Ph.D. Dissertation

Wireless Sensor Networks in the Future Internet of Things: Density, Mobility, Heterogeneity and Integration

Marc Barceló Lladó

Thesis Advisors: José López Vicario and Antoni Morell Pérez
Department of Telecommunications and Systems Engineering,
Escola d'Enginyeria,
Universitat Autònoma de Barcelona (UAB).

September, 2015

To my parents and my brother,

"They did not know it was impossible, so they did it."

- Mark Twain

Abstract

Wireless sensor networks (WSNs) are expected to revolutionize the way we live, work, and interact with the physical environment. Although WSNs have been in the spotlight of the research community for the last decade, their performance in practical implementations is still far behind the theoretical results. This is mainly due to the practical issues that arise in real-life scenarios. As a result, WSNs are generally limited to simple environmental sensing applications. The aim of this thesis is to reduce the gap between the theoretical and real potential of WSNs, and therefore increase their integration in society. In particular, this thesis focuses on the following four practical obstacles: high node density, node mobility, traffic heterogeneity and integration into the future Internet of Things (IoT). First, we deal with the interference problem in high density sensor deployments. We address this issue proposing a pragmatic joint routing, transmission power control and channel allocation approach, built upon the well-known RPL (Routing Protocol for Low-Power and Lossy Networks). This reduces the average packet collisions and the energy consumption of WSNs. Second, we address the low communication reliability and robustness in WSNs with mobile nodes. In particular, we propose a solution that combines RPL with a position-based routing approach based on Kalman filtering. This provides the efficiency and reliability of RPL, and also includes mobility support for non-static nodes. Third, we study the problem of QoS (Quality of Service) provisioning in WSNs managing heterogeneous traffic. With this in mind, we propose a multi-tree approach based on the construction of multiple RPL Instances. This constructs multiple virtual topologies to address the particular requirements of each traffic flow individually. Finally, we focus on the efficient integration of wireless sensors into Cloud-based IoT platforms. In particular, we propose a formulation to orchestrate the resource utilization of the whole network, taking advantage of the recent advances in virtualization and mobile cloud computing. This optimizes the overall consumption, considering the capabilities and limitations of each node, while satisfying the service requirements and the individual users' demands.

Acronyms

API	Application Programming Interface
AODV	Ad Hoc On-demand Distance Vector
bps	bits per second
BS	Base Station
C-RPL	Cooperative RPL
CSDP	Cloud Service Distribution Problem
CSMA-CA	Carrier Sense Multiple Access - Collision Avoidance
CTP	Collection Tree Protocol
DAO	Destination Advertisement Object
DIO	DODAG Information Object
DIS	DODAG Information Solicitation
DODAG	Destination Oriented Acyclic Graph
DSR	Dynamic Source Routing
EO	End Office
ETX	Expected Transmissions
GBR	Gradient-Based Routing
GEAR	Geographic and Energy Aware Routing
GPS	Global Positioning System
HO	Head Office
HR	Hierarchical Routing
IEEE	Institute of Electrical and Electronics Engineers

IETF	Internet Engineering Task Force
ILP	Integer Linear Programming
IO	Intermediate Office
IoT	Internet of Things
KP-RPL	Kalman Positioning RPL
LEACH	Low-Energy Adaptive Clustering Hierarchy
LLN	Low-Power and Lossy Network
MAC	Medium Access Control
MANET	Mobile Ad-hoc Network
MaxPDR	Maximum PDR
MinAP	Minimum Aggregated Power
MIPS	Million of Instructions Per Second
MP2P	Multipoint-to-Point
OF	Objective Function
pdf	probability density function
P2P	Point-to-Point
P2MP	Point-to-Multipoint
PDR	Packet Delivery Ratio
PRR	Packet Reception Rate
PL	Path Loss
QoS	Quality of Service
RPL	Routing Protocol for Low-Power and Lossy Networks
ROLL	Routing Over Low-Power and Lossy networks
SDN	Software-Defined Networking
SDP	Service Distribution Problem
SHM	Structure Health Monitoring
RSSI	Receive Signal Strength Indicator
TSCH	Time-Slotted Channel Hopping
VANET	Vehicular Ad-hoc Network

vFSN	Virtual Fog Service Network
WSN	Wireless Sensor Network
WSSN	Wireless Sensor Surveillance Network

Contents

Chapter 1

Introduction

1.1. Motivation

Wireless sensor networks (WSNs) have attracted much attention of researchers during the last decade. Despite their variety, all sensor networks have certain fundamental features in common. These are composed of low cost devices that are networked via low power wireless communications, to which traditional sensors (e.g., mechanical, thermal, biological, chemical, optical, magnetic) may be attached to measure physical phenomena. These sensor nodes, which may be deployed randomly all over the sensing area, collaborate among themselves to reach a common goal. Some of the benefits of wireless sensing versus wired approaches are: lower installation and maintenance costs, increased flexibility, broader sets of applications, and larger deployments.

Since the very beginning, this technology promised to revolutionize the way we live, work, and interact with the physical environment. In the present, wireless sensors are being deployed on roads, in cities, forests, factories, houses, and many other scenarios. This large scale sensing platform enables the user to have access to a wide range of applications, particularly in the fields of positioning, distributed detection and monitoring. Some examples are environmental monitoring [Rao15], industrial automation [Gun13], agriculture [San15], natural disaster prediction [Arj15], homeland security [Bar13b], smart buildings [Sur15], healthcare [Zha14], transportation [Hu15] and structure health monitoring [Liu15].

Besides the advantages of wireless sensors, their limited communication, computing, storage, and energy resources, has motivated the research community to consider WSNs as a new paradigm in contrast with general data networks. In particular, the communication constraints (e.g., 250 kbps in IEEE 802.15.4) narrow the feasible solutions to those that do not need to use the radio resources extensively. The computational constraints of commercial wireless sensors (e.g., typical microcontroller speed around 8-48 MHz) do not allow strategies demanding heavy computation. The memory limitation (e.g., typically around 128-512 KB) restrict the use of strategies that need significant storage resources. Finally, their limited battery capacity (e.g., typical battery resources provide around 27 kJ) limit their duty cycle. On the other hand, the well known wireless channel dynamics severely affect them, due to their low transmission power (e.g., around 3 dBm), the traffic congestion at their frequency bands, and their low-gain omnidirectional antennas.

The particularities and constraints of WSNs have inspired a vast literature to exploit this technology as efficiently as possible. Since wireless sensors may be deployed in inaccessible locations in which it is not possible to replace or recharge their batteries for practical reasons, the research community has mainly focused on reducing their energy consumption [Pan14]. It is well known that the radio transceiver is the most energy demanding part of the wireless sensor [Wan06]. Therefore, energy efficient communications are imperative to increase the network lifetime of WSNs. The most relevant mechanisms to reduce their energy consumption are:

- Sleep scheduling [Guo12]: The wireless sensor radios generally have four operation modes: transmission, reception, idle and sleep. Idle, transmission and reception modes consume a similar amount of energy. Therefore, sleep scheduling techniques put nodes into sleep mode (i.e., radio off) whenever possible, since it consumes much less energy.

- Energy efficient MAC (Medium Access Control) protocols [Hua12]: Their objective is to avoid collisions so that two interfering nodes do not transmit at the same time, and therefore avoid wasting energy resources with retransmissions.
- Data-driven approaches [Jia15]: Their goal is to reduce the amount of bits transmitted by reducing the data redundancy. These take advantage on the spatial and temporal correlation of measurements. Some examples are data prediction, data compression and data aggregation.

At the beginning, WSNs where planned to be dedicated systems, where highly specialized protocols were used within the WSN, and vendor-specific gateways were used to provide network connectivity with the IP world. Therefore, gateways and sensors often have to be from the same vendor in order to be compatible [Ish13]. Recently, the RPL (Routing Protocol for Low-power and lossy networks) standard has been proposed by the IETF (Internet Engineering Task Force) ROLL (Routing Over Low power and Lossy networks) working group to bring IPv6 to WSNs, so that sensor nodes can be natively addressed and connected through the IP protocol with the rest of the Internet. As a result, a new breed of WSN services and applications are expected in the near future, since they are anticipated to be a key element of the future Internet of Things (IoT). In this thesis, we analyze which are the key issues that need to be addressed to enable these services and propose solutions to throw some light onto these problems. In particular, we visualize future WSN deployments as highly dense, mobile, multi-purpose networks, perfectly integrated into the rest of the IoT. Then, we focus on each of these features throughout this thesis.

1.1.1. Node Density

The node density of wireless sensor deployments is expected to remarkably increase, particularly in urban scenarios. Note that larger WSNs, probably from different providers, are expected to coexist in the same sensing areas [Xia13]. Therefore, it is crucial to reduce the interferences among sensors, without decreasing the network throughput.

Although some commercial wireless sensors have multi-power and multi-channel capabilities [Atm09], which certainly alleviate the problem of interferences among sensors, sometimes these cannot be efficiently exploited due to practical issues, such as hardware or software incompatibilities.

1.1.2. Node Mobility

Thanks to the reduction of size and weight of commercial wireless sensors, these will be attached to mobile entities more easily. However, communication protocols generally assume that WSNs are static [Pan14], and hence most of these strategies are not practical in scenarios with mobile nodes. Note that in this case the network must dynamically adjust itself to variations on the distances among nodes, sudden obstacles, disabled nodes, and so on. This introduces additional challenges that need to be considered in order to enable robust communications in practical scenarios [Pen10].

1.1.3. Traffic Heterogeneity

Although current wireless sensors are generally application-specific nodes developing simple tasks, the increasing processing capabilities of commercial sensors will enable more advanced services, which may combine multiple independent tasks. Moreover, wireless sensors are envisioned as infrastructure resources, which means that they may run multiple third-party applications on-demand [Mit12]. Note that each application has its own particularities in terms of data generation (e.g., continuous, intermittent), traffic pattern (e.g., intermittent, periodic, bursty) and QoS requirements (e.g., lifetime, reliability, delay) [Wan10]. Therefore, the management of the traffic generated by each application must be addressed individually.

1.1.4. Integration into the Internet of Things

The advances in mobile computing and software defined networking, enable WSNs to be connected with the Cloud and also be part of the Internet of Things (IoT). Then, the extensive processing resources of the Cloud can be combined with the ubiquity of wireless sensors. Moreover, the data sensed by different WSNs may be shared on a bigger scale, enabling a more efficient exploitation of the physical infrastructure [Mis14b]. This will extend the capabilities of WSNs far beyond sensing and tracking applications, particularly in smart cities, grids and transportation networks. However, the heterogeneity of devices and services complicates the efficient orchestration of these platforms.

In this thesis, we address the above mentioned challenges from a pragmatic perspective, taking into account not only the hardware limitations of wireless sensors, but also the issues that may arise with their implementation in real-life WSNs. Moreover, the compatibility of the proposed solutions with the existent networking standards is also a major requirement. With this in mind, we design them to be compatible with the well-known RPL protocol, which is an energy efficient protocol that has been already standardized by the IETF (Internet Engineering Task Force).

1.2. Outline and Research Contributions

In this section, we outline the contents of the thesis and the major contributions. *Chapter 1* discloses the main challenges that need to be addressed to enable the forthcoming WSN applications and their integration in society (See Figure 1.1). *Chapter 2* illustrates the background for the following chapters. *Chapter 3* introduces a pragmatic joint routing, power control and channel allocation approach to enable higher density deployments. *Chapter 4* introduces a robust routing strategy for WSNs with both static and mobile nodes to deal with the issues that arise in real-life WSNs with mobile nodes. *Chapter 5* introduces a multi-tree routing approach to efficiently manage multiple traffic types

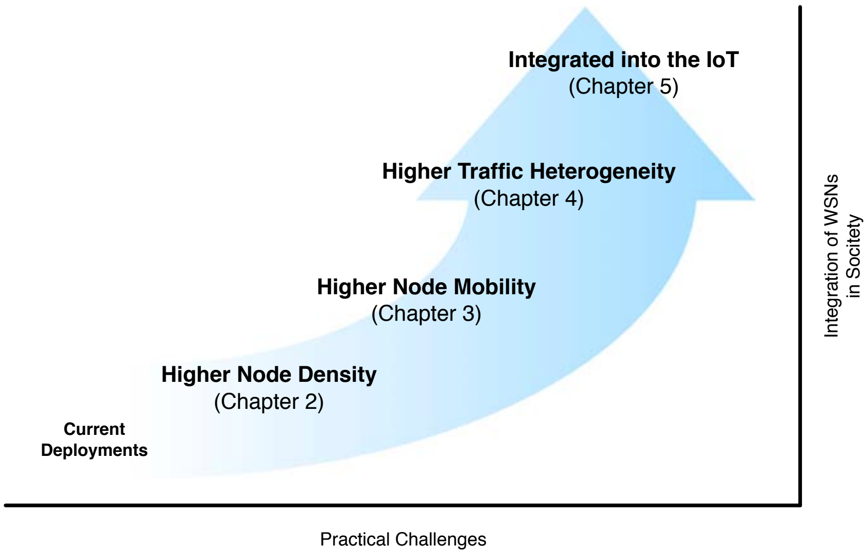


Figure 1.1. Growth of the integration of wireless sensor networks in society thanks to the development of practical strategies.

in a single WSN. *Chapter 6* introduces a minimum cost flow formulation for IoT-Cloud platforms to address the integration of WSNs into the future IoT. Finally, *Chapter 6* concludes the thesis and prospects our future work. In the following, we discuss the details of the contributions.

Chapter 3 - Joint Routing, Channel Allocation and Power Control to Support High Density Deployments

Nowadays, commercial wireless sensors have multi-power and multi-channel capabilities. These effectively reduce the network consumption and avoid the high collision probability that may arise in convergecast networks. However, practical issues arise in cross-layer implementations. In this chapter, we enhance RPL to obtain a joint routing, transmission power control and channel allocation solution for real-life WSNs. Two different strategies (MinAP and MaxPDR) are designed and implemented in a WSN testbed with commercial motes. We compare these strategies with the original RPL and also with other standardized routing protocols using simulations. We also provide experimental results to show that the network performance is improved both in terms of reliability and energy consumption.

The contributions of this chapter have been included in a journal article:

- Barcelo, M.; Correa, A.; Vicario, J.L.; Morell, A., "Joint routing, channel allocation and power control for real-life wireless sensor networks", in Transactions on Emerging Telecommunication Technologies, January 2014.

This has been inspired by the previous conference publications:

- Barcelo, M.; Correa, A.; Vicario, J.L.; Morell, A., "Joint routing and transmission power control for Collection Tree Protocol in WSN", IEEE International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC), September 2013.
- Barcelo, M.; Vicario, J.L.; Seco-Granados, G.; Puig, J.M.; Laborda, J.M., "Multichannel routing algorithm for cluster-tree wireless sensor networks in aerospace applications", IEEE Fly By Wireless (FBW), June 2011.

Chapter 4 - Position Assisted Routing to Support Node Mobility

Mobility is still one of the greatest challenges in WSNs, since energy efficient routing strategies are generally designed for static networks. In particular, RPL has been proven to be very efficient in static WSNs. However, its slow response to topology changes and its huge signalling cost to keep up-to-date routes in the presence of mobile nodes makes it inefficient in mobile scenarios. In this chapter, we introduce KP-RPL (Kalman Positioning - RPL), a novel routing strategy for WSNs with both static and mobile nodes. The objective of KP-RPL is to provide robust routing, taking into account the issues that arise in real-life WSNs with mobile nodes (e.g., sudden obstacles, interferences, estimation errors). This extends RPL with a new metric for mobile nodes that combines Kalman positioning and blacklisting. The simulation results show that the reliability and the robustness of the network in bad channel conditions is enhanced, compared to existent approaches. Moreover, thanks to the Kalman filter, the minimum rate that is necessary for positioning is reduced and the network lifetime extended.

The contributions of this chapter have been submitted for journal publication in:

- Barcelo, M.; Correa, A.; Vicario, J.L.; Morell, A.; Vilajosana, X.,"Position Assisted RPL using Kalman Filtering in Wireless Sensor Networks with Mobile Nodes", (submitted to) IEEE Sensors Journal, 2015.

A preliminary position-aware routing protocol was included in the following conference publication:

- Barcelo, M.; Correa, A.; Vilajosana, X.; Vicario, J.L.; Morell, A., "Novel routing approach for the TSCH mode of IEEE 802.15.4e in wireless sensor networks with mobile nodes", IEEE Vehicular Technology Conference (VTC), September 2014.

Chapter 5 - Multi-Tree Routing to Manage Heterogeneous Traffic

Advanced WSN applications may need to develop multiple tasks that involve sensing, processing and gathering data from different sensing units. This heterogeneous data may have multiple and sometimes opposite sets of requirements. In these scenarios, different objective functions must be combined, and therefore traditional single-tree routing approaches are not efficient. On the contrary, RPL virtually splits the network into multiple RPL Instances, which transport each kind of data according to its particular requirements. However, this protocol does not define any mechanism to decide the nodes that must belong to each instance, and this decision has a strong impact in the network energy consumption and performance. With this in mind, in this chapter we introduce C-RPL (Cooperative - RPL), which creates multiple instances following a cooperative strategy among nodes with different sensing tasks. As a result, the energy consumption, the complexity and the cost of the nodes is reduced compared to RPL, since they are active less time, perform fewer tasks and are equipped with less sensing hardware. In this chapter, we also propose a novel fairness analysis for networks with multiple instances, showing that C-RPL achieves a better tradeoff in terms of performance and energy consumption than RPL.

The contributions of this chapter have been submitted for journal publication in:

- Barcelo, M.; Correa, A.; Vicario, J.L; Morell, A., "Cooperative interaction among multiple RPL instances in wireless sensor networks", (submitted to) Computer Communications, 2015.

Preliminary multi-tree and cooperative multi-tree routing approaches were introduced in the following conference publications:

- Barcelo, M.; Correa, A.; Vicario, J.L; Morell, A., "Cooperative multi-tree sleep scheduling for surveillance in wireless sensor networks", IEEE Military Communications Conference (MILCOM), November 2013.
- Barcelo, M.; Correa, A.; Vicario, J.L; Morell, A., "Multi-tree routing for heterogeneous data traffic in wireless sensor networks", IEEE International Conference on Communications (ICC), June 2013.

Chapter 6 - IoT-Cloud Formulation to Integrate WSNs into the Future Internet of Things

The confluence of the Cloud and the Internet of Things (IoT) enables a new breed of applications that create real-time valuable information via the analysis of live data from a wide range of sensing devices. Wireless sensors are planned to be a key element in this new paradigm, and therefore their efficient integration is essential. In this chapter, we address this as a minimum cost mixed-cast flow problem. This is mathematically formulated using only linear constraints and solved via integer linear programming. We refer to this formulation as the IoT-Cloud SDP (Service Distribution Problem), since it finds the placement of virtual service functions over an IoT-Cloud platform. We focus on energy consumption as the major driver of todays network and cloud operational costs.

The heterogeneous set of IoT-Cloud network components are characterized according to their associated sensing, compute and/or transport capacity and energy efficiency. We solve the IoT-Cloud SDP for an illustrative set of smart city services, in which users provide and request information using their smart devices, showing that this is a flexible solution to efficiently orchestrate IoT-Cloud networks.

The contributions of this chapter are included in a journal article that is being prepared for publication:

- Barcelo, M.; Correa, A.; LLorca, J.; Vicario, J.L; Morell, A.; Tulino, A., "IoT Cloud Service Optimization in Smart Cities". (Manuscript in preparation).

The SDP was first formulated for cloud networks in the following conference publication:

- Barcelo, M.; Llorca, J.; Tulino, A.; Narayan, R., "The Cloud Service Distribution Problem in Distributed Networks", IEEE International Conference on Communications (ICC), June 2015.

Additional contributions

Besides the research contributions covered in this thesis, some other topics related to WSNs have also been considered during this period.

The study of a PCA-based data aggregation technique has been submitted for publication in the following journal article:

- Morell, A.; Correa, A.; Barcelo, M.; Vicario, J.L., "Data Aggregation and Principal Component Analysis in WSNs" (submitted to) IEEE Wireless Communications, 2015.

The contributions related with the design of positioning systems based on Kalman filtering are included in the following journal and conference publications:

- Correa, A.; Barcelo, M.; Morell, A.; Vicario, J.L., "Enhanced inertial-aided indoor tracking system for wireless sensor networks: A review," IEEE Sensors Journal, September 2014.
- Correa, A.; Barcelo, M.; Morell, A.; Vicario, J.L., "Distance-based tuning of the EKF for indoor positioning in WSNs," European Signal Processing Conference (EUSIPCO), September 2014.
- Correa, A.; Morell, A.; Barcelo, M.; Vicario, J.L., "Navigation system for elderly care applications based on wireless sensor networks," European Signal Processing Conference (EUSIPCO), August 2012.

The results regarding the exploitation of multiple receivers around the body for indoor positioning systems have been published in the following conference publication:

- Correa, A.; Barcelo, M.; Morell, A.; Vicario, J.L., "Indoor pedestrian tracking system exploiting multiple receivers on the Body" International Conference on Indoor Positioning and Indoor Navigation (IPIN), October 2014.

Finally, an extension of this article including additional details and experimental results has been submitted for journal publication in:

- Correa, A.; Barcelo, M.; Morell, A.; Vicario, J.L., "Indoor pedestrian tracking with on-body multiple receivers" (submitted to) IEEE Sensors Journal, 2015.

Chapter 2

Background

2.1. Summary

In this chapter, we provide a background overview of the most relevant topics in the following chapters. First, we briefly introduce gradient-based routing with a focus on the RPL protocol, since the routing strategies presented in Chapters 3-5 are built upon this protocol. Second, we introduce the multi-channel and multi-power approaches, which are effective solutions to deal with packet collisions in high density WSNs (Chapter 3). Third, we present position-based routing as an efficient mechanism to deal with the mobility of nodes (Chapter 4). Fourth, we discuss the practical challenges that appear in WSNs managing heterogeneous traffic, and also a general classification of the most relevant kinds of traffic in WSNs (Chapter 5). Finally, we introduce cloud computing and motivate the transition to fog computing, which is a key concept to efficiently integrate wireless sensors into the future IoT (Chapter 6).

2.2. Gradient-Based Routing

In this section, we motivate the use of gradient-based routing in general, and the RPL protocol in particular, in front of other routing approaches.

WSNs are mainly used for monitoring purposes. Therefore, the communication in these networks is frequently many-to-one (i.e., the base station collects data from many sensor nodes). In this scenario, tree-based routing is extensively used because it can reduce the amount of sending data by using data aggregation techniques. Many strategies can be used to create a routing tree. In general, the most interesting approaches can be classified in clustering protocols, position-based routing protocols and self-organizing coordinate protocols [Wat11]. Clustering protocols, such as low-energy adaptive clustering hierarchy (LEACH) [Hei00], impose a structure to balance the energy consumption of nodes. Each node determines in each round whether it becomes a cluster head or a leaf node using a stochastic algorithm. Then, leaf nodes transmit to cluster heads, which aggregate and compress the data and forward it to the sink. These techniques assume that all nodes can communicate with a cluster-head, but the accuracy of this assumption depends on the transmission ranges of the wireless sensors and the particular scenario. As a result, the performance of the network may be reduced due to the use of unreliable links. On the other hand, position-based routing, such as geographic and energy aware routing (GEAR) [Yu01], constructs the routes according to the position of nodes. This reduces the complexity of the routing process, but the obtention of the geographic information may not be possible. Moreover, the inevitable positioning errors reduce the network performance [Pen11]. In contrast, self-organizing coordinate protocols, such as gradient-based routing (GBR) [Man01], do not need to estimate the position of the nodes. This avoids using GPS chips, indoor positioning systems, or programming the position of each node manually. This last category, and more precisely the gradient-based approach, has been found the most suitable mechanism for convergecast WSNs by the IETF, through the ROLL working group [Wat11].

Gradient-based routing is based on the transmission of control messages (also referred to as pilots or beacons) to estimate the link qualities. According to the link quality estimator, the network can be optimized in terms of reliability, energy consumption, latency, bandwidth, and so on. GBR is the classical gradient routing protocol, but there exist more advanced gradient-based implementations, such as CTP (Collection Tree Protocol) [Gna09], RPL (Routing Protocol for Low-Power and Lossy Networks) [Win12] and the hierarchical routing defined in ZigBee [Cuo07]. CTP is a routing protocol specially designed for relatively low traffic rates. It is widely used in practical WSN implementations. In fact, CTP has been included in TinyOS 2.x and also in OMNET++. For this reason, it has been widely used in the recent years for both research [Vil10] and commercial products. For instance, CitySee and GreenOrbs are using this protocol for ecological surveillance purposes, and Powernet for energy consumption monitoring. RPL is a more recent protocol that provides a similar performance in terms of reliability, energy consumption and protocol overhead. In addition to the features of CTP, this can successfully communicate with IP devices in the greater Internet using IPv6 addresses [Ko11]. Therefore, we have adopted this protocol in the design of our routing algorithms.

2.2.1. RPL: Routing Protocol for Low-Power and Lossy Networks

RPL is an energy efficient and reliable protocol designed for (LLNs) Low-Power and Lossy Networks, such as WSNs. This protocol constructs convergecast trees, referred to as DODAGs (Destination Oriented Acyclic Graphs), with one or multiple roots that act as sinks, according to a predefined objective function (OF). The OF defines how RPL nodes select and optimize routes, and should be defined according to the network requirements. This also defines the concept of RPL Instance that groups one or multiple DODAGs with a common OF (See Figure 2.1). For the sake of simplicity, in this thesis we assume that each RPL Instance includes only one DODAG, and each DODAG has a single sink.

According to the particular OF defined for each DODAG, the nodes compute their relative distance to the sink, referred to as Rank [Vil14]. This metric is also used to avoid and detect loops in order to ensure that packets make forward progress within the DODAG, since it must monotonically decrease towards the DODAG destination. The typical OF in RPL is based on the minimum number of expected transmissions

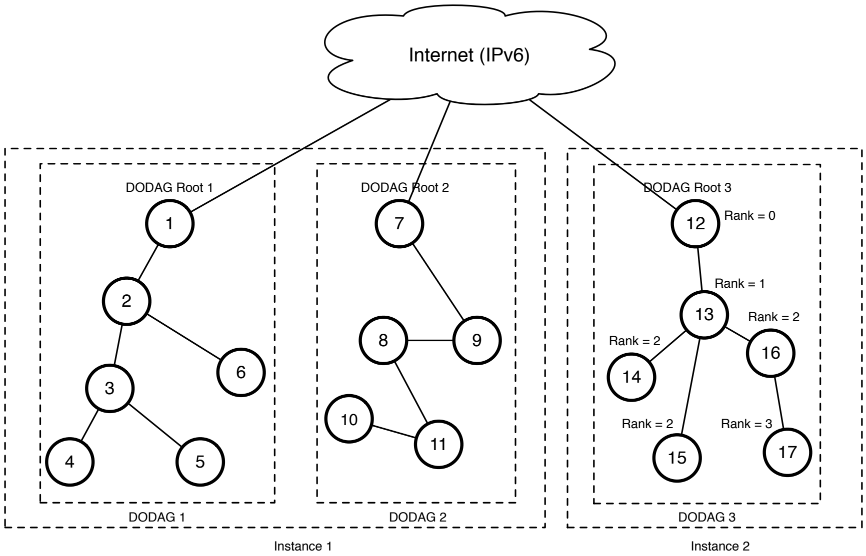


Figure 2.1. A RPL network with three DODAGs in two instances.

(ETX). Basically, this metric indicates how many times a message must be transmitted on average to reach its final destination. For instance, the i-th anchor node estimates its end-to-end ETX, when forwarding to the j-th anchor node, as the ETX to reach the j-th anchor node ($ETX_{i,j}$), plus the end-to-end ETX of this node (ETX_j), that is:

$$ETX_i = ETX_{i,j} + ETX_j, \tag{2.1}$$

where the ETX between two nodes is computed as the reciprocal of their packet delivery ratio (PDR). Then:

$$ETX_i = \frac{1}{PDR_{i,j}} + ETX_j. \tag{2.2}$$

RPL defines three different control packets: DODAG Information Object (DIO), DODAG Information Solicitation (DIS) and Destination Advertisement Object (DAO):

- DIO packets are sent in broadcast and propagate Ranks in order to construct and maintain the DODAG. A DIO packet includes information that allows a node to discover a RPL Instance, to learn its configuration parameters, and also to select a DODAG parent set. The most relevant fields in a DIO packet are the 128-bit

DODAGID, which uniquely identifies a DODAG using an IPv6 address, the 8-bit *RPLInstanceID*, which identifies the instance in the network, and the 16-bit *Rank*. The rate of DIOs is adjusted dynamically by a trickle timer according to the network needs [Lev11].

- DIS packets are sent in unicast to solicit DIOs from a neighbour node, for instance to discover nearby DODAGs. When a node receives a DIS packet from another node, it must unicast a DIO packet to this node.

- DAO packets are sent in unicast upwards along the DODAG to the selected parent in storing mode, or to the DODAG root in non-storing mode [Gad12]. This is a mechanism to collect information about the network topology.

Multiple kinds of traffic are supported in RPL: multipoint-to-point (MP2P), point-to-multipoint (P2MP), and point-to-point (P2P). MP2P is the traffic pattern of applications collecting information from multiple sources at a single location. This is the most common traffic in WSNs. P2MP traffic is generated in applications that disseminate information from a single node to multiple nodes. P2P traffic is also supported to enable nodes to communicate among themselves.

In terms of security, RPL supports message confidentiality and integrity. In particular, RPL has three basic security modes: unsecured, preinstalled and authenticated. In the first mode, RPL control packets are sent without additional security mechanisms. In the second mode, RPL Instances have preinstalled keys that are applied to generate secured RPL messages. In the third mode, nodes may join a RPL Instance as a leaf, but they need to obtain a preinstalled key from an authentication authority to act as a router.

The interested reader is referred to [Win12] for further details on RPL and illustrative examples.

2.3. Realistic Multi-Channel and Transmission Power Control

In this section, we introduce the multi-channel and multi-power strategies, which define the frequency channel and the transmission power of wireless sensors, respectively. These have been used in Chapter 3 to both reduce the energy consumption and alleviate the packet collision problem in highly dense WSNs.

2.3.1. Multi-Channel Control

Channel allocation has been extensively studied in the literature to increase the bandwidth of single channel MAC protocols by parallelizing transmissions among neighboring nodes [Hua12]. In single channel WSNs under heavy load, many messages are lost due to collisions among wireless sensors [Jov11]. However, using multiple channels, the potential interferences are avoided assigning different channels to nodes within a distance of two-hops (See Figure 2.2). Moreover, network throughput can potentially be increased. Some examples of multi-channel MAC protocols are MMSN [Zho06], TMCP [Wu08] and MC-LMAC [Dur08].

In practical implementations, several issues arise with many multi-channel protocols [Sai14]. Some of them involve heavy computation that cannot be implemented in complexity constrained nodes, such as wireless sensors [Haj11]. In other cases, they require additional hardware resources, such as multiple transceivers [Li11]. Moreover, it is also important to take into account the additional overhead that these protocols introduce, which may be very high. Due to these issues, most of multi-channel protocols cannot be implemented in practice. It is also important to mention that node-independent channel allocation may not be feasible in tree-based networks, since nodes would be required to adapt their reception channel for each son node. Note that channel hops can be a major source of transmission errors due to channel mismatches and time synchronization errors,

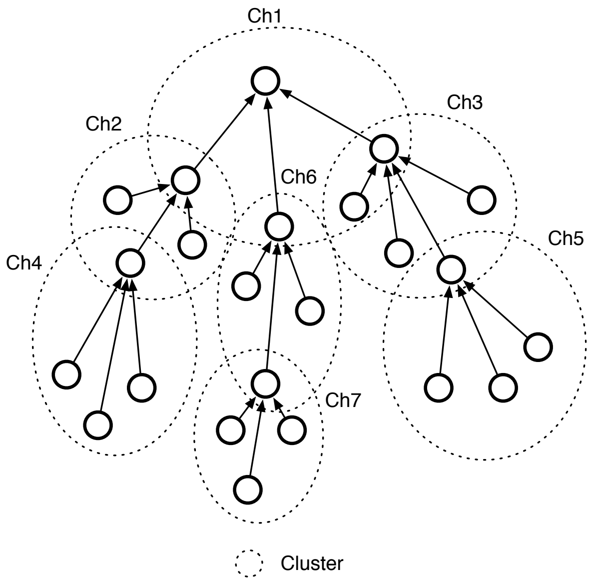


Figure 2.2. Example of a channel assignment scheme in WSNs that allocates a single transmission and reception channel to each node using clusters.

since both nodes should be set on the same channel at the same time [Hua12].

Summarizing, multi-channel protocols for WSNs must not require complex processing, should not introduce significant overhead, and should not enforce particular hardware requirements in order to make possible its implementation in commercial motes. Moreover, it is recommended to allocate the channels according to the network topology in order to benefit from the node distribution and avoid frequent channel changes.

2.3.2. Transmission Power Control

The purpose of transmission power control techniques is mainly twofold: i) limiting multiuser interference to increase single-hop throughput, and ii) reducing power consumption to prolong battery lifetime. On one hand, the interferences among nodes should be minimized to reduce collisions and increase the throughput. On the other hand, since the radio transceiver is the main source of energy consumption in WSNs, the transmission power should be defined according to the current needs. Some examples of power control strategies proposed in the literature are presented in [Bra09], [Meg11], [Fu12].

The reliability of a link is closely related to the received signal strength [Ram06]. According to the channel conditions, the transmitter should increase or decrease the transmit power, either to increase the signal strength at the receiver, or to reduce the energy consumption and channel interferences with other nodes (See Figure 2.3). It is also important to consider that additional routes can be discovered using a higher transmit power, which could be more energy efficient than the routes already discovered. On the other hand, a reliable link reduces the number of average retransmissions due to unreceived packets, decreasing the channel utilization of that link and therefore potential collisions. Therefore, in practice the transmit power should not be adjusted without considering routing information. Moreover, the complexity and the additional overhead of the transmission power assignment should also be considered. Finally, the particular range and granularity of power levels has also a relevant impact on the efficiency of these strategies [Cot14], and this may vary in each commercial platform.

Therefore, a pragmatic transmission power control scheme for WSNs should take into account the particular hardware characteristics. Moreover, it should be decentralized and simple to reduce the communication overhead and spend the minimum computing resources.

2.4. Position-Based Routing

In this section, we introduce position-based routing that is an efficient solution for mobile WSNs, highlighting the impact of realistic conditions in its performance. In Chapter 4, we have combined this technique with gradient-based routing to provide robust and reliable routing in WSNs with mobile nodes.

Several WSN applications require the combination of static nodes, also referred to as anchors, and nodes attached to mobile entities (e.g., people, equipment, goods). Some examples are elderly health monitoring, animal tracking, search and rescue and vehicular networks. However, providing robust and reliable routing in this kind of networks

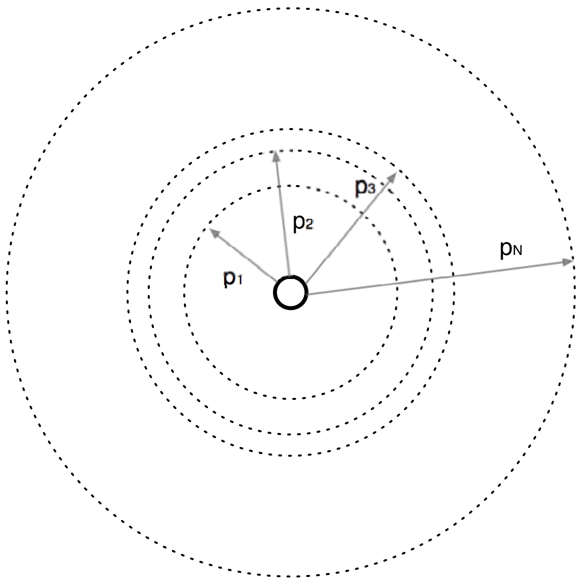


Figure 2.3. Transmission range of a wireless sensor using different transmit power levels (assuming omnidirectional antennas).

is still one of the greatest challenges in WSNs [Cad13]. Note that additional issues must be considered, since mobility may lead to frequent node disconnections, and a high signalling cost is required to create and repair routes. Moreover, this information may be frequently out of date at the moment of taking the routing decisions, leading to inefficient routing decisions [Gad14]. Routing strategies based on the position of nodes, referred to as position-based routing or geographical routing, alleviate these problems by eliminating the need for topology maintenance [Kar00]. Since routing decisions are not based on routing tables, they do not need to exchange and maintain routing information, reducing the memory requirements of nodes, and also the reaction time to sudden topology changes [Sar14].

The main assumption that relies on this protocols is that sensors are aware of their position, the position of their neighbors, and the position of the destination. The position of anchor nodes can be assumed to be error free (i.e., sophisticated positioning systems can be used, since their position has to be set only once). However, the position of mobile nodes is unknown, and therefore it has to be frequently estimated. Since GPS (Global

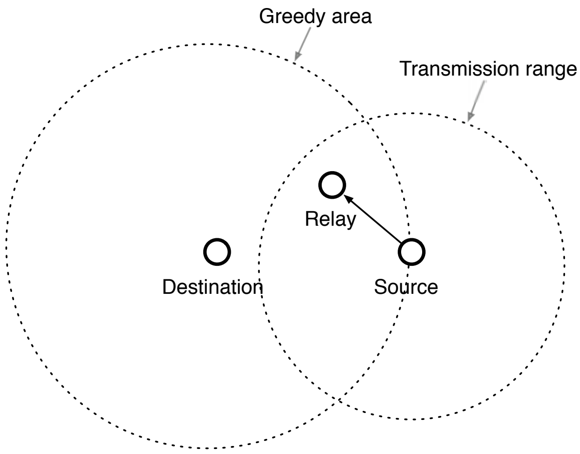


Figure 2.4. Greedy packet forwarding.

Positioning System) positioning may not feasible in WSNs, due to the absence of a GPS module or to GPS reception problems, indoor positioning systems are frequently used. Then, sensors broadcast their position to the rest of nodes.

The classical geographic routing algorithm uses a greedy strategy that forwards packets to the neighbour that is closer to the destination (See Figure 2.4). Then, in a multi-hop scheme, the packets advance towards the destination in each hop. This avoids the creation of loops by strictly forwarding packets to nodes located closer to the destination than themselves. However, since forwarding decisions are locally optimal, these may not lead to globally optimal paths. On the other hand, nodes may not find any candidate neighbor, resulting in a dead-end node. Then, recovery routines must be applied to avoid these situations. Face routing is an alternative to greedy routing in order to guarantee delivery [Kim05]. This uses the concept of planarization, which constructs planar sub-graphs that contain no intersecting edges. However, it has a low energy efficiency and it is not efficient in networks that are not located in planar surfaces. This motivates hybrid greedy-face routing [Kuh03], which generally use greedy forwarding and switch to face routing when nodes cannot find any candidate neighbour (i.e., any node that is closer to the sink than itself).

Most of position-based routing strategies, assume accurate location information, but positioning systems have a limited accuracy that depends on the technology and the environment. On the other hand, the so-called unit disk graph model is generally used, assuming that nodes may reach any node located at a distance equal or lower than its transmission range. However, in real implementations the transmission ranges are not uniform, the antennas are not omnidirectional, the links are not bidirectional, and the reliabilities of links are not boolean. Both positioning [Pen11] and transmission model [Zam08] inaccuracies should be considered by the routing algorithm, since these cannot be completely avoided. Note that these have a strong impact in the network reliability, particularly in the presence of mobile nodes.

2.5. Management of Heterogeneous Traffic

In this section, we illustrate the problem of managing multiple kinds of traffic in a WSN. Moreover, a general classification of the main kinds of traffic in WSN applications is provided. We have addressed the problem of considering the QoS requirements of heterogeneous traffic in Chapter 5.

With the increasing computational capacities of wireless sensors, a higher number of WSN applications will combine multiple tasks, and different kinds of measurements, providing more advanced and complex services. This generates multiple traffic flows coexisting at the same network, each of them with its particular QoS requirements (See Figure 2.5). Moreover, these requirements may even change over time in order to satisfy the users' demands. For instance, SHM (Structure Health Monitoring) [Har10] systems may need to collect information coming from different sensing units, such as pressure, vibration or temperature. Moreover, they also need to send alarm messages in case of broken sections or systems failures. In addition, continuous messages are broadcast for external monitoring and calibration. This heterogeneity should be taken into account to meet the service requirements, as well as for the proper utilization of the limited

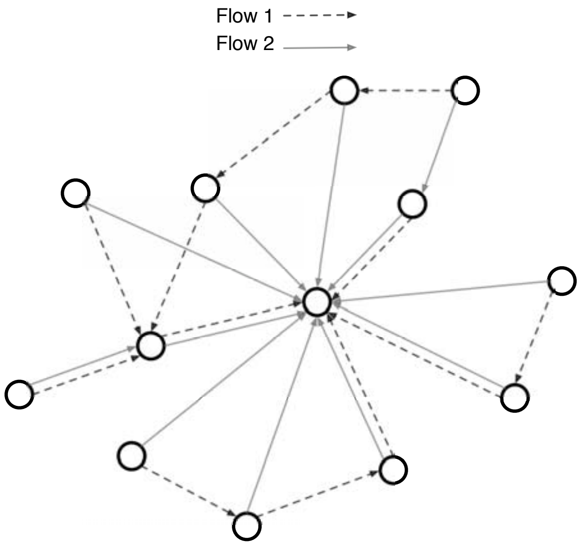


Figure 2.5. Example of a network with two convergecast trees.

resources of wireless sensors.

Traditional routing approaches are generally focused on a particular QoS metric, such as delay, reliability, bandwidth, security, energy consumption, and so on. In order to consider multiple QoS metrics, multi-objective routing [Alw13] combines them in a single metric. However, due to the contradictory relationship among the QoS requirements, it is not always possible to find a solution that satisfies all the parameters [Mie03]. Therefore, these strategies look for a balanced solution among the different objective functions. However, this may not be feasible in applications imposing hard-QoS constraints, such as a minimum delay or a maximum end-to-end reliability. In this case, data packets must be categorized into different levels or classes and forwarded accordingly [Ari13]. In convergecast networks, multi-tree routing schemes are an efficient mechanism to provide this differentiation at the network layer [Bar13a]. Multi-tree routing algorithms may provide congestion control, load balancing and QoS differentiation, creating an independent tree for each traffic flow according to its particular requirements. Then, each node has a different parent assigned for each tree in the network.

The RPL protocol considers the multi-tree solution to deal manage heterogeneous traffic.

In particular, this protocol creates multiple logically independent RPL Instances that may run concurrently. Each instance may be optimized with a particular objective function, which should be correlated with the traffic requirements. The identifier of the instance is defined in the *RPLInstanceID* field of the data and control packets. Nodes may belong to multiple RPL Instances, and they could act as a router in some instances, and as a leaf node in others, but they can only join to one DODAG per instance. However, it is worth noting that RPL does not specify how to create and operate with them, and this is set for future specifications [Win12].

2.5.1. Data Traffic Classification

Although a low energy consumption, a high reliability and a reduced latency is always desired, each type of data has its particular requirements. Moreover, the transmission pattern may also be different (e.g., bursty transmissions, periodic transmissions, or in some cases without transmissions during days or months). Nevertheless, the main kinds of data can be grouped in event detection, non-critical monitoring and critical monitoring [Wan10].

2.5.1.1. Event Detection Traffic

In event detection tasks, it is crucial to receive the information as soon as possible in order to turn on the actuators or send an alarm. Therefore, a reduced delay is the main requirement of this kind of traffic. Since nodes do not need to send information if the event has not been detected, this traffic is generally isolated. On the other hand, the transmission consumption is not critical since these occur just occasionally. Furthermore, note that the reliability is neither as important as delay since many packets can be sent in the case of alarm. Then, it is enough to receive one of these packets, since all of them contain the same information. Although the latency of the network may depend on many factors, such as the sleeping strategy or the transmission scheduling, we assume that the latency in multi-hop networks is proportional to the number of hops, since the packet must be decoded, processed and coded again in each hop. However, in a particular implementation the delay metric should be adapted to consider the particularities of the MAC layer.

2.5.1.2. Non-Critical Monitoring Traffic

Certain tasks, such as ambient conditions monitoring, require periodic transmissions from every node to the sink. In this case, the delay and the reliability are not critical. Instead, the energy consumption becomes the most critical parameter, since they require periodic data transmissions. For instance, temperature, light or pressure measurements, are highly correlated in time and space. The measurements lost during the transmission can be easily estimated using the past measurements (i.e., time correlation) or even with the measurements coming from other motes (i.e., space correlation). Therefore, the routing strategy should prioritize the energy consumption in this case.

2.5.1.3. Critical Monitoring Traffic

Some applications require a high reliability. This is the case of applications managing critical data, such as positioning systems or vital signs monitoring. In these applications, it is essential to assure that as many packets as possible reach the sink. Moreover, their traffic pattern strongly depends on the application requirements. For instance, in a positioning application, the mobile node should continuously transmit information as long as the mobile entity is moving. It is easy to notice that the performance of these applications is strongly related to the network reliability, since each packet contains relevant information. Therefore, the routing algorithm should provide the maximum packet reliability.

2.6. From Cloud to Fog Computing

In this section, we introduce cloud computing and also motivate the transition from cloud computing to fog computing in order to move processing closer to the end-users. In Chapter 6, we address the integration of WSNs into Cloud-based IoT platforms taking into account the advantages of fog computing.

2.6.1. Overview of Cloud Computing

Cloud computing is a cutting edge technology that makes use of a shared pool of configurable computing resources that can be rapidly provisioned and released to store and process data in third-party servers. Then, personal devices become interfaces to powerful data centers. This model reduces the necessity of people and companies of buying dedicated hardware using a pay-as-you-go model. The main advantages of cloud computing are infrastructure cost reduction, greater flexibility, elasticity and optimal resource utilization. Then, major companies, such as Amazon, Google, IBM, Cisco and Alcatel-Lucent have invested in cloud computing and offer cloud-based solutions for individuals and businesses.

Cloud computing offers many different services, which form the cloud computing stack (See Figure 2.6). They describe how cloud services are made available to clients. These can be categorized in [Asl12]:

- IaaS: Infrastructure as a Service. This is the lowest level of the stack. It provides infrastructure components to clients, such as virtual machines, servers, networks and firewalls. Amazon Web Services is a large IaaS provider.
- PaaS: Platform as a Service. This layer provides application programming interfaces (APIs) to clients to interact with databases and web servers, which automatically scale according to the users' requirements. Google AppEngine is a popular PaaS provider.

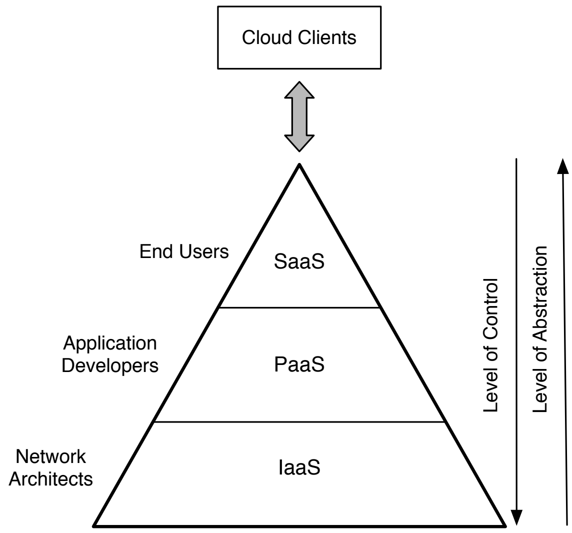


Figure 2.6. Cloud computing stack.

- SaaS: Software as a Service. This provides complete online software solutions already developed, such as virtual desktops or social media platforms. SaaS is currently a huge market and still has a strong growth potential. According to Forrester's annual industry outlook, global SaaS software revenues are forecasted to reach $106 billions in 2016.

In practice, most of current cloud services are a combination of these services, and therefore the line between them is becoming blurred.

2.6.2. Distributed Cloud Computing

Already, connected devices have reached 9 billion, and this number is expected to grow more rapidly and reach 24 billion by 2020 [Gub13]. Internet traffic will soon be dominated by the consumption of resource and interaction intensive cloud services and applications from resource-limited communication end points. Moreover, the number of latency-sensitive applications, such as real-time services, is also growing. Both end user experience and overall network efficiency will push for a fundamental transformation of todays centralized network and cloud architectures towards a highly distributed converged cloud-network platform, composed of a large number of cloud nodes distributed across an increasingly meshed converged metro-core [Bar15]. In this massively distributed cloud network, and with the introduction of network functions virtualization (NFV) and software-defined networking (SDN), virtual cloud service functions can be dynamically and elastically instantiated over commodity servers at multiple cloud locations close to the end users and interconnected via a programmable network fabric. This way, a cloud network operator can host a variety of services and applications over a common distributed physical infrastructure, reducing both capital and operational expenses, while delivering high quality of experience (QoE).

2.6.3. Fog Computing

With this in mind, fog computing, also referred to as edge computing, extends the cloud computing paradigm to the edge of the network. In general, a fog network is a highly virtualized platform, composed of a huge number of heterogeneous devices, that provides computation, storage, and networking services between end devices and traditional cloud computing data centers. Fog nodes may be resource-poor devices, such as set-top-boxes, access points, routers, base stations, smart devices, or low complexity cloud nodes, such as cloudlets [Sat09] or micro-clouds [Shi13], placed at the edge of the network. The integration of end-devices into the Cloud, motivates the definition of additional service models besides IaaS, PaaS and SaaS, such as SAaaS (Sensing and Actuation as a Service), which includes sensors and actuators.

In addition to the benefits of cloud computing, fog computing provides support for latency-sensitive applications, since it allows applications to run as close as possible to the end-users, mobility support and location awareness [Sto14]. Moreover, this reduces the utilization of the cloud resources when these are not strictly necessary, partially storing and processing at edge devices. Some scenarios in which fog computing has a huge potential are smart grids, smart buildings, smart cities, connected vehicles, augmented reality and mobile big data analytics [Bon12], [Yi15].

Although fog computing has several advantages in front of cloud computing, the design and management of fog networks is still a challenging task. In [Yi15], the following issues are identified:

- Network connectivity: Emerging techniques, such as SDN and NFV, are required to provide flexible services that integrate heterogeneous devices, such as smart devices and wireless sensors.
- QoS requirements: The individual capacities of fog nodes and the particular service requirements must be considered.
- Computation offloading: Offloading decisions must react to the network changes as quickly as possible.
- Provisioning and resource management: Efficient resource management models are required to avoid wasting storage, computing and energy resources.
- Other issues, such as accounting, billing, interfacing model and security. These are out of the scope of this thesis.

The identified issues mainly arise due to the heterogeneity of fog nodes and services, and the dynamism of fog networks in terms of nodes, network access and resources.

Chapter 3

Joint Routing, Channel Allocation and Power Control to Support High-Density Deployments

3.1. Summary

The multi-channel and multi-power capabilities of commercial sensors are planned to be very important in future WSN implementations. This is due to the increasing sensor density in some scenarios, such as smart cities. Note that the interferences among sensors reduce the network performance, due to packet losses, and increase its energy consumption, due to packet retransmissions. However, incompatibilities may arise when multi-channel and multi-power strategies are designed regardless of the algorithms applied to the rest of layers. However, the complexity of cross-layer approaches may be too high to be applied in WSNs.

In this chapter, we propose a pragmatic joint routing, transmission power control and channel allocation scheme to address this problem. This is integrated in the well-known RPL (Routing Protocol for Low-Power and Lossy Networks) to make it compatible with this standardized current networking standard. We propose two different approaches: MinAP (Minimum Aggregated Power) and MaxPDR (Maximum Packet Delivery Ratio). The first approach is designed for applications that demand a very low energy

consumption, while the second approach is for applications with high reliability requirements.

3.2. Introduction

3.2.1. Motivation and Previous Work

WSNs are composed of a large number of distributed sensor nodes. These share the same communication channel, and thus interferences and collisions are prone to happen. Since the number of WSN applications is constantly growing, this problem is expected to be very significant in the near future. On one hand, wireless sensors are generally designed to be application specific (i.e., deployed to perform a particular task), and therefore multiple WSNs may need to coexist to provide advanced services [Raw14]. On the other hand, network operators provide their own sensing platform [Mis14b]. As a result, an increasingly number of wireless sensors will coexist, particularly in densely populated scenarios, such as urban areas.

Packet collisions reduce both the performance and the energy efficiency of the network. In order to deal with this problem, many energy efficient communication strategies have been proposed. These can be implemented in the physical layer (e.g., reduce the transmission power), the medium access control (MAC) layer (e.g., reduce the collision probability) or the network layer (e.g., find energy efficient routes) [Sha13].

Concerning the network layer, tree-based routing is extensively used in convergecast networks, (i.e., networks where all traffic is sent to a single node). One of its advantages is the possibility of setting up the network in a completely distributed manner, reducing the traffic overhead compared to centralized approaches. This also allows the reduction of the amount of sending data by using data aggregation and compression techniques. One of the most interesting solutions for the construction of this kind of networks is the gradient-based routing approach [Wat11]. This technique uses control messages to evaluate the quality of the wireless links. The link quality estimator metric may focus on different criteria, such as reliability, number of hops, node battery level, energy consumption, and so on.

On the other hand, concerning the physical and MAC layers, transmission power control and channel allocation strategies can be very effective to increase the network lifetime and reduce the collision probability of the network [Dur12]. Power control strategies adjust the transmission power of each node at the minimum level that guarantees a tolerable reliability at the receiver [Bra09], [Meg11], [Fu12], [Cor08]. On the other hand, multi-channel approaches distribute the frequency resources available [Zha12], [Mor13], [Bac10], [Hua12]. As a result, multiple nodes can transmit in parallel, increasing the throughput, and alleviating collisions and interferences.

The combination of individual solutions at each layer may not be straightforward and could lead to inconsistencies [Bar13c]. Therefore, some cross-layer strategies have been proposed in the literature [ElB11], [Li11], [Luo10]. However, most of these solutions are too complex to be applied in practical WSNs, unless drastic simplifications are considered. Note that the wireless dynamics and the limited hardware capabilities of wireless sensors must be considered in real-life implementations.

In this chapter, we propose a pragmatic cross-layer approach for real-life WSNs. The objective is to design an implementable joint routing, transmission power control and channel allocation solution. Instead of designing an unfeasible cross-layer approach, we enhance the standardized routing protocol RPL (Routing Protocol for Low-Power and Lossy Networks) [Win12]. This protocol has been selected because it considers many important issues that appear in real-life WSNs and it can successfully communicate with devices outside the WSN [Ko11]. More specifically, we propose two different approaches to enhance RPL, MinAP (Minimum Aggregated Power) and MaxPDR (Maximum Probability Delivery Ratio). They modify the link estimator, the path selection and the packet forwarder of RPL, and can be introduced in this protocol without any additional signalling cost.

3.2.2. Contributions

The main contributions of this chapter are:

- We propose a pragmatic cross-layer strategy combining routing, transmission power control and channel allocation for real life WSNs. Although these problems are extensively studied separately, from the best of our knowledge a joint solution for real-life WSNs has not been proposed before.
- We design a metric for applications that prioritize the energy consumption (MinAP), and another for applications with high reliability requirements (MaxPDR). Moreover, the second approach adjusts the tradeoff between the network energy consumption and reliability using a configurable parameter.
- We implement MinAP and MaxPDR in a commercial WSN platform in order to provide experimental results of their performance in a real environment, and compare them with existent solutions.

3.2.3. Organization of the Chapter

The remaining of this chapter is organized as follows: Section 3.3, presents an overview of the proposed joint routing, channel allocation and transmission power control solution. This also introduces the MinAP and MaxPDR metrics and the multi-channel mechanism. Section 3.4, evaluates the performance of the proposed solution using simulations. This compares MinAP and MaxPDR with LEACH, GBR and the hierarchical routing of ZigBee in terms of reliability, energy consumption and collision probability for different scenarios. Section 3.5, provides experimental results of the proposed approach in a commercial WSN platform composed of 13 IRIS motes. Finally, Section 3.6 summarizes this chapter and presents the main conclusions.

3.3. Joint Routing, Channel Allocation and Power Control

The problems of routing, channel allocation and transmission power control are mostly studied separately. This is because the optimal cross-layer solution becomes a complex optimization problem [ElB11]. Since WSNs can only deal with low complexity tasks, a different approach is necessary. We design a pragmatic cross-layer solution built upon RPL. This avoids both the incompatibility issues that may arise with the combination of several individual techniques, and the difficulties that may appear in the implementation of unpractical cross-layer approaches in commercial WSNs.

3.3.1. Overview

The proposed RPL extension modifies the link quality estimation, the path selection and the packet forwarder of this protocol. More precisely, the link estimator considers a different metric. Instead of using an ETX-based metric, two alternative metrics are defined, referred to as MinAP and MaxPDR. The first one focuses on energy consumption and the second one on network reliability. The path selection has also been modified according to the estimates of the two proposed metrics. Finally, the packet forwarder is also modified since packets can be transmitted using multiple transmission power levels and multiple frequency channels. Therefore, the energy consumption and the packet collisions can be reduced. The overall differences between the original RPL and the proposed RPL-based approach are illustrated in Figure 3.1.

Following the structure of RPL, the proposed approach uses DIO packets to compute the Rank of each node. Each node composes a list with potential parents and their corresponding Ranks. According to these Ranks, routes are formed by the most preferred parent of each node. As in RPL, the rate of these packets is dynamically adjusted by the Trickle algorithm [Lev11]. This algorithm reduces the DIOs rate if the topology is

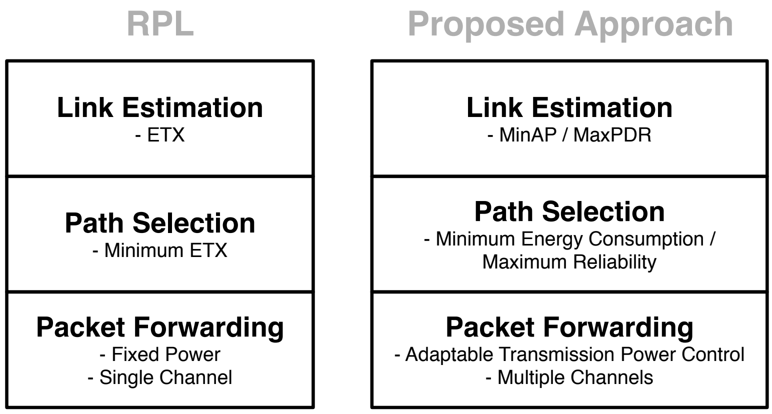


Figure 3.1. Comparison between the original RPL and the proposed approach.

stable, and increases this rate whenever a change in the topology needs to be propagated. However, instead of broadcasting DIOs at a unique transmission power level, they are transmitted at multiple power levels (L_{DIO}), being $L_{DIO} = 0$ and $L_{DIO}{=}L_{max}$ the minimum and maximum available levels, respectively.

The tree construction process also establishes the transmission power and the transmission channel of each node. These two parameters are used to send unicast packets, such as data messages towards the sink, control packets going downward, ACKs (Acknowledgements), and so on. However, DIOs are transmitted in broadcast at multiple power levels, from 0 to L_{max}. On the other hand, these packets are broadcast using a previously defined common channel in order to assure that all nodes are able to listen these packets. Therefore, nodes listen to this channel periodically in order to maintain their Rank and avoid potential loops in the DODAG. In order to schedule the transmission and reception of data and control packets (DIOs and DISs), nodes divide the total time in data and control periods. The time dedicated to each period must be defined according to the particular network requirements (i.e., a longer data period increases the network throughput, and a longer control period increases the adaptability of the network to sudden changes due to interferences, obstacles, and so on). In particular, a control period duration of 5% of the superframe suffices in our implementation, since nodes are static.

New nodes can join the DODAG at any time. To do so, the new node broadcasts a DIS packet using the common channel. Then, nodes receiving this packet will reset their Trickle timer, hence start transmitting DIOs with the lowest interval. Finally, the new node uses these DIO packets to compute the Rank associated with the rest of nodes based on the Objective Function. Note that nodes are continuously broadcasting DIOs at multiple transmission powers. Therefore, new nodes will periodically receive DIOs at the highest power level from each node located at one hop distance.

Since nodes compose a list of potential parents, the process of restoring the tree connectivity due to new obstacles, interferences, node failures and so on, is straightforward. Once a node detects that it has lost the connectivity with its parent, because it does not receive its ACKs, the list of potential parents is updated and the new parent is selected accordingly. Finally, the transmission power level and the transmission channel are also updated.

3.3.2. MinAP: Minimum Aggregated Power

This strategy is mainly focused on reducing the transmission power as much as possible to minimize the energy consumption and reduce the interferences among nodes. Then, DODAGs are constructed using the minimum end-to-end aggregated transmission power. The block diagram of this strategy can be found in Figure 3.2.

In this strategy, each node selects its own transmission power, being B the transmitter and A the receiver ($L_{tx_{A,B}}$), according to:

$$L_{tx_{A,B}} = \lfloor L_{max} \left(1 - PDR_{A,B}\right) \rfloor, \tag{3.1}$$

where $PDR_{A,B}$ is the link PDR between nodes A and B, that is defined as:

$$PDR_{A,B} = \frac{R_{A,B}}{P_B}, \tag{3.2}$$

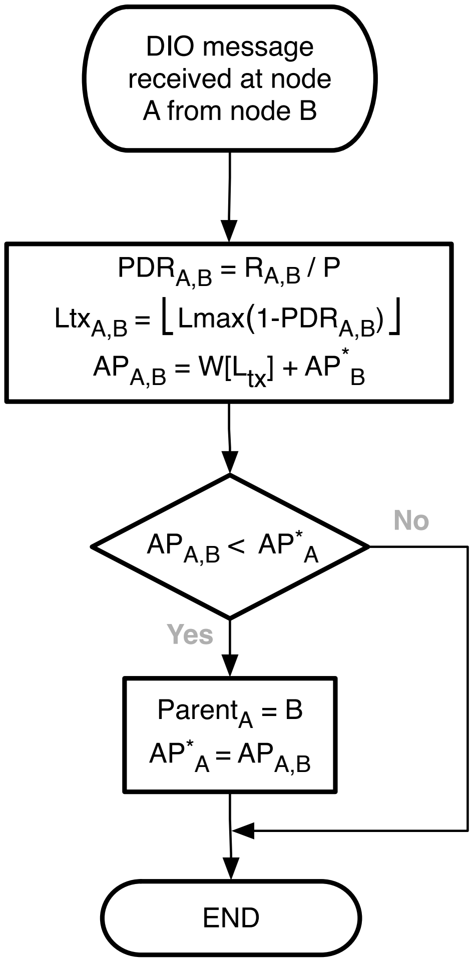


Figure 3.2. Block diagram of MinAP.

where $R_{A,B}$ is the number of DIO messages correctly received at node A from node B, and P_B is the total number of DIO messages sent by B.

The aim of this power selection mechanism is to select the minimum transmission power to ensure reliable reception of packets. According to (3.1), the transmission power of node A to node B decreases with the number of DIOs correctly received from B. For instance, if node A has received the 50% of DIOs sent by node B, it selects its transmission power level as $L_{max}/2$ (or the level below), assuming that it has received all DIOs sent with a higher transmission power level. Note this assumes the high correlation among the PDR of a link with the RSSI measured at the receiver shown in [Ram06]. This particular mechanism has been proposed due to its simplicity and compatibility with the RPL protocol.

According to the radio transceiver specifications, each transmission power level ($L_{tx_{A,B}}$) has an associated value in Watts ($W_{A,B}$). Since these equivalences are known, the total aggregated power (AP) from node A to the sink, using node B as the first relay (AP_A^B), can be computed as:

$$AP_A^B = \sum_{i \in p_{A,B}} W_i, \qquad (3.3)$$

$$p_{A,B} = \{l_1, l_2, ..., l_S\}, \qquad (3.4)$$

where $p_{A,B}$ is the path from A to the sink, using B as the first relay, and $l_1, l_2, ..., l_S$, are the set of links included in this path, being l_1 the link from A to B and l_S the last one to reach the sink. However, the knowledge of all this information in A would require a lot of signalling. Alternatively, this is computed in a decentralized manner as follows:

$$AP_A^B = W_{l_1} + AP_B^*, \qquad (3.5)$$

where AP_B^* is the aggregated power of the best path from B to the sink, and it is broadcast by B in its control messages. Note that this is important to reduce the communication overhead.

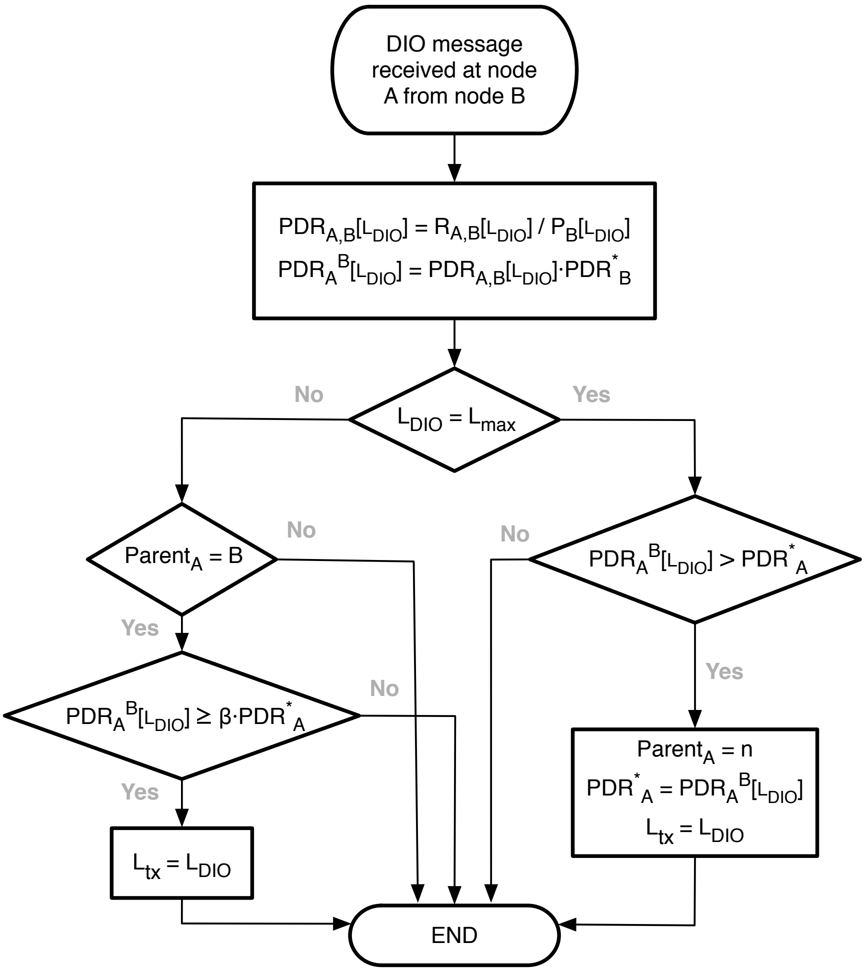


Figure 3.3. Block diagram of MaxPDR.

Finally, node A defines its parent as the node that provides the lowest aggregated power value (AP^*_A).

3.3.3. MaxPDR: Maximum Packet Delivery Ratio

This strategy is mainly focused on maximizing the network reliability. In this case, nodes select the path with the highest end-to-end PDR. Then, the transmission power of each node is reduced as much as possible according to a configurable parameter β. The block diagram of this strategy can be found in Figure 3.3.

In multi-hop networks, the path from each node to the sink must be considered in order to estimate the reliability of the network. In this strategy, only the DIO messages sent using $L_{DIO} = L_{max}$ are considered to define the routes. This has been considered due to the highest reliability of RSSI measurements at higher transmission powers. The rest of DIOs are used to reduce the transmission power of wireless sensors as much as possible. To do so, a different PDR value is calculated for each transmission power level.

The end-to-end PDR from A using B as the first relay (PDR_A^B) is:

$$PDR_A^B = \prod_{i \in p_{A,B}} PDR_i, \tag{3.6}$$

$$p_{A,B} = \{l_1, l_2, ..., l_S\}, \tag{3.7}$$

For each transmission power level, this can also be calculated in a decentralized manner as follows:

$$PDR_A^B[L_{DIO}] = PDR_{l_1}[L_{DIO}] \cdot PDR_B^*, \tag{3.8}$$

where PDR_B^* is the PDR associated with the best path from B to the sink, and it is broadcast by B. Finally, the node with the highest end-to-end PDR (PDR_A^*) is set as the parent of A.

Finally, if node B has been selected by A as its parent, node A defines its transmission power as the lowest one that fulfills the following condition:

$$PDR_A^B[L_{DIO}] \geq \beta \cdot PDR_A^*, \tag{3.9}$$

where β is a threshold parameter that can be adapted, from 0 to 1, according to the reliability requirements. This adjusts the willingness of nodes to reduce their transmission power. Note that high β values are more demanding in terms of reliability, whereas low values tend to relax it.

3.3.4. Multi-Channel Control

The objective of multi-channel strategies is to increase the spectral efficiency of the network in order to reduce the high collision probability that may arise in single-channel tree networks.

In this section, we present a channel allocation process that uses the signalling of the routing process. Note that the scope of the chapter is not to define a new multi-channel MAC protocol. Instead, the objective is to include a channel assignment mechanism in RPL without increasing its signalling cost. Moreover, potential issues that may arise with the combination of multiple algorithms in the MAC and network layers are avoided.

The proposed method distributes the frequency resources using the cluster structures automatically created in the routing process, which are composed by a single parent node and all its child nodes. In order to avoid time-frequency synchronization errors, a single frequency channel is associated to each cluster (See Figure 2.2). In particular, the transmission channel (TxCh) of each node is selected according to the reception channel (RxCh) of its parent node:

$$TxCh = RxCh\left[Parent\right]. \tag{3.10}$$

From the DIO packets sent by neighbour nodes, each node knows which reception and transmission channels have been selected by those nodes. This is valuable information, since these are the channels that are used by nodes within a distance of two hops, and they must be avoided to reduce network collisions. With this information all the nodes in the network define their own Channel Utilization vector($ChUt$) as:

$$ChUt = \{u_1, ..., u_k, ..., u_F\}, \tag{3.11}$$

where each value corresponds to the number of neighbour nodes that use each of the F parallel channels. Using this vector, RxCh is selected as the channel with the lowest number of nodes associated:

$$RxCh = \arg\min_{k}(ChUt). \tag{3.12}$$

At the end of the training process, each node has two channels assigned (TxCh and RxCh). In order to reduce time-frequency synchronization errors, non-leaf nodes remain in RxCh unless they need to transmit their own information or to relay information coming from other nodes.

Figure 3.4 summarizes the routing, transmission power control and channel allocation processes.

3.4. Simulation Results

This section evaluates the performance of the proposed methods and compares them with the original RPL [Win12] in terms of energy consumption, reliability and collision probability. For the sake of comparison, we also include LEACH [Hei00], GBR [Man01] and the hierarchical routing used in ZigBee [Cuo07] (for simplicity referred here to as ZigBee) (See Section 2.2). Simulations are conducted in Matlab.

The scenario consists of N sensing nodes transmitting temperature measures through the DODAG to a single sink node. They are randomly deployed on a LxL m^2 surface, and the sink is situated in the middle of the square. Data aggregation has not been considered, hence each packet is transmitted independently at each hop. Nodes operate at the 16 different channels defined by the IEEE 802.15.4-2003 standard [IEE11] in the 2.4 GHz frequency band. They use channel 11 to broadcast and receive DIOs and the rest of channels (12 to 26) to send data and other unicast messages. A total of 200 realizations are averaged. In each realization, nodes send their temperature measurements to the sink through the network.

First, a collision-free multiple access scheme has been considered to evaluate the re-

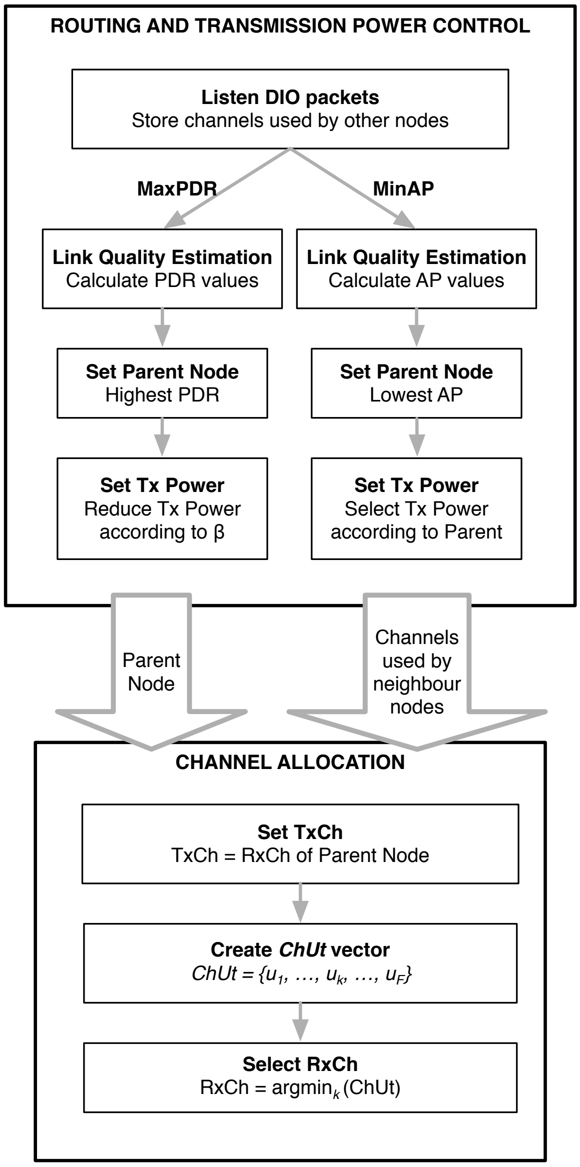


Figure 3.4. Overview of the routing, transmission power control and channel allocation processes.

liability and the energy consumption of each strategy. After that, we have used a CSMA-CA scheme to show the collision probability reduction due to the multi-channel approach.

The path loses (PL) are modeled using the one slope log-distance path loss model [MS05], which is commonly employed in current simulation frameworks (e.g., ns-2):

$$PL(dB) = PL_0(dB) - 10\alpha log(d) + \gamma, \tag{3.13}$$

where PL_0 is the path loss at the reference distance d_0 (1 m), α is the path loss exponent, d is the communication distance, and γ is zero mean Gaussian noise with variance σ^2 that models the attenuation caused by flat fading. In these simulations, the following physical parameters are assumed, σ^2=6 [Rap01], α=3 [Pu12] and PL_0=50 dB (empirically measured using IRIS motes). Moreover, the sensitivity value (i.e., -91 dBm) and the multiple transmission power levels of the RF230 transceiver [Atm09] (i.e., 16 levels from -17.2 dBm to 3 dBm) are considered.

3.4.1. Reliability

Figure 3.5 compares the reliability in terms of end-to-end PDR, considering 30 nodes and different area deployment dimensions. In these simulations, the threshold parameter of MaxPDR (β) is set to 1. The impact of this parameter is evaluated in Section 3.4.3. The results show that MaxPDR, with and without transmission power control, obtains the highest reliability, even higher than the original RPL and ZigBee. Note that these construct the routes using end-to-end ETX estimates, while MaxPDR uses PDR estimates. For instance, in a deployment area of 80x80 m^2, MaxPDR achieves a PDR of 0.97 (0.98 without power control), while RPL and ZigBee obtain 0.85 and 0.92, respectively. On the other hand, both GBR and MinAP present a lower PDR in the same situation (i.e., 0.61 and 0.68, respectively). This is because these construct the routes based on the number of hops and the aggregated power, respectively. The

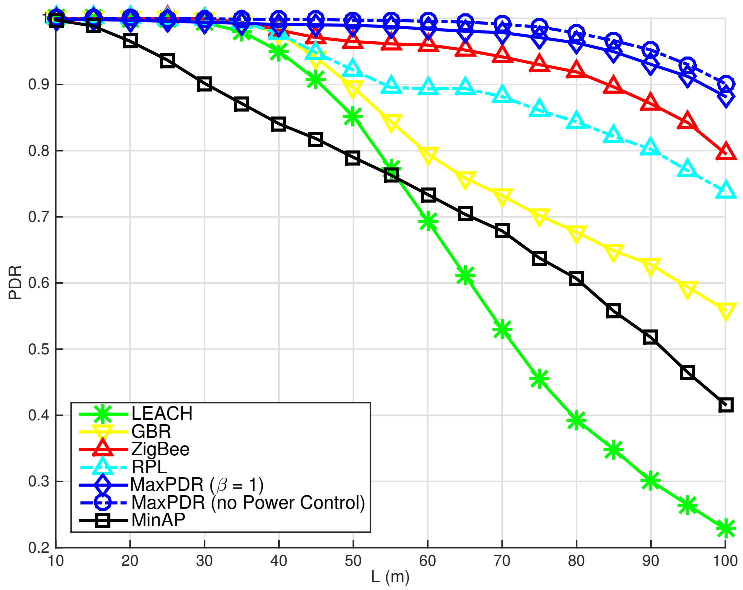


Figure 3.5. Reliability in terms of PDR for different deployment areas.

poor performance of LEACH in large deployment areas is because it considers that all nodes can communicate with each other. Therefore, cluster-heads are not selected according to the particular scenario. This is the reason why gradient routing approaches generally have a higher reliability than clustering approaches, particularly in large-scale deployment areas.

3.4.2. Energy Consumption

Figure 3.6 presents the average aggregated power of 30 nodes for different deployment areas. From this figure, it can be observed that the gradient-based strategies adjust their power consumption according to the scenario, but LEACH does not. In most cases, MaxPDR with β=1 and MinAP present a lower energy consumption than GBR, ZigBee and RPL thanks to their transmission power control mechanism. In fact, MinAP can reduce the aggregated power of RPL in more than 2.5 mW. The transmission power control is particularly useful in small deployment areas, since the average aggregated

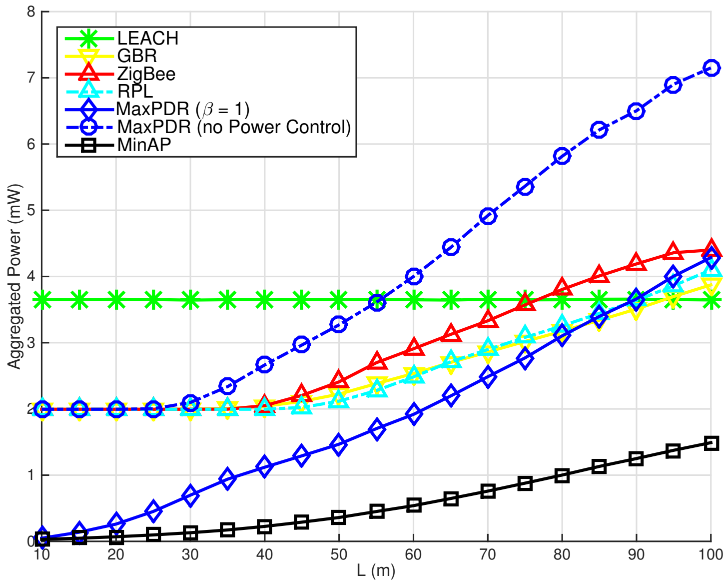


Figure 3.6. Energy consumption in terms of aggregated power (mW) for different deployment areas.

power is 99% lower.

In Figure 3.7, the aggregated power is shown for different network densities. The sensing area is 100x100 m^2. While GBR, ZigBee , RPL and MaxPDR with β=1 show a similar consumption, it is interesting to observe that MinAP has a considerably lower slope compared with the rest of strategies. Thus, its energy consumption is substantially much lower, particularly in high density networks. For instance, MinAP reduces the aggregated power of RPL more than 75% in the case of networks composed of 100 nodes. Note that the aggregated power of MaxPDR without power control is very high. This is because the PDR metric is very demanding in terms of energy consumption, since it finds the most reliable routes without considering their energy consumption associated. Hence, it is very important to reduce this consumption by adjusting the transmission power of wireless sensors.

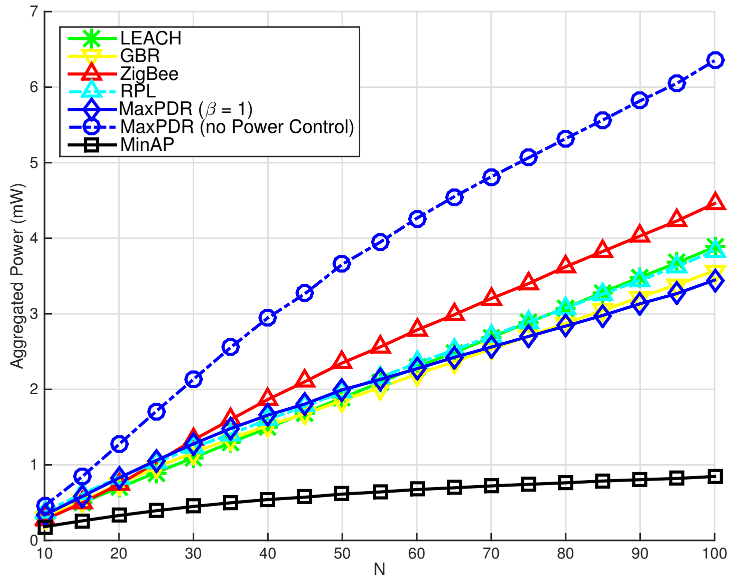


Figure 3.7. Energy consumption in terms of aggregated power (mW) for different node densities.

3.4.3. Reliability vs. Consumption Tradeoff

In convergecast networks, the packets relayed toward the sink may need several transmissions. This increases the energy consumption and reduces the reliability of the network. On one hand, the energy consumption can be reduced by lowering the transmission power of each hop. On the other hand, a high transmission power increases the reliability of each link and also reduces the number of hops to reach the sink. Consequently, there is a tradeoff between these two parameters, since both should be considered.

In Figure 3.8, the PDR vs. aggregated power of each strategy is represented in the particular case of 30 nodes deployed in a 100x100 m^2 area. The confidence intervals (90%) of each value are included. Note that these intervals are large because nodes are randomly deployed. In this figure, multiple β values (from 0.1 to 1) have been considered for MaxPDR. Unfortunately, MaxPDR does not achieve a higher reliability than MinAP using equal or less aggregated power. However, it is important to highlight that MaxPDR increases the flexibility of the network, since this can be adjusted according to the specific

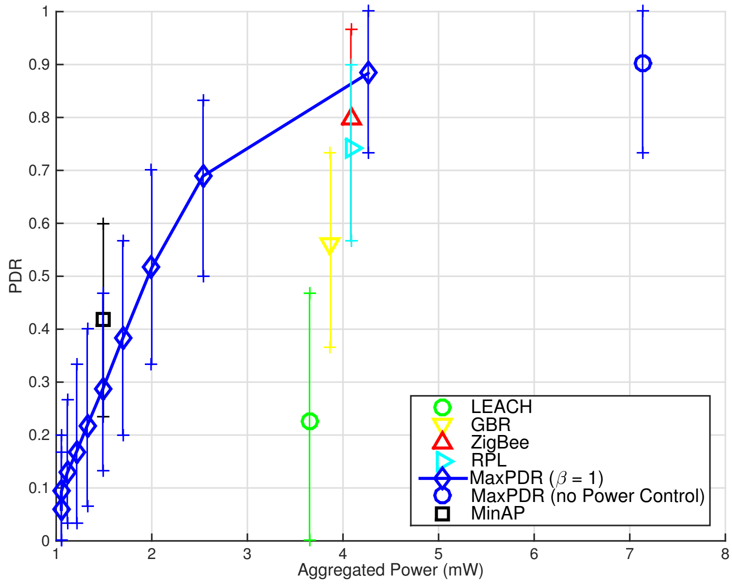


Figure 3.8. Reliability vs. energy consumption tradeoff in terms of PDR vs. aggregated power (mW).

network requirements of each application using β. We can observe that for β values lower than 1, MaxPDR obtains a lower PDR than RPL and ZigBee, but its aggregated power is much lower. For instance, using β=0.9 the PDR of MaxPDR is 6% and 12.5% lower, respectively, while the aggregated power is 39% lower.

3.4.4. Collision Probability

In Figure 3.9, the collision probability of MaxPDR and MinAP is calculated with and without the proposed channel allocation mechanism. In addition, RPL, GBR and ZigBee, and a multi-channel version of these, have also been included to illustrate the problem of single-channel networks. Multiple transmission probabilities have been considered to represent different network traffic patterns, (i.e., a transmission probability of 1 means that nodes transmit continuously). A total of 30 nodes are randomly deployed in a 100x100 m^2 area.

The results show that the single-channel versions have a very high collision probability,

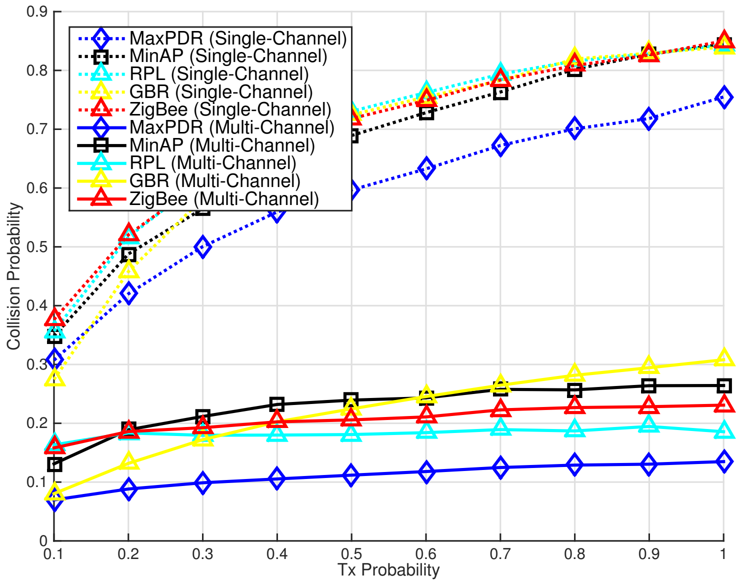


Figure 3.9. Collision probability in MaxPDR for different transmission probabilities.

especially in scenarios with high throughput requirements. Note that using a multi-hop approach, a single packet may require several transmissions to reach the sink. Consequently, the collision probability is increased and the network throughput is reduced. On the other hand, using a multi-channel approach the collision probability becomes much lower. In fact, using MaxPDR or MinAP with the proposed channel allocation, the collision probability is always below 12%. As a result, the collision probability can be reduced up to 85%, compared with the original RPL, by adding this strategy. Note that the remaining collisions mostly come from the multi-channel hidden terminal problem. Therefore, they could be avoided assuming an alternative MAC protocol instead of CSMA-CA [Bac10].

3.5. Experimental Evaluation

In this section, MinAP and MaxPDR are implemented in commercial wireless sensors and evaluated using a WSN testbed. The objective is to prove their implementability and evaluate their performance in a real life application. In the version of MaxPDR without power control, the minimum, medium and maximum transmission power levels have been used.

3.5.1. WSN Testbed

The WSN testbed is composed of 13 IRIS motes from MEMSIC [MEM] (See Figure 3.10), which are compatible with the IEEE 802.15.4 standard, distributed as shown in Figure 3.11. IRIS motes are equipped with the Atmel ATMega 1281 8-bit microcontroller [Atm14], 8 KB of RAM and the RF230 transceiver [Atm09]. This operates at the 2.4 GHz band, it uses up to 16 channels, and its transmission power range is adjustable from -17.2 dBm to 3 dBm and divided in 16 steps. This transceiver has two operating modes: the basic mode, and the extended mode. In our implementation, the basic mode has been chosen in order to better observe the impact of each strategy, since the extended mode includes additional features that are out of the scope of this chapter (e.g., automatic address filtering, automatic acknowledgement). In particular, transmissions are scheduled according to a simple time synchronized mechanism based on the ID of each node. Therefore, the proposed approach is also compatible with more advanced time synchronized mechanisms.

Each sensing mote transmits temperature and light measurements to the sink node every second. Nodes have been deployed without any line of sight to favor multi-hop transmissions. Moreover, the network has been located in a region with interferences coming from other systems operating at the same frequency band, such as WiFi and Bluetooth.

3.5.2. Experimental Results

The mean values of PDR and aggregated power obtained using each implementation are shown in Table 3.1. These results indicate that MaxPDR provides the maximum

Figure 3.10. IRIS mote [MEM].

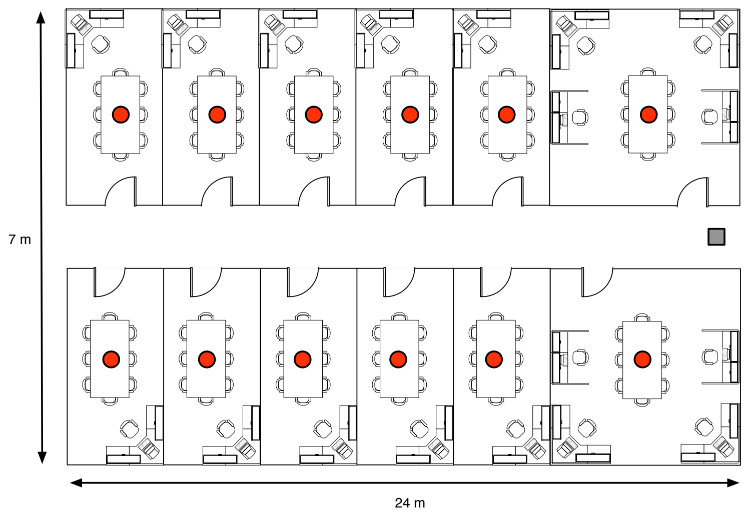


Figure 3.11. WSN testbed composed of 12 sensing motes (red spots) and a sink mote (grey square).

Table 3.1. Experimental Results. PDR and aggregated power (mW).

	MaxPDR (no Power Control)			**MaxPDR**
	P_{max}	P_{med}	P_{min}	(β=1)
PDR	0.95	0.89	0.50	0.94
AP (mW)	4.489	1.256	0.049	0.805

	MinAP	**RPL**	**ZigBee**
PDR	0.65	0.91	0.92
AP (mW)	0.077	2.539	2.826

reliability when the maximum transmission power is used (i.e., 0.95). However, this is reduced as the transmission power is reduced, obtaining 0.89 and 0.5 using the medium and minimum levels, respectively. On the other hand, applying the power control with β=1, a high degree of reliability is also achieved (i.e., 0.94). In fact, this reliability is even higher than the reliability obtained by RPL and ZigBee (i.e., 0.91 and 0.92, respectively), confirming the simulation results presented in Section 3.4. Moreover, MaxPDR with β=1 also has a low aggregated power, reducing more than 3 times the aggregated power of the original RPL, and more than 5.5 times compared with MaxPDR without power control.

Obviously, the lowest aggregated power is found using MaxPDR with the minimum transmission power. In contrast, it shows poor performance in terms of reliability. On the other hand, MinAP has also a very low aggregated power (97% lower than the original RPL), and it adjusts the transmission powers according to the scenario. As a result, this strategy is the best option for applications with very low energy consumption requirements, and no strict reliability constraints. For instance, this is the case of periodic environment measuring applications, such as temperature or light monitoring. The reason is that these kinds of measures are highly correlated in time and space. Thus, unreceived measures can be efficiently estimated using previous information or even using measures received from other nodes.

3.6. Summary and Conclusion

In this chapter, we address the node density problem in WSNs, enhancing the widely implemented RPL routing protocol. We propose a pragmatic joint routing, channel allocation and transmission power control solution for real life WSNs. Two different routing metrics are proposed, referred to as MinAP and MaxPDR, which combine reliability and energy efficiency criteria. Both include a power control mechanism to adapt the transmission power according to the channel conditions and the QoS requirements. This uses the DIO packets broadcast in RPL to not increase the overall signalling cost. Moreover, a simple multi-channel mechanism is also included. This assigns a single pair of channels for transmission and reception to each node to avoid time-frequency synchronization errors. MinAP and MaxPDR have been compared using simulation and experimental results with the original RPL, and also with LEACH, GBR and the gradient-based routing protocol adopted in ZigBee.

The simulation results show that the proposed approach increases the reliability, and reduces the aggregated power and the collision probability of RPL. More specifically, MaxPDR can increase by more than 10% its PDR and MinAP can reduce its aggregated power up to 99%. It is also worth noting that MaxPDR can be adapted using the parameter β to deal with the reliability vs. energy consumption tradeoff. The experimental results show that MaxPDR achieves a PDR of 0.94, compared to 0.91 for the original RPL, and it also reduces its aggregated power by 68%. On the other hand, MinAP reduces its aggregated power by 97%, being a very interesting solution if the energy consumption of the network is more important than its reliability. These results have verified the implementability of the proposed RPL-based approach and confirmed the promising results obtained through simulations.

Chapter 4

Position Assisted Routing to Support Node Mobility

4.1. Summary

In the near future, commercial wireless sensors will be easily attachable to mobile entities, such as people, wild animals, vehicles, and so on. However, node mobility adds significant challenges to the network routing due to the frequent topology changes. Therefore, routing strategies designed for static WSNs are not efficient in these scenarios. On the other hand, the routing protocols designed for mobile WSNs generally assume perfect positioning information, but the positioning accuracy is limited, particularly in indoor environments. Therefore, practical strategies are necessary for WSNs with mobile nodes to further enhance their integration in society.

In Chapter 3, we introduced a RPL-based routing approach for energy efficient routing in WSNs. The results have shown that this is a reliable and efficient routing strategy for WSNs with static nodes. Unfortunately, the slow response to topology changes of RPL and its high signalling cost to keep up-to-date routes in the presence of mobile nodes makes it not appropriate for mobile scenarios. In this chapter, we further extend RPL by introducing KP-RPL (Kalman Positioning - RPL), a novel routing strategy for WSNs with both static and mobile nodes. The objective of KP-RPL is to provide

robust routing, taking into account the issues that arise in real-life WSNs with mobile nodes. This combines RPL with Kalman positioning and blacklisting to develop a new RPL metric. The simulation results show that the reliability and the robustness of the network in harsh conditions is enhanced, compared to existent approaches. Moreover, thanks to the Kalman filter, the rate of packets for positioning can be reduced to extend the network lifetime.

4.2. Introduction

4.2.1. Motivation and Previous Work

Wireless sensor networks are planned to be dynamic network infrastructures, combining static and mobile nodes. However, the implementation of wireless sensor networks in scenarios with mobile nodes is particularly challenging [Cad13]. This is because the mobility of nodes deteriorates the network performance due to the continuous changes in the network topology [Zam08]. Since energy efficient routing strategies are generally designed for static WSNs [Pan14], they are not practical in mobile scenarios without significant changes. Moreover, the additional issues that appear in real-life scenarios (e.g., obstacles, interferences, estimation errors), reduce the robustness of the communications [Pen11].

The problem of mobility in wireless networks has been extensively studied in mobile ad-hoc networks (MANETs). Some examples are the protocols ad hoc on-demand distance vector (AODV) [Per99] and dynamic source routing (DSR) [Joh01]. However, routing algorithms designed for MANETs are not recommended for WSNs due to their low energy efficiency [Gad06]. One of the main reasons is that they consider many-to-many routes, while communications in WSNs are generally many-to-one. The high energy efficiency of clustering approaches (e.g., LEACH [Hei00]), has motivated researchers to adapt them in order to support node mobility in WSNs (e.g., LEACH-M [Kim06], LEACH-ME

[Kum08], CBR [Aww09]). Nevertheless, these suffer from the typical reliability problems of clustering approaches (See Section 2.2). Another routing approach that supports mobility is geographical routing [Kar00], which uses a greedy forwarding mechanism whereby each node forwards a packet to the neighbor that is closest to the destination. Since this method does not explicitly consider the link reliabilities, it is not recommended for lossy networks. In [Zam08], the authors introduce the *PRR*x*distance* metric, which considers both the relative position and the Packet Reception Rate (PRR). This strategy increases the network reliability compared to traditional geographical routing. However, this strategy can still lead to the construction of unreliable paths, since it exclusively uses local information in the forwarding decisions. Therefore, a different approach is necessary to overcome the lack of reliability in this kind of scenarios.

Gradient-based protocols, such as RPL [Win12] or CTP (Collection Tree Protocol) [Gna09], are reliable and efficient strategies for static WSNs. However, as shown in [Gun13] and [Cob14], the mobility of nodes deteriorates their performance. Note that since routes are defined in a proactive fashion, nodes need to keep up-to-date routing tables. Therefore, the signalling required to maintain up-to-date routes is too costly in terms of energy resources and overhead. Recently, some works tried to enhance RPL in mobile environments. In [Lee12], RPL is implemented in Vehicle Ad-hoc Networks (VANETs), and the impact of various RPL parameters is studied in these networks. In particular, the authors modify the trickle algorithm and propose a new loop avoidance mechanism. In [Saf12] and [Saa11], the mobility problem is addressed using mobile sinks. They adapt the rate of control messages according to the speed of mobile nodes in order to increase the energy efficiency of the network. In [Kor12], RPL is modified to differentiate between mobile and static nodes, prioritizing routing through static nodes rather than mobile nodes. In [Tia13], geographical information is included in RPL for WSNs assisting VANETs. The authors propose a new routing metric to improve RPL in terms of packet delivery ratio and overhead. In [Gad14], the authors propose Co-RPL, which extends RPL with the Corona mechanism for the localization of mobile nodes in WSNs. In Co-RPL, mobile nodes forward to the best neighbor within the same corona (i.e., circular area around the DODAG roots). In general, the previously mentioned approaches enhance RPL in scenarios with mobile nodes, since they take into account important issues that appear in these scenarios. However, they do not consider the inevitable positioning errors existent in real-life networks. Note that robustness is essential in WSNs, since low power communication links are very sensitive to channel dynamics.

In this chapter, we introduce KP-RPL (Kalman Positioning - RPL) in order to enhance RPL using positioning information from Kalman filtering. KP-RPL divides the network routing into: i) routing among static nodes and ii) routing among mobile and static nodes. On the one hand, we use RPL among static nodes, using ETX (Expected Transmissions) as the link quality metric. On the other hand, the routing among mobile and static nodes follows a position-based approach. First of all, each mobile node generates its own confidence region, which includes its most likely positions, taking into account the channel conditions. Then, this position is refined using Kalman filtering. Previous to the routing decision, mobile nodes generate a blacklist to discard nodes that may not be reachable due to positioning errors. Finally, nodes use end-to-end ETX estimates to select the best routing paths in each transmission time slot.

4.2.2. Contributions

The main contributions of this chapter are:

- We propose a novel routing approach, referred to as KP-RPL, that combines RPL with position-based routing. RPL provides reliable communications and position-based routing provides mobility support. This is complemented with Kalman filtering to reduce the impact of channel dynamics.
- A blacklisting approach that improves the network reliability in the presence of positioning errors. This is applied taking into account the accuracy of positioning estimates.

4.2.3. Organization of the Chapter

The remaining of this chapter is organized as follows: Section 4.3, presents an overview of KP-RPL, differentiating the different: phases, positioning, blacklisting and routing. Section 4.4, explains the positioning method of mobile node, which is based on confidence regions and Kalman filtering. Section 4.5, motivates blacklisting and explains how this is applied to reduce the impact of positioning errors. Section 4.7, evaluates KP-RPL and compares it with other position-based routing approaches in terms of positioning accuracy, PDR and ETX for different scenarios. Finally, Section 4.8 summarizes this chapter and presents the main conclusions.

4.3. KP-RPL: Kalman Positioning - RPL

Currently, the RPL routing protocol does not incorporate efficient mechanisms for handling mobile nodes, since it was designed to meet the QoS requirements of static low-power and lossy networks (LLNs), such as wireless sensor networks. Its slow response to topology changes results in the selection of suboptimal paths to the root and connectivity losses, which may lead to severe degradation of the network performance.

In this chapter, we introduce KP-RPL in order to include mobility support in RPL using position information. The block diagram of KP-RPL is shown in Figure 4.1. This divides the routing problem into two different subproblems: anchor to anchor and mobile to anchor routing. On one hand, the anchor to anchor links are constructed using the traditional RPL algorithm [Win12]. On the other hand, the mobile to anchor links are selected combining end-to-end ETX estimates, which are obtained using the positions estimated by the Kalman filter, with blacklisting. Note that we avoid constructing links among mobile nodes, since otherwise there would be two sources of positioning errors, which may lead to frequent network disconnections.

First of all, each mobile node defines its own confidence region using RSSI (Receive

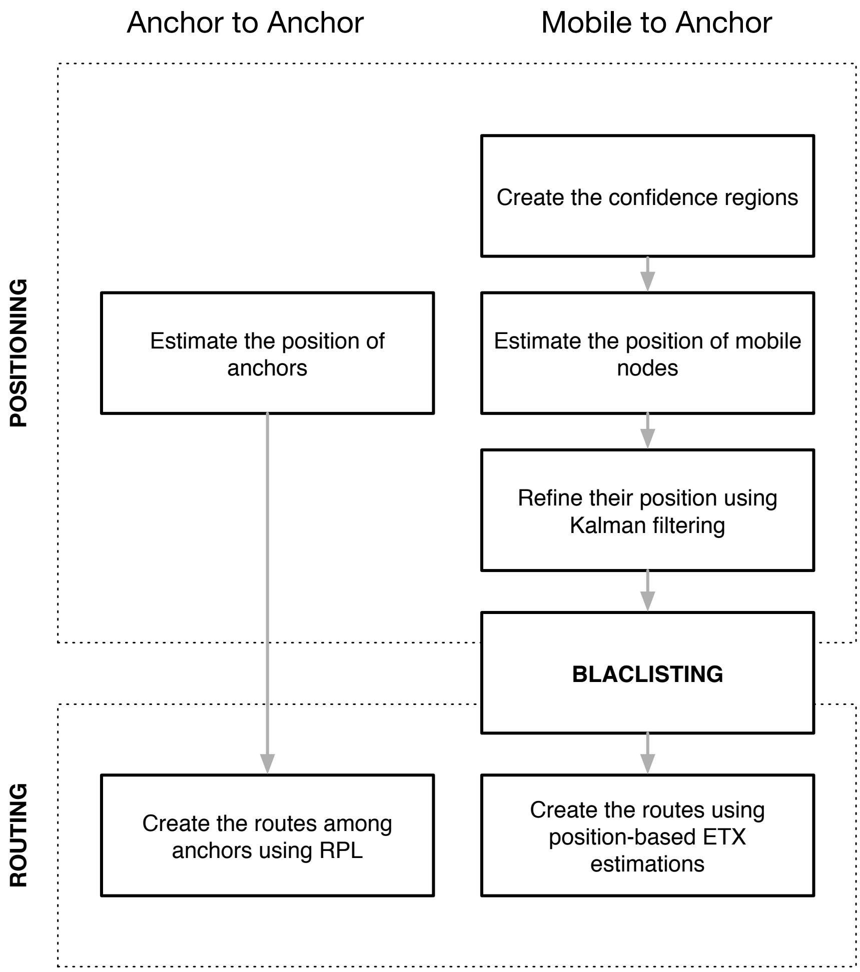


Figure 4.1. Block diagram of KP-RPL.

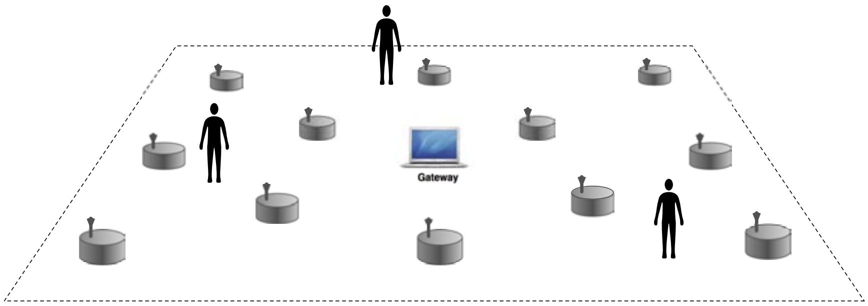


Figure 4.2. WSN composed of static and mobile sensors, which are attached to people.

Signal Strength Indicator) measurements from anchor nodes. This region is the area where this node is most likely to be located, taking into account the uncertainty of RSSI measurements. Then, each mobile node defines its position as the location with the highest probability within its own confidence region. This position is refined using Kalman filtering, which combines it with velocity estimates. Note that the low memory and computational requirements of Kalman filtering make it suitable for low power wireless sensors [Rib10]. Before making the forwarding decision, each mobile node creates a blacklist to discard potentially unreliable links, taking into account the variance of the position estimated. Finally, the end-to-end ETX values of each possible route are estimated, and the routing decisions are taken accordingly.

In the following sections, the positioning, blacklisting and routing phases are detailed.

4.4. Positioning Phase

Static nodes, also referred to as anchor nodes, are located in a well-known position so it can be assumed that the information about their location do not contain errors. On the other hand, the positions of mobile nodes have to be estimated frequently (See Figure 4.2). In this section, we introduce the positioning approach that is used by mobile nodes prior to their routing decisions.

4.4.1. Positioning based on Confidence Regions

In this section, we explain how mobile nodes estimate their position by overlapping the confidence region of various anchor nodes, which are computed using RSSI measurements as we will show in this section. This approach has been chosen instead of well-known methods, such as least squares, because it defines the influence of each anchor node individually. Then, the RSSI measurements from closer anchor nodes have a stronger impact in positioning than those measured from distant anchors.

In order to create these regions, we assume that mobile nodes know the statistics of the wireless channel that is time-varying and follows the one slope log-distance path loss model, as assumed in Chapter 3. The value of an instantaneous RSSI measurement at the k-th time period following this model is [Pat03]:

$$RSSI^k = P_{1m} - 10\alpha \log_{10} d - \gamma, \tag{4.1}$$

where P_{1m} is the received power at 1 meter from the transmitter, α is the path-loss exponent, d is the transmission distance and $\gamma \sim \mathcal{N}\left(0, \sigma_\gamma^2\right)$ is zero mean Gaussian noise that models the shadowing effects.

Following this model, the maximum likelihood estimator of the distance, between the mobile node and an anchor node, using RSSI measurements is [Pat03]:

$$\hat{d}^k_{RSSI} = 10^{\frac{RSSI^k - P_{1m}}{10\alpha}}. \tag{4.2}$$

Note that $\hat{d}^k_{RSSI}$ does not follow a Gaussian distribution due to the exponential relationship between the distance and the RSSI value. In fact, the distance estimation follows a log-normal distribution, that is:

$$\ln \hat{d}^k_{RSSI} \sim \mathcal{N}\left(\ln d^k, \sigma_d^2\right), \tag{4.3}$$

where d^k is the real distance to the anchor node and $\sigma_d = (\sigma_\gamma \ln 10) / (10\alpha)$ is the standard deviation of the distance estimation [Li07]. Then, the probability density function (*pdf*) of this estimation is:

$$f_d\left(\hat{d}\right) = \begin{cases} \frac{1}{\sqrt{2\pi}\sigma_d \hat{d}} \exp -\frac{\left(\ln \hat{d} - \ln d\right)^2}{2\sigma_d^2} & if \quad \hat{d} > 0 \\ 0 & if \quad \hat{d} \leq 0, \end{cases} \tag{4.4}$$

where d is the real distance. From the *pdf* of the distance estimation, we create the confidence region of the a-th anchor node, referred to as C_a, as the area where the mobile node is located with confidence probability q_{LN}. Then, assuming omnidirectional antennas, C_a is a circle with radius r that fulfills $\Pr\{d^k \leq r\} = q_{LN}$, centered on the a-th anchor node location, where q_{LN} is defined as:

$$q_{LN}(r) \simeq \int_{0^+}^{r} \frac{1}{\sqrt{2\pi}\sigma_d x} \exp\left(-\frac{\left(\ln x - \ln \hat{d}_{RSSI}^k\right)^2}{2\sigma_d^2}\right) \mathrm{d}x. \tag{4.5}$$

Then, the m-th mobile node defines its confidence region by overlapping the confidence regions of the L closest anchor nodes (Figure 4.3), that is:

$$C_m = C_1 \cap C_2 \cap ... \cap C_L. \tag{4.6}$$

Within its confidence region, the mobile node estimates its position, referred to as $\hat{\mathbf{p}}_m$, evaluating a set of discrete points uniformly distributed inside C_m, defined as $\mathcal{C}_m$. As far as the probability is almost constant in small areas, a density of 2 points/m^2 suffices in our particular implementation. Then, the mobile node estimates the probability to be located in each point of $\mathcal{C}_m$ using (4.4). Assuming that the *pdfs* of different anchor nodes are independent, the mobile node computes its position $\hat{\mathbf{p}}_\mathbf{m}$ as:

$$\hat{\mathbf{p}}_\mathbf{m} = \max_{\mathbf{c}} \prod_{a \in \mathcal{A}^p} q_{LN}(d_{c,a}) \quad \forall \mathbf{c} \in \mathcal{C}_m, \tag{4.7}$$

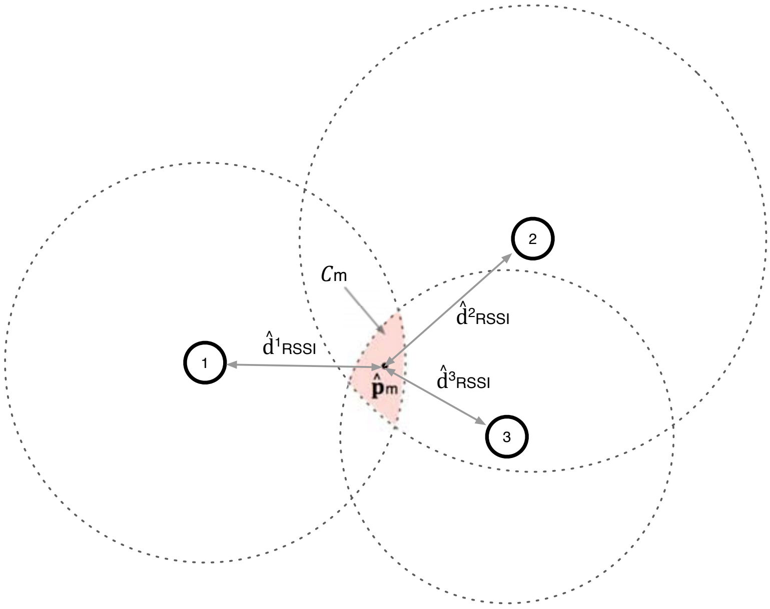


Figure 4.3. Confidence region of a mobile node created overlapping the confidence regions of three anchor nodes.

where $\mathcal{A}^p$ is the set of anchor nodes considered for positioning and $d_{c,a}$ is the euclidean distance from the c-th point to the a-th anchor node (i.e., $\|\mathbf{c} - \mathbf{p}_a\|^2$).

4.4.2. Enhanced Positioning using Kalman Filtering

Once the position of the mobile node has been estimated from its confidence region, KP-RPL combines it with velocity estimates using Kalman filtering [Kal60]. The Kalman filter is a set of mathematical equations that provides an efficient computational solution of the least-squares method [Wel95]. In the context of WSNs, this has been extensively used in numerous applications, particularly for tracking and navigation [Cor14]. Its low processing and memory requirements [Rib10] and the positioning accuracy obtained with this filter motivate us to implement it in KP-RPL. In this section, we introduce the basics of Kalman filtering, and the interested reader is referred to [May79] for a detailed explanation.

A Kalman filter is articulated in two steps: i) the prediction step, and ii) the correction step. In the prediction step, the state vector $\mathbf{x}^{k-1}$, which includes the information of the mobile node at the (k-1)-th time instant, is updated using the transition matrix $\mathbf{F}$. On one hand, we model the state vector at the k-th time instant $\mathbf{x}^k$ in a two dimensional space by means of its position and velocity as:

$$\mathbf{x}^k = \begin{bmatrix} p_x^k & p_y^k & v_x^k & v_y^k \end{bmatrix}^T, \tag{4.8}$$

where p_x^k, p_y^k and v_x^k, v_y^k represent the position and the velocity in Cartesian coordinates, respectively. On the other hand, the transition matrix models the movement of the node, which in our case is assumed to have constant velocity in x and y. Then, this matrix is defined as:

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{4.9}$$

where T is the transmission period of the mobile node. Using this matrix, the state vector $\mathbf{x}^k$ is computed as:

$$\mathbf{x}^k = \mathbf{F}\mathbf{x}^{k-1} + \mathbf{v}^k, \tag{4.10}$$

where $\mathbf{v}^k$ is a zero mean Gaussian noise with covariance matrix $\mathbf{Q}$.

On the other hand, the state covariance matrix at the k-th time instant $\mathbf{P}^k$, which estimates the covariances of the positioning and velocity estimates in $\mathbf{x}^k$, is updated as:

$$\mathbf{P}^k = \mathbf{F}\mathbf{P}^{k-1}\mathbf{F}^T + \mathbf{V}\mathbf{Q}\mathbf{V}^T, \tag{4.11}$$

where $\mathbf{V}$ is the state noise covariance matrix, which models possible acceleration changes, that is defined as:

$$\mathbf{V} = \begin{bmatrix} 0.5T^2 & 0 \\ 0 & 0.5T^2 \\ T & 0 \\ 0 & T \end{bmatrix}. \tag{4.12}$$

Note that we have not considered the position and velocity measurements at the k-th instant yet, since $\mathbf{x}^{k-1}$ is an input coming from the previous iteration. In the correction step, we take into account the vector of measurements at the k-th instant $\mathbf{z}^k$ that is defined as:

$$\mathbf{z}^k = \begin{bmatrix} \tilde{p}_x^k & \tilde{p}_y^k & \tilde{v}_x^k & \tilde{v}_y^k \end{bmatrix}^T, \tag{4.13}$$

where $\tilde{p}_x^k$, $\tilde{p}_y^k$ and $\tilde{v}_x^k$, $\tilde{v}_y^k$ are the position and velocity measurements in Cartesian coordinates, respectively. Note that the m-th node obtains $\tilde{p}_x^k$ and $\tilde{p}_y^k$ from $\hat{\mathbf{p}}_\mathbf{m}$, while $\tilde{v}_x^k$, $\tilde{v}_y^k$ are measured by its accelerometer.

Then, we correct the state vector and the state covariance matrix with $\mathbf{z}^k$ as [Wel95]:

$$\mathbf{x}^k = \mathbf{x}^k + \mathbf{K}^k(\mathbf{z}^k - \mathbf{H}\mathbf{x}^k), \tag{4.14}$$

and

$$\mathbf{P}^k = (\mathbf{I} - \mathbf{K}^k\mathbf{H})\mathbf{P}^k, \tag{4.15}$$

where $\mathbf{H}$ is the measurement matrix, which maps the measurements into the state vector. The Kalman gain $\mathbf{K}$ is computed as [Wel95]:

$$\mathbf{K}^k = \mathbf{P}^k\mathbf{H}^T(\mathbf{H}\mathbf{P}^k\mathbf{H}^T + \mathbf{R})^{-1}, \tag{4.16}$$

where $\mathbf{R}$ is the covariance matrix of measurements. Since the covariance of positioning is difficult to estimate, this matrix has to be defined with a calibration process.

Finally, from the first two components of $\mathbf{x}^k$ we obtain the new position estimation $\hat{\mathbf{p}}_m^*$, and from $\mathbf{P}^k$ we obtain the estimated variance of this position $\hat{\sigma}_k$. In Section 4.5, we

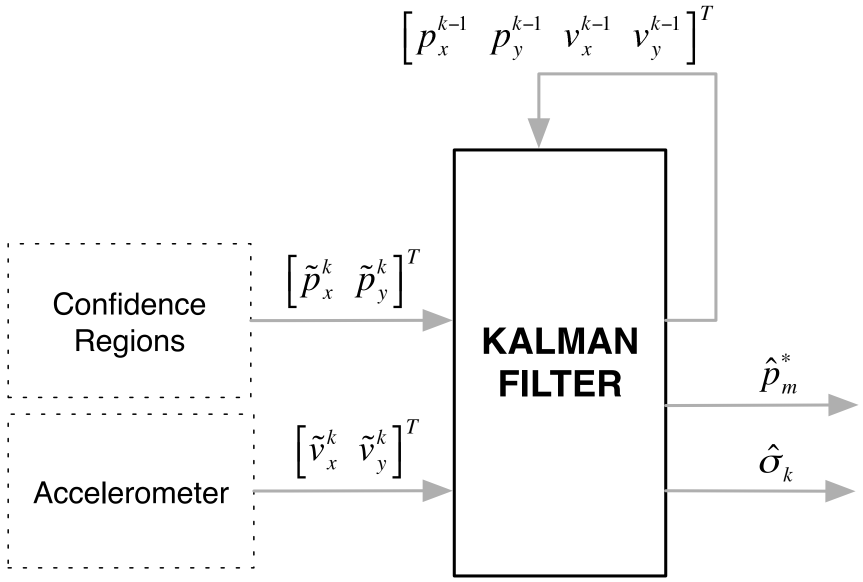


Figure 4.4. Inputs/outputs of the Kalman filter.

use the output of the Kalman filter to discard the nodes that may be unreachable from the routing process. In Section 4.6, the quality estimation of links is computed with $\hat{\mathbf{p}}_m^*$.

Figure 4.4 shows the inputs and outputs of the Kalman filter.

4.5. Blacklisting Phase

Due to the limitations of positioning algorithms, it is impossible to avoid position inaccuracies, which can cause a relevant degradation in the performance of position-based routing algorithms [Pen11].

In order to diminish their impact in the reliability of the network, in KP-RPL mobile nodes create a blacklist before making any routing decision. This list includes the nodes that could be unreachable, taking into account the variance of position estimates. Basically, we force that the candidate anchor nodes of mobile nodes must be reachable from any point inside their confidence region.

After the Kalman filter, mobile nodes construct their confidence regions again, taking into account the refined position and its variance. It is important to note that the distances between this position and anchor nodes do not follow a log-normal distribution. Assuming that the position error of $\hat{\mathbf{p}}_m^*$ is independent in x and y and Gaussian $\mathcal{N}\left(0, \hat{\sigma}_k^2\right)$, these distances follow a Rician distribution [Pap02]. Then, the confidence regions after Kalman filtering, with confidence probability q_R, are circles with radius r that fulfill $Pr\{d^k \leq r\} = q_R$, centered on the anchor node location. The confidence probability q_R is defined as:

$$q_R(r) \simeq \int_{0^+}^{r} \frac{x}{\hat{\sigma}_k^2} \exp\left(-\frac{x^2 + \hat{d_K}^2}{2\hat{\sigma}_k^2}\right) I_0\left(\frac{x\hat{d_K}}{\hat{\sigma}_k^2}\right) \mathrm{d}x, \tag{4.17}$$

where $\hat{d_K}$ is the euclidean distance from the position estimated by the Kalman filter to the anchor node, and I_0 is the modified Bessel function of the first kind with zero order. Then, the m-th mobile node defines $\mathcal{C}_m^*$, from the intersection of the newly generated confidence regions, as in Section 4.4.1. Then, its set of candidate anchor nodes $\mathcal{K}_m$ includes the anchor nodes that fulfill the following condition:

$$\|\mathbf{c} - \mathbf{p}_a\|^2 \leq r_{tx}, \quad \forall \mathbf{c} \in \mathcal{C}_m^*, a \in \mathcal{A}, \tag{4.18}$$

where $\mathcal{A}$ is the set of anchor nodes in the network, and r_{tx} is the transmission range of mobile nodes.

Thanks to the newly generated confidence regions, only the anchors that can be reachable by the mobile node (i.e., nodes in $\mathcal{K}$), taking into account its position and velocity estimates, are considered in the routing phase. Then, the probability of reaching the anchor node selected is increased, since the impact of positioning errors is lower, and therefore the robustness of the network is enhanced.

4.6. Routing Phase

The routing problem in KP-RPL is decomposed into: i) anchor to anchor routing and ii) mobile to anchor routing. This strategy exploits the simplicity and reliability of RPL among static nodes, and provides mobility support to specific nodes using position-based routing.

4.6.1. Anchor to Anchor Routing

Gradient-based routing is an energy efficient and reliable solution to construct convergecast graphs, as we have shown in Chapter 3. Following the same approach, we use the RPL protocol for the routing among anchor nodes, since this is the most prominent standard for LLNs. RPL constructs a convergecast directed acyclic graph rooted at the sink node, referred to as DODAG (Destination Oriented Acyclic Graph), using control messages referred as DIOs (DODAG Information Objects) (See Section 2.2.1).

The Rank of anchor nodes is the number of expected transmission (ETX). Basically, this metric indicates how many times a message must be transmitted on average to reach its destination. The i-th anchor node estimates its end-to-end ETX, when forwarding to the j-th anchor node, as the ETX to reach the j-th anchor node ($ETX_{i,j}$), plus the end-to-end ETX of this node (ETX_j). The ETX between two nodes is computed as the reciprocal of their packet delivery ratio (PDR), then:

$$ETX_i = \frac{1}{PDR_{i,j}} + ETX_j. \tag{4.19}$$

4.6.2. Mobile to Anchor Routing

Unfortunately, RPL is designed for static networks, and the mobility of nodes deteriorates its performance [Cob14]. In addition, the signalling required to maintain up-to-date routes is too costly in terms of energy resources and overhead [Gun13]. In order to overcome these issues, in KP-RPL mobile nodes make their routing decisions using positioning information, which is obtained from the Kalman filter. Moreover, mobile nodes only consider their set of candidate anchor nodes ($i.e., \mathcal{K}$) as possible parents.

Following the RPL approach, the mobile nodes select their parents according to the ETX from themselves to the sink. Then, the end-to-end ETX of the m-th mobile node (ETX_m), when transmitting through the a-th anchor node, is estimated as the ETX from itself to this anchor node ($ETX_{m,a}$), plus the ETX from the anchor to the sink (ETX_a):

$$ETX_m = ETX_{m,a} + ETX_a. \tag{4.20}$$

On one hand, ETX_a can be obtained from the Rank of the a-th anchor node, which is computed using (4.19). Ranks are included in the *Rank* field of DIO packets, and the mobile node may solicit this information using a DIS packet, as explained in [Win12].

On the other hand, the $ETX_{m,a}$ is estimated as the reciprocal of $PDR_{m,a}$. According to [Ram06], this PDR can be estimated as the probability of receiving a packet at the k-th time period with RSSI higher than the sensitivity of the transceiver ($RSSI_{th}$), that is:

$$PDR_{m,a} = Pr\{RSSI^k_{m,a} \geq RSSI_{th}\}. \tag{4.21}$$

Assuming that RSSI measurements decay with distance following the log-distance path loss model in (4.1), then:

$$PDR_{m,a} = Pr\{\gamma \leq P_{1m} - 10\alpha \log_{10} d_{m,a} - RSSI_{th}\}. \tag{4.22}$$

Since γ is Gaussian $\mathcal{N}\left(0, \sigma^2_\gamma\right)$, we can estimate $PDR_{m,a}$ as:

$$PDR_{m,a} = \Pr\{\gamma \leq X\} = \int_{-\infty}^{X} \frac{1}{\sqrt{2\pi}\sigma_\gamma} \exp\left(-\frac{\gamma^2}{2\sigma^2_\gamma}\right) \mathrm{d}\gamma, \tag{4.23}$$

where X has been used for the sake of clarity instead of $P_{1m} - 10\alpha \log_{10} \hat{d}_{m,a} - RSSI_{th}$.

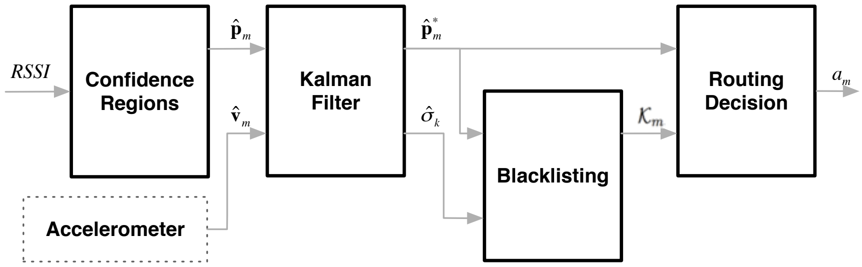


Figure 4.5. Overview of the communication among modules in KP-RPL (mobile nodes).

Note that $d_{m,a}$ has been substituted by $\hat{d}_{m,a}$, since we estimate the distances between mobile and anchor nodes using the position estimated by the Kalman filter ($\hat{\mathbf{p}}^*_m$).

Finally, the m-th mobile node selects its parent a_m, from the set of candidate anchor nodes $\mathcal{K}_m$, using (4.20) as:

$$a_m = \arg\min_k \left(\frac{1}{PDR_{m,k}} + ETX_k \right), \quad \forall k \in \mathcal{K}_m. \tag{4.24}$$

Note that (4.24) strongly depends on the accuracy of positioning, since $PDR_{m,k}$ is estimated based on $\hat{\mathbf{p}}^*_m$. Moreover, it also depends on the effectiveness of the blacklisting strategy, since this decides the anchor nodes in $\mathcal{K}_m$.

In Figure 4.5, we summarize the communication among modules in KP-RPL, from the creation of confidence regions to the final routing decision.

4.7. Simulation Results

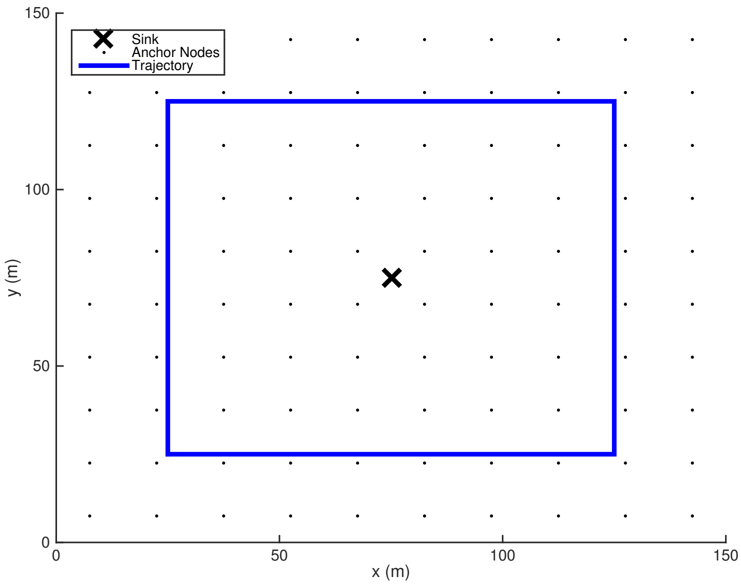
In this section, the performance of KP-RPL, with and without blacklisting, is evaluated using Matlab and compared with two existent geographical routing approaches. A version of KP-RPL without Kalman filtering, referred to as P-RPL, is also included in the simulations to observe the impact of Kalman filtering. In this case, the blacklisting and routing decisions are taken using the position estimates obtained from the confidence regions. We have considered the typical geographical routing forwarding criterion (i.e., nodes transmit to the node closest to the sink, as long as it is reachable) [Pen11], and also the strategy proposed in [Zam08] (*PRR* x *distance*). In the latter case, each mobile node multiplies the reliability of each link by the distance improvement towards the destination, and then it selects the node that maximizes this value. We assume that both geographical routing approaches use the well-known least squares algorithm to estimate the position of mobile nodes. For the sake of a fair comparison, geographical routing is only considered for the mobile to anchor communications, being anchor to anchor communications always managed by the RPL protocol.

The received power at 1 m has been empirically measured using IRIS motes (i.e., P_{1m}= -47 dBm). A path loss exponent α=3 [Pu12] and a path loss variance σ_γ^2=6 [Rap01] are assumed. Moreover, the sensitivity value and the maximum transmission power of the IRIS transceiver (i.e., RF230 [Atm09]), have been considered (-91 dBm and 3 dBm, respectively).

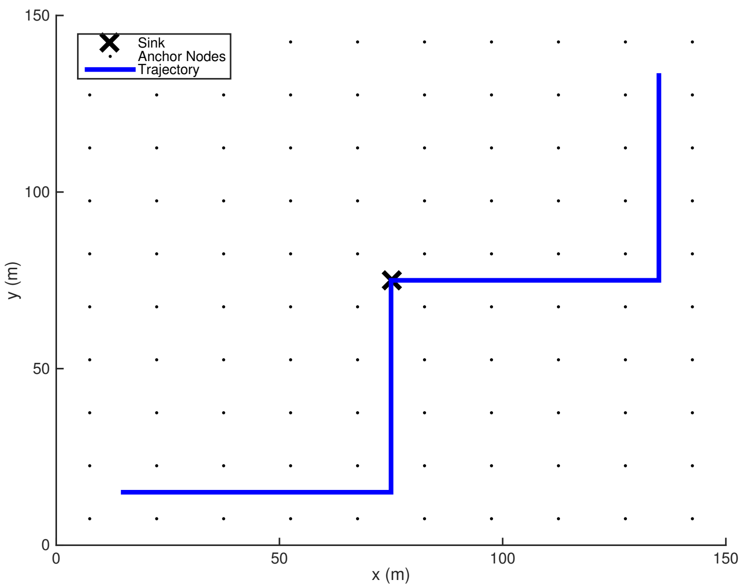
The scenario is composed of a mobile node, which moves across a 150x150 m^2 sensing area, and 100 anchor nodes uniformly distributed within this area in a square grid formation. The mobile node moves with a speed of 2 m/s and transmits a packet per second. In particular, we consider the trajectories of the mobile node in Figure 4.6. An accelerometer provides speed estimates in x and y. The error of this measurements in each direction follows a Gaussian distribution $\mathcal{N}\left(0, \sigma_s^2\right)$, with zero mean and $\sigma_s = 0.5$ m/s. In order to better observe the performance of each algorithm, we assume that the MAC layer is able to avoid collisions among packets.

The mobile node considers its 4 closest anchor nodes for positioning, selected according to its instantaneous RSSI measurements. Note that a higher number of anchor nodes may reduce the positioning error, but it would also increase the computing complexity of positioning. In both P-RPL and KP-RPL, the confidence regions of anchor nodes are created using the 95% confidence intervals.

The most relevant parameters considered in these simulations are summarized in Table 4.1.



(a) Square trajectory.



(b) Zig-zag trajectory.

Figure 4.6. Trajectories of the mobile node.

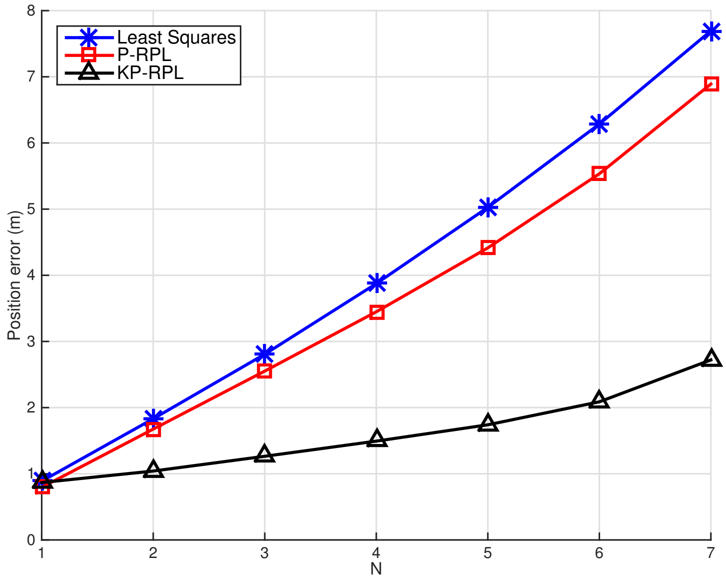
Table 4.1. Simulation Parameters

Parameter	**Value**
Sensing area	150x150 m^2
Anchor nodes	100
Speed of mobile nodes	2 m/s
Transmission period	1 s
Transmit power	3 dBm
Sensitivity of the receiver	-91 dBm
Power received at 1 m (P_{1m})	-47 dBm
Path loss exponent (α)	3
Path loss variance (σ_γ^2)	6

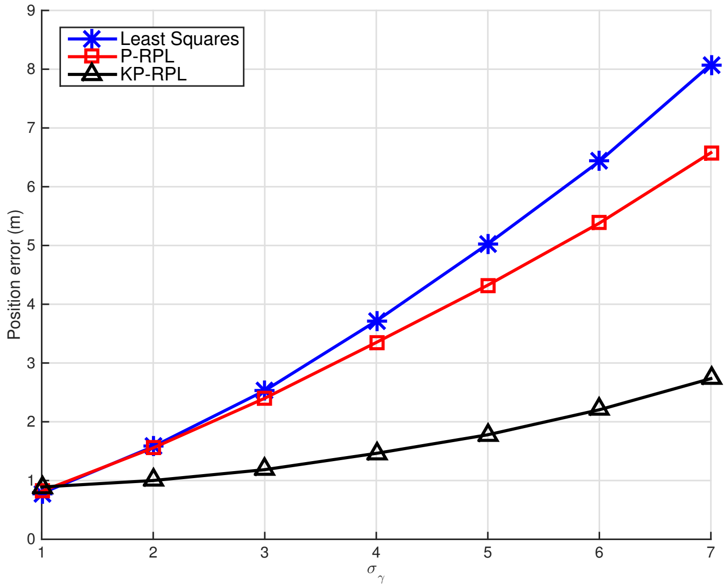
4.7.1. Robustness against Channel Conditions

First of all, we evaluate the impact of the channel conditions in KP-RPL. In Figure 4.7, we compare the positioning error of KP-RPL and P-RPL for different σ_γ. For the sake of comparison, we also include the positioning error of the least squares algorithm. As shown in this figure, we can obtain a better positioning accuracy using confidence regions than using least squares. The reason is that confidence regions are created taking into account the channel conditions, and the impact of RSSI measurements depends on the distance from the mobile node to each anchor node. We can also observe that Kalman filtering significantly enhances the positioning robustness thanks to the correction of RSSI measurements using the velocity estimates from the accelerometer and the prediction model. For instance, if $\sigma_\gamma = 7$ the positioning error in both square and zig-zag trajectories is reduced more than 5 meters compared to least squares.

In Figure 4.8, we evaluate the reliability in terms of PDR for different channel conditions. This shows that, even without blacklisting, both P-RPL and KP-RPL are more reliable than geographical routing. Moreover, blacklisting increases their PDR up to 12%. This figure also shows that in some cases P-RPL has a higher PDR than KP-RPL (See Figure 4.8b). It is important to remark that the routing metric of both protocols is ETX, and although the PDR and ETX of a link are reciprocal values, in multi-hop networks they are not equivalent. This is because ETX also depends on the number of hops (i.e.,



(a) Square trajectory.



(b) Zig-zag trajectory.

Figure 4.7. Position error for different σ_γ.

$ETX_{a,b,c} = ETX_{a,b} + ETX_{b,c}$, $PDR_{a,b,c} = PDR_{a,b}PDR_{b,c}$). Therefore, in order to increase the ETX of the network, KP-RPL reduces its PDR compared to P-RPL in these cases. We evaluate this tradeoff in Section 4.7.5.

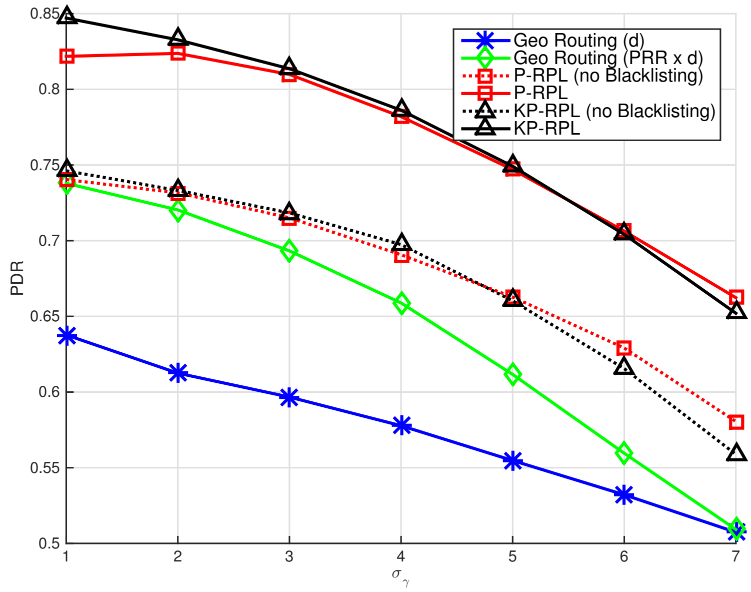
4.7.2. Number of Concurrent Positioning Nodes

In RPL, the nodes use the trickle algorithm [Lev11] to adapt the rate of DIOs according to the current needs. With this algorithm, the energy consumption of these nodes can be reduced when the network routes do not need to change frequently. However, position-based routing needs frequent RSSI measurements, since the mobile node needs to estimate its position before making any forwarding decision. As a result, the rate of DIOs must be kept constant, leading to high battery consumption of anchor nodes. Then, it is important to evaluate the impact of reducing the number of concurrent anchor nodes that broadcast DIO packets to reduce the network consumption.

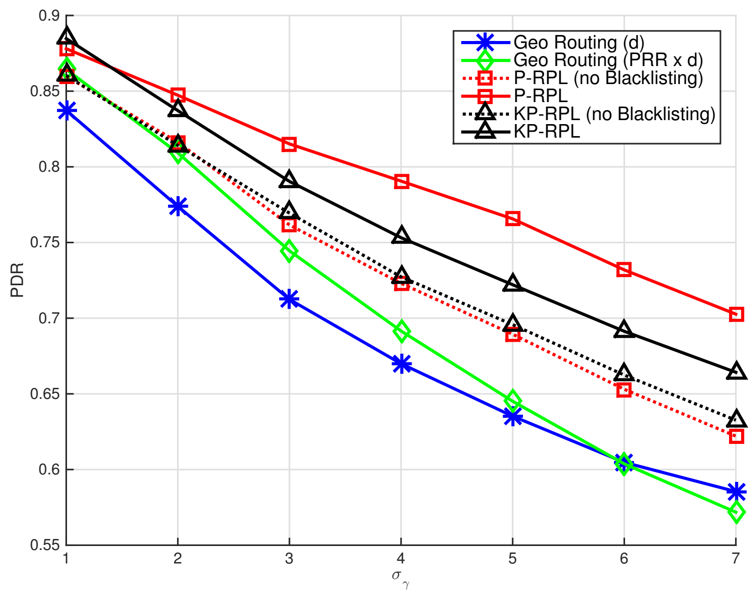
Assuming that anchor nodes do not coordinate their DIOs, in Figure 4.9 we evaluate the average ETX for different activation probabilities (i.e., active anchor nodes broadcast DIOs). In both trajectories, we observe a higher robustness of KP-RPL in front of low activation probabilities than the rest of approaches. In fact, we can observe that KP-RPL only needs an activation probability of 0.2 to achieve the same performance than keeping all the anchor nodes always active. On the contrary, the rest of approaches need an activation probability higher than 0.5 to achieve a similar performance.

4.7.3. Density of Anchor Nodes

A large density of anchor nodes enhances the positioning accuracy and the network reliability, but it also increases the network cost. In order to observe the impact of the density of anchor nodes, Figure 4.10 and Figure 4.11 show the positioning error and the ETX for different densities, respectively. In this case, we have assumed a random deployment of anchor nodes, since these cannot always be distributed in a square grid

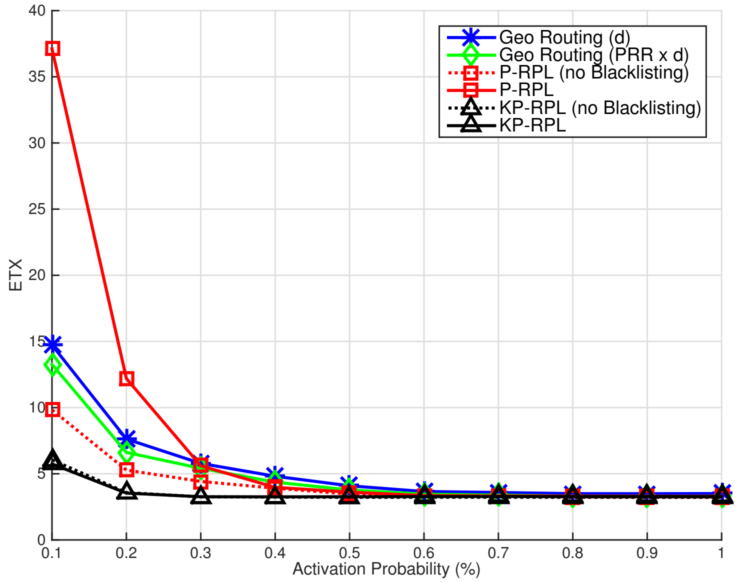


(a) Square trajectory.

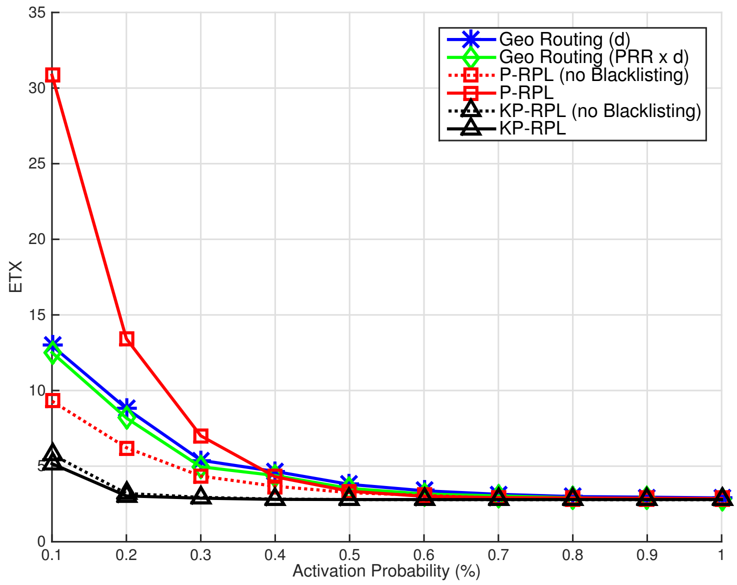


(b) Zig-zag trajectory.

Figure 4.8. Packet delivery ratio for different σ_γ.



(a) Square trajectory.



(b) Zig-zag trajectory.

Figure 4.9. Expected number of transmissions for different activation probabilities.

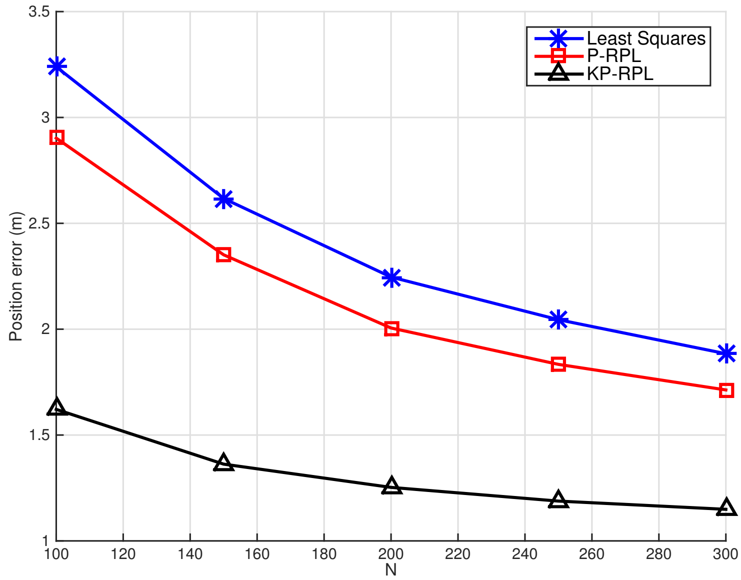


Figure 4.10. Position error for different anchor node densities (Square Trajectory). The anchor nodes are distributed randomly.

formation due to their number. We can observe that both in terms of position error and ETX, the density of anchor nodes can be lower using KP-RPL than with the rest of strategies for a given target performance. Note that KP-RPL obtains a lower position error using 100 anchor nodes than least squares using 300 anchor nodes. This means that Kalman filtering is particularly effective in low density deployments.

4.7.4. Energy Consumption

In Figure 4.12, we compare the average power consumption spent for transmitting data packets from the mobile node to the sink. We assume the transmission consumption of the RF230 transceiver (i.e., 16.5 mA), the typical size of IEEE 802.15.4 data packets (i.e., 127 bytes), and a bitrate of 250 kbps. Besides the packet rate, this consumption mainly depends on the average ETX of the network. In the simulation results, we can observe that in most cases KP-RPL has the lowest energy consumption. This is particularly lower in scenarios with bad channel conditions, thanks to the higher robustness of KP-RPL

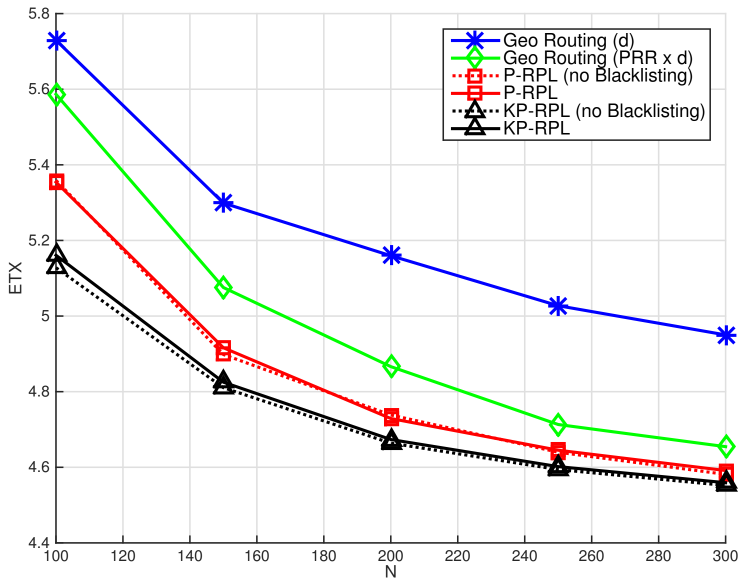
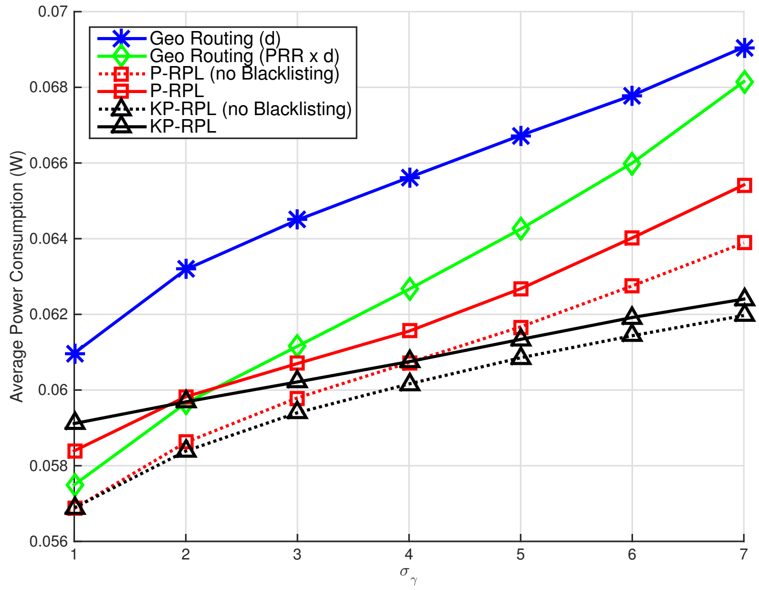


Figure 4.11. ETX for different anchor node densities (Square Trajectory). The anchor nodes are distributed randomly.

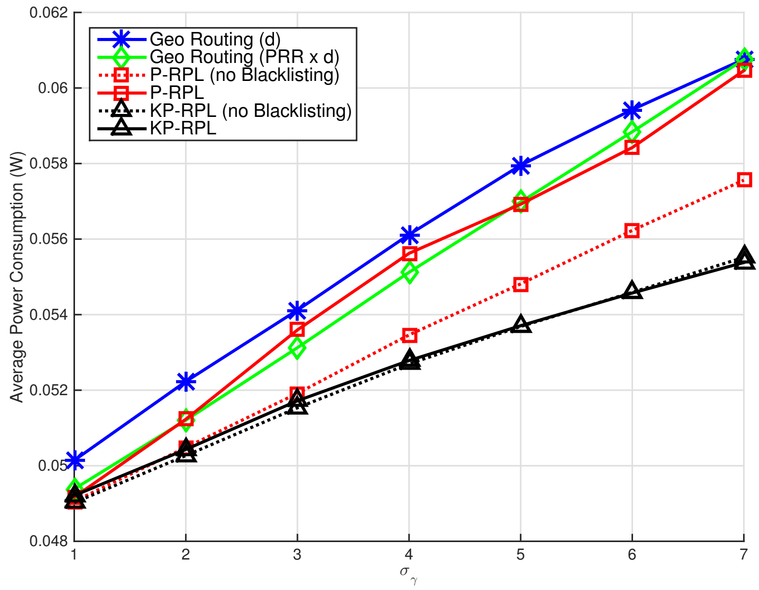
shown in Section 4.7.1. Therefore, KP-RPL reduces the overall energy consumption of the network thanks to: i) the lower power that is necessary to transmit data packets, and ii) the lower rate of positioning packets that is required to achieve accurate positioning, as we have shown in Section 4.7.3.

4.7.5. ETX vs PDR Tradeoff

We have observed in Figure 4.8 that in some cases KP-RPL has a lower PDR than P-RPL. This is because of the ETX vs. PDR tradeoff caused by blacklisting. On one hand, the ETX of the network may be reduced if the confidence regions are too conservative (i.e., mobile nodes may not be considering reliable paths). On the other hand, Figure 4.8 has shown that blacklisting increases the average PDR. Therefore, the size of the confidence regions has a relevant impact on both ETX and PDR, and this impact may be different in KP-RPL and P-RPL. In order to evaluate this tradeoff, Figure 4.13 shows the values of ETX and PDR for different blacklisting criteria. We create the confidence



(a) Square trajectory.



(b) Zig-zag trajectory.

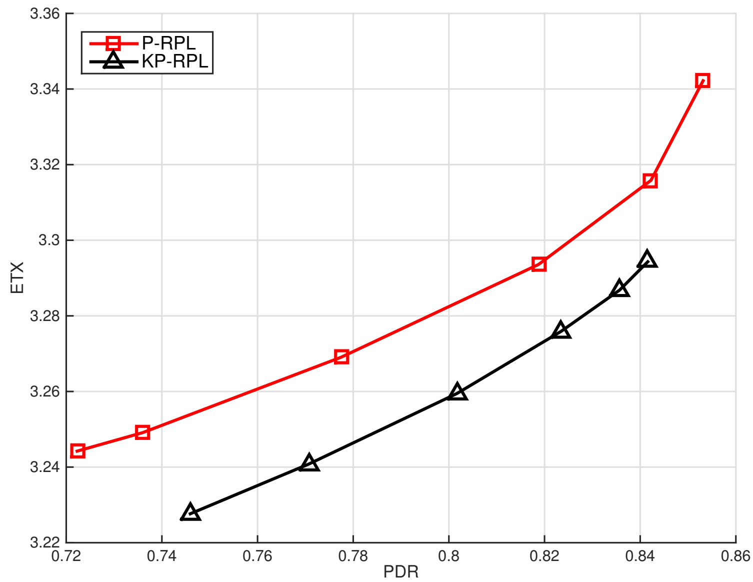
Figure 4.12. Average power consumption (W) for different channel conditions.

regions of anchor nodes using different confidence probabilities ($q = q_{LN} = q_R$), where low values create small blacklisting areas, while high values tend to create larger areas in order to be more restrictive. We can see that the tradeoff can be managed by adjusting the size of the confidence regions. Note that certain applications may require a minimum PDR, and therefore q has to be increased, even if it also increases the average ETX.

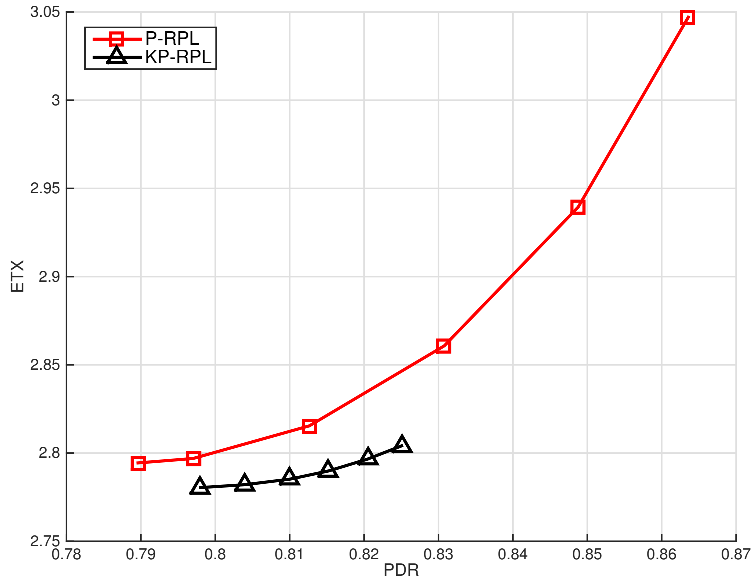
4.8. Summary and Conclusions

The implementation of WSNs with mobile nodes in real-life scenarios is a challenging task, particularly in environments with bad channel conditions, such as industrial scenarios. In this chapter, we address this issue with KP-RPL, which further enhances RPL enabling it to cope with both static and mobile nodes. KP-RPL combines RPL with a robust, yet simple positioning algorithm, based on the construction of confidence regions and Kalman filtering. Moreover, a blacklisting strategy is applied to avoid considering unreliable links. We have shown that Kalman filtering improves the routing decisions of mobile nodes, since it enhances their position estimates. On the other hand, blacklisting provides robustness in front of the inevitable positioning errors of mobile nodes, which are due to the inaccuracies of RSSI measurements and velocity estimates.

In general, the simulation results show that KP-RPL is a reliable routing approach that is also robust to channel conditions. We have evaluated the average ETX of KP-RPL in different network conditions and mobile node trajectories, observing that it increases the reliability of the network up to 25%, compared to geographical routing. The results have shown that blacklisting enhances the network reliability, but it may also increase the average ETX of the network. This generates a tradeoff that can be managed with the size of the confidence regions, where the ETX is improved with smaller areas, and larger areas increase the PDR. We have also evaluated KP-RPL for different anchor node densities, since this has an impact on the positioning accuracy of the network. The results show that the number of anchor nodes can be reduced using KP-RPL, compared



(a) Square trajectory.



(b) Zig-zag trajectory.

Figure 4.13. Tradeoff between PDR and ETX using different confidence probabilities: $q = 0.382$ (0.5σ) (bottom-left), $q = 0.682$ (1σ), $q = 0.866$ (1.5σ), $q = 0.954$ (2σ), $q = 0.988$ (2.5σ) and $q = 0.998$ (3σ) (top-right).

to geographical routing approaches. Thus, the infrastructure cost of the network is lower. In terms of energy consumption, we have shown that KP-RPL enhances the overall network consumption. On one hand, using KP-RPL the number of simultaneously active anchor nodes can be reduced, since this needs less nodes for positioning. On the other hand, the average energy consumption due to the transmission of data packets is also lower, thanks to its higher reliability. As a result, the lifetime of both anchor and mobile nodes is extended.

Chapter 5

Multi-Tree Routing to Manage Heterogeneous Traffic

5.1. Summary

The QoS requirements of the traffic generated by WSNs collecting a single phenomenon can be easily managed by designing the routing protocol properly. However, the increasing sensing and computational capabilities of wireless sensors enables the implementation of more advanced services, which may combine data from different sensing units. Moreover, in the near future WSNs are planned to be shared infrastructures that may be used by multiple on-demand services. This generates multiple traffic flows in the same WSNs with their own QoS requirements.

In Chapters 3 and 4, homogeneous traffic is assumed. In order to manage heterogeneous traffic, RPL may virtually split the network into multiple RPL Instances, which are constructed according to a predefined objective function. However, this protocol does not define any mechanism to decide the nodes that must belong to each instance, and this decision has a strong impact in the network energy consumption and performance. With this in mind, in this chapter we introduce C-RPL (Cooperative - RPL), which creates multiple instances following a cooperative strategy among nodes with different sensing tasks. As a result, the energy consumption, the complexity and the cost of the wireless

sensors is reduced compared to RPL, since they are active less time, perform fewer tasks and are equipped with less sensing hardware. In this chapter we also propose a novel fairness analysis for networks with multiple instances, showing that C-RPL achieves a better tradeoff, in terms of performance and energy consumption, than RPL with non-cooperative instances.

5.2. Introduction

5.2.1. Motivation and Previous Work

Many advanced WSN applications need to develop multiple tasks. For instance, SHM (Structure Health Monitoring) [Har10] systems may need to collect information coming from different sensing units, such as pressure, vibration or temperature. Moreover, they also need to send alarm messages in case of broken sections or systems failures. In addition, continuous messages are broadcast for external monitoring and calibration. Wireless Sensor Surveillance Networks (WSSNs) may also need to combine multiple traffics, such as event detection traffic, to detect intrusions or smoke, and positioning information of mobile entities [Bar13b]. All this heterogeneous traffic needs to be properly managed by the network. For instance, latency is a critical requirement in event detection applications [Bag10]. On the contrary, critical monitoring tasks may admit a certain delay in some cases, but they require a high reliability [Luo11]. On the other hand, ambient monitoring applications may not have strict delay or reliability constraints, but a low energy consumption becomes crucial because these measurements are transmitted periodically [Pan14].

Multi-objective routing approaches, such as [Mal06], [Alw13], consider multiple criteria simultaneously to handle different QoS requirements. These strategies find a tradeoff solution taking into account multiple objective functions at the same time. For instance, in [Che10] the authors propose a solution to increase the lifetime and throughput of the network, while reducing its latency. In [Yet12], the aggregate energy consumption and delay are optimized using genetic algorithms. However, the requirements of different kinds of traffic may not be satisfied with multi-objective routing, since they are not addressed individually. Note that frequently, these requirements have contradictory relationships among them [Mie03]. For instance, the lowest delay to the destination is generally found using the minimum number of hops, while the solution with the lowest energy consumption may require many shorter (i.e., low-power) transmissions (See Section 2.5).

Generally, the individual QoS requirements of multiple traffic flows are addressed through buffering or prioritized Medium Access Control (MAC) mechanisms [Ari13]. At the network layer, it is also possible to provide QoS differentiation through multiple routing trees. Although this mechanism has been mainly used in the literature for load balancing [Chu11], or to increase the network robustness in front of faulty links [Wei07], it can also be used to efficiently manage the QoS requirements of heterogeneous traffic [Lon13]. In fact, the well-known RPL (Routing Protocol for Low-Power and Lossy Networks) protocol [Win12] has adopted this approach. In order to address multiple traffic flows, this protocol virtually splits the network into multiple RPL Instances, where each instance uniquely defines a set of one or more DODAGs (Destination Oriented Directed Acyclic Graphs) with a common objective function. Then, each instance can be used to address the particular requirements of a specific kind of traffic. In [Raj14], two RPL instances are defined to individually manage latency constrained and high priority traffic in Smart Grids. In particular, they construct each RPL Instance according to the minimum number of hops and the minimum expected number of transmissions (ETX), respectively. In [Lon13], the authors differentiate between nodes for monitoring purposes, and nodes for high priority traffic (e.g., alarms). Then, they create one RPL Instance that is only composed of nodes from the first group, and also a second instance that groups nodes from both groups. However, not any of these strategies define how to dynamically select the nodes that belong to each instance, and therefore this decision must be taken in advance. Note that the best solution depends on the objective function of each instance and the particular network conditions. Therefore, RPL could either follow a low-consumption strategy (i.e., each node belongs only to the instance associated to its tasks) or a high-reliability strategy (i.e., any node may belong to any instance as long as it senses or forwards the kind of traffic associated to that instance). On one hand, the first strategy may reduce the network performance, due to the reduction of the node density of each instance. On the other hand, the second strategy does not consider that nodes associated to different tasks may have different duty cycles. As a result, this solution may not be energy efficient, since it does not prioritize the communication among nodes with the same duty cycle. Then, many nodes have to extend their active time, thus increasing their energy consumption.

In this chapter, we propose a cooperative RPL-based strategy (C-RPL) to manage this tradeoff. This defines the nodes that belong to each instance, referred in C-RPL to as C-RPL Instances, following a cooperative strategy. Taking into account the selfish nature of nodes, coalitions are created according to a utility function that considers the tradeoff between the performance and the energy consumption. From a game theoretical perspective, the solution of this cooperation problem, such as the solution of the WSN cooperation problem in [Md13], has been found to be equivalent to the well-known prisoner's dilemma game [Axe81]. Briefly, this is a two person zero game that describes a situation where two players increase their utility if they both cooperate, but if a player decides not to cooperate while the other cooperates, its utility gain is even higher than cooperating. As a result, players will never cooperate (i.e., Nash equilibrium of the prisoner's dilemma game). This game is suitable for studying complex interactions among players, such as the cooperation among RPL instances, since rational actions do not cause the Pareto optimality.

On the other hand, since multiple performance criteria may be involved (e.g., latency, bandwidth, energy consumption), it is important to distribute the network resources in a "fair" manner (i.e., considering the requirements of each traffic in the network). Although many definitions can be found in the literature to evaluate fairness [Shi14], such as weighted fairness, max-min fairness or proportional fairness, to the best of the authors knowledge, none of these definitions have been used before when different objective functions are considered in the same network. In this chapter, we propose a metric to evaluate the overall network fairness in networks with multiple instances, which may have different objective functions.

5.2.2. Contributions

The main contributions of this chapter are:

- We address the performance and energy efficiency issues that may appear in RPL in the presence of heterogeneous traffic. Then, we propose a novel approach (C-RPL) that coordinates the RPL Instances to form energy efficient coalitions according to their individual objective functions and the network conditions.
- We propose a mechanism to evaluate fairness in networks with multiple RPL Instances. This evaluates the distribution of the existent network resources to address the different and sometimes contradictory objectives of each instance.

5.2.3. Organization of the Chapter

The remaining of this chapter is organized as follows: Section 5.3 provides some preliminary results to validate multi-tree routing approaches in real-life scenarios. Section 5.4, motivates and presents C-RPL, detailing the mechanism to create the C-RPL Instances. This also provides an illustrative example of C-RPL applied to a WSN with three different kinds of traffic. Section 5.5, proposes a metric to evaluate fairness in networks with multiple instances. Section 5.6 compares the performance of C-RPL (with and without the cooperation game) with RPL for different scenarios. Finally, Section 5.7 summarizes this chapter and presents the main conclusions.

Table 5.1. Instances Coexisting in the Network

Traffic	Objective Function
Positioning information	Maximum PDR
Ambient conditions & Battery	Minimum aggregated power
Fall detection	Minimum number of hops

5.3. Preliminary Results on Multi-Tree Routing

First of all, we evaluate the performance of a simple multi-tree routing strategy in a real scenario. The objective of this is section is to validate this approach in a commercial WSN platform, since to the best of the authors knowledge, it has not been implemented before.

We develop a habitat monitoring application that provides: i) the position of a mobile entity, ii) habitat environment conditions, and iii) movement and falling detection of the mobile entity, using an assembled accelerometer. In order to efficiently manage each traffic flow, multiple trees are constructed according to their main requirements, which are defined in Section 2.5 (See Table 5.1). The interested reader is referred to [Cor12] in order to find detailed information about the positioning algorithm of the mobile node.

The platform is composed of 14 MEMSIC IRIS sensors, which are equipped with the RF230 transceiver chip [Atm09]. The mobile node is attached to the mobile entity. This receives information from the rest of nodes, computes its own position, and transmits it to the nearest node. The sink node, which is connected to the computer, gathers the information coming from the network. The rest of nodes are distributed strategically to cover all the points needed for positioning and ambient conditions monitoring. Once the nodes have been deployed, a different tree is created for each traffic flow. Figure 5.1 is a representation of the application, showing the position of the mobile node (red spot), temperature information, battery status and alarms (movement and falling of the mobile entity). The ambient conditions measurements are sent every 2 seconds, the positioning

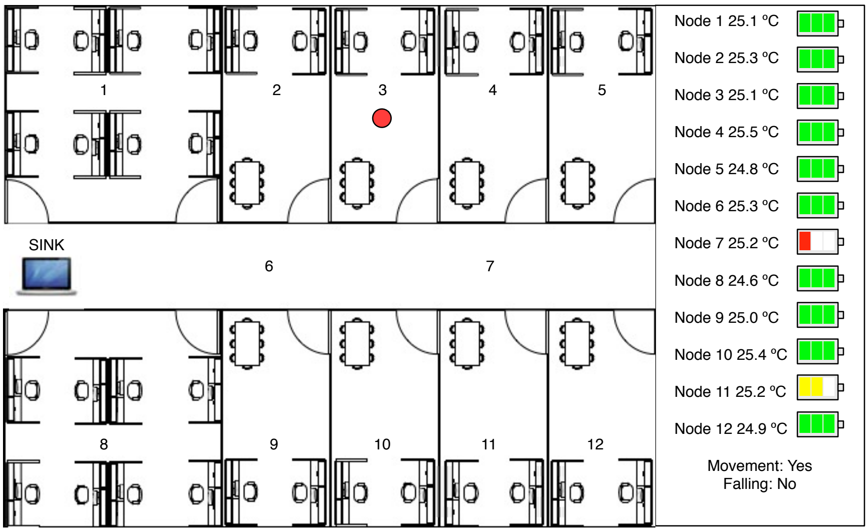


Figure 5.1. Example of a habitat monitoring application

Table 5.2. Experimental Results of a Multi-Tree Application

	Alarms	**Temp. & Battery**	**Positioning**
Hops (#)	1,25	1,92	2,25
AP (mW)	2,49	0,077	4,489
PDR (%)	85,60	65,26	95,20

messages every 0.5 seconds, and the event detection messages are only sent when the event occurs. The average results obtained through the experimental evaluation are presented in Table 5.2.

It can be observed that it is possible to construct a tree optimized for each kind of traffic. Using the proposed solution, the lowest delay and a reliability of 85.6% is guaranteed for the alarm messages. On the other hand, the ambient conditions monitoring has a very low consumption per packet associated. This is very important because of their periodicity. Therefore, the reduction of the aggregated power associated to these packets significantly extends the network lifetime. Finally, the positioning measures are received with a PDR higher than 95%, which is considered to fulfill the high reliability demands of positioning applications.

Although this simple multi-tree approach has been shown to be efficient managing multiple QoS requirements, we have assumed that all nodes participate in all trees. Note that in a real application, the tasks of each node should be defined according to the service needs, to avoid wasting energy resources. This enables the network to put nodes into sleep mode according to sleep scheduling strategies. In general, the radio transceivers of wireless sensors have four states: send, receive, idle and sleep. In the receive and send states nodes are sending or receiving information, respectively. In the idle state, nodes monitor the wireless channel for incoming packets, and in the sleep state nodes are inactive. Although, the transmit state consumes the highest amount of energy (e.g., the RF230 transceiver consumes 16.5 mA in the transmit state, and 15.5 mA in the receive and idle states [Atm09]), the idle consumption is the most relevant in WSNs, since wireless sensors generally spend much more time waiting for packets than transmitting or receiving them. For instance, a node that periodically transmits a packet per second, with a typical packet duration of 5 ms, spends around 99.5% of its time in the idle state. On the contrary, the sleep state consumes much less energy (e.g., the RF230 transceiver consumes 20 nA in this state). Since the idle consumption wastes important energy resources, we should reduce the time that wireless sensors remain unnecessarily active in order to increase the network lifetime.

In order to illustrate this issue, in Figure 5.2 we show the consumption spent in idle and send states of a WSNs (assuming the network parameters in [Bar13b]), using Matlab. In particular, we compare the consumption of a single-tree and a multi-tree strategy, with and without cooperation among trees. We can observe that the consumption for transmitting may be an order of magnitude lower than the idle consumption in a single-tree strategy. However, if we create multiple trees and allow them to cooperate, the idle consumption can be reduced without affecting the network performance, since the average active time of sensors is reduced.

In addition, we also need to consider the compatibility of the multi-tree strategy with the current networking standards. With this in mind, in Section 5.4 we develop C-RPL (Cooperative-RPL), which is a multi-tree routing approach build upon RPL.

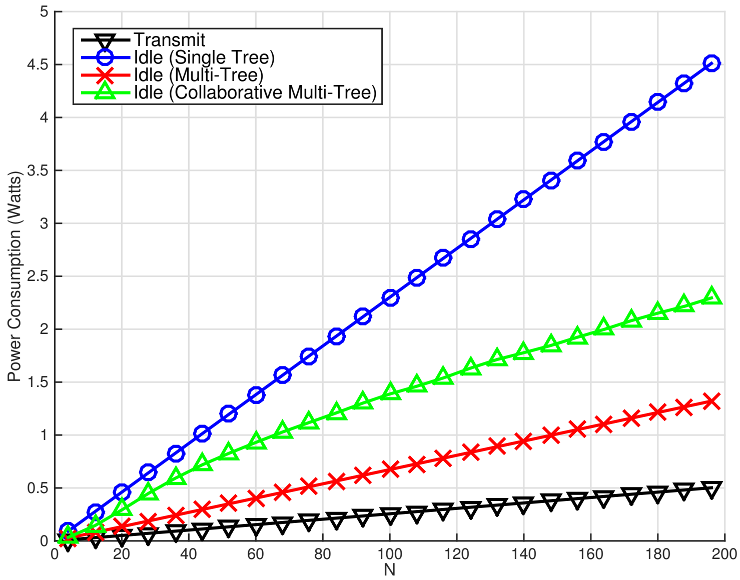


Figure 5.2. Comparison of the power consumption in transmit and idle states in a multi-tree network. The network parameters in [Bar13b] are assumed.

5.4. C-RPL: Cooperative - RPL

In order to efficiently reduce the active time of sensors and address the multiple QoS requirements of WSNs with heterogeneous traffic, we divide the network in multiple trees, which group nodes according to their tasks in the network, and make them collaborate only if it is necessary (See Figure 5.3). In particular, C-RPL uses the capability of RPL to construct multiple RPL Instances, and then creates coalitions among them according to their needs. This assumes selfish instances, which aim to maximize their performance with the minimum energy consumption.

We assume that nodes support slotted listening duty cycling [Tse02], which is a simple mechanism to reduce the idle energy consumption of the network. Basically, nodes periodically wake-up and remain fully operative for a specific time period. The percentage of time that a node is active defines its duty cycle, which must be set according to the QoS requirements, where high duty cycles are required for urgent data, while low duty cycles

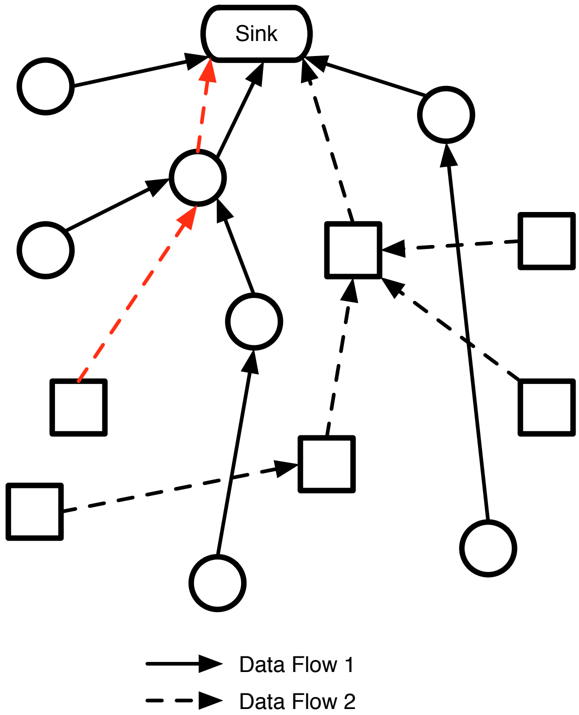


Figure 5.3. Example of a wireless sensor network divided in two different trees, which cooperate among them (red arrows).

may be used for non-critical data. Accordingly, we assume that each kind of traffic has its own duty cycle associated, and they are not overlapped in time.

In this section, we first discuss the issues of RPL with heterogeneous traffic that motivate C-RPL. Then, we explain how instances evaluate their utility related to each possible coalition. Finally, we explain the cooperation game among RPL Instances that is used to define the C-RPL Instances.

5.4.1. Problem Statement

RPL Instances operate independently from each other, and therefore their construction is also independent. The nodes that may belong to each instance need to be defined in advance, reducing the flexibility of the routing process. Then, a particular RPL Instance may be either composed of: i) the nodes with tasks associated with this instance, or ii) all the nodes in the network. In the first case, nodes only need to be active during the active time assigned to their respective instance or instances. In the second case, any node may belong to any instance as long as it senses or forwards traffic associated to that instance. Therefore, in this case nodes need to be active during the time assigned to its traffic, and also during the active time assigned to the rest of instances in which they participate. Clearly, the first strategy may reduce the network performance due to the reduction of the node density of each instance. On the other hand, the second strategy may waste energy resources because of including nodes with different duty cycle in the same DODAG. As a result, many nodes have to extend their active time, and thus increasing the average energy consumption of the network.

In C-RPL, the nodes that belong to each instance do not need to be predefined, and therefore C-RPL Instances can be constructed dynamically according to the objective function of each instance, the location of nodes and the particular network conditions. In general, C-RPL defines the nodes of each C-RPL Instance following a cooperative approach among the nodes that develop the same task in the network. The objective of each instance is to maximize its utility, taking into account the possible coalitions that can be formed with the rest of instances in the network. A cooperation strategy among instances is adopted, rather than a cooperation strategy among nodes, to prioritize the group interests in front of individual interests. Note that local decisions among nodes may also have an impact in the rest of the network. For example, if a node increases its transmission rate due to a collaboration with another node, it is also increasing the transmission rate of the nodes that forward its packets to the sink through the DODAG. This increases their energy consumption and it may even cause congestion problems in these nodes.

5.4.2. Rank Computation and Parent Selection

Following the RPL approach, each node selects its parent node using the concept of Rank, which is defined according to the objective function. In this sense, the main difference between both approaches is that in C-RPL nodes compute additional ranks to evaluate the possible coalitions that may be created in the network. These coalitions may be composed of one or multiple instances.

In general, the n-th node defines its Rank and parent related to each coalition and objective function, as:

$$r_{n,o}^{*,C} = \min_{k \in \mathcal{C}} r_{n,o}^{k,C} \quad \forall C \in \mathcal{P}, o \in \mathcal{O}, \tag{5.1}$$

$$p_{n,o}^{C} = \arg\min_{k \in \mathcal{C}} r_{n,o}^{k,C} \quad \forall C \in \mathcal{P}, o \in \mathcal{O}, \tag{5.2}$$

where C is the coalition, $\mathcal{C}$ is the set of nodes that belong to this coalition, o is the objective function, $\mathcal{P}$ is the set of possible coalitions in the network, and $\mathcal{O}$ is the set of objective functions.

In networks with more than three instances, we limit the cooperation game to the coalitions between two instances, and the coalition between all the instances (referred to as the grand coalition). This is to prevent the number of possible coalitions to grow exponentially with the number of instances. Note that when the node density is low, the instances will tend to create the grand coalition in order to improve their performance. On the other hand, when the node density is high enough, the instances will tend to create small coalitions in order to reduce their energy consumption. This is shown in Section 5.6.

In C-RPL, all nodes need to broadcast the information associated with each possible coalition. Currently, in RPL each DIO (DODAG Information Object) packet has a unique *RPLInstanceID* and *Rank* fields. This may constrain the construction of multiple RPL Instances since the signalling overhead has to be increased. In fact, the management of multiple instances has been identified as one of the open issues in RPL that needs to be addressed in the following versions [Win12]. Although this is out of the scope of this thesis, we propose a more flexible size of DIO packets, allowing to include multiple *RPLInstanceID* and *Rank* fields. Then, although the size of each DIO is slightly increased, the nodes do not need to broadcast a different DIO for each possible C-RPL

Instance.

5.4.3. Coalition Utilities

The creation of larger C-RPL instances may increase the performance of the network. However, since coalitions are energy consuming, a utility function is used to decide which coalitions, if any, are created.

When two instances agree to collaborate, they both share their nodes. Since they have different duty cycles, the nodes from different instances must coordinate their active times in order to communicate. In a slotted listening duty cycle scheme, this means that either the transmitter or the receiver must remain active during the active time assigned to the other instance, increasing its duty cycle. We define that the node that adjusts its duty cycle in C-RPL is always the transmitter. On the other hand, the nodes forwarding packets from other instances increase their transmission probability, since they have to transmit additional packets. In general, both instances increase their energy consumption by forming a coalition due to the additional duty cycle of their nodes. Then, there exists a tradeoff between performance and energy consumption that all instances need to evaluate before forming a coalition. The performance of each node is characterized by its Rank, and its energy consumption is estimated taking into account its duty cycle and average transmission rate in that coalition. In particular, we consider the utility of each node in the coalition C, for a given objective function o, using linear aggregation as:

$$u_{n,o}^{C} = -\left(\rho \bar{r}_{n,o}^{*,C} + (1-\rho)\bar{e}_{n,o}^{*,C}\right), \tag{5.3}$$

where $\bar{r}_{n,o}^{*,C}$ and $\bar{e}_{n,o}^{*,C}$ are the mean-variance normalization of $r_{n,o}^{*,C}$ and $e_{n,o}^{*,C}$, respectively. This is applied to homogenize the Rank and energy consumption values. Note that the utility is inversely proportional to both Rank and energy consumption. A weighted linear utility has been chosen, since it allows the network designer to easily adjust the willingness of nodes to increase their Rank at the expenses of increasing their energy consumption using the cooperation parameter ρ. This must be configured according to the application requirements.

Then, the instance I computes the utility of the coalition C as the minimum utility, when its associated objective function o_I is considered, among the nodes in $\mathcal{I}$ (i.e., set of nodes that belong to this instance):

$$U^C_{I,o_I} = \min_{n \in \mathcal{I}} u^C_{n,o_I}. \tag{5.4}$$

Note that the minimum utility is considered to avoid unfair solutions among nodes, such as instances that overload a particular node.

5.4.4. Cooperation Game among RPL Instances

Once the utilities of each possible coalition have been computed, the instances must decide which coalitions are created in order to form the C-RPL Instances. We model instances as selfish entities that aim to maximize their performance with the minimum energy consumption. In this section, we show that without any coordination, the instances would never agree to cooperate among them.

Following a game theoretical approach, we can model this game similarly to the prisoner's dilemma game [Axe81] (Table 5.3). This game defines R (reward) as the utility when both players cooperate, S (sucker) as the utility when a player cooperates but the other player does not, T (temptation) stands for the utility when a player does not cooperate but the other player does. Finally, if neither cooperate, they both obtain a reward of P (punishment). The relation among them is $T > R > P > S$.

Table 5.3. Prisoner's Dilemma Game

		Player 2	
		Cooperate	Not Cooperate
Player 1	Cooperate	R_1,R_2	S_1,T_2
	Not Cooperate	T_1,S_2	P_1,P_2

Comparing the prisoner's dilemma game with the cooperation game among two RPL Instances (Table 5.4), we can observe that they are very similar. In this game, each instance is a player, where I_1 is the instance that requests the collaboration and I_2 is the instance that decides whether to collaborate or not. U^c is the utility when an instance cooperates (i.e., shares its nodes), and U^n when it does not. Note that the *Not Cooperate - Cooperate* case is not considered here because this would mean that I_2 accepts a collaboration from I_1 that has not been requested.

Table 5.4. Cooperation Game among RPL Instances

		I_2	
		Cooperate	Not Cooperate
I_1	Cooperate	U_1^c,U_2^c	U_1^c,U_2^n
	Not Cooperate	-	U_1^n,U_2^n

Since the utility function considers both the performance and the energy consumption, U^c can be either higher or lower than U^n. However, analyzing this game, we can find that the Nash equilibrium is the *Not Cooperate - Not Cooperate* solution. This is because either if I_2 is willing to cooperate or not, I_1 always increases its utility by defecting to cooperate, standing that I_2 cooperates (i.e., the Rank remains equal and the consumption is lower). As a result, even if I_2 accepts to cooperate, I_1 will also defect to cooperate because in this new situation its utility is higher if it does not share its nodes. Since this may not be the solution that maximizes their utilities, it is necessary to mediate in the construction of coalitions.

In C-RPL, the cooperation game is managed by the instances, but supervised by the sink. The closest node of each instance to the sink (referred to as instance head), computes the utility of its respective instance for each possible coalition. Then, each instance head finds the best coalition C^* and defines the C-RPL Instance, broadcasting this decision to the rest of nodes, which set their parent nodes accordingly. In case that the rest of instances in C^* do not join the coalition (i.e., this is not their best coalition), the instance head updates C^* and broadcast it again to the rest of nodes, creating a different C-RPL Instance. Instance heads periodically evaluate if their C^*

Table 5.5. Instances Coexisting in the Network

Instance	Traffic	Objective Function
I_H	Event detection	H: Min. Hops
I_E	Non-critical monitoring	E: Min. ETX
I_P	Critical monitoring	P: Max. PDR

is still the best possible coalition. On the other hand, the coalitions are supervised by the sink, which has knowledge about the performance that each instance should obtain. Using these values as a reference, the sink can detect if any instance in a coalition does not collaborate. In order to punish this instance, the sink should not acknowledge its packets. In this case, that instance would have to collaborate again in order to continue receiving ACKs. This is summarized in Figure 5.4.

5.4.5. Example of C-RPL

Finally, we present an illustrative example of C-RPL in a network with three instances, which manage three different kinds of traffic. In order to cover the main kinds of traffic that are present in WSNs (see Section 2.5), we consider event detection, non-critical monitoring and critical monitoring traffic. A different objective function is defined for each traffic according to their main requirements (i.e., latency, consumption and reliability). In particular, in this example we consider the instances in Table 5.5.

The computation of Ranks is related to their objective function. Then, each node computes its own Ranks as follows:

$$r_{n,H}^{*,C} = \min_{k \in \mathcal{C}} r_{n,H}^{k,C} = \min_{k \in \mathcal{C}} \left(Hops_n^k + r_{k,H}^{*,C} \right) \quad \forall C \in \mathcal{P}, \tag{5.5}$$

$$r_{n,E}^{*,C} = \min_{k \in \mathcal{C}} r_{n,E}^{k,C} = \min_{k \in \mathcal{C}} \left(ETX_n^k + r_{k,E}^{*,C} \right) \quad \forall C \in \mathcal{P}, \tag{5.6}$$

$$r_{n,P}^{*,C} = \min_{k \in \mathcal{C}} r_{n,P}^{k,C} = \min_{k \in \mathcal{C}} \left(\frac{1}{PDR_n^k} r_{k,P}^{*,C} \right) \quad \forall C \in \mathcal{P}. \tag{5.7}$$

The value of $Hops_n^k$, ETX_n^k or PDR_n^k is estimated by the n-th node using the DIOs

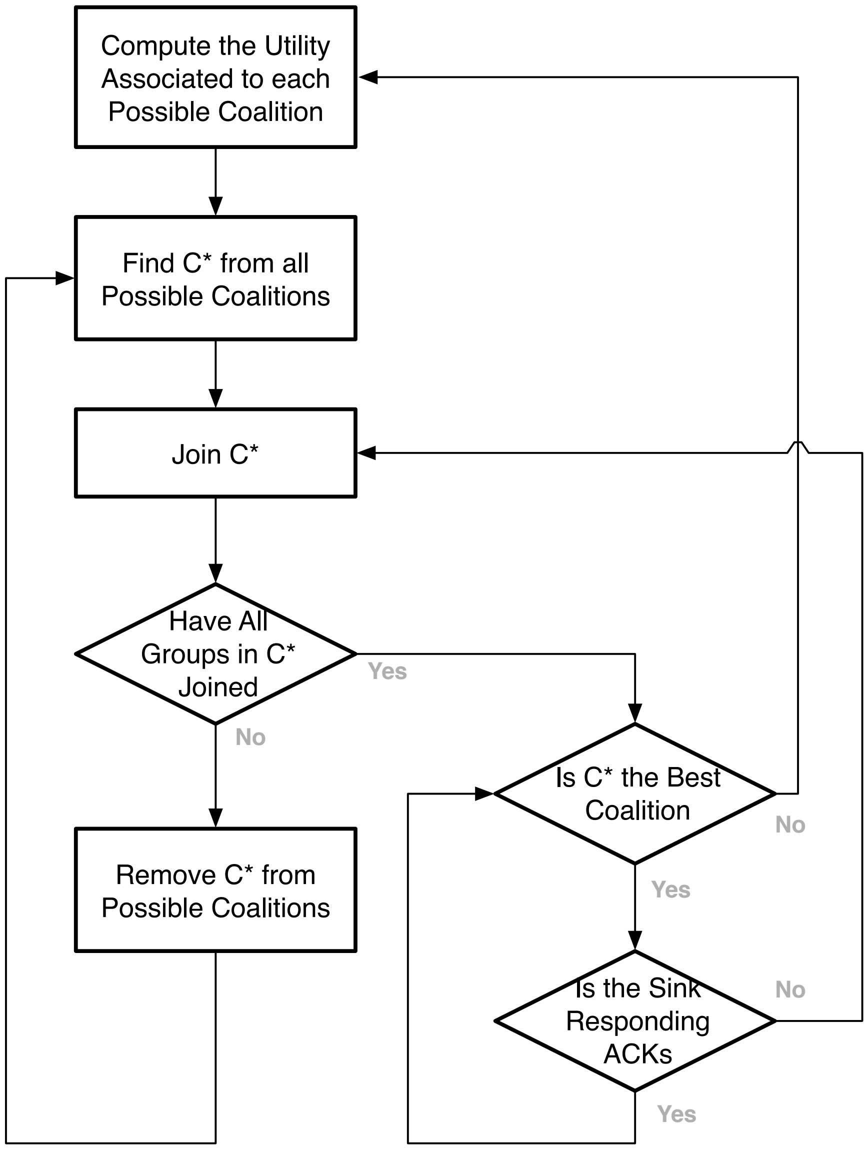


Figure 5.4. Block diagram of the construction of C-RPL Instances

coming from the k-th node [Win12]. On the other hand, the values of $r^{*,C}_{k,H}$, $r^{*,C}_{k,E}$ or $r^{*,C}_{k,P}$ are broadcast by the k-th node within its DIO packets. Note that Ranks increase in the direction of the leaf nodes and decrease in the direction of the sink node.

Figure 5.5 illustrates the possible coalitions evaluated by I_E. This instance evaluates the C-RPL Instance constructed using only its own nodes (Figure 5.5a), the C-RPL Instances constructed in coalition with each other group: $\{I_E, I_P\}$ and $\{I_E, I_H\}$ (Figure 5.5b), and also the C-RPL Instance that groups the whole network: $\{I_E, I_P, I_H\}$ (Figure 5.5c). For each of these possible coalitions, the n-th node in I_E computes a different Rank:

$$r^{*,1}_{n,E} = \min_k r^{k,1}_{n,E} \quad \forall k \in I_E, \tag{5.8}$$

$$r^{*,2}_{n,E} = \min_k r^{k,2}_{n,E} \quad \forall k \in \{I_E, I_P\}, \tag{5.9}$$

$$r^{*,3}_{n,E} = \min_k r^{k,3}_{n,E} \quad \forall k \in \{I_E, I_H\}, \tag{5.10}$$

$$r^{*,4}_{n,E} = \min_k r^{k,4}_{n,E} \quad \forall k \in \{I_E, I_P, I_H\}. \tag{5.11}$$

On the other hand, the rest of the instances also evaluate all their possible coalitions. Then, the nodes in I_E compute some additional Ranks, in this case using the metrics of the other instances. In particular, the Rank using the nodes in I_E and the nodes of each of the other instances: $\{I_E, I_P\}$ and $\{I_E, I_P\}$ (Figure 5.6.a), and also the Rank using the whole network considering each respective metric $\{I_E, I_P, I_H\}$ (Figure 5.6b and Figure 5.6c). Then, the n-th node of I_E computes the following additional Ranks:

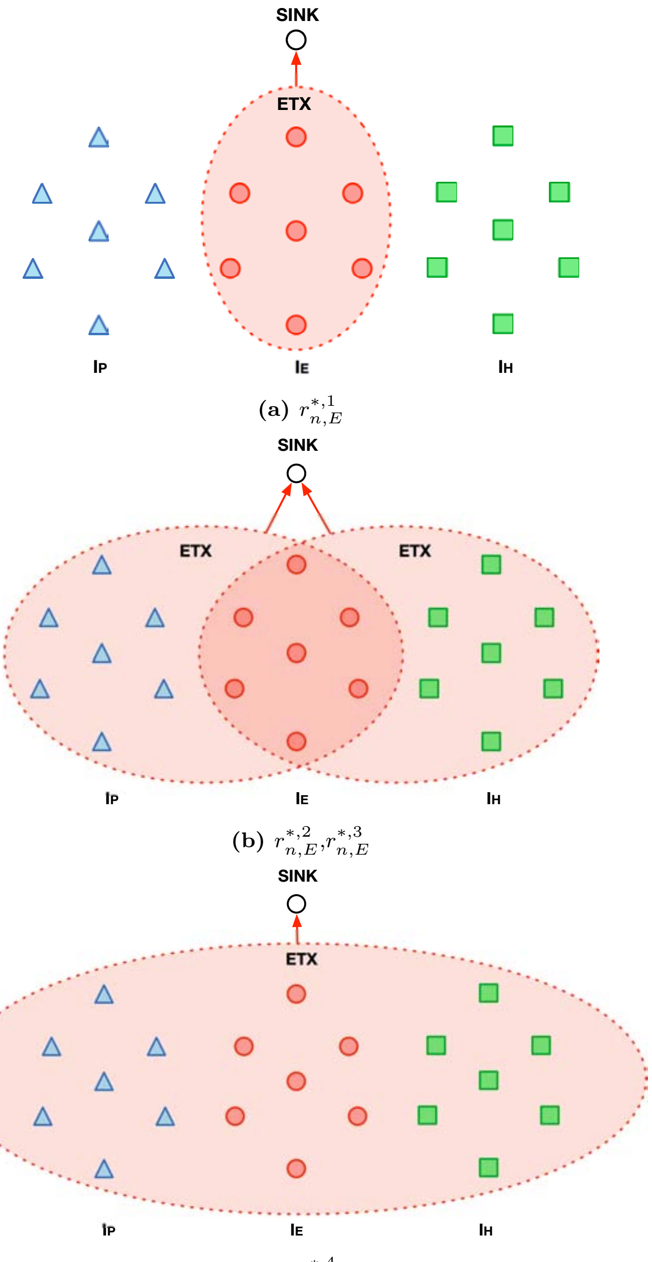
$$r^{*,5}_{n,P} = \min_k r^{k,5}_{n,P} \quad \forall k \in \{I_E, I_P\}, \tag{5.12}$$

$$r^{*,6}_{n,H} = \min_k r^{k,6}_{n,H} \quad \forall k \in \{I_E, I_H\}, \tag{5.13}$$

$$r^{*,7}_{n,P} = \min_k r^{k,7}_{n,P} \quad \forall k \in \{I_E, I_P, I_H\}, \tag{5.14}$$

$$r^{*,8}_{n,H} = \min_k r^{k,8}_{n,H} \quad \forall k \in \{I_E, I_P, I_H\}. \tag{5.15}$$

Summarizing, Ranks $r^{*,1}_{n,E}$ to $r^{*,4}_{n,E}$ are used by I_E to evaluate the possible coalitions that



(a) $r^{*,1}_{n,E}$

(b) $r^{*,2}_{n,E}$,$r^{*,3}_{n,E}$

(c) $r^{*,4}_{n,E}$

Figure 5.5. Possible coalitions evaluated by I_E. Circles indicate the nodes available to construct the C-RPL Instance. The color of the circle indicates the Rank metric used to evaluate the coalition

can be formed with I_P and I_H, and are computed using the metric ETX. On the other hand, Ranks $r_{n,P}^{*,5}$ to $r_{n,H}^{*,8}$ are used to inform the rest of instances about the performance that they would obtain by creating a coalition with I_E. Note that these are computed using the objective functions of the other instances, which are the minimum number of hops and the maximum end-to-end PDR, in this case.

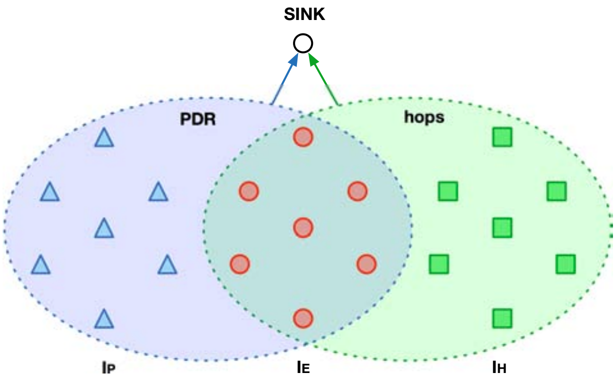
5.5. Fairness in Networks with Multiple Instances

The collaboration among instances can improve their performance, but it also increases their energy consumption. Without control, these collaborations may overload a particular instance, depleting the batteries of its nodes too fast. This situation may cause the loss of specific sensing capabilities. On the other hand, it is also important that these collaborations try to satisfy all their different requirements. Therefore, it is important that the available resources are shared in a “fair” manner among instances. Nevertheless, it is important to notice that the objective here is not to distribute the resources uniformly, since instances perform different tasks with different energy consumption demands.

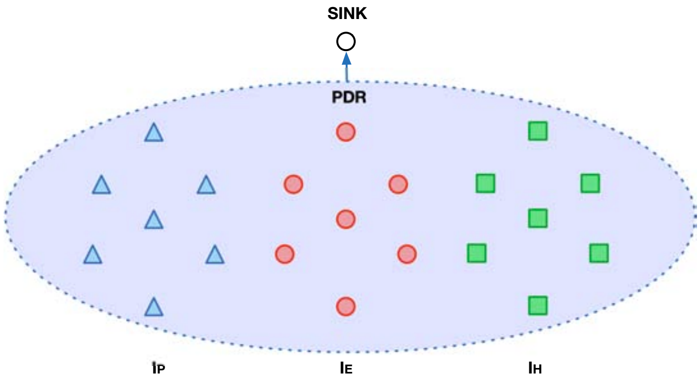
In order to evaluate fairness, many definitions can be found in the literature (e.g., weighted fairness, max-min fairness, proportional fairness) [Shi14]. However, to the best of the authors knowledge, not any of these definitions has been used before to evaluate fairness in networks with multiple instances. In this chapter, we propose to use the following metric:

$$F = \min_{I} \left(\frac{1}{|\mathcal{I}|} \sum_{n \in \mathcal{I}} (x_n - x_n^*) \right) \quad \forall I \in \mathcal{K}, \tag{5.16}$$

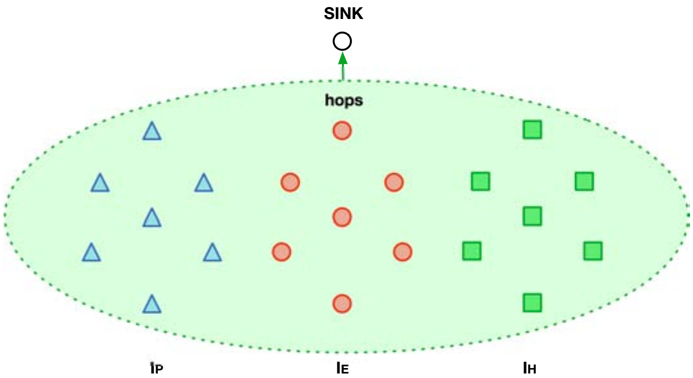
where $\mathcal{K}$ is the set of instances in the network, x_n is the utility achieved by the n-th node, and x_n^* is the optimum utility of this node. Since we evaluate fairness among instances, the average deviation of the nodes in each instance is computed. Note that the maximum value of fairness is found when $x_n = x_n^*$ $\forall n$ (i.e., x_n is always equal or lower than x_n^*). Then, F is always zero or negative.



(a) $r^{*,5}_{n,P}$



(b) $r^{*,6}_{n,H}$,$r^{*,7}_{n,P}$



(c) $r^{*,8}_{n,H}$

Figure 5.6. Possible coalitions that involve I_E evaluated by other instances. Circles indicate the nodes available to construct the C-RPL Instance. The color of the circle indicates the Rank metric used to evaluate the coalition.

In order to consider the tradeoff between performance and energy consumption, we evaluate fairness in both senses. In particular, the fairness in terms of Rank (F_r) and energy consumption (F_e) are defined as follows:

$$F_r = \min_I \left(\frac{1}{|\mathcal{I}|} \sum_{n \in \mathcal{I}} \left(\bar{r}^{*,*}_{n,o_I} - \bar{r}^{*,C^*}_{n,o_I} \right) \right) \quad \forall I \in \mathcal{K}, \tag{5.17}$$

$$F_e = \min_I \left(\frac{1}{|\mathcal{I}|} \sum_{n \in \mathcal{I}} \left(\bar{e}^{*,*}_{n,o_I} - \bar{e}^{*,C^*}_{n,o_I} \right) \right) \quad \forall I \in \mathcal{K}, \tag{5.18}$$

where o_I is the objective function of I, C^* is the coalition selected by I, and $\bar{r}^{*,*}_{n,o_I}$ and $\bar{e}^{*,*}_{n,o_I}$ are the optimum mean-variance normalized Rank and energy consumption of the n-th node, respectively. These are the best values among all the possible coalitions in terms of Rank and energy consumption.

This metric evaluates how the network resources are balanced among the instances in the network, taking into account the deviation of each node from its best possible solution. In particular, a very low value in F_r means that there is an instance that is not receiving enough resources, while a low value in F_e means that the network is overloading a particular instance in terms of energy consumption. Since we want to avoid both extreme situations, it is important to consider the tradeoff between both metrics.

5.6. Simulation Results

In this section, we compare the performance of C-RPL and RPL in a WSN with heterogeneous traffic using Matlab. In particular, we evaluate their performance for different ρ values (i.e., cooperation parameter), node densities and traffic loads. Finally, we compare each strategy in terms of fairness.

5.6.1. Simulation Environment

In RPL, the nodes that may belong to each instance must be defined in advance. In this context, we consider two different versions of RPL, which construct their RPL Instances using different criteria. In the first version, simply referred to as RPL, RPL Instances may be composed of any node in the network, regardless of its specific tasks. In the second version, referred to as RPL II (RPL with Independent Instances), nodes belong only to the instance related to its tasks, and therefore to a unique RPL Instance. For the sake of comparison, we also include a second version of C-RPL, referred to as C-RPL NCG (C-RPL with No Cooperation Game), in which each instance constructs the C-RPL Instance that maximizes its utility independently. Note that this strategy is not equivalent to RPL, since this considers not only the Rank, but also the energy consumption of each possible C-RPL Instance using (5.4). This has been included to show the lack of fairness that may arise without the cooperation game among instances.

We assume a slotted listening duty cycling mechanism that adjusts the duty cycle of each node according to its tasks in the network. Therefore, nodes remain active only if they are sensing, sending or expecting to receive data. Moreover, in order to better observe the impact of each routing strategy in the network performance, we assume that there are no collisions among packets.

The simulation scenario considers a total of N sensing nodes, randomly deployed in a 150x150 m^2 area. The information collected by them is gathered at the sink node, which is located in the middle of the sensing area.

The path losses (PL) are modeled following the log-distance path loss model:

$$PL(dB) = PL_0(dB) - 10\alpha log(d) + \gamma, \tag{5.19}$$

where PL_0 is the path loss at the reference distance d_0 (1 m), α is the path loss exponent, d is the communication distance, and γ is the attenuation caused by flat fading, which has zero mean and variance σ^2. The value of PL_0 has been empirically measured using IRIS motes (i.e., PL_0= 50 dB). A path loss exponent α=3 [Pu12] and a path loss variance σ^2=6 [Rap01] are assumed. Moreover, we assume the sensitivity value (-91 dBm), the maximum transmission power (3 dBm) and the current consumptions in sleep, idle and transmit states (20 nA, 15.5. mA and 16.5 mA, respectively) of the IRIS transceiver (i.e., RF230 [Atm09]). Table 5.6 summarizes the main simulation parameters.

Table 5.6. Simulation Parameters

Parameter	**Value**
General	
Sensing area (L x L)	150 x 150 m^2
Path loss at 1 m (PL_0)	50 dB
Path loss exponent (γ)	3
Flat fading variance (σ^2)	6
Packet size	127 bytes
Radio Transceiver	
Maximum data rate	250 kbps
Current consumption:	
Transmit state (I_{tx})	16.5 mA
Idle state (I_{idle})	15.5 mA
Sleep state (I_{sleep})	20 nA
Transmission power	3 dBm
Sensitivity	-91 dBm
Supply voltage (V_{DD})	3 V

In particular, the sink collects event detection, critical monitoring and non-critical monitoring measurements, coming from the wireless sensors, which generate up to 250 kbps. The size of packets is assumed to be 127 bytes (i.e., standard packet size in IEEE 802.15.4). Each kind of data is sensed by a different instance, which are composed of the same number of nodes. Since instances may sense different magnitudes, the traffic originated at different instances cannot be generally aggregated. On the other hand, the traffic from the same instance can be compressed using data aggregation techniques, since they may be spatially and temporally correlated. In particular, we assume a compression rate of 80% [Cap12] (i.e., nodes can combine up to 5 packets into a single packet on average). The duty cycle, the average packet transmission rate and the objective function of each instance are defined according to the particularities of each kind of data (See Table 5.7).

Table 5.7. Traffics Managed by the Network

Traffic	Duty Cycle	Packets/s	Objective
Event detection	20 %	0.01	Min. Hops
Non-critical monitoring	5 %	0.1	Min. ETX
Critical monitoring	10 %	0.2	Max. PDR

5.6.2. Cooperation Parameter

In Figures 5.7 to 5.9, we show the impact of the cooperation parameter ρ in the network performance, in terms of the particular objective function of each instance. Although individual ρ values could have been considered for each node or instance, we assume a uniform ρ, for the sake of simplicity. Each instance is composed of 40 nodes. In general, we can observe that instances do not cooperate until a minimum ρ is considered. This is because the network prioritizes the energy consumption in front of Rank (i.e., Hops, ETX and PDR) when using low ρ values. On the other hand, when the cooperation parameter is increased, C-RPL and C-RPL NCG tend to the RPL solution, since instances are more prone to collaborate with other instances, as long as they increase their performance. We can also observe that without the cooperation game, the instances start creating larger C-RPL Instances using lower ρ values. This is because these are decided individually by each instance. In this case, the best performance is always found using RPL, but in situations with congestion problems this may not be always the case, as we discuss in Section 5.6.4.

In Figure 5.10, the average energy consumption is shown for different ρ values. As we have observed before, the network prioritizes the energy consumption if low ρ values are considered. The average power consumption in C-RPL increases with ρ, from 5.5 mW to 8.5 mW, due to the additional communication among instances. This causes that more nodes increase their duty cycles, and therefore their energy consumption also increases.

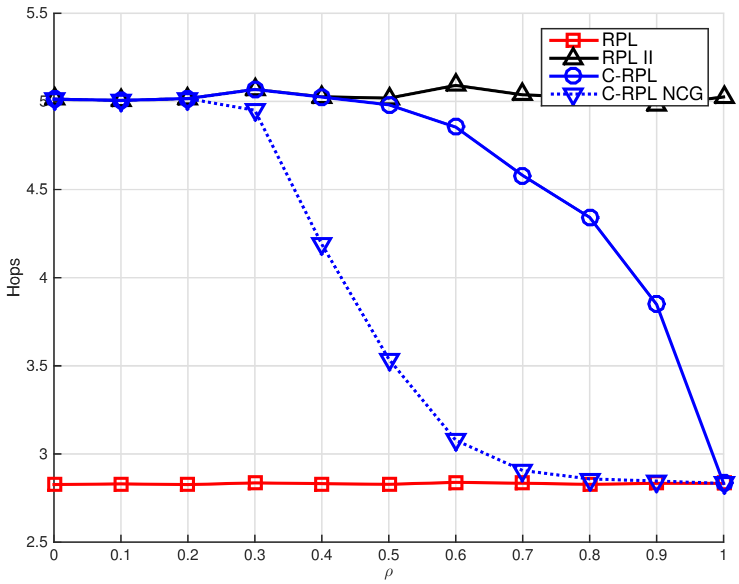


Figure 5.7. Average number of hops of I_H.

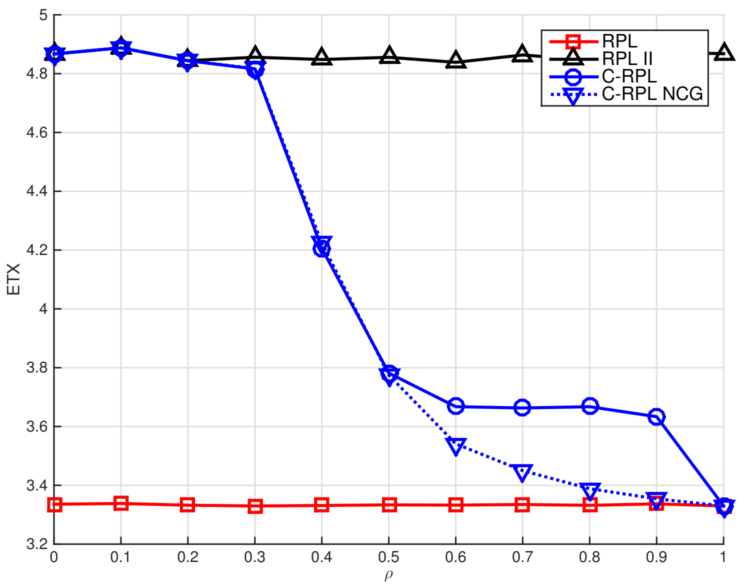


Figure 5.8. Average ETX of I_E.

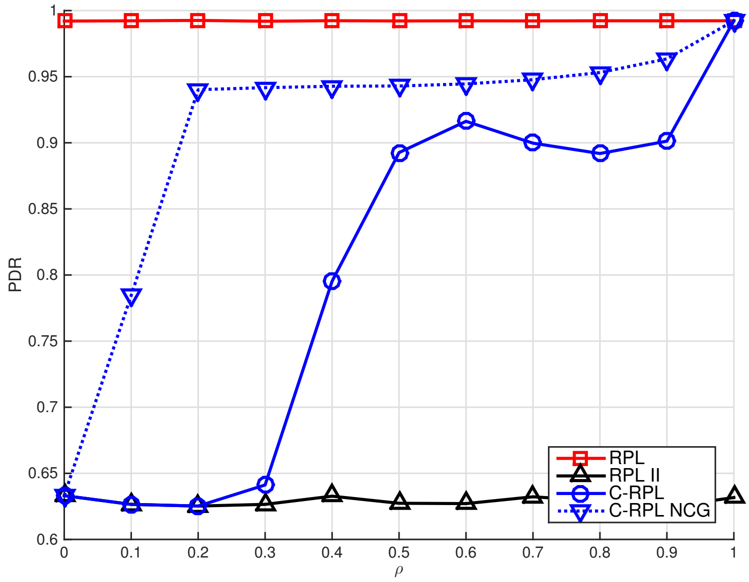


Figure 5.9. Average PDR of I_P.

For the same reason, C-RPL NCG, has a higher consumption than C-RPL. In fact, this may be even higher than in RPL.

5.6.3. Impact of Node Density

In Figures 5.11 to 5.13, we evaluate the performance of RPL and C-RPL for different network densities, considering ρ=0.9 and the same number of nodes for each instance ($N/3$). In networks with a low density of nodes, the best performance is achieved by RPL. On the contrary, when instances are independent from each other (i.e., RPL II), the number of possible routes is reduced, and therefore their performance is lower. When the density of the RPL Instances increases, RPL II improves its performance, since the communication distances are shorter. In general, the performance of C-RPL is an intermediate value between both solutions. Note that the performance of each instance could be individually adjusted with individual ρ parameters.

As we can observe in Figure 5.13, a node density above 130 nodes generates network

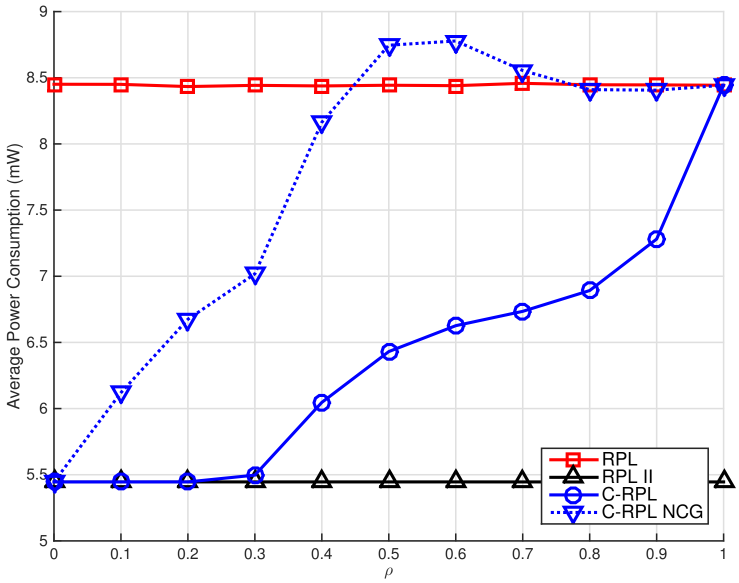


Figure 5.10. Average power consumption (mW) of the network for different ρ values.

congestion in RPL, reducing the network performance. Note that I_P is the instance that generates the highest amount of traffic, and therefore it is congested at lower node densities than the rest of instances. However, the impact of node density is different in C-RPL, since C-RPL avoids creating C-RPL Instances with overloaded nodes. We discuss the network congestion problem in the next section.

In general, C-RPL reduces the average number of coalitions with the node density, since instances do not need to create large C-RPL Instances to have a good performance. As a result, the average energy consumption of C-RPL can be always lower than RPL. This can be observed in Figure 5.14, where C-RPL reduces the average power consumption of RPL around 17%.

5.6.4. Impact of Traffic Load

The packet rates in the previous sections were not high enough to cause congestion problems with 120 nodes. However, bottlenecks may frequently appear in convergecast

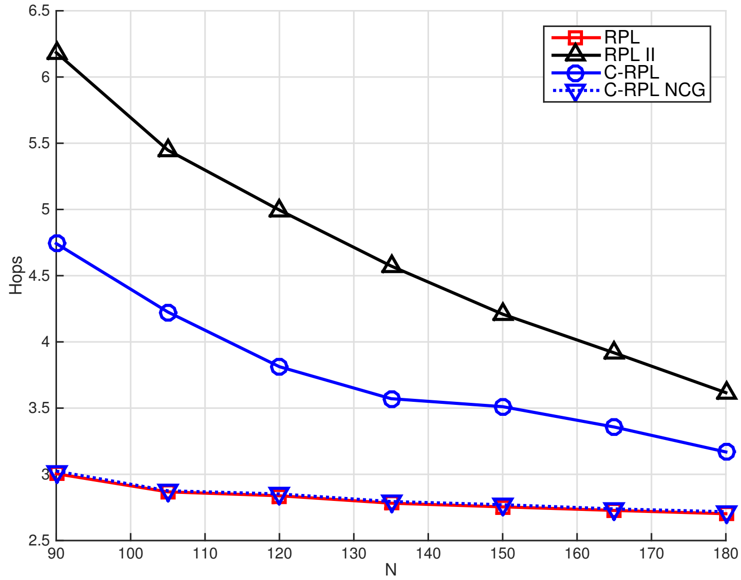


Figure 5.11. Average number of hops of I_H.

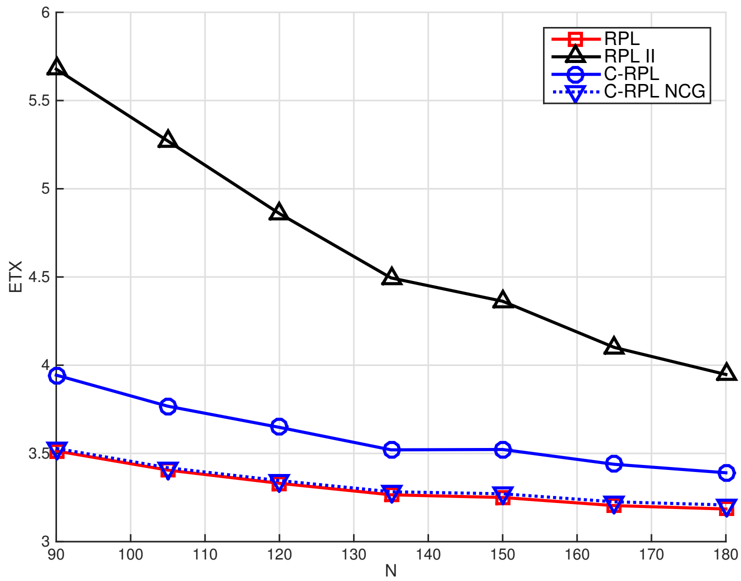


Figure 5.12. Average ETX of I_E.

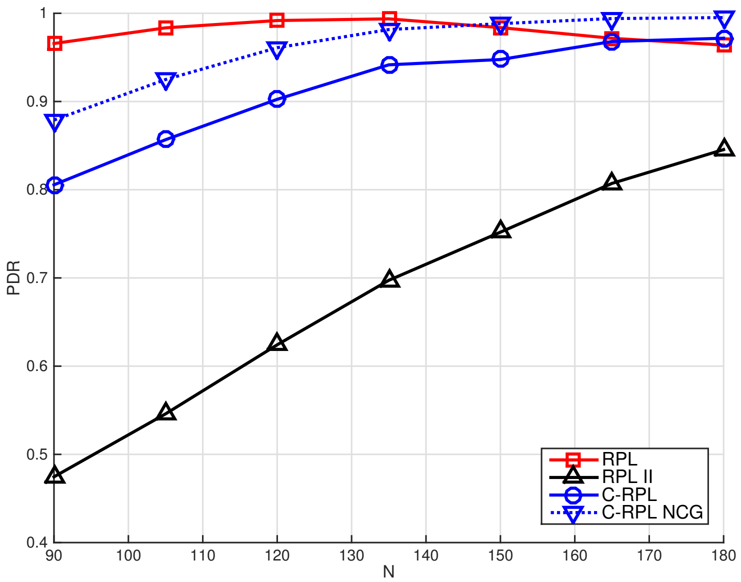


Figure 5.13. Average PDR of I_P.

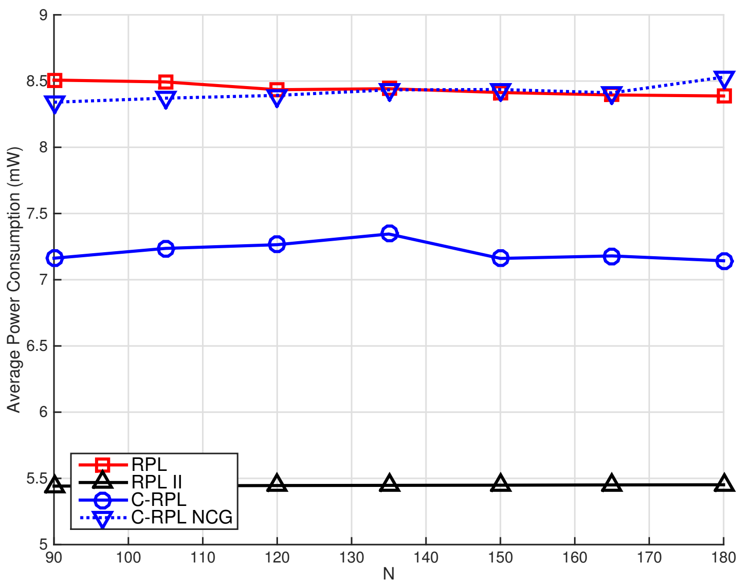


Figure 5.14. Average power consumption (mW) for different node densities.

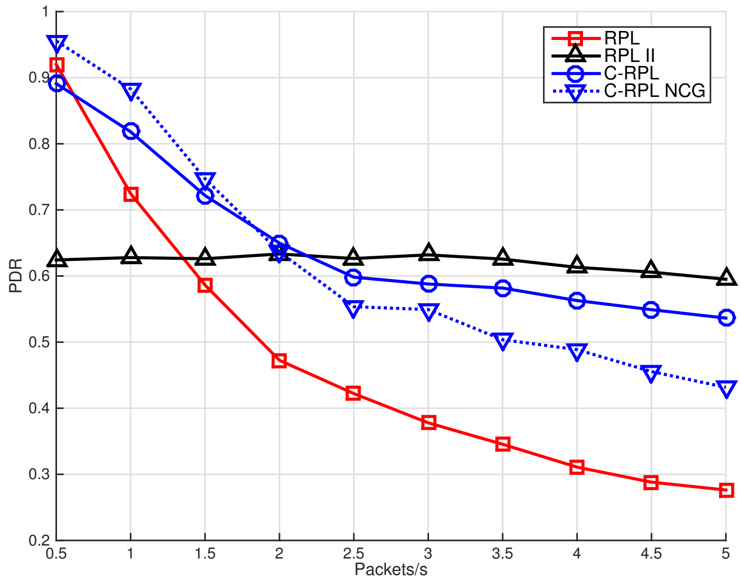


Figure 5.15. Average PDR of I_P for different packet rates.

networks when the traffic is moderate, since the traffic is many-to-one. In this case, some nodes may not be able to forward all packets during their corresponding duty cycle. In Figure 5.15, we show the impact of traffic congestion using different packet rates in I_P, while the packet rates of the rest of instances are not modified. We can observe that RPL is the most affected strategy due to the higher number of communications among nodes from different instances. As a result, a large amount of traffic cannot be aggregated, causing congestion problems in the network. For example, when each node in I_P generates 4 packets/s, the PDR is around 0.3. On the other hand, when instances are completely independent, the traffic can always be aggregated and therefore its impact is much lower (above 0.6 in the same case).

C-RPL manages this issue adjusting the cooperations according to the traffic conditions. When the traffic is low, the instances tend to form larger C-RPL Instances to achieve a better performance by collaborating among them. However, when the traffic is high enough to cause congestion problems, the C-RPL Instances tend to be smaller in order to reduce the amount of traffic that cannot be aggregated.

5.6.5. Fairness

In applications with multiple sets of requirements, the network must not be focused in one specific task. Instead, it must distribute the available resources in a fair way to satisfy the requirements that each traffic demands. In Figures 5.7 to 5.14 we have shown that C-RPL and C-RPL NCG are intermediate solutions among RPL and RPL-II in terms of performance and energy consumption, which can be adapted using ρ. In this section, we show that C-RPL distributes the network resources more fairly than RPL. Figure 5.16 shows the fairness of each strategy in terms of energy consumption and Rank for different values of ρ. As we can observe, using RPL II the consumption fairness is close to zero, but the worst Rank fairness is obtained (around -1.17). On the other hand, using RPL we can obtain a better Rank fairness, but this is much worse in terms of consumption fairness (around -1.45). We can also observe here that using C-RPL we can obtain intermediate solutions according to the ρ parameter. For instance, using $\rho = 0.8$ C-RPL obtains a fairness around -0.75, both in terms of Rank and consumption. Note that intermediate solutions mean that the resources are fairly shared, and not any instance is overloaded.

In this figure, we can also compare the fairness degree of C-RPL when the cooperation game is applied or not. In general, we can observe that C-RPL avoids solutions with a low consumption fairness thanks to the cooperation game. On the other hand, C-RPL NCG tends to provide a better Rank fairness, regardless of its consumption fairness.

5.7. Summary and Conclusions

In this chapter, we have presented C-RPL, a multi-tree routing approach to deal with the traffic heterogeneity in WSNs. The objective is to adapt the active time of wireless sensors according to the network needs, and also satisfy the particular QoS requirements of each traffic flow. As we did in the previous chapters, we base our strategy on the

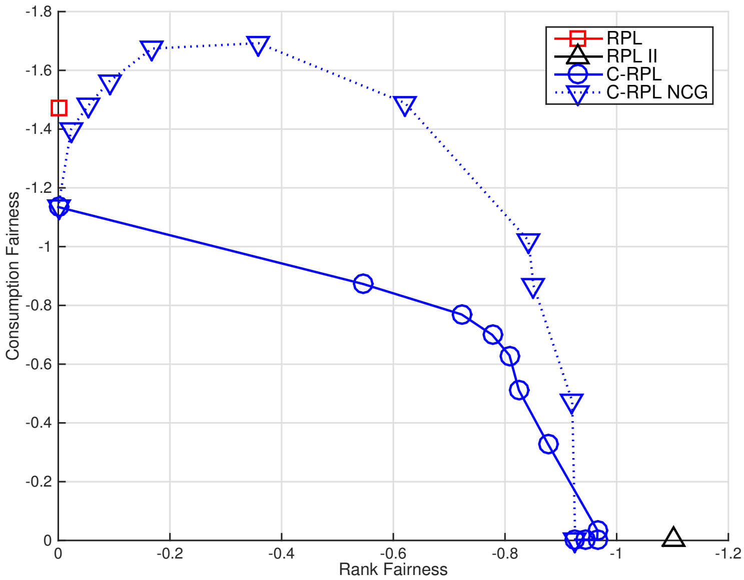


Figure 5.16. Rank and consumption fairness for $\rho = 0$, 0.5, 0.6, 0.7, 0.8, 0.9 and 1, where the smallest ρ is located on the right bottom side.

RPL protocol for the sake of compatibility with the current networking standards. In particular, we use the capability of RPL to create virtual subnetworks, referred to as RPL Instances, which can be constructed with different objective functions. A multi-tree strategy has been adopted due to its simplicity and compatibility with the RPL protocol. This is particularly interesting in WSNs managing traffics with opposite sets of QoS requirements, and therefore they need to be addressed individually. Note that otherwise, multi-objective routing could be applied.

C-RPL creates energy efficient instances, referred to as C-RPL Instances, following a cooperative strategy among nodes with different sensing tasks. Since nodes would never agree to cooperate, C-RPL coordinates them to create coalitions that improve the trade-off between their own performance and energy consumption using the cooperation parameter ρ. This can be individually defined for each DODAG according to the application requirements.

The simulation results show that C-RPL adjusts the C-RPL Instances according to the individual objective function of each traffic flow and the current network conditions. In particular, we have observed that C-RPL tends to create large instances when the node density is low. Otherwise, it constructs smaller instances to reduce the energy consumption of the network and avoid congestion problems. Besides, we have proposed a fairness analysis for networks with multiple instances. This measures the distribution of the network resources when multiple objective functions coexist in the network. The results show that C-RPL obtains a more balanced fairness than RPL-based approaches with non-cooperative instances.

Chapter 6

IoT-Cloud Formulation to Integrate WSNs into the Future Internet of Things

6.1. Summary

The confluence of distributed cloud networking and the IoT will enable a new range of services in "smart" environments (e.g., smart cities, smart grids, smart transportation systems). Wireless sensors are planned to be a fundamental part in this new paradigm and therefore their efficient integration is critical. On one hand, the IoT-Cloud augments the computing resources of sensors and extends their battery life. On the other hand, the data gathered by different sensing platforms can be shared on a bigger scale, enabling a more efficient exploitation of the physical infrastructure.

In Chapters 3, 4 and 5, WSNs were assumed to be isolated networks that merely collect specific data to be gathered at a central server. However, thanks to the recent advances in network programability and virtualization, wireless sensors can be integrated into IoT-Cloud networks as distributed sensing and computing resources. In this chapter, we formulate this problem as a minimum cost flow problem using only linear constraints. The objective is to the find the optimal placement of virtual functions over the IoT-Cloud that meets user requests, satisfies network resource capacities, and minimizes overall network cost. We solve this problem for an illustrative set of smart city services,

where users interact with the city using their smart devices.

6.2. Introduction

6.2.1. Motivation and Previous Work

The Internet of Things (IoT) connects many heterogeneous devices, such as wireless sensors, RFID devices, smartphones, wearables and connected vehicles, which produce and/or consume information in real-time. When connecting the IoT to the Cloud, information collected from multiple locations can be processed and analyzed to produce meaningful information, which can be accessed remotely [Ala13]. At the same time, the intrinsic limitations of lightweight mobile devices (e.g., battery life, processing power, storage capacity) can be alleviated by taking advantage of the extensive resources in the Cloud using offloading techniques [Kum10]. The resulting IoT-Cloud paradigm enables the network designers to implement a new breed of services and applications (e.g., health system monitoring, traffic control, energy management, vehicular networking), which are expected to define the essence of next generation smart environments (e.g., smart cities, smart homes, smart grids [Bot14]).

With the increasing number of services and connected devices, IoT traffic is expected to grow dramatically in the coming years [Wen14]. In order to satisfy the QoS requirements of end users and increase the network efficiency, cloud networks are becoming increasingly distributed (i.e., composed of a large number of dispersed cloud nodes). This has motivated the placement of low complexity cloud nodes in close proximity to the device layer, such as cloudlets [Sat09] or micro-clouds [Shi13]. In this context, and with the particularities of the IoT in mind, in [Bon12] the authors propose to expand the Cloud paradigm up to the device layer, creating the so-called Fog, making the analogy with a cloud close to the ground. In fog networks, the end-layer can handle part of the computational tasks [Zhu15]. As a result, the latency and resource utilization of the network resources can be improved compared to traditional cloud computing [Aaz14].

Wireless sensors are expected to assume a crucial role in IoT-Cloud networks thanks to their ubiquity and low cost. Therefore, it is very important to efficiently integrate them into the Fog. This means that they can be orchestrated together with the rest of cloud nodes and IoT devices. Note that although wireless sensors have been conceived to be application-specific, these can become cloud ready infrastructures using software-defined networking (SDN) [Nas14]. The platforms integrating WSNs into the Cloud are referred in the literature to as Sensor-Cloud architectures [Ala13]. These combine the extensive processing resources of cloud servers with the ubiquity of wireless sensors. Moreover, the data gathered by different WSNs, which generally belong to different operators, may be shared on a bigger scale [Xia13], [Mis14b]. In [Luo12], the authors propose Sensor-OpenFlow using the separation of the control and data planes applied in OpenFlow [McK08]. This makes the underlying network (i.e., data plane) programmable by manipulating a user-customizable flow table for each sensor. In [Mit12], the authors present a Sensor-Cloud architecture that makes use of the Contiki operating system [Dun04]. This aims to provide a platform that can obtain any type of data from different heterogeneous sensing infrastructures. In [Asl12], the authors introduce a service oriented architecture, referred to as WSN-SOrA, which is designed to orchestrate the service provisioning in large scale WSNs. In [Mis14a], the importance of the gateway in Sensor-Cloud architectures is highlighted. Then, an optimal gateway selection approach is proposed to minimize the network delay. In [Mis14b], the authors study the performance enhancements that can be obtained using Sensor-Cloud platforms over traditional WSN architectures. The main security issues that may appear in these networks are addressed in [Ahm14].

The previously cited works focus on the architectural and implementation challenges associated to Sensor-Cloud platforms. However, it is still necessary to mathematically formulate the optimization of services in IoT-Cloud networks, taking into account the heterogeneity of sensing, transmission, and computing resources across the physical infrastructure, as well as the tight IoT QoS requirements. Then, in this chapter we formulate the problem of finding the optimal resource allocation in IoT-Cloud networks that meets user requests, satisfies network resource capacities and minimizes overall network consumption. We refer to this problem as the service distribution problem (SDP). In the context of cloud networks, we introduced the cloud service distribution problem (CSDP) in [Bar15]. The CSDP is formulated as a minimum cost mixed-cast flow problem [Ahu93], in which cloud services are represented by a service graph that encodes the relationship between input and output information flows. This solution jointly optimizes the use of compute, storage and transport resources in arbitrary cloud network topologies. Moreover, this is able to capture flexible service chaining, resource consolidation savings, unicast and multicast delivery, and latency constraints. However, the generality of the CSDP, and in particular of its cost function, while valuable in its intention to capture a wide range of business models, leads to a mixed integer linear program of exponential complexity.

In this chapter, we focus on the optimization of IoT services in IoT-Cloud platforms. We adopt a linear pay-as-you-go cost model, in which the cost of a service is proportional to its use of the physical infrastructure. In addition, we include specific models that characterize the particularities of IoT-Cloud networks (wireless access, wireline cloud metro) and end devices (e.g., battery life, processing power, sensing capability, radio technology) of essential importance in IoT environments, and not yet characterized in the existing literature. We provide an efficient linear programming formulation to the IoT-Cloud SDP that can be solved in polynomial time, enabling the dynamic and elastic configuration of IoT-Cloud resources for real-time IoT services based on short-term demand estimations. Then, we solve the IoT-Cloud SDP for an illustrative set of smart city services. In particular, we consider three illustrative services that require the orchestration of the sensing cloud and device layers.

6.2.2. Contributions

The main contributions of this chapter are the following:

- We mathematically formulate the service distribution problem in IoT-Cloud networks as a minimum cost mixed-cast flow problem. This can be efficiently solved via linear programming.
- We solve the IoT-Cloud SDP for an illustrative set of smart city services that allow users the consumption of augmented information resulting from the real-time processing of live smart city streams.

6.2.3. Organization of the Chapter

The remaining of this chapter is organized as follows: Section 6.3, motivates IoT-Cloud platforms in smart cities and its advantages in relation to centralized cloud strategies. Section 6.4, introduces the system model and presents the concept of service graph. Section 6.5, formulates the service distribution problem in IoT-Cloud platforms. Section 6.6, shows the energy consumption of illustrative smart city services using the IoT-Cloud SDP. Finally, Section 6.7 summarizes this chapter and presents the main conclusions.

6.3. IoT-Cloud in Smart Cities

Smart cities are a clear example of the huge potential of the IoT. These connect many wired and wireless devices to provide services that enhance the wellbeing of people (Figure 6.1). Smart cities aim to improve many city services, such as the system health monitoring, traffic mobility, waste management, energy management and healthcare.

The general structure of smart cities includes a device layer, a transmission layer and an application layer (Figure 6.2). The device layer provides raw data from sensing devices

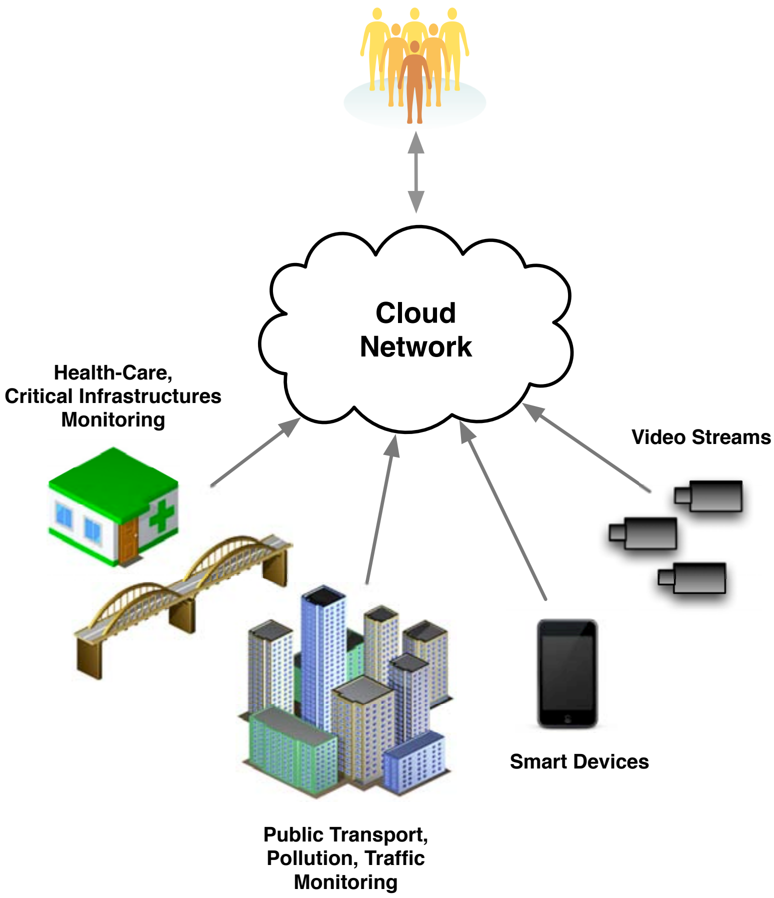


Figure 6.1. Smart city architecture.

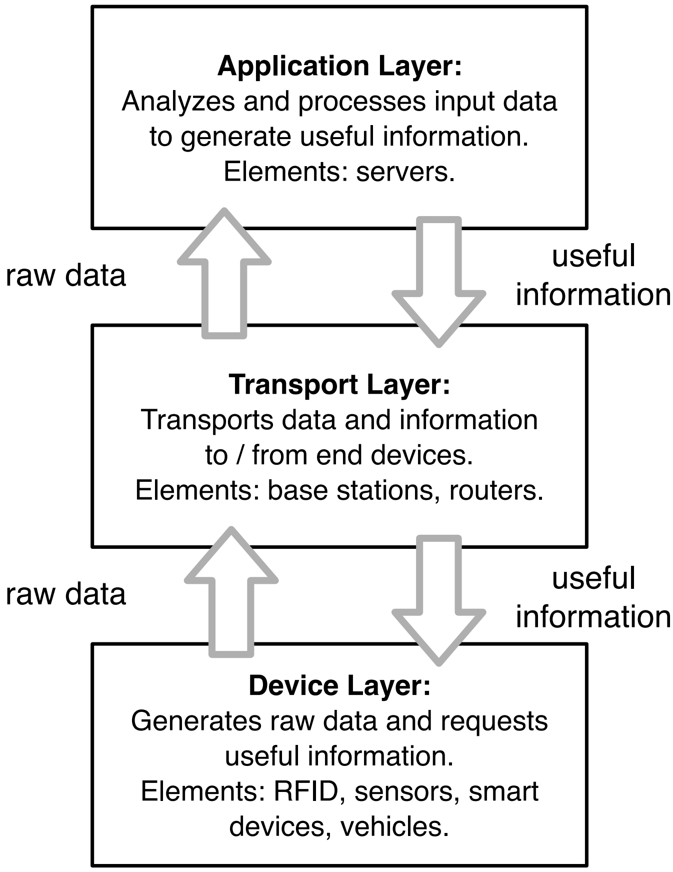


Figure 6.2. Smart cities layers

and also requests processed information. In general, there are nodes that only provide data (e.g., distributed sensors, cameras, RFID, GPS), nodes that only request information (e.g., actuators, public screens), and nodes that may both provide and request information (e.g., smart devices). The network layer transports both the raw data to the cloud servers, and the processed data to the end devices. This layer combines wired and wireless links, with different technologies. Finally, the application layer processes and analyzes raw data to generate the information requested by the device layer.

Traditionally, a centralized cloud scheme in smart cities was considered to be the most efficient solution, since the consolidation of processing resources in a single location reduces the operational cost of the network. However, recent works indicate that the network performance can be improved by doing more processing close to the end users [Sat15], [Ha13]. In this context, [Sat09] and [Shi13] are initiatives to place distributed near-user cloud nodes. Thanks to the recent advances on mobile computing [Lei13] and virtualization [Nas14], the device layer can also be integrated into the cloud network. Then, smart cities can be modeled as converged IoT-Cloud networks in order to efficiently manage their resources, enhance their QoS, and reduce their energy consumption [Bon12].

In this scenario, sensors, smartphones, connected vehicles, and the rest of mobile devices, are not simple endpoints, but storing, sensing and computing resources. With this level of abstraction, they can be orchestrated together, and also interact with the cloud resources. Some of the advantages of the IoT-Cloud paradigm in smart cities are:

- Low latency: the processing can be placed at the edge of the network in order to support latency sensitive applications.
- Geographical distribution: It allows widely distributed cloud systems, since the device layer is an active part of the Cloud.
- Large scale sensor networks: The interoperability of WSNs from different providers creates larger virtualized sensing platforms.
- Location aware and mobility support: The current location of users can be used to provide service mobility.

In the following section, we introduce the system model considered in this chapter, which captures the high heterogeneity of IoT-Cloud platforms in terms of services and devices.

6.4. System Model

In this chapter, we represent an IoT-Cloud network as a graph. Note that we do not model the network as we did in the previous chapters, since we must consider different types of nodes (e.g., wireless sensors, smart devices, cloud servers) and links (e.g., ZigBee, Wifi, 4G, optical). Moreover, we also need to capture the different classes of services in “smart” environments, which may collect and/or distribute data. Then, using graph-based representations the network can be mathematically formulated in a more formal way, and therefore its optimization is simplified.

6.4.1. Network Model

We consider a network modeled as a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with V vertices and E edges representing the set of network nodes and links, respectively. Each node is characterized by its particular energy resources (e.g., power grid, battery), processing resources (e.g., processor, microprocessor) and data acquisition resources (e.g., camera, sensors, I/O interfaces). These are connected by wired and wireless links, which are characterized by their particular transport capacity. In particular, we denote with c_u^{pr} and c_u^{sn} the data processing and acquisition capacities (in bps) at node $u \in \mathcal{V}$. Analogously, we denote with e_u^{pr} and e_u^{sn} the data processing and acquisition efficiencies (in Watts per bit) at node $u \in \mathcal{V}$. Finally, c_u^{tr} and e_{vu}^{tr} denote the capacity (in bps), and the efficiency (in Watts per bit) of the link $(v, u) \in \mathcal{E}$. Besides the particularities of the physical links, these also consider the transmitter and receiver efficiencies and capacities.

6.4.2. Service Model

We denote with $\mathcal{O}$ the set of information objects or flows that can be processed, captured, or transported over the network, and with $\mathcal{P}$ the set of virtual functions that can be implemented. An information object o is, in general, the output of a function $p_o \in \mathcal{P}$ that requires the set of objects $\mathcal{Z}(o) \subset \mathcal{O}$ as input. For example, in a personalized video streaming application, the processed video stream o results from the combination of multiple video streams (i.e., $\mathcal{Z}(o)$) via the video processing function p_o.

We represent a cloud service ϕ by a rooted tree $\mathcal{T}_\phi = (\mathcal{A}_\phi, \mathcal{O}_\phi)$, referred to as the service

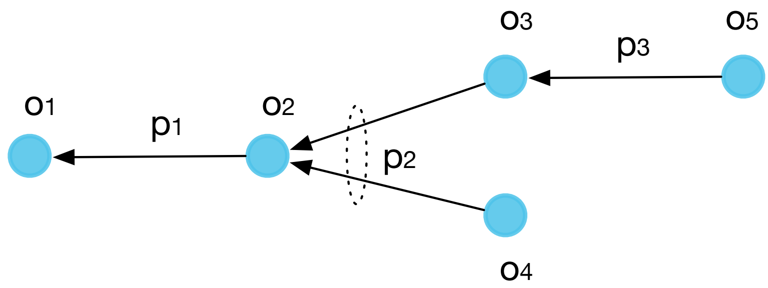


Figure 6.3. An example of a service graph, $\mathcal{T}_\phi = (\mathcal{A}_\phi, \mathcal{O}_\phi)$, with $|\mathcal{O}_\phi| = 5$ objects, $|\mathcal{A}_\phi| = 4$ edges, and $|\mathcal{P}| = 3$ virtual functions.

tree. For any node $o \in \mathcal{O}_\phi$, there is a directed edge $(z, o) \in \mathcal{A}_\phi$ for all $z \in \mathcal{Z}(o)$, as shown in Figure 6.3. Hence, the set of objects $\mathcal{Z}(o) \subset \mathcal{O}$ required to generate object o via function p_o are represented as the children of o in the service tree $\mathcal{T}_\phi$. In particular, the root of the service tree $r_\phi \in \mathcal{O}$ represents the final information object that needs to be delivered to the end user(s). Note that r_ϕ has no outgoing links in $\mathcal{T}_\phi$, as it is not the input to any other object for cloud service ϕ, and the leaves of the rooted tree have no incoming links, as they represent source information that cannot be created by a cloud service function. The service tree hence encodes the relationship between all the objects necessary to deliver the final product to the end user(s) via the required cloud functions. It is important to note that each object o is uniquely characterized by the pair $(p_o, \mathcal{Z}(o))$. This allows different objects to share the same input, but be created via different functions, or to share the same function, but have different inputs.

6.4.3. Service Requirements

When a user requests a cloud service ϕ, the user is, in essence, requesting the final information object or flow represented by the root of the service graph, $r_\phi \in \mathcal{R}$. We denote with B_o the bitrate (in bps) associated with content object $o \in \mathcal{O}$.

In terms of QoS, we denote with h_{vu} and r_{vu} the transport delay (in seconds) and reliability (in terms of packet delivery ratio) associated with the link $(v, u) \in \mathcal{E}$, respectively, and with h_u the processing delay at cloud node $u \in \mathcal{V}$. We use $H_{d,o}$ to denote the maximum delay allowed for the delivery of content object $o \in \mathcal{O}$ at destination $d \in \mathcal{V}$. Moreover, E_u (in Watts) stands for the maximum power consumption of node $u \in \mathcal{V}$, taking into account its expected lifetime. Note that E_u may be ∞ for non battery powered nodes.

Once introduced the system model, in the following section we formulate the IoT-Cloud SDP.

6.5. The IoT-Cloud Service Distribution Problem

The IoT-Cloud SDP is formulated as a minimum cost mixed-cast flow problem. This is characterized by the following flow variables:

- User-object flows: are characterized by a triplet (d, o, z), which indicates that the given flow is carrying information of object $z \in \mathcal{O}$, used to deliver final product $o \in \mathcal{O}$ at destination $d \in \mathcal{V}$. In particular, $f_{vu}^{tr,d,o,z}$, $f_{vu}^{sn,d,o,z}$, $f_u^{pr_i,d,o,z}$ and $f_u^{pr_o,d,o,z}$ are binary variables that indicate whether object z is carried (i.e., tr), captured (i.e., sn) or processed (i.e., pr) by link $(v, u) \in \mathcal{E}$ or node $u \in \mathcal{V}$ for final product $o \in \mathcal{O}$ at destination $d \in \mathcal{V}$. Note that we differentiate between $f_u^{pr_i,d,o,z}$ and $f^{pr_o,d,o,z}$, which denote the input and output flows of the processing unit at node $u \in \mathcal{V}$ associated with triplet (d, o, z). Figure 6.4 illustrates the network flows associated with a given triplet (d, o, z) at node $u \in \mathcal{V}$, where pr represents the processing unit that hosts virtual functions and $q_u^{d,o,z}$ is a binary demand parameter that indicates if node $u \in \mathcal{V}$ requests object $z \in \mathcal{O}$. Note that $q_u^{d,o,z} = 0$ if $u \neq d$ or $z \neq o$, since users only request final information objects for themselves.

- Global flows: f_{vu}^{tr}, f_u^{sn} and f_u^{pr}, determine the total amount of information flow carried, captured or processed at a given physical node/link, respectively.

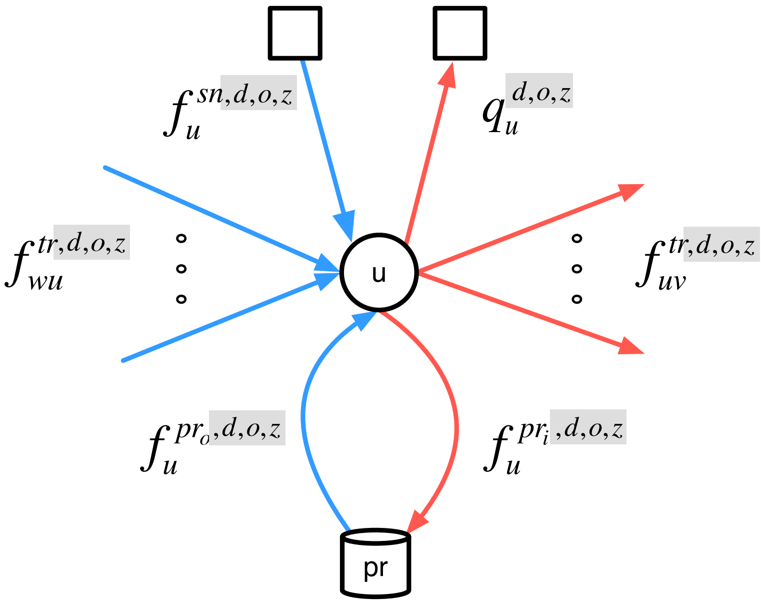


Figure 6.4. Generalized flow conservation at node $u \in \mathcal{V}$, where pr represents the processing unit that hosts virtual functions at node u.

Objective Function

We define a generic cost associated with the capturing (k_u^{sn}), processing (k_u^{pr}), and transport (k_{vu}^{tr}) of traffic over the physical resources of the network. For instance, these may be defined in terms of capital cost (\$/bit), energy consumption (W/bit), environmental impact (CO^2/bit), or a combination of multiple costs. Then, the objective function is characterized by the global flows and determines the total cost of the network:

$$\begin{aligned} \text{minimize} \quad & \sum_{(v,u)\in\mathcal{E}} \left(k_{vu}^{tr} f_{vu}^{tr}\right) + \\ & \sum_{u\in\mathcal{V}} \left(k_u^{sn} f_u^{sn} + k_u^{pr} f_u^{pr}\right). \end{aligned} \tag{6.1}$$

Next, we include the problem constraints, which assure the conservation of the total flow, capture the capabilities and capacities of nodes/links, model the unicast and multicast traffic, and define the QoS requirements.

Generalized Flow Conservation Constraints

User-object flows must satisfy demand and flow conservation. By modeling the demand $q_u^{d,o,z}$ as part of the outgoing flow of node u, we can use the following generalized flow conservation constraints:

$$q_u^{d,o,z} + \sum_{w \in \mathcal{N}^+(u)} f_{uw}^{tr,d,o,z} + f_u^{pr_i,d,o,z} = \sum_{v \in \mathcal{N}^-(u)} f_{vu}^{tr,d,o,z} + f_u^{sn,d,o,z} + f_u^{pr_o,d,o,z} \quad \forall u, d, o, z. \tag{6.2}$$

Flow conservation constraints state that the outgoing flow associated with a given triplet (d, o, z) must be equal to the incoming flow for that same triplet, for any node u. As illustrated in Figure 6.4, the outgoing flow is composed of the outgoing transport flows, the processing flow leaving node u towards the processing unit, and the demand flow; while the incoming flow is composed of the incoming transport flows, the capturing flow, and the processing flow going out of the processing unit.

In addition, each processing unit must satisfy the following flow conservation constraint:

$$f_u^{pr_o,d,o,z} \leq f_u^{pr_i,d,o,y} \quad \forall u, d, o, z, y \in \mathcal{Z}(z). \tag{6.3}$$

This constraint makes sure that in order to have a processed flow z for demand (d, o), a flow associated with each of the input objects required to generate z, $y \in \mathcal{Z}(z)$, for demand (d, o), must be present at the input of the processing unit.

Function Availability Constraints

These constraints allow restricting the set of virtual functions that can be implemented at a given node:

$$f_u^{pr_o,d,o,z} = 0 \quad \forall u, d, o, z, p_z \notin \mathcal{P}_u, \tag{6.4}$$

where $P_u \subset P$ is the set of virtual functions available at node u.

Source Constraints

We denote the set of objects that are available in the network as source information and hence input for cloud services as $\mathcal{S} \subset \mathcal{O}$. We define the set $O_u \subset \mathcal{S}$ as the objects that are captured by node u:

$$f_u^{sn,d,o,z} = 0 \quad \forall u, d, o, z \notin O_u. \tag{6.5}$$

$$\tag{6.6}$$

In addition, we need to make sure that source objects $\mathcal{S} \subset \mathcal{O}$ are not created in the network:

$$f_u^{pr_o,d,o,z} = 0 \quad \forall u, d, o, z \in \mathcal{S}. \tag{6.7}$$

Mixed-cast Constraints

Mixed-cast constraints allow modeling the unicast or multicast nature of flows via the corresponding relationship between user-object and global flows. Since a single captured object can be used to satisfy multiple demands, capturing flows are said to be multicast. Hence, user-object capturing flows for the same object, but for different destinations, are allowed to overlap:

$$f_u^{sn,d,o,z} \le f_u^{sn,z} \quad \forall u, d, o, z, \tag{6.8}$$

$$\sum_{z \in \mathcal{O}} f_u^{sn,z} B_z = f_u^{sn} \quad \forall u. \tag{6.9}$$

where the user-object flows are weighted by the size of the object B_z.

On the other hand, the transport and processing flows may be unicast or multicast. If these are multicast:

$$f_{vu}^{tr,d,o,z} \le f_{vu}^{tr,z} \quad \forall u, d, o, z, \tag{6.10}$$

$$\sum_{z \in \mathcal{O}} f_{vu}^{tr,z} B_z = f_{vu}^{tr'} \quad \forall u, \tag{6.11}$$

$$f_u^{pr_o,d,o,z} \le f_u^{pr,z} \quad \forall u, d, o, z, \tag{6.12}$$

$$\sum_{z \in \mathcal{O}} f_u^{pr,z} B_z \gamma_z = f_u^{pr} \quad \forall u. \tag{6.13}$$

where output processing flows are scaled by a factor γ_z that captures the changes (in bps) associated with the generation of object z from its input $\mathcal{Z}(z)$ via function p_z.

If transport and processing are unicast, user-object flows cannot overlap and must be added across both objects and destinations:

$$\sum_{d \in \mathcal{V}} \sum_{o \in \mathcal{O}} \sum_{z \in \mathcal{O}} f_{vu}^{tr,d,o,z} B_z = f_{vu}^{tr'} \quad \forall (v, u), \tag{6.14}$$

$$\sum_{d \in \mathcal{V}} \sum_{o \in \mathcal{O}} \sum_{z \in \mathcal{O}} f_u^{pr_o,d,o,z} B_z \gamma_z = f_u^{pr} \quad \forall u. \tag{6.15}$$

Note that we use $f_{vu}^{tr'}$ instead of f_{vu}^{tr} to denote that the global transport flow in (6.11) and (6.14) does not include the possible packet retransmissions. These are considered in (6.20).

QoS Constraints

In this section, we formulate the constraints that define the QoS requirements in terms of latency, reliability and energy consumption.

Latency

We use $\delta_{d,o,z}$ to denote the (local) delay associated with a particular user-object flow (d, o, z). This is computed as the weighted sum of the delay associated with the transport and processing of (d, o, z) flows:

$$\delta_{d,o,z} = \sum_{(v,u)\in\mathcal{E}} f_{vu}^{tr,d,o,z} h_{vu} + \sum_{u\in\mathcal{V}} f^{pr_o,d,o,z} h_u \quad \forall d, o, z. \tag{6.16}$$

In (6.16), without loss of generality, we assume that input processing flows have zero delay and capture all processing delay with the output processing flows. We then use $\delta_{d,o,z}^{ag}$ to denote the aggregate delay associated with user-object flow (d, o, z), computed as the sum of the local delay, $\delta_{d,o,z}$, plus the maximum across the aggregate delays of all of its input flows $\delta_{d,o,y}^{ag}, \forall y \in \mathcal{Z}(z)$, as:

$$\delta_{d,o,z}^{ag} = \delta_{d,o,z} \quad \forall d, o, z \in \mathcal{S}, \tag{6.17}$$

$$\delta_{d,o,z} + \delta_{d,o,y}^{ag} \leq \delta_{d,o,z}^{ag} \quad \forall d, o, z, y \in \mathcal{Z}(z). \tag{6.18}$$

Finally, the aggregate delay associated with the delivery of final object o at destination d is constrained according to the service requirements:

$$\delta_{d,o,o}^{ag} \leq H_{d,o} \quad \forall d, o. \tag{6.19}$$

Reliability

The reliability of links has a relevant impact on the traffic flow due to the retransmissions caused by packet losses, particularly in low power wireless links. The average number of retransmissions required in link $(v, u) \in \mathcal{E}$ is modeled as the reciprocal of its packet delivery ratio:

$$f_{vu}^{tr} = f_{vu}^{tr'}/r_{vu} \quad \forall (v, u). \tag{6.20}$$

The value of r_{vu} is between zero (i.e., disconnected nodes) and 1 (i.e., ideal link), and therefore it increases the actual transport flow.

Energy Consumption

The total energy consumption due to transport, processing and acquisition at node u must be lower than its maximum energy consumption:

$$\begin{aligned} &\sum_{(u,w)\in\mathcal{E}} f_{uw}^{tr} e_u^{tx} + \sum_{(v,u)\in\mathcal{E}} f_{vu}^{tr} e_u^{rx} + \\ &f_u^{sn} e_u^{sn} + f_u^{pr} e_u^{pr} \leq E_u \quad \forall u. \end{aligned} \tag{6.21}$$

Note that we separate the impact of the transport flow f_{vu}^{tr} with e_v^{tx} and e_u^{rx}, which denote the transmission and reception efficiency (in Watts per bit) of nodes u and v, respectively.

Capacity Constraints

Global flows must satisfy capacity constraints, as:

$$f_{vu}^{tr} \le c_{vu}^{tr} \quad \forall (v,u), \tag{6.22}$$

$$f_{u}^{sn} \le c_{u}^{sn} \quad \forall u, \tag{6.23}$$

$$f_{u}^{pr} \le c_{u}^{pr} \quad \forall u. \tag{6.24}$$

Integer/Fractional Flow Constraints

User-object flows can either be binary or fractional. Accordingly, these can be defined as:

$$f_{vu}^{tr,d,o,z}, f_{u}^{pr_i,d,o,z}, f_{u}^{pr_o,d,o,z}, f_{u}^{sn,d,o,z} \in \{0,1\} \quad \forall (v,u), u, d, o, z, \tag{6.25}$$

$$\tag{6.26}$$

or

$$f_{vu}^{tr,d,o,z}, f_{u}^{pr_i,d,o,z}, f_{u}^{pr_o,d,o,z}, f_{u}^{sn,d,o,z} \in [0,1] \quad \forall (v,u), u, d, o, z. \tag{6.27}$$

As shown in [Llo13], binary flow variables in mixed-cast flow problems can be relaxed if: a) all flows are unicast, b) the network topology is a tree, or c) the network can perform intra-session network coding (e.g., random linear coding). Then, the polynomial complexity of the IoT-Cloud SDP may be reduced in these practical cases.

6.6. Simulation Results

In this section, we solve the IoT-Cloud SDP to minimize the overall network consumption of three illustrative smart city services via the linear programming solver Xpress-MP. In

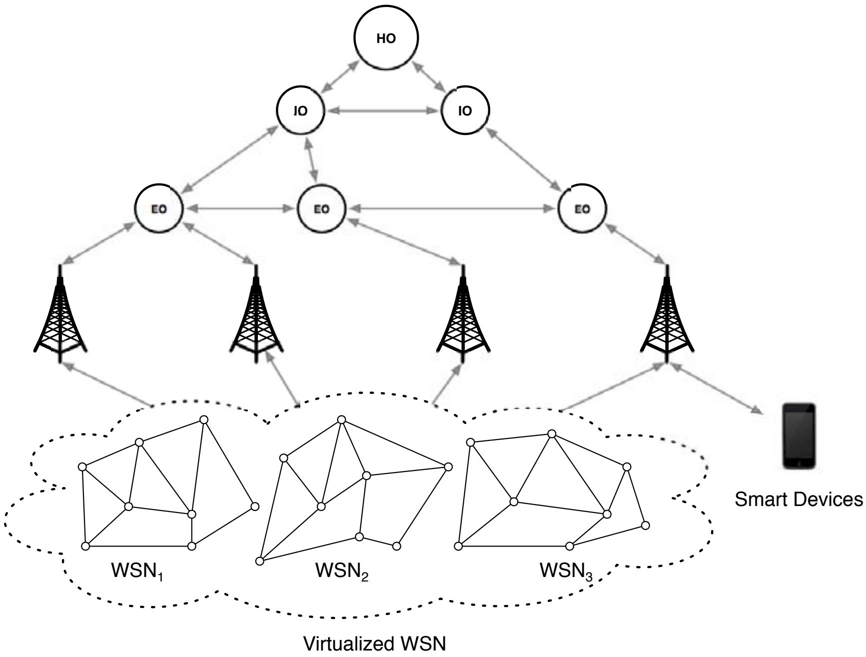


Figure 6.5. Structure of the smart city.

particular, we analyze: i) the virtual fog service network solution (vFSN), which allows tasks to be processed anywhere in the network, ii) a distributed cloud approach, which places tasks in any cloud node, and iii) a fully distributed approach, which allocates all the processing at the devices level.

6.6.1. Smart City Scenario

We consider the smart city network in Figure 6.5 that includes wireless sensors, smart devices (e.g., smartphones, tablets, smart glasses), base stations and cloud nodes. We assume a hierarchical cloud infrastructure that is composed of end offices (EOs), intermediate offices (IOs) and a Head Office (HO) [Bar15]. Base stations are equipped with a network gateway and also with a low complexity cloud instance, such as a cloudlet [Sat09]. Three different WSNs collect environmental information. Each WSN is composed of 25 sensors, uniformly distributed over a sensing area of 250x250 m^2.

Precisely modeling the processing capacity (in MIPS) and efficiency (in MIPS/W) of the smart city nodes is difficult, due to the limited information disclosed by network operators. Since this is out of the scope of this chapter, we compute approximate values using information extracted from [Vis15] (i.e., cloud equipment), [Alt15] (i.e., smart devices), and [Xia10] (i.e., wireless sensors), which are presented in Tables 6.1 and 6.2. Note that we assume that the efficiency of cloud nodes increases at higher layers, thanks to the consolidation of processing tasks. Since the capacity and efficiency of smart devices vary considerably depending on the kind of device, we consider different realistic values in the simulations.

6.6.2. Simulation Details

The processing complexity of tasks (in instructions per bit), depends on the particular application. In this section, we assume that simple tasks, such as data aggregation, have a complexity of 100 instructions per bit. On the other hand, the tasks that require a more complex data analysis require a higher number of instructions per bit (i.e., 500-5000 in our simulations).

We assume that the statistics of the wireless channel are known. This is time-varying and follows the two-slope log-distance path loss model described in [Kyö07] for urban micro-cell scenarios. Then, the path losses (PL) are modeled as follows:

$$\begin{aligned} PL(dB) = PL_0(dB) + 20 - 12.5\alpha + \\ 10\alpha \log(d) + 3\log(f_c/5) + \gamma, \end{aligned} \tag{6.28}$$

where PL_0 is the path loss at the reference distance d_0 (1 m), α is the path loss exponent, d is the communication distance, f_c is the system frequency (in GHz) and $\gamma \sim \mathcal{N}\left(0, \sigma_\gamma^2\right)$ is zero mean Gaussian noise that models the shadowing effects. This model assumes α=2.8 and σ=4 dB. Moreover, a collision-free MAC layer is assumed to obtain results independent from the specific MAC mechanism. Please note that in this chapter we have considered a slightly different path loss model than in previous chapters to better model smart city scenarios.

Taking into account the general characteristics of ZigBee transceivers, we assume f_c=2.4 GHz, $PL_0(dB)$=35 dB [MS05], a transmit power of 3 dBm, a sensitivity value of -91 dBm [Atm09] and omnidirectional antennas. We also consider that the packets generated at wireless sensors have a length of 127 bytes (i.e., standard packet size in IEEE 802.15.4).

The maximum energy consumption of wireless sensors is limited by their expected battery duration. Note that the replacement of their batteries is a costly operation and therefore, nodes should not deplete their batteries before the scheduled maintenance. With this in mind, in the simulations we assume a minimum lifetime of each sensor of 1 year. This is introduced in (6.21) of the IoT-Cloud SDP formulation, taking into account that they are equipped with a battery that provides 27 kJ [Siv10].

The maximum latency may be as low as 10 ms in latency sensitive applications, such as cognitive assistant services [Sat15]. On one hand, strict delay requirements pushes the processing closer to the end user. On the other hand, using the most powerful nodes and the fastest links the computation and transmission times can be reduced, and these are generally closer to the network core. In these simulations, we narrow the set of feasible solutions of the IoT-Cloud SDP using (6.19) taking into account a maximum end-to-end delay of 10 ms.

The range of applications that smart cities can provide is very wide. In this section, we focus on the following examples: augmented reality, autonomous sensing and actuation, and city monitoring.

6.6.3. Augmented Reality Service

The real world and a computer generated view are combined in augmented reality services to create the perception of a new reality. Recent products providing augmented reality are Google Glass, Sony SmartEyeglass and Microsoft HoloLens. In the con-

Table 6.1. Capacities and efficiencies of fog nodes

	Capacity	**Efficiency**
Cloud Node (HO)	53.5 Million MIPS	500 MIPS/W
Cloud Node (IO)	26 Million MIPS	200 MIPS/W
Cloud Node (EO)	13 Million MIPS	133 MIPS/W
Cloud Node (BS)	6.5 Million MIPS	100 MIPS/W
Wireless Sensor	1 MIPS	4167 MIPS/W

Table 6.2. Capacities and efficiencies of links

	Capacity	**Efficiency**
Optical	4480 Gbps	12.6 nJ/bit
4G (Down/Up)	72/12 Mbps	76.2/19 μJ/bit
WiFi	150 Mbps	300 nJ/bit
ZigBee	250 kbps	100 nJ/bit

text of smart cities, the real-time video stream captured by a smart device may be combined with an informative view of the city. In general, smartphones contain the necessary elements to implement augmented reality services (i.e., internet connexion, sensors, camera). However, the high computational requirements of these applications and the limited battery resources of smartphones motivates the network operators to consider offloading approaches [Alt15]. In this context, fog computing can extend the capabilities of end devices, without introducing the high latency of centralized cloud approaches.

In this section, we consider a scenario in which end users request augmented video streams using their smart devices. This is the combination of a real-time video stream, captured with the camera of the smart device, personalized information, downloaded from the Internet, and data from the city gathered by the wireless sensors. We assume that 2000 instructions per bit are required to generate the augmented video. Half of users access the network using 4G and the other half use a WiFi connexion. Note that the video captured and the augmented video must be send in unicast, but the data from wireless sensors can be sent in multicast to avoid duplicate packets.

In Figure 6.6, we compare the average power consumption for different processing effi-

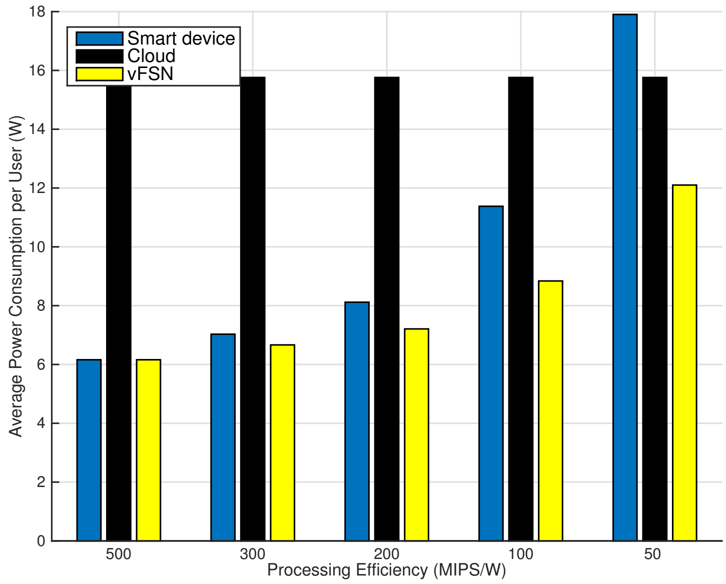


Figure 6.6. Average power consumption (W) of an augmented reality application for different smart device efficiencies (MIPS/W).

ciencies. These are considered to capture the multiple kinds of end devices sharing the smart city infrastructure. On average, end users request an augmented reality video that results from the combination of 200 kbps of video, 50 kbps of personalized information and 1 packet/s from each wireless sensor. The simulation results show that vFSN adjusts the offloading decisions according to the processing efficiency of end devices. The processing tends to be allocated at the devices layer if their efficiency is high enough, while low efficient devices offload their processing to the cloud servers. In general, users accessing the network using WiFi, tend to use the cloud resources frequently. However, the access cost of 4G networks, push the processing to the end devices, as observed also in [Vis15].

In Figure 6.7, the overall power consumption of the network for multiple augmented video bitrates is shown. These represent different video qualities defined according to the users' preferences. We assume that the end devices are smartphones with a processing efficiency of 1000 MIPS/W and 4.5 Wh. We constrained the solutions of the IoT-SDP

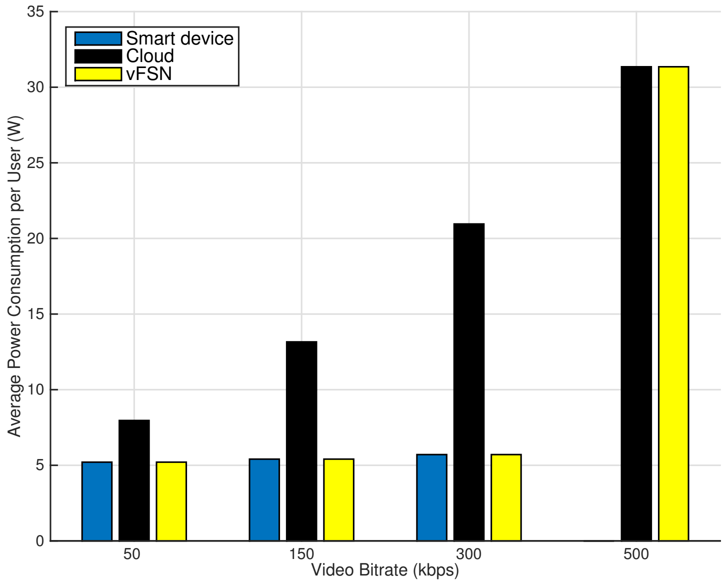


Figure 6.7. Average power consumption (W) of an augmented reality application for different video qualities (kbps). Smartphones must provide at least 5 hours of service.

to support at least 5 hours of augmented video service before depleting the smartphone batteries. In this figure, we observe that the video is processed by the smartphones, even with low video bitrates. This is due to the higher energy efficiency of smartphones, compared to the efficiency of cloud servers. The difference in terms of consumption between local processing and cloud processing increases with the bitrate. Besides the additional processing cost, note that to process the video at the Cloud, this must be transported from the smartphone to a cloud node and then transported back to the smartphone, once the additional information has been included. However, due to the limited energy resources of smartphones, users demanding an augmented video with high quality (e.g., 500 kbps) must offload their processing to the Cloud. Otherwise, these cannot provide the minimum hours of service.

6.6.4. Autonomous Sensing and Actuation Service

Wireless sensors and actuators have been traditionally application specific. However, considering them as fog nodes, their tasks in the network can be adapted to the service requirements, increasing or reducing their activity dynamically.

In this section, we consider a WSN that monitors a particular phenomenon in the city and activates the actuators accordingly. There are 5 of these sensors that are also actuators. Nearby wireless sensors may combine their measurements using data aggregation techniques. This reduces traffic by a compression rate that depends on the spatial and temporal correlation of data. A realistic compression rate of 50% [Cap12] is assumed in this simulations. Based on these measurements, a simple decision of turning on/off each actuator is taken.

In Figure 6.8, we compare the power consumption for different packet generation rates at wireless sensors. Note that a high rate increases the accuracy of the service, but also increases the energy consumption of wireless sensors. Then, in a real application this should be defined according to the service requirements. As we can observe, the sensing platform can manage the actuators without the assistance of the cloud infrastructure as long as the packet rate is low (i.e., lower than 2 packets per second). However, at higher rates this processing has to be offloaded to the Cloud in order to keep the nodes alive for at least 1 year without the replacement of their batteries.

In Figure 6.9, we consider different processing complexities, representing different kinds of data aggregation. Similarly to what has been observed in Figure 6.8, for low aggregation complexities the wireless sensors process everything themselves to avoid using the cloud resources. However, this is not possible if the processing cost is higher than a certain value (i.e., 70 instructions per bit). Then, this has to be placed at the cloud layer.

In both cases, the vFSN solution places the processing at the sensing layer as long as it is possible to take advantage of its higher efficiency. In case that the packet rate or the

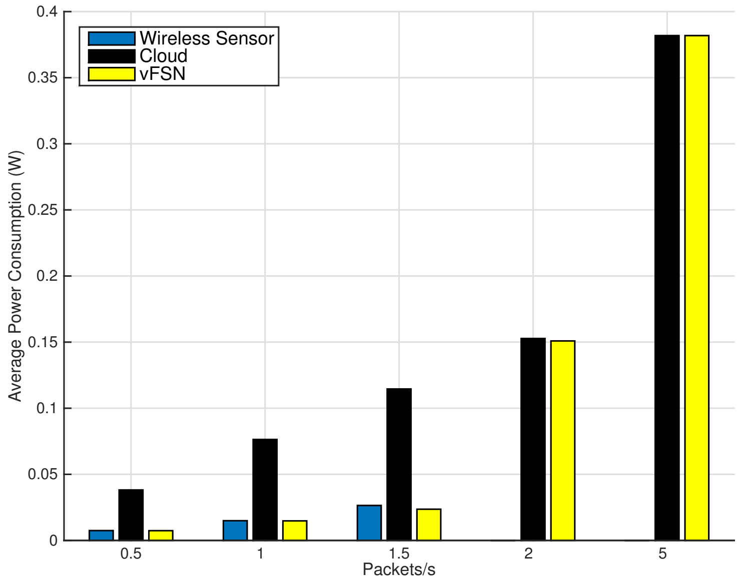


Figure 6.8. Average power consumption (W) of an autonomous sensing actuation platform for different packet generation rates (packets/s).

processing complexity increase above a certain threshold, this is automatically placed at the cloud layer.

6.6.5. City Monitoring Service

Thanks to the deployment of large scale wireless sensor networks, it is possible to provide real-time information about the city. In this example, users request personalized information that is generated from the analysis of the data gathered by three different WSNs. For instance, in a mobility assistant service, these WSNs could provide the speed of vehicles, pollution levels and congestion at the city access, respectively. Then, different mobility options could be obtained from this data (e.g., the fastest route, the most eco-friendly route or the best public transport alternative). Note that the measurements from wireless sensors can be sent in multicast, since they can be used to generate multiple information objects. However, the personalized information can only be sent in multicast for these users requesting the same information at the same time. Otherwise,

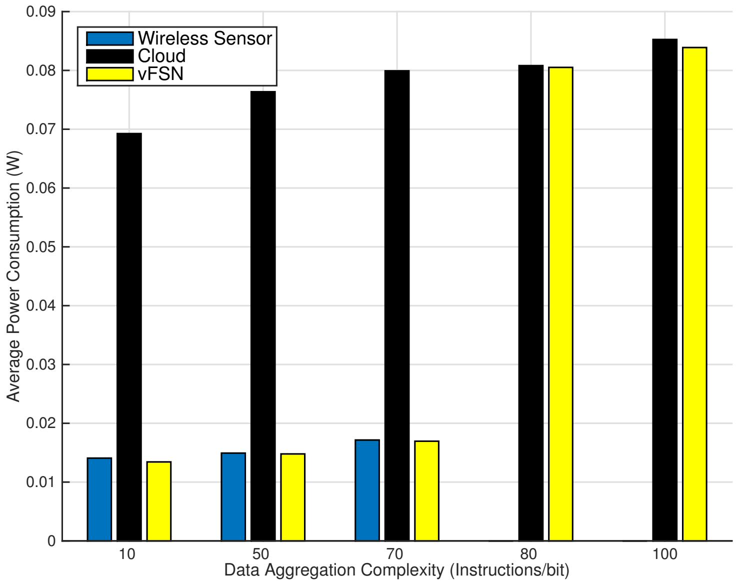


Figure 6.9. Average power consumption (W) of an autonomous sensing actuation platform for different data aggregation complexities (instructions/bit).

this must be sent in unicast.

We assume smart devices with a processing efficiency of 1000 MIPS/J. The processing complexity of analyzing the data from each WSN is 500 instructions per bit, while the complexity of generating the personalized information is 5000 instructions per bit. An average rate of 1 packet per second is generated by each wireless sensor.

In Figure 6.10 and Figure 6.11, we compare the power consumption per user for different average number of users requesting simultaneously the same information using WiFi and 4G, respectively. The bitrate of this information is assumed to be 200 kbps. Note that if this is processed and transported in multicast through the cloud layer, the processing of this information can be consolidated in a specific cloud location. Nevertheless, we assume that users receive the information in unicast from the base stations, since 4G multicast is not available commercially yet. In Figure 6.10, we observe that if there are more than 2 simultaneous WiFi users, it is more efficient to consolidate the processing at a cloud location than processing it locally, even if the transport cost of the requested information

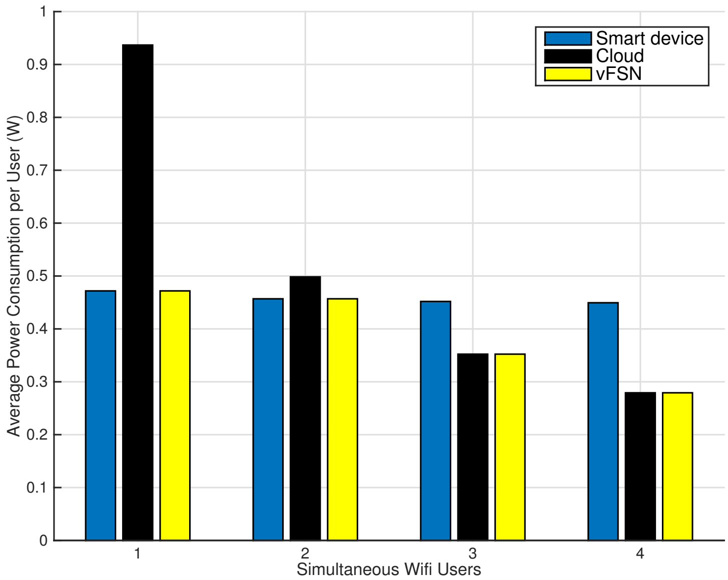


Figure 6.10. Average power consumption (W) of a city monitoring application for different WiFi users requesting the same information at the same time.

is higher. However, for users accessing the network using 4G (Figure 6.11), the additional transport cost, which is much higher in 4G networks, cannot be compensated with a relatively low number of users. Note that this cost could be reduced using energy efficient 4G multicast streaming schemes [Yoo14].

In Figure 6.12, we compare the average power consumption for different bitrates of personalized information. A high bitrate represents applications providing detailed information, while low bitrates indicate that the application is only providing the most relevant information. We consider the same number of WiFi and 4G users. The same information is simultaneously requested by 4 users on average. As we can observe, if users request a low bitrate, the processing tends to be placed at the cloud layer to reduce the processing cost by consolidating it in a single location. However, the cloud approach becomes less efficient at high bitrates due to the transport cost from the Cloud to the smart devices, particularly to 4G devices. Then, vFSN moves the processing closer to the users in order to reduce this cost, at the expenses of reducing the processing

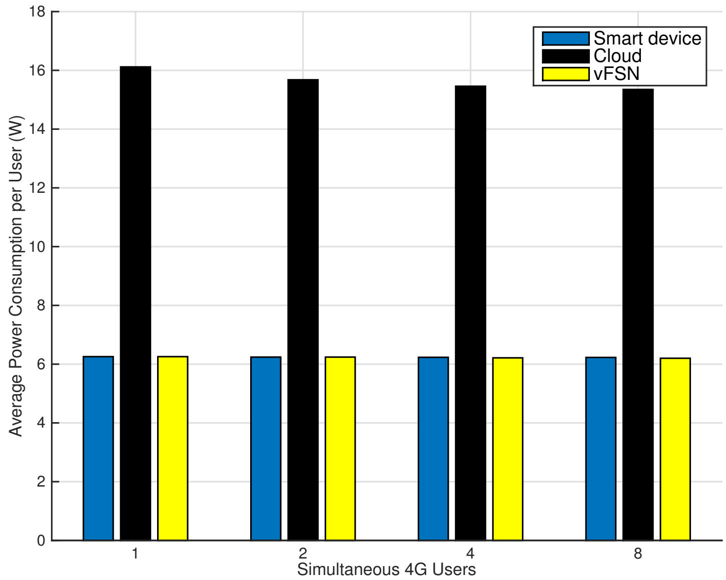


Figure 6.11. Average power consumption (W) of a city monitoring application for different 4G users requesting the same information at the same time.

consolidation.

6.7. Summary and Conclusions

The future internet will combine the Internet of Things (IoT) and the Cloud. In this new paradigm, wireless sensors become cloud ready infrastructures and therefore, they can be orchestrated together with the rest of the IoT.

In this chapter, we formulate the service distribution problem (SDP) in IoT-Cloud networks, referred here to as the IoT-Cloud SDP. This finds the optimal placement of virtualized functions over the network, taking into account the heterogeneous capacities and limitations of end devices. We have implemented the IoT-Cloud SDP in three illustrative smart city services: augmented reality, autonomous sensing and actuation, and city monitoring. In the first example, the simulation results show that vFSN adjusts the offloading decisions according to the processing efficiencies of end devices. In

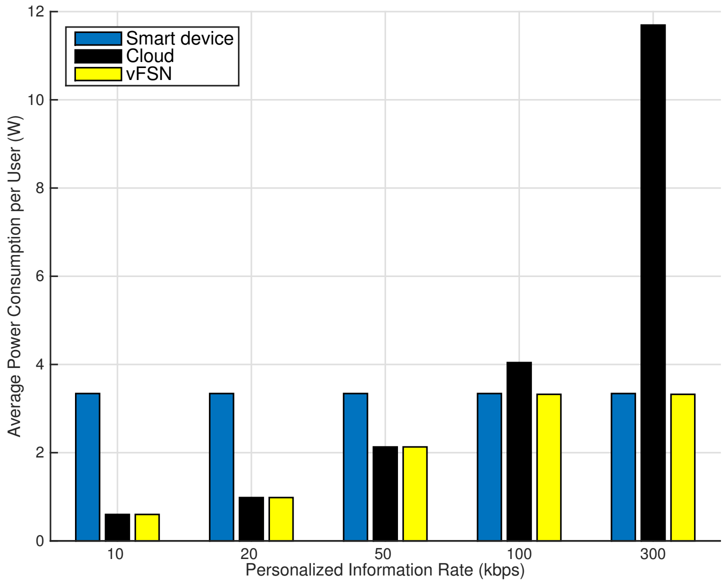


Figure 6.12. Average power consumption (W) of a city monitoring application for different personalized information rates (kbps).

particular, the augmented reality video of highly efficient devices is processed locally, to reduce latency and traffic going through the network, while the processing of low efficient devices tend to be offloaded to the Cloud. In the second example, we have observed that vFSN moves the processing of the information from wireless sensors with a high data generation rate to the Cloud in order to extend their lifetime. On the other hand, the processing is placed at the sensing platform when this rate and the processing complexity are low. In the third example, we have shown that vFSN takes the offloading decisions of a city monitoring application according to the personalized information bitrate and the number of simultaneous users requesting the same information. While a high bitrate moves the processing closer to the end user in order to reduce its transport cost, a high popularity pushes the processing back to the Cloud in order to consolidate the processing tasks.

In general, we have shown that the IoT-Cloud SDP captures the tradeoffs that appear in IoT-Cloud platforms due to the heterogeneity of services, network technologies and devices. Then, using a fully virtualized approach, individual placement decisions can be taken in order to minimize the energy consumption of these networks, and hence reduce their cost and environmental impact.

Chapter 7

Conclusions and Future Work

In this thesis, we have addressed several challenges that are expected to be critical in the integration of WSNs into the future Internet of Things. In *Chapter 3*, we have enhanced RPL with a pragmatic channel allocation and transmission power control. Both techniques are expected to be crucial in future WSNs, due to the increasing density of wireless sensors in many scenarios, such as smart cities. In *Chapter 4*, we have designed a robust routing approach that deals with the challenges originated by the mobility of nodes. Note that with the reduction of the size of wireless sensors, they are going to be easily attached to mobile entities for monitoring, localization or tracking applications. In *Chapter 5*, we have addressed the traffic heterogeneity problem with a cooperative multi-tree routing strategy. The increasing complexity of applications and the collaboration among different WSNs originates independent traffic flows with individual requirements. Finally, in *Chapter 6* we have formulated the integration of wireless sensors into Cloud-based IoT platforms. As a result, wireless sensors can extend their hardware limitations and share their measurements in a bigger scale.

7.1. Conclusions

First of all, in *Chapter 1* we have discussed the main motivations and presented the outline of this thesis. We have also introduced the main research contributions, which are contained in Chapters 3-6.

In *Chapter 2*, we have presented the background for the following chapters. We have first introduced gradient-based routing, with emphasis on the standardized RPL protocol, since this is the core of the proposed solutions in *Chapters 3-5*. Second, we have introduced the multi-channel and multi-power mechanisms for WSNs from a pragmatic perspective. Third, we have explained position-based routing and the impact of realistic hardware and channel conditions on its reliability. Fourth, we have presented the problem of managing heterogeneous traffic in a single WSN, and a general classification of the most important traffics in WSNs. Finally, we have motivated the transition from cloud computing to fog computing, its most relevant advantages, and also some challenges that need to be addressed to enable this paradigm shift.

In *Chapter 3*, we have proposed a pragmatic joint routing, channel allocation and transmission power control solution to enable high density wireless sensor deployments. This provides multi-channel and multi-power capabilities to RPL that can be adjusted according to the channel conditions and the QoS requirements. We have shown that it efficiently reduces the energy consumption and the packet collisions of the network, without adding significant complexity to RPL. Two different routing metrics are proposed, referred to as MinAP and MaxPDR, which combine reliability and energy efficiency criteria. MinAP is more suitable for applications with tight energy consumption constraints, and MaxPDR is recommended for applications that demand a high network reliability. These have been compared through simulations with RPL and also with LEACH, GBR and the gradient-based routing protocol adopted in ZigBee. These simulations have shown that the proposed extension increases the reliability, and reduces the aggregated power and the collision probability of RPL. It is also worth noting that MaxPDR can be adapted using a configurable parameter β to deal with the reliability vs. energy consumption tradeoff. Moreover, the proposed strategies have also been validated in a WSN testbed with commercial motes. The experimental results have confirmed their implementability and also the promising results obtained through simulations.

In *Chapter 4*, we have addressed node mobility in WSNs with KP-RPL, which further extends RPL enabling it to cope with mobile nodes. This combines RPL with a robust, yet simple positioning algorithm, based on the construction of confidence regions and Kalman filtering. Moreover, blacklisting is applied to avoid disconnections due to positioning errors. Thanks to these additional algorithms, KP-RPL reduces the impact of the positioning inaccuracies that are introduced by the RSSI measurements and velocity estimates. Thus, mobile nodes can make better routing decisions. We have evaluated KP-RPL in different network conditions and trajectories of the mobile nodes, observing that it increases the reliability of the network, compared to geographical routing. Moreover, the impact of bad channel conditions in KP-RPL is significantly lower. We have observed that blacklisting increases the network reliability, but it also increases the average ETX of the network, since this is a conservative strategy. This generates a tradeoff that can be managed with the size of the confidence regions, where the ETX is improved using smaller areas, and larger areas increase the PDR. We have also evaluated KP-RPL in scenarios where anchor nodes are not continuously broadcasting positioning packets. In this situation, KP-RPL is particularly efficient because it needs a lower number of positioning nodes than geographical routing thanks to the Kalman filer. Then, the activity of anchor nodes can be reduced in order to increase their lifetime. Moreover, the lower ETX achieved by KP-RPL reduces the energy consumption of mobile nodes for transmitting data packets. Therefore, KP-RPL extends the overall lifetime of the network.

In *Chapter 5*, we have extended RPL with C-RPL to deal with the QoS requirements of multiple traffic flows. This creates energy efficient instances, referred to as C-RPL Instances, following a cooperative strategy among nodes with different sensing tasks. Since nodes would never agree to cooperate, C-RPL coordinates them to create coalitions that improve the tradeoff between their own performance metric and energy consumption, using a cooperation parameter ρ. The simulation results show that C-RPL efficiently creates C-RPL Instances according to their individual objective functions and the particular network conditions. In particular, we have observed that C-RPL tends to create large instances when the node density is low. Otherwise, it constructs smaller instances to reduce the energy consumption of the network and also to avoid congestion problems. Besides, we have proposed a fairness analysis for networks with multiple instances that measures the distribution of the network resources when multiple objective functions coexist in the network. The results show that C-RPL obtains a higher fairness degree than RPL approaches with non-cooperative instances.

Finally, in Chapter 6 we have formulated the IoT-Cloud SDP to efficiently solve the service distribution problem in Cloud-based IoT platforms. This considers the novel paradigm of fog computing, which extends the Cloud up to the device layer. The IoT-Cloud SDP orchestrates wireless sensors together with the rest of devices in the IoT to find the placement of virtual functions across the network that minimizes the overall network cost, meeting users' requests and satisfying the network resource capabilities. We have solved the IoT-Cloud SDP for three illustrative smart city services: augmented reality, autonomous sensing and actuation, and city monitoring. The results show that the task offloading is adjusted according to the energy efficiency of end devices. Highly efficient devices can reduce the communication costs processing tasks locally, while low efficient devices tend to offload their tasks to not waste battery resources. The generation rate of wireless sensors also has an impact on the solution, since it introduces traffic in the network that needs to be processed somewhere. This is processed by the wireless sensors as long as their lifetime constraints are satisfied, but it has to be processed at the Cloud otherwise. On the other hand, a high bitrate of personalized information moves the processing closer to the end user in order to reduce its transport cost. However, highly popular information tends to be processed at the Cloud in order to consolidate the processing tasks at a single location.

7.2. Future Work

In general, the contributions presented in this thesis can be further developed as follows:

- Considering state-of-the-art low-power MAC protocols in our simulations.
- Validating the simulation results obtained using Matlab with network simulators, such as ns-3.
- Extending the experimental results with additional scenarios and larger deployments.
- Combining the RPL-based techniques proposed in Chapters 3 to 5.

Moreover, there are some topics derived from the research presented in each chapter that remain as future work.

In Chapter 3, it would be interesting to explore the combination of other multi-channel and multi-power mechanisms with a low signalling cost and RPL. Moreover, the collision problem among sensors could be experimentally studied using a larger WSN testbed.

In Chapter 4, KP-RPL has been studied for mobile nodes following a predefined trajectory, and it could be interesting to observe its performance in scenarios with unpredictable moving patterns. In addition, an experimental evaluation of KP-RPL in a commercial WSN testbed has not been provided yet.

In Chapter 5, it would be interesting to study a more distributed C-RPL that applies the cooperation game among nodes instead of groups of nodes. This would reduce its signalling cost and also increase the granularity of its solutions. Moreover, an experimental comparison among RPL and C-RPL would also be very significant.

In Chapter 6, a more precise energy efficiency model of cloud servers, smart devices and wireless sensors would improve the accuracy of the simulation results. In addition, the

IoT-Cloud SDP could also be applied to other smart environments, such as smart homes, smart grids and smart transportation systems.

Bibliography

[Aaz14] Aazam, M. and Huh, E.N. "Fog computing and smart gateway based communication for cloud of things". In "Future Internet of Things and Cloud (FiCloud), 2014 International Conference on", pages 464–470. Aug 2014. doi: 10.1109/FiCloud.2014.83.

[Ahm14] Ahmed, M.; Cui, H.; and Huang, X. "Smart integration of cloud computing and mcmc based secured wsn to monitor environment". In "Wireless Communications, Vehicular Technology, Information Theory and Aerospace Electronic Systems (VITAE), 2014 4th International Conference on", pages 1–5. May 2014. doi:10.1109/VITAE.2014.6934449.

[Ahu93] Ahuja, R.K.; Magnanti, T.L.; and Orlin, J.B. *Network Flows: Theory, Algorithms, and Applications*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993. ISBN 0-13-617549-X.

[Ala13] Alamri, A.; Ansari, W.S.; Hassan, M.M.; Hossain, M.S.; Alelaiwi, A.; and Hossain, M.A. "A survey on sensor-cloud: Architecture, applications, and approaches". *International Journal of Distributed Sensor Networks*, 2013(917923), 2013.

[Alt15] Altamimi, M.; Abdrabou, A.; Naik, S.; and Nayak, A. "Energy cost models of smartphones for task offloading to the cloud". *Emerging Topics in Computing, IEEE Transactions on*, PP(99):1–1, 2015. ISSN 2168-6750. doi: 10.1109/TETC.2014.2387752.

[Alw13] Alwan, H. and Agarwal, A. "MQoSR: A multiobjective qos routing protocol for wireless sensor networks". *ISRN Sensor Networks*, 2013:12, 2013.

[Ari13] Arifuzzaman, M.; Matsumoto, M.; and Sato, T. "An intelligent hybrid MAC with traffic-differentiation-based QoS for wireless sensor networksqos for wireless sensor networks". *Sensors Journal, IEEE*, 13(6):2391–2399, June 2013. ISSN 1530-437X. doi:10.1109/JSEN.2013.2252163.

[Arj15] Arjun, D.; Bala, A.; Dwarakanath, V.; Sampada, K.; Prahlada Rao, B.; and Pasupuleti, H. "Integrating cloud-wsn to analyze weather data and notify saas user alerts during weather disasters". In "Advance Computing Conference (IACC), 2015 IEEE International", pages 899–904. June 2015. doi: 10.1109/IADCC.2015.7154835.

[Asl12] Aslam, M.; Rea, S.; and Pesch, D. "Service provisioning for the wsn cloud". In "Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on", pages 962–969. June 2012. ISSN 2159-6182. doi:10.1109/CLOUD.2012.132.

[Atm09] Atmel. "AT86RF230: Low power 2.4 ghz transceiver for zigbee, IEEE 802.15.4, 6LoWPAN, RF4CE and ISM applications", 2009.

[Atm14] Atmel. "Atmega 1281datasheet", February 2014.

[Aww09] Awwad, S.; Ng, C.; Noordin, N.; and Rasid, M. "Cluster based routing protocol for mobile nodes in wireless sensor network". In "Collaborative Technologies and Systems, 2009. CTS '09. International Symposium on", pages 233–241. May 2009. doi:10.1109/CTS.2009.5067486.

[Axe81] Axelrod, R. and Hamilton, W.D. "The evolution of cooperation". *Science*, 211(4489):1390–1396, 1981.

[Bac10] Bachir, A.; Dohler, M.; Watteyne, T.; and Leung, K. "MAC essentials for wireless sensor networks". *Communications Surveys Tutorials, IEEE*, 12(2):222 –248, quarter 2010. ISSN 1553-877X. doi:10.1109/SURV.2010.020510.00058.

[Bag10] Baghyalakshmi, D.; Ebenezer, J.; and Satyamurty, S. "Low latency and energy efficient routing protocols for wireless sensor networks". In "Wireless Communication and Sensor Computing, 2010. ICWCSC 2010. International Conference on", pages 1 –6. jan. 2010. doi:10.1109/ICWCSC.2010.5415892.

[Bar13a] Barcelo, M.; Correa, A.; Lopez Vicario, J.; and Morell, A. "Multi-tree routing for heterogeneous data traffic in wireless sensor networks". In "Communications (ICC), 2013 IEEE International Conference on", pages 1899–1903. June 2013. ISSN 1550-3607. doi:10.1109/ICC.2013.6654799.

[Bar13b] Barcelo, M.; Correa, A.; Lopez Vicario, J., J.; and Morell, A. "Cooperative multi-tree sleep scheduling for surveillance in wireless sensor networks". In "Military Communications Conference, MILCOM 2013 - 2013 IEEE", pages 200–205. Nov 2013. doi:10.1109/MILCOM.2013.43.

[Bar13c] Barcelo, M.; Correa, A.; Vicario, J.L.; and Morell, A. "Joint routing and transmission power control for collection tree protocol in WSN". In "Personal Indoor and Mobile Radio Communications (PIMRC), 2013 IEEE 24th International Symposium on", pages 1989–1993. Sept 2013. ISSN 2166-9570. doi: 10.1109/PIMRC.2013.6666470.

[Bar15] Barcelo, M.; Llorca, J.; Tulino, A.M.; and Raman, N. "The cloud service distribution problem in distributed cloud networks". In "Communications

(ICC), 2015 IEEE International Conference on", pages 344–350. June 2015. doi:10.1109/ICC.2015.7248345.

[Bon12] Bonomi, F.; Milito, R.; Zhu, J.; and Addepalli, S. "Fog computing and its role in the internet of things". In "Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing", MCC '12, pages 13–16. ACM, New York, NY, USA, 2012. ISBN 978-1-4503-1519-7. doi:10.1145/2342509.2342513.

[Bot14] Botta, A.; de Donato, W.; Persico, V.; and Pescape, A. "On the integration of cloud computing and internet of things". In "Future Internet of Things and Cloud (FiCloud), 2014 International Conference on", pages 23–30. Aug 2014. doi:10.1109/FiCloud.2014.14.

[Bra09] Bravos, G. and Kanatas, A.G. "Integrating power control with routing to satisfy energy and delay constraints in sensor networks". *European Transactions on Telecommunications*, 20(2):233–245, 2009. ISSN 1541-8251. doi: 10.1002/ett.1248.

[Cad13] Cadger, F.; Curran, K.; Santos, J.; and Moffett, S. "A survey of geographical routing in wireless ad-hoc networks". *Communications Surveys Tutorials, IEEE*, 15(2):621–653, Second 2013. ISSN 1553-877X. doi: 10.1109/SURV.2012.062612.00109.

[Cap12] Cappiello, C. and Schreiber, F.A. "Experiments and Analysis of Quality- and Energy-Aware Data Aggregation Approaches in WSNs". In "QDB 2012, 10th International Workshop on Quality in Databases", 2012.

[Che10] Chen, Y.; Nasser, N.; Salti, T.E.; and Zhang, H. "A multipath QoS routing protocol in wireless sensor networks". *Int. J. Sen. Netw.*, 7(4):207–216, 2010. ISSN 1748-1279. doi:10.1504/IJSNET.2010.033204.

[Chu11] Chung, T.P.; Lin, T.S.; Zheng, X.Y.; Yen, P.L.; and Jiang, J.A. "A load balancing algorithm based on probabilistic multi-tree for wireless sensor networks". In "Sensing Technology (ICST), 2011 Fifth International Conference on", pages 527 –532. 28 2011-dec. 1 2011. ISSN 2156-8065. doi: 10.1109/ICSensT.2011.6137035.

[Cob14] Cobarzan, C.; Montavont, J.; and Noel, T. "Analysis and performance evaluation of rpl under mobility". In "Computers and Communication (ISCC), 2014 IEEE Symposium on", pages 1–6. June 2014. doi:10.1109/ISCC.2014.6912471.

[Cor08] Correia, L. and Nogueira, J. "Transmission power control techniques for mac protocols in wireless sensor networks". In "Network Operations and Management Symposium, 2008. NOMS 2008. IEEE", pages 1049 –1054. april 2008. ISSN 1542-1201. doi:10.1109/NOMS.2008.4575277.

[Cor12] Correa, A.; Morell, A.; Barcelo, M.; and Vicario, J. "Navigation system for elderly care applications based on wireless sensor networks". In "Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European", pages 210–214. Aug 2012. ISSN 2219-5491.

[Cor14] Correa, A.; Barcelo, M.; Morell, A.; and Lopez Vicario, J. "Enhanced inertial-aided indoor tracking system for wireless sensor networks: A review". *Sensors Journal, IEEE*, 14(9):2921–2929, Sept 2014. ISSN 1530-437X. doi:10.1109/JSEN.2014.2325775.

[Cot14] Cotuk, H.; Bicakci, K.; Tavli, B.; and Uzun, E. "The impact of transmission power control strategies on lifetime of wireless sensor networks". *Computers, IEEE Transactions on*, 63(11):2866–2879, Nov 2014. ISSN 0018-9340. doi:10.1109/TC.2013.151.

[Cuo07] Cuomo, F.; Della Luna, S.; Monaco, U.; and Melodia, F. "Routing in zigbee: Benefits from exploiting the ieee 802.15.4 association tree". In "Communications, 2007. ICC '07. IEEE International Conference on", pages 3271–3276. June 2007. doi:10.1109/ICC.2007.542.

[Dun04] Dunkels, A.; Gronvall, B.; and Voigt, T. "Contiki - a lightweight and flexible operating system for tiny networked sensors". In "Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks", LCN '04, pages 455–462. IEEE Computer Society, Washington, DC, USA, 2004. ISBN 0-7695-2260-2. doi:10.1109/LCN.2004.38.

[Dur08] Durmaz Incel, O.; Jansen, P.; and Mullender, S. "MC-LMAC: A multi-channel mac protocol for wireless sensor networks", 2008.

[Dur12] Durmaz Incel, O.; Ghosh, A.; Krishnamachari, B.; and Chintalapudi, K. "Fast data collection in tree-based wireless sensor networks". *Mobile Computing, IEEE Transactions on*, 11(1):86 –99, jan. 2012. ISSN 1536-1233. doi:10.1109/TMC.2011.22.

[ElB11] ElBatt, T. and Andersen, T. "A cross-layer framework for multiple access and routing design in wireless multi-hop networks". *Wireless Communications and Mobile Computing*, 11(8):1155–1167, 2011. ISSN 1530-8677. doi:10.1002/wcm.902.

[Fu12] Fu, Y.; Sha, M.; Hackmann, G.; and Lu, C. "Practical control of transmission power for wireless sensor networks". In "Network Protocols (ICNP), 2012 20th IEEE International Conference on", pages 1 –10. 30 2012-nov. 2 2012. doi:10.1109/ICNP.2012.6459981.

[Gad06] Gadallah, Y. "A comparative study of routing strategies for wireless sensor networks: Are manet protocols good fit?" In "Proceedings of the 5th International Conference on Ad-Hoc, Mobile, and Wireless Networks", ADHOC-NOW'06, pages 5–18. Springer-Verlag, Berlin, Heidelberg, 2006. ISBN 3-540-37246-6, 978-3-540-37246-2. doi:10.1007/118147643.

[Gad12] Gaddour, O. and KoubíA, A. “Survey rpl in a nutshell: A survey”. *Comput. Netw.*, 56(14):3163–3178, September 2012. ISSN 1389-1286. doi: 10.1016/j.comnet.2012.06.016.

[Gad14] Gaddour, O.; Koubaa, A.; Rangarajan, R.; Cheikhrouhou, O.; Tovar, E.; and ABID, M. “Co-rpl: Rpl routing for mobile low power wireless sensor networks using corona mechanism”. In “Industrial Embedded Systems (SIES), 2014 9th IEEE International Symposium on”, pages 200–209. June 2014. doi: 10.1109/SIES.2014.6871205.

[Gna09] Gnawali, O.; Fonseca, R.; Jamieson, K.; Moss, D.; and Levis, P. “CTP: Robust and efficient collection through control and data plane integration”. Tech. rep. sing-09-01, Stanford University, April 2009.

[Gub13] Gubbi, J.; Buyya, R.; Marusic, S.; and Palaniswami, M. “Internet of things (iot): A vision, architectural elements, and future directions”. *Future Gener. Comput. Syst.*, 29(7):1645–1660, September 2013. ISSN 0167-739X. doi: 10.1016/j.future.2013.01.010.

[Gun13] Gungor, V.C. and Hancke, G.P. *Industrial Wireless Sensor Networks: Applications, Protocols, and Standards*. CRC Press, Inc., Boca Raton, FL, USA, 1st edition, 2013. ISBN 1466500514, 9781466500518.

[Guo12] Guo, P.; Jiang, T.; Zhang, Q.; and Zhang, K. “Sleep scheduling for critical event monitoring in wireless sensor networks”. *Parallel and Distributed Systems, IEEE Transactions on*, 23(2):345–352, 2012. ISSN 1045-9219. doi: 10.1109/TPDS.2011.165.

[Ha13] Ha, K.; Pillai, P.; Lewis, G.; Simanta, S.; Clinch, S.; Davies, N.; and Satyanarayanan, M. “The impact of mobile multimedia applications on data center consolidation”. In “Cloud Engineering (IC2E), 2013 IEEE International Conference on”, pages 166–176. March 2013. doi:10.1109/IC2E.2013.17.

[Haj11] Hajiaghayi, M.; Dong, M.; and Liang, B. “Optimal channel assignment and power allocation for dual-hop multi-channel multi-user relaying”. In “INFOCOM, 2011 Proceedings IEEE”, pages 76–80. April 2011. ISSN 0743-166X. doi:10.1109/INFCOM.2011.5935298.

[Har10] Harms, T.; Sedigh, S.; and Bastianini, F. “Structural health monitoring of bridges using wireless sensor networks”. *Instrumentation Measurement Magazine, IEEE*, 13(6):14 –18, december 2010. ISSN 1094-6969. doi: 10.1109/MIM.2010.5669608.

[Hei00] Heinzelman, W.; Chandrakasan, A.; and Balakrishnan, H. “Energy-efficient communication protocol for wireless microsensor networks”. In “System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on”, page 10 pp. vol.2. jan. 2000. doi:10.1109/HICSS.2000.926982.

[Hu15] Hu, X.; Yang, L.; and Xiong, W. "A novel wireless sensor network frame for urban transportation". *Internet of Things Journal, IEEE*, PP(99):1–1, 2015. ISSN 2327-4662. doi:10.1109/JIOT.2015.2475639.

[Hua12] Huang, P.; Xiao, L.; Soltani, S.; Mutka, M.; and Xi, N. "The evolution of MAC protocols in wireless sensor networks: A survey". *Communications Surveys Tutorials, IEEE*, PP(99):1 –20, 2012. ISSN 1553-877X. doi:10.1109/SURV.2012.040412.00105.

[IEE11] IEEE. *IEEE Standard 802.15.4, "Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)*, June 2011.

[Ish13] Ishaq, I.; Carels, D.; Teklemariam, G.K.; Hoebeke, J.; Abeele, F.V.d.; Poorter, E.D.; Moerman, I.; and Demeester, P. "Ietf standardization in the field of the internet of things (iot): A survey". *Journal of Sensor and Actuator Networks*, 2(2):235, 2013. ISSN 2224-2708. doi:10.3390/jsan2020235.

[Jia15] Jian, X.; Geng, Y.; Zhengyu, C.; and Qianqian, W. "A survey on the privacy-preserving data aggregation in wireless sensor networks". *Communications, China*, 12(5):162–180, May 2015. ISSN 1673-5447. doi:10.1109/CC.2015.7112038.

[Joh01] Johnson, D.B.; Maltz, D.A.; and Broch, J. "Ad hoc networking". chapter DSR: The Dynamic Source Routing Protocol for Multihop Wireless Ad Hoc Networks, pages 139–172. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2001. ISBN 0-201-30976-9.

[Jov11] Jovanovic, M.; Djordjevic, G.; Nikolic, G.; and Petrovic, B. "Multi-channel media access control for wireless sensor networks: A survey". In "Telecommunication in Modern Satellite Cable and Broadcasting Services (TELSIKS), 2011 10th International Conference on", volume 2, pages 741–744. Oct 2011. doi:10.1109/TELSKS.2011.6143217.

[Kal60] Kalman, R.E. "A new approach to linear filtering and prediction problems". *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[Kar00] Karp, B.N. "Geographic routing for wireless networks". Technical report, Harvard University, 2000.

[Kim05] Kim, Y.; Govindan, R.; Karp, B.; and Shenker, S. "On the pitfalls of geographic face routing". In "in Proc. of DIALM-POMC", pages 34–43. ACM Press, 2005.

[Kim06] Kim, D.S. and Chung, Y.J. "Self-organization routing protocol supporting mobile nodes for wireless sensor network". In "Computer and Computational Sciences, 2006. IMSCCS '06. First International Multi-Symposiums on", volume 2, pages 622–626. June 2006. doi:10.1109/IMSCCS.2006.265.

[Ko11] Ko, J.; Dawson-Haggerty, S.; Gnawali, O.; Culler, D.; and Terzis, A. "Evaluating the Performance of RPL and 6LoWPAN in TinyOS". In "Proceedings of the Workshop on Extending the Internet to Low power and Lossy Networks", IP+SN '11. April 2011.

[Kor12] Korbi, I.; Ben Brahim, M.; Adjih, C.; and Saidane, L. "Mobility enhanced rpl for wireless sensor networks". In "Network of the Future (NOF), 2012 Third International Conference on the", pages 1–8. Nov 2012. doi:10.1109/NOF.2012.6463993.

[Kuh03] Kuhn, F.; Wattenhofer, R.; and Zollinger, A. "Worst-case optimal and average-case efficient geometric ad-hoc routing", 2003.

[Kum08] Kumar, G.; Vinu Paul, M.; and Jacob, K. "Mobility metric based leach-mobile protocol". In "Advanced Computing and Communications, 2008. ADCOM 2008. 16th International Conference on", pages 248–253. Dec 2008. doi:10.1109/ADCOM.2008.4760456.

[Kum10] Kumar, K. and Lu, Y.H. "Cloud computing for mobile users: Can offloading computation save energy?" *Computer*, 43(4):51–56, April 2010. ISSN 0018-9162. doi:10.1109/MC.2010.98.

[Kyö07] Kyösti, P.; Meinilä, J.; Hentilä, L.; Zhao, X.; and Jämsä, T. "Winner ii channel models". Technical report, EBITG, TUI, UOULU, CU/CRC, NOKIA, 2007.

[Lee12] Lee, K.; Sudhaakar, R.; and Ning, J. "A comprehensive evaluation of RPL under mobility". *International Journal of Vehicular Technology*, 2012:10, 2012.

[Lei13] Lei, L.; Zhong, Z.; Zheng, K.; Chen, J.; and Meng, H. "Challenges on wireless heterogeneous networks for mobile cloud computing". *Wireless Communications, IEEE*, 20(3):34–44, June 2013. ISSN 1536-1284. doi:10.1109/MWC.2013.6549281.

[Lev11] Levis, P.; Clausen, T.; Hui, J.; Gnawaliand, O.; and Ko, J. "Rfc 6206: The trickle algorithm". In "Internet Engineering Task Force", March 2011.

[Li07] Li, X. "Collaborative localization with received-signal strength in wireless sensor networks". *Vehicular Technology, IEEE Transactions on*, 56(6):3807–3817, 2007. ISSN 0018-9545. doi:10.1109/TVT.2007.904535.

[Li11] Li, J.; Guo, X.; and Guo, L. "Joint routing, scheduling and channel assignment in multi-power multi-radio wireless sensor networks". In "Performance Computing and Communications Conference (IPCCC), 2011 IEEE 30th International", pages 1 –8. nov. 2011. ISSN 1097-2641. doi:10.1109/PCCC.2011.6108087.

[Liu15] Liu, X.; Cao, J.; Song, W.Z.; Guo, P.; and He, Z. "Distributed sensing for high-quality structural health monitoring using wsns". *Parallel and Distributed Systems, IEEE Transactions on*, 26(3):738–747, March 2015. ISSN 1045-9219. doi:10.1109/TPDS.2014.2312911.

[Llo13] Llorca, J. and Tulino, A. "The content distribution problem and its complexity classification". Technical report, Alcatel-Lucent, 2013.

[Lon13] Long, N.T.; Uwase, M.P.; Tiberghien, J.; and Steenhaut, K. "QoS-aware cross-layer mechanism for multiple instances RPL". In "Advanced Technologies for Communications (ATC), 2013 International Conference on", pages 44–49. Oct 2013. ISSN 2162-1020. doi:10.1109/ATC.2013.6698074.

[Luo10] Luo, J.; Rosenberg, C.; and Girard, A. "Engineering wireless mesh networks: Joint scheduling, routing, power control, and rate adaptation". *Networking, IEEE/ACM Transactions on*, 18(5):1387 –1400, oct. 2010. ISSN 1063-6692. doi:10.1109/TNET.2010.2041788.

[Luo11] Luo, H.; Tao, H.; Ma, H.; and Das, S. "Data fusion with desired reliability in wireless sensor networks". *Parallel and Distributed Systems, IEEE Transactions on*, 22(3):501 –513, march 2011. ISSN 1045-9219. doi:10.1109/TPDS.2010.93.

[Luo12] Luo, T.; Tan, H.P.; and Quek, T. "Sensor openflow: Enabling software-defined wireless sensor networks". *Communications Letters, IEEE*, 16(11):1896–1899, November 2012. ISSN 1089-7798. doi:10.1109/LCOMM.2012.092812.121712.

[Mal06] Malakooti, B. and Thomas, I. "A distributed composite multiple criteria routing using distance vector". In "Networking, Sensing and Control, 2006. ICNSC '06. Proceedings of the 2006 IEEE International Conference on", pages 42–47. 2006. doi:10.1109/ICNSC.2006.1673115.

[Man01] Mani, C.S. and Srivastava, M.B. "Energy efficient routing in wireless sensor networks". In U. McLean, editor, "Military Communications Conference Proceedings on Communications for Network-Centric Operations: Creating the Information Force", volume 1, pages 357–361. October 2001.

[May79] Maybeck, P.S. *Stochastic Models: Estimation and Control: v. 1: Estimation and Control:*, volume 1 of *Mathematics in Science and Engineering*. Elsevier Science, 1979. ISBN 9780080956503.

[McK08] McKeown, N.; Anderson, T.; Balakrishnan, H.; Parulkar, G.; Peterson, L.; Rexford, J.; Shenker, S.; and Turner, J. "Openflow: Enabling innovation in campus networks". *SIGCOMM Comput. Commun. Rev.*, 38(2):69–74, March 2008. ISSN 0146-4833. doi:10.1145/1355734.1355746.

[Md13] Melo de, P.; Cunha, F.; and Loureiro, A. "A distributed protocol for cooperation among different wireless sensor networks". In "Communications (ICC), 2013 IEEE International Conference on", pages 6035–6039. June 2013. ISSN 1550-3607. doi:10.1109/ICC.2013.6655566.

[Meg11] Meghji, M. and Habibi, D. "Transmission power control in multihop wireless sensor networks". In "Ubiquitous and Future Networks (ICUFN), 2011 Third International Conference on", pages 25 –30. june 2011. doi: 10.1109/ICUFN.2011.5949130.

[MEM] MEMSIC. "IRIS datasheet".

[Mie03] Mieghem, P.V. and Kuipers, F.A. "On the complexity of QoS routing". *Computer Communications*, 26:376–387, 2003.

[Mis14a] Misra, S.; Bera, S.; Mondal, A.; Tirkey, R.; Chao, H.C.; and Chattopadhyay, S. "Optimal gateway selection in sensor-cloud framework for health monitoring". *Wireless Sensor Systems, IET*, 4(2):61–68, June 2014. ISSN 2043-6386. doi: 10.1049/iet-wss.2013.0073.

[Mis14b] Misra, S.; Chatterjee, S.; and Obaidat, M. "On theoretical modeling of sensor cloud: A paradigm shift from wireless sensor network". *Systems Journal, IEEE*, PP(99):1–10, 2014. ISSN 1932-8184. doi:10.1109/JSYST.2014.2362617.

[Mit12] Mitton, N.; Papavassiliou, S.; Puliafito, A.; and Trivedi, K. "Combining cloud and sensors in a smart city environment". *EURASIP Journal on Wireless Communications and Networking*, 2012(1):247, 2012. doi:10.1186/1687-1499-2012-247.

[Mor13] Morell, A.; Vilajosana, X.; Vicario, J.L.; and Watteyne, T. "Label switching over ieee802.15.4e networks". *Transactions on Emerging Telecommunications Technologies*, pages n/a–n/a, 2013. ISSN 2161-3915. doi:10.1002/ett.2650.

[MS05] Martinez-Sala, A.; Molina-Garcia-Pardo, J.M.; Egea-Ldpez, E.; Vales-Alonso, J.; Juan-Llacer, L.; and Garcia-Haro, J. "An accurate radio channel model for wireless sensor networks simulation". *Communications and Networks, Journal of*, 7(4):401–407, Dec 2005. ISSN 1229-2370. doi:10.1109/JCN.2005.6387982.

[Nas14] Nastic, S.; Sehic, S.; Le, D.H.; Truong, H.L.; and Dustdar, S. "Provisioning software-defined iot cloud systems". In "Future Internet of Things and Cloud (FiCloud), 2014 International Conference on", pages 288–295. Aug 2014. doi: 10.1109/FiCloud.2014.52.

[Pan14] Pantazis, N.; Nikolidakis, S.; and Vergados, D. "Energy-efficient routing protocols in wireless sensor networks: A survey". *Communications Surveys Tutorials, IEEE*, PP(99):1–41, 2014. ISSN 1553-877X. doi: 10.1109/SURV.2012.062612.00084.

[Pap02] Papoulis, A. and Pillai, S. *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill series in electrical engineering: Communications and signal processing. Tata McGraw-Hill, 2002. ISBN 9780070486584.

[Pat03] Patwari, N.; Hero, A.; Perkins, M.; Correal, N.; and O'Dea, R. "Relative location estimation in wireless sensor networks". *Signal Processing, IEEE Transactions on*, 51(8):2137–2148, 2003. ISSN 1053-587X. doi: 10.1109/TSP.2003.814469.

[Pen10] Peng, B. and Kemp, A. "Impact of location errors on geographic routing in realistic wsns". In "Indoor Positioning and Indoor Navigation (IPIN), 2010 International Conference on", pages 1–7. Sept 2010. doi:10.1109/IPIN.2010.5647292.

[Pen11] Peng, B. and Kemp, A. "Energy-efficient geographic routing in the presence of localization errors". *Computer Networks*, 55(3):856 – 872, 2011. ISSN 1389-1286. doi:http://dx.doi.org/10.1016/j.comnet.2010.10.020.

[Per99] Perkins, C. and Royer, E. "Ad-hoc on-demand distance vector routing". In "Mobile Computing Systems and Applications, 1999. Proceedings. WMCSA '99. Second IEEE Workshop on", pages 90–100. Feb 1999. doi:10.1109/MCSA.1999.749281.

[Pu12] Pu, C.C.; Lim, S.Y.; and Ooi, P.C. "Measurement arrangement for the estimation of path loss exponent in wireless sensor network". In "Computing and Convergence Technology (ICCCT), 2012 7th International Conference on", pages 807–812. 2012.

[Raj14] Rajalingham, G.; Gao, Y.; Ho, Q.D.; and Le-Ngoc, T. "Quality of service differentiation for smart grid neighbor area networks through multiple RPL instances-rpl instances". In "Proceedings of the 10th ACM Symposium on QoS and Security for Wireless and Mobile Networks", Q2SWinet '14, pages 17–24. ACM, New York, NY, USA, 2014. ISBN 978-1-4503-3027-5. doi:10.1145/2642687.2642695.

[Ram06] Raman, B.; Chebrolu, K.; Madabhushi, N.; Gokhale, D.Y.; Valiveti, P.K.; and Jain, D. "Implications of link range and (in)stability on sensor network architecture". WiNTECH '06, pages 65–72. ACM, New York, NY, USA, 2006. ISBN 1-59593-539-8. doi:10.1145/1160987.1161000.

[Rao15] Rao, S. and Mehta, N. "Energy harvesting wsns for accurately estimating the maximum sensor reading: Trade-offs and optimal design". *Wireless Communications, IEEE Transactions on*, 14(8):4562–4573, Aug 2015. ISSN 1536-1276. doi:10.1109/TWC.2015.2422811.

[Rap01] Rappaport, T. *Wireless Communications: Principles and Practice*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2nd edition, 2001. ISBN 0130422320.

[Raw14] Rawat, P.; Singh, K.D.; Chaouchi, H.; and Bonnin, J.M. "Wireless sensor networks: A survey on recent developments and potential synergies". *J. Supercomput.*, 68(1):1–48, April 2014. ISSN 0920-8542. doi:10.1007/s11227-013-1021-9.

[Rib10] Ribeiro, A.; Schizas, I.; Roumeliotis, S.; and Giannakis, G. "Kalman filtering in wireless sensor networks". *Control Systems, IEEE*, 30(2):66–86, April 2010. ISSN 1066-033X. doi:10.1109/MCS.2009.935569.

[Saa11] Saad, L. and Tourancheau, B. "Sinks mobility strategy in IPv6-based WSNs for network lifetime improvement". In "New Technologies, Mobility and Security

(NTMS), 2011 4th IFIP International Conference on", pages 1–5. Feb 2011. ISSN 2157-4952. doi:10.1109/NTMS.2011.5720597.

[Saf12] Safdar, V.; Bashir, F.; Hamid, Z.; Afzal, H.; and Pyun, J.Y. "A hybrid routing protocol for wireless sensor networks with mobile sinks". In "Wireless and Pervasive Computing (ISWPC), 2012 7th International Symposium on", pages 1–5. July 2012. doi:10.1109/ISWPC.2012.6263665.

[Sai14] Saifullah, A.; Xu, Y.; Lu, C.; and Chen, Y. "Distributed channel allocation protocols for wireless sensor networks". *Parallel and Distributed Systems, IEEE Transactions on*, 25(9):2264–2274, Sept 2014. ISSN 1045-9219. doi:10.1109/TPDS.2013.185.

[San15] Santoshkumar and Udaykumar, R. "Development of wsn system for precision agriculture". In "Innovations in Information, Embedded and Communication Systems (ICIIECS), 2015 International Conference on", pages 1–5. March 2015. doi:10.1109/ICIIECS.2015.7192904.

[Sar14] Sara, G. and Sridharan, D. "Routing in mobile wireless sensor network: a survey". *Telecommunication Systems*, 57(1):51–79, 2014. ISSN 1018-4864. doi:10.1007/s11235-013-9766-2.

[Sat09] Satyanarayanan, M.; Bahl, P.; Caceres, R.; and Davies, N. "The case for vm-based cloudlets in mobile computing". *Pervasive Computing, IEEE*, 8(4):14–23, Oct 2009. ISSN 1536-1268. doi:10.1109/MPRV.2009.82.

[Sat15] Satyanarayanan, M.; Schuster, R.; Ebling, M.; Fettweis, G.; Flinck, H.; Joshi, K.; and Sabnani, K. "An open ecosystem for mobile-cloud convergence". *Communications Magazine, IEEE*, 53(3):63–70, March 2015. ISSN 0163-6804. doi:10.1109/MCOM.2015.7060484.

[Sha13] Shamsan Saleh, A.M.; Ali, B.M.; Rasid, M.F.A.; and Ismail, A. "A survey on energy awareness mechanisms in routing protocols for wireless sensor networks using optimization methods". *Transactions on Emerging Telecommunications Technologies*, pages n/a–n/a, 2013. ISSN 2161-3915. doi:10.1002/ett.2679.

[Shi13] Shiqiang, et al, W. "Mobile micro-cloud: Application classification, mapping, and deployment". In "Annual Fall Meeting of ITA (AMITA)", 2013.

[Shi14] Shi, H.; Prasad, R.; Onur, E.; and Niemegeers, I. "Fairness in wireless networks:issues, measures and challenges". *Communications Surveys Tutorials, IEEE*, 16(1):5–24, First 2014. ISSN 1553-877X. doi:10.1109/SURV.2013.050113.00015.

[Siv10] Sivagami, A.; Pavai, K.; Sridharan, D.; and SatyaMurty, S. "Estimating the energy consumption of wireless sensor node: Iris". *International Journal on Recent Trends in Engineering and Technology*, 3(4):141–143, May 2010.

[Sto14] Stojmenovic, I. and Wen, S. "The fog computing paradigm: Scenarios and security issues". In "Computer Science and Information Systems (FedCSIS), 2014 Federated Conference on", pages 1–8. Sept 2014. doi:10.15439/2014F503.

[Sur15] Suryadevara, N.; Mukhopadhyay, S.; Kelly, S.; and Gill, S. "Wsn-based smart sensors and actuator for power management in intelligent buildings". *Mechatronics, IEEE/ASME Transactions on*, 20(2):564–571, April 2015. ISSN 1083-4435. doi:10.1109/TMECH.2014.2301716.

[Tia13] Tian, B.; Hou, K.M.; Shi, H.; Liu, X.; Diao, X.; Li, J.; Chen, Y.; and Chanet, J.P. "Application of modified rpl under vanet-wsn communication architecture". In "Computational and Information Sciences (ICCIS), 2013 Fifth International Conference on", pages 1467–1470. June 2013. doi:10.1109/ICCIS.2013.387.

[Tse02] Tseng, Y.C.; Hsu, C.S.; and Hsieh, T.Y. "Power-saving protocols for IEEE 802.11-based multi-hop ad hoc networks". In "INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE", volume 1, pages 200–209 vol.1. 2002. ISSN 0743-166X. doi:10.1109/INFCOM.2002.1019261.

[Vil10] Vilajosana, X.; Llosa, J.; Pacho, J.C.; Vilajosana, I.; Juan, A.A.; Vicario, J.L.; and Morell, A. "Zero: Probabilistic routing for deploy and forget wireless sensor networks". *Sensors*, 10(10):8920–8937, 2010. ISSN 1424-8220. doi:10.3390/s101008920.

[Vil14] Vilajosana, X. and Pister, K. "Minimal 6TiSCH configuration". Internet-draft, 6TiSCH, April 2014.

[Vis15] Vishwanath, A.; Jalali, F.; Hinton, K.; Alpcan, T.; Ayre, R.; and Tucker, R. "Energy consumption comparison of interactive cloud-based and local applications". *Selected Areas in Communications, IEEE Journal on*, 33(4):616–626, April 2015. ISSN 0733-8716. doi:10.1109/JSAC.2015.2393431.

[Wan06] Wang, A. and Sodini, C. "On the energy efficiency of wireless transceivers". In "Communications, 2006. ICC '06. IEEE International Conference on", volume 8, pages 3783–3788. June 2006. ISSN 8164-9547. doi:10.1109/ICC.2006.255661.

[Wan10] Wang, Q. "Traffic analysis and modeling in wireless sensor networks and their applications on network optimization and anomaly detection". *Network Protocols and Algorithms, Macrothink Institute*, 2(1), 2010.

[Wat11] Watteyne, T.; Molinaro, A.; Richichi, M.; and Dohler, M. "From manet to ietf roll standardization: A paradigm shift in wsn routing protocols". *Communications Surveys Tutorials, IEEE*, 13(4):688–707, 2011. ISSN 1553-877X. doi:10.1109/SURV.2011.082710.00092.

[Wei07] Wei, W. and Zakhor, A. “Multiple tree video multicast over wireless ad hoc networks”. *Circuits and Systems for Video Technology, IEEE Transactions on*, 17(1):2–15, Jan 2007. ISSN 1051-8215. doi:10.1109/TCSVT.2006.885719.

[Wel95] Welch, G. and Bishop, G. “An introduction to the kalman filter”. Technical report, Chapel Hill, NC, USA, 1995.

[Wen14] Wenge, R.; Zhang, X.; Dave, C.; Chao, L.; and Hao, S. “Smart city architecture: A technology guide for implementation and design challenges”. *Communications, China*, 11(3):56–69, March 2014. ISSN 1673-5447. doi:10.1109/CC.2014.6825259.

[Win12] Winter, T.; Thubert, P.; Brandt, A.; Clausen, T.; Hui, J.; Kelsey, R.; Levis, P.; Pister, K.; Struick, R.; and Vasseur, J. “Rpl: Ipv6 routing protocol for low-power and lossy networks”. In “IETF RFC 6550”, March 2012.

[Wu08] Wu, Y.; Stankovic, J.; He, T.; and Lin, S. “Realistic and efficient multi-channel communications in wireless sensor networks”. In “INFOCOM 2008. The 27th Conference on Computer Communications. IEEE”, pages 1193 –1201. april 2008. ISSN 0743-166X. doi:10.1109/INFOCOM.2008.175.

[Xia10] Xiao, Y.; Chen, H.; and Haizhon., F. *Handbook on Sensor Networks*. World Scientific Publishing Co., 2010. ISBN 978-981-283-730-1.

[Xia13] Xiao, Y.; Simoens, P.; Pillai, P.; Ha, K.; and Satyanarayanan, M. “Lowering the barriers to large-scale mobile crowdsensing”. In “Proceedings of the 14th Workshop on Mobile Computing Systems and Applications”, HotMobile ’13, pages 9:1–9:6. ACM, New York, NY, USA, 2013. ISBN 978-1-4503-1421-3. doi:10.1145/2444776.2444789.

[Yet12] Yetgin, H.; Cheung, K.; and Hanzo, L. “Multi-objective routing optimization using evolutionary algorithms”. In “Wireless Communications and Networking Conference (WCNC), 2012 IEEE”, pages 3030–3034. April 2012. ISSN 1525-3511. doi:10.1109/WCNC.2012.6214324.

[Yi15] Yi, S.; Li, C.; and Li, Q. “A survey of fog computing: Concepts, applications and issues”. In “Proceedings of the 2015 Workshop on Mobile Big Data”, Mobidata ’15, pages 37–42. ACM, New York, NY, USA, 2015. ISBN 978-1-4503-3524-9. doi:10.1145/2757384.2757397.

[Yoo14] Yoon, J.; Zhang, H.; Banerjee, S.; and Rangarajan, S. “Video multicast with joint resource allocation and adaptive modulation and coding in 4g networks”. *Networking, IEEE/ACM Transactions on*, 22(5):1531–1544, Oct 2014. ISSN 1063-6692. doi:10.1109/TNET.2013.2279887.

[Yu01] Yu, Y.; Govindan, R.; and Estrin, D. “Geographical and energy aware routing: A recursive data dissemination protocol for wireless sensor networks”. In “UCLA Computer Science Department Technical Report”, pages 1–11. 2001.

[Zam08] Zamalloa, M.Z.n.; Seada, K.; Krishnamachari, B.; and Helmy, A. "Efficient geographic routing over lossy links in wireless sensor networks". *ACM Trans. Sen. Netw.*, 4(3):12:1–12:33, June 2008. ISSN 1550-4859. doi:10.1145/1362542.1362543.

[Zha12] Zhang, G.a.; Gu, J.y.; Bao, Z.h.; Xu, C.; and Zhang, S.b. "Joint routing and channel assignment algorithms in cognitive wireless mesh networks". *Transactions on Emerging Telecommunications Technologies*, pages n/a–n/a, 2012. ISSN 2161-3915. doi:10.1002/ett.2560.

[Zha14] Zhang, Y.; Sun, L.; Song, H.; and Cao, X. "Ubiquitous wsn for healthcare: Recent advances and future prospects". *Internet of Things Journal, IEEE*, 1(4):311–318, Aug 2014. ISSN 2327-4662. doi:10.1109/JIOT.2014.2329462.

[Zho06] Zhou, G.; Huang, C.; Yan, T.; He, T.; Stankovic, J.; and Abdelzaher, T. "Mmsn: Multi-frequency media access control for wireless sensor networks". In "INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings", pages 1–13. April 2006. ISSN 0743-166X. doi:10.1109/INFOCOM.2006.250.

[Zhu15] Zhu, C.; Sheng, Z.; Leung, V.; Shu, L.; and Yang, L. "Toward offering more useful data reliably to mobile cloud from wireless sensor network". *Emerging Topics in Computing, IEEE Transactions on*, 3(1):84–94, March 2015. ISSN 2168-6750. doi:10.1109/TETC.2014.2364921.