Tunisian Republic
Higher Education and
Scientific Research in Tunisia

University of Sfax
The Faculty of Economics
and Management of Sfax

Dissertation

Submitted in partial fulfillment of the requirements for the degree of

Master

in Information Systems and New Technologies

An Experimental Study for the Performance Evaluation and Optimization of Link Quality Estimators in Wireless Sensor Networks

Prepared by

Maissa Ben Jamaa

Defended the 27th of July 2010 in the presence of jury:

Mr. Ahmed HADJ KACEM	President
Mr. Khalil DRIRA	Referee
Mr. Mohamed JMAIEL	Advisor
Mr. Anis KOUBÂA	Co-Advisor
Mrs. Nouha BACCOUR	Co-Advisor

University year 2009-2010

Acknowledgments

First of all, I would like to express my extreme gratitude to Prof. Dr. Mohamed Jmaiel, for giving me the chance to join his work team, as well as for supervising and guiding my Master research. His kindness and support were helpful and motivating.

Moreover, I would like to thank my co-supervisor, Dr. Anis Koubâa, for the interest he showed toward this work, his valuable suggestions and fruitful discussions.

Further, a special thank goes to Mrs. Nouha Baccour, for her instantaneous responsiveness, her outstanding help and her constant enthusiasm toward this work. She was always available to give me both academic and moral support. Her constructive advices and her patience were very important to promote the elaboration of this Master Thesis.

Also, I would like to express my respectful thanks to Prof. Dr Ahmed Hadj Kacem, Professor of Computer Science in the Faculty of Economics and Management of Sfax-Tunisia, for giving me the honor of being the president of my Master jury.

Furthermore, I deeply thanks Dr. Khalil Drira, Research Associate at LAAS-CNRS, for accepting the reviewing and the judgment of my Master work.

My sincere gratitude goes to Prof. Dr Habib Youssef, Prof. Dr Mario Alves, Prof Dr. Leandro Buss Becker and all RadiaLE team for their academic support.

I am also very grateful to Denis Lima Do Rosario, Dr. Geoffrey Werner Challen and all MOTELAB team for their experimental help.

Finally, I deeply thank my dear parents and my dear sisters for their infinite encouragement and for their love.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

Wireless Sensor Networks (WSNs) are an instance of wireless networks consisting of a set of small and battery powered devices equipped with sensing and communication capabilities. These devices are intended to collect measurements from their surrounding environment, such as temperature, sound, pressure, motion etc and then to report them over radio links to a base station (referred also as sink node). However, several studies have shown that radio links in wireless sensor networks are unreliable for communication due to the erratic variation of their quality over time and space [53, 57]. An important mechanism to cope with this unreliability at higher layer protocols specifically routing protocols, is link quality estimation [10]. In fact, link quality estimation eases the distinguishing between links having good, medium and bad quality. Therefore, transmitting over bad quality links is avoided which decreases excessive retransmissions caused by packet losses and consequently decreases power consumption. Further, estimating the link quality enables sensor nodes to transmit their data over good quality links which improves the network performance in terms of throughput and energy-efficiency. Ensuring such performance is closely related to the accuracy of link quality estimates. Indeed, poor link quality estimates can cause 200% or greater slowdown in network throughput [21]. Thus, evaluating the performance of Link Quality Estimators (LQEs) becomes mandatory. This master thesis addresses the performance evaluation and optimization of a set of Link Quality Estimators (LQEs). In [10], an extensive comparative performance study of five well-known LQEs has been conducted, using TOSSIM 2 simulator [35]. In this study, the authors first analyzed the statistical properties of LQEs independently of higher-layer protocols. Then, they investigated their impact on the Collection Tree Routing protocol (CTP) [2]. This work was a first and outstanding step that might help network designers to choose the most appropriate LQE for their higher layer protocols. However, the performance analysis of LQEs, together with related conclusions were based on TOSSIM2 simulation. The validity of the simulation results greatly depends on the accuracy of TOSSIM2 channel model. Therefore, it is important to evaluate the performance of LQEs based on real experimentation, in order to get more realistic and trust-able observations. In[11], the authors extended the first evaluation methodology presented in [10] (i.e. the analysis of LQEs

statistical properties) to validate their new estimator, called F-LQE (Fuzzy Link Quality Estimator). However, the validation of F-LQE might be not conclusive as it is based on simulation. Furthermore, the impact of F-LQE on higher layer protocols has not been investigated.

1.2 Objectives

The main objective of this master thesis is to replicate the comparative simulation study of LQEs conducted in [10] into an experimental study on real WSNs platform in order to provide more rigorous conclusions. The study presented in [10] involves five representative LQEs, namely PRR (Packet Reception Ratio), WMEWMA (Window Mean Exponential Weight Moving Average), ETX (Expected Transmission count), RNP (Requested Number of Packets), and Fourbit. On the other hand, F-LQE, a new LQE based on Fuzzy logic has been introduced and validated through simulations in [11]. This master aims also to involve F-LQE at the experimental study of LQEs, which allows its experimental validation by comparing its performances to existing LQEs. An optimization of F-LQE for better performances is also envisaged in this master.

1.3 Contributions

The contributions of this master thesis are both practical and theoretical.

1.3.1 Theoretical Contributions

The first contribution is to evaluate the performance of existing LQEs (PRR, WMEWMA, ETX, RNP and Fourbit) based on real experimentations by studying their statistical properties as well as their impact on CTP routing protocol.

The second contribution [12] is to participate at the validation of F-LQE by comparing its statistical properties to those of the existing LQEs.

The third contribution is to study the impact of FLQE on CTP routing protocol.

The fourth contribution is to optimize FLQE implementation for better performance by using both simulation and real experimentations.

1.3.2 Practical Contributions

The first contribution [9] is to participate at the implementation and the deployment of an experimental testbed called RadiaLE, which aims at automating the experimental performance evaluation of LQEs. RadiaLE adopts the first evaluation methodology introduced in [10], which consists in analyzing the statistical properties of LQEs independently of higher layer protocols, namely MAC and routing protocols.

The second contribution is to update RadiaLE functionalities in order to adopt the second evaluation methodology introduced in [10], which consists in investigating the impact of LQEs on CTP routing protocol.

1.4 Research Context

This Master Thesis was developed within ReDCAD research unit at ENIS (Ecole Nationale d'Ingénieurs de Sfax, University of Sfax), in collaboration with (1) CISTER Research Unit at ISEP/IPP (Instituto Superior de Engenharia do Porto/ Instituto Politécnico do Porto) (2) COINS Research Unit at CCIS/IMAMU (College of Computer and Information Sciences / Al-Imam Mohamed bin Saud University) and (3) the Department of Automation and Systems (DAS) of the Federal University of Santa Catarina (UFSC). Another institution has partially contributed to the development of this thesis, which is Prince Research Unit, at the University of Sousse. This collaboration was in the context of the Ph.D. work of Nouha Baccour, which is performed in collaboration between ReDCAD research unit at ENIS (Ecole Nationale des Ingénieurs de Sfax) and CISTER Research Group at ISEP/IPP (Instituto Superior de Engenharia do Porto/Instituto Politécnico do Porto). One practical objective in Nouha's PhD is the experimentation of LQEs using real deployment to validate the performance of a newly proposed fuzzy link quality estimator, F-LQE by comparing its performance to those of existing LQEs. For that purpose, Nouha Baccour has proposed two evaluation methodologies:

The first evaluation methodology consists in analyzing statistical properties of LQEs, independently of higher layer protocols, namely MAC and routing protocols. Hence, Nouha has designed RadiaLE, a test-bed that allows for the performance evaluation of LQEs based on this methodology. RadiaLE comprises the hardware components of the WSN test-bed (TelosB nodes, USB cables/hubs) and a software tool for setting up and controlling the experiments and also for analyzing the collected data, allowing for LQEs evaluation. The implementation of RadiaLE functionalities as well as its deployment at Porto has been done jointly by Maissa Ben Jamaa and Denis Do Rosario in the context of their master thesis work. The work of Maissa consisted mainly of implementing a MATLAB application for analyzing the experimental data and generating graphs and statistics. On the other hand , the work of Denis was mainly the deployment of RadiaLE at ISEP/IPP. A joint work between Maissa and Denis was the implementation of a Java application for the experiment control.

The second evaluation methodology consists in studying the impact of LQEs on higher layer protocols - specifically CTP routing protocol. Particularly, this study aims at (*i.*) the identification of F-LQE limitations when integrated to CTP and then (*ii.*) its optimization in order to get better performances. This study has been carried out in the context Maissa Ben Jamaa master thesis.

1.5 Structure Of The Thesis

The remainder of this Thesis is structured as follows. Chapter 2 gives a brief overview about WSNs technology followed by a description of WSNs radio link characteristics. Then, a review of most representative Link Quality Estimators (LQEs) in the literature is given.
The performance evaluation strategy consists in experimentally (1) studying the statistical properties of existing LQEs and then (2) investigating their impact on a higher layer protocol. For that, chapter 3 is devoted to present our experimental testbed called RadiaLE exploited to evaluate the performance of different LQEs. Chapter 3 starts by surveying the state of the art of existing testbeds designed for the experimentation of WSN. Chapter 4 and 5 describe respectively the experiments setup, scenario and results of both statistical properties and impact on routing studies. Chapter 6 introduces our strategy and our experimental results for the optimization of FLQE Link Quality Estimator. FLQE [12] is a recently proposed Link Quality Estimator. A general conclusion and future work are finally presented at the end of this Thesis.

Part I

General Background

Chapter 2

Wireless Sensor Networks and Link Quality Estimation

2.1 Introduction

Wireless Sensor Networks (WSNs) are more and more tightly integrated at different practical real life levels such as security ensuring, environment monitoring, home automation, enemy tracking, etc. This high integration illustrates how powerful this recent technology is. Indeed, the deployment of a WSN boosts human interaction with his physical environment in terms of fastening and easing data collection. However, efficient data collection relies on reliable communication. Radio (wireless) links are often lossy and error prone. Thus, fulfilling reliable communication remains one of the main challenges for wireless networks in general, and WSNs in particular. In fact, it has been shown in several studies [53, 57, 56, 33, 8, 60] that radio links in WSNs often have asymmetric and irregular qualities over time and space which makes the communication over them very unreliable and losses prone. To address this challenge, many work have been investigated in developing mechanisms called "Link Quality Estimators" (LQEs) exploited to distinguish between high, medium and bad quality radio links. Link quality estimation is utmost important in improving higher layer protocols efficiency. Indeed, considering the routing case; tracking, identifying and utilizing links with highest quality for packet forwarding increases network performance in terms of energy saving and successful packet delivery increasing.
To elaborate on this issue, this chapter presents a brief overview on WSNs, followed by a description of WSNs radio links characteristics. Then, examples of link quality estimations applicabilities jointly with a review of most representative LQEs in literature are given.

2.2 Wireless Sensor Networks (WSNs)

2.2.1 Overview

Wireless Sensor Networks (WSNs) are self-configured wireless networks composed of a set of cooperative, small and battery powered devices equipped with sensing

and communication capabilities. These devices (referred to sensor devices or sensor nodes or motes) are intended to collect measurements from their environment of deployment. Being often massively deployed, they are designed with severe constrained and low cost resources.
WSNs invention has been an immediate result of the conjunction of advances in microelectronics and wireless communication technologies, coupled with the need to easily envision of a wide range of real world applications [16, 39]. With the asset of its unprecedented flexibility and low cost, we are witnessing today, a growing integration of this new promising technology in several monitoring and tracking interest domains such as health care, agriculture, security, military surveillance, home automation and environment monitoring.
According to [44] and as depicted in Figure 2.1, a typical architecture of a WSN consists of four basic components: (*i.*) an assembly of remotely distributed or localized sensors (*ii.*) an interconnecting network (usually, but not always, wireless-based) (*iii.*) a central point of information clustering said Clustering Node (referred also as sink node) and (*iv.*) a set of computing resources at the central point (or beyond, i.e: at a Final Processing Node) to handle data correlation, event trending, status querying, and data mining.

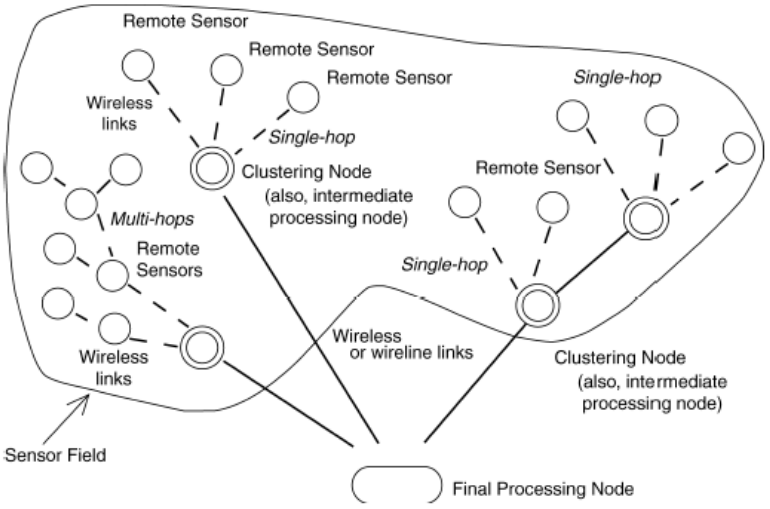


Figure 2.1: Typical architecture of a WSN [44]

The main objective of WSN deployment (i.e measurements collection) in itself seems to be non-complex; however the ensuring of its correct operation wraps serious challenges. Indeed, in one hand, due to the increasing demand of cheap and small size devices, sensor nodes are designed with limited resources in terms of energy, memory, computation and transmission capabilities. Yet, the performance requirements are limitedly relaxed. On the other hand, sensor nodes have to perform several activities including sensing, computing and communication. They also have to strive to operate for a long period. Hence, the key challenge is how we could ensure correct operating and long-lived network in spite of resources scarcity impeding? The more

works in the domain try to find a compromise answer to this question by developing advanced hardware platforms and advanced software solutions, the more new research topics raise and the more considerable efforts have to be devoted. Hardware advances interest in optimizing and improving internal architecture of sensor nodes and in reducing their costs [55]. Software advances interest in addressing issues such as energy saving, network lifetime increasing, security ensuring, connectivity maintain, mobility managing etc by developing sophisticated protocols, softwares and standards.
An overview about these advances is presented in what follows.

2.2.2 Hardware Advances

Being thrown in the target environment, an operating sensor device has to support four basic functionalities:

1. Data sensing from the surrounding environment
2. Data storing
3. Data processing
4. Communicating with other sensor devices in the network

Ensuring these functionalities requires the availability of five basic hardware components when designing a sensor device as illustrated in the next Figure.

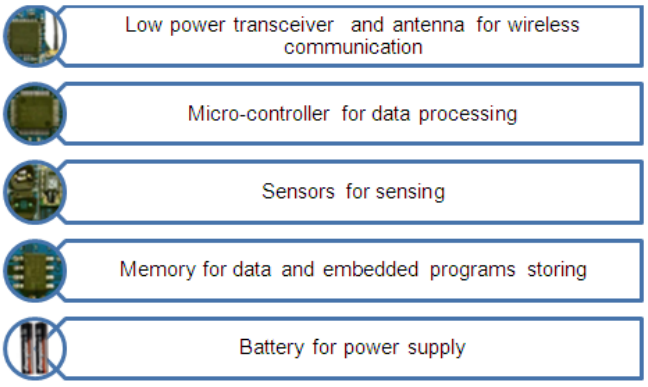


Figure 2.2: Sensor device: Hardware components

The design of these hardware components has witnessed seminal advances in order to tackle energy saving problem and in order to improve network efficiency. Consider Crossbow [1] products as typical examples; we can see, by skimming the chronicle of manufactured sensor devices, a clear reduction in active and sleep power with an improvement in data rate, memory sizes and applications support. Furthermore, recent platforms are designed to be IEEE 802.15.4 standard compliant which

is a standard designed for low cost, low power networking and Low Rate Wireless Personal Area Networks (LRWPANs). Figure 2.3 outlines this chronicle.

Mote Type	*René2 2000*	*Mica 2001*	*Mica2 2002*	*MicaZ 2004*	*Telosb 2004*	*Imote2 2007*
						
Micro-Controller						
Type	ATMegaL 163	ATMegaL 128			IT MSP430	Intel PXA271 XScale
Program memory (KB)	16	128			48	32768
RAM (KB)	1	4			10	256
Current Draw (Active mode)	15 (mW)	5.5 mA	8 mA		1.8 mA	31 mA
Current Draw (Sleep mode)	45(µW)	<20(µA)	<15(µA)		5.1 µA	390 µA
Wakeup Time (µs)	36	180			6	
Communication						
Radio	TR1000		CC1000	CC2420		
Data rate (kbps)	10	40	38,4	250		
Modulation scheme	OOK	ASK	FSK	O-QPSK		
Programming and sensor interface						
Communication	IEEE 1284 (programming) and RS232 (requires additional hardware)				USB	
Applications						
		• Security, Surveillance and Force Protection • Environmental Monitoring • Large Scale Wireless Networks • Distributed Computing Platform		• Indoor Building Monitoring and Security • Acoustic, Video, Vibration and Other High Speed Sensor Data • Large Scale	• Platform for low power research development • Wireless sensor network experimentation	• Digital Image Processing • Condition Based Maintenance • Industrial Monitoring and Analysis • Seismic and Vibration Monitoring

Figure 2.3: The chronicle Of Crossbow manufactured sensor devices

According to [41], the power of WSN technology can be fully harnessed only if proper software solutions are made available to application developers in addition

to hardware advances.

2.2.3 Software Advances

A WSN can be viewed in two levels [42]. The first one is the network level representing mainly the wireless communication between sensor nodes. The second is the sensor node level consisting of hardware, radio, CPU and embedded softwares. For the first level, the presence of communication standards is necessary in order to ensure the interoperability between sensor nodes of different manufacturers. While for the second level, the presence of operating systems is required to provide common services typically are (*i.*) hardware management of sensors, radios, CPU, and I/O buses, (*ii.*) tasks coordination, (*iii.*) power management, (*iv.*) adaptation to resource constraints, and (*v.*) networking [17]. Non exhaustive descriptions of WSN related Operating Systems (OSs) and communication standards are presented in what follows.

2.2.3.1 Operating Systems

Operating Systems (OS) designed for Wireless Sensor Network (WSN) differ greatly from traditional ones, especially because of resources and energy constraints of sensor devices. Several operating systems have been designed with respect to these particular characteristics such as "TinyOS", "Mantis", "Contiki" and "Emstar". The following section gives a brief snapshot about these OSs.

2.2.3.1.1 TinyOS TinyOS is an event driven execution model and component-oriented architecture operating system developed by Berkeley EECS Department. It provides a rich library of components including hardware abstractions (chipcon, sensor, memory, microcontroller, I/O buses, etc), resources management facilities (resource arbitration, power management) and communication protocols (MAC, routing, etc). Developing applications under TinyOS requires being familiarized with NesC which is C like programming language. Currently, TinyOS is widely used by over 500 research groups and companies. It can be considered as the leader among the other available operating systems designed for sensor networks. Its well reputation is justified by the following provided options:

1. Accessibility for free and open source code downloads.
2. Ease of installation and use.
3. Availability of large amount of documentations and online tutorials.
4. The support of broad motes platforms especially those manufactured by Crossbow company

According to [43], TinyOS suffers from two major disadvantages: the non-preemption due to its event driven execution model and the non-memory protection as no virtual memory concept or memory management exists in it.

2.2.3.1.2 Mantis Mantis (multimodal system for networks of in situ wireless sensors) is a preemptive multithreaded embedded operating system for wireless sensor networks. Mantis multithreading enables sensor nodes to natively interleave complex tasks with time sensitive tasks [28]. Mantis applications are written in C and executed as threads. The Network stack and the scheduler are also implemented as threads [43]. Threads maintaining is managed by the kernel. In fact, the kernel maintains a thread table specifying thread priority, thread handler and other thread related information. According to [43], Mantis suffers from the overheads of context switching and the memory allocated per each thread. Such overhead is discouraged for resource constrained systems like WSN.

2.2.3.1.3 Emstar EmStar is not exactly an operating system but a programming model and software framework for creating Linux-based sensor network applications. Emstar enables unmodified TinyOS applications execution within the EmStar environment through its wrapper layer EmTOS. EmStar provides also two hybrid modes that combine simulation with real wireless communication and sensors in the environment. Each of these modes run the same code and use the same configuration files, allowing developers to seamlessly iterate between the convenience of simulation and the reality afforded by physically situated devices [25].

2.2.3.1.4 Contiki Contiki is an event driven execution model operating system supporting multithreading as an optional application level library [43]. The kernel of Contiki supports two types of events: synchronous and asynchronous. Synchronous events require the immediate running of the event handler and are required to run to completion [40]. Asynchronous events, in contrast, could be dispatched at a later time. In Contiki, communication protocol stack, device drivers, sensors data handling are implemented as service. Each service has an interface indicating its id and an implementation. Since services are replaceable [40], service implementation can be changed at run time.

2.2.3.1.5 Discussion The previously presented OSs belong mainly to two opposite models: event-driven and thread-driven models. Each model has its unique characteristics and its proper pros and cons. The following tables outline the advantages and the disadvantages of each model based on the work published in [40].

Table 2.1: Advantages and disadvantages of the event-driven model

Advantages	Disadvantages
Concurrency with low resources	Event-loop is in control
Complements the way networking protocols work	Program needs to be chopped to subprograms
Inexpensive scheduling technique	Bounded buffer producer-consumer problem
Highly portable	High learning curve

Table 2.2: Advantages and disadvantages of the thread-driven model

Advantages	Disadvantages
Eliminates bounded buffer problem	Complex shared memory
Programmer in control of program	Expensive context switches
Automatic scheduling	Complex stack analyses
Real-time performance	High memory footprint
Low learning curve	Not portable due to stack manipulation
Simulates parallel execution	Performs better on multiprocessors

2.2.3.2 Communication Standards

In order to achieve cost reduction through mass production [5] and to ensure the interoperability between devices of different manufacturers, a number of communication standards have been established. A brief overview of the most well-known ones are presented next.

2.2.3.2.1 Zigbee Zigbee is a standardized network protocol stack built on top of the IEEE 802.15.4 network specification. In fact, the two lowest layers: Physical and Medium Access Control layers are provided by the IEEE 802.15.4, while ZigBee defines two other layers: the Application Layer (APL) and the Network Layer (NWL).
The physical layer uses Sequence Spread Spectrum (SSS) modulations schemes and operates on one of three possible unlicensed frequency bands :

- 868.0-868.6 MHz: Europe, allows one communication channel (2003, 2006)
- 902-928 MHz: North America, up to ten channels (2003), extended to thirty (2006)
- 2400-2483.5 MHz: worldwide use, up to sixteen channels (2003, 2006)

The MAC controls the access to the medium using the CSMA-CA mechanism according to two different modes: beacon enabled and non-beacon enabled modes. In the non-beacon enabled mode all nodes in the network can transmit their packets whenever the channel is Idle. In contrast, in the beacon enabled mode, nodes in the network can only transmit their packets in predefined time slots. The time slots assignment is defined in a superframe, periodically sent by a coordinator node. The superframe specifies for each node the specific time slot during which it can transmit and receive its packets. The superframe may also contain a common slot during which all nodes in the network compete by using CSMA-CA mechanism to access the channel.
The goal of the Network Layer is to manage the routing policy, the security procedures and the establishment of the topology and its control.
Finally, the Application Layer is responsible of the forwarding of messages between devices and the maintaining of binding tables. These binding tables match devices

according to their services and their needs.
Zigbee classifies devices according to their functionalities on three types:

- *Zigbee Coordinator (ZC)*: It is the most sophisticated device among others as it is the responsible of the initiation of the network, the maintaining of its overall information and coordinates the bridging with other networks.
- *Zigbee Router (ZR)*: It is an intermediate device working as a relay between ZED devices to ensure their distributed multi-hop communication.
- *Zigbee End Device (ZED)*: it works an ordinary sensor node that collects measurement from the surrounding environment and that communicates with its parent (either a coordinator or a router).

2.2.3.2.2 Bluetooth It is a short range wireless communication standard operating in the unlicensed 2.4-GHz frequency band. Bluetooth is invented by telecoms vendor Ericsson in 1994.
Bluetooth connects devices together in a star-shaped topology (a piconet) and applies master and slave concept. In fact, Bluetooth devices have to wait until the master of the piconet allows them to communicate [24]. Communication time in Bluetooth is divided into slots. The odd-numbered slots are reserved for the master device, while the even-numbered ones are reserved to slave devices. Slave devices can only transmit in response to a query from the master device. Further, even devices have data to transmit, they must wait until the master device authorize them [24].
The major drawback of Bluetooth is that the maximum number of simultaneously interconnected devices is limited : Only one master and up to seven slave devices per piconet.

2.2.3.2.3 WirelessHart Highway Addressable Remote Transducer (HART) is a wireless communications standard suitable for industrial applications such as process control, measurement and management applications. The main advantages of this standard are its reliability, security, compatibility with existing devices and more importantly its energy efficiency. The reliability is traduced by its capability of coexisting with other wireless networks in the vicinity (immunity against interference) thanks to adopted modulation scheme (channel hopping) and the usage of time-synchronized messaging. The security is ensured through the usage of encryption, authentication, key management, and other open industry-standard security practices [6]. And the energy efficiency is guaranteed by the Smart Data Publishing and other techniques that make batteries, solar and other low-power options practical for wireless devices [6].
A WirelessHART network consists of three basic components:

- *Wireless field devices*: they are connected to process or plant equipments.
- *Gateways*: they enable the communication between the wireless field devices and the host applications connected to a high-speed backbone or other existing plant communications network.

- *Network manager*: it configures the network and schedule communication between devices. It also manages the routing and network traffic. It can be integrated into the gateway, host application, or process automation controller.

2.2.3.3 Discussion

Existing operating systems and communication standards are two facilities given to application developers in order to ease the building of optimized, energy efficient and inter-operable WSN software solutions. Nevertheless, application developers still have to deal with challenges related to WSNs when designing new applications, algorithms and network protocols. Examples of these challenges are:

- **Insecurity:** Sensor nodes are usually thrown in hostile environments. Numerous kinds of attacks can thereby threaten the confidentiality, the integrity and the operation of the network.
- **Frequent topology changes:** Sensor nodes mobility and their batteries drain lead to frequent changes in the network topology requiring thus an instantaneous adaptation at communication protocols level.
- **Erratic quality communication links:** While propagating though the wireless medium, transmitted signals are subject of different phenomena leading to their distortion. This makes the quality of communication links extremely erratic. A need to carefully track the quality of these communication links becomes necessary. Link quality estimation emerges as an important mechanism to determine how good communication links are. We particularly interest in this Master thesis to Link Quality Estimations. The following section elaborates more on this mechanism.

2.3 Link Quality Estimation

In wireless communication, interference, multi-path effects, harsh environment conditions have a huge impact on the correct signal propagation. Interference is caused when simultaneous transmissions are done by near operating wireless networks sharing the same transmission channel. Multi-path effects consist in signals absorption, attenuation, reflection and scattering when hurting obstacles, smooth and rough surfaces. More the signal strength is strong more it will be resilient against these communication hurdles. However, in Wireless Sensor Networks and because of stringent cost and energy constraints, sensor nodes are hampered by a low-power radio transceiver which emits very low strength signals. Therefore, while propagating though the wireless medium, these signals are weak enough to be resilient against unwanted modifications. Consequently and comparing to other wireless networks, the quality of radio links found in Wireless Sensor Networks experiences erratic variation over time and space making the communication over them very unreliable. Numerous studies in the field [27, 18, 53, 34, 56, 57, 59, 50, 60, 33, 8, 23] have been conducted in order to understand radio links unreliability by grasping their particular characteristics. In these studies, several experiments have been performed in

different environment of deployment and under different network settings. All these studies agreed on several observations, which are presented in what follows.

2.3.1 WSNs Radio Links Characteristics

2.3.1.1 Non-isotropic connectivity

Figure 2.4 shows that the Received Signal Strength Indicator (RSSI) varies continuously when incrementally changing of the propagation direction from the sender [57]. This variability leads to non-isotropic Packet Reception Ratio (PRR) which means that the PRR does not have the same value in all directions from the source as depicted in Figure 2.5.

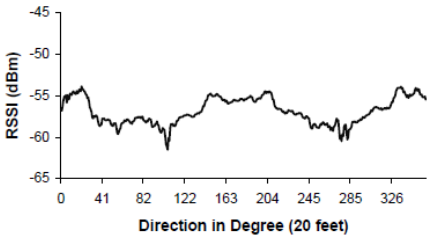


Figure 2.4: Non-isotropy in Received Signal Strength Indicator (RSSI) [57]

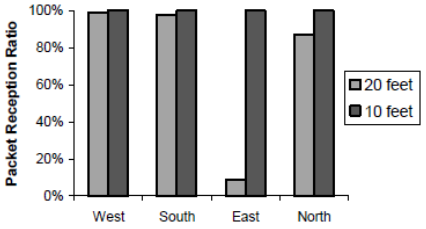


Figure 2.5: Non-isotropy in Packet Reception Ratio (PRR) [57]

2.3.1.2 Existence Of a Gray Area

Empirical studies [54, 56] have shown the existence of a three reception regions [Figure 2.6]: Effective region (called also Connected Region), clear region (called also disconnected region) and transitional region (called also gray area). The connected region is characterized by high Packet Reception Ratio (indicated as Reception Success Rate in the next Figure) stable and symmetric links [60]. The extents of this

effective region increase with transmit power [54]. The transitional region is characterized by the presence of unreliable and asymmetric links [60]. Finally the disconnected region is characterized by the absence of practical links for transmission [60].

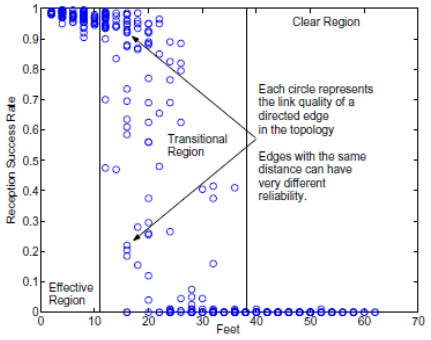


Figure 2.6: Reception regions [54]

2.3.1.3 Non Monotonically Distance Decay

As it may be observed from the previous Figure, the Packet Reception Ratio (PRR) (indicated as "reception success rate" in the Figure) of links within the gray area does not monotonically decay with distance. This means that nodes that are geographically far away from the source may get better PRR than nodes that are geographically closer.

2.3.1.4 Temporal variability:

Tracking the PRR as function of time shows its continuous variation. This variation is mainly observed when links are situated in the gray area. Figure 2.7 highlights this behavior.

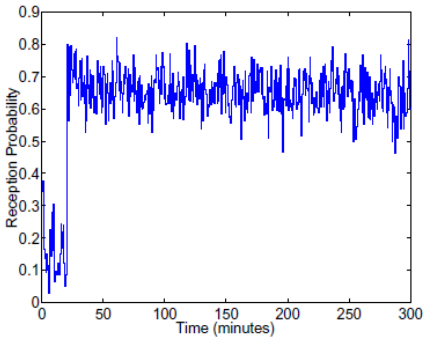


Figure 2.7: PRR temporal variability [54]

2.3.1.5 Asymmetry

Asymmetry is quantified by the difference between Packet Reception Ratio (PRR) for a link pair between i and j [56] as mentioned bellow:

$$|PRRi \rightarrow j - PRRj \rightarrow i| \tag{2.1}$$

More this difference is great more the link is considered as highly asymmetric. [27] and [60] argue that hardware variance as well as differences between sensor node in transmit power and reception sensitivity are identified as the cause of link asymmetry. Further [60] argues that link asymmetry play a significant role on the extent of the transitional region.

2.3.1.6 Discussion

All of these findings have questioned the validity of the idealized binary model which assumes a disc-shape reception region where packets are received only within a certain distance separating the sender and the receiver nodes[60]. The binary model assumption is definitely unrealistic for two major reasons: (1) two sensor nodes placed at the same distance to the sender may have extremely different packet reception ratio (e.g. 10% and 90%) (2) there is a non-zero probability of receiving a packet at a long distance from the sender [27].
Further, several studies [27, 33, 58] have shown that these characteristics can have a major impact on the performance of higher layer protocols [60]. Authors of [58] demonstrate that links unreliability can have a negative impact on routing protocols, particularly geographic forwarding schemes [60]. In [27], authors show that flooding mechanisms for data dissemination can be significantly affected by link asymmetry even by using the simplest one. De Couto et al have also deduced that the traditional routing metric "Shortest path" works poorly in WSN due to links unreliability as minimizing the hop-count maximizes the distance traveled by each hop, which is likely to minimize signal strength and thus minimize the Packet Reception Ratio [21]. We conclude that links unreliability becomes a challenge for higher layer protocols designer. Many works have been investigated in addressing this challenge by developing mechanisms called "Link Quality Estimators" (LQEs) exploited to distinguish between links with high, medium and bad quality. Link quality estimation is utmost important in improving higher layer protocols efficiency. The following section sheds the light on examples of Link Quality Estimation applicabilities.

2.3.2 Link Quality Estimation Usefulness

Link quality estimation can be applied in various contexts, such as routing, rate adaptation and mobility managing [52].

- **Routing:** Traditional routing protocols rely on hop-count information to route data. By consequence, they select links with arbitrary quality. However, routing over good quality links instead of over arbitrary ones, increases network performance in terms of energy saving and successful packet delivery increasing.

In fact, transmitting over good quality links reduce the number of transmissions required to successfully deliver a packet to the next-hop, which in effect reduces energy consumption. Moreover, link quality estimation enhances the selection of the most stable routes for communication [10]. Stable routes are built by selecting good quality links and discarding bad quality ones which in effect increases the end-to-end probability of message delivery and minimizes the route re-selection operation triggered by links failure [10].

- **Rate adaptation:** Rate adaptation is referred to the mechanism of selecting one out of multiple available transmission rates at a given time [22]. The effectiveness of a link adaptation scheme depends on how fast it can respond to the wireless channel variation [22]. Thus, the transmitter can select the adequate transmission data rate that with being sustained by the current link quality information [52].
- **Mobility managing:** Link quality estimate can be used to determine when to hand over a mobile user to another cell or sector, or when to deploy relays to create in real time a multi-hop network [46]. Relays are created in real time to extend the communication range through multihop relaying when the range of single-hop wireless communication is limited by distance or harsh radio propagation conditions [46].

In this context, several Link Quality Estimators (LQEs) have been designed. The next section reviews the most reported LQEs in literature.

2.3.3 Examples Of Link Quality Estimators (LQEs)

Link quality estimators are link quality measurement metrics exploited to determine how good communication links are. Several Link Quality Estimators (LQEs) have been reported in the literature. These LQEs can be classified to three classes: receiver side estimators, sender side estimators and hybrid side estimators.

NB: Given a sensor node, we refer by "uplink", the link on which the sensor node sends its packets and by "downlink", the link on which the sensor node receive packets.

2.3.3.1 Receiver side estimators

Receiver side estimators are computed at the receiver sensor node based on the received traffic and estimate the downlink quality. We name:

2.3.3.1.1 RSSI Acronym of Received Signal Strength Indicator, RSSI is read form an 8 bit register in the radio chipcon. Consider for example CC2420 radio chipcon [7], RSSI acquisition start by sampling the signal strength over the first 8 symbols following the Start of Frame Delimiter (SFD) of the received packet. Then CC2420 averages the measured values and stores the computed value in the register

RSSI_VAL. Based on the work of [48], RSSI can provide a quick estimate of whether an downlink link is in the gray region or not.

2.3.3.1.2 LQI Acronym of Link Quality Indication, LQI is also a hardware metric provided by the CC2420 radio chipcon. As RSSI, LQI is an average correlation value of samples computed over the first 8 symbols following the Start of Frame Delimiter (SFD) of the received packet. Based on the work of [48], LQI can provide an estimate of where in the gray region a link is.

2.3.3.1.3 SNR Acronym of Signal to Noise Ratio, SNR is a measure quantifying how much a signal has been corrupted by noise. SNR expressed in decibel (dB), is the ratio of signal power to the noise power corrupting the signal. SNR is a hardware metric reflecting roughly the channel quality in terms of noise presence.

2.3.3.1.4 PRR Acronym of Packet Reception Ratio, PRR is defined as the number of successfully received packets over the number of sent packets. The number of sent packets is the sum of received and lost packets. The receiver infers the losses in packet reception by tracking the sequence numbers and counting gaps between them. PRR estimates are computed for each window of "w" received packets, as follows:

$$PRR(w) = \frac{Number\ of\ received\ packets}{Number\ of\ sent\ packets} \tag{2.2}$$

2.3.3.1.5 WMEWMA Acronym of Window Mean Exponential Weighted Moving Average, WMEWMA uses a Exponential Weighted Moving Average (EWMA) filter to combine recently and previously PRR computed estimates. We say that WMEWMA smooths PRR estimates in order to provide transient fluctuations resistant metric. EWMA filter is a function that gives greater/lower weight to more recent measurements and exponentially decrease/increase weight of older ones. The rate of decrease is governed by a smoothing factor α which ranges between 0 and 1. WMEWMA can is calculated as follows:

$$WMEWMA(w) = \alpha * WMEWMA + (1 - \alpha) * PRR \tag{2.3}$$

2.3.3.1.6 ETX Acronym of Expected Transmission count, ETX approximates the required number of retransmissions to successfully deliver a packet. ETX takes into account link asymmetry by estimating the uplink quality from the sender to the receiver, denoted as PRRup, as well as the downlink quality from the receiver to the sender, denoted as PRRdown [10]. By combining PRRup and PRRdown, ETX provides an estimation of the bidirectional link quality, expressed as follows:

$$ETX(w) = \frac{1}{PRRdown * PRRdup} \tag{2.4}$$

2.3.3.1.7 F-LQE Acronym of Fuzzy Link Quality Estimator, FLQE, a recently proposed estimator, combines four link quality properties namely, packet delivery (SPRR), ASymmetry Level (ASL), Stability Factor(SF), and channel quality quantified by SNR (ASNR). SPRR defines the capacity of the link to successfully deliver data, ASL is represents the difference in connectivity between the uplink and the downlink, SF quantifies the variability level of the link, and the ASNR reflects the degree of noise in the communication channel. Each of these proprieties is defined as a fuzzy variable. The overall quality of the link is traduced in a fuzzy rule:

IF the link has high packet delivery AND low asymmetry AND high stability AND high channel quality THEN it has high quality

The above rule is translated to the following equation:

$$\begin{aligned}\mu(i) = \beta * min(\mu_{SPRR}(i), \mu_{ASL}(i), \mu_{SF}(i), \mu_{ASNR}(i)) + \\ (1-\beta) * mean(\mu_{SPRR}(i), \mu_{ASL}(i), \mu_{SF}(i), \mu_{ASNR}(i))\end{aligned} \tag{2.5}$$

where: μ_{SPRR}, μ_{ASL}, μ_{SF} and μ_{ASNR} are the membership functions of the fuzzy variables. The next Figure depicts their definitions.

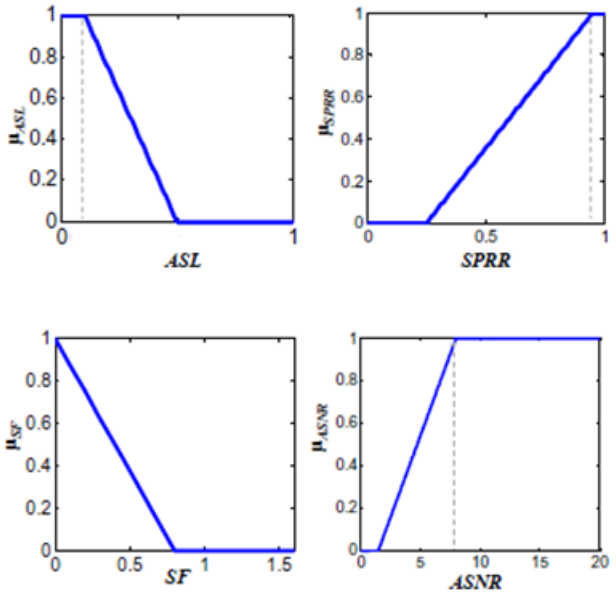


Figure 2.8: F-LQE membership functions [11]

Finally, F-LQE uses a EWMA filter to smooth its estimates as follow:

$$FLQE(w) = \alpha * FLQE + (1-\alpha) * 100 * \mu(i) \tag{2.6}$$

2.3.3.2 Sender side estimators

Sender side estimators are computed at the sender sensor node based on the sent traffic and estimate the uplink quality. We name:

2.3.3.2.1 RNP Acronym of Requested Number of Packets, RNP counts the required number of packet transmissionsretransmissions (txrtx) before a successful reception. This metric is computed for each w transmitted (tx) and re-transmitted (rtx) packets as follows:

$$RNP(w) = \frac{Number\ of\ tx\ and\ rtx}{Number\ of\ successfully\ sent\ packets} \tag{2.7}$$

The number of successfully sent packets is determined by the sender as the number of acknowledged packets.

2.3.3.3 Hybrid side estimators

Hybrid side estimators are computed at the sender sensor node based on both sent and received traffics. We name for example:

2.3.3.3.1 Fourbit Fourbit estimator is an estimator computed at the sender sensor node level and approximates the packet retransmissions count based on statistics collected from received and sent packets. In fact based on "w1" received packets, the node computes the WMEWMA estimate and derives an approximation of the RNP, denoted as estETXdown, as follows:

$$estETXdown(w) = \frac{1}{WMEWMA} - 1 \tag{2.8}$$

Further, the sender computes RNP, denoted as estETXup, based on "w2" transmitted/retransmitted data packets to the receiver. Finally, Fourbit combines both estETXup and estETXdown metrics via the EWMA filter, in order to obtain an estimate of the bidirectional link expressed as follows:

$$Fourbit(\alpha, w) = \alpha * Fourbit + (1 - \alpha) * estETX \tag{2.9}$$

Where estETX corresponds to estETXup or estETXdown.

2.3.3.4 Discussion

In [10], it has been argued that although RSSI, LQI and SNR have the advantage of not requiring additional computation overhead, they are judged as not sufficient to characterize the holistic link quality as they are measured uniquely based the first 8 symbols of a received packet and not the whole packet. RSSI, LQI and SNR are considered as **Hardware estimators** [10] as they are directly extracted from the radio chipcon. Other estimators are considered as **Software estimators**. Software estimators enable to count or approximate either the reception ratio (PRR) or the average number of packet transmissions/retransmissions (RNP) before its successful reception [10]. Hence, they can further be classified as PRR−counting estimators and RNP−counting estimators [10]
The following tables outlines the characteristics of each mentioned estimator.

Table 2.3: Characteristics of different LQEs

	Software/Hardware	Location	Direction
RSSI	Hardware	Receiver	Unidirectional
LQI	Hardware	Receiver	Unidirectional
SNR	Hardware	Receiver	Unidirectional
PRR	Software	Receiver	Unidirectional
WMEWMA	Software	Receiver	Unidirectional
ETX	Software	Receiver	Bidirectional
FLQE	Hybrid	Receiver	Bidirectional
RNP	Software	Sender	Unidirectional
Fourbit	Software	Hybrid	Bidirectional

2.4 Conclusion

In this chapter we have given an overview of Wireless Sensor Network (WSN) technology and highlighted its radio link irregularities. Link quality estimation emerges as an important mechanism for higher layer protocols to overcome this pitfall. In this regard, several Link Quality Estimators (LQEs) have been proposed. Link Quality Estimators are either receiver side, sender side or hybrid side estimators. Link quality estimation is utmost important in improving higher layer protocols efficiency. However, ensuring such performance is closely related to the accuracy of link quality estimates. Indeed, poor link quality estimates can cause a 200% or greater slowdown in network throughput [21]. Thus, evaluating the performance of Link Quality Estimators (LQEs) remains mandatory.
Prior works [10, 11] have studied the performance of PRR, WMEWMA, ETX, RNP, Fourbit and FLQE using TOSSIM2 simulator. In [10], the authors have first analyzed the statistical properties of (PRR, WMEWMA, ETX, RNP, Fourbit) independently of higher-layer protocols. Then, they investigated their impact on the Collection Tree Routing protocol (CTP) [2]. In [11], the authors extended the first evaluation methodology presented in [10] to validate their new proposed estimator (FLQE). However, the validation of FLQE might not be conclusive as it is based on simulation. Furthermore, the impact of FLQE on higher layer protocols has not been investigated. Therefore, it is important to evaluate the performance of different LQEs based on real experimentation, in order to get more realistic and trust-able observations.
The next part of this Thesis describes in depth our experimental study by firstly introducing our benchmarking tool exploited to conduct the real experimentations and then by presenting LQEs statistical properties and their impact on routing.

Part II

Experimental Study

Chapter 3

RadiaLE: A Framework For Benchmarking LQEs In WSNs

3.1 Introduction

Simulation and real experimentation are two fundamental techniques that are widely used in the research community to conduct experimentations such as experiments for a performance evaluation study. Simulation provides a rapid prototyping solution but it lacks accuracy as it usually relies on simplified and stochastic models. In contrast, real experimentation provides much more accuracy and fidelity. Since the performance evaluation of LQEs highly depends on real network conditions, conducting such study trough simulation shows limited extent and depends on the accuracy of the simulator. For that reason, it has been proposed, in order to replicate the study presented in [11], to use or to build a benchmarking tool commonly called testbed that automates the performance evaluation of different LQEs. This testbed will be firstly used to analyze and understand the statistical properties of link quality estimators independently of any external factor, such as collisions and routing. Only the impact of the physical layer and the data link layer (retransmissions) are considered. Towards this goal, the present chapter starts by surveying the state of the art of existing testbeds designed for the experimentation of WSNs and then describes our implemented RadiaLE testbed that fit more with our requirements.

3.2 Experimental Testbeds For WSNs: Literature Review

Testbed designed for the experimentation of WSNs, consists, generally, of hardware and software components 3.1. The hardware components encompass sensor nodes, programming boards, computers and connectors (Ethernet connection, USB connection, secondary wireless connection etc). In contrast, software components, encompass usually:

- User interfaces (scripts or a GUI) for the interaction with the deployed sensor network.

- A database for permanent data storage. This data are gathered during experiments' running.
- A serial forwarded enabling the communication between sensor nodes and user interfaces.
- An embedded operating system enabling sensor nodes programming.

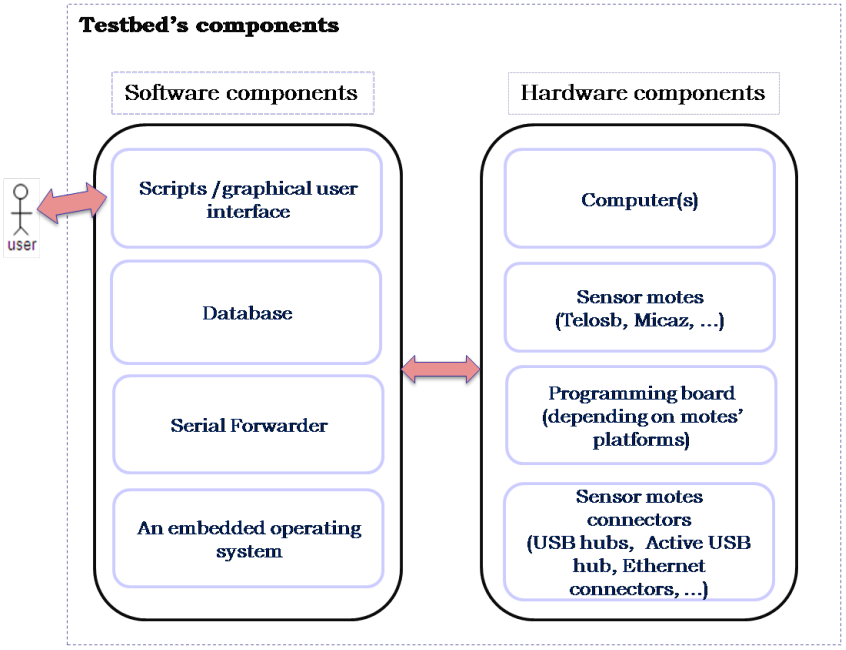


Figure 3.1: Testbed's components

Many testbeds have been designed for the experimentation of WSNs. They can be classified to two classes: **fixed topology based** testbeds and **flexible topology based** testbeds. Testbeds belonging to the first class have been designed to offer open access to a permanently pre-deployed wireless sensor network where the user can change neither its topology nor the environment where it is deployed. In contrast, testbeds belonging to the second category have been designed to give users ability to define their own topology and environment of deployment.

3.2.1 Fixed Topology Based Testbeds

3.2.1.1 MoteLab

MoteLab [29] is a remotely accessible testbed developed by Harvard University. Through, its web interface, MoteLab enables for registered users to interact with available sensor nodes by installing Tinyos binary images, scheduling and running experiments. Sensor nodes consist of 190 TMote Sky motes, deployed in an indoor environment, in a fixed fashion over 3 floors of the "Electrical Engineering and Computer Science" building at Harvard University. Each mote is powered from wall

power and is connected to the Ethernet network of the building. The advantage of being connected to the Ethernet network is providing a separate control network over which it facilitates the direct capture of debugged data sent over serial ports and the quick sensor nodes reprogramming.
Motelab software architecture consists of the four following components as depicted in Figure 3.2:

- A web interface
- job daemon.
- DB logger
- And a SQL database server

The web interface: it is the front-end of subscribed users to create experiments, called "jobs" in Motelab jargon, by firstly uploading the executable files: binary images obtained from TinyOS operating system, and class files defining the structure of messages sent over serial ports. Secondly by selecting involved job sensor nodes count and ids, third by assigning programs to motes and optionally enabling 250 Hz power data collection on Mote 118. And finally, by setting job execution starting time and duration which is limited by the authorized user quota.

The job daemon: it is the most important component in Motelab architecture as it orchestrates jobs setting up including node reprogramming and jobs scheduling.

The DB logger: being connected to each node, DB logger responsibility consists in the reception of messages sent by sensor nodes over serial ports and their storage in a SQL database.

The SQL database server: it contains a set of databases storing two separated information: job generated data and testbed state. In fact, once subscribed, Motelab creates for the new user a database where his related experiments settings and debugged data will be stored. There is also another kind of database in this SQL server holding all testbed information, including users' details and attributed quotas; sensor nodes' states; information about uploaded files; job properties; and the testbed reservation schedule.

To our best knowledge, Motelab designers were the pioneer in providing public accessible and permanent available Testbed. May be also, the numerous useful provided features were the basic assets behind its well reputation and its wide usage in the research community. However, authors of [20] have questioned the efficiency of the scheduling mechanism that Motelab adopts. That's why they proposed a solution, called Mirage: A Microeconomic Resource Allocation System.

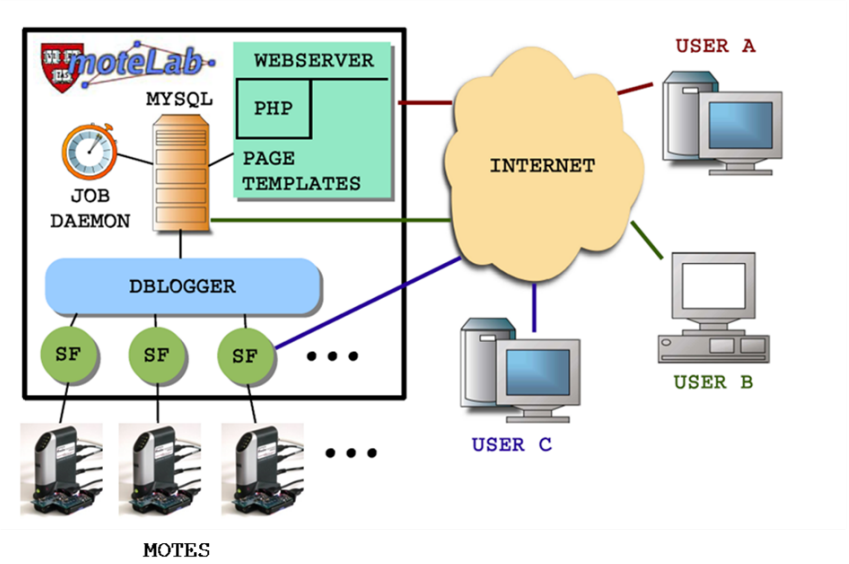


Figure 3.2: Motelab components [29]

3.2.1.2 Mirage

Despite its popularity, Mirage [20] is not a proper testbed. It is a resource allocation scheduling mechanism operating on Intel Research Berkeley's SensorNet testbed which hitherto deploys 100 MICAZ motes and 50 Mica2Dot motes. SensorNet's motes are connected to an out-of-band communication channel consisting in Ethernet network and are connected to power supply. As the same case with Motelab testbed, this channel enables users to remotely control, reprogram and retrieve data sent over serial ports independently of nodes' wireless communication channel. The resources allocation policy defined by Mirage, applies auction based mechanism to arbitrate the access to the testbed. Indeed, a registered user competes for testbed resources by submitting bids specifying resource combinations of interest in space/-time (e.g., "any x MICAZ motes for y hours anytime in the next z days") along with a maximum amount which he is willing to pay.
To use Mirage in order to obtain access to SensorNet testbed, a user is invited to do three steps:

1. Register for an account by either creating a new project or joining an existing one.

2. Bid for resources in the auction.

3. Claim a resource allocation after winning in the auction.

Register for an Account: Users, in Mirage are classified into two categories: project owners and project users. Each one must register first for an account. Then he has to create a project or to be associated to an available one. In order to create a project, a project owner must be enabled by Mirage administrators. However,

in order to be associated to an available one, project user must be enabled by the project owner. In doing so, passing through Mirage administrators' centralized bottleneck is removed.

Bidding in the Auction: For each project, Mirage attributes a bank account where virtual currencies are stored. Each bank account is initialized with a controlled baseline amount of currency. Such control will prevent users monopolizing the access to the testbed by bidding arbitrarily high values. This currency is used to bid in auction. Users submit bids to the auction using a two-phase process. First, they use the resource discovery service to find candidate nodes that meet their constraints. Second, they place bids in the auction using the Mirage bidding language[20]. Formally, a bid bi in Mirage is specified as follows:

$$b_i = (v_i, s_i, t_i, d_i, f_{min}, f_{max}, n_i, ok_i) \tag{3.1}$$

$[s_i\ t_i]$: specifies the interval where the experiment starting time should belong.
di: indicates the experiment duration.
$[f_{min}\ f_{max}]$: indicate that mote should operate on an unused frequency in the range $[f_{min}\ f_{max}]$, f_{min} and f_{max} are in MHZ.
n_i, ok_i: indicates that the user wants any combination of ni motes from the set ok_i. We note that ok_i is the set of motes obtained through resource discovery process.

Claim a resource allocation: For the winner bid, Mirage allocates a concrete set of motes for a specific period of time for all users on the winner user's project. After that and in order to communicate directly with the allocated motes, the user has to download motes' IP addresses list. Finally, he is invited to (1) login to mirage.berkeley.intel-research.net, (2) access to the motes using TCP/IP and (3) start his experiment.

Both Mirage (SensorNet) and Motelab testbeds, in their earlier versions, deploy ***non-USB motes platforms*** which require additional custom interfaces for their reprogramming. The emergence of standardized USB motes' platforms and the simplicity of their integration with external systems have motivated the authors of [51] to build the TWIST testbed that fully supports the USB 2.0 standard capabilities.

3.2.1.3 TWIST: TKN Wireless Indoor Sensor network Testbed

TWIST [51] is an indoor testbed designed by the "Technische Universität" of Berlin. Similarly to Motelab and Mirage (SensorNet), TWIST allows a public access to an experimental wireless sensor network encompassing 204 sensor nodes (102 TmoteSky nodes+102 eyesIFX nodes) spanned over three floors of the FT building at the TU Berlin campus.
TWIST provides basic functionalities to interact with the deployed network such node's configuration, network-wide programming, out-of-band extraction of debugged information and gatewaying of application data[51].
In this testbed, all the sensor nodes communicate with servers and a control station through special devices called super nodes 3.3.

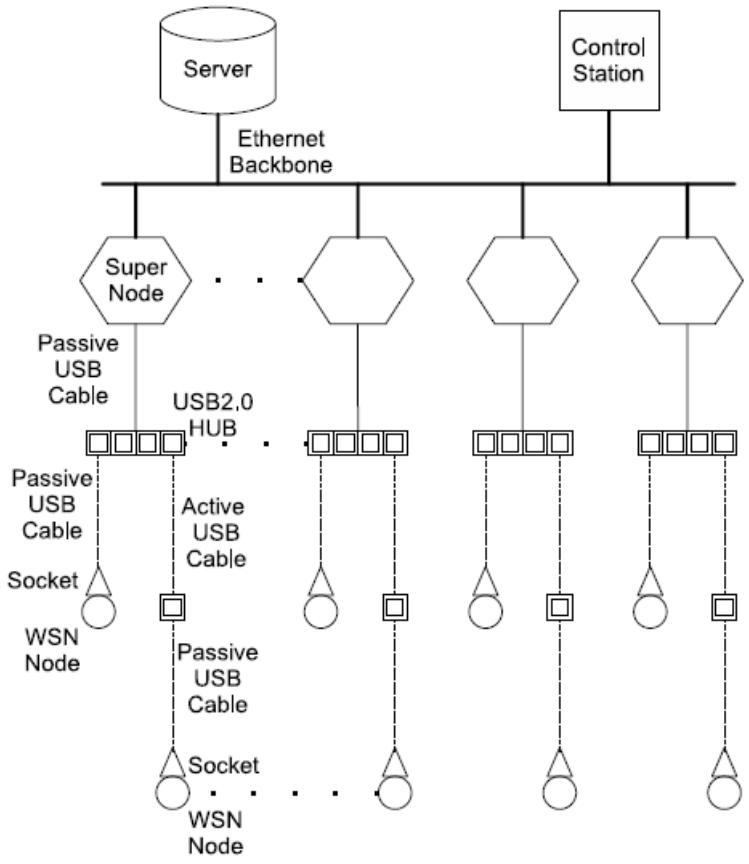


Figure 3.3: TWIST hardware architecture [51]

Servers maintain databases that store users' experimental data. Control station enables to configure launch and monitor experiments under the testbed. Finally, super-nodes, being connected to servers through Ethernet, allow programs, written under TinyOS operating system, installing on motes, power controlling and debugged data collection. Motelab, Mirage and TWIST use wired connections as back channel to access to the deployed wireless sensor network. In [31], it has been argued that such wired connections are cumbersome and do not allow large-scale deployment. Therefore, they designed DSN testbed, a flexible topology based testbed for the experimentation of WSN.

3.2.2 Flexible Topology Based Testbeds

3.2.2.1 DSN: Deployment Support Network

DSN is a non-permanent and flexible testbed developed by the ETH of Zurich. Through its innovative architecture, DSN provides a software and hardware solutions for scalable and quickly deployed WSN.
The key contribution of DSN testbed consists in building a wireless back channel

(backbone) between sensor network and data storage servers. Through this backbone, a user can:

- Retrieve data reported from deployed motes.
- Reprogram them.
- Control them by sending commands.
- Monitor their status by doing coordinated fault injection and profile their power consumption (battery voltage and current consumption).

The architecture of this backbone contains two building blocks as depicted in Figure 3.4:

- **DSN nodes:** consist in BTnode rev3 Bluetooth based nodes attached to sensor devices via a programming and debugging cables called wired target interface. DSN nodes form autonomous multi-hop Bluetooth based wireless network to report collected data sent from corresponding attached sensor nodes.
- **DSN server:** is connected to the Bluetooth network formed by DSN nodes and provides the client web based interface. Over this interface, the client can interact (communicate and control) and monitor the status of DSN nodes. The information flow goes from the client over the DSN-server to the DSN nodes and finally to the corresponding sensor nodes (i.e the sensor node directly attached to the DSN node) and vice versa. The DSNserver decouples the client from the target WSN both in time and space[31]. In particular, data from the sensor nodes are stored in a database and can be requested anytime, and commands can be scheduled on the DSN-nodes. Separation in space is given through the client interface that allows for an IP-based remote access[31].

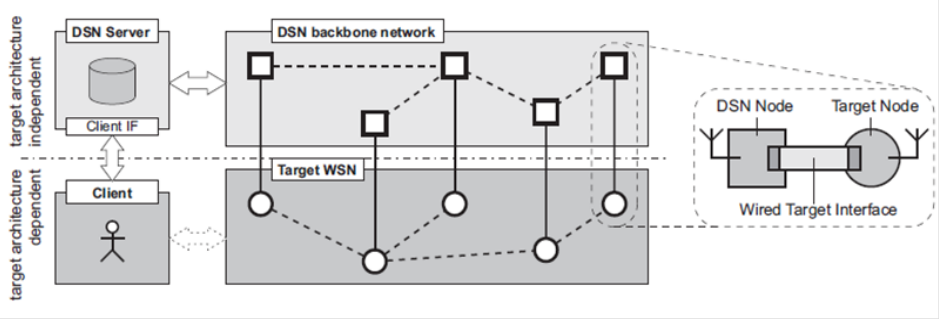


Figure 3.4: Architecture of DSN backbone [31]

In the context of his master project, Patrice Oehen has added to DSN testbed a tool for links measurements named: DSN Analyzer [Fig. 3.5].
DSN Analyzer is a java application used to control, monitor and reprogram Siemens A80 sensor nodes as well as to perform links measurements. Links measurements

are done through packet-statistics collection. Collecting a row of packet statistics is a preliminary step toward the computation of LQEs.
To our best knowledge, the first testbed designed to perform link measurements was SCALE: A tool for Simple Connectivity Assessment in Lossy Environments.

Figure 3.5: DSNAnalyzer [3]

3.2.2.2 SCALE: A tool for Simple Connectivity Assessment in Lossy Environments

Scale [18] is a "C" programmed tool compatible MICA1 and MICA2 motes, developed for the tuning and the assessing of connectivity across any deployed wireless sensor network. SCALE is integrated in the Emstar operating system and it focuses on facilitating the computation of Packet Reception Ratio using the same hardware platform and in the same environment targeted for deployment in the absence of concurrent transmission [18]. Each node in SCALE runs a software stack allowing for sending and receiving probe packets in a round robin fashion.
This software stack encompasses three modules:

- **Conntest:** it is charged of the sending and the receiving of probe packets and doing the control coordination among nodes (when to start/stop sending packet probes).
- **LinkStats:** it is charged of maintaining the packet reception statistics from all neighbors and the managing of the mode of communication between PC

and the motes. There are two mode of communication either serially or over UDP network. LinkStats provides respectively for each of one a corresponding driver: **MoteNic** and **Udpd**.

- **Emrun:** it orchestrates the dependency between modules according to the chosen configuration. For example, if a module terminates unexpectedly, emrun automatically restarts it and the other modules can reconnect to it without losing state.

SCALE provides also a visualization tool called Connview [Fig. 3.6] that offers two facilities.

1. The checking of the experiment status in real time.
2. The analysis and the display of the final experimental results.

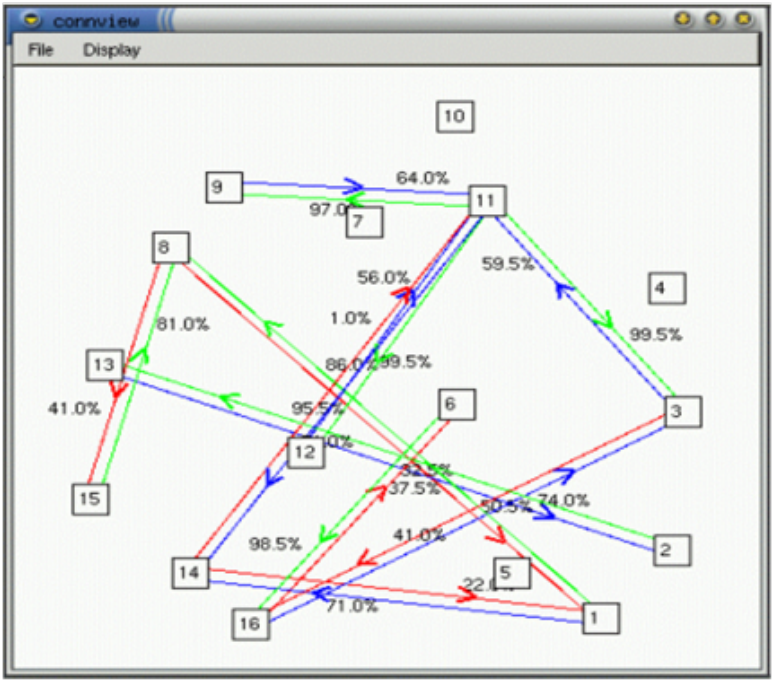


Figure 3.6: SCALE Connview interface [18]

Connview allows also the turning on/off of any node or any link, color links based on the percentages of packet reception ratio, display of asymmetric links, etc...

The compatibility with old platforms becomes a bottleneck in SCALE design, especially with the emergence and the increasing usage of new manufactured 802.15.4 compliant sensor nodes such Telob and MicaZ mote. SWAT the Stanford Wireless Analysis Tool overcomes this handicap.

3.2.2.3 SWAT: Stanford Wireless Analysis Tool

SWAT [47] is a software tool enabling link measurements over 802.15.4 and 802.11 networks [Fig. 3.7]. Through its software features, SWAT allows researchers to

- Configure their network through a HTML/PHP based User Interface (UI).
- Run their experiments
- Distill collected data into relevant plots.

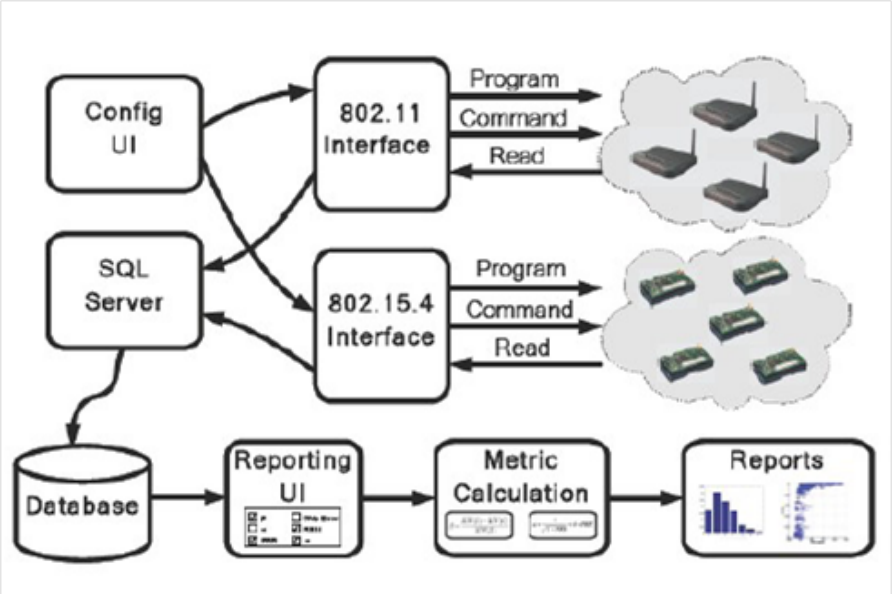


Figure 3.7: SWAT architecture [47]

Through the HTML UI, users can specify the experiment settings. They include network type (802.15.4/802.11), the list of node, the number of packets to send, the inter-packet interval, the type of transmission either in broadcast or in unicast, enabling or disabling CSMA backoff, the channel, the transmission power, enabling or disabling link layer acknowledgements, enabling or disabling link layer retransmissions (for unicasts transmissions), maximum retransmission count, the bit rate (for 802.11 networks), enabling or disabling noise sampling and noise sampling rate. After setting up the experiment parameter, the HTML/PHP interface invokes Phyton scripts to ensure host-mote communication for performing specific operations, namely sending commands to motes (to control them) and storing raw packet-statistics retrieved from motes into a MySQL database. Once the experiment has finished, SWAT allows users to specify which of the supported metrics they would like to calculate from the stored data. Supported metrics include hitherto packet delivery temporal and spatial correlations, noise floor distribution, received signal strength to reception ratio correlation, link asymmetries, and reception ratio distribution over time.

3.2.3 Discussion

Fixed topology based testbeds are not much suitable for our case of study, as they suffer from several weaknesses: The physical topology of sensor nodes as well as the environment conditions cannot be managed which is important in evaluating LQEs. Flexible topology based testbeds cope with this shortcoming, however they don't fit much well with our research objective. For instance, using DSN testbed demands the usage of BTnode rev3 Bluetooth devices. This kind of devices is not available in our Research Unit and it will be very hard and expensive to bring them. Further, although SCALE [18] and SWAT [47] have been devoted for links measurements, they still present some limitations. In fact, in one hand, SCALE is only compatible with old platforms (MICA 1 and MICA 2 motes) which do not support the LQI metric. This metric has been shown as important to understand and analyze channel behavior in WSNs [49, 15]. On the other hand, SWAT is not practical for large-scale experiments, as some configuration tasks are performed manually especially the designation of the experimental nodes. Furthermore, link measurements are collected at receiver's side only. This shortcoming doesn't enable the computation of sender side estimator like RNP or FourBit.
According to these observations none of existing testbeds is suitable for the experimentation of different LQEs. Therefore, it has been proposed to introduce a new one that provides the required features enabling the performance analysis of link quality estimators. An in-depth description of our testbed is presented in what follows.

3.3 RadiaLE: A Framework For Benchmarking Link Quality Estimators In WSNs

3.3.1 Overview

RadiaLE [9] is a testbed designed in major part by Nouha Baccour in the context of her PhD thesis to which this master thesis belongs. It is implemented in collaboration between ReDCAD research unit at ENIS (Ecole Nationale d'Ingénieurs de Sfax, University of Sfax), CISTER Research Unit at ISEP/IPP (Instituto Superior de Engenharia do Porto/ Instituto Politécnico do Porto) and the Department of Automation and Systems (DAS) of the Federal University of Santa Catarina (UFSC).
The objective behind the development of RadiaLE consists in enabling the performance evaluation of LQEs by analyzing their statistical properties, independently of any external factor, such as collisions and routing, only the impact of the physical layer and the data link layer (retransmissions) are considered. These statistical properties will be exploited to make conclusions about the stability and the degree of over-estimation of studied link quality estimator.
The global architecture of RadiaLE as depicted in Figure 3.8 encompasses hardware and software components. The hardware components consist principally in the deployed sensor nodes. Software components are a set of applications developed to program mote, to define experiments settings, to automate packets-statistics collection, and to study of statistical properties of different LQEs by analyzing collected data.

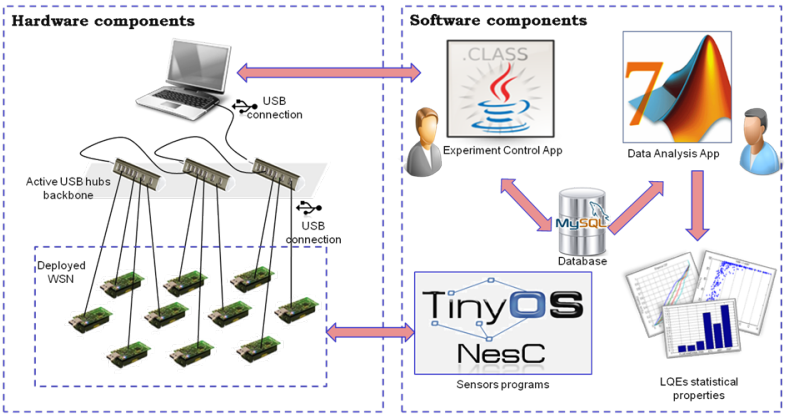


Figure 3.8: RadiaLE components

3.3.2 RadiaLE Implementation

3.3.2.1 Hardware components

The hardware components as illustrated in Figure 3.8 comprise motes, a USB backbone and a laptop.

- **Motes:** RadiaLE supports platforms integrating a Universal Serial Bus (USB) such as Telosb or Tmote sky. The motes are programmed in NesC over TinyOS 2.x operating system.
- **USB backbone:** All the motes are connected to a standard laptop PC using a combination of USB cables and active USB hubs constituting an USB backbone. This backbone is used as a logging/control channel between the motes and the PC.
- **PC:** A RadiaLE user interacts with the mote through an Experiment control application installed on the PC.

3.3.2.2 Software components

RadiaLE software components consist in three applications:

- A NesC application to be installed directly on the motes. This application has been developed under TinyOS 2.x operating system.
- An experiment control java application (ExpCtrApp) providing a graphical user interface to ensure *(i.)* motes programming and control, *(ii.)* network configuration, and *(iii.)* link measurements storage.
- A data Analysis Matlab application (DataAnlApp) providing a second graphical user interface for different LQEs, computation, tuning, and performance analysis (including graph generation).

- A MySQL database for permanent experiments link measurements storage.

A detailed descriptions of ExpCtrApp and DataAnlApp are presented next.

3.3.2.2.1 The experiment Control application (ExpCtrApp) The ExpCtrApp offers several functionalities including:

Motes programming and control [Fig. 3.9]: The ExpCtrApp automatically detects the motes connected to the PC and programs them by installing the compiled NesC application. The NesC application defines a set of protocols for any bidirectional communication between motes and between motes and the ExpCtrApp. The ExpCtrApp also, exchanges commands with the motes to control data transmission according to the experiment settings defined at the network configuration phase.

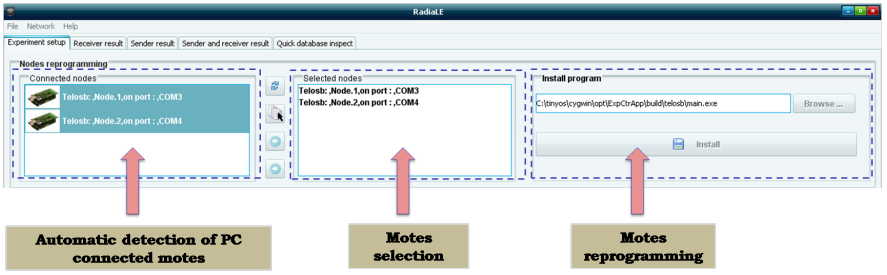


Figure 3.9: Motes programming and control interface

Network configuration [Fig. 3.10]: Using ExpCtrApp, the user can specify a set of parameters, including the traffic pattern, number of packets, inter-packet interval, packet size, radio channel, transmission power, link layer retransmissions on/off and maximum retransmission count. Once these settings are communicated to the motes, they start performing their tasks.

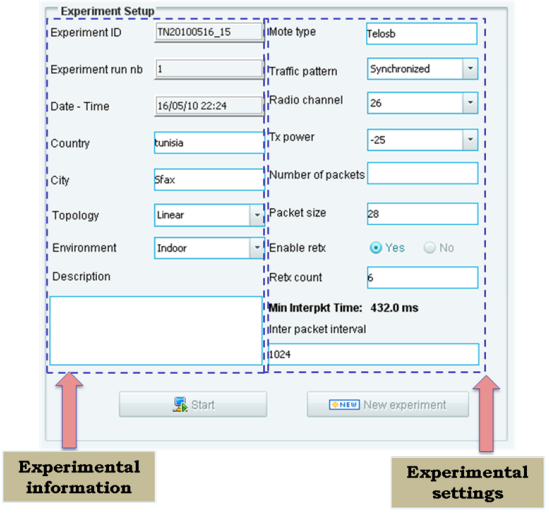


Figure 3.10: Network configuration interface

Most of the studies on link quality use single Burst traffic when experimenting WSN. The sending policy in this traffic pattern consists in: each node sends a burst of packets to each of its neighbors and then passes the token to another node to send its burst. This kind of traffic pattern cannot accurately capture the link asymmetry property as the two directions (uplink and downlink) will be assessed in separate time windows. RadiaLE copes with this shortcoming and integrates two traffic patterns as showed in both Figure 3.11 and Figure 3.12 (Burstyx and Bursty) which enhance the accuracy of link asymmetry assessment. Given a link $Ni \leftrightarrow Nj$; In the Bursty traffic and after receiving **"start sending command"** from the PC, the mote Ni sends a first burst of packets to Nj. When it finishes, it notifies the PC by a **"data end sending report msg"**, to allow Nj sending its first burst of packets to Ni. This operation is repeated for a pre-fixed number of bursts. However, in the synchronized traffic, Ni and Nj are synchronized to exchange packets (one packet a time). For that, the PC sends a **"start sending command"** to each mote indicating the beginning of transmission time so that the mote sends its data in an exclusive time slot (to avoid collisions). When each node finishes the sending of its data, it notifies the PC by a **"data end sending report msg"**.

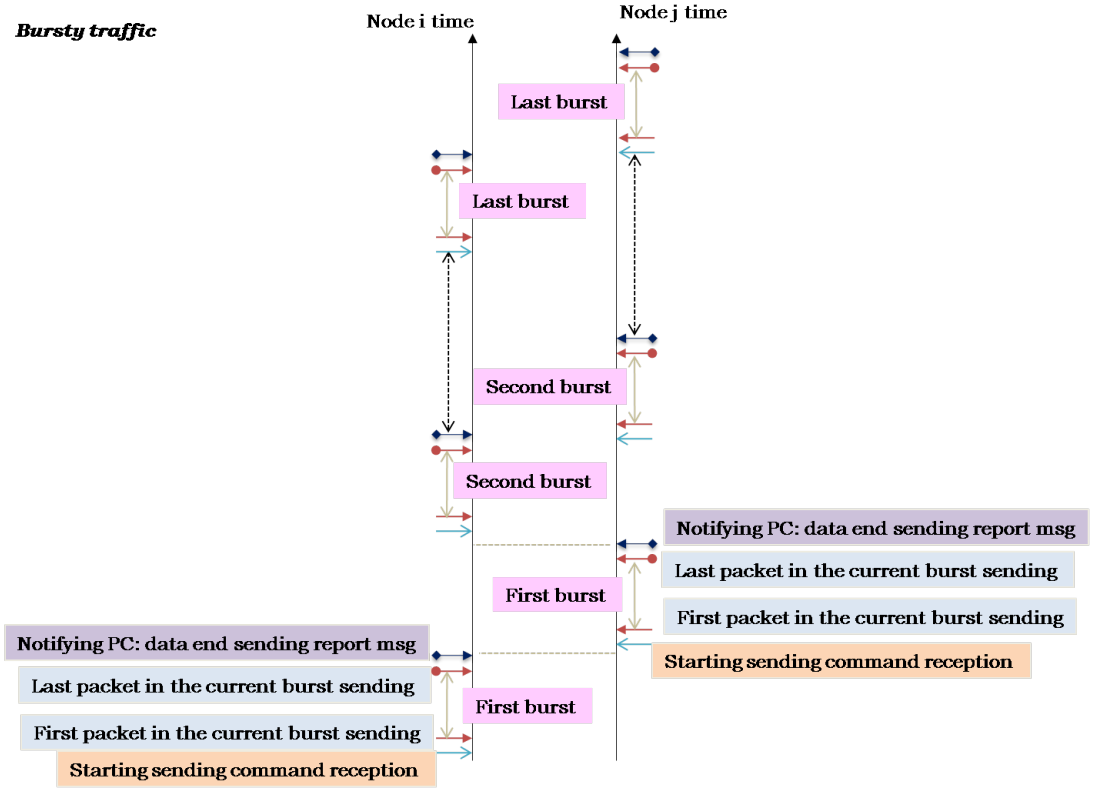


Figure 3.11: Bursty traffic

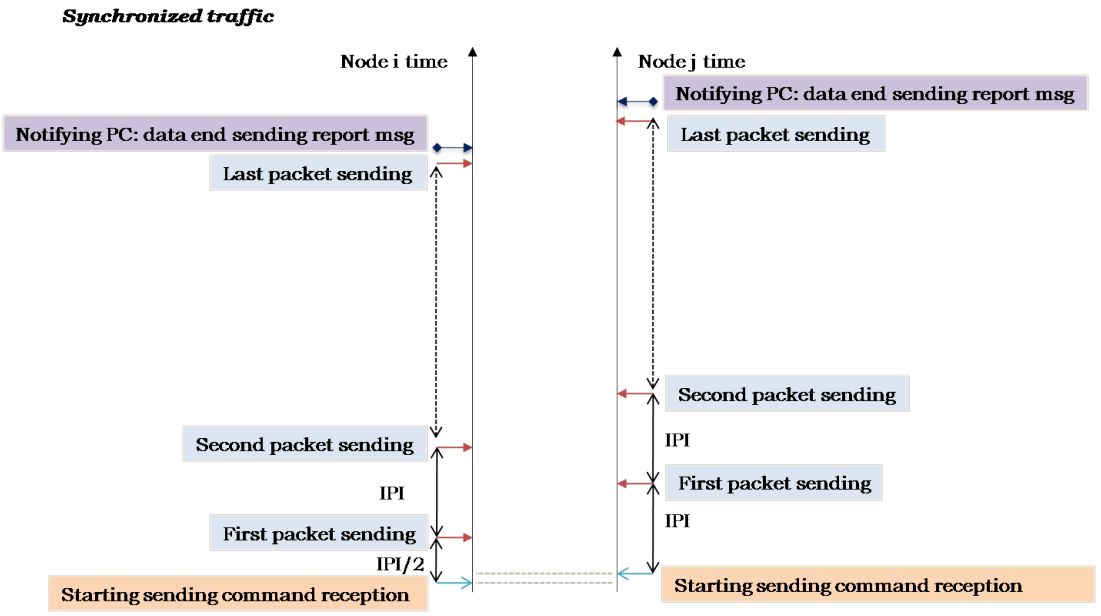


Figure 3.12: Synchronized traffic

Data logging: The main task of the motes is to exchange data traffic in order to collect link measurements such as packets' sequence number, sender and receiver ids, packet retransmission counts, LQI, Signal-to-Noise Ratio (SNR), Received Signal Strength Indicator (RSSI), background noise, etc...
Link measurements are sent though the USB backchannel to the ExpCtrApp, which in turn stores them into a MySQL database.
ExpCtrApp also provides: *(i.)* a Network Viewer [Fig. 3.13] to visualize the network map and the link quality metrics (e.g. PRR, RSSI) in real-time and *(ii.)* a Database Inspector [Fig. 3.14] to view in real-time data retrieved from the sensor nodes.

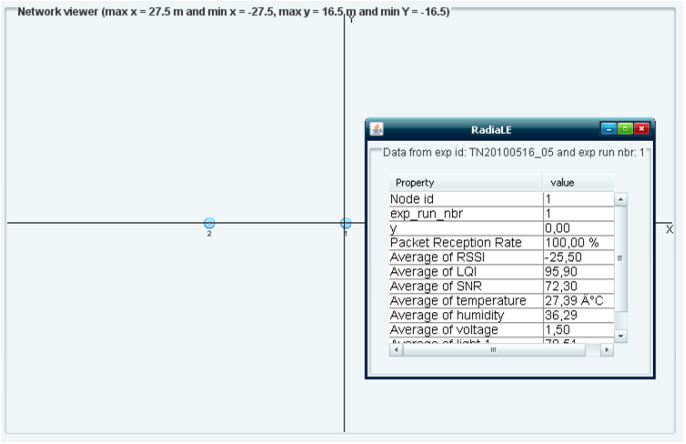


Figure 3.13: Network Viewer

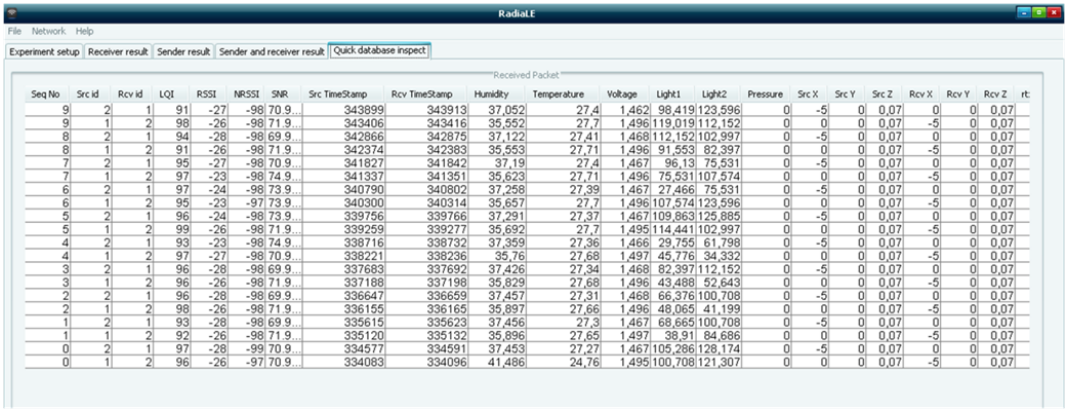


Figure 3.14: Database Inspector

3.3.2.2.2 The data analysis application (DataAnlApp)

The ExpCtrApp offers two basic functionalities: *(i.)* Links characteristics study (link characterization) and *(ii.)* Link quality estimator statistical properties study. To use DataAnlApp, a user is invited first to connect to the database where experiments measurements were stored as depicted in Figure 3.15

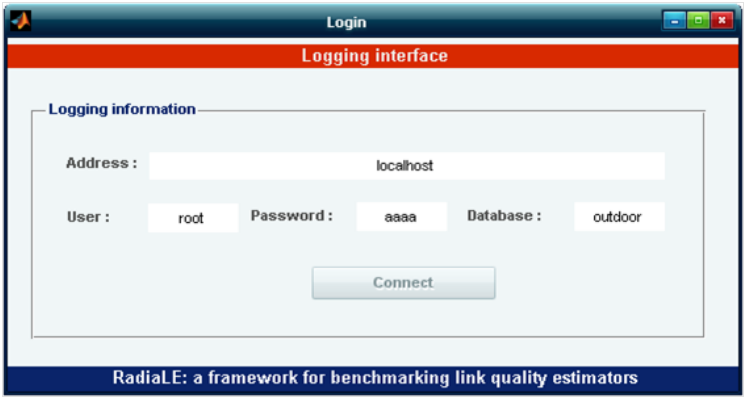


Figure 3.15: Database connection interface

After selecting which experiment measurements will be studied, a user chooses either link characterization functionality or link estimation functionality.

Links characterization: Figure 3.17 depicts the link characterization interface. This interface provides a set of configurable graphs, allowing the study of underlying links' properties. Such graphs help to design new LQEs by understanding the channel behavior.

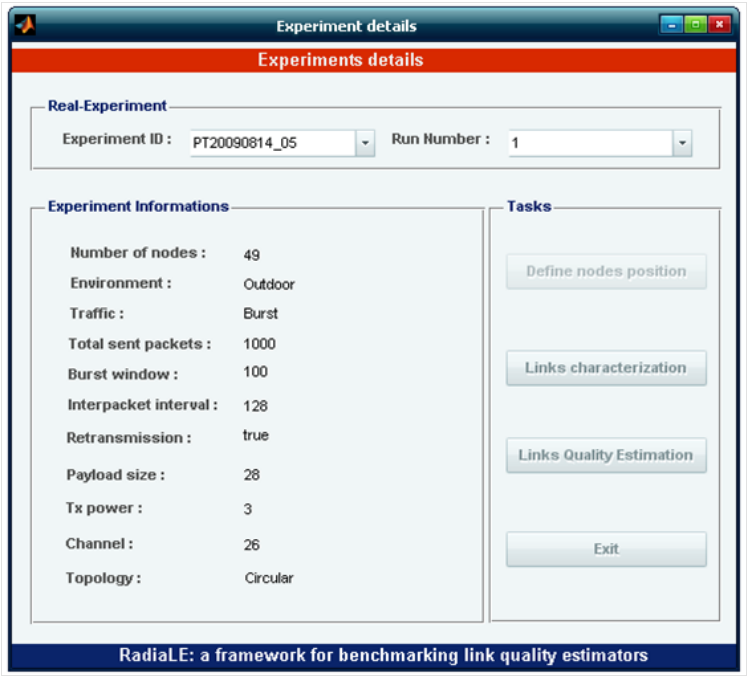


Figure 3.16: Main DataAnlApp interface

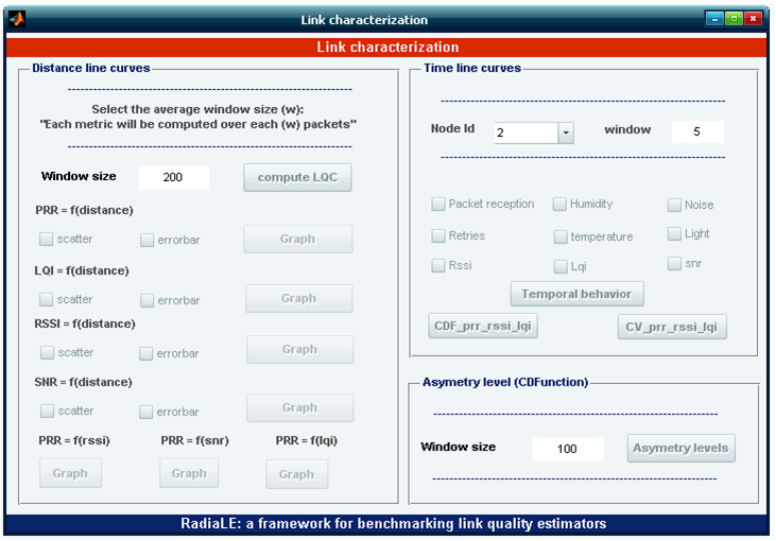


Figure 3.17: Link Characterization interface

Link quality estimation: Figure 3.18 depicts the link quality estimation interface. This interface provides assistance to RadiaLE users to evaluate and optimize their LQEs. It enables the generation of various statistical graphs, such as the empirical cumulative distribution function and the coefficient of variation of link quality estimates. By analyzing these graphs the stability and the over-estimation of LQEs can be evaluated. Furthermore, DataAnlApp integrates others interesting

curves, such as the evolution according to time and the distribution according to space (scatter plot) of each LQE.

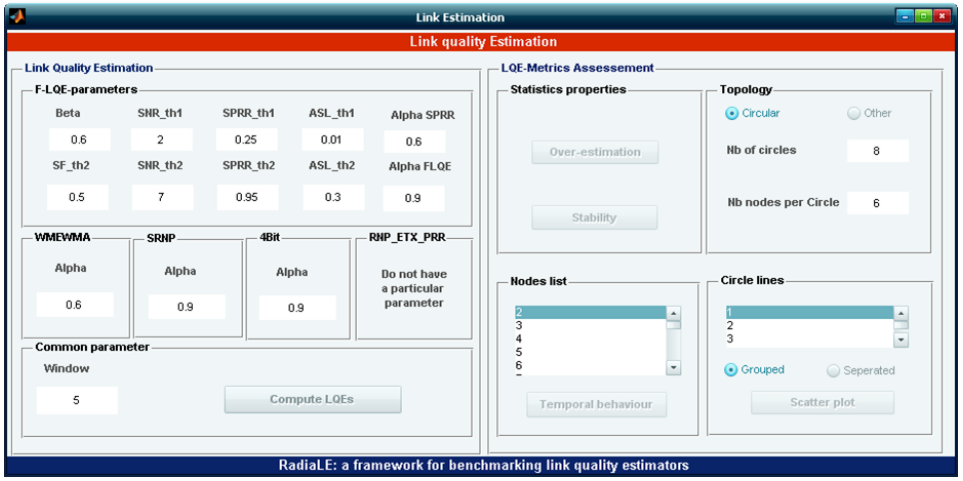


Figure 3.18: Link Quality Estimation interface

Hitherto, DataAnlApp integrates the set of well-known LQEs which are PRR, WMEWMA, ETX, RNP, SRNP, FourBit and FLQE. Other LQEs can be easily integrated and compared to existing LQEs, due to the flexibility and completeness of the collected empirical data.

3.4 Conclusion

In this chapter we started by reviewing the state of the art of existing testbeds designed for the experimentation of WSNs and then we presented RadiaLE, an experimental benchmarking tool that automates the experimental evaluation and the design of Link Quality Estimators. RadiaLE has several advantages compared to existing testbeds such as providing abstractions to the implementation details and the flexibility and completeness of the collected data. Further, RadiaLE testbed has allowed us to conduct an experimental study in order to analyze and understand the statistical properties of different LQEs independently of any external factor, such as collisions and routing. LQEs statistical properties study is presented in the next chapter.

Chapter 4

LQEs Statistical Properties

4.1 Introduction

LQEs statistical properties study reflects LQEs' performance, in terms of reliability and stability. Reliability refers to the ability of the LQE to correctly characterize the link state with neither **over-estimating** nor **under-estimating** the quality of the link. The **stability** refers to the ability of the LQE to resist to transient (short-term) variations, also called fluctuations, in link quality[**?**]. The current chapter elaborates on LQEs statistical study by firstly introducing our experimental methodology and then by presenting and discussing the results of the conducted experiments.

4.2 LQEs' Statistical Properties Study

4.2.1 Experiments Setup

The proposed methodology to evaluate the performance of LQEs consists first, in establishing a rich set of links with different qualities, second in creating bidirectional data traffic over links to gather different measurements required for the computation of LQEs and finally in analyzing gathered data, computing LQEs and studying their statistical properties.
The conducted experiments were performed in an outdoor environment in a garden (no people walking around motes) at ISEP/Porto as part of the Master thesis of Denis Lima Do Rosario.
The experimental setup consists of 49 Telosb motes connected to a control station (PC) via a USB backbone. The 49 Telosb motes run the TinyOS 2.x operating system with our implemented NesC application. A Telosb mote is a battery powered wireless module with USB programming capabilities. Its wireless transceiver is an IEEE 802.15.4 compliant chipcon working in the 2.4 GHz frequency band reaching up to 250 kbps as data rate. Telosb embeds an 8MHz Texas Instruments MSP430 microcontroller (10k RAM, 48k Flash) working at ultra low power consumption, and integrates Humidity, Temperature, and Light sensors. For enabling data transmission over long USB cables, we have used the HU-5870V USB hubs of TRUST company.
The 49 Telosb motes (from mote $N1$ to mote $N49$) have been spanned over the

grass according to a circular shape topology as depicted in Figure 4.1. The mote $N1$ is placed in the central of circle, while the 48 other motes (from $N2$ to $N49$) are divided in 8 set of 6 motes. Each set of motes is placed in a circle around $N1$. The inter-distance "Y" between two consecutive circles is fixed to 0.75 meter, while the first circle is distant from $N1$ X meters equals to 3. The decision of choosing circular topology wasn't arbitrary. Since our first objective in this study is establishing a rich set of links with different qualities, exploiting WSN links properties was the intuitive solution toward achieving this goal. In fact, as outlined in chapter 2, the connectivity (quantified by the Packet Reception Ratio) of WSN links is neither isotropic nor monotonically decay according to distance. Hence, links $N1 \leftrightarrow Ni$ where $2 \leq$ i ≤ 49 will have various qualities. Further, extensive experiments were done in Porto in order to capture the extent of the gray area. "X" and "Y" have thus been chosen so that links $N1 \leftrightarrow Ni$ where $2 \leq i \leq 49$ fall in this gray area.

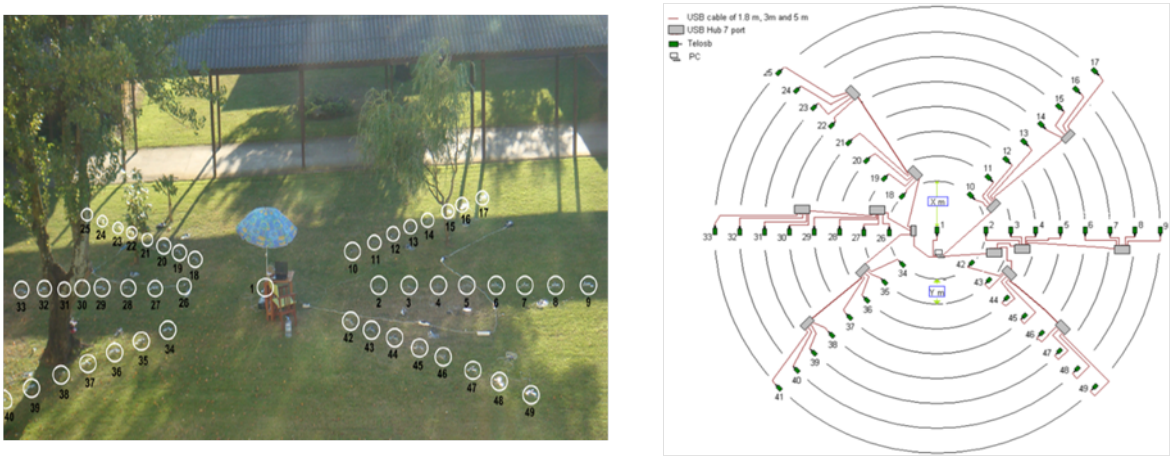


Figure 4.1: Circular topology

4.2.2 Experiments Scenarios And Results

4.2.2.1 Scenarios

In addition of exploiting link properties to create a rich set of links with different qualities, we have exposed LQEs in study, to different network configuration. In fact, we performed three scenarios where in each one we vary a certain parameters in order to study their impact on the behavior of the LQE. We considered:

- The impact of day and night (changes in terms of temperature and humidity)
- The impact of the packet size (we used the minimum and the maximum sizes)
- The impact of the channel (the channel 26 is an interference free channel with 802.11 networks in the vicinity).

Links under estimation were the unidirectional links $Ni \rightarrow N1$ and the chosen traffic pattern was the bursty traffic. Recall the description of a bursty traffic: Given a

link $Ni \leftrightarrow Nj$; Ni sends a first burst of packets to Nj. When it finishes, it notifies the PC, to allow Nj sending its first burst of packets to Ni. This operation is repeated for a pre-fixed number of bursts. In our scenario, we have fixed the number of packets per burst to 100 and the number of burst to 10.
The transmission power has been set to -25 dBm (it is the minimum transmission power as we were constrained by the garden extents) and the inter packet interval was set to 100 ms.
The duration of each experiment was approximately 8hs. Figure 4.2 depicts the chosen scenarios.

	Climate	Environment	Trafic type	Packet size	Tx power	Channel	Retx
Scenario 1	Night, day	outdoor	Burst 1	28	-25	26	6
Scenario 2	Day	outdoor	Burst 1	28, 114	-25	26	6
Scenario 3	Day	outdoor	Burst 1	28	-25	20, 26	6

Traffic pattern	# of Pkts to send per burst	Inter Packet Interval	# of Bursts
Burst1	100	100 ms	10

Figure 4.2: Experiments scenarios

4.2.2.2 Results

4.2.2.2.1 Over−Under Estimation

The over or the under estimation of LQEs was assessed by studying the distribution of link quality estimates, illustrated by empirical cumulative distribution function, CDF. In statistics theory, The cumulative distribution function (CDF) describes the probability distribution of a real-valued random variable "X". For every real number x, the CDF of a real-valued random variable "X" is given by:
$F_X(x) = P(X \leq x)$
Where "P" is the probability: that the random variable "X" takes a value less or equal to x.

The results of our experiments are depicted in next Figures[1]:

[1]PRR, SPRR (WMEWMA) and FLQE share the same scale, as the same case for RNP, FourBit and ETX. That's why we separate between Figures

DAY / NIGHT

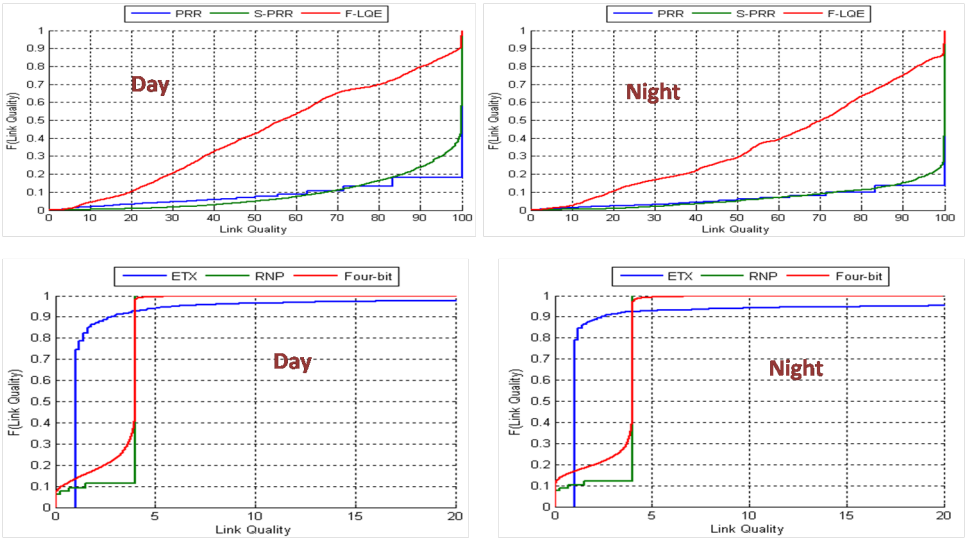


Figure 4.3: Over-under Estimation: Day/Night Impact

MIN PACKET LENGTH / MAX PACKET LENGTH

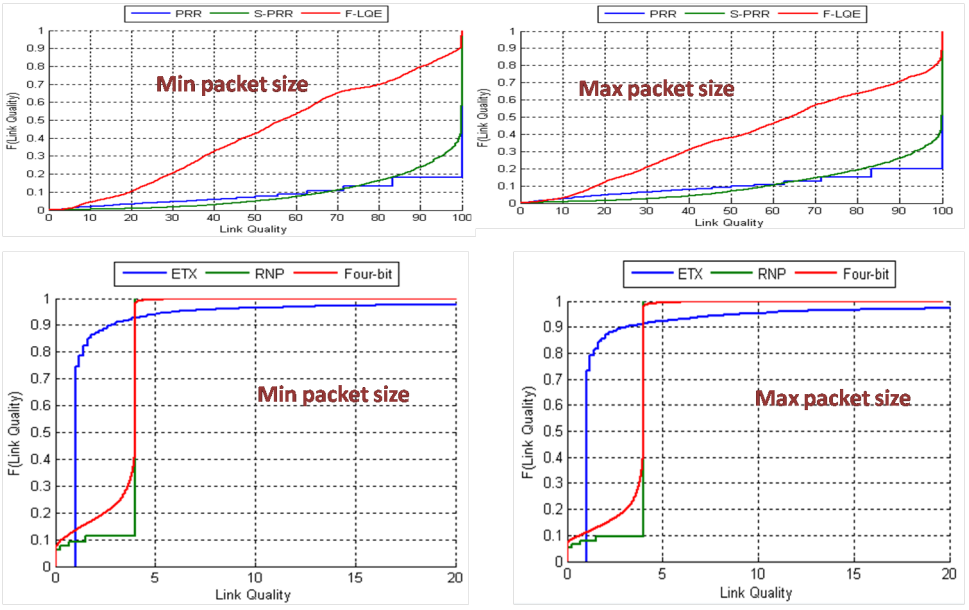


Figure 4.4: Over-under Estimation: Packet Size Impact

CHANNEL 26 / CHANNEL 20

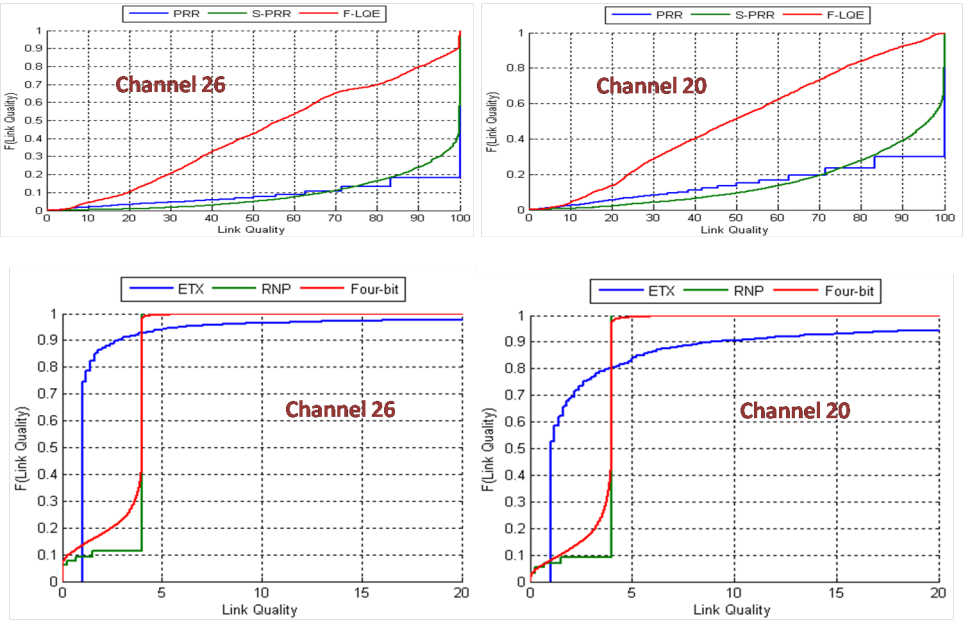


Figure 4.5: Over-Under Estimation: Channel Impact

From the above Figures we can see that PRR, SPRR (WMEWMA) and ETX overestimate link quality as they judge most of the links to have good quality (for the case of PRR and SPRR: about 80% of estimates are $\geq$ 90% and for the case of ETX only about 25% of estimates are $\geq$ 4 which is the maximum value). In contrast, RNP, and Fourbit underestimate link quality as they consider most of the links as bad (for the case of RNP: about 90% of estimates are $\geq$ 4 and for the case of FourBit about 60% of estimates are $\geq$ 4). Fourbit is shown to be less pessimistic than RNP as its computation includes PRR values.
The underestimation of RNP and FourBit is justified by the fact that they are not able to determine if the packets are received after retransmissions or not. The discrepancy between results of the first estimators group (PRR, WMEWMA and ETX) and the second estimators group (RNP and FourBit) is due to that most of the packets transmitted over the link are correctly received (high PRR) but after a certain number of retransmissions (high RNP). Further, PRR, WMEWMA and ETX (i.e: receiver side estimators) tend to overestimate the quality of links as they depend on packet reception event in order to update their estimates. In fact, given the Figure 4.6, If we consider an estimation window equals to 4 then PRR, WMEWMA and ETX estimators will judge the link quality as excellent based on their first estimates. However during the following period (packet losses period), these estimators will maintain this excellent link quality estimate as no reception is done. The absence of packet reception inhibits the updating of the link quality. This fact justifies the overestimation.

Comparing to other estimators, FLQE is in between. FLQE provides reasonable

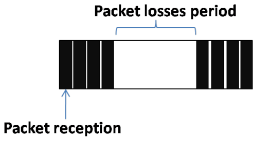


Figure 4.6: Major drawback of receiver side estimators: During packet looses period, estimate values can not be updated

link quality estimates (neither overestimates nor underestimates link quality). Furthermore, the distribution of link quality estimates is nearly a uniform distribution, which means that FLQE is able to distinguish between links having different link qualities.

4.2.2.2.2 Stability

A radio link may show transient quality fluctuations due to transient changes in the environment of deployment or also due to the nature of low power radio transceivers that sensor nodes use. These transceivers have been shown to be very prone to noise. LQEs should resist to these fluctuations and provide stable link quality estimates. This property is of paramount importance in WSNs. For instance, routing protocols do not have to reroute information when a link quality show transient degradation, because rerouting is a very energy and time consuming operation.[12].
To assess the stability of LQEs, we will measure the coefficient of variation of their estimates. In statistics theory, the Coefficient of Variation (CV) is a dimensionless number that characterizes dispersion relative to the mean. CV has the advantage of facilitating the comparison among random variables of different units [45]. The results of our experiments are depicted in next Figures:

DAY / NIGHT

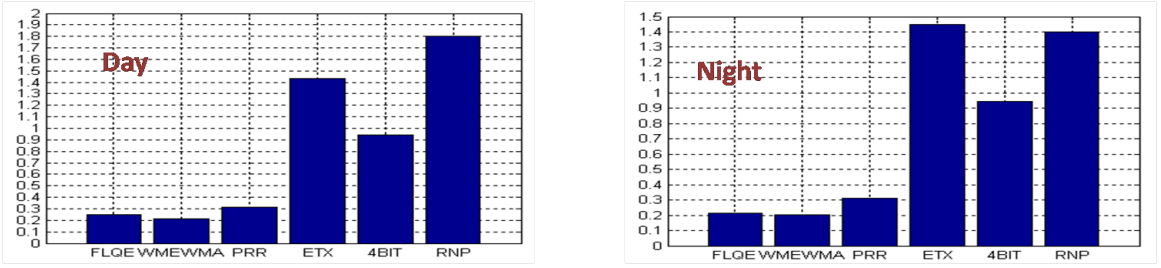


Figure 4.7: Stability: Day/Night Impact

MIN PACKET LENGTH / MAX PACKET LENGTH

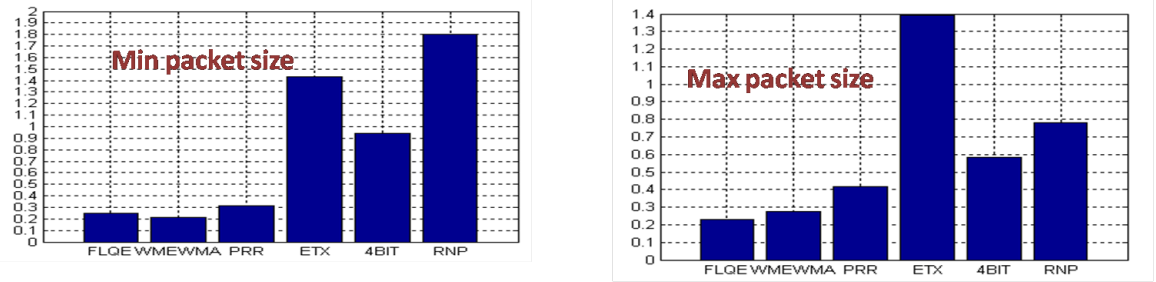


Figure 4.8: Stability: Packet Size Impact

CHANNEL 26 / CHANNEL 20

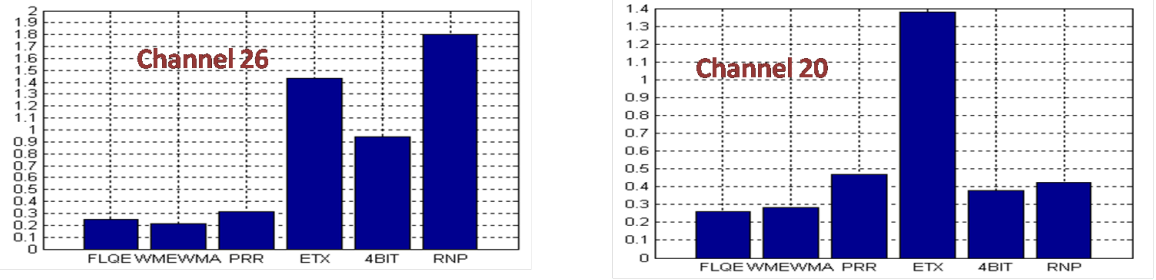


Figure 4.9: Stability: Channel Impact

From the above Figures we can see that utmost the time FLQE is the most stable estimator. Further, except ETX, estimators that tend to overestimate link quality are stable, in contrast estimators that tends to underestimate link quality are unstable.
The reason behind the instability of ETX is that when the PRR tends to 0 (very bad link) the ETX will tend to infinity, which increase the standard deviation of ETX link estimates [12].

4.3 Conclusion

In this chapter we addressed the performance evaluation of existing LQEs by analyzing their statistical properties. To achieve this goal, we used our own implemented RadiaLE which is a testbed for the benchmarking of Link Quality Estimators.
Experiment results show that FLQE demonstrates grater performance than other LQEs, namely PRR, ETX, WMEWMA, FourBit and RNP.
Studying the statistical properties of different estimators was an important step to

get prior conclusions about each LQE performance. However, studying their impact on higher layer protocols will be much more conclusive.
Towards this end, the next chapter elaborates on the impact of LQEs on CTP tree routing protocol.

Chapter 5

LQEs Impact On Routing

5.1 Introduction

In this chapter we pursue the performance evaluation study of LQEs by investigating their impact on CTP, a Collection Tree routing Protocol. This performance study is conducted with both simulation using TOSSIM2 simulator and real experimentation using MoteLab testbed combined with an extended version of RadiaLE. Indeed, as it was mentioned in the previous chapter, simulation provides a rapid prototyping solution, however real experimentation provides much more accurate, realistic and trust-able results. To elaborate on the impact on routing study, this chapter starts by giving a brief overview of CTP routing protocol. Next, a description of the experimental methodology is introduced. Finally, the results of experiments are presented and discussed.

5.2 CTP: a Collection Tree Routing Protocol

CTP (Collection Tree Protocol) is a multi-hop routing protocol provided by Tinyos operating system (v2). The CTP routing policy consists in building a tree towards the sink node (referred also as collection point [30]) based on links quality information.
We identify for that three types of node in the network:

- *The sink node:* It is the root of the tree to which all transmitted data packet are forwarded.
- *The parent node:* The established tree has a hierarchical structure. Each parent has a certain number of childs. Except the root node, each parent can be a child of another parent situated above it in the hierarchy.
- *The child node:* Each child is associated to a single parent. Each child can be also a parent for other nodes situated bellow it in the hierarchy.

The CTP implementation [2] has three basic components which are:

- *Estimating Engine:* Each node estimates the quality of links joining it to its neighbors, using Fourbit link quality estimator (2.9).

- *Routing Engine:* Each node in the network, elects the neighbor with the best path quality as its parent. This operation is done by all the nodes in the network, except the root node, in order to build the tree shape topology.
- *Forwarding Engine:* The data packets generating and forwarding are managed in this component. Each node maintains a forwarding queue storing the pending data packets and schedules their transmission to the next hop.

The following sections describe in depth the functionalities of each component.

5.2.1 CTP Components

5.2.1.1 Link Quality Estimation Engine

The link quality estimation engine includes the implementation of Fourbit estimator which relies on both beacon-driven and data-driven estimation. The beacon-driven estimates are computed based on the received beacons traffic, while the data-driven estimates are computed based on the sent data packets. As outlined in chapter 2, Fourbit computes the estETXdown (2.8) estimates based on the beacons traffic by tracking the gap between sequence numbers. The update of estETXdown is trigged after the reception of BLQ beacons. BLQ represents to the **beacon window**. Further, based on the number of transmissions and retransmissions of sent data packets, Fourbit computes the estETXup (RNP). The estETXup is updated after DLQ transmissions/retransmissions. DLQ represents to the **data window**. Finally Fourbit combine both values through a EWMA filter (2.9).
To store link quality estimates, each node maintains a *"link table"* composed of a set of entries representing links with its neighbors. This table contains the following information:

- The address of neighbor
- The last sequence number of last received beacon
- The number of received beacons during a beacon window
- The number of failed beacons during a beacon window
- Flags indicating some states about each neighbor
- The beacon driven estimate value (2.8)
- The Fourbit estimate value (2.9)
- The number of received acks for the sent data packets during a data window
- The number of transmission and retransmissions that were done during a data window

Each node periodically broadcasts beacons containing a subset of these links. This has the advantage of sending receiver side estimates to senders as senders could not know these information alone.
The replacement policy in the *"link table"* is governed by the usage of a compare and a pin bit. If the pin bit is applied to an entry of the *"link table"*, this means that this entry could not be removed until the bit is cleared. The compare bit is checked when a beacon is received and it indicates whether the route provided by the sender of the beacon is better than the route provided by one or more of the entries in the link table.

5.2.1.2 Routing Engine

This engine is responsible of the building and the controlling of the tree topology. Each node in the network maintains a *"routing table"* where routing data are stored. Each entry in the table contains basically the following information:

- The addresses of neighbors
- The addresses of neighbors parents
- The neighbors costs. A node cost is the sum of the parent cost and the quality of the link (link cost) joining him with its parent. As node cost computation is a recursive relation, CTP sets the initial cost which is the cost of the root node to zero. We note that the tree routes building starts from the top of the tree which means from the root node.

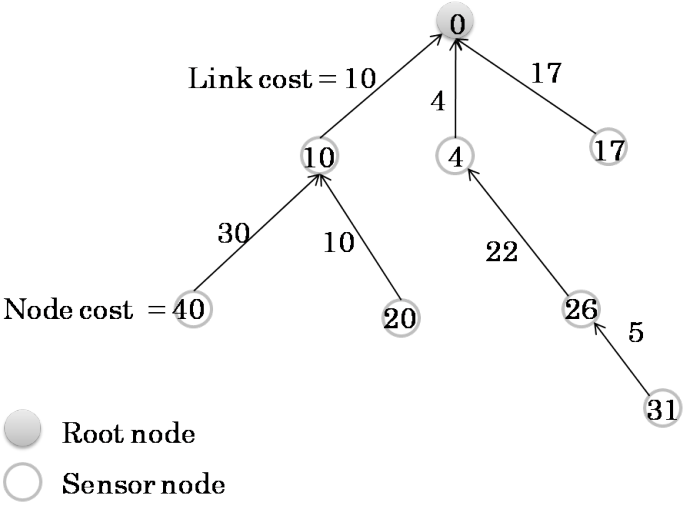


Figure 5.1: Relationship between Link cost and Node cost

- Indication if the node is congested

The filling and the updating of the *"routing table"* is maintained by beacons sending. These beacons are sent according to the trickle algorithm based timer (briefly said trickle timer) [36]. Trickle timer is used in CTP to address link dynamics by sending few beacons in stable topologies and by sending heavy load beacons in initialization phase or when loops are detected by the forwarding engine.

5.2.1.3 Forwarding Engine

Data packets generation, transmission to the next hop, retransmission when necessary and acknowledgment passing to the link quality estimation engine are managed in this component. For that, each node maintains a forwarding queue containing locally generated and forwarded packets. A packet will be ejected from the queue if it has been acknowledged or if it has reached the maximum allowed retransmissions count. Single-hop transmission duplicates caused by lost acknowledgments in addition to routing inconsistencies are also detected by this engine. Routing inconsistencies are detected if a node receives a data packet from a neighbor with cost less than its own as the cost must always decrease from child to parent. In this case, the forwarding engine drops the packet and triggers the topology updating task.
The following section summarizes the working of CTP.

5.2.2 CTP: How It Works

A network consists initially of orphan nodes spread in the deployment area. Each node starts by broadcasting a set of control beacons with a very high rate. Control beacons are used to explore neighbors and to exchange topology information which are mainly the node's current parent and node's cost (initially set of 0xFFFF). A control beacon contains also two control bits: the Congestion and Pull bits. The Congestion bit is used to indicate if a node is congested while the Pull bit allows a node to pull advertisements from its neighbors, in order to quickly discover its local neighborhood [30]. When a node receives control beacons from the root node then it elects it as a parent and changes its cost to the quality of "the joining link". The topology information contained in control beacons received from other neighbors are taken into account and stored in the routing table only if these neighbors are not orphan otherwise they will be neglected. An orphan node is identified by a parent address equals to infinity (0xFFFF). To select a parent, the orphan node searches the best neighbor existing its routing table. The best neighbor is the node who minimizes the orphan node cost. The orphan node cost will be the sum of the selected neighbor cost and the quality of "the joining link". The "joining link" quality is got from the link quality estimation engine. Once a node has a parent, the sending control beacon rate is reduced and the node monitors the quality of "the joining link" through the sent data traffic. As changing routes too quickly can harm the efficiency of the protocol, a node switches routes if it only believes the other route is significantly better than its current one [30].

5.3 LQEs Impact On Routing Study

In order to evaluate the impact of LQEs on the CTP protocol, we performed experiments with both simulation and real experimentation. Simulation was used to have a rapid and global idea about the performance of each estimator when FLQE is included, however real experimentation was used to retrieve more accurate and realistic results.
The performance evaluation considers the following metrics:

- *The packet delivery ratio (PDR)*: It represents the ratio of the total number of successfully received packets in the network to the total number of transmitted packets in the network.
- *The number of retransmissions for each received packet (Rtx)*: it is inferred from the average number of packet re-transmissions over the network of a packet before it is correctly delivered to the root. The number of retransmissions information reflects the amount of wasted energy, as a retransmission is an energy greedy operation.
- *The average of parent changes (PrtChgt)*: It refers the stability of the network.
- *The average of hop count (HopC)*: It refers to the length of the tree.

5.3.1 Simulation Study

5.3.1.1 Experiment Settings

5.3.1.1.1 Simulation Environment

TinyOS(v2) operating system offers an event driven simulator called "TOSSIM" enabling the simulation of entire TinyOS applications without requiring any changes in the code. TOSSIM works by replacing used components with simulation implementations. The level at which components are replaced is very flexible. TOSSIM can replace a packet-level communication component for packet-level simulation. TOSSIM is a discrete event simulator this means when it runs, it pulls events (sorted by time) from the event queue and executes them. Depending on the level of simulation, simulation events can represent hardware interrupts or high-level system events (such as packet reception). TOSSIM supports two programming interfaces, Python and C++.
This simulator has been shown to provide accurate wireless channel models, which improves the correctness of our simulation results. However, TOSSIM does not enable currently power measurements gathering.
To run experiments with TOSSIM, a user is invited first to define simulation configuration which include the defining of nodes number, their coordinates, the channel and radio parameters.
TOSSIM integrates the most common radio propagation model which is the log-normal shadowing path loss model defined by the following equation:

$$PL(d) = PL(d_0) + 10nlog_{10}(\frac{d}{d_0}) + X_\sigma \tag{5.1}$$

Where "d" is the transmitter-receiver distance, "n" the path loss exponent (rate at which signal decays), "X" a zero-mean Gaussian (in dB) with standard deviation σ (multi-path effects), "d_0" a reference distance and $PL(d_0)$ the power decay for this distance.

5.3.1.2 Simulation Scenarios

To run simulation experiments, we consider an 81-nodes multi-hop network where nodes use Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) as MAC protocol, and CTP as routing protocol like in [10]. In order to alleviate congestion and collisions, only node with odd ids send data traffic, others are used as **relay nodes** **5.2**. Packet retransmission has been activated. Further, nodes begin their data packet transmission after a delay of 10 min (in order to enable the establishment of the topology). Each simulation is repeated 5 times. The experiment duration is 40 minutes. Sensor nodes are deployed in a non-uniform grid fashion as depicted in Figure 5.2 in an indoor environment. The sink node is located at coordinates (0,0). The grid unit varies in 4,14 meters as defined in [10]. As ETX estimator requires a constant rate, we have changed the trickle timer by a periodic timer for all estimators in study.

NB: We highlight that nodes implementing PRR, WMEWMA or FLQE estimators select their next hop based on the quality of the downlink and not of the uplink link. We highlight that data packets are sent over the uplink.

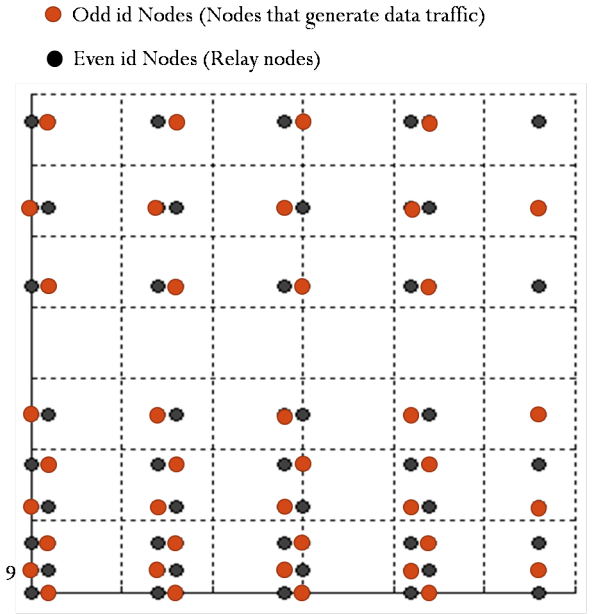


Figure 5.2: Non-uniform grid

5.3.1.3 Simulation results

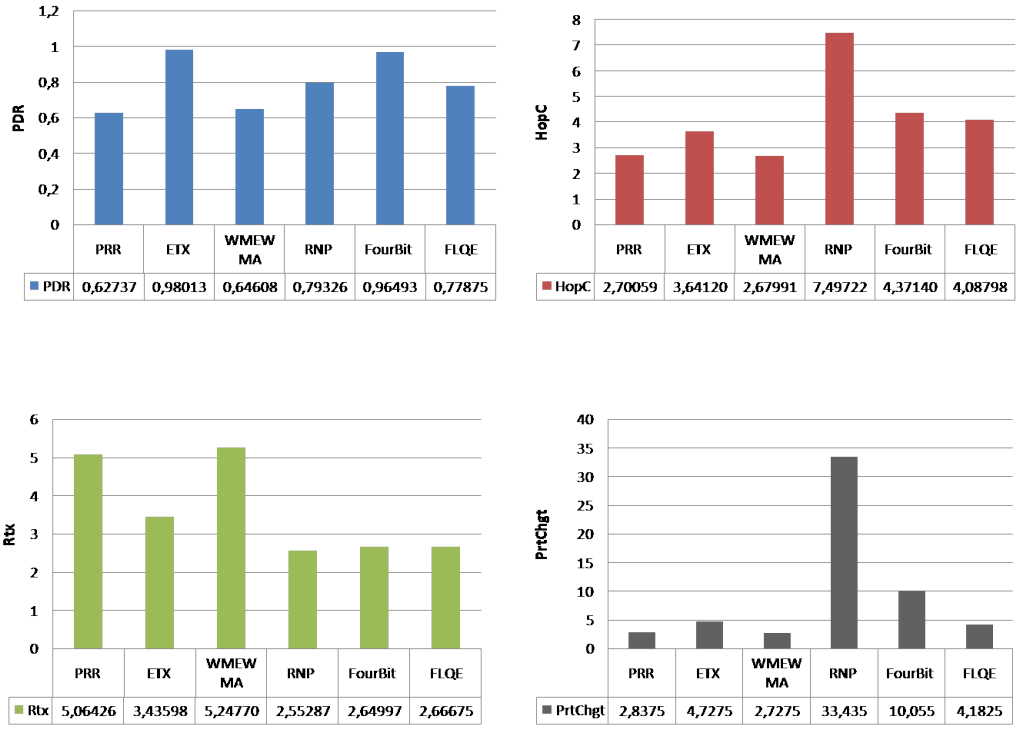


Figure 5.3: Simulation results

5.3.1.4 Discussion

Based on chapter 2, the studied Link Quality Estimators (LQEs) generally enable to count or approximate either the reception ratio (PRR) or the average number of packet transmissions/retransmissions (RNP) before its successful reception [10]. We name the first category of Link Quality Estimators as PRR−counting LQEs and we name the second category as RNP−counting LQEs. PRR, WMEWMA belong to the first category and ETX, RNP and FourBit belong to the second one. FLQE, in contrast, could not be associated to none of these categories as it has different philosophy. In fact, instead of counting or approximating either packet reception ratio or the average number of packet transmissions/retransmissions, FLQE gives a percentage of quality for a given link based on several properties combined using fuzzy logic.

By analyzing the above Figures, we can clearly see that RNP−counting LQEs show better performance in terms of packet delivery ratio (PDR) comparing to others. Further, by using RNP−counting LQEs, the CTP routing protocol builds its collection tree with longer paths which is not the case when using PRR−counting LQEs. This result could be justified to the fact that a sensor node tends to minimize the retransmissions count by selecting its nearest neighbors as eventual parents. Such deduction could be strengthened by analyzing the average retransmissions count. In fact, it is clear from the above Figures that RNP−counting estimators usage

significantly reduces packets retransmissions and thus enhances the energy saving. It is also important to note that the estimator that approximates the number of packet transmissions/retransmissions (ETX) shows less performance in terms of energy consumption reducing than those that count this value (FourBit and RNP). Indeed, ETX is a receiver side estimator. Thus its computation requires the reception of packets. Any lateness in packet reception leads to updating time increasing and consequently a sensor node exercises more retransmissions with the current bad link until its replacement. More the updating time is longer more the link quality estimator becomes less-responsive to link quality degradation. That's why, sender side estimators (RNP and FourBit) lead to the most unstable topology (frequent parent changes). In the other side, WMEWMA and PRR are the most stable LQEs but lead to lower packet delivery ratio and higher packet retransmissions count. In general, we can say that estimators that count or approximate the number of packet retransmissions (RNP-counting estimators) provide better performance than those that count packet reception (said PRR-counting estimators). FLQE in between gives medium performance.
FLQE has a similar performance with RNP in terms of packet delivery and packet retransmission but it is more stable and the routing paths are shorter.
The observed results about FLQE performance contradicts what it has been found in the statistical study. As we strongly believe that FLQE is a powerful estimator that may outperform existing LQEs, we ask the following question: Does the simulation abstractions are the causes behind this unexpected result?
To answer to this question, we need to re-experiment existing LQEs in a realistic environment. The following section elaborates on our real experimentation study.

5.3.2 Real Experimentation Study

5.3.2.1 Experiment Settings

5.3.2.1.1 Real Experimentation Environment

To conduct a large scale experimentation we have chosen to conduct our experiments using MoteLab testbed [29] jointly with an extended implementation of RadiaLE [9]. An overview about MoteLab testbed is already present in chapter 3. The usage of MoteLab is justified by the following reasons:

- It provides an open and free access to a large scale wireless sensor network consisting (theoretically) of 190 Tmote Sky sensor nodes [29].
- It is easy and quick to use (Based on our experience)
- It is often permanently available.

5.3.2.1.2 Real experimentation scenarios Our experiments are conducted in an indoor environment as MoteLab is deployed in Havard university building and the experiment network contains 13 nodes. Included nodes are:
3;9;10;13;15;16;17;18;19;20;23;25;26 where node number 3 is the root of the CTP tree. Sensor node are disposed as depicted in the next Figure.

Figure 5.4: MoteLab node positions

Packet retransmission has been activated and beacon traffic is constant. The transmission power is set to -15dBm and the chosen transmission channel is 26 (Interference free channel). Each experiment takes 40 minutes (10 min for the topology establishment and 30 min for data packets sending) and had been performed between midnight and 8AM (MoteLab time).
We summarize the experiment settings in the following table.

Table 5.1: First Impact on routing experiments settings

Data traffic	Constant (1 packet per 8 seconds)
Beacon traffic	Constant (1 beacon per second)
Selected Nodes	Depicted in Figure 5.4
Number of active nodes	13
Transmission power	-15dBm
Channel of transmission	26 (interference free channel)
Experiment times	between midnight and 8AM (MOTELAB time)
Experiment duration	40 min (Node start sending only beacon traffic in the first 10 min and then send data for 30 min)

5.3.2.2 Real experimentation results

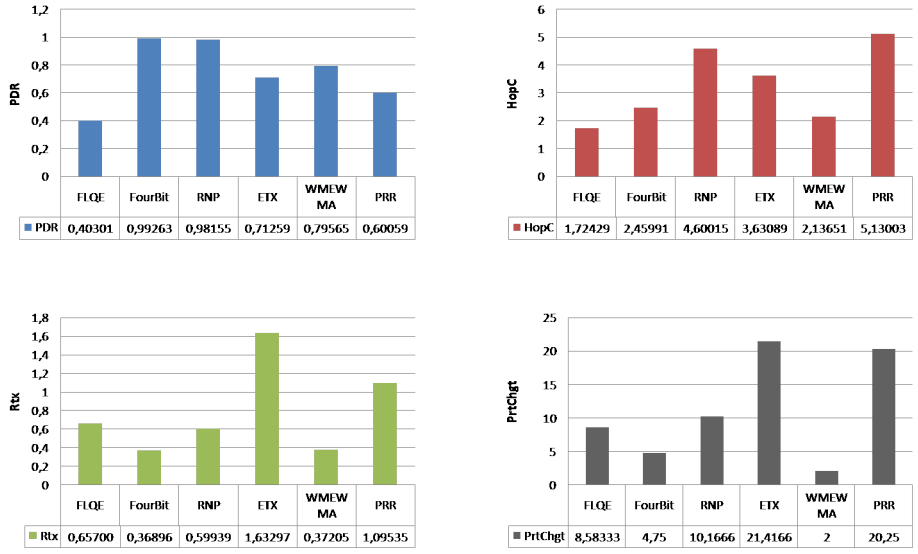


Figure 5.5: Real Experimentation results

The following Figure regroups both real experimentation and simulation results.

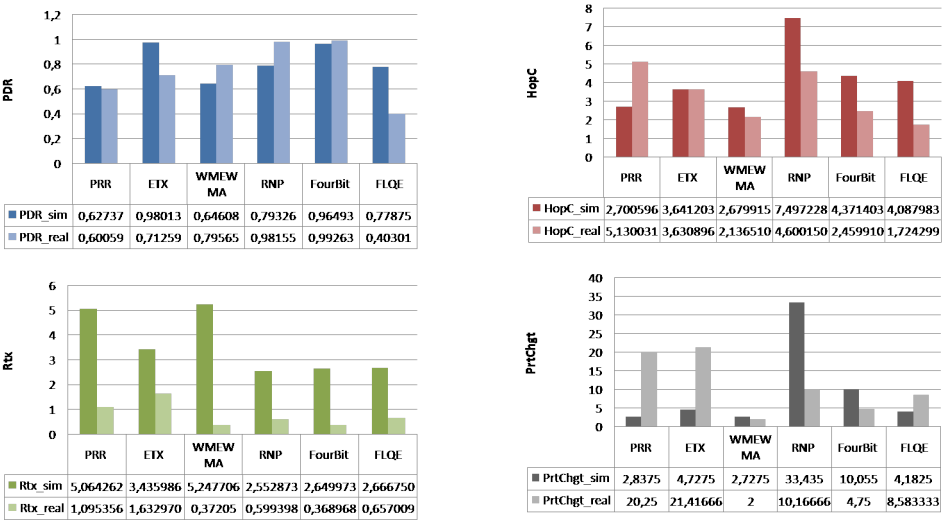


Figure 5.6: Real Experimentation and Simulation results

5.3.2.3 Discussion

It is clear from the above Figures that simulation results are quite different from real experimentation results. This discrepancy is mainly caused by the stochastic models that the simulator uses and by the difference is some experiment settings

especially the network density. In dense networks, collisions become more frequent leading to frequent packets failure and leading also to an increase in the retransmissions count. This observation justifies why simulation results show higher number of packet retransmissions and longer routes for all estimators than those of real experimentations.
By exploring the real experimentation results, we can conclude that FourBit is the best estimator among existing LQEs and FLQE followed by PRR are the worst ones. Comparing to other LQEs, FourBit has the advantage of exploiting information extracted from both received beacons traffic and sent data traffic. Other LQEs such as FLQE, PRR, ETX, WMEWMA and RNP rely exclusively on either received beacons traffic or sent data traffic. Consequently, the update of link quality estimates of FLQE, PRR, ETX and WMEWMA happens only at beacons reception and it is the same case for RNP, as a sensor node has a fresh idea about the RNP estimates only with its current parent to which it is sending its data traffic[1].
Further, we can conclude that LQEs estimating the quality of the uplink (sender side estimators) outperforms (in terms of Packet Delivery Ratio (PDR) and in terms of energy waste (Rtx)) other estimators that confuse the quality of the uplink by the quality of the downlink. This confusion is absolutely faulty where highly asymmetric links are present. In CTP, the distinguishing between the quality of both link directions is utmost important as the parent changing event should be trigged only if a degradation is detected on the link over which data are sent.
By comparing real experimentation and simulation results we may re-derive some conclusions about the performance of different LQEs. Indeed, RNP−counting LQEs, except ETX, still show better performance in terms of packet delivery ratio (PDR) comparing to others: more than 90% for RNP and FourBit. Furthermore, a concordance in both simulation and real experimentation results is shown in what concerns FourBit, RNP and WMEWMA estimators. FourBit and RNP are unstable but leading to a higher PDR however WMEWMA is the most stable LQE but leading to a lower PDR in comparison to RNP and FourBit. Nevertheless, it is important to note the PDR of WMEWMA is acceptable (about 80%).
ETX, PRR and FLQE real experimentation results are different from those of simulation. PRR and ETX present the worst behavior as the network topology is highly instable and the number of retransmissions is remarkably huge comparing to other LQEs. However ETX is better than PRR in terms of packet delivery. This instability could be justified to the presence of bursty reception ratio links where the PRR estimates for a given link fluctuate erratically between bad values and excellent values. The smoothed version of PRR (WMEWMA) is relatively resilient to this sudden fluctuations.
Also, by comparing real experimentation and simulation results, we can see that the performance of FLQE was remarkably dropped. In order to understand why such degradation is done. We recall some details about FLQE implementation. FLQE estimator assess the channel quality through SNR metric. In TOSSIM simulator, this measure is directly extracted from the software components. In real experimentation SNR computation is split in two phases: the first one consist in extracting

[1]Retransmissions are enabled for data traffic only. Beacons are transmitted in broadcast, so no retransmission scheme is set.

RSSI from the received packet and then in measuring the noise. By separating noise from the received signal, a sensor node can compute the signal to noise ratio for the received packet. This computation requires a high memory overhead, which may harm the good execution of sensor node embedded programs such as the routing protocol.

5.4 Conclusion

In this chapter, we investigated the impact of LQEs on high layer protocols particularly on CTP tree routing protocol. The reason behind CTP selection is that CTP relies on link quality estimation to build the routes.
From both simulation and experiment results we found that FourBit outperforms existing LQEs. PRR and FLQE are the worst LQEs. FLQE estimator shows poorer performance when it is experienced in a real world scenario than in simulation. This observation opens two questions: Are we correctly exploiting FLQE estimates? Can we optimize FLQE implementation in order to get better performance?
In the following chapter we will try to answer to these two questions.

Chapter 6

FLQE Parameters Callibration and Optimization

6.1 Introduction

As outlined in the previous chapter, FLQE estimator shows poor performance when it was integrated at CTP protocol and experienced in a real world scenario. This result contradicts which it was concluded in the first evaluation methodology (LQEs statistical properties study). An eventual optimization of FLQE implementation may contribute to the enhancement of its performance.
The current chapter elaborates on our methodology to improve FLQE performance. For that, it starts by describing the eventual improvements which could be applied on original FLQE implementation. Then the experiment results got after these improvements appliance are presented and discussed.

6.2 FLQE: Potential Improvements

To improve the performance of FLQE in order to outperforms existing LQEs, we have applied numerous modifications to the original FLQE, which leaded to different new versions of FLQE. In this section, we describe the eventual improvements that we could apply to FLQE implementation:

- FLQE is a receiver side estimator, so given a sensor node **A**, the quality of its uplink (the link over which data traffic is sent) is computed at a receiver node **B**.
  ⇒ It will be more rigorous if **A** selects its next hop based on the quality computed by **B** sensor node.
  This first improvement will be called ***Accuracy I***. **The number 2** in figures 6.2 and 6.3 refers to the FLQE implementation where the sensor node selects its next hop based on the link quality estimates computed by its neighbors. If the sensor node selects its next hop based on the dowlink quality, FLQE versions will be libeled by **the number 1**.

- Receiver side estimators suffer from the lack of responsiveness as they detect the degradation of the link quality only if a reception is done. A link suffering from a lot of packet losses will take more time to detect such degradation.
  ⇒ Exploiting the retransmission count information may enhance the responsiveness of FLQE.
  This second improvement will be called ***Responsiveness I***. **The letter R** in figures 6.2 and 6.3 refers to the FLQE implementation where ***Responsiveness I*** is applied.

- SNR is a channel quality indicator that requires a huge memory overhead. LQI is another channel quality indicator requiring less memory overhead than SNR. LQI has been shown as important metric to understand and analyze channel behavior in WSNs [48, 49, 15].
  The following figure clearly shows how SNR based FLQE implementation **(S-FLQE)** is more memory greedy than LQI based FLQE implementation **(L-FLQE)**.

Figure 6.1: Memory consumption difference between S-FLQE and L-FLQE

⇒ may use LQI for channel quality assessment instead of SNR.
This third improvement will be called ***Channel I***. The letters S and L in figures 6.2 and 6.3 refer to the FLQE implementation based respectively on SNR or LQI.

By applying these improvements on the original FLQE implementation, we got the following versions as depicted in next figure.

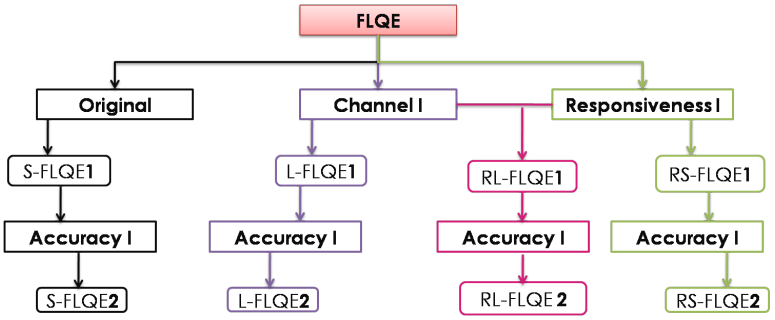


Figure 6.2: Different FLQE implementation versions after potential improvements appliance

- In routing protocols, the routing metric has great impact on their good operation. The routing metric in CTP is the sum of links forming the route qualities. In FLQE case, we need to find the adequate routing metric that leads to better performance. Logically, a good quality route is a route where the overall links forming it, have good quality (i.e The first link has good quality AND the second link has good quality AND .... AND the last link has good quality). Hence, if we traduce this relation mathematically, we found that the best routing metric should be the "product" of links forming the route qualities.
  Sensor nodes usually store link quality estimates in an integer format. Applying the product of link qualities forming long routes leads to either the usage of extremely long size variables or precision loss. For example, given a route formed by six links where each link has a quality equals to 10%. By applying the product operation to compute the route quality, we found that the results will be equal to 1000000. In this case, a sensor should manage a 32bit size variables however computation overheads are not advised in WSNs context. If the sensor node manage only 16bit size variables, 1000000 may will be trancated to 65536.
  Being inspired by the way that CTP uses to take into account PRR and WMEWMA estimates, we tested the use of the "Sum of Inverse of FLQE values" as a routing metric.

After the appliance of all the improvements, we found 12 FLQE versions. The following figure summarizes them.
We note by X-FLQE1 (X=S,L,RS or RL) the FLQE versions that uses the Sum of link quality estimates as routing metric and these estimates are those of the downlink. We note by X-FLQE2 the FLQE versions that uses the Sum of link quality estimates as routing metric and these estimates are those of the uplink. The uplink link quality estimates are sent by the neighbors of the sensor node. Moreover, we note by X-FLQE3 the FLQE versions that uses the Sum of the inverse of link quality estimates as routing metric and these estimates are those of the uplink.

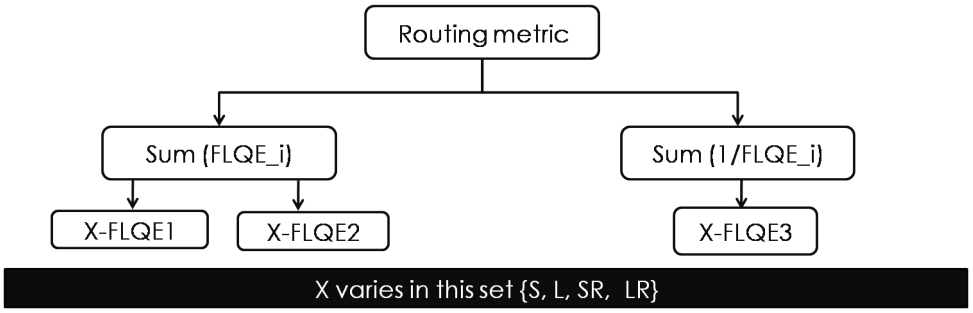


Figure 6.3: Different FLQE implementation versions resulting from the routing metric change

6.3 Experimental Study

6.3.1 Experiments Settings

6.3.1.1 Experimental Environment and Scenarios

FLQE optimization experiments are conducted in an indoor environment using MoteLab testbed as we did in the previous chapter. We also kept the same experiment settings summarized as follows:

Table 6.1: Second Impact on routing experiments settings

Data traffic	Constant (1 packet per 8 seconds)
Beacon traffic	Constant (1 beacon per second)
Selected Nodes	Depicted in Figure 5.4
Number of active nodes	13
Transmission power	-15dBm
Channel of transmission	26 (interference free channel)
Experiment times	between midnight and 8AM (MOTELAB time)
Experiment duration	40 min (Node start sending only beacon traffic in the first 10 min and then send data for 30 min)

6.3.2 Experiments Results

The performance evaluation metrics under consideration are Packet Delivery Ratio (PDR), Hop Count (HopC), Retransmission number (Rtx) and parent changes (PrtChgt). The following Figures outline the found results.

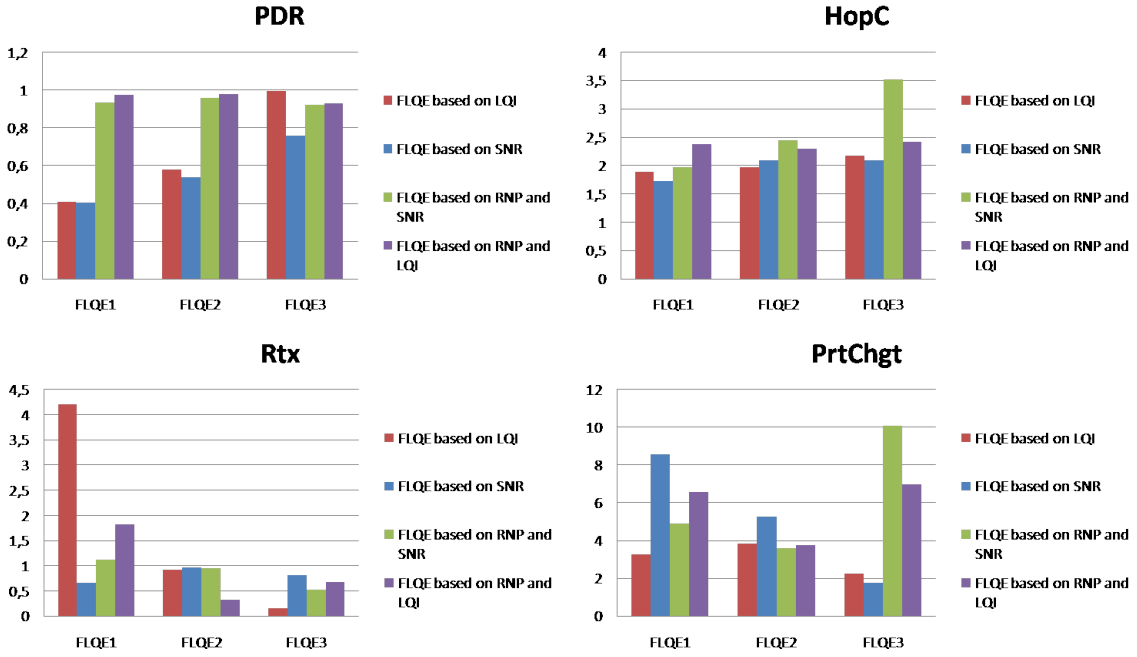
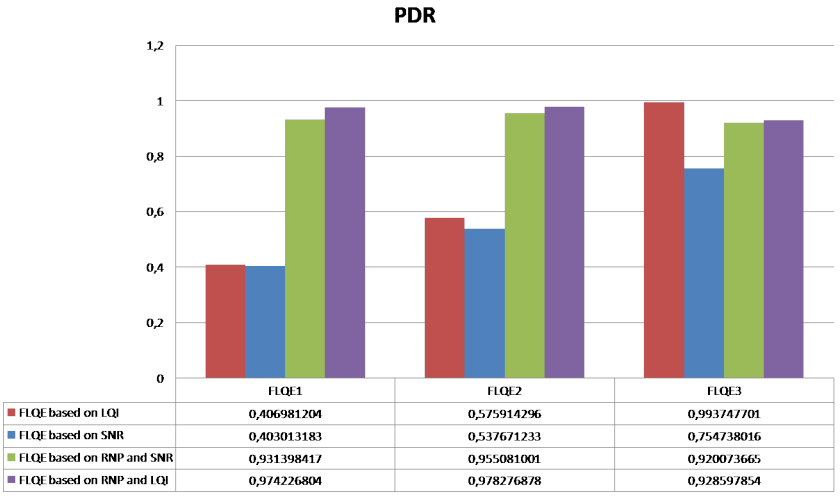
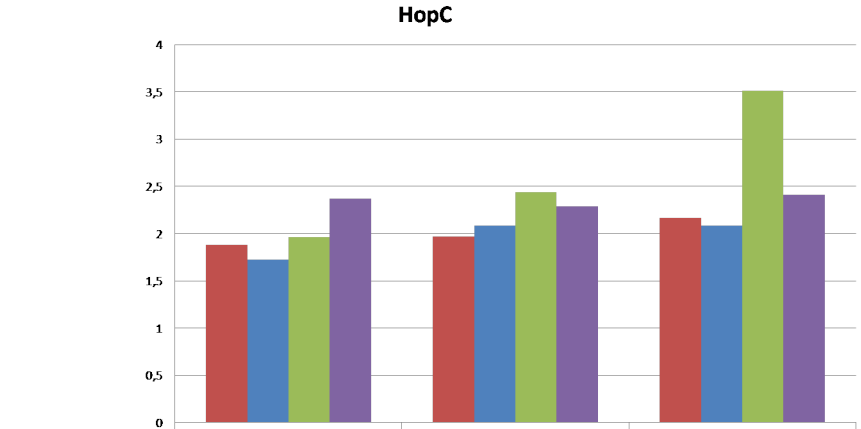


Figure 6.4: FLQE Optimization Results



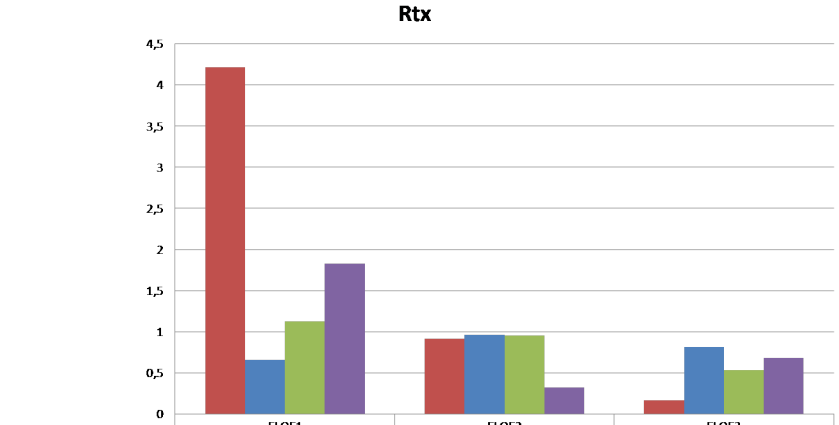
	FLQE1	FLQE2	FLQE3
FLQE based on LQI	0,406981204	0,575914296	0,993747701
FLQE based on SNR	0,403013183	0,537671233	0,754738016
FLQE based on RNP and SNR	0,931398417	0,955081001	0,920073665
FLQE based on RNP and LQI	0,974226804	0,978276878	0,928597854

Figure 6.5: FLQE PDR Optimization Results



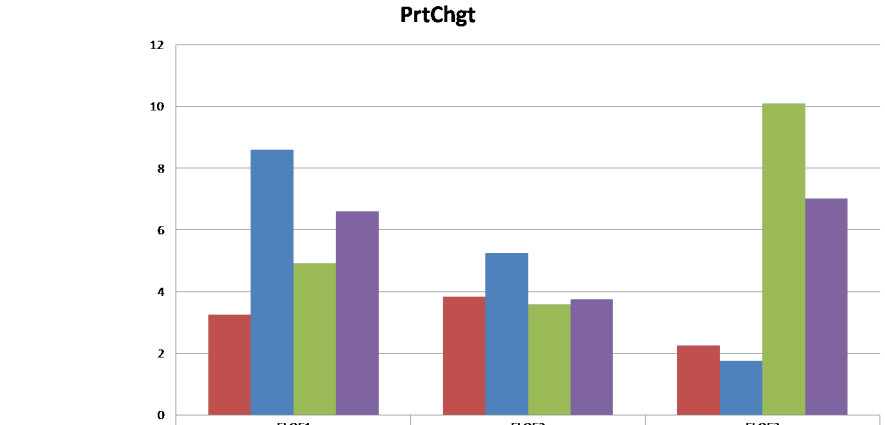
	FLQE1	FLQE2	FLQE3
FLQE based on LQI	1,883129123	1,967286722	2,166173205
FLQE based on SNR	1,724299065	2,084217976	2,087641556
FLQE based on RNP and SNR	1,96128423	2,436777178	3,514011209
FLQE based on RNP and LQI	2,37037037	2,289800527	2,409163347

Figure 6.6: FLQE HopC Optimization Results



	FLQE1	FLQE2	FLQE3
FLQE based on LQI	4,208294062	0,91533034	0,160251665
FLQE based on SNR	0,657009346	0,959660297	0,809453471
FLQE based on RNP and SNR	1,122757318	0,951811874	0,527221777
FLQE based on RNP and LQI	1,82388511	0,321791494	0,680876494

Figure 6.7: FLQE Rtx Optimization Results



	FLQE1	FLQE2	FLQE3
FLQE based on LQI	3,25	3,833333333	2,25
FLQE based on SNR	8,583333333	5,25	1,75
FLQE based on RNP and SNR	4,916666667	3,583333333	10,08333333
FLQE based on RNP and LQI	6,583333333	3,75	7

Figure 6.8: FLQE PrtChgt Optimization Results

6.3.3 Discussion

From the above figures we observe that FLQE performance may be improved, by the tuning of some implementation details.
For example considering the correct link direction in parent selection contributes to the increasing of the PDR, the decreasing of number of packet retransmissions and the increasing of the network stability by the decreasing of the number of parent changes. To consider the correct link direction, a sensor node should receive FLQE estimates from its neighbors.

First hint: FLQE estimates are better exploited if the routing protocol considers the quality of the link over which the data is sent in its decisions.

Furthermore, it is clear from the above figures that FLQE based on LQI metric is generally better than FLQE based on SNR. LQI usage has the advantage of reducing memory overhead and computation overhead caused by SNR usage. LQI metric is directly extracted from the chipcon hardware, however SNR requires several computation operations to be extracted. We conclude that memory and computation overhead impacts FLQE performance.

Second hint: It is recommended to use LQI as channel quality indicator instead of SNR at it less memory greedy.

Another result that can be observed is that the routing metric definitely plays a potential role in the rigorous exploitation of FLQE estimates. As we can see XR-FLQEi versions (where X=L or S and i=1 or 2 or 3) leads to better performance when using the sum of FLQE as routing metric rather than using the sum of 1/FLQE.
In opposite X-FLQEi versions (where X=L or S and i=1 or 2 or 3) leads to better

performance when using the sum of 1/FLQE as routing metric rather than using the sum of FLQE.

Third hint: It is recommended to judiciously select the routing metric in order to the better exploitation of FLQE estimates.

Based on the improvement results, the best FLQE version is L-FLQE3. By comparing L-FLQE3 performance with those of existing LQEs, we found that L-FLQE3 outperforms existing LQEs. The following Figure shows that L-FLQE3 has the highest packet delivery ratio, the minimized number of packet retransmissions and its usage leads to the most stable network.

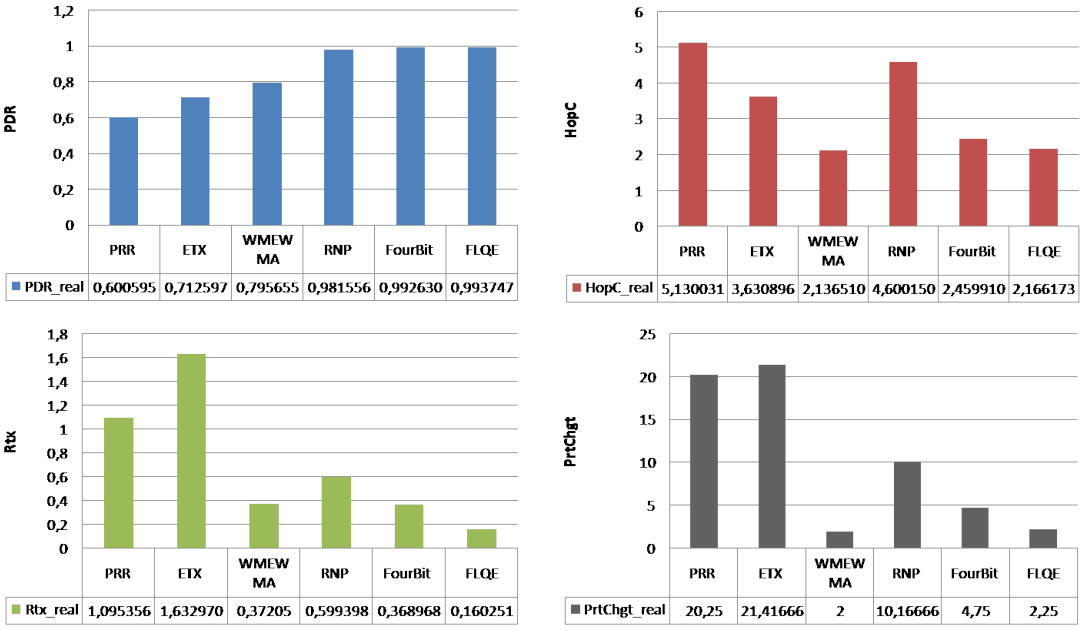


Figure 6.9: The FLQE out-performance of existing LQEs after improvements appliance

6.4 Conclusion

In this chapter, we tried to improve FLQE performance by the calibration of its parameters. These improvements are applied to some FLQE properties for example: using LQI instead of SNR or using RNP instead of SPRR. Further, these improvements are also applied to the way the CTP protocol exploits FLQE estimates to build the routes. We finally converged to a version that outperforms existing LQEs.

Part III

Conclusion and Future Work

General conclusion and future work

In this Master Thesis we conducted a performance evaluation study of a set of most reported LQEs and tried to optimize FLQE, a recently proposed LQE.
Our experimental methodology is split in two phases: the first one consists in studying the statistical properties of LQEs and the second one consists in investigating their impact on higher layer protocols particularly routing protocols. The selected routing protocol was CTP as it relies on link quality estimation to build the routes.
For the first phase of the experimental study, we were invited to implement a testbed called RadiaLE aiming at automating the performance evaluation of different LQEs by analyzing their statistical properties.
In the second study, we performed our experiments using simulation and real experimentation. Simulation provides a rapid prototyping solution but it lacks accuracy as it usually relies on simplified and stochastic models. In contrast, real experimentation provides much more accurate and trust-able results. Simulation experimentations were done using TOSSIM2 simulator and Real experimentations were done using extended RadiaLE combined to MOTELAB testbed in order to perform large scale experiments. Our prior results shows that FourBit is the most powerful estimator in terms of enabling energy saving and maximizing packet delivery. However, by applying some optimization on FLQE implementation, FLQE succeeded to outperform existing LQEs. Roughly the best estimators are L-FLQE3, FourBit, WMEWMA and ETX.
The following table summarizes the overall performance of all studied LQEs.

High → Low

Statistical properties						
Over-Estimation	PRR	WMEWMA	ETX	FLQE	FourBit	RNP
Stability	FLQE	WMEWMA	PRR	FourBit	RNP	ETX
Impact on CTP routing (real experimentations results)						
Delivery	L-FLQE3	FourBit	RNP	WMEWMA	ETX	PRR
Retransmissions (energy consumption)	ETX	PRR	RNP	WMEWMA	FourBit	L-FLQE3
Paths lengths	PRR	RNP	ETX	Fourbit	L-FLQE3	WMEMWA
Network stability	WMEWMA	L-FLQE3	FourBit	RNP	PRR	ETX

Figure 6.10: Different LQEs performances

In our study, we investigated the impact of LQEs on routing context in a single scenario, we plan in our future work extending this study in other scenarios by changing for example the environment of deployment and by increasing the network density. In addition, we plan to investigate LQEs impact on other contexts such mobility managing.

Part IV

Publications List

Publications

Maissa Ben Jamaa, Nouha Baccour, Anis Koubaa, Denis do Rosarioy, Mario Alves, Leandro Becker, Habib Youssef, and Mohamed Jmaiel. Demo Abstract: A TestBed for the evaluation of Link Quality Estimators in WSNs, The First International School on Cyber-Physical and Sensor Networks (SensorNets 2009), Monastir, Tunisia, December 17-22 2009. (Best Demo Award).

Nouha Baccour, Maissa Ben Jamaa, Denis do Rosario, Anis Koubaa, Habib Youssef, Mario Alves and Leandro B. Becker. A TestBed for the Evaluation of Link Quality Estimators in Wireless Sensor Networks, The ACS/IEEE Workshop Future Trends on Ad-hoc and Sensor Networks (FT-ASN 2010), Tunisia, May 16-19, 2010.

Nouha Baccour, Maissa Ben Jamaa, Denis do Rosario, Anis Koubaa, Mario Alves, Leandro B. Becker, Habib Youssef, and Hossein Fotouhi. Demo Abstract: RadiaLE: a framework for benchmarking link quality estimators, The 7th European Conference on Wireless Sensor Networks (EWSN 2010), Coimbra, Portugal, February 17-19, 2010.

Nouha Baccour, Anis Koubaa, Habib Youssef, Maissa Ben Jamaa, Denis do Rosario, Mario Alves and Leandro Becker. F-LQE: A Fuzzy Link Quality Estimator for Wireless Sensor Networks, The 7th European Conference on Wireless Sensor Networks (EWSN 2010), Coimbra, Portugal, February 17-19, 2010.

Part V

Bibliography

Bibliography

[1] Crossbow: http://www.xbow.com.

[2] Ctp:collection tree protocol: http://www.tinyos.net/tinyos-2.x/doc/html/tep123.html.

[3] Dsnanalyzer: Backend for the deployment support network.

[4] Motelab: http://motelab.eecs.harvard.edu.

[5] thalesresearch: http://www.thalesresearch.com/.

[6] Why wirelesshart: http://www.hartcomm.net/protocol/training/resources/wihartresources/v

[7] Chipcon cc2420: Data sheet, 2009.

[8] Daniel Aguayo, John Bicket, Sanjit Biswas, Glenn Judd, and Robert Morris. Link-level measurements from an 802.11b mesh network. In *In SIGCOMM*, pages 121–132, 2004.

[9] Nouha Baccour, Maissa Ben Jamaa, Denis do Rosario, Anis Koubaa, Mario Alves, Leandro B. Becker, Habib Youssef, and Hossein Fotouhi. Demo abstract: Radiale: a framework for benchmarking link quality estimators.

[10] Nouha Baccour, Anis Koubaa, Maissa Ben Jamaa, Habib Youssef, Marco Zuniga, and Mario Alves. A comparative simulation study of link quality estimators in wireless sensor networks. In *17th IEEE/ACM International Symposium on Modelling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS'09)*, 2009.

[11] Nouha Baccour, Anis Koubaa, Habib Youssef, and Mario Alves. Simulation analysis and validation of a fuzzy link quality estimator. In *SensorNets '09: The First International School on Cyber- physical and Sensor Networks*, Monastir, Tunisia.

[12] Nouha Baccour, Anis Koubaa, Habib Youssef, Maissa Ben Jamaa, Denis do Rosário, , Mario Alves, and Leandro B.Becker. F-LQE: A Fuzzy Link Quality Estimator for Wireless Sensor Networks. In *7th European Conference on Wireless Sensor Networks (EWSN 2010)*, LNCS 5970, pages 240–255, Coimbra, Portugal, February 2010. Springer.

[13] Jan Beutel, Matthias Dyer, Mustafa Yücel, and Lothar Thiele. Demo abstract: Development and test with the deployment-support network.

[14] Sanjit Biswas and Robert Morris. Exor: Opportunistic multi-hop routing for wireless networks. In *in SIGCOMM*, pages 133–144, 2005.

[15] Carlo Alberto Boano, Thiemo Voigt, Adam Dunkels, Fredrik Osterlind, Nicolas Tsiftes, Luca Mottola, and Pablo Suarez. Poster abstract: Exploiting the lqi variance for rapid channel quality assessment. In *IPSN '09: Proceedings of the 2009 International Conference on Information Processing in Sensor Networks*, pages 369–370, Washington, DC, USA, 2009. IEEE Computer Society.

[16] Azzedine Boukerche. *Algorithms and Protocols for Wireless Sensor Networks*. Wiley-IEEE Press, 2008.

[17] Anmol Sheth James Carlson Sash Bhatti Hui Dai Jing Deng Richard Han Brian Shucker, Jeff Rose. *handbook of sensor networks algorithms and architectures*, chapter 6, page 174. A JOHN WILEY and SONS, INC., PUBLICATION, 2005.

[18] Alberto Cerpa, Naim Busek, and Deborah Estrin. Scale: A tool for simple connectivity assessment in lossy environments. Technical report, 2003.

[19] Alberto Cerpa, Jennifer L. Wong, Miodrag Potkonjak, and Deborah Estrin. Temporal properties of low power wireless links: modeling and implications on multi-hop routing. In *MobiHoc '05: Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, pages 414–425, New York, NY, USA, 2005. ACM.

[20] Brent N. Chun, Philip Buonadonna, Alvin Auyoung, Chaki Ng, David C. Parkes, Jeffrey Shneidman, Alex C. Snoeren, and Amin Vahdat. Mirage: A microeconomic resource allocation system for sensornet testbeds. In *In Proceedings of the 2nd IEEE Workshop on Embedded Networked Sensors*, 2005.

[21] Douglas S. J. De Couto, Daniel Aguayo, John Bicket, and Robert Morris. A high-throughput path metric for multi-hop wireless routing. In *MobiCom '03: Proceedings of the 9th annual international conference on Mobile computing and networking*, pages 134–146, New York, NY, USA, 2003. ACM.

[22] Qiao Daji and Choi Sunghyun.

[23] Douglas S. J. De Couto, Daniel Aguayo, Benjamin A. Chambers, and Robert Morris. Performance of multihop wireless networks: shortest path is not enough. *SIGCOMM Comput. Commun. Rev.*, 33(1):83–88, 2003.

[24] Callaway Jr Edgar H. *handbook of sensor networks algorithms and architectures*, chapter 8, page 250. A JOHN WILEY and SONS, INC., PUBLICATION, 2005.

[25] J. Elson, S. Bien, N. Busek, V. Bychkovskiy, A. Cerpa, D. Ganesan, L. Girod, B. Greenstein, T. Schoellhammer, T. Stathopoulos, and D. Estrin. Emstar: An environment for developing wireless embedded systems software, 2003.

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

[27] Deepak Ganesan, Bhaskar Krishnamachari, Alec Woo, David Culler, Deborah Estrin, and Stephen Wicker. Complex behavior at scale: An experimental study of low-power wireless sensor networks, 2002.

[28] Ana-Beln Garca-Hernando, Jos-Fernn Martnez-Ortega, Juan-Manuel Lpez-Navarro, Aggeliki Prayati, and Luis Redondo-Lpez. *Problem Solving for Wireless Sensor Networks*. Springer Publishing Company, Incorporated, 2008.

[29] Werner-Allen Geoffrey, Swieskowski Patrick, and Welsh Matt. Motelab: A wireless sensor network testbed. In *IPSN '05: Proceedings of Information Processing in Sensor Networks*, pages 483–488, Piscataway, NJ, USA, 2005. IEEE Press.

[30] Omprakash Gnawali, Rodrigo Fonseca, Kyle Jamieson, David Moss, and Philip Levis. Collection tree protocol. In *SenSys '09: Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, pages 1–14, New York, NY, USA, 2009. ACM.

[31] Beutel Jan, Dyer Matthias, Yucel Mustafa, and Lothar Thiele. Demo abstract: Development and test with the deployment-support network. In *EWSN '07: Proceedings of the 4th European Conference on Wireless Sensor Networks*, pages 195–211. Springer,, 2007.

[32] C.E. Koksal and H. Balakrishnan. Quality-aware routing in time-varying wireless networks. *IEEE Journal on Selected Areas of Communication Special Issue on Multi-Hop Wireless Mesh Networks*, pages 1984–1994, 2006.

[33] David Kotz, Calvin Newport, and Chip Elliott. The mistaken axioms of wirelessnetwork research. Technical report, 2003.

[34] D. Lal, A. Manjeshwar, F. Herrmann, E. Uysal-Biyikoglu, and A. Keshavarzian. Measurement and characterization of link quality metrics in energy constrained wireless sensor networks. *IEEE Global Telecommunications Conference (IEEE GLOBECOM)*, 2003.

[35] Philip Levis, Nelson Lee, Matt Welsh, and David Culler. Tossim: accurate and scalable simulation of entire tinyos applications. In *SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 126–137, New York, NY, USA, 2003. ACM.

[36] Philip Levis, Neil Patel, David Culler, and Scott Shenker. Trickle: A self-regulating algorithm for code propagation and maintenance in wireless sensor networks. In *In Proceedings of the First USENIX/ACM Symposium on Networked Systems Design and Implementation (NSDI*, pages 15–28, 2004.

[37] Y. Li, J. Chen, R. Lin, and Z. Wang. A reliable routing protocol design for wireless sensor networks. *IEEE International Conference on Mobile Adhoc and Sensor Systems Conference*, 2005.

[38] Linlan Liu, Jun Li, Jian Shu, Zhenhua Wu, and Yebin Chen. Cci-based link quality estimation mechanism for wireless sensor networks under perceive packet loss. *Journal of Software*, 5(4), 2010.

[39] P M. Wightman M A. Labrador. *Topology Control in Wireless Sensor Networks*. Springer, 2009.

[40] Sudip Misra, Isaac Woungang, and Subhas Chandra Misra. *Guide to Wireless Sensor Networks*. Springer Publishing Company, Incorporated, 2009.

[41] Luca Mottola and Gian Pietro Picco. Programming wireless sensor networks: Fundamental concepts and state-of-the-art.

[42] Heemin Park, Weiping Liao, King Ho Tam, Mani B. Srivastava, and Lei He. A unified network and node level simulation framework for wireless sensor networks.

[43] Adi Mallikarjuna V. Reddy, A.V.U. Phani Kumar, D. Janakiram, and G. Ashok Kumar. Wireless sensor network operating systems: a survey. *Int. J. Sen. Netw.*, 5(4):236–255, 2009.

[44] Kazem Sohraby, Daniel Minoli, and Taieb Znati. *Wireless Sensor Networks: Technology, Protocols, and Applications*. Wiley-Interscience, 2007.

[45] T. T Soong. *Fundamentals of probability and statistics for engineers*. Wiley, 2004.

[46] Michael R. Souryal, Johannes Geissbuehler, Leonard E. Miller, and Nader Moayeri. Real-time deployment of multihop relays for range extension. In *MobiSys '07: Proceedings of the 5th international conference on Mobile systems, applications and services*, pages 85–98, New York, NY, USA, 2007. ACM.

[47] K. Srinivasan, M A. Kazandjieva, M. Jain, E. Kim, and P. Levis. Swat: Enabling wireless network measurements. *Demonstration at the Sixth ACM Conference on Embedded Networked Sensor Systems*, 2008.

[48] Kannan Srinivasan, Prabal Dutta, Arsalan Tavakoli, and Philip Levis. Understanding the causes of packet delivery success and failure in dense wireless sensor networks. Technical report, In Technical report SING-06-00, 2006.

[49] Kannan Srinivasan and Philip Levis. Rssi is under appreciated. In *In Proceedings of the Third Workshop on Embedded Networked Sensors (EmNets*, 2006.

[50] Lei Tang, Kuang-Ching Wang, Yong Huang, and Fangming Gu. Channel characterization and link quality assessment of ieee 802.15.4-compliant radio for factory environments. *IEEE Trans. Industrial Informatics*, 3(2):99–110, 2007.

[51] Handziski Vlado, Kopke Andreas, Willig Andreas, and Wolisz Adam. Twist: a scalable and reconfigurable testbed for wireless indoor experiments with sensor networks. In *REALMAN '06: Proceedings of the 2nd international workshop on Multi-hop ad hoc networks: from theory to reality*, pages 63–70, New York, NY, USA, 2006. ACM.

[52] Andreas Wapf and Michael R. Souryal. Measuring indoor mobile wireless link quality. In *ICC*, pages 1–6. IEEE, 2009.

[53] Alec Woo and David Culler. Evaluation of efficient link reliability estimators for low-power wireless networks. Technical Report UCB/CSD-03-1270, EECS Department, University of California, Berkeley, 2003.

[54] Alec Woo, Terence Tong, and David Culler. Taming the underlying challenges of reliable multihop routing in sensor networks. In *SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 14–27, New York, NY, USA, 2003. ACM.

[55] Jennifer Yick, Biswanath Mukherjee, and Dipak Ghosal. Wireless sensor network survey. *Comput. Netw.*, 52(12):2292–2330, 2008.

[56] Jerry Zhao and Ramesh Govindan. Understanding packet delivery performance in dense wireless sensor networks. In *SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 1–13, New York, NY, USA, 2003. ACM.

[57] Gang Zhou, Tian He, Sudha Krishnamurthy, and John A. Stankovic. Impact of radio irregularity on wireless sensor networks. In *MobiSys '04: Proceedings of the 2nd international conference on Mobile systems, applications, and services*, pages 125–138, New York, NY, USA, 2004. ACM.

[58] Gang Zhou, Tian He, Sudha Krishnamurthy, and John A. Stankovic. Models and solutions for radio irregularity in wireless sensor networks. *ACM Trans. Sen. Netw.*, 2(2):221–262, 2006.

[59] M. Zuniga and B. Krishnamachari. Analyzing the transitional region in low power wireless links. *IEEE SECON: First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks*, pages 517–526, 2004.

[60] M. Zuniga and B. Krishnamachari. An analysis of unreliability and asymmetry in low-power wireless links. *ACM Trans. Sen. Netw.*, 3(2):7, 2007.