Master of Science

LINK QUALITY ESTIMATION FOR INDUSTRIAL WIRELESS SENSOR NETWORKS USING BURSTINESS DISTRIBUTION METRIC

The Graduate School of the University of Ulsan

Department of Electrical and Computer Engineering

NGOC HUY NGUYEN

LINK QUALITY ESTIMATION FOR INDUSTRIAL WIRELESS SENSOR NETWORKS USING BURSTINESS DISTRIBUTION METRIC

Supervisor: Professor MYUNG KYUN KIM

A Thesis

Submitted to

The Graduate School of the University of Ulsan

In partial Fulfillment of the Requirements for the Degree of

Master of Science

By

NGOC HUY NGUYEN

Department of Electrical and Computer Engineering

University of Ulsan, Korea

NOVEMBER 2020

LINK QUALITY ESTIMATION FOR INDUSTRIAL WIRELESS SENSOR NETWORKS USING BURSTINESS DISTRIBUTION METRIC

This certifies that this Thesis of Ngoc Huy Nguyen is approved by:

Committee Chair: Professor Seok Hoon Yoon

윤석훈

Committee Member: Professor Insoo Koo

Insoo Koo 구인

Committee Member: Professor Myung Kyun Kim

김명균

Department of Electrical and Computer Engineering

University of Ulsan

November 2020

DEDICATED TO

My parent, my brother, and the people who have helped me.

ABSTRACT

Industrial wireless sensor network applications are requiring low-power operations, deterministic communications, and end-to-end reliability. However, it is very difficult to achieve this goal because of link burstiness and interference. In this study, we propose a novel link quality estimation mechanism named burstiness distribution metric, which uses the distribution of burstiness in the link to deal with the variations of the wireless link. First, we estimate the link quality by counting the number of consecutive loss packet happenings in each bi-direction link at the receiver node. Based on that, we created a burstiness distribution list and estimate the number of transmissions. The node uses the burstiness distribution metric to choose the best route for the routing and calculate the number of transmissions to reach the reliability target in the scheduling algorithm. Our simulation in Cooja Simulator – the simulator of Contiki showed that our proposal can be used as the input metric to choose the minimal number of transmission packet on each link for the routing protocol and calculate the number of transmission in scheduling to achieve energy efficiency and reliability target in the industrial wireless sensor network.

TABLE OF CONTENTS

ACKNOWLEDGEMENT

The thesis presents the research that I worked at the Computer Network Lab at the University of Ulsan. Professor Myung Kyun Kim, the supervisor for my MSc thesis work, his support for this research is very important. I would like to express my sincere gratitude for his support, guidance, and encouragement for my work at the University of Ulsan. I also would like to thank all my colleagues in the lab, with whom I had a lot of fascinating and beneficial conversations, with whom I published my research articles as well as conference proceedings.

Thanks to BK21 Plus, which offered me funding for research and an important and promising research topic. I had the opportunity to work with many excellent researchers. My skills improved during project group discussions.

Finally, I would like to share a lot of my achievements with my family in Vietnam. Particularly for my parents, who have always supported me. To my girlfriend, who are the ones I want to share all my happiness with. To my friends at Ulsan University who gave me a colorful life during my studies in Ulsan.

Republic of Korea, Ulsan, NOVEMBER 2020

NGOC HUY NGUYEN

LIST OF FIGURES

LIST OF TABLES

Chapter 1: INTRODUCTION

1.1 Overview.

In several applications, wireless sensor networks are deployed, including monitoring applications, radiation checks, leakage detection, process control applications. Compared to the use of wired networks, they offer many benefits such as low cost and high versatility [1]. The data is obtained by the sensors and then sent to the sink (based station). Especially, in the industrial application, have extreme latency requirements, energy efficiency, and reliability. The reliability of data transmission is decided by the link quality between every two sensor nodes. However, the link quality can be effect by a lot of causes such as internal interference, external-interference, self-interference, and burstiness link. The interference link is the physical property of the environment that interferes with the transmission of packets between various connections. Burstiness link is another physical property that implies transmissions of the wireless link do not contain a fixed failure probability and have periods of continuous transmission errors. Because of these non-deterministic wireless links, it's difficult to supply reliability for packet transmit over the wireless networks. Although the causes will make the link quality between two sensor nodes can be changed. However, estimating the quality of the link is not easy, since the sensor node is limited to the energy consumption and the complexity computational. Therefore a link quality measurement that is low computational complexity and low energy consumption for wireless sensor networks is necessary to study.

The routing protocol is crucially important to reduce latency, energy efficiency, and reliability. There are some studies to design a routing protocol in low-power multi-hop networks such as RPL [2], LOADng [3], LRP [4] to seek effective routes. The above protocols must keep their footprint minimal and choose the right path for data traffic to the route. This initial link quality calculation must therefore be careful and precise to choose the best path to sink node to achieve a low-cost reliability objective. The authors in [5] and [6] try to solve the Burstiness link problem by proposed some mechanisms. Srinivasan et al. [5] proposed a metric called β, showing that an inter-packet delay of 500 ms can avoid the burst of the link. While 500ms of delay is a drawback for the network in the industrial field, we don't have to wait the time to prevent packet loss when a link burst occurs. Munir et al. [6] proposed a maximum burst length metric (denoted Bmax), estimated by long experiment data. In accordance with this scheduling algorithm, the routing algorithm called least-burst-route is used to minimize the total of worst-case burst lengths over all links in the route. However, the algorithm to calculate the Bmax value is gathered from the experiment measure link with the long sequence of data-trace before settings the network. This makes it impossible to implement this algorithm to the ad hoc network.

1.2 Contribution of the Thesis.

The main purpose of this study is to evaluate link quality estimators and we proposed a new metric called the Burstiness Distribution Metric, is calculated by counting the number of consecutive loss packet then makes the distribution of burstiness of each link. Our proposal can be easy to apply to the ad-hoc network. Moreover, we apply the Burstiness Distribution Metric to calculate the number of transmissions for the Path Collision-aware Least Laxity First (PCLLF) [7] scheduling algorithm to reach a reliable target for the specific application.

1.3 Structure of the Thesis.

In this part, the structure of the thesis is described. The ideas behind wireless sensor networks are presented in Chapter 2. Then the various networking protocols of WSNs such as IEEE 802.15.4 standards and its amendment IEEE 802.15.4e are discussed further. In addition, Chapter 2 introduces the Contiki OS and its updated Contiki-NG version (Next Generation), as well as the network stacks currently supported. Finally, a summary of the simulator software Cooja is accustomed to evaluate this work is presented.

The discussed in Chapter 3 mentioning the related work that has been done on the link quality estimators. Furthermore, a summary survey for the measure link quality topic is discussed in this chapter. Chapter 4 explains the proposed algorithm and the implementations in the Contiki-NG. Chapter 5 describes the configurations of the simulation work in the Cooja Simulator to evaluate the program implementations of the proposed link quality estimator. Finally, the simulation results are shown and the key conclusions of this review are discussed. In Chapter 6, the conclusions of the thesis are presented.

Chapter 2: BACKGROUND

The principles behind wireless sensor networks, their key functions, and the IEEE 802.15.4 specification are discussed in this chapter. In addition, the IEEE 802.15. 4e amendment with the Time Slotted Channel Hopping (TSCH) is explained. By maintaining nodes time-synchronized at the MAC layer, TSCH is a framework that helps to ensure network stability. Finally, an overview of the Contiki, Contiki-NG, Cooja Simulator is briefly presented.

2.1 Wireless sensor networks (WSNs).

Wireless sensor networks are a board system that typically consists of several cheap, low-power with connectivity capability and more functional sensor nodes to sense the physical object and execute actions. They are equipped with a transceiver module, power source, memory, sensors, and microprocessors, and therefore not only can sense the physical environment but also process and send data. The standard sensor node architecture is shown in Figure 1. Communication takes place over a wireless medium and together they can monitor. WSN applications with the above specifications may appear very limited to high-end applications such as radiation, ground mines, and warning systems for nuclear threats. However, the number of applications and areas is very enormous and continues to grow along with recent technological developments such as environmental applications, home automation, health care, control and tracking systems,...

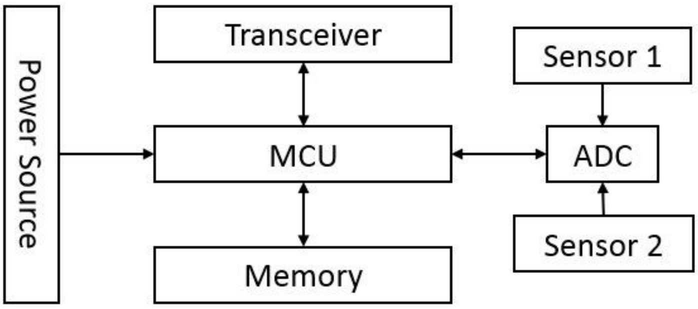


Figure 1. The standard sensor node architecture.

Wireless has great advantages over traditional wired networks, which get rids of the cables thus lowering the cost, simplifying for design network, installation, and maintenance. Moreover, they can be placed in a harsh environment where wired solutions are impossible [8].

2.2 IEEE 802.15.4 standard.

Several protocols and standards have been developed over the past few decades to establish connectivity and network functionality for wireless sensor networks. The IEEE 802.15.4 [9] standard aimed at allowing applications within wireless personal area networks with comparatively low throughput and latency requirements.

IEEE 802.15 Task Group 4 released the original standard edition in 2003. Subsequently, several updates were released in 2006 and 2011 [10]. In addition, a significant range of academic and industrial sensor nodes for the study is funded to meet the growing demand for low-power and short-range networks.

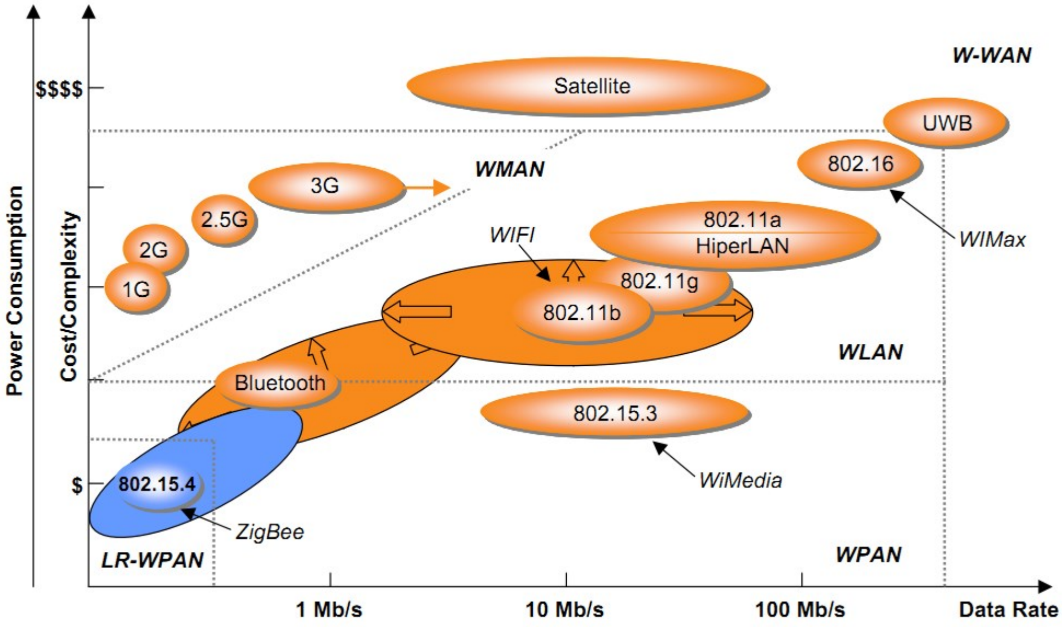


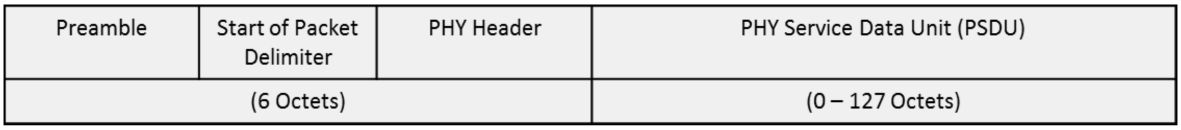
Figure 2. The key features comparison of wireless standards.

The standard defines the low-power physical (PHY) layer and medium access control (MAC) layer of the network stack. The IEEE 802.15.4 standard compare with other standards as shown in Figure 2 [11].

Frequency	Region	Channel	Modulation	Bit rate
868-868.6 MHz	EU, Japan	1	BPSK	20 kbps
902-928 MHz	USA	10 (2003 rel) 30 (2006 rel)	BPSK	40 kbps
2400 MHz	Global	16	O-QPSK	250 kbps

Table 1. IEEE 802.15.4 standard frequency, modulation, and data rate.

The data rates of IEEE 802.15.4 standard approximately 20, 40, and 250 kbps. Depending on the specific application, we can use Radio Frequency (RF) bands and digital modulation mechanisms to comply with our data rate expectation. In addition, both star and peer-to-peer network topologies are supported by the standard. The PAN Coordinator performs the role of a border router in the star topology where the IEEE 802.15.4 sensor nodes can connect with the internet world via it. With peer-to-peer networks, sensor nodes will connect with all other sensor nodes in their radio range, and the sensor data packet will be transmitted to the PAN Coordinator depending on the routing protocol. Different mechanisms, such as Carrier-Sense Multiple Access with Collision Avoidance (CSMA/CA), are used for the MAC layer to access the channel. In addition, there are four kinds of MAC frames in the protocol: a data frame, a beacon frame, an acknowledgment frame, and a MAC command frame. The IEEE 802.15.4 packet structure has four important fields, which can be seen in Figure 3: a four-octet (32-bit) preamble for synchronization, a packet ID operating as the start of the packet delimiter (8-bit), a PHY header containing the length (8-bit) of the Physical Layer Service Data Unit (PSDU), and finally a PSDU field containing the actual data.



Preamble	Start of Packet Delimiter	PHY Header	PHY Service Data Unit (PSDU)
(6 Octets)			(0 – 127 Octets)

Figure 3. The IEEE 802.15.4 packet structure.

Finally, the IEEE 802.15.4 includes two network device groups: full-function devices (FFDs), reduced function devices (RFDs). FFDs are devices that have full functionality levels. They can be used to transmit and receive data but can route data from other nodes as well. However, RFDs are devices that have a reduced functionality level. They are normally an end node, and may generally be sensors or switches. RFDs will only talk to FFDs and they do not have the routing capability. These RFDs are also referred to as child devices as they require other parent devices to connect with. The special FFDs node that manages the IEEE 802.15.4 network is the coordinator. It also sets up the IEEE 802.15.4 network in addition to the usual FFDs functions and operates as the network coordinator or manager.

2.3 IEEE 802.15.4e TSCH

Although the IEEE 802.15.4 protocol was first released in 2003, additional updates and enhancements were made in 2006 and 2011 to cover various aspects of the specification [10]. In 2012, a new update to the 802.15.4-2006/2011 revisions, IEEE 802.15.4e [12], was released to strengthen previous MAC protocols and networking modes while meeting the evolving criteria of implementations in manufacturing environments for time-critical requirements. Several general functional enhancements and the following new MAC behavior modes were added, such as Asynchronous Multi-Channel Adaptation (AMCA), Deterministic and Synchronous Multi-channel Extension (DSME), Low Latency Deterministic Network (LLDN), and Time-Slotted Channel Hopping (TSCH).

The TSCH [13] is one of the leading MAC behavior modes for IEEE 802.15.4e built to support the industrial and automotive industries. Time-slotted access and multichannel with channel hopping are integrated into the TSCH mode, appropriate for multi-hop networks. One of the key features was the TSCH mitigates two of the main causes of link failure external interference and multi-path fading. External interference occurs when simultaneous transmissions operate on the same frequency band will collide and introduce packet loss. Multi-path fading is the radio waves that will take many different paths to the destination, which will then receive multiple signals. Depending on several factors, this will create a destructive effect that results in packet loss [14]. In addition, several network topologies, such as star, tree, and mesh, are supported by TSCH. TSCH has dedicated and shared links. Shared links are slots that can be allocated to more than one transmitter, allowing simultaneous access by several nodes depending on the collision process. The dedicated link can prevent collision since it allows only one communication cell between one transmitter and one receiver at the same time. TSCH's key priorities are to support larger network capability, low-power, low latency, and high reliability. Multichannel TSCH supports channel hopping based on up to 16 different channels. The channel is defined by the offset of the channel, and the integer value has a range of 0 to 15. At the same time slot, more nodes will communicate using separate channels that are defined by their channel offset. Thus we can allocate up to 16 dedicated links to a time slot.

The TSCH MAC does not change the physical layer configuration entirely. It can run on any hardware compatible with the IEEE 802.15.4 standard.

2.3.1 Slotframe structure.

In TSCH MAC, time is divided into timeslots. The slot frame structure consists of several timeslots of 10ms duration typically. All sensor nodes are tightly synchronized. Each pair of sensor nodes can exchange a maximum data frame size and receive an acknowledgment (ACK) during this time slot.

When the ACK is not received, the frame retransmission is delayed until the next time slot assigned is reached. The total number of slots that have elapsed is the absolute slot number (ASN), a timeslot counter increases by 1 each timeslot, and it is used to synchronize nodes in a WSN. When joining nodes has received ASN, they use this to calculate which channel to communicate on. All scheduling cells have a specific slot offset and channel offset. For example, if node A has a transmitting slot to node B on channel offset 1, node B will have a receiving slot for node A on channel offset 1. The channel offset is then transformed to a frequency using the function below:

$$f = F[(ASN + channelOffset)\%Nch] \quad (1)$$

where the Nch is the number of channels, F is the lookup table function of available channels.

2.3.2 Channel hopping.

Equation (1) presents the channel hopping function in TSCH, where several frequencies can be returned at various time slots for the same link. The channel hopping system ensures that within their designated timeslot, all available channels are used for a single connection, thus reducing the effects of interruption and multi-path fading and thereby increasing the efficiency of the network.

2.3.3 TSCH scheduling.

The IEEE 802.15.4e standard [12] does not determine how to build, optimize, and manage the timeslot schedule. It just shows how the MAC layer will perform the schedule. Some several scheduling strategies were proposed such as centralized scheduling, distributed scheduling, and autonomous scheduling. The Central Entity is responsible for creating and optimizing the network schedule with unified scheduling and uploading the schedule for sensor nodes. However, the distributed scheduling doesn’t have Central Entity for computes scheduling, it takes decisions locally based on the control packet they are negotiating with their neighbors. With the autonomous scheduling, all the schedule is generated by the sensor node without any control packet transfer with their neighbors. Figure 4 shows an example of the TSCH schedule.

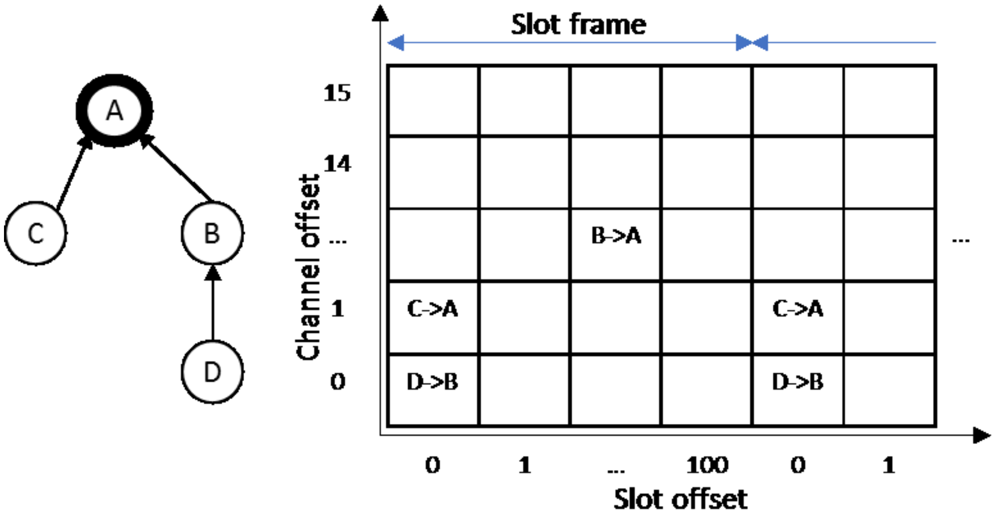


Figure 4. An example TSCH schedule.

2.4 Contiki, Contiki-NG, and Cooja Simulator.

Many applications and operating systems (OS) have been deployed in the last few decades to resolve various requirements in the WSN field. The tools and toolchains used to do the practical work and assess the implementations needed are defined in this section.

2.4.1 Contiki, Contiki-NG OS.

In WSN, the OS differs from the traditional OS used in other devices such as PC, mobile phones,… It is a lightweight OS that allows simple programming to be used by device developers on embedded systems such as sensor nodes. A broad variety of WSN OSs, such as RIOT, OpenWSN, TinyOS, Contiki, and Contiki-NG, are available for use and testing.

Contiki [15] is a lightweight WSN OS designed for platforms with constrained resources. The Real-time OS (RTOS) features are not entirely supported by Contiki. In order to incorporate the strengths of event-driven processes and preemptive threads, Contiki implements a hybrid model. It implemented protothreads to provide event-driven resources while allowing optional preemptive multithreading with an application library that can only be connected to applications that expressly need this preemptive function in a particular library.

The next generation of the Contiki project is Contiki-NG. It offers a low-power, RFC-compliant IPv6 networking stack that enables Internet connectivity. Contiki-NG started as a fork of the Contiki operating system with five goals obtained. Focus on dependable (reliable and secure), standard-based IPv6 communication; Focus on modern IoT platforms, e.g. ARM Cortex M3, …; modernize the structure, configuration, logging, and platforms, to reflect the goals above; Improve the documentation, both code API, module description, and tutorials; Implement a more agile development process, with easier inclusion of new features, and with periodic releases.

2.4.2 Network stacks in Contiki, Contiki-NG.

Contiki [15] is an open-source, lightweight, and multi-tasking operating system built with memory-constrained networked embedded systems and wireless sensor networks. Because Contiki is open-source and has many different contributors that often cause the documentation to be a little short. Apart from its source code comments and illustrations, the lack of adequate documentation made it difficult to obtain enough details about its network stack. There are some small and significant changes to the operating system from the predecessor Contiki-OS to the new Contiki-NG. The former core directory is renamed to OS. The app's directory is now moved to OS, and top-level directory dev, CPU, and platform are now under one directory called the arch. [16]

Three types of network stacks [17] can be used in Contiki: uIP's and rime stack. The uIP stack is a simple TCP/IP network suite implementation that offers IPv4 networking functionality and was later expanded to include IPv6 capabilities. The rime stack provides a series of primitives for custom networking to allow low-power wireless networks to communicate using lightweight layering and the ability to create complex abstractions. Contiki uses a five-layer network stack that, given the computing and memory limitations of most networked embedded systems, is approximately close to the TCP/IP model, but simpler. At the same time, as seen in Figure 5, it also includes the typical seven layers of the Open Systems Interconnection (OSI) model. In the following paragraphs, the definition of each layer is briefly clarified.

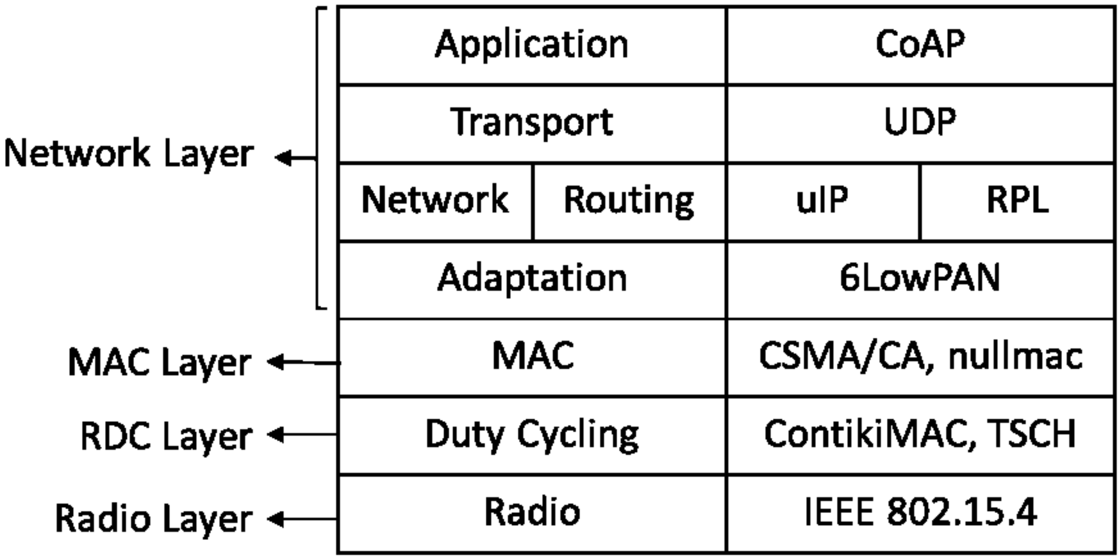


Figure 5. Contiki, Contiki-NG Network stack, and some example protocols

Radio Layer

In the Contiki, Contiki-NG model, the radio or physical layer is the lowest layer. This layer is done by a sensor node radio module which specifies how the input data is organized and constructed for transmission to the upper layers of the network. Most of the radio layers run on the IEEE 802.15.4 mechanism.

Frame Layer

The framer layer is not seen in Figure 5 of the network stack but is placed between the layer of the radio and the layer of the RDC. Like the majority of the layers, the framer layer does not have a standard layer implementation. The frame layer is made up of a series of auxiliary functions that are used to build and parse the frame.

RDC Layer

In the network stack, the Radio Duty Cycling (RDC) layer plays a crucial role as it substantially influences the nodes' energy consumption by requiring the nodes to switch their radio transceivers off as they sleep and ensure that their radio transceivers are awake to transmit or receive the packet. Three defined RDC protocols are currently provided by Contiki: LPP, X-MAC, and ContikiMAC. Based on the original Low-Power Probing protocol, the LPP protocol was developed, thereby improving power consumption. The X-MAC of Contiki is the same as the X-MAC protocol, thereby enhancing some networking and power consumption aspects. Finally, ContikiMAC was developed to expand the low-power listening mechanisms used by subsequent RDC protocols while enhancing energy efficiency at the same time. Based on low-power listening specifications, Contiki-NG uses the ContikiMAC protocol. Time Slotted Channel Hopping (TSCH) is used by ContikiMAC as part of the IEEE 802.15.4e-2012 MAC amendment layer.

MAC Layer

Located on the top of the RDC layer is the Medium Access Control (MAC) layer. In the Contiki and Contiki-NG net stacks, it also plays a crucial role, as it determines how the nodes will connect when the network is congested. The MAC layer is responsible for preventing collisions and retransmitting packets in the event of a collision. Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) and carrier nullmac are two MAC protocols to be used. CSMA/CA offers a number of functions, such as detecting the channel/radio media before sending back-off if another node is sent, waiting for a certain period of time-based on the RDC protocol used, and retransmitting missed packets after collisions. Alternatively, nullmac does not have any handling at the MAC layer, since it only forwards packets from the radio driver to the upper layer, and vice versa, and hence technically has a higher packet loss ratio than CSMA.

Network Layer

In Contiki and Contiki-NG, the network layer is the topmost layer where it covers the different functions of the sublayer, as seen in Figure 5. Until they are sent out it is largely responsible for packing the packages. In other words, by changing such packet frames to suit the upper sublayer format, such as IPv6, it provides different networking and routing functions for received packets before being sent to other nodes. The routing protocol used by Contiki is Routing Protocol for Low-power and Lossy Networks (RPL). By generating an acyclic routing graph starting from the root node, called the DODAG, it is responsible for finding the best path that the transmitted packets can take (Destination Oriented Directed Acyclic Graph). Contiki-NG is supported by ContikiRPL, and RPL Lite is the updated version. The transport and application sub-layers are the last two uppermost sub-layers. The means of communication between the source and the destination nodes are defined by the transport sub-layer protocol, such as the User Datagram Protocol (UDP). At the top, the sublayer of the framework acts as an interface between lower layers and host applications and vice versa. The IETF Constrained Application Protocol (CoAP), a low-power deployment program that seeks to leverage any traditional duty cycle protocol and achieve low-energy consumption, is one of the existing protocols in the application sub-layer.

2.4.3 Cooja Simulator.

A Java-based simulator designed to simulate WSN sensors running ContikiOS is Cooja [18]. It includes a variety of functions for controlling the performance of the nodes of the sensor. Along with the simulator, some functionality is provided, such as the MSPsim device emulator, mobility plug-ins, and the power trace feature. For the emulation of sensor nodes such as Tmote Sky and Zolertia Z1, based on the MSP430 microcontroller, the MSPsim emulator can be used by Cooja. There are three main properties of Cooja's simulated motes: a storage memory containing the software code needed for testing, a mote sort that can be exchanged between several motorcycles by using the same source code, and peripherals for hardware.

Chapter 3: RELATED WORK

3.1 Wireless Link Quality Estimation.

3.1.1 Category of Link Quality Estimators.

The estimator of the link can be split into two groups: hardware-based and software-based estimators. The Link Quality Indicator (LQI), the Received Signal Strength Indicator (RSSI), and the Signal-to-Noise Ratio (SNR) are some examples of hardware-based estimators. They are hardware built-in value so they do not require any overhead computing. However, hardware-based estimation methods not provide an exact estimation, as found and recorded in previous research studies. Furthermore, these calculations are measured only for packets that have been successfully received. Therefore, where the radio connection suffers from unnecessary packet loss, the transmission efficiency may be overestimated by not considering packet losses. The categories of some distinct relation consistency metrics are described in Figure 6 based on their form of estimation.

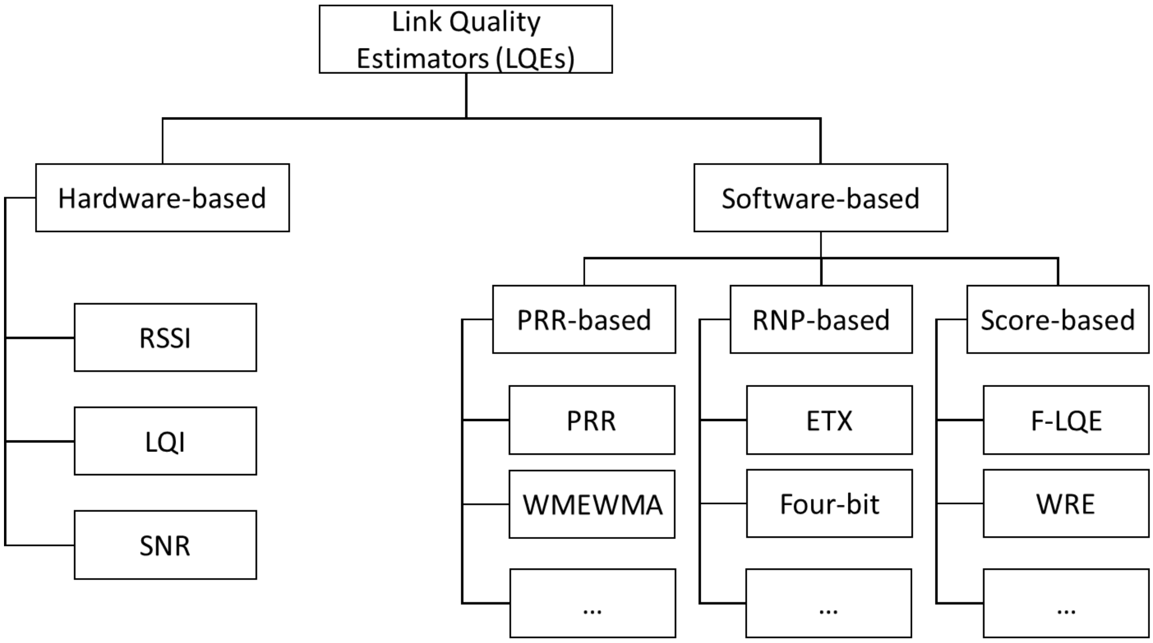


Figure 6. Categories of some Link Quality Estimators.

On the other hand, the number of received and sent packets is calculated by software-based LQEs, and various methods are followed to quantify these software-based relation efficiency estimators. On the sender side, certain software-based relation output estimators are measured, while others are measured on the receiver side [19]. These estimators help one to count or estimate either the rate of reception or the average number of transmissions/re-transmissions of packets needed before their complete reception. It is possible to classify software-based LQEs into three groups, such as PRR-based:

either counting or approximating the PRR, RNP-based: either counting or approximating the number of packet transmissions, and score-based: providing a score that identifies the quality of the link.

Such link quality estimators are simple, but they have been commonly used in routing protocols. The main difference between hardware and software-based estimators is that hardware-based estimators rely on received packet data and do not account for the loss of packets.

3.1.2 Hardware-based Estimators.

The IEEE 802.15.4 standard proposed two ways to assess the link quality after receiving a packet which are RSSI and LQI. RSSI is used to describe the power which is presented by received radio signal strength in dBm and coarsely correlated with the distance. However, RSSI only relies on received packets on the receiver side, not account for the number of the lost packet, and the interference can influence the RSSI value. The LQI measurement is based on the received packets but the IEEE 802.15.4 standards do not define the computation method for LQI value. The standard only states that the range of LQI value is from 0 to 255. Moreover, to develop a general computation using this metric is very difficult since the different vendor has different ways to calculate LQI value. Some studies used the RSSI or LQI to estimate the channel quality.

Noda et al. [20] proposed a new channel measurement based on the channel availability over time. This significantly quantified the utilization of the spectrum. However, in the presence of multipath fading, it may have problems estimating the channel quality.

Audéoud and Heusse [21] studied the correlation between the Received Signal Strength Indicator (RSSI), Link Quality Indicator (LQI), and Packet Delivery Ratio (PDR). RSSI is a weak indication, loosely correlated with PDR. The work was therefore based on the use of LQI since it gives more valuable information. However, the LQI may overestimate the quality of the channel in impulsive noise scenarios, since the LQI does not account for the packet losses.

Eskola and Heikkilä [22] proposed a method to classify wireless channel disturbances related to line-of-sight changes and radio interference. However, the proposed method did not give any metrics that apply to protocols for improving network performance.

Gomes et al. [23] proposed a Link Quality Estimator (LQE) node, dedicated to real-time link quality estimation using the obtained information and RSSI from received packets. They used RSSI and LQI values to infer the Packet Reception Rate (PRR) of the given links. Unfortunately, RSSI and PDR have proven to be only loosely correlated in many situations. The hardware-based estimators do not require computation resources because they use built-in hardware metrics. However, they do not have precise measurements, as found and recorded in previous studies.

3.1.3 Software-based Estimators.

The PRR and the Required Number of Packets (RNP) are some examples of software-based estimators based on the upper layer calculated information. RNP is calculated based on the transmitted packets from senders and is more reactive when compared with PRR. Thus, as long as the traffic is created by the sender, RNP can assess the quality of the link. However, if packets are successfully received after being retransmitted many times, RNP can underestimate the quality of the link. Some RNP-based estimators are Window Mean with Exponentially Weighted Moving Average (WMEWMA), Expected Transmission Count (ETX), and Four-Bit (FB). WMEWMA uses the EWMA filter as the key estimation technique; the PRR is calculated and then smoothed to the previously calculated PRR, which gives a more reliable yet sufficiently agile approximation compared to the PRR [19]. The average-based LQEs show poor reactivity. In order to overcome that, the authors [24] proposed the Kalman Filter-based Link Quality Estimator (KLE). ETX [25] considers link asymmetry by calculating the PRR of the backward link and the forward link. However, Koksal and Balakrishnan [26] found that in congested networks, passive monitoring ETX overloaded, since a large number of nodes did not receive packets to calculate ETX. FB [27] link estimation is designed with four bits of information. FB integrates RNP and WMEWMA with an EWMA filter to approximate the number of retransmissions. The authors in [28] performed a simulation to compare five LQEs: PRR, RNP, WMEWMA, ETX, and FB in smart-grid environments. The simulation result showed that ETX and FB showed better performance since ETX and FB considered the link asymmetry. However, ETX and FB did not consider burstiness in the links during their measurements.

Chapter 4: ALGORITHM AND IMPLEMENTATION

4.1 System and Network Model.

In this study, we concentrated on converging multi-hop networks consisting of one coordinator (sink) node in tree topology and some sensor nodes. During network formation, all nodes are deployed randomly and they have identification numbers. If they have a parent-child relationship and exchange information via this link, the correspondence between two nodes is called a tree-link. Sensor data is produced by sensor nodes and periodically transmitted to the sink via the tree-link.

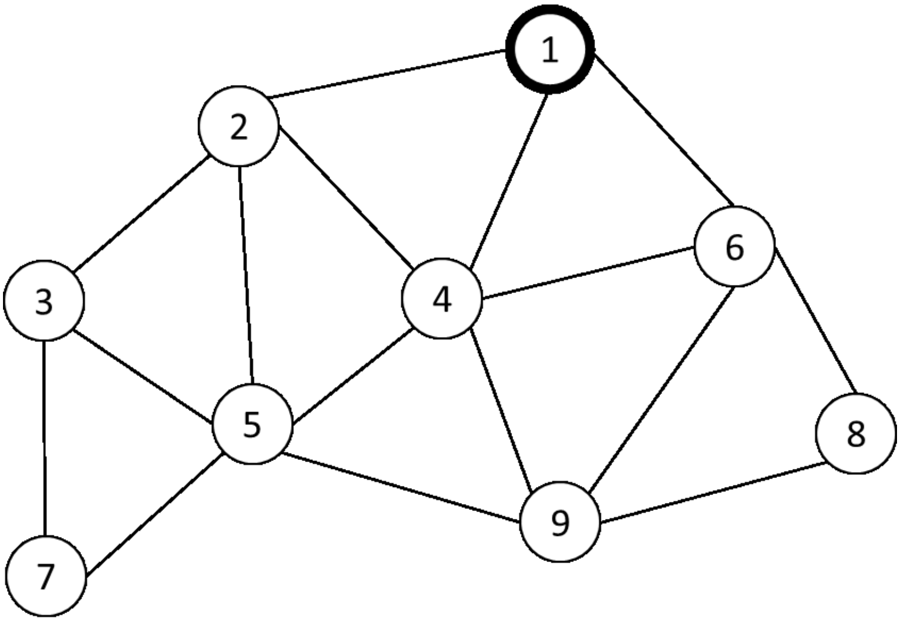
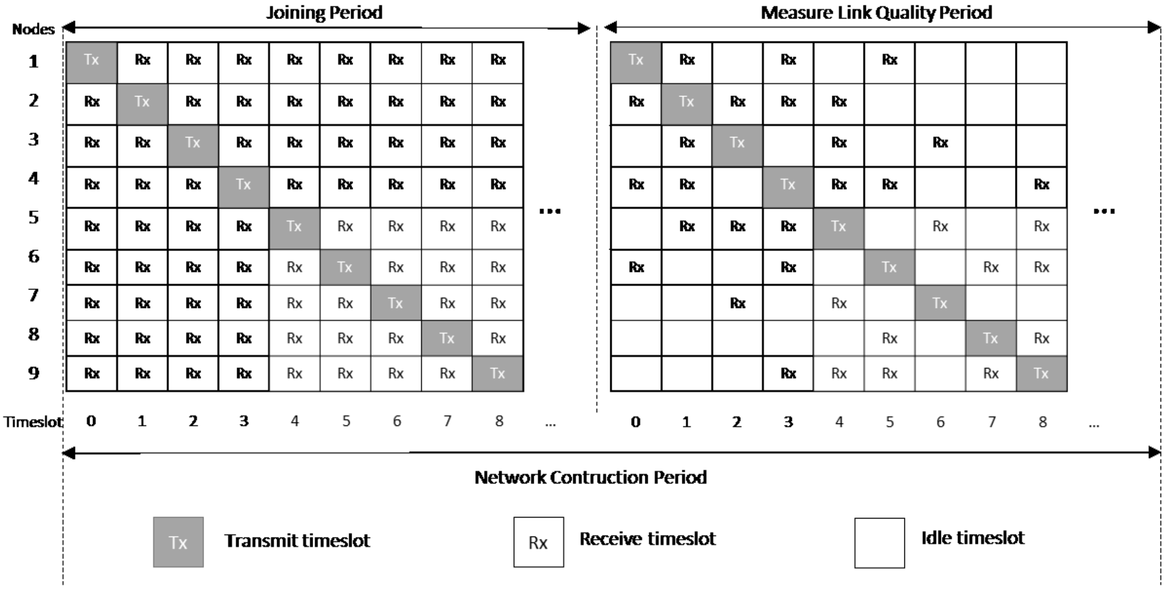


Figure 7. An example network.

Given the fixed number of periodic nodes generated sensor data in the network N = {N_1, N_2, N_3, …, N_n} with N_1 is the Coordinator, and from N_2 to N_n is the sensor nodes. We assume all the sensor nodes are stationary and deployed randomly. An example network is shown in Figure 7. If the two nodes are in the radio range of each other, they are connected by an edge. Time Slotted Channel Hopping (TSCH) is the MAC protocol used in which time slot and channel are allocated to all nodes in the network to connect to the parent node or child node that is collision-free with the neighbor node. We use the advantage of the Orchestra [29] schedule to make the collision-free during joining and measure link quality.

In the Network Construction Period, we define the Joining Period (JP) and Measure Link Quality Period (MLQP). In the JP, when a node starts, it auto-assign the schedule base on Node-ID with the rule each node autonomous assigns transmit (Tx) at timeslot calculated by (Node-ID – 1) since the timeslot value starts from 0, and receive (Rx) timeslot in all the other timeslots. For example with Node

1 in Figure 8, it assigns TX at timeslot 0 and Rx from timeslot 1 to 8 with the example network in Figure 7. Then all the sensor nodes start scanning mode in the same channel to detect the Enhanced Beacon (EB) packet broadcasting from the Coordinator (Node 1) to join the network. A timeslot counter named Absolute Slot Number (ASN) is defined by TSCH. The ASN starts from 0 when the network is created from the Coordinator, then it increments by 1 at each timeslot. After a node joins the network, it will become a broadcasting node to broadcast itself EB packet to expanding the network. When all the sensor nodes join the network, it will update the schedule base on the neighbor has the radio in range then switch to MLQP as shown in Figure 8. The schedule is updated to reduce the redundant assigned timeslot.



Nodes									
1	Tx	Rx	Rx	Rx	Rx	Rx	Rx	Rx	Rx
2	Rx	Tx	Rx	Rx	Rx	Rx	Rx	Rx	Rx
3	Rx	Rx	Tx	Rx	Rx	Rx	Rx	Rx	Rx
4	Rx	Rx	Rx	Tx	Rx	Rx	Rx	Rx	Rx
5	Rx	Rx	Rx	Rx	Tx	Rx	Rx	Rx	Rx
6	Rx	Rx	Rx	Rx	Rx	Tx	Rx	Rx	Rx
7	Rx	Rx	Rx	Rx	Rx	Rx	Tx	Rx	Rx
8	Rx	Rx	Rx	Rx	Rx	Rx	Rx	Tx	Rx
9	Rx	Rx	Rx	Rx	Rx	Rx	Rx	Rx	Tx
Timeslot	0	1	2	3	4	5	6	7	8

1	Tx	Rx		Rx		Rx			
2	Rx	Tx	Rx	Rx	Rx				
3		Rx	Tx		Rx		Rx		
4	Rx	Rx		Tx	Rx	Rx			Rx
5		Rx	Rx	Rx	Tx		Rx		Rx
6	Rx			Rx		Tx		Rx	Rx
7			Rx		Rx		Tx		
8						Rx		Tx	Rx
9				Rx	Rx	Rx		Rx	Tx
	0	1	2	3	4	5	6	7	8

Figure 8. Orchestra's schedule assigns in NCP.

4.2 Measure Link Quality Principle.

To measure the link quality between two sensor nodes, probe packets are used in which the number of probes is predefined. Each node generates a probe packet, P(node_id, sequence), in which sequence is a sequence of probe packets generated, and it increases from 1 up to the number of probes. Upon receiving P(node_id, sequence), the receiver calculates the burstiness value by using the sequence number in P for the sender node and then saving it to the burstiness distribution list (BDL). If a burstiness value duplicates others in the BDL, the burstiness time count for this burstiness value increases by 1. For example, let us consider the measurement link quality for nodes 1 and 4 in Figure 7, where node 1 received a probe packet from node 4 with the sequence number 1, denoted by P(4, 1). The next received probe packet from node 4 is P(4, 4), which means probe packets P(4, 2) and P(4, 3) were lost due to burstiness, and thus the burstiness value is 2. At this time, the current received probe packet is P(4, 4), and the receiver will receive the next probe packet and will again calculate the burstiness value. Assume that the probe packet is P(4, 7) the next time, so the burstiness value is again 2, but the burstiness time count will increase by 1 to indicate that burstiness value 2 was duplicated. Let the result of a transmission per link at the ith sequence be “S” if it was successful and “F” if it failed. With the above example of node 1 and node 4, we have the result of the transmissions after seven probe packets as “SFFSFFS”. After finishing the measure link quality (MLQ) period, we have the BDL of a link with 1000 probe packets transmitted, as shown in Table 2.

Burstiness value	0	1	2	3	4
Burstiness time count	634	129	31	2	1

Table 2. Burstiness distribution list of a link after finishing the measure link quality period.

Consider the relationship between burstiness value and burstiness time count in the BDL. The burstiness value is defined as the number of consecutive losses during probe packet transmission, and the burstiness time count is defined as the number of times the burstiness value appeared. For example, in Table 1, after transmitting 1000 probe packets, a burstiness value of 0 occurred 634 times, a burstiness value of 1 occurred 129 times, and so on. It means that in the 1000 probe packets transmitted, 634 times the result of transmission was “SS” and 129 times the result of transmission was “SFS”, and so on. With a link of good quality, the burstiness value will be small, whereas the burstiness time count will be large for the minimum burstiness value. Conversely, a bad link has a higher burstiness value and a low burstiness time count for the minimum burstiness value.

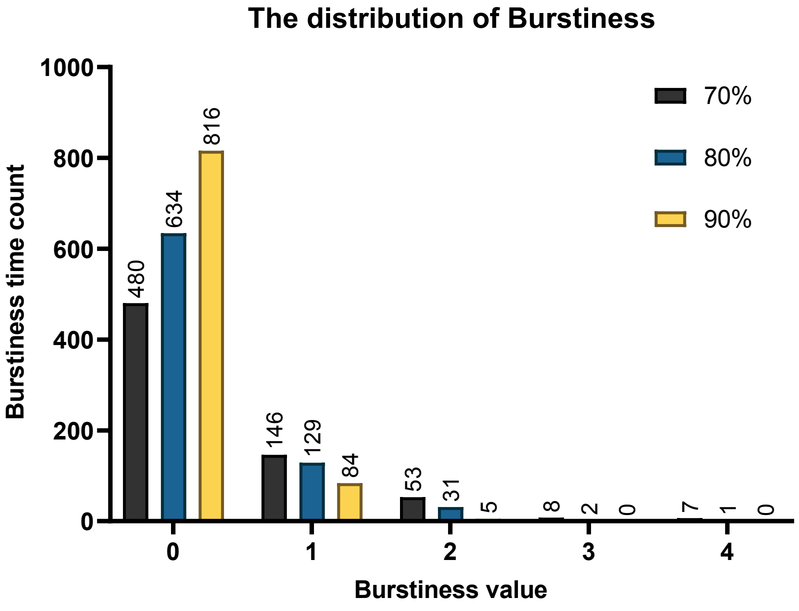


Figure 9. The distribution of Burstiness compares with link PRR.

For example, Figure 9 shows the simulation result from one pair of nodes with the different configurations of the PRR value. In Figure 9, the distribution of the burstiness values is compared with a link PRR of 70%, 80%, and 90%, when the number of probes is 1000 packets. Figure 9 describes how with a good link (PRR 90%) we have a burstiness time count with a minimum burstiness value at 816 and a maximum burstiness value of 2. On the other hand, with bad link quality (PRR 70%), we have a burstiness time count with a minimum burstiness value of 480 and a maximum burstiness value of 4.

4.3 Calculate Burstiness Distribution List.

Algorithm 1: Calculate the burstiness distribution list

```
Input   : Incoming probe P
Output  : BDL
Initialize nbrlist is empty, BDL is empty
When node y send probe packet:
    P = (y, seqno(y));
When node x received probe packet from node y:
  If (y not belong to x's nbrlist)
      nbrlist ← y, y's seqno;
  Else
      burstiness_value(y) = seqno(y) – nbrlist[y].seqno(y) – 1;
      If(burstiness_value(y) not in y's BDL)
            burstiness_time_count(y) = 1;
            (burstiness_value(y), burstiness_time_count(y)) → y's BDL;
      Else
            BDL[burstiness_value(y)].burstiness_time_count(y)++;
      Endif
      nbrlist[y].seqno(y) ← seqno(y);
  Endif
```

First, the neighbor list (nbrlist) and the BDL are set to empty. When sensor node x receives a probe packet from node y, it saves node y's ID and sequence to its nbrlist if node y is not already on node x's nbrlist. If node y is on node x's nbrlist, node x will calculate the burstiness value for node y as follows:

$$\text{burstiness\_value}(y) = \text{seqno}(y) - \text{nbrlist}[y].\text{seqno}(y) - 1$$

In which, seqno(y) is the sequence number in the new incoming probe packet from node y, and nbrlist[y].seqno(y) is the previous sequence number of node y saved in the neighbor list.

The parameter burstiness_time_count(y) refers to a burstiness_value(y) set to 1, and the pair {burstiness_value(y), burstiness_time_count(y)} will be saved in node y's BDL if this value is not already on node y's BDL. If burstiness_value(y) is on node y's BDL, node x will increase the burstiness_time_count(y) referring to burstiness_value(y) by 1. Then, it updates the seqno(y) value in the neighbor list.

This process will repeat during the measure link quality period. The detailed algorithm is in Algorithm 1, and the flowchart is in Figure 10.

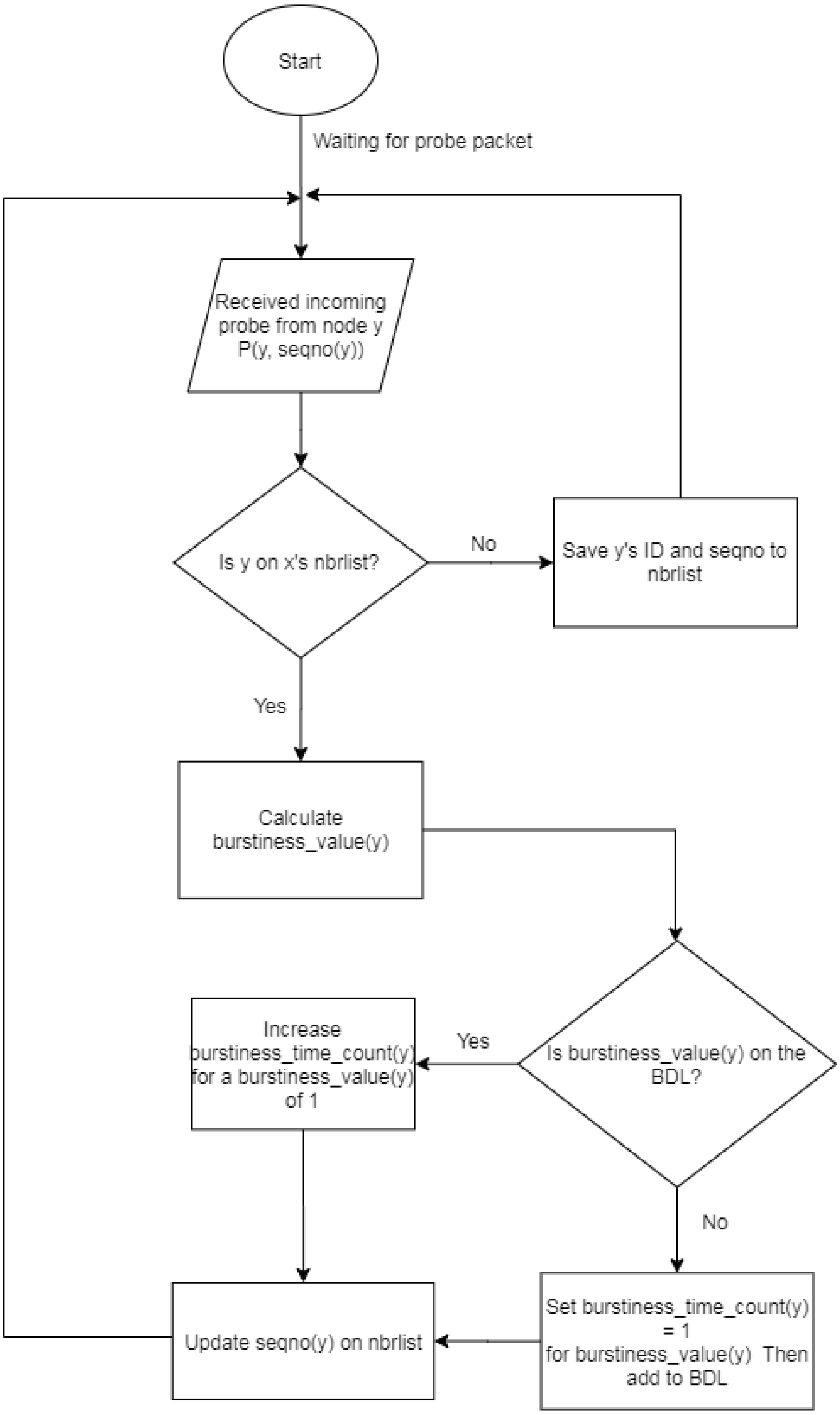

Start
Waiting for probe packet
Received incoming probe from node y P(y, seqno(y))
Is y on x's nbrlist?
No
Save y's ID and seqno to nbrlist
Yes
Calculate burstiness_value(y)
Is burstiness_value(y) on the BDL?
Yes
Increase burstiness_time_count(y) for a burstiness_value(y) of 1
No
Set burstiness_time_count(y) = 1 for burstiness_value(y) Then add to BDL
Update seqno(y) on nbrlist

Figure 10. The flowchart of the main procedures in Algorithm 1.

4.4 Calculate the Burstiness Distribution Metric.

Algorithm 2: Calculate Burstiness Distribution Metric

```
Input    : Ascending sorted burstiness distribution list (BDL),
           number of probes (Nprobes), target PRR (PRRtarget), hop count (h)
Output :  Burstiness distribution metric (Bdist)
: Calculate end-to-end loss ratio by using Equation (1)
: Calculate Nloss_threshold by using Equation (2)
: Initialize the Ncurrent_loss = Bmax*BDL[Bmax]
: For loop in BDL from i = Bmax to 0
:     If(Ncurrent_loss ≤ Nloss_threshold)
:         Ncurrent_loss += i * BDL[i];
:     Else
:         Bdist = i + 1;
:         Break;
:     End if
: End for
: Return Bdist
```

To calculate the Bdist metric, we consider the possible loss ratio for each link, and determine the Bdist value from the BDL. We assume that a target PRR is considered. For example, to reach a PRR target of 99% for the data transmission period, the loss ratio should be lower than 1% for each link. If the route has h hops, then every link that the route goes through has a loss ratio lower than $\sqrt[h]{0.01}$. First, based on the hop count of the current node and target PRR, the end-to-end loss ratio ($\mathrm{e2e}_{\mathrm{loss\ ratio}}$) that we accepted to reach the PRR target is calculated by Equation (1):

$$\mathrm{e2e}_{\mathrm{loss\ ratio}} = 1 - \sqrt[\mathrm{h}]{\mathrm{PRR}_{\mathrm{target}}} \qquad (2)$$

Then, we can calculate the number of packet loss threshold (denoted Nloss_threshold) corresponding to the number of probes (denoted Nprobes) with Equation (2):

$$\mathrm{N_{loss\_threshold}} = \mathrm{e2e_{loss\ ratio}} \times \mathrm{N_{probes}} \qquad (3)$$

The number of current packet loss (denoted Ncurrent_loss) is calculated by multiply pair of {burstiness_value, burstiness_time_count} at ith in BDL as follows:

Ncurrent_loss = i * BDL[i];

in which, *i* is burstiness value, BDL[i] is burstiness_time_count of *i*.

The Bdist metric is set equal to the burstiness value, such that the total of Ncurrent_loss lower than, or equal to, the Nloss_threshold. The detailed calculation of Bdist is in Algorithm 2 and the flowchart for Algorithm 2 is in Figure 11.

For example, a one-hop link has a target PRR of 99%, and the number of probes is 1000 in order to measure the link quality, which means that 1% of 1000 packets can be lost. The number of packets that can be lost during data transmission is 10 packets. In the BDL in Figure 9, we can calculate the Bdist value for successful packet transmission as 2 since the consecutive loss of two times transmission in the link at a PRR of 90% was 10 packets, equal to the allowed packet losses.

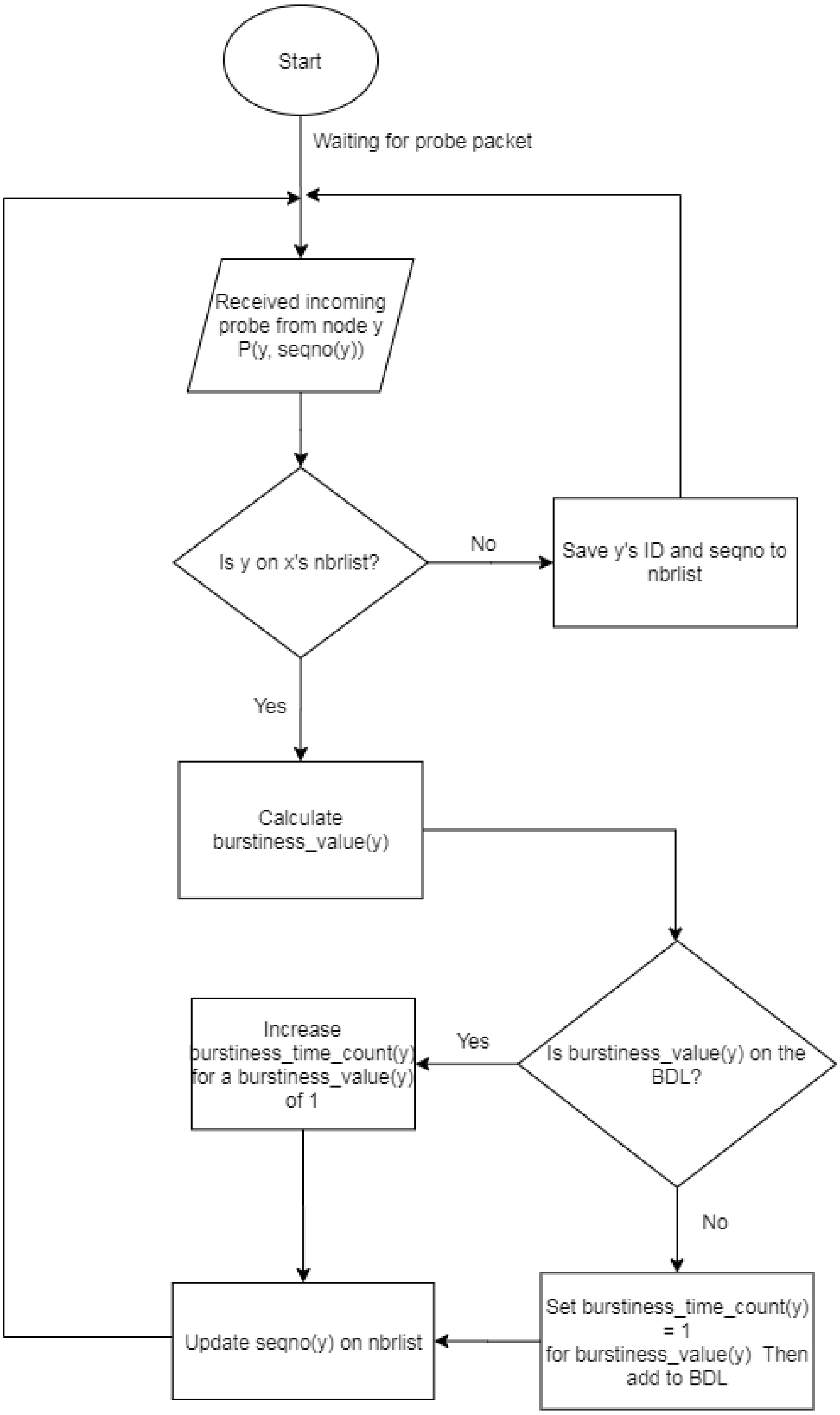

Start
Waiting for probe packet
Received incoming probe from node y P(y, seqno(y))
Is y on x's nbrlist?
No
Save y's ID and seqno to nbrlist
Yes
Calculate burstiness_value(y)
Is burstiness_value(y) on the BDL?
Yes
Increase burstiness_time_count(y) for a burstiness_value(y) of 1
No
Set burstiness_time_count(y) = 1 for burstiness_value(y) Then add to BDL
Update seqno(y) on nbrlist

Figure 11. The flowchart of the main procedures in Algorithm 2.

Chapter 5: EVALUATION RESULTS

By using the Cooja simulator, the Bdist metric was compared with schemes such as ETX and PRR. The key parameters are presented in Table 3.

Number of nodes	10
Number of probes	1000
Target PRR	99%
Link quality of the channel	70% - 90%
Number of the data packets	1000
Network area	100m x 100m
Payload size	26 bytes
Simulation time for each scenario	About 55 min

Table 3. Summary of the simulation set-up parameters.

The link quality of the channel for each link indicates the percentage of successful packets at the receiver, compared with the number of packets from senders. For example, a channel link quality at 70% indicates that the receiver will successfully receive 70 packets if sent 100 packets. The number of probes to measure the link quality and the number of data packets to evaluate the measured link quality metric are both 1000.

5.1 Relationship between the Number of Retransmissions and Network Performance.

To evaluate the effect of the number of retransmissions on the performance of the network, a simple network with one sink (gateway) and nine sensor nodes was considered. The link quality of the channel was fixed, while the number of retransmissions varied from 0 to 3, and we examined the packet reception rate at the sink with each value for the number of retransmissions.

The results in Figure 12 show that since the sensor nodes were not allowed retransmissions, the PRR at the sink achieved a low PRR: 66.3%. When the sensor nodes were allowed retransmissions, the PRR improved since the number of retransmissions increased. It is shown that when the link quality is not very good, the number of retransmissions will decide the network performance. Therefore, to estimate the number of transmissions is very important.

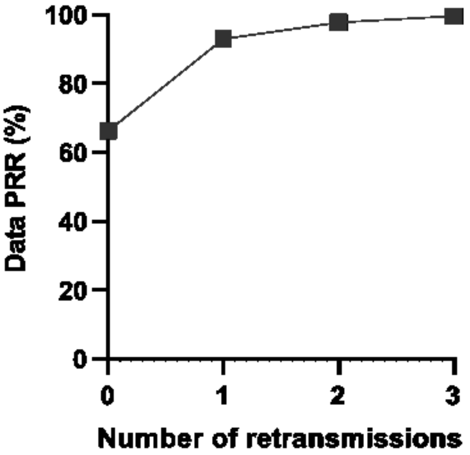


Figure 12. Relationship between the number of retransmissions and the packet reception rate.

5.2 Effect of the Hop Count on Network Performance.

We evaluated the effect of the hop count on PRR. A simple linear network with the hop count varying from 1 to 4 was considered. We compared our proposed approach with schemes like ETX and PRR. Each scheme estimates the number of transmissions by itself. We examined the packet reception rate at the sink for each value of the hop count.

The results are presented in Figure 13, where it is indicated that our proposed method achieved a very high PRR, even at the highest hop count. That is because our proposed method provided a good estimation of the number of transmissions by using the target PRR to estimate link quality. Under other schemes, the PRR decreased since the hop count increased. This is because, with the two other schemes, the estimate of the number of transmissions was not very good.

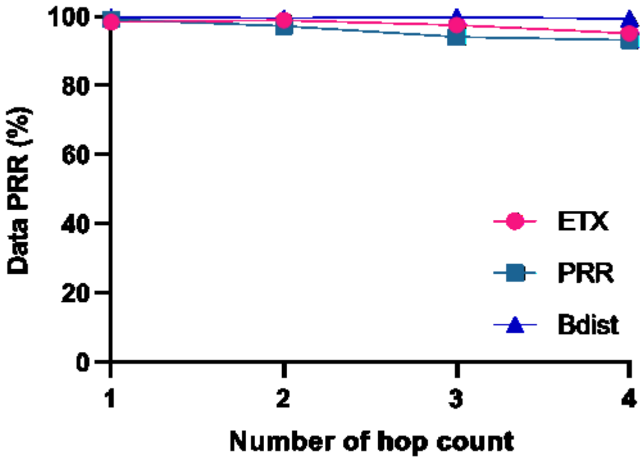


Figure 13. Effect of the hop count on network performance.

5.3 Evaluating the Network with other Estimation Schemes.

In this examination, nine sensor nodes were deployed randomly, and the link quality between every two nodes was set randomly from 70% to 90%. Each sensor node generated 1000 packets periodically. The Time Slotted Channel Hopping (TSCH) MAC protocol was used with the Path Collision-aware Least Laxity First (PCLLF) [26] scheduling algorithm to guarantee that no collisions occurred in the network. We collected the packet reception rate of each sensor node at the sink.

The resulting PRR data for each sensor node at the sink are presented in Figure 14, where the PRR of the proposed scheme almost achieved the target PRR with the estimated number of transmissions for each sensor node. It shows that the algorithm for calculating the number of transmissions worked well. Furthermore, the other schemes achieved a much lower PRR since the estimation of the number of transmissions by ETX and PRR was not very good. This is because ETX and PRR did not consider the burstiness that happens in the links.

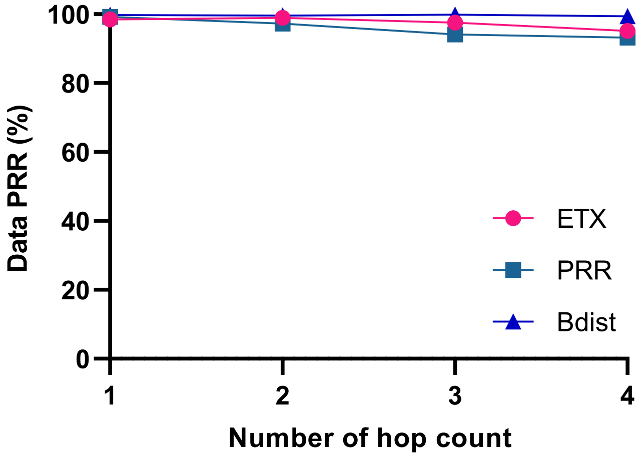


Figure 14. The data PRR of sensor nodes under several estimation schemes.

5.4. Evaluating Networks of Several Types.

In this examination, we ran nine sensor nodes deployed with a link quality between two nodes set randomly between 70% and 90%. We compared network schemes using minimal scheduling, orchestra scheduling, and PCLLF scheduling, with Bdist as the metric value and PCLLF with no retransmission configuration. Each sensor node generated 1000 packets and transmitted using the time slot assigned by the scheduling algorithm. We collected the data packet reception rate at the sink node.

Figure 15 shows the data PRR of each sensor node at the sink. From Figure 15, the network performance of the proposed Bdist metric had the highest reliability compared with the other network schemes. Almost all sensor nodes achieved the target PRR by using the number of transmissions calculated with our method. Other network schemes showed lower performance since they did not consider the retransmissions.

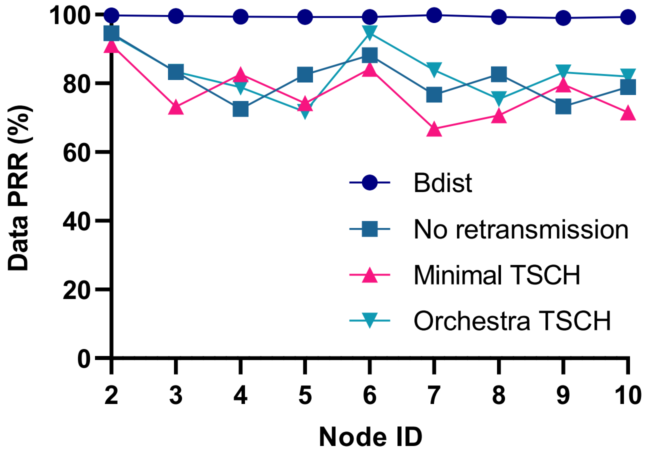


Figure 15. The PRR data of sensor nodes under several network schemes.

Chapter 6: CONCLUSION

In this study, we proposed an LQE to apply to industrial wireless sensor networks with high reliability. Based on the burstiness property of wireless links, we estimated the number of transmissions required to reach the PRR target by using the burstiness distribution list proposed in Chapter 4. We proved by simulation in Cooja that our approach estimated the number of transmissions that can reach the target PRR, as we expected. Based on the simulation in which we compared our approach with some RNP-based methods (ETX and PRR), our proposal can be used as the input metric to calculate the number of transmissions in scheduling for industrial wireless sensor networks. We conclude that our approach is highly suitable for industrial wireless sensor networks that require high reliability for data transmissions. In future work, we will evaluate our approach in regards to the real devices for monitoring and controlling systems in industrial environments.

Probe packet distribution makes increasing the overhead of the network. However, it can be helpful in guaranteeing the reliability of data transmission when the link quality is not very good. Especially in real applications, we cannot guarantee that the link quality can be stable every time. Every link in the network may have a very different link quality. Therefore, estimating the link quality to guarantee the reliability of data transmission is more important than energy consumption in the industry application.

PUBLICATIONS

International Journal

[1] N. H. Nguyen and M. K. Kim, “Link Quality Estimation from Burstiness Distribution Metric in Industrial Wireless Sensor Networks,” *Energies*, vol. 13, no. 23, p. 6430, 2020.

International Conferences

[1] N. H. Nguyen, D. L. Mai, and M. K. Kim, “Performance Evaluation of Linear LoRa Network Protocol,” in *IT Convergence and Security*, Springer, pp. 69–76.

REFERENCES

[1] B. Sinopoli, C. Sharp, L. Schenato, S. Schaffert, and S. S. Sastry, “Distributed control applications within sensor networks,” *Proc. IEEE*, vol. 91, no. 8, pp. 1235–1246, Aug. 2003, doi: 10.1109/JPROC.2003.814926.

[2] T. Winter *et al.*, “RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks.,” *rfc*, vol. 6550, pp. 1–157, 2012.

[3] T. Clausen, J. Yi, and U. Herberg, “Lightweight on-demand ad hoc distance-vector routing-next generation (LOADng): protocol, extension, and applicability,” *Comput. Networks*, vol. 126, pp. 125–140, 2017.

[4] H.-J. Audéoud and M. Heusse, “Experimental Comparison of Routing Protocols for Wireless Sensors Networks: Routing Overhead and Asymmetric Links,” in *2017 29th International Teletraffic Congress (ITC 29)*, 2017, vol. 1, pp. 55–62.

[5] K. Srinivasan, M. A. Kazandjieva, S. Agarwal, and P. Levis, “The β-factor: measuring wireless link burstiness,” in *Proceedings of the 6th ACM conference on Embedded network sensor systems*, 2008, pp. 29–42.

[6] S. Munir, S. Lin, E. Hoque, S. M. S. Nirjon, J. A. Stankovic, and K. Whitehouse, “Addressing burstiness for reliable communication and latency bound generation in wireless sensor networks,” in *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*, 2010, pp. 303–314.

[7] A. Darbandi and M. K. Kim, “Path collision-aware real-time link scheduling for TSCH wireless networks,” *KSII Trans. Internet Inf. Syst.*, vol. 13, no. 9, pp. 4429–4445, 2019, doi: 10.3837/tiis.2019.09.006.

[8] J. Zheng and A. Jamalipour, *Wireless sensor networks: a networking perspective*. John Wiley & Sons, 2009.

[9] J. A. Gutierrez, M. Naeve, E. Callaway, M. Bourgeois, V. Mitter, and B. Heile, “IEEE 802.15. 4: a developing standard for low-power low-cost wireless personal area networks,” *IEEE Netw.*, vol. 15, no. 5, pp. 12–19, 2001.

[10] IEEE Computer Society., “IEEE Standard for Local and Metropolitan Area Networks- Specific Requirements Part 15.4:Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specification for Low-Rate Wireless Personal Area Networks (LRWPANs),” pp. 1–314, 2011.

[11] J. A. Gutierrez, “IEEE Std. 802.15. 4. Enabling Pervasive Wireless Sensor Networks,” *Eat. Corp., Berkeley Uni*, 2005.

[12] 802.15. 4e-2012, “IEEE standard for local and metropolitan area networks—part 15.4: low-rate wireless personal area networks (LR-WPANs) amendment 1: MAC sublayer.” IEEE Standard, 2012.

[13] D. De Guglielmo, G. Anastasi, and A. Seghetti, “From ieee 802.15. 4 to ieee 802.15. 4e: A step towards the internet of things,” in *Advances onto the Internet of Things*, Springer, 2014, pp. 135–152.

[14] T. Watteyne, C. Adjih, and X. Vilajosana, “Lessons learned from large-scale dense IEEE802. 15.4 connectivity traces,” in *2015 IEEE International Conference on Automation Science and Engineering (CASE)*, 2015, pp. 145–150.

[15] A. Dunkels, B. Gronvall, and T. Voigt, “Contiki-a lightweight and flexible operating system for tiny networked sensors,” in *29th annual IEEE international conference on local computer networks*, 2004, pp. 455–462.

[16] “Contiki-NG Wiki.” [Online]. Available: https://github.com/contiki-ng/contiki-ng/wiki/. [Accessed: 09-Oct-2020].

[17] “Network Stack - Contiki.” [Online]. Available: http://anrg.usc.edu/contiki/index.php/Network_Stack. [Accessed: 09-Oct-2020].

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

[19] G. Kirubasri and N. U. Maheswari, “A study on hardware and software link quality metrics for wireless multimedia sensor networks,” *Int. J. Adv. Netw. Appl.*, vol. 8, no. 3, p. 3103, 2016.

[20] C. Noda, S. Prabh, M. Alves, C. A. Boano, and T. Voigt, “Quantifying the channel quality for interference-aware wireless sensor networks,” *ACM SIGBED Rev.*, vol. 8, no. 4, pp. 43–48, 2011.

[21] H.-J. Audéoud and M. Heusse, “Quick and efficient link quality estimation in wireless sensors networks,” in *2018 14th Annual Conference on Wireless On-demand Network Systems and Services (WONS)*, 2018, pp. 87–90.

[22] M. Eskola and T. Heikkilä, “Classification of radio channel disturbances for industrial wireless sensor networks,” *Ad Hoc Networks*, vol. 42, pp. 19–33, 2016.

[23] R. D. Gomes, D. V Queiroz, A. C. Lima Filho, I. E. Fonseca, and M. S. Alencar, “Real-time link quality estimation for industrial wireless sensor networks using dedicated nodes,” *Ad Hoc Networks*, vol. 59, pp. 116–133, 2017.

[24] A. Woo and D. E. Culler, *Evaluation of efficient link reliability estimators for low-power wireless networks*. Computer Science Division, University of California Oakland, Calif, USA, 2003.

[25] D. S. J. De Couto, D. Aguayo, J. Bicket, and R. Morris, “A high-throughput path metric for multi-hop wireless routing,” in *Proceedings of the 9th annual international conference on Mobile computing and networking*, 2003, pp. 134–146.

[26] C. E. Koksal and H. Balakrishnan, “Quality-aware routing metrics for time-varying wireless mesh networks,” *IEEE J. Sel. areas Commun.*, vol. 24, no. 11, pp. 1984–1994, 2006.

[27] R. Fonseca, O. Gnawali, K. Jamieson, and P. Levis, “Four-bit wireless link estimation.,” in *HotNets*, 2007.

[28] V. C. Gungor and M. K. Korkmaz, “Wireless link-quality estimation in smart grid environments,” *Int. J. Distrib. Sens. Networks*, vol. 8, no. 2, p. 214068, 2012.

[29] S. Duquennoy, B. Al Nahas, O. Landsiedel, and T. Watteyne, “Orchestra: Robust mesh networks through autonomously scheduled TSCH,” in *Proceedings of the 13th ACM conference on embedded networked sensor systems*, 2015, pp. 337–350.