Energy Efficient RPL Routing Protocol in Smart Buildings

by

Elnaz Rezaei

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2014

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Energy is an important factor that must be considered by multi-hop wireless mesh routing protocols because most sensors are powered by batteries with a limited capacity. We focus on the industry-standard RPL (Routing Protocol over Low-power and lossy networks) routing protocol that must find energy-efficient paths in low-power and lossy networks. However, the existing RPL objective functions route based on hop-count and ETX (expected transmission count) metrics alone, ignoring the energy cost of data transmission and reception. We address this issue in two ways. First, we design an objective function for RPL that finds paths that require, in expectation, the minimum amount of energy. Second, we design a probing mechanism which configures the transmission power of sensors to minimize energy consumption. The proposed approach is implemented and evaluated using simulations as well as on a small testbed with two Zolertial Z1 motes.

Acknowledgements

I would like to take this time to thank my supervisor, Prof. Keshav, who has supported me throughout my thesis with his patience and knowledge. It has been an honor to work with him. I appreciate his commitment to the success of his students and I consider myself fortunate to have had him as an advisor.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Emerging low-powered networked devices, such as sensors, are being widely deployed in smart object networks, such as for building automation. Network companies, such as Cisco and Ericsson, predict this growth will reach billions of devices over the next ten years [24]. Smart object networks are typically built over *low-power and lossy networks* (*LLNs*) and need a routing protocol to establish and maintain multi-hop connectivity.

Because of unique characteristics of LLNs, the Routing Protocol for LLNs (RPL) was developed and standardized by IETF[1] in 2012 [27]. RPL is an IP-based routing protocol that supports a wide range of applications over LLNs. In RPL, topologies are built proactively to minimize an *objective function* while obeying certain constraints. While the logic of exchanging routing control messages has been specified in the standard documents (called Requests for Comments or RFCs), the choice of an objective function has been left open in RPL. Therefore, different networks can define different objective functions to meet their requirements. The aim of this work is to describe an objective function that minimizes the expected energy cost of transmitting a packet from a sensor node to the root of a wireless sensor network.

There are two contributions in this work. First, we propose a new objective function for RPL which selects paths with the minimum expected transmission energy. For this purpose, we design a new routing metric called E_s (expected transmission energy). Second, we design an adaptive transmission power control algorithm that minimizes the expected

[1] Internet Engineering Task Force

energy cost of transmitting a packet on a wireless link, and that is responsive to changes in the wireless environment.

1.1 Background

Each smart object consists of a microprocessor, a transceiver module, a sensor or actuator, and a power source. Smart objects can measure and collect data across the environment and also to trigger various actions [25]. Smart Objects are constrained in terms of memory, processing power and energy. Additionally, smart objects communicate over wireless low power lossy networks (LLNs), such as IEEE 802.15.4. Lossy links are characterized by link with high bit error rate, frequent packet drops, and instability. Scarce resources, link characteristics and different application requirements made ROLL[2] Working Group to design a new protocol, RPL[3].

RPL is designed to find multi-hop paths from a set of sensor nodes to a *root*. Specifically, RPL is designed to support collection-based networks in which nodes are formed into a directed acyclic graph to handle multi-point to point traffic. In RPL, each node selects a set of parents that are potential next-hops on a path towards a root. In this way, in RPL, unlike a tree, a DAG[4] topology is formed. However, point-to-multipoint and multipoint-to-multipoint traffic are also supported. Compared to a tree structure, a DAG structure lets a node have more than one parent, which is more tolerant of node and link failures.

Figure 1.1 shows an example DAG with eight nodes $\{a, b, c, d, e, f, g, S\}$. Node S is the root which controls other nodes and collects the measurements from them. Node S is the parent of nodes a and b which means that nodes a and b can communicate with S over a one-hop path. Note that node b is the parent of node d, while node d is parent of node g. Hence, node g connects to the root by a three-hop path.

Another difference between RPL and other routing protocols is that it supports the *objective function* concept. An objective function is a metric that is used to choose among alternative paths: when there is such a choice, the one with the lowest value of the objective function is chosen. For example, in Figure 1.1, node c can reach S either using node a or node b. In this case, it will choose the path which has the lower value of the objective

[2]Routing Over Low-power and Lossy networks
[3]Routing Protocol over Low-power and lossy networks
[4]Note that DAG stands for *Directed acyclic graph*

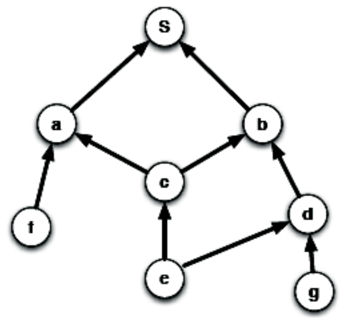


Figure 1.1: A RPL simple example [5]

function. The objective function can be set based on the application requirements. For example, delay is the main concern in a control application, while energy consumption is the main concern in a data collection scenario. In a control application, the objective function can be to minimize delay. Hence, sensors are ranked based on their latency to the root. Next, nodes select parents based on their rank to select a path to the root. This shows the need to select appropriate parents for nodes to achieve a desired level of performance.

The RPL control messages, DAG construction, and communication paradigms are specified in IETF drafts [11]. RPL control messages are defined as a new type of ICMPv6 control messages. There are three control messages: *DIO*, *DAO*, and *DIS*.

DIO[5] messages are multicast periodically from the root and repeated by its neighbors. DIO messages have information about the DAG structure, including the root's unique identifier, routing metrics, rank, and other network parameters. In other words, the DIO messages create the DAG and the upward paths toward the root. The DIO messages are sent via multicast and scheduled using a Trickle timer[16]. So, that they are sent less frequently when the network is stable.

DAO[6] messages are unicast by a node to its parents to build downward routes from the root towards the leaves. Therefore, point to multi-point traffic is routed using paths constructed by DAO messages. When a node joins a network, it can wait to receive a DIO message to detect a DAG or can send a DIS[7] message to solicit a DIO message from its neighbors.

[5]Dag Information Object
[6]Destination Advertisement Object
[7]DAG Information Solicitation

1.2 Related Work

Most existing RPL implementations use one of two routing metrics: *hop-count* and *ETX*[8]. The hop-count objective function ranks the nodes based on their distance to the root. By the *path cost* we mean a scalar value computed as a function of the link or node characteristics along the end-to-end path toward the root. Therefore, when the objective function is hop-count, the path cost is the hop-count number toward the root. For example, the hop-count path cost of the (g, d, b, s) path in Figure 1.1 is three. Although the hop-count objective function is simple, it does not consider the lossy and low-power properties of the network. Therefore, it finds a path with shorter length but perhaps a higher energy consumption than a path with more hops.

The second objective function, which is the expected transmission count, ETX, tries to measure the link reliability between any two nodes in a wireless network. The ETX value of a link indicates how many times we expect to retransmit a packet to successfully deliver the packet over the link. The more lossy the link between two nodes is, the more times retransmission is required and the higher the ETX value. When the objective function uses the ETX metric, nodes are ranked based on their link quality towards the root. Assume node g in Figure 1.1 has the least distance to the root. If there is an obstacle between node g and S which reduces the link quality[9], the ETX value is increased. Consequently, node g is ranked to a high value and is placed at a higher level[10] (a less desirable location) in the DAG.

The hop-count metric does not consider the LLN[11] properties, so may choose paths that have a high energy cost. Similarly, the ETX metric only considers link loss, and does not consider the transmission power needed to reduce link loss levels. However, the energy cost of packet transmission is an important metric in smart object networks and specifically in smart buildings. This motivates our work in coming up with a new objective function for RPL.

Recently, the *remaining energy* routing metric was proposed [14]. Network life is defined as the time when the first node of the network has used its battery completely. The

[8] Expected transmission count

[9] The obstacle would absorbs the signal and the receiver does not receive anything.

[10] Root is always ranked to 1. Rank is monotonically increasing as we move downwards from the root to the leaves.

[11] Low-power and lossy network

objective function thus increases the network life time. The remaining energy metric considers a node's remaining energy to forward a packet through the root. A battery model is used to predict the lifetime of a node as a scalar value. In each path, the node with minimum remaining energy defines the path cost. This method only aims to postpone the time when the first node dies, however, it can waste energy by ignoring the link property. The battery model used is also complex [18]. This has motivated us to focus on the end-to-end transmission energy that considers both the lossy links (link metric) and low-power nodes (node metric) in routing.

1.3 Outline

The rest of this thesis is organized as follows. First we state the problem statement in Chapter 2. Next, we propose the idea of *energy efficient RPL* to make the RPL energy efficient. Accordingly, a new metric, E_s, is proposed which considers the lossy and low-power attributes together in Chapter 3. We propose a transmission power control for RPL based on E_s in Chapter 4. In Chapter 5, the simulation settings and results are presented. Chapter 6 concludes the thesis.

Chapter 2

Problem Statement

2.1 RPL Implementation Methodology

Smart objects usually are highly constrained in terms of physical size, available memory, CPU power and battery life. Additionally, they mainly communicate over wireless low power lossy networks, or LLNs. There are many different types of application each with its own specific routing requirements. For instance, a critical alarm application requires a *time sensitive* path for alarms; consequently the routing objective is to find a short delay and reliable path [25].

Scarce resources, lossy link characteristics and different routing requirements have motivated the IETF to design a new routing protocol for smart object networks. The IETF standard Routing Protocol over Low Power and Lossy Networks, RPL, is designed for smart object networks. RPL is an IP-based distance vector protocol.

RPL assumes the existence of a *root* node that collects measurements and/or controls nodes in a wireless sensor network. RPL forms an abstract gradient called DAG, Directed Acyclic Graph, started at the *root* node [26]. Other nodes are assigned a rank depending on their distance from the root based on an *objective function*. By distance, we do not mean the physical distance necessarily. Distance is a network metric defined by the objective function, such as delay and hop-count. Each node selects a set of parents (rather than a single parent, as would be the case in a tree) such that the ranks of the parent nodes is lower than the current rank of the node itself. This results in the formation of a DAG. In order to compute the path, an *objective function* is defined based on the routing metric. Given a set of neighbors, a node chooses parents such that the objective function of the path to the root from that parent is minimized. This is a generalization of distance vector routing, where the objective function plays the role of a general distance function.

Note that RPL is an application-aware protocol that aims to support a variety of LLN applications, such as home automation, building automation, and industrial automation. Thus, RPL provides a general framework in which to create application-specific DAGs by means of control messages. In other words, the objective function is an input to RPL from the application. Objective function design is mentioned as an open issue in [19].

To implement RPL in practice, we first need to specify the target application. Second, the main application requirements must be defined. Third, the objective function should be designed based on the application requirements. The objective function is a function of link metrics and/or node metrics which assigns a cost to each path. Based on the defined objective function, the routing metrics are formalized. For example, if an objective function is a function of the node's remaining energy, the routing metric is the node's remaining energy. So, we need to design and implement how to measure the node's remaining energy.

Our goal is to determine an appropriate objective function when RPL is used in a smart home or building for sensor measurements. We now discuss the application environment for smart buildings.

2.2 RPL in Smart Buildings

Since different applications have different specifications and routing requirements, we narrowed the application area to smart buildings. We believe that the primary constraint in smart buildings is *energy* because sensors are mainly battery powered and have a restricted source of energy. For example, Zolertia Z1 sensors work with two AA batteries. After installing the system, locating and changing dead batteries is difficult. Other sensors are powered by parasitic sources of supply, such as vibrations or solar panels, and therefore must use energy sparingly.

The Z1 is a low power wireless module compliant with IEEE 802.15.4 and Zigbee protocols. Its core architecture is based upon the MSP430 micro-controllers and CC2420

IC	Operating Range	Current Consumption	Notes
MSP430f2617	1.8V to 3.6V	0.1µA 0.5µA 0.5mA < 10mA	OFF Mode Standby Mode Active Mode @1MHz Active Mode @16MHz
CC2420	2.1V to 3.6V	<1µA 20µA 426µA 18.8mA 17.4mA	OFF Mode Power Down IDLE Mode RX Mode TX Mode @ 0dBm
ADXL345	1.8V to 3.6V	0.1µA 40uA to 145uA	Standby Active Mode
M25P16	2.7V to 3.6V	1µA 4mA to 15mA	Deep Power Down Active Mode
TMP102	1.4V to 3.6V	1µA 15µA	Shutdown Mode Active Mode

Figure 2.1: Approximate Current Consumption of Z1 modules [29]

radio transceivers by Texas Instruments[1][29]. Figure 2.1 shows the current consumption of the different Zolertia Z1 modules [27]. We see that the radio transceiver module is the main energy consumer because the current consumption of other modules is less than the transceiver module current consumption.

In order to reduce sensor's energy consumption, the transceiver module needs to operate optimally. We target transmission energy as the main focus of our work. Specifically, the objective function is to find the path with minimum expected transmission energy. Hence, we design a routing metric to indicate the transmission energy consumption over each link. We call this link metric E_s. We discuss this metric in the next chapter.

[1]Note that ADXL345, M25P16, and TMP102 denote the accelerator, external flash chip, and temperature sensor respectively.

Chapter 3

Es Routing Metric

A routing metric is used by the routing protocol to choose paths. Packets are sent on a path along which the routing metric is minimized. In this chapter, we present the design of a routing metric that minimizes energy consumption for packet transmission along a path. We first model the RPL parent selection mechanism. Next, we explain the Es routing metric.

3.1 RPL Parent Selection Model

Let (x, y) denote the link between node x and node y. We assume that nodes are immobile and that link quality is variable. Link quality affects network performance in WSNs[1] using a poor link results in wasted resources and unreliable service. Hence, the network layer needs to estimate the link quality and to consider it in routing. Link quality estimation can be done by hardware-based methods, such as LQI[2] [12], RSSI[3] [22], and SNR[4] [21]; or by software-based methods, such as ETX[3]. Let $lq(x, y)$ denote the link quality estimation for the link (x, y) where this quality is variable over time. Note that $lq(x, y)$ is also called *link metric* in routing protocols, such as RPL. Let $At(x)$ be the node metric(attribute) of node x. An example of such an attribute would be its remaining battery life.

[1] Wireless Sensor Networks
[2] Link quality index
[3] Received signal strength indicator
[4] Signal to noise ratio

Let $Nbr(x) = \{n_1, n_2, ..., n_l\}$ denote the set of nodes which are in the transmission range of node x. We call this the neighbor set of node x. Let $R(x)$ denote the set of nodes from which node x receives a signal. $Nbr(x)$ and $R(x)$ are not equal, because nodes can have different transmission power in the network. So, for example, x may receive a signal from a neighbor y, but y may not be able to receive a transmission from x because of the lower transmission power used by x.

Let $rank(x)$ indicate the rank of node x in the DAG. In RPL, node p is a parent of node x, if the following conditions hold:

$$p \in Nbr(x) \tag{3.1a}$$
$$x \in Nbr(p) \tag{3.1b}$$
$$rank(p) < rank(x) \tag{3.1c}$$
$$\tag{3.1d}$$

We call set $P(x) = \{p_1, p_2, ..., p_l\}$ the parent set of node x. Node x receives DIO messages from each $p \in P$. Note that it is an implementation choice whether x sends a DAO packet to the parent set, P, or only to its preferred parent.

The best parent denoted $\hat{p}$ is the solution of the following optimization problem.

$$max\ f(lq(p,x), At(p)) \tag{3.2a}$$
$$s.t. p \in P, \tag{3.2b}$$
$$At(p) > Threshold_1, \tag{3.2c}$$
$$lq(p,x) > Threshold_2 \tag{3.2d}$$

Note that the objective function f is a function of the link metric and the node metric.

Two objective functions have been suggested for RPL: *hop-count* and *ETX* [13]. In *hop-count*, the objective function is to select parents and paths only based on the hop-count property of the node, independent of the link quality, or $f(lq(p,x), At(p)) = f(At(p)) = -(hop\text{-}count(p))$, where $hop\text{-}count(p)$ is the number of hops taken on the shortest path from the root to node p.

Expected number of transmission or ETX is a receiver-side link estimator to account for lossy links [6]. Note the ETX value indicates how many times on average a packet needs to be retransmitted over link layer to successfully be delivered to the receiver. The more lossy the link between two nodes, the higher the ETX value. When objective function utilizes the ETX metric, nodes are ranked based on their link quality towards the root. Consequently, paths with the highest end-to-end throughput is selected in lossy networks.

In standard implementations of RPL, the ETX value for a link is updated by a transmitter after sending a data packet. Such implementations initially assume that a link has the worst quality[5] and this estimate is revised over time. This update technique has an inherent problem, in that the ETX metric is only updated on paths where data traffic is already being sent, i.e., to the preferred parent. So, if some other neighbor node were to become a better parent due to a change in link quality, this node would not be discovered. In other words, suppose that ρ shows the selected preferred parent of x. Since x sends the data packets only to ρ, only $ETX(x, \rho)$ is updated. Therefore, ρ needs not necessarily be the best parent, p hat, that is, $\rho \neq \hat{p}$.

3.2 Es Routing Metric

The motivation for our work is to define a routing metric that allows RPL to find the best path in terms of energy consumption, despite changes in link quality over time. The first step is to estimate the minimum energy required to successfully send a packet on each link.

A link's min-energy is the minimum required energy to transmit a packet over it. For example, Figure 3.1 shows a network of four nodes. Node 4 has two parents: $Nbr(4) = \{2, 3\}$. Accordingly two paths exist from node 4 to node 1. We are interested in an energy-efficient path, which is a path with minimum energy consumption. However, we first need to find the energy consumption of each link: $(4, 2)$, $(4, 3)$, $(2, 1)$, and $(3, 1)$.

3.3 Designing the Es metric

Table 3.1 explains the notations used in this section.

[5]ETX value of links are initialized to five.

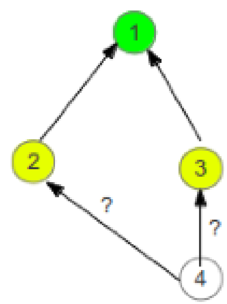


Figure 3.1: A network setting

Notation	Description
$P(l)$	Transmission power of link l
$\lambda(l)$	Link rate of link l
$\bar{L}$	Average packet size
E	Energy consumption per transmission
t	Packet reception probability
$\boldsymbol{E_s}$	Energy consumption for a successful transmission
E_s	Es routing metric

Table 3.1: Notation Table

Let $P(l)$ and $\lambda(l)$ denote the transmission power and the link transmission rate of link l respectively. If $\bar{L}$ is the average size[6] of a packet sent on link l, the energy consumption per transmission, $E(l)$ is given by Equation 3.3.

$$E(l) = P(l) * \frac{\bar{L}}{\lambda(l)} \tag{3.3}$$

Let t denote the packet reception probability. A packet is received in the first transmission with probability t, in the second transmission with probability $t * (1 - t)$ and so on. Therefore, the expected transmission count, ETX, is calculated as:

$$\begin{aligned} ETX = 1 * r + 2 * t(1-t) + 3 * t(1-t)^2 + ... = \Sigma_{i=0}^{\infty}(i+1) * t(1-t)^i \\ = (1-t)(1+\frac{1}{t}) \end{aligned} \tag{3.4}$$

Thus, the average energy consumption for a successful transmission on link l, $\boldsymbol{E_s}$, is:

$$\boldsymbol{E_s} = ETX * P(l) * \frac{\bar{L}}{\lambda(l)} \tag{3.5}$$

An energy efficient routing protocol finds paths and hops with minimum $\boldsymbol{E_s}$. Although $\boldsymbol{E_s}$ is a function of $(ETX, P, \bar{L}, \lambda)$, $\bar{L}$ and λ are constant[7]. Therefore, we can define a new routing metric called E_s which approximates $\boldsymbol{E_s}$ as:

$$E_s = ETX * P(l) \tag{3.6}$$

We configure RPL to rank nodes based on the E_s metric. Note E_s is an additive metric, that is, E_s on a path is the sum of E_s on each link along the path. Therefore, a path with minimum E_s value is selected to route the packets from a node to the root. Note that the E_s metric is defined independent of the technique used to choose the transmission power used on a link. In the next chapter, we will focus on the optimal choice of this transmission power.

In Chapter 5, we evaluate the effectiveness of this metric and compare its energy usage with that of paths computed using ETX on some sample network topologies.

[6]Note that L also includes a packet's header.

[7]Since we do not change the coding level in WiFi, we can assume λ is constant.

Chapter 4

Transmission Power Control

Radio communication quality changes over time and over space which makes static transmission power control, TPC, inefficient in WSNs. The goal of transmission power control is to provide the energy efficiency at the physical layer by dynamically adjusting the transmitter's output power. In addition to energy, transmission power control affects link quality, interference, and connectivity. A lower transmission power results in lower communication range and less interference. Thus, overhearing of a neighbor's packet is reduced. The greater traffic rate, the more severe the interference caused by a transmitter. The interference caused by a high-traffic rate transmitter can be mitigated, to some extent, by the use of TPC techniques. However, reducing the transmission power increases the delay and changes the network topology. Therefore, it is necessary to choose a transmission power level that is neither too low nor too high.

In RPL, transmission power and the choice of preferred parent are correlated. In this chapter, we propose a practical method to choose the transmission power such that the energy cost of data transmission to the root is minimized. In order to find the path with minimum E_s, we discuss the concept of a potential preferred parent as well as a simple probing technique to choose the preferred parent from one of the set of potential preferred parents. To the best of our knowledge, this is the first work on the transmission power control in RPL.

4.1 Related Work

New radio hardware, such as *Texas Instruments CC2420* provide programmable RF output power which can be controlled by a register during runtime. Note that CC2420 is a 2.4 GHz IEEE 802.15.4 compliant RF transceiver chip designed for low power wireless applications [4]. It can be programmed to eight different output power levels as shown in Figure 4.1

PA_LEVEL	TXCTRL register	Output Power [dBm]	Current Consumption [mA]
31	0xA0FF	0	17.4
27	0xA0FB	-1	16.5
23	0xA0F7	-3	15.2
19	0xA0F3	-5	13.9
15	0xA0EF	-7	12.5
11	0xA0EB	-10	11.2
7	0xA0E7	-15	9.9
3	0xA0E3	-25	8.5

Figure 4.1: Typical current consumption and output power settings for CC2420 [4].

Despite the availability of transmission power control on the CC420 chip, which is widely used in wireless sensor netwoks, we are aware of only a single related work on the use of TPC for WSNs [17]. In this work, each node builds a predictive model for each link based on the linear correlation between RSSI[1] and transmission power. According to the empirical results, the RSSI and transmission power relation is not linear. However, the linear model is used as an approximation, because of feedback control theory which forms a closed loop. This method performs dynamic power control in MAC layers that use TDMA[2]; however, it has not been evaluated in CSMA[3] MACs, such as ZigBee. Our work evaluates dynamic TPC in the ZigBee CSMA network. Also, we do not assume a linear relationship between RSSI and transmission power.

A second problem with this approach is that it evaluates performance in low interference environments, such as a grass field, a parking lot, and a corridor. Moreover, its complexity is significant. Restricted memory does not allow us to implement this method in the context of RPL in $Z1$ motes[4]. Hence we are unable to perform a side-by-side evaluation of this approach compared to ours in our testbed.

[1]Received Signal Strength Indicator
[2]Time division multiple access.
[3]Carrier sense multiple access.
[4]Zolertia Z1 has only 8KB RAM.

4.2 Preferred parent approach

Table 4.1 explains the notation and parameters used in this section.

Notation	**Description**
$RS\hat{S}I(p)$	recorded RSSI for parent p
$RSSI(p)$	current RSSI for parent p
ρ	preferred parent
T	nodes in the transmission range of node x
$P(x)$	parent set of node x
$Pt(x)$	potential preferred parent set of node x
$age(p)$	Time duration from the last received packet from node p till now.
$PRR(x,p)$	forwarding packet reception ratio from node x to node p
Tx-$array$	Available transmission powers to be programmed

Table 4.1: Notation Table

The key idea behind our approach is to send a series of probes from a node to its set of potential preferred parents in order to determine the best transmission power. There are two phases during which the preferred parent and the transmission power are evaluated: *initialization* phase and *environment change* phase. When a node joins the network, its transmission power has to be set properly. This is done during the initialization phase. In addition, during ongoing transmissions, we monitor the packet delivery ratio to detect changes in the environment. Specifically, if the packet delivery ratio drops below a *threshold*, environment change is declared.

Algorithm 1 shows the overall scheme to find the preferred parent. This is executed either during initialization or because the packet delivery ratio to the current parent has fallen below a threshold. In this case, node x probes a subset of its parents called its potential preferred parent set. At the end of probing, node x updates its preferred parent. Note that the preferred parent is the candidate parent with the lowest value of E_s.

Input: Node x, set P
Result: Preferred parent
send packet to the current preferred parent ρ ;
update $PRR(x, \rho)$;
if *(PRR(ρ) < threshold) or (state==Initialization)* **then**
  $Pt \leftarrow$ Potential preferred parents(P); `/* Algorithm 2. */`
  Probe Pt set and find the least transmission power for each p in Pt ;
  `/* Algorithm 3 */`
  $\rho \leftarrow$ Find-preferred-parent(Pt);
end

Algorithm 1: Overall algorithm for finding the preferred parent

Algorithm 2 explains how node x selects up to three parents with minimum ranks as its potential preferred parents as long as they satisfy conditions on their age and their *RSSI*.

Note that we use RSSI to select potential preferred parents in our algorithm. The CC2420 actually provides two link metrics: *RSSI*[5] and *LQI*[6]. However, RSSI is a better link quality indicator when its value is above the sensitivity threshold of CC2420 which is -87 dBm [22].

[5] Received signal strength indicator.
[6] Link quality indicator.

Input: P (Parent set)
Result: Pt (Potential parent set)
Temp $\leftarrow$ ϕ;
for *each* $p \in P$ **do**
    if *age(p)*$<$ I_{max} *and* $RS\hat{S}I(p) > RSSI_{threshold}$ **then**
        Temp $\leftarrow$ $Temp \cup \{p\}$;
    end
end
if *size(Temp)*<4 **then**
    Pt $\leftarrow$ Temp ;
end
else
    Temp $\leftarrow$ Sort Temp by rank;
    Pt $\leftarrow$ First three elements of Temp ;
end

Algorithm 2: Potential preferred parent algorithm

4.3 Setting transmission power

Transmission power selection tries to minimize the end-to-end E_s to the root. A node sends probes with each transmission power setting, and finds the E_s value for each $p \in Pt(x)$. Consequently, the parent with minimum E_s is declared to be the preferred parent for that specific transmission power. The parent with minimum E_s value across all different transmission powers is selected as preferred parent.

```
Input:  Node x and Pt(x)
Result: Preferred parent and transmission power to use to communicate with this
        preferred parent
Tx-array ← [ 0.003, 003, 0.1, 0.2, 0.3, 0.5, 0.8, 1];
Es-array ← [100, 100, 100, 100, 100, 100, 100, 100];
Index ← 4;
increase-tx ← 0;
for each p ∈Pt do
|  while 0 < Index < 8 do
|  |  ETX ← probe(p) ;                                    /* Algorithm 4  */
|  |  Es-array(Index) ← ETX*Tx-array(i) ;
|  |  if Index==4 then
|  |  |  if ETX == 1 then
|  |  |  |  increase-tx ← 0 ;
|  |  |  end
|  |  |  else
|  |  |  |  increase-tx ← 1 ;
|  |  |  end
|  |  end
|  |  Min-Es-Index ← Index(Find-Min(Es)) ;   /* This is the index of the
|  |  transmission power value with the lowest Es so far  */
|  |  if increase-tx==1 then
|  |  |  Index++;
|  |  end
|  |  else
|  |  |  Index- -;
|  |  end
|  end
end
Min-Es-Index ← Index(Find-Min(Es)) ;
Set T_x to Tx-array(Min-Es-Index);
```

Algorithm 3: Choosing transmission power

4.4 Probing

We now discuss how probing is actually carried out. For each potential parent and each transmission power value, *probing-window* probes are unicasted because link measurements are more accurate with unicast packets resemble actual data transmission over the link [3]. Note that CSMA provides link-level acknowledgments or retransmissions which is used to compute $ETX(p), \forall p \in Pt$.

If N_t is the total number of transmissions of node x and N_{ack} is the total number of ACK messages, the packet reception ratio of x to ρ is defined by Equation 4.1.

$$PRR(\rho) = \frac{N_{ack}}{N_t} \tag{4.1}$$

Note $ETX(x, y)$ and $PRR(y)$ are inversely correlated. If e indicates the link error rate, Equation 4.2 represents the relationship between PRR, ETX, and e.

$$ETX = \frac{1}{PRR} = \frac{1}{1-e} \tag{4.2}$$

Probe packets are sent staggered to avoid transmission synchronization and interference.

Input: Probing-time τ, *Max-Num-Transmission*, Node p, Probing window
Result: ETX(p)

```
totalProbes ← 0;
PR̂R(p) ← 0;
while totalProbes < probing-window do
    set-timer(τ);
    wait(timer);
    send a probe;
    t_x ← number of transmissions by MAC layer;
    totalProbes ← totalProbes+1;
    if t_x < Max-Num-Transmission then
        PRR ← 1/t_x;
    end
    else
        PRR ← 0;
    end
    PR̂R(p) ← (0.1 * PRR + 0.9 * P̂RR);
end
ETX(p) ← 1/P̂RR;
```

Algorithm 4: Probing algorithm

In the implementation, node x sends 10 probe packets to each potential preferred parents in 10 seconds. Probe messages are implemented as *ICMP6* packets.

Chapter 5

Evaluation

In this chapter, we first discuss our decisions regarding the evaluation platform. Second, we evaluate the proposed E_s metric and transmission power control mechanism in some example topologies by simulation. Finally, we present a real (but tiny) implementation of our transmission power control algorithm on a two-node network.

5.1 Decision Choices

We discuss and introduce the evaluation platforms which we used in this project.

5.1.1 Evaluation Platform

We evaluate our algoritms on two platforms:

- The Cooja simulator that simulates the Contiki operating system, several sensor motes, and the radio environment.
- The Zolertia Z1 mote that runs the Contiki operating system.

Contiki is an open source operating system designed for low-power wireless devices [8]. Contiki is implemented in C and is also fully simulated by a network simulator called *Cooja* that is implemented in JAVA. In addition to simulating every aspect of Contiki faithfully, Cooja also implements the radio duty cycle effects on nodes running Contiki. Contiki provides communication using μIP, a full light IPv6 stack for tiny microprocessor.

Despite its skimpy documentation, we chose the Contiki and Cooja platform for three reasons. First, a simple version of RPL, called *ContikiRPL* [24], is implemented within Contiki. Second, experimental objective functions can be implemented and evaluated in the simulator and then transferred to real sensor nodes for further experiments and analysis (Figure 5.1). This makes it easier to test our ideas. Third, Cooja emulates different types of radio environments and signal fading, including the unit disk model and Rayleigh fading (discussed in more detail below).

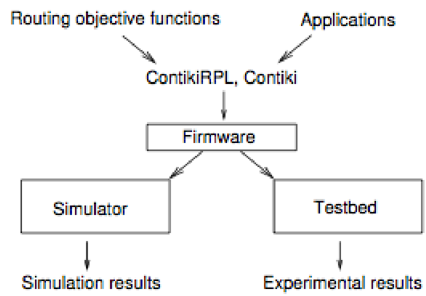


Figure 5.1: Contiki, Cooja, and ContikiRPL platform [24]

Finally, Cooja is capable of simulating different sensor drivers, which can affect mote power consumption and routing.

5.1.2 Zolertia Z1

We tested some aspects of our work on Zolertia Z1 motes. This is a low power platform compliant with IEEE 802.15.4 (Zigbee). The Zolertia Z1 mote contains an *MSP430* microcontrollers and a *CC2420* radio transceiver. We selected Zolertia Z1 for two reasons. First, Z1 supports Contiki OS. Second, Zolertia provides extensive documentation on programming its motes.

5.1.3 Propagation Models in Cooja

The randomness of wireless channels affects the development and performance of routing protocols in WSNs. A radio propagation model mathematically characterizes radio wave

propagation in the channel, modeling issues such as path loss and interference. For example, path loss may happen because of the environment, the propagation medium, the distance between transmitter and receiver, and the height of antennas [2]. Indoor and outdoor environments have different propagation models.
Cooja provides five different radio propagation models:

- No radio traffic
- Unit Disk Graph Medium (UDGM)-constant Loss
- Unit Disk Graph Medium (UDGM)-distance Loss
- Directed Graph Radio Medium (DGRM)
- Multi-path Ray-tracer Medium (MRM)

In the UDGM-constant loss model, the transmission range is modeled as a disk: sensors inside the disk receive all messages and sensors outside the disk receive none. Although RSSI is a function of transmission power, this model ignores transmission power, interference, and asymmetric links. Therefore, it is too simplistic for our work.

The UDGM-distance loss model extends the UDGM-constant loss model by taking interference into account. It defines two additional parameters for a network: *Success-Ratio TX* and *Receive-Ratio RX*. Therefore, interference happens at transmitter by probability $1 - \textit{Receive-Ratio}TX$ and at receiver by probability $1 - \textit{Receive-Ratio}RX$. In this model, links are assumed symmetric and other wireless environment parameters, such as reflection, are ignored. In this model, links are assumed symmetric and other wireless environment parameters, such as reflection, are ignored.

The DGRM model defines a four-element vector for each link: (receive ratio, propagation delay, RSSI, LQI). Interference only is considered at receiver. Compared to previous models, DGRM supports asymmetric links. However, it does not support different transmission powers. The RSSI value is always constant regardless of changing the link length.

The most complex model is the MRM propagation model that uses a ray tracing technique. This model takes refraction, reflection and diffraction into account. Obstacles are considered as a signal strength attenuator. MRM also supports asymmetric links and changing transmission power.

We use the MRM propagation model as it is best aligned with the reality. This is in contrast to previous work using Cooja, that uses the UDGM-distance loss propagation model [1], [15]. To the best of our knowledge, this is the first work to use the MRM propagation model to evaluate the RPL protocol.

5.2 Simulations

In this section, we evaluated the E_s metric performance in three different topologies. For all simulations, the MRM propagation model is used. In MRM, RSSI is a function of distance, output power, obstacle, and interference. We compare E_s metric with only the ETX metric because it has shown that *hop-count* metric results in paths with shorter length and higher power consumption in lossy networks [1].

5.2.1 A 2-node Network

Figure 5.2 shows a network of two nodes: node *1* as root and *2* as sender.

We first evaluate the value of ETX as the transmission power of node 2 takes on all possible values, as shown in Figure 4.1. Node 2 sends one *hello message* per minute for a simulation time of one hour.
Figure 5.3 illustrates the ETX value over time for different transmission power levels. We define the largest possible ETX value, which corresponds to a complete inability to transfer a packet over a radio link $ETX_{max} = 10$ and initialize ETX (recall that this is computed as a exponentially weighted moving average) to 5. For low values of transmission power($Tx = 3$[1] and $Tx = 7$) the ETX value quickly increases and converges to ETX_{max} indicating that, at these transmission powers, the receiver does not receive any packets from the sender. For transmission power levels over 11, the ETX decreases over time from the initial value of 5 and converges to 1. However, the ETX value is fluctuating for $Tx = 11$ indicating that this is a marginal transmission power. This simple experiment shows that always using the maximum transmission power wastes energy: it is possible to reduce transmission power from the maximum value to a lower value with no loss in link quality.

[1] Note that Tx shows the transmission power index (level).

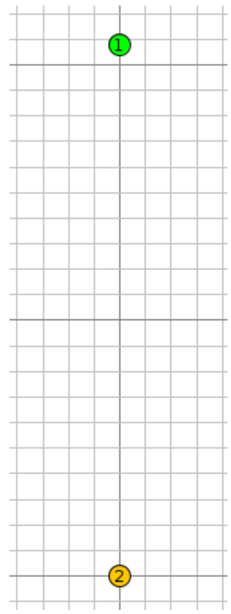


Figure 5.2: Experiment with 2 nodes

Table 5.1 shows the transmission power in milliWatts, $RSSI$, $E\bar{T}X$[2], and E_s for each transmission power.

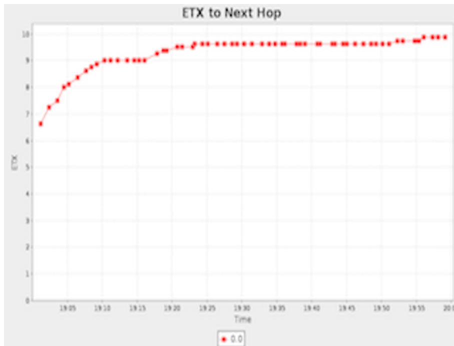
T_x level	T_X(mW)	RSSI	$E\bar{T}X$	$E_s(mW)$
3	0.0032	-111.41	10	0.0297
7	0.0316	-101.32	9.2	0.2909
11	0.1	-96.42	1.6	0.1600
15	0.1995	-93.41	1	0.1995
19	0.3162	-91.41	1	0.3162
23	0.5012	-89.41	1	0.5012
27	0.7943	-87.41	1	0.7943
31	1	-86.41	1	1.0

Table 5.1: Measurements of the two-node network

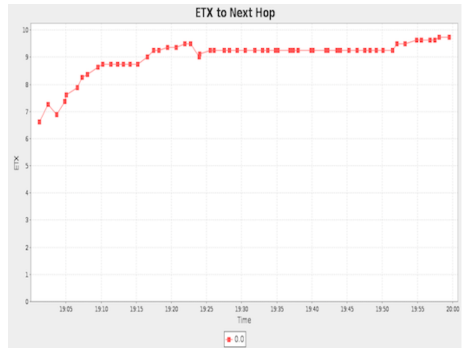
Figure 5.4 illustrates the E_s values over transmission power levels.

Not that the E_s function over Tx levels is a convex function with a minimum at $Tx = 11$.

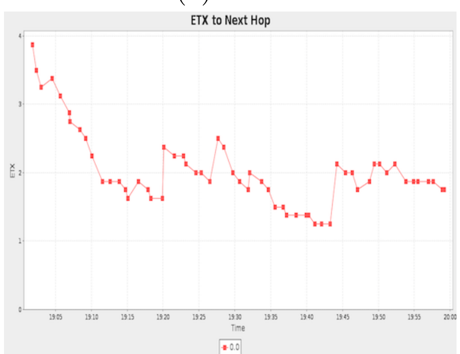
[2] $E\bar{T}X$ indicates the mean of the ETX.



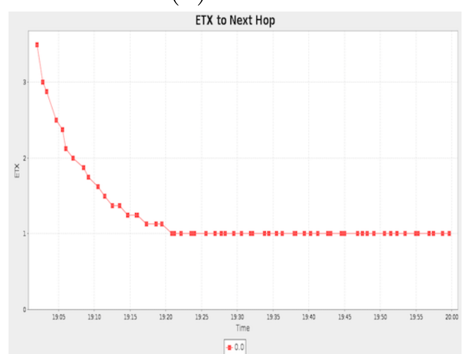
(a) Tx=3



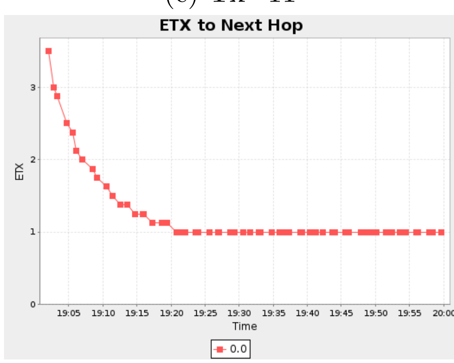
(b) Tx=7



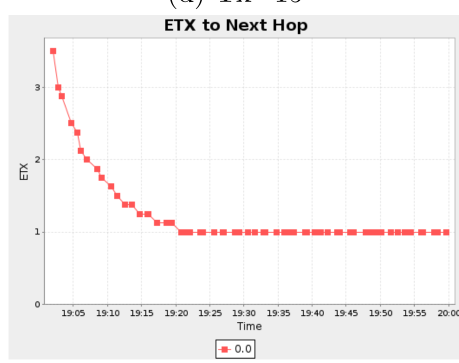
(c) Tx=11



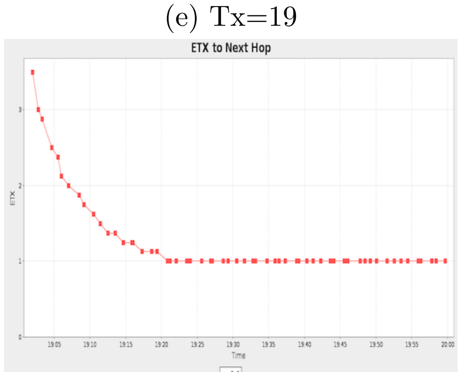
(d) Tx=15



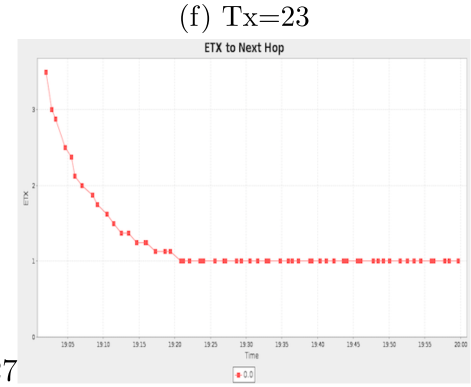
(e) Tx=19



(f) Tx=23



(g) Tx=27



(h) Tx=31

Figure 5.3: ETX over time (minutes).

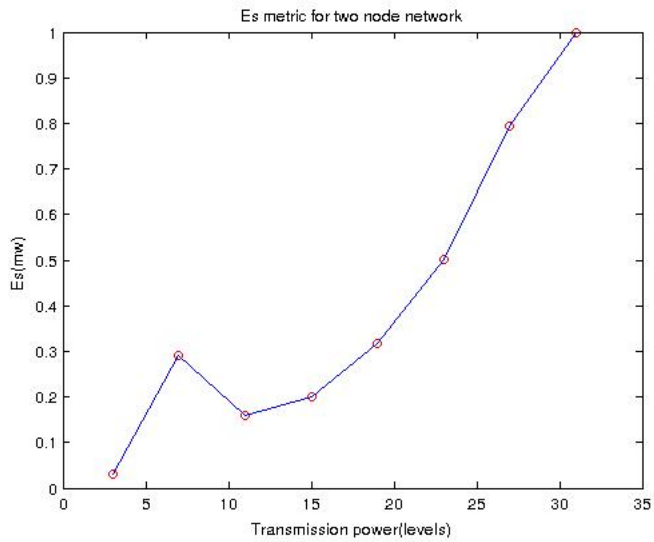


Figure 5.4: E_s values over transmission power levels

5.2.2 A 6-node Network

We now consider a somewhat larger network with 6 nodes. Here, we show how path selection happens after the movement of a node. Figure 5.5 shows the network topology. Node 1 is the root (receiver). Nodes 2 to 6 send a packet to node 1 every minute. Table 5.2 shows the transmission power used at each node.

The root sends at the maximum power. The potential parents send at a fixed transmission power level of 23, and node 6 uses one of eight different transmission powers.

If node 6 selects different transmission power values, it results in different preferred parents and ETX values for each link. Table 5.3 shows the ETX values measured by node 6 to its candidate parents for each transmission power level. The ETX value of link $(6, 1)$ is one when its transmission power level is greater than or equal to 15. Accordingly, node 1 is selected as the preferred parent of node 6 and other candidate parents need not be probed for these transmission power levels. However, node 6 probes nodes $\{3, 4, 5\}$ at $Tx = 7, 11$ because $ETX(6, 1)$ at these power levels is more than one.

Different transmission power values cause different choices preferred parents and ETX values for each link. Consequently, this results in the E_s path cost values for node 6 as is

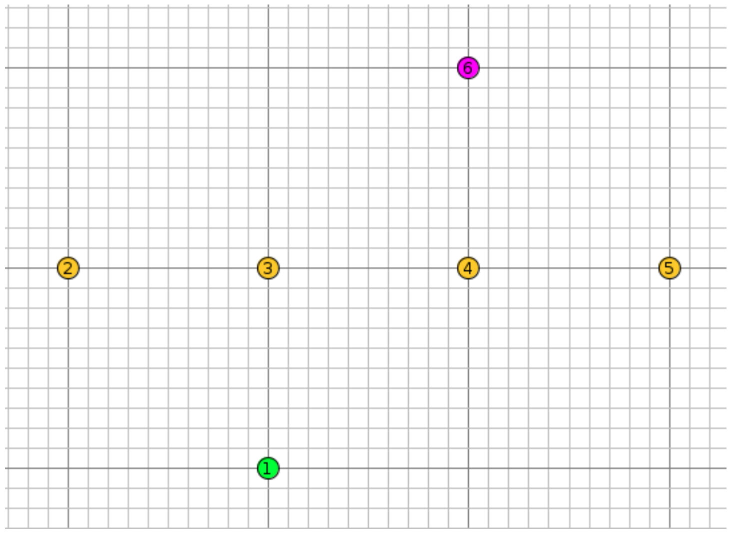


Figure 5.5: A six node network topology

Node ID	Node location	Node output power
1	$(200, 300)$	31
2	$(100, 200)$	23
3	$(200, 200)$	23
4	$(300, 200)$	23
5	$(400, 200)$	23
6	$(300, 100)$	3,7,11,15,19,23,27,31

Table 5.2: Location and power settings of nodes

Transmission Power Levels	3	7	11	15	19	23	27	31
Node 1	-	9.14	2.21	1	1	1	1	1
Node 2	-	8.03	5	5	5	5	5	5
Node 3	-	9.29	2.47	5	5	5	5	5
Node 4	-	6.80	2.23	5	5	5	5	5
Node 5	-	8.23	3.5	5	5	5	5	5

Table 5.3: ETX values measured by node 6 to its candidate parents

shown in Table 5.4 for node 6. The minimum E_s value is for $Tx = 15$ that results in node 6 selecting node 1 as its preferred parent.

Transmission Power Levels	3	7	11	15	19	23	27	31
Node 1	-	0.274	0.221	0.2	0.3	0.5	0.8	1
Node 2	-	0.741	1	1.5	2	3	4.5	5.5
Node 3	-	0.779	0.747	1.5	2.0	3.0	4.5	5.5
Node 4	-	0.704	0.723	1.5	2.0	3.0	4.5	5.5
Node 5	-	0.747	0.85	1.5	2.0	3.0	4.5	5.5

Table 5.4: E_s path cost values for node 6

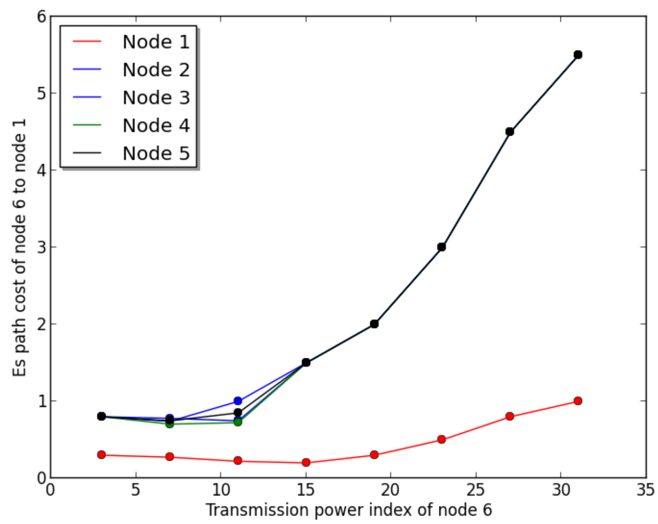


Figure 5.6: Es path value for node 6

The overall network energy consumption of the proposed approach is shown for different transmission power level choices for node 6 in Table 5.5. Over the course of the simulation run, the network energy consumption for the standard RPL was $3.41mJ$. Therefore, the proposed approach improved the network energy consumption by 26% over standard RPL.

Node 6 transmission power level	3	7	11	15	19	23	27	31
Network energy consumption(mJ)	-	3.78	2.74	2.53	2.62	2.83	3.18	3.41

Table 5.5: Network Energy consumption in a static 6 node network

Dynamic Network

Recall that each simulation is run for an hour. In this simulation, at time $t = 20min$, the location of node 1 is changed from $(200, 300)$ to $(4000, 300)$ and an obstacle is placed which blocks all signals from node 1 to node 6 as shown in Figure 5.7.

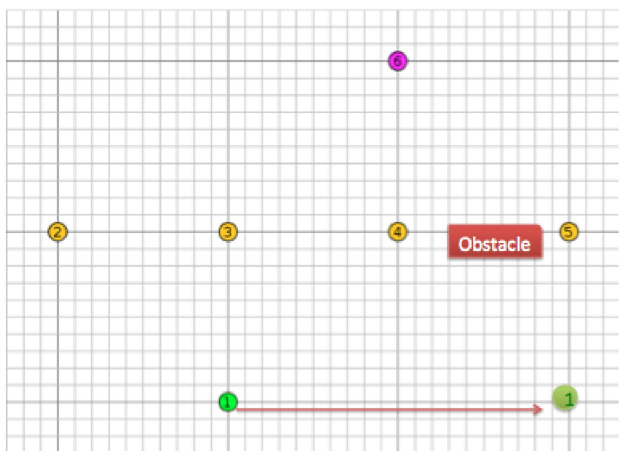


Figure 5.7: Changing the network topology

In Table 5.6 each column shows the transmission power levels chosen by node 6 and each row shows the ETX value to that potential preferred parent. Each cell shows the the ETX value of a potential parent at a specific transmission power of node 6. Because of very low transmission power at level 3, the first column is left empty. Note that node 2 is not a potential parent because we only choose the best three neighbors as potential parents.

Figure 5.8 shows the Es path value of node 6 to node 1 for different transmission power levels of node 6. The figure appears to show that the lowest E_s value is for the direct path to the root node 1 when the transmission power level is 7. However, this is an invalid conclusion because the ETX to node 1 at any power level is very high (see Table 5.6) so the packet delivery ratio is very small. The standard RPL implementation arbitrarily assigns

Transmission Power	3	7	11	15	19	23	27	31
Node 1	-	8.82	5.85	5.31	4.74	5.73	5.73	5.73
Node 3	-	7.32	2.07	1	1	1	1	1
Node 4	-	6.29	1.18	2.10	2.10	2.10	2.10	2.10
Node 5	-	8.56	2.04	2.13	2.10	2.10	2.10	2.10
ρ	-	4	4	3	3	3	3	3

Table 5.6: ETX values measured by node 6 to its candidate parents

a packet that is not delivered at all a retransmit count of 10. This results in the ETX and therefore E_s value to be much smaller than it should be (the retransmit count should really be infinite). In our work, therefore, we automatically ignore links whose ETX value exceeds 5. This allows us to ignore such spurious E_s values.

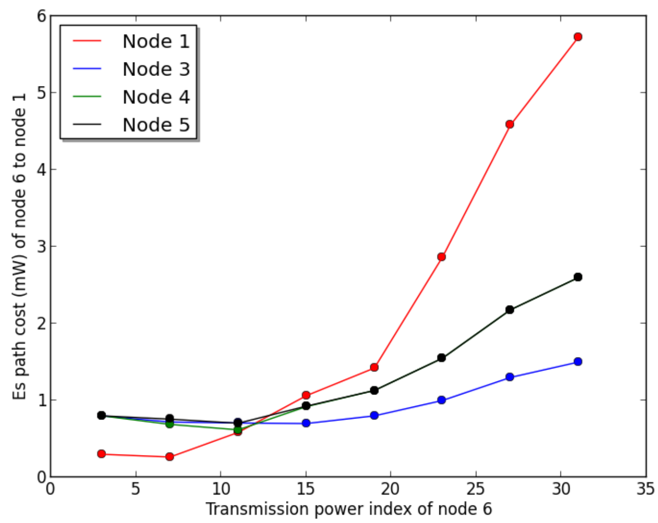


Figure 5.8: Es path cost for node 6

The total energy consumption when using standard RPL in Figure 5.7 is $9.87mJ$. The proposed approach improves the energy consumption. Table 5.7 shows the network energy consumption over different transmission power levels of node 6. Note that the proposed approach energy consumption at $Tx = 15$ is $3.524mJ$ which is 36% of the standard RPL energy consumption. Therefore, the network energy consumption is improved by 64% over

Node 6 transmission power level	3	7	11	15	19	23	27	31
Network energy consumption(mJ)	-	4.980	3.524	3.904	4.183	5.374	6.866	7.886

Table 5.7: Network Energy consumption in a dynamic 6 node network

standard RPL.

5.2.3 A 25-node Network

Figure 5.9 shows a grid topology of 25 nodes. Node 1 is the root and other nodes are clients which sends a report every minute during the one hour simulation time. Figure 5.10 shows a snapshot of the experiment.

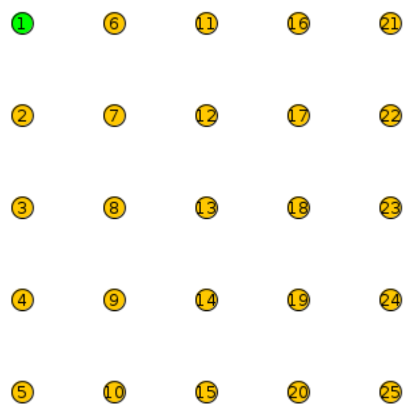


Figure 5.9: A 25-node network

The network energy consumption of the standard RPL is $35.653mJ$. We also run the proposed approach on all sensors, resulted in $31.02mJ$ energy consumption and 12% improvement.

5.3 Real Implementation

In order to test our ideas, we also implemented transmission power control on a pair of Zolertia Z1 motes: one *root* and one *client*. The nodes are located 2 meters apart and use the embedded antenna for communication. Client node sends 150 *hello packets* at the rate

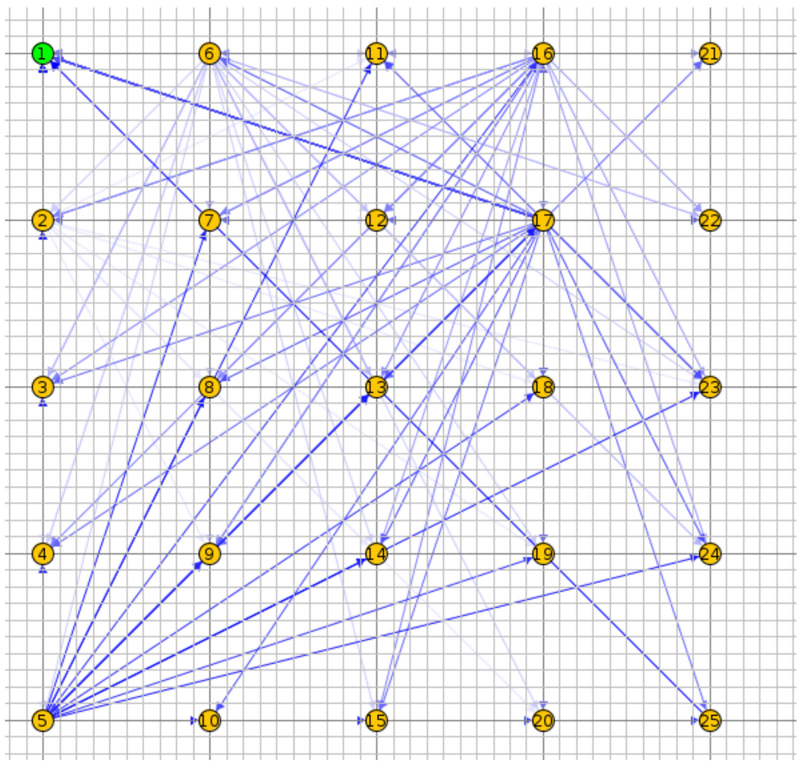


Figure 5.10: A snapshot of the experiment

of one packet per second rate.

Let t_i be the transmission time for sending $Packet_i$ and set $T_j = \{t_1, t_2, ..., t_{150}\}$ be set of transmission times for sending 150 packets when transmission power is j. We measure T_j, for all possible transmission power levels using by a Contiki application called Powertrace [9] that we describe next.

Because of radio duty cycling mechanisms in WSNs, estimating the energy consumption can be inaccurate. Powertrace is a run-time energy profiling utility that measures the time spent in each of several different activities, such as packet transmission, packet reception, and CPU active time. This allows us to precisely determine the time spent by mote in each activity. Recall that the Zolertia Z1 motes use CC2420 RF transceiver module which provides programmable transmission power[4]. If t_i and P denote the transmission time and transmission power, Equation 5.1 is used to compute the energy consumption to send a packet at that power level. Note that P is obtained from the CC2420 data sheet, as showing in Figure 4.1.

$$E_i = t_i * P \tag{5.1}$$

We use the *Bootstrapping method* to calculate the mean transmission energy and its confidence interval for sending one packet. The bootstrap is a non-parametric statistical method for computing confidence intervals [10].

The expected transmission energy to send a packet is shown in Figure 5.11. Note that the error bars are plotted in blue vertical line and are mostly too small to be seen, showing high confidence.

It is clear that the expected transmission energy is a convex function of transmission power level with a global minimum at $Tx = 7$. Hence, the optimal transmission power in this scenario is 7.

Comparing Figure 5.4 and Figure 5.11, we find a similar trend in both simulation and implementation: A convex behavior of E_s over Tx. Therefore, the simulation results and implementation results are consistent. Note that, because the experiment duration and number of transmitted packets are different, the minimum E_s occurs at different transmission powers: at $Tx = 7$ in implementation and at $Tx = 11$ in simulation. This

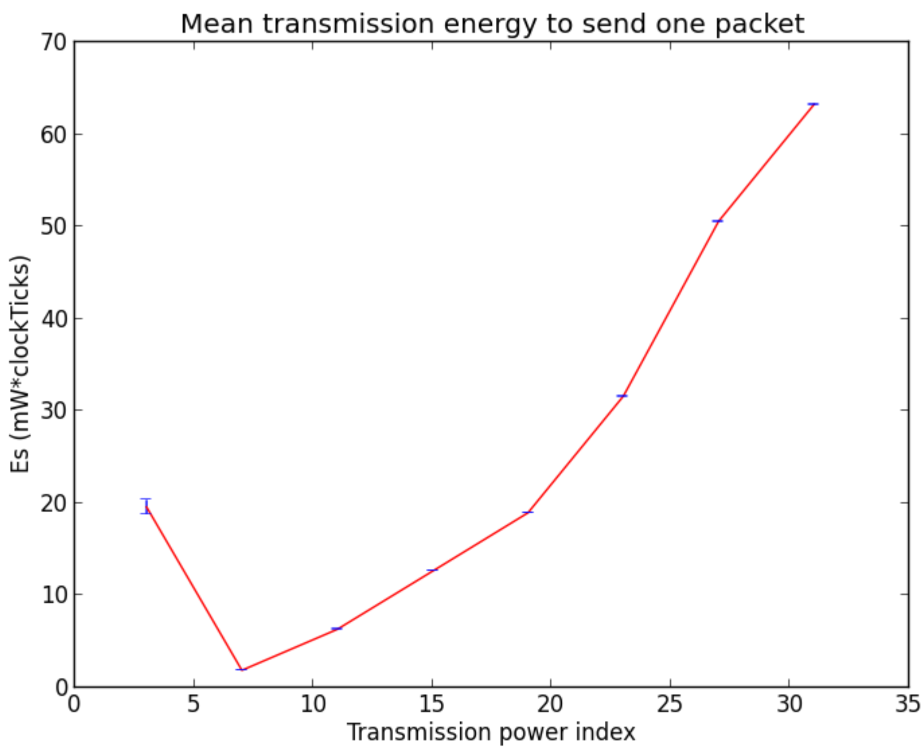


Figure 5.11: Expected transmission energy over transmission power per packet

experimentally validates our simulation results and indicates that the simulation results obtained for more complex topologies are likely to be fairly accurate.

Chapter 6

Conclusions and Future Work

This thesis presents techniques to improve the energy efficiency of the industry-standard RPL routing protocol. We take a top-down approach to design an objective function for RPL, as discussed in Chapter 2. Accordingly, our goal is to develop an energy-aware objective function for smart buildings. This objective function minimizes the expected transmission energy. We also propose a practical method to choose the transmission power level at each node such that the energy cost of data transmission to the root is minimized.

Our work has made one simplifying assumption. We assumed that the transmit power is the dominant radio power consumer. This is valid in some chips like XBee (its transmit power is 10 times more than its receive power [28]). However, some motes also use noticeable power in the receive mode in order to de-modulate the signal. We can include the receive power in the E_s formulation to make it accurate for motes with noticeable receive power.

6.1 Future work

6.1.1 Es Extension

Note that Objective Function Zero (OF0) specified in *RFC 6552* uses three specific parameters to implement the rank: *step-of-rank*, *rank-factor*, and *stretch-of-rank*[23].

The step-of-rank parameter is defined to indicate the link quality estimation, such as ETX value. The rank-factor parameter is defined to classify links. For example, we can distinguish between links with different properties, such as powered over battery-powered. The stretch-of-rank parameter allows an implementation to stretch the step-of-rank to place a node in a higher depth in the DAG. Thus, the parent set of a node is increased. Let Sr and Rf indicate the stretch-of-rank and the rank-factor respectively.

If $R(x)$ shows the rank of the node x and the node p is the parent of node x, Equation 6.2 explains the general rank computation implementation in RFC 6552 [23].

$$R(x) = R(p) + RankIncrease \tag{6.1}$$

$$RankIncrease = (Rf * Sp + Sr) * MinHopRankIncrease \tag{6.2}$$

We designed the E_s to minimize the transmission energy consumption. Thus, E_s increases the *energy efficiency* of a network. However, *energy consumption balancing* property aims for uniform energy dissipation in sensors. Let S_e denote the remaining energy of node x which is a *node metric*. Let E_s indicate the link metric of (x, ρ). We can consider energy efficiency and energy balancing together, if we look at E_s as the step-of-rank and S_e as the rank-factor. Therefore, we can combine link metric and node metric to find the efficient path while *network life time* is also maximized.

6.1.2 Exploration of ETX

Although the ETX estimation method of ContikiRPL does not have control and implementation overhead, it can be inefficient because it can keep using a bad/moderate quality link even though an alternative good quality link is available. Let ETX and ETX' indicate the real value of ETX and the estimated value of ETX respectively in Figure 6.1. Node 4 selects node 2 as its preferred parent. However, node 3 is a better parent. Node 4 makes this error because it does not monitor link quality on link $(4, 3)$.

To implement a probing-like behavior in RPL, links can be initialized optimistically to 1 [7] resulting in frequent changes to the preferred parent. This increases the churn and convergence time in the network.

RPL aims to find the paths with minimum ETX to save energy and other network resources. However, finding paths with minimum ETX itself requires nodes to probe links which is costly in terms of resources. Thus, finding the optimal solution for Problem 3.2

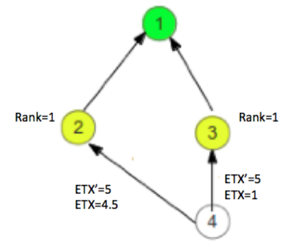


Figure 6.1: An example to show inefficiency of ContikiRPL ETX

can be expensive in terms of energy consumption and computation. However, we can add this probing cost itself to the energy minimization problem, and discussed next.

For each parent, $p \in P$, let $lq(x, p)$ denote the link quality estimation for the link (x, p). Let $At(x)$ be the node metric(attribute) of node x. The parent that maximized the objective function f is the best node to be the node x's parent. In order to find the best parent $\hat{p}$, node x needs to probe the neighbors while probing costs S. The probability of finding $\hat{p}$ is higher, if node x probes all its neighbors. Let α denote the coefficient of first term of the objective function f. Therefore, S and $\alpha * f(lq(p, x), At(p))$ are inversely correlated. Problem 2 formulates the preferred parent selection problem considering the searching cost. This would allow us to select the preferred parent, ρ, that is nearly as good as the best parent, $\hat{p}$, at a reasonable cost.

Problem 2

$$max\ (\alpha * f(lq(p, x), At(p)) - S)\ s.t. \tag{6.3a}$$

$$p \in P, \tag{6.3b}$$

$$At(p) > Threshold_1, \tag{6.3c}$$

$$lq(p, x) > Threshold_2 \tag{6.3d}$$

This problem can be modeled as an *active reinforcement learning* problem to find the trade-off between *exploitation* and *exploration*. Exploitation greedily maximizes the current utility estimates of an agent and exploration maximizes the long-term well-being or curiosity of an agent [20].

In future work, we propose to use two practical schemes to solve problem 2: *GLIE*[1] and

[1]Note that GLIE stands for Greedy in the limit of infinite exploration.

Gittins index [20]. In GLIE method, a node selects its preferred parent randomly $\frac{1}{t}$ [2] of the time and selects its preferred parent greedily rest of the time. A node can also weight its neighbors according to their RSSI.

The second approach is a Gittins index based method [20]. We can compute a Gittins index for each candidate parent based on number of probes sent to the parent and how much it has paid off. The pay off amount depends on the link's ETX. If we probe a good quality link the pay off value is positive, otherwise it is zero. Generally, selecting nodes with highest uncertainty and highest expected return is the optimal exploration policy. We intend to study this more closely in future work.

[2] t is a constant value.

References

[1] Hazrat Ali. A Performance Evaluation of RPL in Contiki. *Master's Thesis, Blekinge Institute of Technology, Sweden*, (ELnaz), 2012.

[2] M. A. Alim, M. M. Rahman, M. M. Hossain, and A. Al Nahid. Analysis of Large Scale Propagation Models for Mobile Communications in Urban Area. *International Journal of Computer Science and Information Security*, abs/1002.2:135–139, February 2010.

[3] Nouha Baccour, Anis Koubaa, Luca Mottola, Marco Antonio Zuniga, Habib Youssef, Carlo Alberto Boano, and Mario Alves. Radio link quality estimation in wireless sensor networks: a survey. *ACM Transactions on Sensor Networks (TOSN)*, 8(4):34, 2012.

[4] CC2420 Chipcon. CC2420 datasheet-2.4 GHz IEEE 802.15. 4/ZigBee-Ready RF Transceiver (Rev. B). Texas Instrument, 2007.

[5] Thomas Clausen, Ulrich Herberg, and Matthias Philipp. A critical evaluation of the IPv6 Routing Protocol for Low Power and Lossy Networks (RPL). In *2011 IEEE 7th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pages 365–372. IEEE, October 2011.

[6] Douglas S. J. De Couto, Daniel Aguayo, John Bicket, and Robert Morris. A high-throughput path metric for multi-hop wireless routing. *Wireless Networks*, 11(4):419–434, July 2005.

[7] Sébastien Dawans, Simon Duquennoy, and Olivier Bonaventure. On link estimation in dense RPL deployments. In *Local Computer Networks Workshops (LCN Workshops), 2012 IEEE 37th Conference on*, pages 952–955. IEEE, 2012.

[8] Adam Dunkels. Contiki: The Open Source Operating System for the Internet of Things. *http://www.contiki-os.org/*.

[9] Adam Dunkels, Joakim Eriksson, Niclas Finne, and Nicolas Tsiftes. Powertrace: Network-level power profiling for low-power wireless networks. Technical Report T2011:05, Swedish Institute of Computer Science, 2011.

[10] Bradley Efron and Robert J Tibshirani. *An introduction to the bootstrap*, volume 57. CRC press, 1994.

[11] Olfa Gaddour and Anis Koubâa. RPL in a nutshell: A survey. *Computer Networks*, 56(14):3163–3178, September 2012.

[12] Omprakash Gnawali, Rodrigo Fonseca, Kyle Jamieson, Maria Kazandjieva, David Moss, and Philip Levis. CTP: An efficient, robust, and reliable collection tree protocol for wireless sensor networks. *ACM Transactions on Sensor Networks*, 10(1):1–49, November 2013.

[13] Omprakash Gnawali and P Levis. The ETX Objective Function for RPL. *IETF Internet draft*, (Internet-Draft), 2010.

[14] Patrick Olivier Kamgueu, Emmanuel Nataf, Thomas Djotio Ndié, and Olivier Festor. Energy-based routing metric for RPL. January 2013.

[15] Jeonggil Ko, Joakim Eriksson, Nicolas Tsiftes, Stephen Dawson-Haggerty, Andreas Terzis, Adam Dunkels, and David Culler. Contikirpl and tinyrpl: Happy together. In *Workshop on Extending the Internet to Low Power and Lossy Networks (IP+ SN)*, 2011.

[16] Philip Levis, Neil Patel, David Culler, and Scott Shenker. Trickle: A self-regulating algorithm for code propagation and maintenance in wireless sensor networks. In *Proceedings of the 1st Conference on Symposium on Networked Systems Design and Implementation - Volume 1*, NSDI'04, pages 15–28, Berkeley, CA, USA, 2004. USENIX Association.

[17] Shan Lin, Jingbin Zhang, Gang Zhou, Lin Gu, John A Stankovic, and Tian He. ATPC: adaptive transmission power control for wireless sensor networks. In *Proceedings of the 4th international conference on Embedded networked sensor systems*, pages 223–236. ACM, 2006.

[18] Emmanuel Nataf and Olivier Festor. Online Estimation of Battery Lifetime for Wireless Sensors Network. *CoRR*, abs/1209.2234, September 2012.

[19] Tifenn Rault, Abdelmadjid Bouabdallah, and Yacine Challal. Energy-Efficiency in Wireless Sensor Networks: a top-down review approach. *Computer Networks*, 2014.

[20] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 2009.

[21] Kannan Srinivasan, Prabal Dutta, Arsalan Tavakoli, and Philip Levis. An empirical study of low-power wireless. *ACM Transactions on Sensor Networks*, 6(2):1–49, February 2010.

[22] Kannan Srinivasan and Philip Levis. RSSI is under appreciated. In *Proceedings of the Third Workshop on Embedded Networked Sensors*, pages 239–242, 2006.

[23] Pascal Thubert. Objective Function Zero for the Routing Protocol for Low-Power and Lossy Networks (RPL). *IETF Internet draft*, (RFC 6552), 2012.

[24] Nicolas Tsiftes, Joakim Eriksson, Niclas Finne, Fredrik Österlind, Joel Höglund, and Adam Dunkels. A framework for low-power IPv6 routing simulation, experimentation, and evaluation. *ACM SIGCOMM Computer Communication Review*, 40(4):479, August 2010.

[25] Jean-Philippe Vasseur and Adam Dunkels. *Interconnecting smart objects with ip: The next internet*. Morgan Kaufmann, 2010.

[26] Thomas Watteyne, Antonella Molinaro, Maria Grazia Richichi, and Mischa Dohler. From MANET To IETF ROLL Standardization: A Paradigm Shift in WSN Routing Protocols. *IEEE Communications Surveys & Tutorials*, 13(4):688–707, 2011.

[27] T Winter, P Thubert, and R Kelsey. RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks. *IETF Internet draft*, (RFC 6550):1–164, 2012.

[28] Datasheet XBee. XBee-PRO OEM RF Modules. *Product Manual v1. xAx-802.15*, 4, 2013.

[29] Zolertia. Z1 Datasheet. *http://www.zolertia.com/*, 2010.