

Delft University of Technology
Master's Thesis in Computer Science

Network Lifetime Analysis of Data Collection Protocols

Si Li



Network Lifetime Analysis of Data Collection Protocols

Master's Thesis in Computer Science

Embedded Software Section
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft University of Technology
Mekelweg 4, 2628 CD Delft, The Netherlands

Si Li
kuromagician@gmail.com

25th August 2014

Author

Si Li (kuromagician@gmail.com)

Title

Network Lifetime Analysis of Data Collection Protocols

MSc presentation

29th August 2014

Graduation Committee

Prof. Dr. K. G. Langendoen (Chair)

Delft University of Technology

M. A. Zúñiga Zamalloa, Ph.D. (Daily Supervisor)

Delft University of Technology

Dr. Z. Al-Ars (Core Member)

Delft University of Technology

Abstract

When a sensor network is deployed, we fundamentally care about three main outcomes: to obtain as much data as possible (high delivery rate), to obtain data as fast as possible (low latency), and to obtain data for as long as possible (long lifetime). This last metric, called network lifetime, is of great importance and has been widely investigated because sensor nodes are usually battery-operated. However, there is a gap between the many theoretical studies and the very few empirical ones. The aim of this thesis is to bridge that gap.

To achieve our aim, we analyze two well-known data collection protocols: one based on shortest-path trees, called CTP; and the other based on opportunistic routing, called ORW. Both protocols have advantages and disadvantages with respect to the network lifetime. On the one hand, CTP reduces the total number of transmissions in the network, but uses an expensive communication primitive and does not care about load balancing. On the other hand, ORW has the exact opposite characteristics, good load balancing with an efficient communication primitive at the cost of increasing the total number of transmissions. There is hence an open question to solve: *which protocol provides longer lifetimes?*

We tackle the problem from an analytical and a practical perspective. For the analytical part, we improve the accuracy of current energy models for CTP and develop a new energy model for ORW. Our models for CTP are up to 95% more accurate than the state-of-the-art. For the empirical part, we evaluate both protocols on a public testbed with 100 nodes.

Our analytical results show that ORW has longer lifetimes than CTP for high density networks, and that this advantage should vanish in low density networks. Our empirical results validate that ORW is indeed better than CTP under high densities, but for lower densities, our experiments actually show that ORW performs significantly worse than CTP. We show that this unexpected behaviour (according to the model) is due to some inherent flaws in the implementation of ORW.

Preface

This thesis is a step towards understanding network lifetime with different protocols in wireless sensor networks (WSNs). I have always been interested in learning more about WSNs, but it was only during the work of my thesis that I had the opportunity to come in close contact with this research area. I feel lucky to have had the chance to investigate state-of-art protocols in a way that has not been investigated before. I hope this work provides hints for improving the current protocols, as well as for designing future ones.

First I would like to thank my supervisor, Marco Zúñiga. It is his patient guidance and great support that helped me finish my thesis. I learned not only knowledge, but also the way to conduct research and present ideas, which will benefit me throughout my life. I also want to acknowledge Koen Langendoen for hosting me at the Embedded Software group, and Zaid Al-Ars for being a member of my committee. Furthermore, I want to thank to my friends, Yan Li and Namitha, your reviews helped in further improving the quality of my thesis. Finally I am very grateful to my family, your unconditional support and gentle comfort helped me overcome the difficulties in my life.

Si Li

Delft, The Netherlands
25th August 2014

Contents

Preface	v
1 Introduction	1
1.1 Network Lifetime and its Key Components	2
1.2 Challenges	2
1.3 Problem Statement	3
1.4 Contributions	3
1.5 Thesis Organization	3
2 Related Work	5
2.1 Research on Network lifetime	5
2.1.1 Definitions of Network lifetime	5
2.1.2 Efforts to extend network lifetime	6
2.2 Energy Models for Wireless Sensor Network	7
2.3 CTP and ORW	8
2.3.1 BoX-MAC-2	8
2.3.2 Collection Tree Protocol	8
2.3.3 Opportunistic Routing in WSN	9
2.3.4 Pros and Cons	10
3 Duty Cycle Model	13
3.1 Model for CTP	13
3.1.1 Existing Model	14
3.1.2 Improved Model	17
3.2 Model for ORW	20
3.3 Comparison of models	22
4 Empirical Results and Analysis	25
4.1 Experimental Setup	25
4.1.1 General Conditions	25
4.1.2 Testbed Specification	26
4.2 Model Validation	26
4.2.1 Results for CTP	27
4.2.2 Results for ORW	28

4.3	Experiments on network lifetime	30
4.3.1	Influence of Energy Budget	30
4.3.2	Influence of Network Density	33
4.3.3	Influence of Network Scale	34
4.4	Discussion on Peculiar Phenomenon	35
5	Conclusions and Future Work	39
5.1	Conclusions	39
5.2	Future Work	39

Chapter 1

Introduction

After emerging a decade ago, wireless sensor networks (WSN) have distinguished themselves as a key technology in a wide range of areas, such as environmental monitoring [20], precision agriculture [2], military surveillance [13], health care [24] and so on.

Arguably, one of the most important applications of sensor networks in all these areas is *data collection*. In data collection applications, nodes are deployed in the scenario of interest, and they need to transmit information in a multi-hop manner to a particular node in the network called *sink*. Figure 1.1 depicts the simple idea behind data collection.

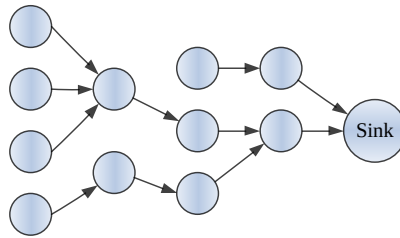


Figure 1.1: An example for data collection applications

Once deployed, we would like the network to deliver data for as long as possible. That is, we would like to maximize the network's lifetime. In most WSN applications, maximizing the network's lifetime is central because nodes usually do not have access to a continuous power supply, and they have to run on batteries to fulfill the given task. Limited by the size of nodes, batteries cannot be too big, and hence, energy becomes a scarce resource where every joule should be spent discreetly [1].

1.1 Network Lifetime and its Key Components

The network lifetime is a function of the lifetime of individual nodes, and nodes belonging to the same network can have widely different lifetimes. Given the dependence on individual node lifetimes, the network lifetime can be defined in many ways. Common definitions include the time passed until the first node in the network dies, or the time when the last node dies.

The lifetime of a node depends on two components: the initial energy level of the batteries and the individual computation, sensing and communication loads. The computation load is usually negligible and most sensors are very energy efficient. Hence, wireless communication is usually the most energy-hungry component.

Considering that radio communication is the main energy drain, the lifetime of the network depends on three main aspects: (i) the cost of transmitting a packet from one node to its neighbor (communication primitive), (ii) the total amount of packets transmitted by all nodes in the network (transmission load), and (iii) the even distribution of packet transmissions among nodes (load balance). Many works, mainly theoretical, have proposed a wide range of techniques to optimize the load balance of the network. But, these studies do not consider the different types of communication primitives present in actual protocols, and do not consider either that specific implementations can change the total number of transmissions in the network.

1.2 Challenges

The aim of this thesis is to study the lifetime of current protocols considering—in a comprehensive manner—: communication primitives, transmission loads and load balancing. Analyzing the lifetime of state-of-the-art protocols is not trivial because they follow conflicting design guidelines. The de-facto standard collection protocol used for the past 10 years, *Collection Tree Protocol (CTP)* [11], aims at minimizing the number of transmissions rather than to achieve load balancing or to use efficient communication primitives. Under these circumstances, a few nodes end up being heavily loaded because they have to perform many (and expensive) transmissions. Overall, these design guidelines lead to a fast depletion of energy in those heavily loaded nodes, which is especially detrimental for the lifetime of large scale networks [18]. On the other hand, recent protocols, like *Opportunistic Routing for Wireless Sensor Networks (ORW)* [16], obtain a more balanced routing by using efficient communication primitives, but at the cost of increasing the number of transmissions (compared to CTP). Given that ORW is a more recent protocol, it has not been used as widely as CTP and it has not been thoroughly investigated either.

1.3 Problem Statement

Considering the current situation, there is an open question that has not been clearly answered yet:

Which data collection method is better in terms of network lifetime? One prioritizing number of transmissions over communication efficiency and load balance (CTP) or vice versa (ORW)?

The problem is further complicated because network lifetime is not a clearly defined terminology. Depending on the specific application, different definitions can be applied. Thus it would be interesting to investigate how these two protocols behave across these various definitions. In this thesis, we do exactly that, we consider the entire spectrum of network lifetime: from the first node that dies, to the last one, including all the fraction of nodes within these two extremes.

1.4 Contributions

This work contains mainly two parts. First, an analytical framework is proposed to understand the cost of communication primitives and load balance in CTP and ORW. Second, experiments are conducted in testbeds to reveal insights about the death process of a network. The overall contributions are:

- Improving the existing energy model for CTP, which enhances the accuracy of the model by up to 95% (Section 3.1).
- Creation of a new model for ORW (Section 3.2), and a comparison between the CTP and ORW models (Section 3.3).
- Showing that ORW has longer lifetimes in high density networks, while CTP is the preferable choice in low density networks (Section 4.3).
- Gaining insights about why ORW performs worse in low density network (Section 4.4).

1.5 Thesis Organization

The remainder of this thesis is organized as follows. We first introduce the related work on network lifetime, energy models, as well as a description of the operation of CTP and ORW in Chapter 2. Then, we present the state-of-the-art (SoA) model and our improved versions in Chapter 3. Next, we provide empirical results validating our analysis in Chapter 4. Last, we conclude our work in Chapter 5.

Chapter 2

Related Work

This chapter discusses the works related to our topic. We first introduce studies investigating the network lifetime in WSNs. Then, we briefly describe the design and operation of CTP and ORW.

2.1 Research on Network lifetime

Network lifetime is one of the most important parameters that needs to be maximized in WSNs. Several studies address this issue, but most of the time they come up with their own definition of network lifetime according to the targeted application. In this section we provide a brief overview of some of these definitions and related efforts to maximize the network lifetime.

2.1.1 Definitions of Network lifetime

- (1) Network lifetime based on number of nodes alive

This is one of the most frequently adopted definitions for network lifetime. And within this definition, many works simply consider the time when the first node dies [21][28]. This definition is sometimes too conservative, as a WSN can still provide a lot of data after the death of the first node – unless every node is critical and the application cannot afford to lose even one node. Some other works, such as [14], define the network lifetime as the time when a fraction of the nodes (f) in the network die. The precise value of f depends on the application itself. This definition is the most appropriate for many of the applications targeted by sensor networks. The most extreme definition of network lifetime is the time when all nodes die [27]. But this last definition is not suitable when network partitions occur. For example, if an operating node does not have a path to the sink, its remaining energy is of no use, but it is still counted as an effective node.

- (2) Network lifetime based on number of nodes connected to the sink

Compared to the previous definitions, this one fits more closely the need of data collection applications. Some studies further argue that the importance of the node should be also considered in the definition of network lifetime [8]. But the need to include the node’s importance only applies if there are some special mechanisms running on the network, like data aggregation.

There are many other network lifetime definitions, but they are either bounded to specific applications [15] or hard to trace [3]. For example, there is a definition based on sensor coverage where there are only a few critical areas of interest to monitor [4], but we do not want to consider such definitions because they are too application specific.

In our work, we will use the definition of network lifetime that focuses on the fraction of dead nodes. Considering a network of n nodes, we will monitor the entire spectrum, going from $1/n$ to n/n . At the same time, we monitor how many of these nodes are still connected to the sink.

2.1.2 Efforts to extend network lifetime

Our work focuses on the analysis of two well known protocols rather than on coming up with a new approach to extend network lifetime. Nevertheless, it is important to describe some of the key techniques used to extend network lifetime, and relate them to the two protocols evaluated in our study.

The first important observation made by the research community is that nodes that are neighbors to the sink are the first ones to deplete their energy. This occurs because, in principle, the closer to the sink, the higher the load nodes have to forward. To overcome this problem, studies have proposed special deployment of nodes (e.g., to deploy more nodes in areas with high traffic) and topology control mechanisms to achieve load balancing [5]. It is important to mention that most of these studies were theoretical. In practice, ORW follow this ‘theoretical’ energy depletion pattern (nodes closer to the sink deplete their energy faster), but CTP does not. CTP constructs a sort of backbone which only loads heavily a few of the nodes close to the sink, and leave other nodes with a low load.

Another important group of methods aimed at maximizing the network lifetime is known as energy aware routing [6]. Those methods try to reach maximal network lifetime by minimizing the energy consumption for a packet to reach its destination. Later works, such as [26], further take the energy remaining in the node into consideration to construct paths —nodes with higher energy reserves are more likely to be selected as forwarders. Neither CTP or ORW take energy-aware methods into account.

Finally, a key observation made for static networks is that no matter what routing protocol is used, it does not directly lead to a good load balancing in the network: nodes around the sink will always have higher load than the

other parts of the network [19]. Thus a mobile sink is proposed to achieve load balancing for the whole network. CTP and ORW are not designed for mobile sinks, hence, this thesis focuses on the analysis of static networks.

2.2 Energy Models for Wireless Sensor Network

In order to investigate the energy consumption in WSNs, many works either adopt or create models to provide foundations for further analysis. There are two main types of models:

(1) Constructed from micro details

Models belonging to this type usually consider many low level details, such as radio fading effect, transmission power, energy consumption from every part of the node, etc. They often use real energy values as a metric, i.e. they calculate how many Joules the node would consume. For example, a widely used and simple model ([21, 17, 12]) has a form similar to:

$$\begin{aligned} P_{sense} &= \alpha_1 b \\ P_{TX} &= (\beta_1 + \beta_2 r^n) b \\ P_{RX} &= \gamma_1 b \end{aligned}$$

where P_{sense} , P_{TX} and P_{RX} stand for power consumption of sensing, transmitting and receiving, respectively. The variable b (in bits/sec) is data rate of the node, and r^n stands for path loss. β_1 and γ_1 represent how much energy is dissipated by the transmitter and receiver.

(2) Constructed from high level statistics

These models usually do not pay much attention to low level details. They are more interested in macro phenomena, and often employ indirect ways to measure the energy consumption instead of actually measuring it, [25] is an example. That model focuses on networks where low power listening (LPL) is applied. In LPL, all nodes are duty-cycled, namely every node sleeps for most of the time, and only wake up for a short period to check the radio transmission to save energy. Since usually radio communication is the major source of energy consumption in sensor nodes [29], several energy models use radio duty-cycle (i.e. the fraction of time that the radio is ‘on’) as an indicator of energy consumption.

In our case, we are constrained by the limitation of large-scale public testbeds which do not allow direct measurement of energy consumption. But high level metrics such as the duty-cycle used in [25] are available. Therefore we choose a high-level model in our work.

2.3 CTP and ORW

In this section we will first introduce BoX-MAC-2 [22], the Medium Access Control (MAC) scheme for both protocols in our work. Then we briefly describe the two protocols we investigate in this thesis.

2.3.1 BoX-MAC-2

Considering that energy consumption is a key concern in WSNs and that radio communication accounts for most of the consumed energy, sensor networks use Medium Access Protocols that keep the radio ‘off’ most of the time. One of the most popular protocols is BoX-MAC-2. This MAC protocol is used by CTP and ORW, and proposes a way to transmit packets in an energy efficient manner:

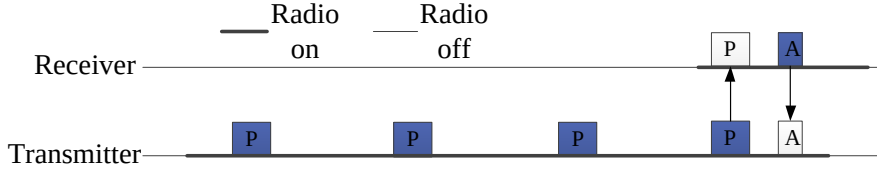


Figure 2.1: Transmission strategy of BoX-MAC-2

As shown in Figure 2.1, BoX-MAC-2 is a packet based MAC. Nodes wake up every wake-up period t_w to check if there is a packet intended from them. Since the transmitter does not have any knowledge about when the receiver will wake up in an asynchronous network, it will send the complete packet repeatedly until the receiver detects the packet and sends back an acknowledgement. Once the acknowledgement is received, the transmission is considered complete. If a node wants to do a broadcast, i.e. to send a packet to all its neighbors, then the transmitter needs to send the strobe of packets for an the entire wake-up period t_w to ensure that every neighbor receives the packet. Hence, in term of energy, broadcast communication costs more than unicasts.

2.3.2 Collection Tree Protocol

CTP, the de-facto standard protocol for data collection applications for the past ten years, has been investigated intensively. It is designed to be highly reliable, robust, efficient and hardware independent [11]. CTP works as follows.

Initially nodes are deployed and they broadcast messages to discover their neighbors and paths to the sink. Nodes estimate their shortest path to the

sink via a metric called ETX [7], which stands for Expected Number of Transmissions. An ETX of 1 means that on average, to successfully deliver a packet, a node needs only one transmission, while an ETX of 2 means that it needs 2 transmissions to deliver a packet. Hence, a lower ETX means a better path. A node's ETX is equal its parent's ETX plus the ETX of the link to this parent. Each node keeps track of its neighbors' ETX values, and chooses the node with the smallest ETX as its parent.

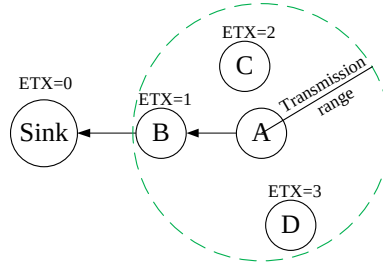


Figure 2.2: Example unicast

Figure 2.2 illustrates the concept of ETX: there are 3 nodes, B, C and D within the transmission range of node A. The sink has an ETX of 0, node B who is the closest node to the sink has an ETX of 1, which means when A wants to deliver a packet to the sink, it will choose the node with the lowest ETX, namely node B as the next hop. Overall, the aim of the ETX metric is to minimise the number of transmissions made by nodes, which is a good thing to do to extend the lifetime of the network. But as we will see, this is achieved using a costly communication primitive (unicasts) and by overloading some of the nodes.

In order to maintain the protocol's resilience to dynamics in the network, nodes broadcast periodic beacons to send updated values of their own ETX to keep the best routes up to date.

2.3.3 Opportunistic Routing in WSN

ORW is a new protocol, proposed in 2012, that uses a completely different approach to build routes, as compared to CTP. ORW works as follows:

After deployment, the sink node sends broadcast beacons to form a gradient around itself. After receiving these beacons, the nodes will perform the same broadcast to propagate the routing information. The metric used in ORW is called Expected Duty Cycle (EDC), which estimates the expected duty cycle needed to reach the sink. Notice that compared to CTP, ORW aims at reducing the duty cycle and not the number of transmissions.

When forwarding a packet, ORW follows an opportunistic approach. Once a node needs to transmit a packet, instead of choosing an specific parent with

the smallest EDC, a node will just send out the packet with its own EDC and a predefined EDC-threshold for the next hop. Any node that wakes up first with an EDC value that is lower than the predefined EDC threshold, receives the packet, acknowledges it and forwards the packet to the next node following the same procedure [10]. This communication primitive is called anycast.

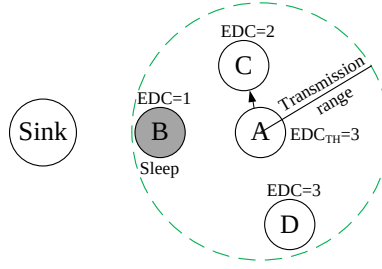


Figure 2.3: Example anycast

Figure 2.3 illustrates the concept of anycast: a packet generated by A has a forwarding threshold of 3 in terms of EDC. Any neighbor having an EDC lower than 3 is qualified to forward the packet. Hence, under this condition, nodes B and C both satisfy the criteria. If node C wakes up before B, A will forward the packet to C, even though node B has a lower EDC. If node B wakes up before node C, node B will forward the packet. If node D wakes up before B and C, it will silently ignore the packet. Thus, at any given time, ORW chooses a good route, not necessarily the best one. While this opportunistic method may increase the path length, and hence, the number of transmissions, it has the advantage of having a more balanced load and a using a more efficient communication primitive.

Since every packet in ORW contains routing information in the header fields, nodes can update their own routing information by overhearing the transmissions occurring on the channel.

2.3.4 Pros and Cons

In Section 1.2 we mentioned three key aspects influencing the network lifetime: communication primitives, transmission loads and load balancing. We will now discuss the pros and cons of CTP and ORW regarding these three aspects.

For CTP we have:

- Communication primitives: Cons
In CTP, a node chooses the node with the lowest ETX as its parent. This makes the node stick to a specific parent, but at the cost of

using expensive unicast transmissions, since a node has to keep on transmitting until *the* particular parent wakes up. The time it takes for a node to wait for the parent's wakeup is called the *rendezvous time*; CTP's unicast leads to long rendezvous times. Furthermore, the regular broadcasts required to maintain the shortest paths to the sink also decreases the lifetime of the nodes.

- Transmission loads: Pros
CTP's routing metric, ETX, aims at minimizing the number of transmissions for a packet to reach the sink. With less transmissions, nodes consume less energy. Thus, network lifetime can benefit from this.
- Load balancing: Cons
If some nodes have a very low ETX, other nodes will tend to select these nodes as their parents, imposing an excessive load on them. This may make such nodes deplete their energy too fast.

For ORW we have:

- Communication primitives: Pros
In ORW, the anycast mechanism utilizes multiple nodes as candidates to forward the packets. Thus a node only needs to transmit the strobe of packets until any of the candidates wakes up. This significantly reduces the expected rendezvous time, which makes anycast a more efficient communication primitive compared to unicast.
- Transmission loads: Cons
ORW's opportunistic nature often leads to a higher number of transmission compared to CTP, as it does not always choose the shortest and best paths to the sink.
- Load balancing: Pros
The opportunistic nature of ORW means that packets choose different paths rather than fixed ones. This vast variety in routing paths leads to a more balanced load compared to CTP.

Based on the above discussion it is not trivial to identify which protocol is better in terms of network lifetime. In the next chapter we derive some simple probabilistic models to compare both protocols.

Chapter 3

Duty Cycle Model

Considering that most public testbeds do not allow direct measurements of energy consumption, researchers have proposed indirect methods to estimate the energy consumption of nodes. By and large, the most widely used method to estimate energy consumption is to keep track of the duty cycle of the radio. As explained in Section 1, in most scenarios radio communication accounts for most of the energy drain, hence, by keeping track of the percentage of a time a radio is kept ‘on’, i.e. by keeping track of its duty cycle, we can estimate the node’s energy consumption.

In this chapter, we introduce the existing duty cycle model for CTP and show how we improve the model. Then we exploit the new CTP model to create a model for ORW, after which a comparison is made between the two models. Throughout our work, we consider that both protocols employ the low power listening method defined by BOX-MAC-2 (described in Section 2.3).

3.1 Model for CTP

Initially, most theoretical studies assumed a very simple model for the radio. The radio was assumed to be always ‘on’, consuming a constant energy in this stage, and an extra consumption of energy was added during packet transmissions. The extra energy used during transmissions depended on the distance between the transmitter and receiver, the longer the distance, the higher the energy used. Once BOX-MAC-2 was designed, this model became obsolete for two reasons. First, in BOX-MAC the radio is kept mainly ‘off’, and second, there is not much difference between the energy used for transmission and reception, because the distances covered by sensor nodes are very short. The authors of BOX-MAC [22] developed a simple duty cycle model for low power listening, and later, the authors of the Broadcast Free Collection Protocol [25] proposed a refined version after considering CTP’s features. These models will be explained in more detail later.

Our models and the SoA models assume that all nodes are duty cycled with the same wakeup interval except the sink, which is always on. Having the sink always 'on' is a fair and common assumption, since most sensor networks deployments actually do this in practice.

We mentioned in the introduction that nodes in the same network can have widely different lifetimes. In order to capture this difference, nodes are divided into three groups, as shown in Figure 3.1: sink neighbors (nodes within one hop from sink), leaves (nodes with forwarding load smaller than 1.5¹) and relays (the rest of the nodes).

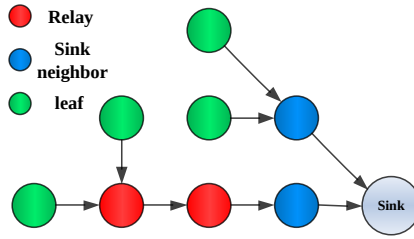


Figure 3.1: Node division

Given that our work extends the model described in [25], we will first describe that model.

3.1.1 Existing Model

The model introduced in [25] considers the five basic communication primitives contributing to CTP's duty cycle: (1) CCA (clear channel assessment) events, (2) Beacon transmissions (3) Beacon receptions, (4) Unicast transmissions, and (5) Unicast receptions. We will now introduce the five aforementioned primitives in detail. Table 3.1 summarizes our notation, and Figure 3.2 and Figure 3.3 will help us understand the beacon and unicast primitives.

(1) CCA events

Nodes keep their radios off most of the time but they wakeup at every interval t_w to perform a channel assessment. The goal of this channel assessment is to see if there are any packets being transmitted in the channel. Denoting t_c as the duration of the channel assessment, the contribution of CCA events to the duty cycle can be denoted as

$$\Delta_{rc} = \frac{t_c}{t_w} \quad (3.1)$$

¹Ideally a leaf node should have a forwarding load of one, namely only forwarding its own packet. But in practice a leaf node will occasionally forward other node's packet. We use this threshold to separate real leaves and relays

Symbol	Value	Description
t_w	0.25s ~ 1s	Time of wake up interval
t_c	12.5 ms	Time required to perform CCA
t_{rx}	25/35 ms	Time for receiving a packet ²
t_{tx}	26/36 ms	Time required to send a packet by the sink's neighbors
t_{IBI}	8 min	Time of beacon interval. Under stable status, one node will send a beacon every 8 minutes
t_{IPI}	1 min	Time of packet interval. Throughout our experiment it's set to 1 minute

Table 3.1: Symbols and their default values

(2) *Beacon transmissions*

At every interval t_{IBI} , nodes send out beacons to broadcast their routing status. Given that nodes wake up every t_w to check the channel, nodes have to transmit their beacons for a duration of t_w to ensure that every neighbor receives the beacon, as shown is Figure 3.2. Hence, the contribution of beacon transmissions to the duty cycle is

$$\Delta_{bs} = \frac{t_w}{t_{IBI}} \quad (3.2)$$

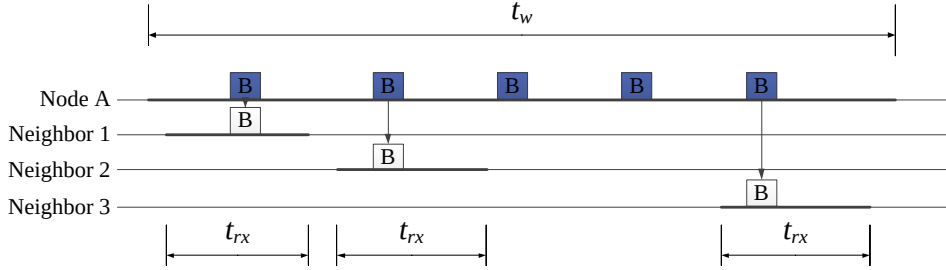


Figure 3.2: A typical broadcast event in CTP

(3) *Beacon receptions*

When the network enters a steady routing state, the number of beacons received within a beacon interval t_{IBI} should be equal to the number of neighbors node i has. Denoting t_{rx} as the time required to receive a packet (shown in Figure 3.2), and N_i as the number of neighbors, the contribution of these reception events to the duty cycle is

$$\Delta_{br} = \frac{t_{rx}}{t_{IBI}} N_i \quad (3.3)$$

²By default, t_{rx} is 25 ms for CTP, and 35 ms for ORW on average.

(4) *Unicast transmissions*

In CTP, every node has a parent that it should forward information to until the information reaches the sink. A node will either transmit its own packet (generated every t_{IPI}), or forward the packets that are generated by other nodes. Figure 3.3 shows a typical unicast process with BOX-MAC. A child node wants to send a packet to its parent node. Any other nodes hearing the ongoing transmission will extend the 'on' time of the radio, but will ignore the packet at the end (because the packet is not intended for that specific node). When the parent node wakes up and detects the packet, it will send an acknowledgement. This process is repeated at each hop until the packet reaches the sink.

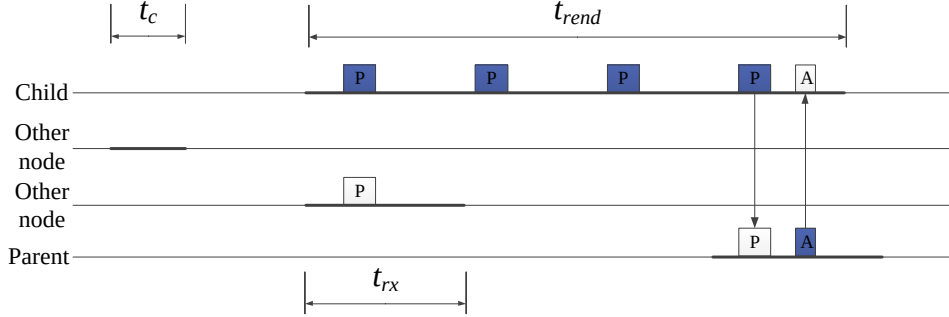


Figure 3.3: A typical unicast transmission in CTP. The bold lines represent the radio 'on' time of the nodes.

This primitive's contribution to the duty cycle is related to two parameters: the amount of time the radio needs to be on until the parent wakes up (rendezvous time), and the link quality (if the link is of poor quality the radio will need to be kept on for a longer time to accommodate retransmissions). To simplify the problem and to capture the essence of low power listening methods, here we assume that all links are perfect, i.e. 100% reliable. This is a fair assumption, because CTP tends to select very good links. Considering that the wakeup time of a potential parent is uniformly distributed within t_w , the expected rendezvous time is $t_w/2$. Denotting F_i as the forwarding load of node i , the contribution to unicast transmissions to the duty cycle is:

$$\Delta_{us}^i = \frac{t_w/2}{t_{IPI}} F_i \quad (3.4)$$

The above equation is only valid for relay nodes and leaf nodes. Considering that the sink's neighbors do not need to wait for the sink to wake up (because the sink is always on), the unicast sending time is just t_{tx} . Hence, the contribution to unicast transmissions for the sink's

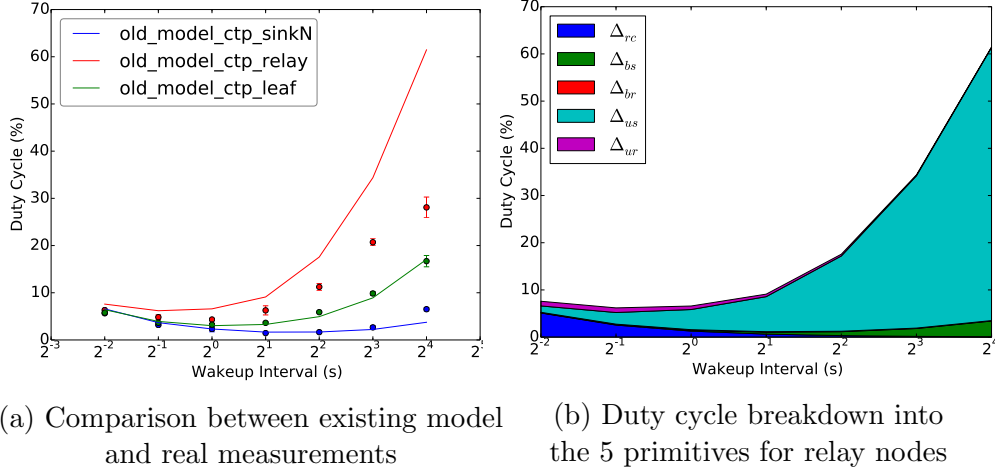


Figure 3.4: Existing model's performance

neighbors is:

$$\Delta_{us}^i = \frac{t_{tx}}{t_{IPI}} F_i$$

(5) *Unicast receptions*

Letting L_i denote the total number of packet receptions (both intended and unintended) at node i , the contribution of unicast receptions to the duty cycle is

$$\Delta_{ur}^i = \frac{t_{rx}}{t_{IPI}} L_i \quad (3.5)$$

Summing up Equations 3.1 to 3.5 we get the final expression for the overall duty cycle of a node:

$$\Delta_{dc}^{CTP} = \Delta_{rc} + \Delta_{bs} + \Delta_{br} + \Delta_{us} + \Delta_{ur} \quad (3.6)$$

3.1.2 Improved Model

We found that the previous model is not that accurate. We conducted experiments in the Indriya testbed [9] with different wakeup intervals t_w . During the experiments we monitored the duty cycle of nodes (using an internal timer to track the time the radio was on), and we also collected information for the required parameters in the model, namely F_i , N_i and L_i . Figure 3.4(a) depicts a clear difference between the modeled values and real measurements. Despite the good match that model's parameters have for leaves and sink neighbors, we observe a dissimilarity as high as 148% for relay nodes. But relays are the most critical group of nodes for network lifetime due to its high forwarding load, and hence we need a more accurate model for them.

After breaking down the energy consumption of nodes into the 5 primitives mentioned in Section 3.1.1, we obtain Figure 3.4(b). We can clearly see that Δ_{us} , namely *unicast transmissions*, constitute the major part of the duty cycle. Thus we can draw the conclusion that the duty cycle of these events is overestimated.

The reason for this overestimation is shown in Figure 3.5. The effect depicted in this figure is also mentioned in [22] but it was neglected and not included in the model:

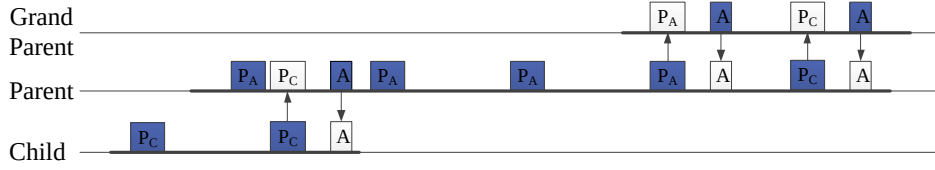


Figure 3.5: One transmission with multiple packets

To explain this phenomenon, we used a two hop communication consisting of a child, parent and grandparent nodes. Before the parent node successfully delivers its packet to the grandparent node, the child node generates a packet and sends it to the parent. At this time, the parent node will first acknowledge the child's packet, and then put the packet into its queue. Once the parent and grandparent establish a connection, the parent node will transmit both packets during a single session (instead of using two sessions: one for each packet). Since the rendezvous time (t_w) is much longer than the time required to transmit a packet, this event is equivalent to transmitting two packets for the cost of one. Due to this event, the forwarding load F_i used in Equation 3.4 is inappropriate and overestimated.

We update the SoA equations with this effect. First, let us assume that during one rendezvous session a node can transmit f_{extra} additional packets besides the original one. Then, Equation 3.4 becomes

$$\Delta_{us} = \frac{t_w/2}{t_{IPI}} \frac{F_i}{1 + f_{extra}} \quad (3.7)$$

We now model the circumstances under which f_{extra} occurs. This effect can be divided into the following two scenarios, illustrated in Figure 3.6:

- (1) If the child node wakes up before the parent node, then it has a window of opportunity of at most t_w to transmit its packet to the parent during the current wakeup period. This window of opportunity is denoted as the blue area in Figure 3.6. Any time longer than t_w will result in letting the parent node receive the packet in the previous wakeup period.

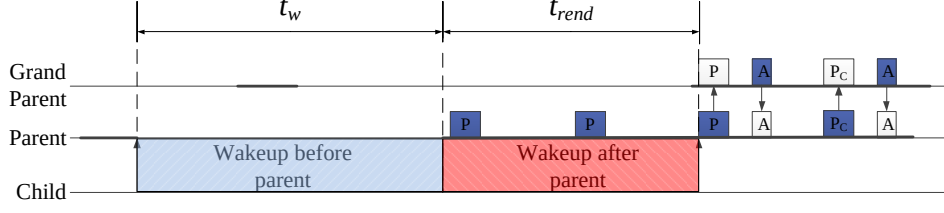


Figure 3.6: The timing that triggers “one transmission with multiple packets” for CTP

- (2) If the child node wakes up after the parent node, then the parent node has an expected rendezvous time of $t_{rend} = t_w/2$ before transmitting its packet to the grandparent. Hence, the child node must wake up within this period, i.e within the red area. Otherwise, the child would need to wait for the subsequent wake up period.

Thus the total time available for the child node to inject a packet to cause a “multi-packet transmission” is $t_w + \frac{1}{2}t_w$. Considering that a node generates a packet every t_{IPI} , the corresponding probability for this phenomenon to occur is $p = \frac{t_w + \frac{1}{2}t_w}{t_{IPI}}$. Letting Q be the size of the node’s transmission queue, and D_i be the number of children of node i , denoting $S = \min(Q, D_i)$, the expected value of f_{extra} is given by:

$$f_{extra} = \sum_{k=1}^S k C_{D_i}^k p^k (1-p)^{D_i-k}$$

But considering that the child node itself may also hold multiple packets, we can use the actual forwarding load F_i to substitute D_i ³. Then we get:

$$f_{extra} = \sum_{k=1}^S k C_{F_i}^k p^k (1-p)^{F_i-k} \quad (3.8)$$

We now can get a new model by substituting Equation 3.8 into Equation 3.7 to replace Equation 3.4. Figure 4.2 shows the results of applying our new model. For the sink’s neighbors the values remain unchanged because the sink is always “on”, and leaves are almost not influenced (for they seldom forward packets). But Equation 3.7 evidently improves the quality of the model for relay nodes.

³Since the computation of combinatorials require an integer, we use the rounding result of F_i . But for convenience, we still use the symbol F_i , and now $S = \min(Q, F_i)$.

3.2 Model for ORW

For ORW, the model is quite similar to the one of CTP, except for the fact that ORW doesn't have beacon events. Strictly speaking, ORW does have something similar to beacons, but it is only used at the very beginning and when a node cannot find a route. Thus it can be neglected. CTP on the other hand, uses beacons aggressively at the beginning, and once the routing topology is formed, it uses beacons every eight minutes to maintain the routes. Similar to what we did in Section 3.1, we propose the following duty cycle model for ORW:

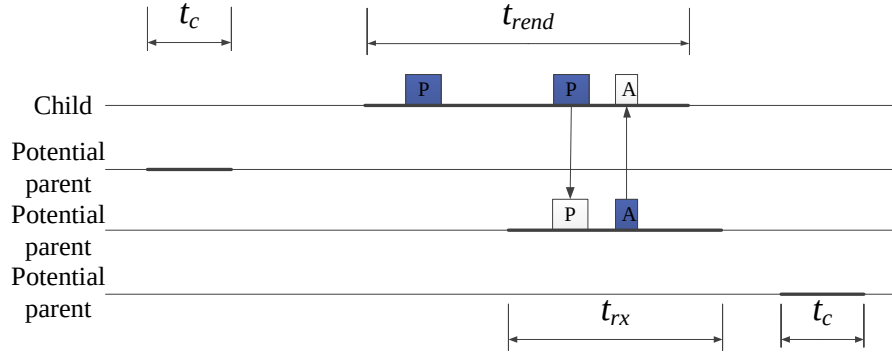


Figure 3.7: A typical anycast transmission in ORW. The bold lines represent the radio 'on' time of the nodes.

(1) *CCA events*

Same as CTP. Please refer to equation (3.1).

(2) *Anycast transmissions*

Figure 3.7 shows a typical anycast process. Compared to unicast, nodes in ORW will utilize any node that (1) wakes up first and (2) provides routing progress towards the sink (instead of selecting a fixed parent). In this way, ORW significantly shortens the rendezvous time because it does not need to wait until the *designated* parent wakes up. The disadvantage of this routing method is that ORW tends to use routes that are longer than CTP's, which increase the total number of transmissions in the network. Letting P_i be the number of potential parents of node i , it can be proved that the expected rendezvous time is $\frac{t_w}{1+P_i}$ if we assume 100% reliable links [10].

ORW also has the same "multi-packet transmission" effect mentioned in Section 3.1.2, because both protocols adopt the same MAC. Figure 3.8 depicts the effect using the same classification used for CTP (before and after the wake up of the parent):

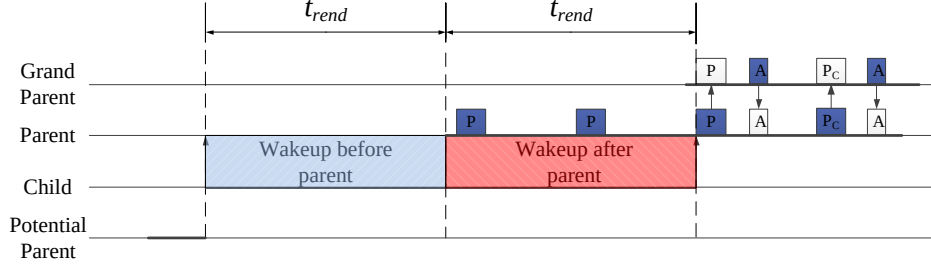


Figure 3.8: The timing that triggers “one transmission with multiple packets” for ORW

(1) The child node wakes up before a potential parent. To trigger the multi packet effect in ORW, a node can wakeup at most t_{rend} before its potential parent. Any time longer than t_{rend} will result in letting another potential parent acquire the packet.

(2) The child node wakes up after its potential parent. Under this circumstance, a node also has t_{rend} to transmit a packet that triggers a multi-packet effect.

Thus the total time available for a child node to inject a packet is $2t_{rend}$, which means that the probability of this effect is $p = \frac{2t_w}{(1+P_i)t_{IPI}}$. Considering this probability, f_{extra} is given by:

$$f_{extra} = \sum_{k=1}^S k C_{F_i}^k p^k (1-p)^{F_i-k}$$

And the final contribution of this primitive to the duty cycle is:

$$\Delta_{as} = \frac{t_w}{(1+P_i)t_{IPI}} \frac{F_i}{1+f_{extra}} \quad (3.9)$$

Again, considering that the sink is always on, for sink neighbors the equation simply becomes

$$\Delta_{as} = \frac{t_{tx}}{t_{IPI}} F_i$$

(3) Anycast receptions

Same as CTP. Please refer to Equation (3.5), but here we represent it with Δ_{ar} .

Overall, the final equation describing the duty cycle of nodes running ORW is:

$$\Delta_{dc}^{ORW} = \Delta_{rc} + \Delta_{as} + \Delta_{ar} \quad (3.10)$$

3.3 Comparison of models

In this section we will make a preliminary comparison between the models derived for CTP and ORW in the previous sections. Table 3.2 lists the primitives used in the models. From the information in this table we can derive two clear results:

- (1) For broadcast related primitives, ORW outperforms CTP under all conditions (because ORW has no broadcast events during its steady state).
- (2) For the other primitives, the two protocols have an almost identical energy consumption, except for the complex unicast and anycast transmissions of non-sink neighbors nodes (top sub-row of “Uni/Anycast Tx”).

	CTP	ORW
CCA events	$\frac{t_c}{t_w}$	$\frac{t_c}{t_w}$
Beacon Tx	$\frac{t_w}{t_{IBI}}$	N/A
Beacon Rx	$\frac{t_{rx}}{t_{IBI}} N_i$	
Uni/Anycast Tx	$\frac{t_w}{2t_{IPi}} \frac{F_i}{1+f_{extra}}$	$\frac{t_w}{(1+P_i)t_{IPi}} \frac{F_i}{1+f'_{extra}}$
	$\frac{t_{tx}}{t_{IPi}} F_i$	$\frac{t_{tx}}{t_{IPi}} F'_i$
Uni/Anycast Rx	$\frac{t_{rx}}{t_{IPi}} L_i$	$\frac{t_{rx}}{t_{IPi}} L'_i$

Table 3.2: Comparison of each factor between CTP and ORW

Therefore we will breakdown the nodes into three groups, sink’s neighbors, relays and leafs to offer a deeper view about the Uni/Anycast transmission on different parts of the network. Table 3.3 lists the results after removing common factors in each row:

Table 3.3 reveals some insights about both protocols (considering only Uni/Anycast transmission):

- (1) If a node has only one parent, then ORW performs identical to CTP for leafs. With two or more potential parents, ORW is much better in terms of duty cycle. Hence, overall, the higher the density the better ORW should perform.
- (2) The sink’s neighbors’ Uni/Anycast transmissions are only related to their forwarding load. Thus a more balanced routing protocol can put

	CTP	ORW
Leaves	$1/2$	$\frac{1}{P_i+1}$
Sink neighbors	F_i^{ctp}	F_i^{orw}
Relays	$\frac{1}{2(1+f_{extra})}F_i^{ctp}$	$\frac{1}{(P_i+1)(1+f'_{extra})}F_i^{orw}$

Table 3.3: Comparison of Uni/Anycast transmission primitives among leaves, sink neighbors and relays

off the first occurrence of failure among sink neighbors. In general, ORW is more balanced than CTP, hence ORW should do a better job in maximizing the minimum lifetime of the sink's neighbors.

However, for relays the model indicates a complex interaction among load balance, communication primitives and particular phenomena, such as the multi-packet effect. Therefore, to obtain a clear comparison, we ran three one-hour experiments with a wakeup interval 1s. Then we extracted the parameters from the traces for the three types of nodes and averaged them. The results are listed in Table 3.4.

	CTP			ORW		
Classes	SN	RL	LF	SN	RL	LF
Forwarding Load	5.96	5.13	1.01	4.22	2.95	1.24
# of Parents	1	1.20	1.04	1.21	10.26	1.21
f_{extra}	N/A	0.1	0.01	N/A	0.01	0

Table 3.4: Parameter-based evaluation of duty cycle. SN: sinks neighbors, RL: relays, LF: leaves.

From Table 3.4 we can get the following information:

- (1) For ORW the effect of multiple transmissions in one rendezvous session can be neglected.
- (2) CTP's relays have higher average load than ORW. Now, considering that Equation 3.7 and Equation 3.9 also capture the cost of communication primitives (besides forwarding load), we hyphotesize that relay nodes in

CTP have a double burden: they perform many transmissions and each transmission is more expensive than in ORW.

Chapter 4

Empirical Results and Analysis

In this chapter we first show testbed results that validate the model constructed in Chapter 3. Next we present additional experiments related to network lifetime and analyze the results. We'll show how the network lifetime is affected by the energy budget, node density and network scale. Last we'll discuss some peculiar phenomena at low densities.

4.1 Experimental Setup

This section introduces the common settings for all the experiments. The settings that are particular to each type of experiment will be discussed in the corresponding sections.

4.1.1 General Conditions

We try to keep each protocol “as it is”, but there are still some things we modify. The main discrepancies between the default implementation and our modifications are list next, together with the reasons for these changes.

- *CCA Time*

This is a parameter that controls the duration of the Clear Channel Assessment (CCA), that is, the amount of time that the node remains awake every time it wakes up. According to [22], the CCA time should be $t_c = t_{backoff} + t_{ack}$, where $t_{backoff}$ is the CSMA back off time, and t_{ack} is the delay of the acknowledgement. The default value is set to 6 ms, which may lead to the following undesirable scenario, depicted in Figure 4.1:

Since the default CCA time is shorter than the interval between two packets, it's possible that the parent node wakes up just between two

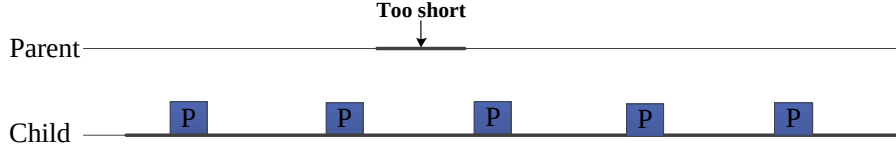


Figure 4.1: CCA time is so short that parent node misses the packets

consecutive transmissions of the child. If this occurs, the child node has to wait for at least another t_w to rendezvous with its parent. For the next time there is also no guarantee that they will detect each other. This is not a desired behavior. It is advised in the TinyOS source code to increase this value according to the specific platform. Thus, we increase the CCA time to about 12 ms, which is just long enough to avoid this potential problem.

- *Packet Interval*

This parameter controls how frequently a node generates data. The default values for CTP and ORW are 2 packet/min and 0.25 packet/min, respectively. In our experiments they are both set to 1 packet/min, which provides a data rate that is common to sensor networks applications, while still guaranteeing that enough data packets are transmitted to analyze the lifetime of the network (some public testbeds only allow slots of 30 minutes and hence the data rate can not be too low).

We always employ the data collected ten minutes after the startup of the network to ensure that the results are not influenced by the high variability of the starting phase.

4.1.2 Testbed Specification

We conduct our testbed experiments in Indriya, a publicly available testbed containing 100 active nodes (July, 2014). The sink is located at the corner of the testbed to ensure the largest possible diameter of the network in terms of number of hops.

4.2 Model Validation

This section validates our model with empirical results from the Indriya testbed. Results for CTP and ORW are presented separately in the following sections. To validate our models for both protocols, we used the following procedure: while running the protocols we measured the duty cycle of the nodes, which represent the ground truth, and we also measured the required

parameters for our models (forwarding load, listening load and number of potential parents). The goal is to observe if the duty cycle estimated by using these parameters is similar to the actual duty cycle measured at the nodes.

4.2.1 Results for CTP

We measure the duty cycle under different wakeup intervals, ranging from 250 ms to 16 s. The duty cycle is calculated as $\Delta = \frac{t_{on}}{t_{all}}$, where t_{on} is the total time when the radio is on, and t_{all} stands for the duration of the measurement. Every experiment lasts for one hour in total. For each wakeup interval we run the experiment three times. The average duty cycles and their corresponding standard deviations are plotted in Figure 4.2. The dashed lines represent the SoA model while the solid lines represent our new model. Table 4.1 quantifies the improvement of our model compared to the SoA model. The improvement of our model is calculated as:

$$improvement = \frac{|\Delta_{oldmodel} - \Delta_{real}| - |\Delta_{newmodel} - \Delta_{real}|}{\Delta_{real}}$$

where Δ_{real} represents the real measured duty cycle, $\Delta_{oldmodel}$ stands for the value calculated by the SoA model, and $\Delta_{newmodel}$ is the value calculated by our model.

$t_w(s)$		2^{-2}	2^{-1}	2^0	2^1	2^2	2^3	2^4
$\frac{\Delta_{model} - \Delta_{real}}{\Delta_{real}} (\%)$	Old Model	19.83	28.37	52.17	45.26	56.38	66.03	118.81
	New Model	19.05	25.36	43.21	29.81	15.83	-3.24	6.43
Improvement(%)		3.90	10.59	17.19	34.14	71.92	95.09	94.59

Table 4.1: Comparison between old and new models against real measurements in detail for CTP

From the results we can clearly see that our new model successfully:

- Reduces the overestimation in the old model by 95.09% for relay nodes. The smallest error against real measure values is only 3.24%.
- Maintains good performance for leave nodes and sink neighbors.

Thus we claim that our model is a good improvement over the SoA.

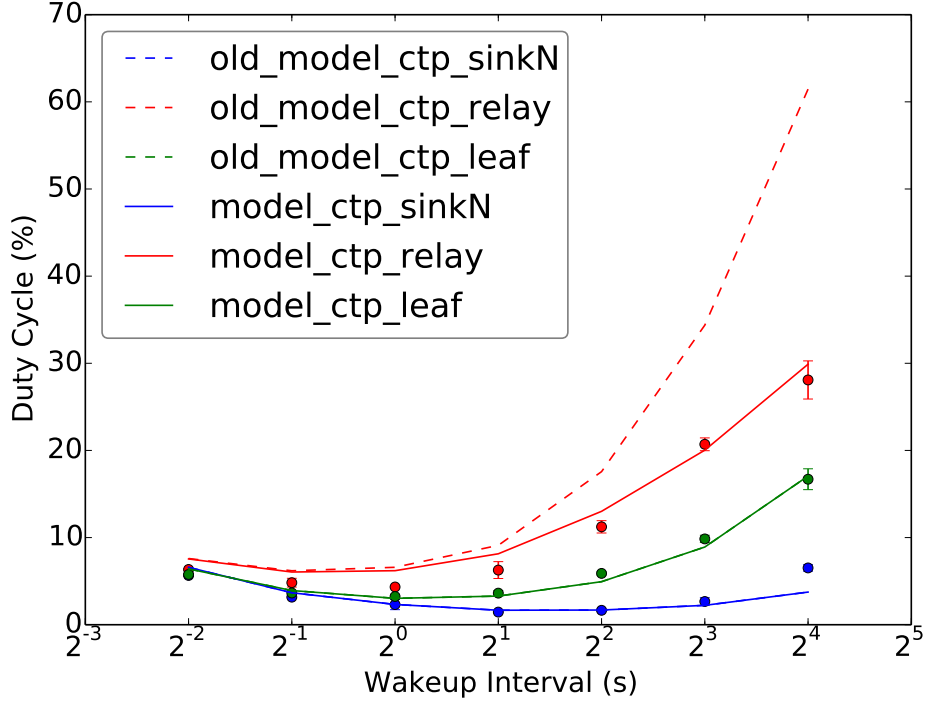


Figure 4.2: Result of new model after applying the fix

4.2.2 Results for ORW

To have a fair comparison with CTP, the experimental setups for ORW are identical to the ones for CTP. Please refer to Section 4.2.1 for more details. Given that there is no existing model for ORW, we quantify the advantages of identifying the problem described in Section 3.1.2 by comparing two models: the model that does not consider the effect in Section 3.1.2 is called “direct ported model”, and the one considering this effect is called the “new model”. Figure 4.3 and Table 4.2 provide our results.

Similarly to CTP, the model for ORW also reduces the overestimation of duty cycle for relays. This improvement confirms that the multiple-packets-transmission event is a phenomenon that needs to be considered.

However, unlike CTP results, we observe an underestimation of the duty cycle for leaves nodes. We hypothesize that this occurs due to our assumption of perfect links. This assumption is not a problem for CTP because CTP tends to choose links with high quality. But, ORW utilizes a pool of potential parents whose links’ quality change continuously in time. This means that at a given time, the actual number of parents that are qualified to forward the packet is less than the total number of parents observed up to that time. For P_i we use the total number of parents, and thus, we overestimate this parameter. Recalling Equation 3.9, we see that if P_i is

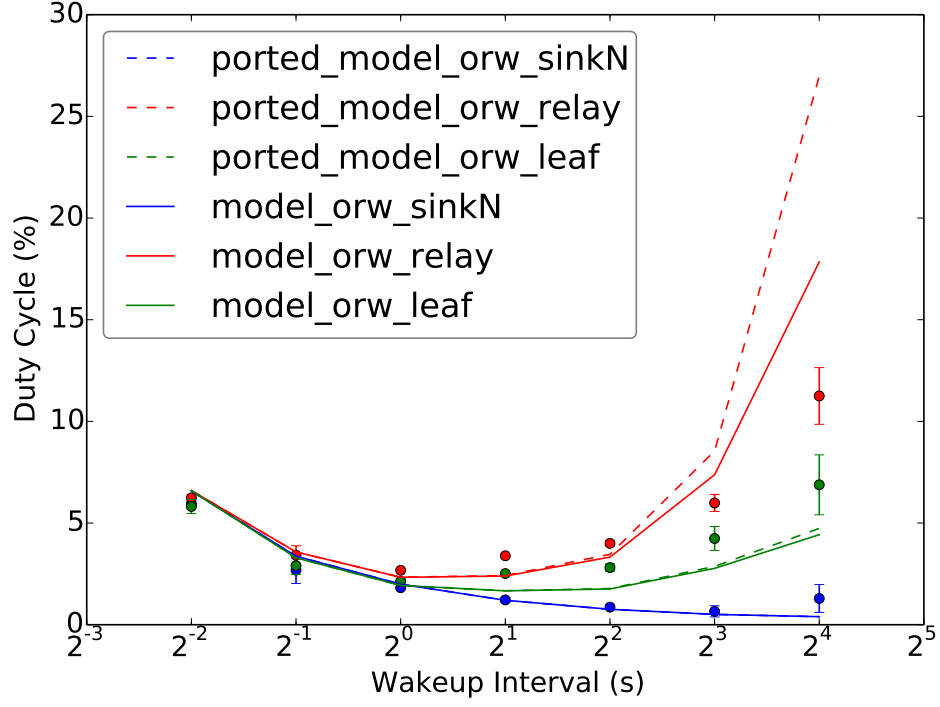


Figure 4.3: Comparison between the direct ported model and the new model for ORW

$t_w(s)$		2^{-2}	2^{-1}	2^0	2^1	2^2	2^3	2^4
$\frac{\Delta_{model} - \Delta_{real}}{\Delta_{real}} (\%)$	Old Model	5.74	4.73	-12.72	-28.61	-13.49	42.71	139.72
	New Model	5.74	4.70	-12.87	-29.37	-16.91	23.18	58.61
Improvement(%)		0.07	0.62	-1.13	-2.64	-25.29	45.72	58.05

Table 4.2: Comparison between the direct ported model and the new model against real measurements for ORW

overestimated, it will result in an underestimation on Δ_{as} . Therefore, when applying our model, a penalty on the potential number of parents should be taken into consideration, depending on the link communication quality.

4.3 Experiments on network lifetime

In this section we conduct experiments related to network lifetime, and explain the results using the models constructed in Chapter 3. We use the radio ‘on’ time as a proxy for energy consumption. Each node is given a fixed amount of energy budget in terms of radio ‘on’ time. Once a node exhausts its budget, it will shut down as if it “dies”. In this way we can capture the dying process of the network.

As mentioned in Section 2.1.1, there are many definitions of network lifetime related to specific applications. We cannot give an evaluation that covers every aspect. In our evaluation we focus on the following points:

- Illustrate the dying process using the total number of nodes alive
- Illustrate the dying process using the number of total connected nodes that have paths to the sink
- Illustrate the dying process for different classes of nodes (sink neighbors, leaves and relays)

4.3.1 Influence of Energy Budget

In this section we present the results for different values of initial energy to reveal the influence of the energy budget on network lifetime. Figure 4.4 to Figure 4.7 depict the dying process of the network when the energy limits are 128s and 256s. Figure 4.4 and Figure 4.6 reflect the dying process in terms of the total number of nodes alive. Every point shows the average result from three experiments, and the standard deviations.

To have a more detailed view of the dying process, Figure 4.5(a) and Figure 4.7(a) compare the total number of received packets by the network (top graph), the total throughput of the network (middle graph) and the dying process of each group of nodes (bottom graph). Figure 4.5(b) and Figure 4.7(b) show the relationship between the dying time and the number of hops to the sink. Nodes with higher loads have darker colors.

From the results we can see that, almost at any point, either in terms of the total number of nodes alive or from the perspective of different classes of nodes, ORW has longer lifetimes than CTP. This is not surprising, as in Chapter 3 we already showed that for every individual node, each transmission in ORW takes shorter rendezvous time than CTP. And from Figure 4.5(b) we can see a more evenly distributed load for ORW, which helps to avoid some nodes from dying too fast.

Overall, having a longer lifetime allows ORW to deliver more packets for the same initial energy budget. And as the energy budget increases, the advantage of ORW becomes stronger.

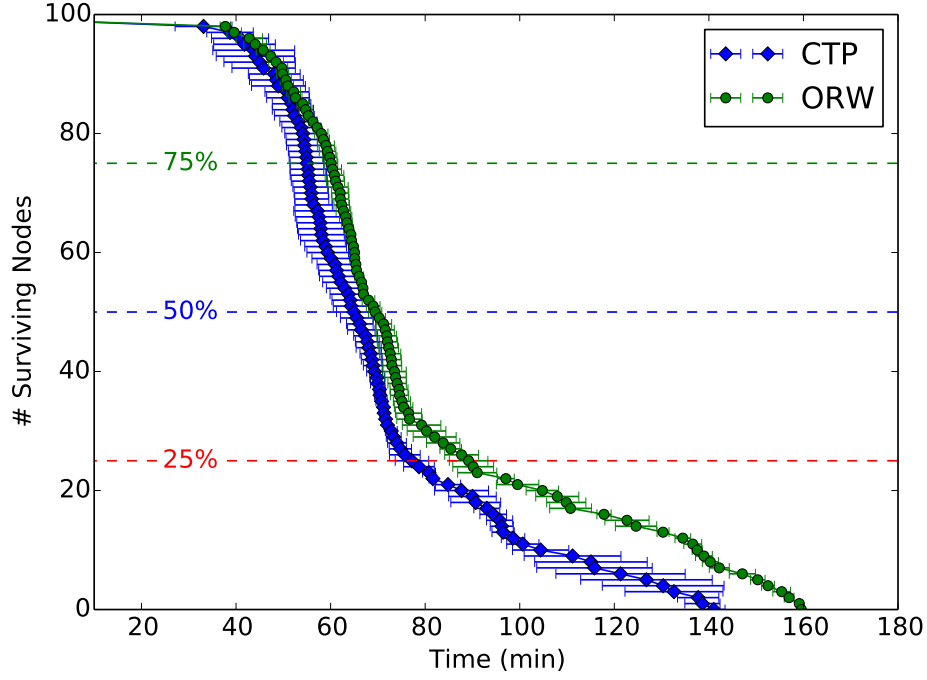


Figure 4.4: Dying process with energy limit 128s in terms of all nodes

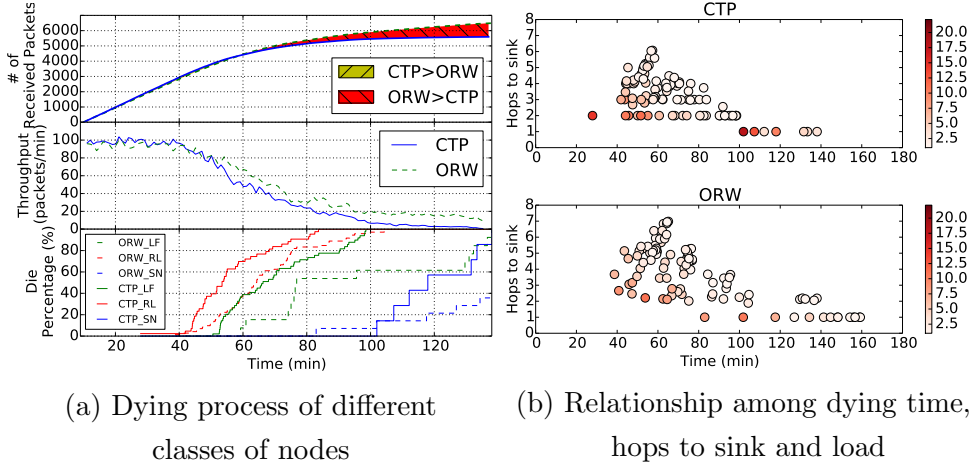


Figure 4.5: Dying process with energy limit 128s of one run

For the experimental setups in this section, we did not observe a big difference between the lifetimes of connected nodes (having a path to the sink) and all nodes (connected and disconnected), shown in Figure 4.8. This is probably due to the good connectivity of the network. With a high density, nodes can easily find new parents when some of its neighbors die. If

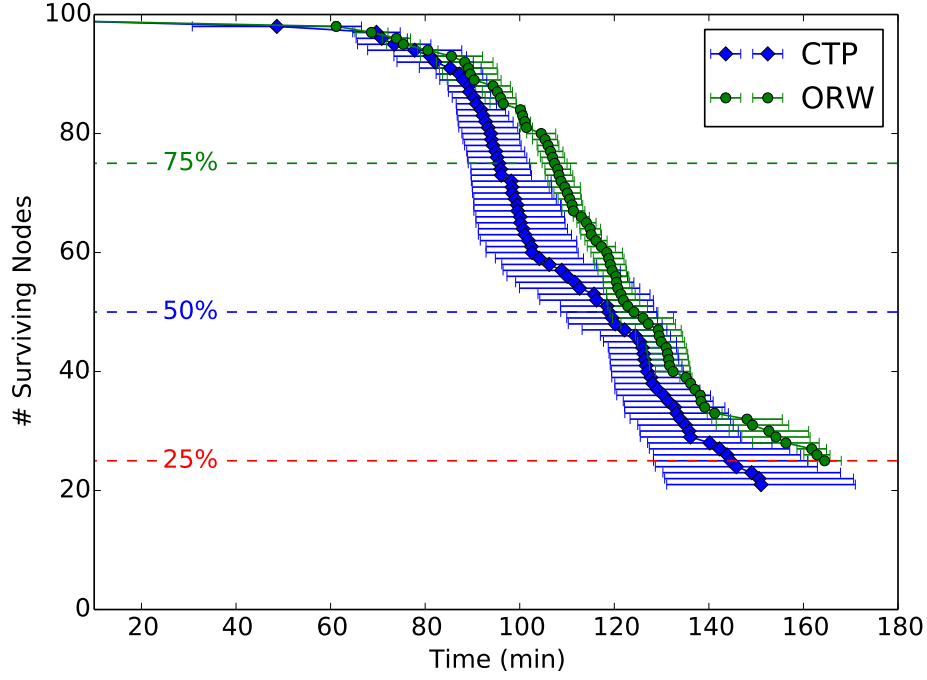


Figure 4.6: Dying process with energy limit 256s in terms of all nodes

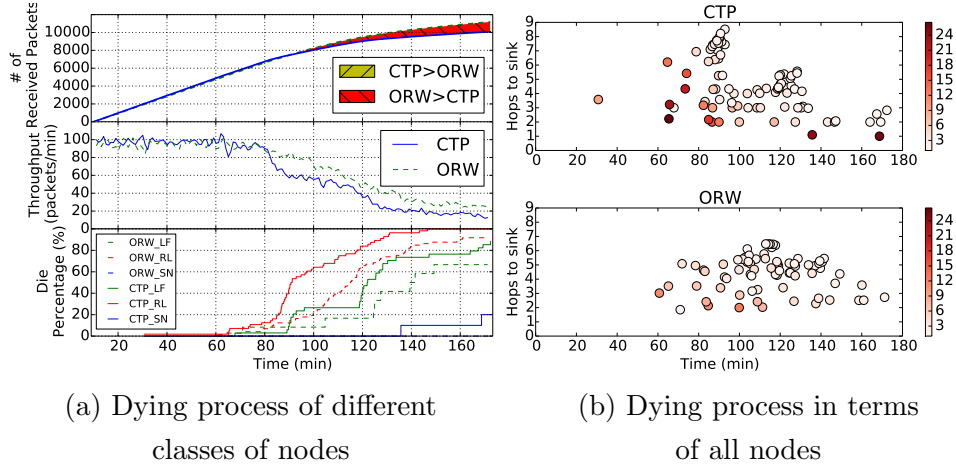


Figure 4.7: Dying process with energy limit 256s of one run

the density is low, there would be fewer nodes that can provide routing progress, thus the number of potential parents P_i would decrease for ORW. Equation 3.9 predicts that if P_i is close to 1, then the advantage of ORW's anycast will diminish. So it would be interesting to explore how the network would perform with lower densities, which will be addressed in the following

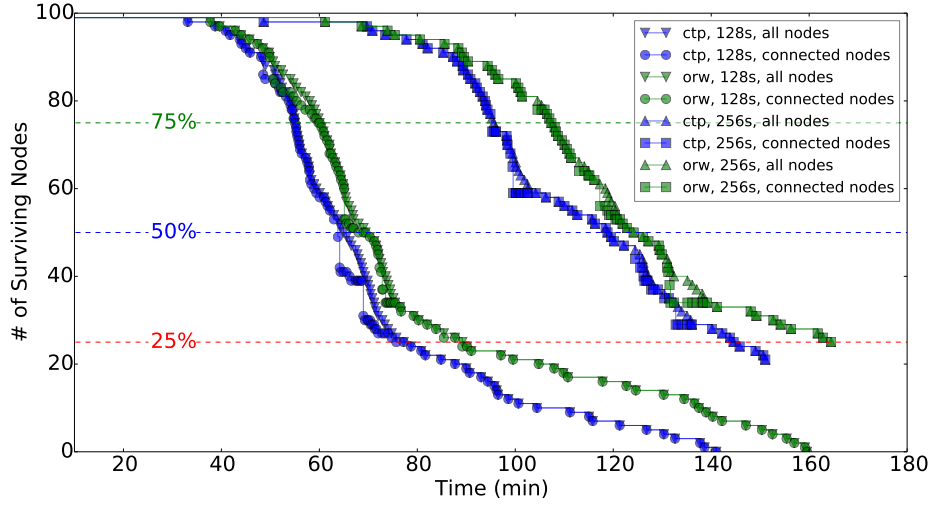


Figure 4.8: Lifetime of connected nodes vs all nodes

section.

4.3.2 Influence of Network Density

In these experiments, we select one third of the total nodes in the network in an even manner. Under this condition, 29 connected nodes are reported. All other parameters are kept the same as in the previous section. The energy limit is set to 128s.

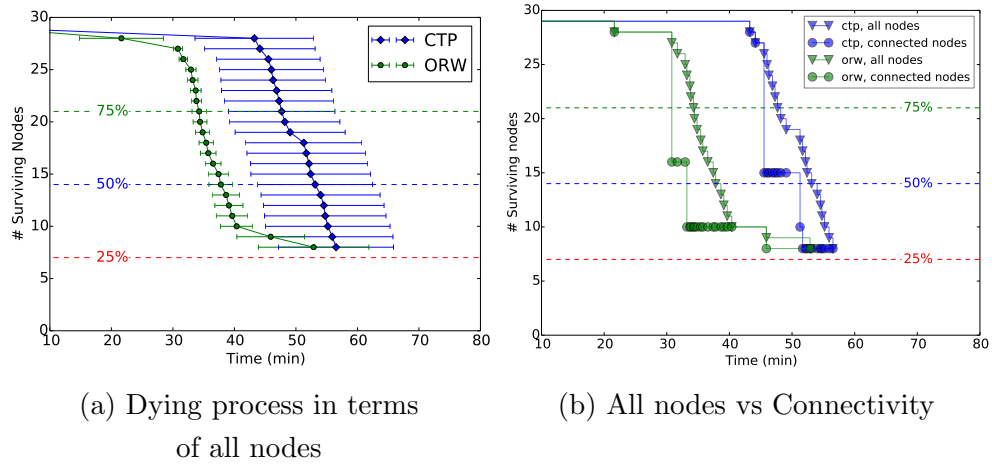


Figure 4.9: Dying process after reducing the density, with energy limit 128s

Figure 4.10 shows the result after reducing the node density. Different from Figure 4.8, there is a significant difference in the dying process. In sparse networks, extending the lifetime of nodes becomes more critical than

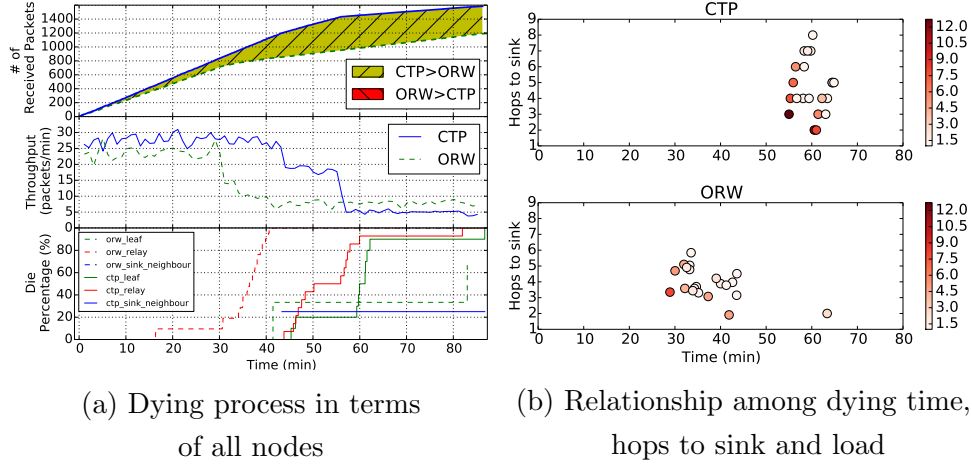


Figure 4.10: Dying process with energy limit 128s of one run, 29 nodes with low density

in dense networks, because some nodes may be the only choice towards the sink for their children. This “bottleneck” effect explains why there is a sharp drop in throughput (Figure 4.10(a)).

Under low densities, we can see clearly that ORW performs worse than CTP in at least two aspects: energy consumption and throughput. For energy consumption, although both protocols die earlier than the experiments in the previous section, ORW suffers more. This is counter intuitive, since even with P_i equals one, our model suggests that ORW should approximate CTP’s performance, rather than performing evidently worse. Another important aspect we should notice is that ORW never reaches the maximum possible throughput, while CTP does a good job. This indicates that ORW may suffer from some inherent issues that may be related to actual implementation rather than the design of the protocol. Before we go into a further discussion, first let us prove that it’s indeed density and not network scale what influences the performance of the two protocols.

4.3.3 Influence of Network Scale

In this section, we run experiments with the same number of nodes used in Section 4.3.2 (29 nodes). We carefully select the nodes so that the density is close to the environment in 4.3.1. Figure 4.11(a) shows that for the most part, the curves of ORW and CTP overlap with each other, suggesting that their performance is similar. Compared to Figure 4.10, the only difference is density. Thus we can confirm that it is a low-density rather than a small-scale what makes ORW perform worse. In order to have a deep understanding about why this happens and to capture some details that cannot be traced by the logging inside the code, we decided to perform

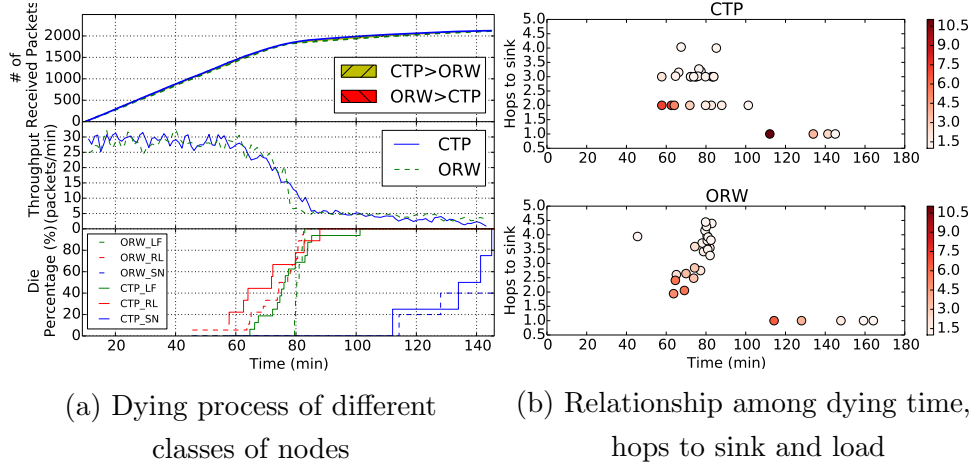


Figure 4.11: Dying process with energy limit 128s of one run, 29 nodes with high density

some simulations.

4.4 Discussion on Peculiar Phenomenon

We run our simulations in Cooja [23], a simulator that was developed for the Contiki operating system. We use the *Unit Disk Graph Model with exponential distance loss* as the radio model. This is an ideal transmission model that does not hold in practice, but it is a good model to identify problems in the implementation of protocols. Each node has a start delay chosen randomly within 10s to avoid synchronization. The network is a 6x6 grid, in which every node can talk to at most eight neighbors around it. We simulated ORW for 1 hour without energy limits. Through simulations we identify the following possible reasons why ORW performs worse in low-density networks:

- (1) *Parent becoming child*

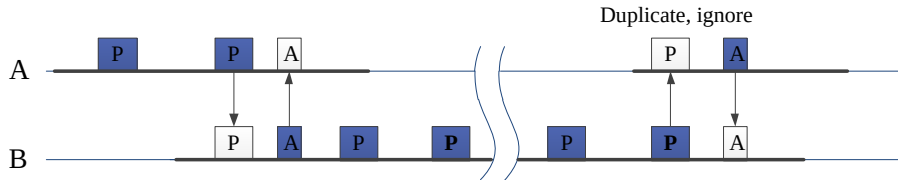


Figure 4.12: Parent becoming child

Figure 4.12 depicts one of the reasons why ORW performs poorly at low

densities. Node A successfully delivers the packet to B, but due to the low density, B cannot find a suitable parent to forward this packet. B has to keep on transmitting until it receives an acknowledgement from some node. If the transmission lasts too long, a penalty will be added to B's routing metric. When B's metric increases beyond a certain threshold, even the original child A will be able to satisfy the requirement, i.e. a loop occurs. At this time, if A receives the packet, it will acknowledge it, because its own routing metric is qualified to forward this packet. Given that node A has already seen this packet before, the packet will be regarded as a duplicate and will be ignored, leading to a packet lost.

- (2) *Receiving multiple packets and sending a single ack to the wrong node*

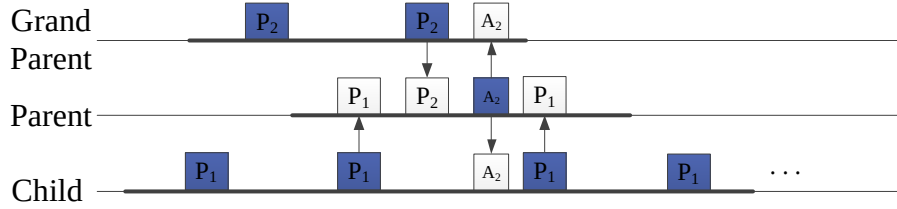


Figure 4.13: Wrong acknowledgement

Figure 4.13 shows another undesirable effect. After the parent node receives the child's packet, a grandparent's packet arrives before the parent issues the required acknowledgement to the child. Normally, the parent node should only acknowledge the packet from child node, but in this case, it wrongly sends out an acknowledgement for the packet from the grandparent node.

This event directly stops the grandparent's transmission, since it receives an acknowledgement and there is no need to continue transmitting the strobe of packets. But the parent node is not able to forward this packet, as it has a higher (worse) routing metric than its grandparent. This cause the packet from grandparent to be lost. In the meantime, the child node continues transmitting its packet, because no acknowledgement has been received for it. This causes an extra transmission time, which increases the duty cycle of the child node.

- (3) *Although an acknowledgement is received, the original sender keeps on transmitting*

As shown in Figure 4.14, although the child node receives the acknowledgement, instead of stopping the transmission of packets, it still keeps

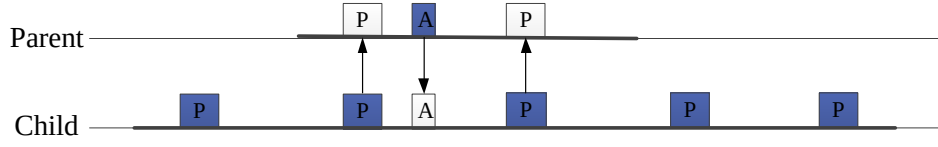


Figure 4.14: Original sender still keep sending after receiving an acknowledgement

sending. This leads to an unnecessary long transmission time, and consequently, to a higher energy consumption.

- (4) *An acknowledgement is sent when receiving the packet for the second time instead of the first*

Instead of acknowledging the packet immediately upon the first reception, the parent node sends out the acknowledgement after the second time it receive the packet (Figure 4.15). Sometimes this is not a problem (except for the extra unnecessary transmission), but in some instances we observe packet suppressions by the parent, which leads to packet losses.

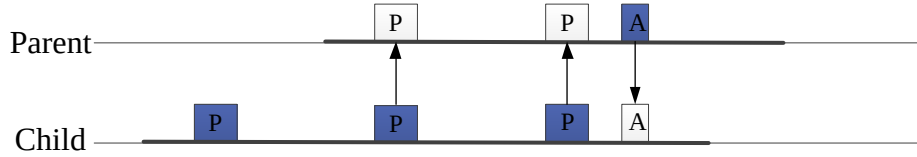


Figure 4.15: Acknowledgement is sent when receiving the packet for the second time

Overall, scenarios 1, 2, and 4, lead to packet losses. Even though all these scenarios can occur in high density networks, sparse networks have a higher probability of triggering such events. Scenario 3 does not necessarily lead to a packet loss, but it increases the nodes' duty cycle since the nodes needs to transmit for an extra time that is not needed. Among the aforementioned scenarios, scenario 2 can also be found in CTP, thus we hypothesize that it may be related to the MAC layer's implementation or hardware issues. The rest of scenarios only occurs in ORW, thus we owe them to the flaws in the implementation of ORW.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

The goal of this thesis is to analyze the lifetime of data collection networks. Our analysis focuses on two well known protocols that have different forwarding mechanisms: CTP and ORW. We constructed new models and validate them in public testbeds. Our models show improve the accuracy of existing CTP models up to 95% for relay nodes. We also construct a new model for ORW and make a preliminary comparison between the models of both protocols. By doing experiments in real testbeds, we find that in terms of network lifetime, ORW outperforms CTP in high density network, which validate the results from our models. On the other hand, CTP is the choice for low density networks. Although our model predicts that ORW should be better, or at least similar to CTP's performance in low density networks, through simulations we show that these deviations from the model are mainly due to some flaws in ORW's code.

5.2 Future Work

Limited by time, we did not have a chance to perform experiments on more testbeds besides Indriya. This could reveal whether our results hold across other environments. Also CTP and ORW are not bounded to specific a MAC scheme. Investigating how they would behave with other MACs is an interesting topic for future research. Last but not least, a careful check should be made on ORW's code to remove its limitations.

Bibliography

- [1] Ian F Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci. Wireless sensor networks: a survey. *Computer networks*, 38(4):393–422, 2002.
- [2] Aline Baggio. Wireless sensor networks in precision agriculture. In *ACM Workshop on Real-World Wireless Sensor Networks (REALWSN 2005)*, Stockholm, Sweden, 2005.
- [3] Manish Bhardwaj and Anantha P Chandrakasan. Bounding the lifetime of sensor networks via optimal role assignments. In *IEEE INFOCOM*, volume 3, pages 1587–1596. INSTITUTE OF ELECTRICAL ENGINEERS INC (IEEE), 2002.
- [4] Mihaela Cardei, My T Thai, Yingshu Li, and Weili Wu. Energy-efficient target coverage in wireless sensor networks. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, volume 3, pages 1976–1984. IEEE, 2005.
- [5] Chih-Yung Chang and Hsu-Ruey Chang. Energy-aware node placement, topology control and mac scheduling for wireless sensor networks. *Computer networks*, 52(11):2189–2204, 2008.
- [6] Jae-Hwan Chang and Leandros Tassiulas. Energy conserving routing in wireless ad-hoc networks. In *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 1, pages 22–31. IEEE, 2000.
- [7] Douglas SJ 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, 2005.
- [8] Isabel Dietrich and Falko Dressler. On the lifetime of wireless sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 5(1):5, 2009.
- [9] Manjunath Doddavenkatappa, Mun Choon Chan, and Akkihebbal L Ananda. Indriya: A low-cost, 3d wireless sensor network testbed. In *Testbeds and Research Infrastructure. Development of Networks and Communities*, pages 302–316. Springer, 2012.
- [10] Euhanna Ghadimi, Olaf Landsiedel, Pablo Soldati, and Mikael Johansson. A metric for opportunistic routing in duty cycled wireless sensor networks. In *Sensor, Mesh and Ad Hoc Communications and Networks (SECON), 2012 9th*

- Annual IEEE Communications Society Conference on*, pages 335–343. IEEE, 2012.
- [11] Omprakash Gnawali, Rodrigo Fonseca, Kyle Jamieson, David Moss, and Philip Levis. Collection tree protocol. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, pages 1–14. ACM, 2009.
 - [12] Gaurav Gupta and Mohamed Younis. Load-balanced clustering of wireless sensor networks. In *Communications, 2003. ICC'03. IEEE International Conference on*, volume 3, pages 1848–1852. IEEE, 2003.
 - [13] Tian He, Sudha Krishnamurthy, John A. Stankovic, Tarek Abdelzaher, Liqian Luo, Radu Stoleru, Ting Yan, Lin Gu, Jonathan Hui, and Bruce Krogh. Energy-efficient surveillance system using wireless sensor networks. In *Proceedings of the 2Nd International Conference on Mobile Systems, Applications, and Services, MobiSys '04*, pages 270–283, New York, NY, USA, 2004. ACM.
 - [14] Keith Hellman and Michael Colagrosso. Investigating a wireless sensor network optimal lifetime solution for linear topologies. *Journal of Interconnection Networks*, 7(01):91–99, 2006.
 - [15] Santosh Kumar, Anish Arora, and Ten-Hwang Lai. On the lifetime analysis of always-on wireless sensor network applications. In *Mobile Adhoc and Sensor Systems Conference, 2005. IEEE International Conference on*, pages 3–pp. IEEE, 2005.
 - [16] Olaf Landsiedel, Euhanna Ghadimi, Simon Duquennoy, and Mikael Johansson. Low power, low delay: opportunistic routing meets duty cycling. In *Proceedings of the 11th international conference on Information Processing in Sensor Networks*, pages 185–196. ACM, 2012.
 - [17] Jian Li and Prasant Mohapatra. An analytical model for the energy hole problem in many-to-one sensor networks. In *IEEE Vehicular Technology Conference*, volume 62, page 2721. IEEE; 1999, 2005.
 - [18] Yunhao Liu, Yuan He, Mo Li, Jiliang Wang, Kebin Liu, and Xiangyang Li. Does wireless sensor network scale? a measurement study on greenorbs. *Parallel and Distributed Systems, IEEE Transactions on*, 24(10):1983–1993, 2013.
 - [19] Jun Luo and J-P Hubaux. Joint mobility and routing for lifetime elongation in wireless sensor networks. In *INFOCOM 2005. 24th annual joint conference of the IEEE computer and communications societies. Proceedings IEEE*, volume 3, pages 1735–1746. IEEE, 2005.
 - [20] Alan Mainwaring, David Culler, Joseph Polastre, Robert Szewczyk, and John Anderson. Wireless sensor networks for habitat monitoring. In *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications, WSNA '02*, pages 88–97, New York, NY, USA, 2002. ACM.
 - [21] Mirela Marta and Mihaela Cardei. Improved sensor network lifetime with multiple mobile sinks. *Pervasive and Mobile computing*, 5(5):542–555, 2009.

- [22] David Moss and Philip Levis. Box-macs: Exploiting physical and link layer boundaries in low-power networking. *Computer Systems Laboratory Stanford University*, 2008.
- [23] Fredrik Osterlind, Adam Dunkels, Joakim Eriksson, Niclas Finne, and Thiemo Voigt. Cross-level sensor network simulation with cooja. In *Local Computer Networks, Proceedings 2006 31st IEEE Conference on*, pages 641–648. IEEE, 2006.
- [24] Chris Otto, Aleksandar Milenkovic, Corey Sanders, and Emil Jovanov. System architecture of a wireless body area sensor network for ubiquitous health monitoring. *Journal of Mobile Multimedia*, 1(4):307–326, 2006.
- [25] Daniele Puccinelli, Silvia Giordano, Marco Zuniga, and Pedro José Marrón. Broadcast-free collection protocol. In *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*, pages 29–42. ACM, 2012.
- [26] Ivan Stojmenovic and Xu Lin. Power-aware localized routing in wireless networks. *Parallel and Distributed Systems, IEEE Transactions on*, 12(11):1122–1133, 2001.
- [27] Di Tian and Nicolas D Georganas. A coverage-preserving node scheduling scheme for large wireless sensor networks. In *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, pages 32–41. ACM, 2002.
- [28] Wei Wang, Vikram Srinivasan, and Kee-Chaing Chua. Using mobile relays to prolong the lifetime of wireless sensor networks. In *Proceedings of the 11th annual international conference on Mobile computing and networking*, pages 270–283. ACM, 2005.
- [29] Wei Ye, John Heidemann, and Deborah Estrin. An energy-efficient mac protocol for wireless sensor networks. In *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 3, pages 1567–1576. IEEE, 2002.