

Delft University of Technology  
Master's Thesis in Embedded Systems MSC

# Opportunistic Routing for Indoor Energy Harvesting Wireless Sensor Networks

Eloi Garrido Barrabés





# Opportunistic Routing for Indoor Energy Harvesting Wireless Sensor Networks

Master's Thesis in Embedded Systems

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

Eloi Garrido Barrabés  
E.GarridoBarrabes@student.tudelft.nl

16th August 2016

**Author**

Eloi Garrido Barrabés (E.GarridoBarrabes@student.tudelft.nl)

**Title**

Opportunistic Routing for Indoor Energy Harvesting Wireless Sensor Networks

**MSc presentation**

August 23, 2016

**Graduation Committee**

Prof.dr. K. G. Langendoen Delft University of Technology

Dr. Venkatesha Prasad Delft University of Technology

Dr. Zaid Al-Ars Delft University of Technology

Ir. V. Rao Delft University of Technology

## Abstract

Internet of Things (IoTs) is envisioned to enable smart spaces such as smart homes, interactive museums or personalized trade in shopping malls. In these smart homes, several sensors and actuators assist in automating tasks of daily life by forming wireless sensor networks (WSNs). Active and Assisted Living (AAL) is one of the important applications of smart homes, wherein activities of elderly persons are mainly monitored and assist them through actuators when required. With a huge number of sensors involved in AAL applications across smart homes, powering the nodes is an important issue. To this end, make use of ambient energy harvesting technologies for enabling perpetual operations of the WSNs.

Due to limited energy harvesting opportunities in indoor environments, the energy levels of nodes in the WSN varies over space and time. Therefore, collecting data reliably over such a network is a significant challenge. There are many proposals in this domain, including MAC, routing, etc., in such WSN. Although several routing protocols exist for WSNs and EH-WSNs, they do not consider the limited energy availability. Consequently, they do not adapt to the conditions, therefore fail in their objective. We propose a novel routing protocol called Harvesting Energy Aware Routing with adaptive Duty Cycling (HEAR-DC), based on the philosophy of using available energy, and routing the data packets opportunistically. HEAR-DC has several components: (i) transmitter-initiated MAC with opportunistic reception and duplicity avoidance; (ii) an energy harvesting aware gradient; (iii) adaptive duty cycling; and (iv) adaptations to support mobile nodes and sparse data traffic.

We also develop a trace-based energy harvesting simulation module for Cooja, which does not exist as yet. With this module, real data traces can be used for evaluating energy-harvesting WSN protocols. We present the design and implementation aspects as well as limitations of this module in the current state.

We evaluate our protocol against the state-of-the-art opportunistic routing protocol ORW. Results show that our proposed protocol achieves a latency reduction of 64% with 92.75% lower packet loss as compared to ORW in the best case of high energy availability. In the worst case, HEAR-DC achieves a substantial latency reduction of 37% and packet loss reduction of 48% compared to ORW. HEAR-DC achieves almost a factor of four lower delay than ORW when only ‘events’ are transmitted.



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Challenges of EH-WSN . . . . .	4
1.2	Problem Statement . . . . .	4
1.2.1	Challenges . . . . .	5
1.3	Contributions . . . . .	6
<b>2</b>	<b>Related Work</b>	<b>9</b>
2.1	ORW . . . . .	10
2.2	ORiNoCo . . . . .	11
<b>3</b>	<b>System Model</b>	<b>13</b>
3.1	Network Architecture . . . . .	13
3.2	Energy Model . . . . .	14
3.2.1	Energy Sources and Harvesters . . . . .	14
3.2.2	Storage Element . . . . .	16
3.2.3	Energy Consumption Model . . . . .	16
3.3	Assumptions . . . . .	17
<b>4</b>	<b>Harvesting Energy Aware Routing Protocol with Adaptive Duty Cycling (HEAR-DC)</b>	<b>19</b>
4.1	Overview of HEAR-DC . . . . .	20
4.2	Transmitter-Initiated MAC . . . . .	21
4.3	Gradient . . . . .	23
4.4	Adaptive Duty Cycle . . . . .	26
4.5	Operation of HEAR . . . . .	27
4.5.1	Bootstrapping . . . . .	27
4.6	Steady State Operation . . . . .	29
4.6.1	Adaptations . . . . .	30
<b>5</b>	<b>Trace-based Energy Harvesting Simulations</b>	<b>33</b>
5.1	Advantages of Trace-Based Simulations for EH-WSNs . . . . .	33
5.2	Data Traces of Energy Arrival . . . . .	34
5.3	Energy Harvesting Module . . . . .	35
5.3.1	Design and Implementation . . . . .	35

5.3.2	Features . . . . .	37
5.4	Limitations and Open Questions . . . . .	38
<b>6</b>	<b>Performance Evaluation</b>	<b>41</b>
6.1	Methodology . . . . .	41
6.2	Scenarios . . . . .	42
6.3	Results . . . . .	43
6.3.1	High Energy Scenario . . . . .	43
6.3.2	Low Energy Scenario . . . . .	45
6.3.3	Protocol Evaluation under Variable Traffic . . . . .	47
6.3.4	Impact of Mobile Nodes . . . . .	47
<b>7</b>	<b>Conclusions and Future Work</b>	<b>51</b>
7.1	Conclusions . . . . .	51
7.2	Future Work . . . . .	53

# Chapter 1

## Introduction

As the world moves towards ubiquitous connectivity, not just between people but also including objects or “things”, a huge surge is expected in the usage of Internet of Things (IoT) devices and end-user applications. Applications range from simple monitoring of temperature to automatic climate and soil control for agricultural applications and connected fridges that place orders for milk when milk is over. The information exchange between devices is not only powering these innovative applications but also making traditional elements *smart*. For instance, today a mirror tells the user about the weather [6] when he stands in front of it, shoes calculate how much the user has walked, and a coffee machine keeps coffee ready in the morning because the alarm clock informed about the user waking up from bed. To enable such applications, the systems usually require “eyes” to obtain information and learn, and “mouths” to talk to each other. Typically, sensors form the *eyes* to measure a physical parameter, and wireless radios work as the *mouth* to communicate data providing our applications with the required tools to add the adjective “smart” to more and more fields every day.

One of the important applications enabled by IoTs is Active and Assisted Living (AAL). AAL encompasses technical systems to support people dwelling in a smart house. In the past, AAL was conceived to support elderly people and individuals with special needs in their daily routine. The primary goal of AAL was to maintain and foster the autonomy of those people. Thus, to increase safety in their lifestyle and in their home environment. However, recently, it has been extended for the well-being of any person, including for healthy persons [2]. The idea is to allow people to age healthily, so that a certain quality of life is met, specifically for the majority of older adults. AAL enables easier access to information to informal caregivers such as family and friends as well as the doctors about the activities of the person in an AAL enabled house. Specifically, with smart sensors deployed in the house, the person’s activities of daily living (ADL) can be monitored easily. Any abnormality can be reported immediately to the caregivers.

To monitor ADLs and the other environmental parameters, many sensors are deployed around the house. For instance, motion sensors to detect where a person is, temperature and humidity sensors to determine the ambiance, wearables and body area sensors to determine respectively the activities and physiological parameters of the person. Fig. 1.1 shows an example deployment of the system in an assisted-living house with many residents. While some sensors may be fixed, wearables and other body sensors are worn by the person making the sensors mobile. These sensors are typically powered by batteries.

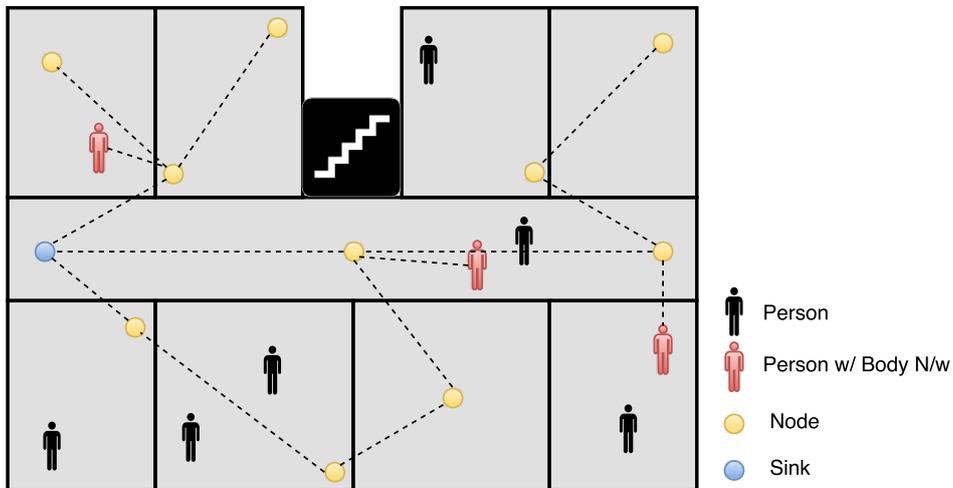


Figure 1.1: Assisted-living house deployment. Composed by emplaced nodes and body sensors.

All the data from the sensors need to be collected at a central place for further processing. Alternatively, to conserve battery, only ‘events’ of certain significance may be reported [32, 31]. We call such systems as event-triggered systems. Events in an AAL scenario can be quite sparse depending on the number of people in the house. For example, if the user is stationary, there is no change in the environment, hence there is no sensor data to report. However, when the user begins to move, the sensor may report this change in value. This reduces the traffic as well as increases energy-efficiency of the node.

Most of the wireless sensor nodes are small, inexpensive, low power consuming devices that are powered by batteries. Typically, most AAL deployments need a multi-hop communication before the data reaches a ‘sink’, forming a wireless sensor network (WSN). The maximum number of hops depends on the deployment. More sensors are expected to be deployed in the near future in order to gain fine-grained observations of ADLs in smart homes. While WSNs have been researched for more than a decade now, energy efficiency and scalability with respect to gathering data for a given

deployment still attract the attention of researchers.

Typically, the sensors deployed are required to last long. However, since the nodes are battery-powered, their lifetime is limited. Much research has been done to enhance energy-efficiency of nodes through algorithms and protocols to extend their lifetimes. However, it is impractical to have battery operated nodes since replenishing them is a laborious task. If nodes are deployed in inaccessible locations, the network has a limited lifetime. Consequently, harvesting energy from ambient sources to power these nodes has attracted attention in recent times. Energy harvesting is a technique that harvests or scavenges a variety of untapped ambient energy sources and converts the harvested energy into electrical energy to recharge storage elements such as batteries or super-capacitors [21]. Energy harvesting can power the nodes perpetually in theory [29] and also it provides secondary benefit of providing the context [31].

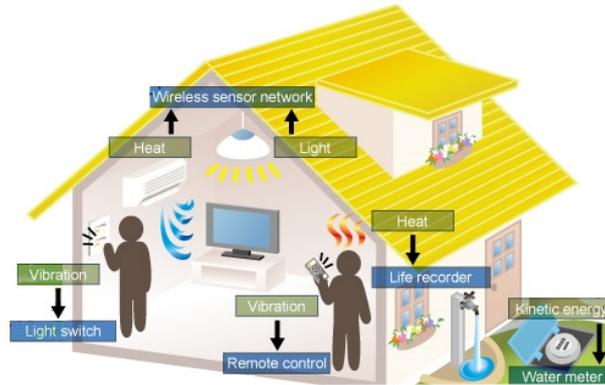


Figure 1.2: Indoor energy harvesting opportunities.

Energy-harvesting technology enables the network to extract energy from surrounding environment, such as solar power [12], mechanical movement [26], heat [13] and fluid flow [3]. Fig. 1.2 shows few opportunities of scavenging energy in a home environment. Energy-harvesting technology provides numerous benefits [21]:

1. Reduce the dependency on battery power: with the harvested energy, nodes eliminate the use of battery power—the harvested ambient energy may be sufficient to eliminate the need for batteries completely.
2. Reduce installation and maintenance cost: self-powered nodes eliminate service visits to replace batteries. This independence of each node

has the potential to boost its market potential into the consumer world since it needs the least intervention from end-users.

3. Provide long-term solutions: reliable nodes with energy-harvesting devices will function as long as the ambient energy is available, which is perfectly suited for long-term applications.

With untapped power sources, energy-harvesting wireless sensor networks (EH-WSNs) can operate perpetually. With EH-WSNs, our goal of collecting sensor data be it event-triggered or periodic data at a sink, reliably and perpetually should be realizable. However, it is not straightforward to do so as there are certain challenges.

## 1.1 Challenges of EH-WSN

Unfortunately, merely replacing the batteries with energy harvesters does not guarantee perpetual network operations nor the desired reliability. This is due to:

1. Ambient energy sources do not provide constant power. While the harvested energy can at times be very low, it can be in excess of the storage capacity of the nodes on other occasions. For instance, statistics show that the difference in available solar power at shadowy, cloudy and sunny environments can be up to three orders of magnitude [30].
2. The harvested energy from ambient sources varies drastically by location and time. For example, consider one node placed next to a window with direct sunlight and the other one positioned in a bookshelf. The amount of energy harvested by these nodes is different during day and night. In either case, the node next to the window has more opportunities to harvest energy than the node in the bookshelf since the latter node probably cannot receive direct light on its solar panel.

These aspects are illustrated with the following example. Fig. 1.3(a) shows an EH-WSN in which all nodes have sufficient energy at time  $t_1$ . However, due to insufficient harvested energy, node 3 dies at time  $t_2$ . The network is still connected through nodes 1 and 4. However, when node 1 dies at  $t_3$ , the network gets disconnected. Therefore, a system that tackles this type of situation and redirects node traffic dynamically is required.

## 1.2 Problem Statement

Evidently, the major problem is to collect data reliably over the dynamic network, which is the focus of this thesis. We target to design a routing protocol, along with a routing metric, for EH-WSNs that enables reliable

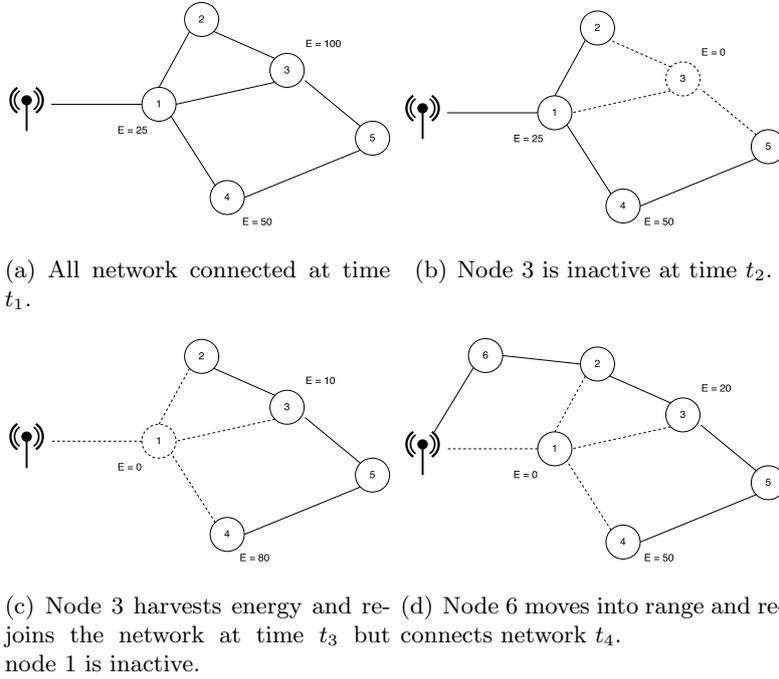


Figure 1.3: Dynamic energy on EH-WSN changes network connections.

data collection taking into account the energy-harvesting variations. By reliability, we mean the ratio of the number data packets received at the sink to the number of packets generated in the nodes. Further, we assume the nodes are powered only by super-capacitors, i.e., by harvested energy. Due to the above-mentioned energy variations, we see there are several challenges to this problem.

### 1.2.1 Challenges

All the challenges are due to the variations in harvested energy. We enlist the challenges in realizing a solution to our problem here.

1. Since the nodes operate mainly in indoor conditions, the amount of energy harvested is very small, typically in the order of  $\mu\text{J}$ . Therefore, nodes dying and re-joining the network are inevitable and must be dealt with.
2. Identifying a reliable route is challenging. It is required to determine how to quantify reliability.
3. Static routes do not work in the dynamic network we are considering to operate in. The protocol overheads can be taxing on the available energy on the nodes. Thus, the overheads must be kept to a minimum

in order to make efficient use of the available energy on the nodes. Frequent updating of the nodes leaving and joining cannot be done, and a suitable low overhead workaround must be sought.

4. Mobile nodes add another dimension of dynamics to an already dynamic network. The protocol must enable these nodes to send data on this network reliably.

There are several energy-efficient routing protocols for WSNs in the literature (e.g., CTP [15]). Most of these protocols have overheads for link-quality assessments and route maintenance. Opportunistic Routing (OR) based protocols avoid these overheads to a great extent [14, 37]. The two important OR protocols in this context are ORW [14] and ORiNoCo [37]. ORW is one of the most well-known OR-based protocol for battery-powered WSNs, but does not adapt to energy variations in EH-WSNs. ORiNoCo has been adapted for EH-WSNs; however, it is passive in adapting to energy variations. This is because ORiNoCo uses receiver-initiated medium access control (MAC) along with adaptive duty cycling<sup>1</sup> to route packets than an active routing metric. This results in packets getting stuck on high energy nodes surrounded by low energy nodes with little progress. Therefore, we propose a novel OR-based routing protocol, Harvesting Energy Aware Routing with adaptive Duty Cycling (HEAR-DC), based on the philosophy of opportunistic use of available energy and wireless channel for routing. In this protocol, we make the high energy nodes take more load than the low energy nodes to achieve reliability. Furthermore, we propose a novel gradient, or routing metric, that allows to actively make the higher energy nodes take more load. However, number of hops can vary also because of limited ranging based on the availability of energy. Sometimes it may be important to send a packet twice on a short a link rather than sending once on a longer link based on availability of energy.

### 1.3 Contributions

This work contributes to the state of the art in the following way:

1. We propose a novel OR-based routing protocol, Harvesting Energy Aware Routing with adaptive Duty Cycling (HEAR-DC), that provides a reliable routing solution for EH-WSNs.
2. We propose a novel gradient to quantify the reliability together with routing progress offered by a node. We identify reliability with the energy level of the node: higher the energy in the node, higher is the

---

<sup>1</sup>Adaptive duty cycling is a technique in EH-WSNs in which the ON period of the node is proportional to the energy present in the storage.

reliability offered by the node. The intuition is, because it has more energy, it can forward the packet sooner in the direction of the sink.

3. We develop an energy-harvesting simulation module for Cooja, which does not exist as yet. This module uses real energy data traces on the nodes to simulate energy harvesting rather than the conventional method of using stochastic distributions. Simulations with this module is a step closer to reality.
4. We evaluate the system through simulations on Contiki OS with real-world energy harvesting datasets. We show that HEAR-DC achieves a substantial latency reduction of 37% and packet loss reduction of 48% compared to ORW when the harvesting energy is low or none for some nodes (worst case).

The major takeaways from this thesis are the following:

1. A novel gradient and routing protocol with low overheads that is designed to provide reliable data collection even in low energy harvesting conditions. This takes EH-WSNs a step forward since HEAR-DC does not rely on energy predictions or location information, and can work with any kind of harvesting technology.
2. The energy-harvesting module for Cooja that enables trace-based simulations, since the simulations are a step closer to reality.



## Chapter 2

# Related Work

Many routing protocols have been proposed for WSNs [28]. There is a different protocol for almost every scenario. Most of these works consider battery-powered networks, wherein one of the inevitable requirements is energy efficiency, i.e., maximize the lifetime of the network under a certain scenario. Most of the protocols carry overheads with them, such as control packets to maintain routing structures/parents in tree based protocols, and link-quality assessment (e.g., Collection Tree Protocol [15]) in order to select links of consistently high reliability and achieve higher energy efficiencies.

We studied several current EH-aware protocols from the literature [28]. Some of the protocols introduce improvements. Protocols like *Randomized Max-Flow* (R-MF) [5], *Energy-opportunistic Weighted Minimum Energy* (E-WME) [25] and *Randomized Minimum Path Recovery Time* (R-MPRT) [23] used the stored energy or a node's harvesting ratio as metric parameter to find a suitable route towards the sink. R-MF focuses to improve throughput but needs information about the full network; R-MPRT only uses harvesting rate as metric, which leads to oscillations. E-WME has lots of losses in low energy conditions and overheads produced by learning about neighborhood state [19].

The Opportunistic Routing (OR) paradigm, on the other hand, reduces overheads and does not maintain routing tables, and provides highly energy-efficient, low-delay data collection possibilities [14]. OR works on the principle of *anycast*, i.e., the packet is broadcasted over the medium but is picked up by a suitable forwarder while the other nodes drop the packet. In OR, the forwarding decision of a packet is delayed until after the transmission, thereby spatial diversity of the wireless channel is exploited. OR in WSNs target energy efficiency and low overhead mechanisms for forwarder selection in duty cycled radios.

In the context of our problem, OR paradigm suits the best since (a) we need low overhead mechanisms to overcome the problem of nodes leaving and rejoining the network; and (b) it is best to delay the forwarder selection

until after transmission as the node with high reliability quotient can forward the packet. We base our solution on OR method as shall be described in Chap. 4. We shall, therefore, summarize two important state-of-the-art OR based routing protocols - Opportunistic Routing in WSNs (ORW) and Opportunistic Receiver-initiated No-overhead Collection (ORiNoCo) protocols in this chapter.

## 2.1 ORW

Opportunistic Routing in Wireless sensor networks (ORW) [14] is an opportunistic routing protocol that targets applications with low and asynchronous duty cycled radios. ORW introduces a new distributed routing metric named Expected number of Duty Cycles (EDC). This metric generates a tree-like structure towards the sink based on the accumulated number of DC through a particular path. ORW is a potential candidate for EH-WSN because ORW nodes forward their packets to the first suitable neighbor who is awake. Thus, it manages to reduce delay and energy consumption, and is resilient to link dynamics.

The metric, EDC, is computed using expected transmission delay and node duty cycle, which is fixed across the network. The simplified formula each node uses to compute locally forwarding cost is given by,

$$EDC(i) = \frac{1}{\sum_{j \in F(i)} p(i, j)} + \frac{\sum_{j \in F(i)} p(i, j) \cdot EDC(j)}{\sum_{j \in F(i)} p(i, j)} + \omega, \quad (2.1)$$

where the first term is the per-hop forwarding delay, the second term models the cost for the rest of the nodes and finally  $\omega$  is the inherent cost of forwarding.  $p(i, j)$  is the probability of successful packet transmission from over link  $(i, j)$  that is measured periodically using beacons or by passive listening. Each node knows the price to forward a packet, its own DC and adds it to the received parent values.

To reduce the forwarding delay of packets, every node in ORW uses a pool of parent nodes that can forward the packets. The forwarder set is computed by adding neighboring nodes sorted by their EDC in ascending order to the forwarder set until a pre-determined threshold EDC is reached. ORW mainly relies on overhearing of transmissions to update its neighbor table as well as link quality estimates.

ORW performs best when deployed in a high-density network where it can exploit routing diversity to forward packets to the sink. However, ORW protocol has not been strictly designed for EH-WSN. Therefore, some issues arise when trying to use it in such scenarios. Firstly, ORW expects nodes to maintain the fixed duty cycle which may not happen in EH-WSNs leading to longer delays or packet losses. Secondly, packets must be periodically

exchanged in order to maintain link quality estimates and forwarder sets. This does not work well when event-based reporting is used. Furthermore, in this case, the EDC values age over time leading to rediscovery of routes before forwarding the packets.

## 2.2 ORiNoCo

Opportunistic receiver-initiated no-overhead collection (ORiNoCo) is another OR based protocol, designed for WSNs but has been adapted for EH-WSNs. ORiNoCo relies on two main components apart from OR, receiver-initiated medium access control (RI-MAC) and load adaptation or adaptive duty cycling.

In RI-MAC, a low power probing method is used. That is, when a sender is ready to transmit, instead of sending, it waits until a receiver broadcasts a beacon. Once this beacon is received, and acknowledged that the receiver provides routing progress, the transmission takes place.

In order to provide a reliable data collection route, ORiNoCo implements an online adaptive duty cycling mechanism. Due to this, a node with more energy attracts more packets. When a node receives more energy, it increases its DC, and hence an increased number of beacons are broadcasted. Thus the possibility of the node to be selected as a forwarder increases. ORiNoCo can utilize Expected Transmission Count or hop count as its routing metric, though it does not explicitly recommend any. Hop count may be preferred since it does not involve any link quality estimation overheads. In Fig. 2.1 we can see an example of how ORiNoCo using RI-MAC works. As observed, the communication is initiated by the receiver which starts broadcasting beacons to the channel and awaits a sender to acknowledge the transmission.

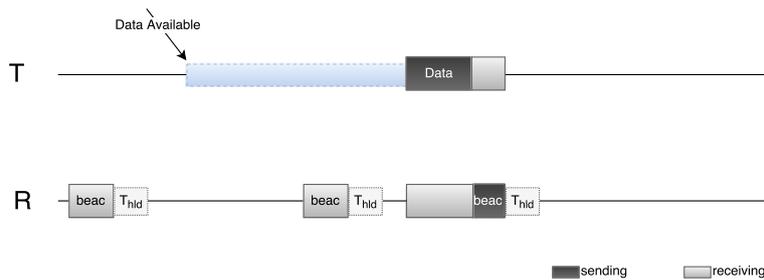


Figure 2.1: Communication in ORiNoCo [7].

Adapted ORiNoCo encounters some difficulties in more challenging indoor EH scenarios. Firstly, when there is excess of energy, beacons from multiple nodes may collide. Although RI-MAC introduces some randomness to avoid collisions, as density and energy level increase, the collisions become unavoidable. Secondly, in the low energy scenario, when the energy

levels are low, the route chosen may not be reliable anymore since packets get attracted to nodes with slightly better forwarding metric when the receiver had energy. However, since the metric does not include any energy harvesting components in them, a good routing decision will not be made. We particularly address this aspect through our novel gradient.

## Chapter 3

# System Model

Two essential requirements for our AAL scenario: (i) the ability to be available, i.e., be operational for extended periods of time, and (ii) scalable, i.e., be able to include more technologies (e.g., new on-body sensors) over time. In this chapter, we shall describe a network architecture to support such an AAL system which fits our study case. Furthermore, we shall also specify the energy models for the nodes since we consider all nodes to be energy-harvesting.

### 3.1 Network Architecture

We depict a typical deployment of the AAL network in Fig. 3.1. The network consists of several heterogeneous devices - wearables and statically deployed wireless sensors - and a sink or a gateway node to collect and process the data of activities of daily living. As shown in the figure, the nodes communicate with the sink over multiple hops. As can be seen, each node can have different energy levels.

Similar to AlarmNet [39], we consider two types of nodes: *emplaced* nodes and *wearables* or mobile nodes. The emplaced nodes are statically deployed in living spaces to measure temperature, light and other environmental parameters along with assisting in determining a user's activity. For example, emplaced nodes can detect if the user is in the vicinity of a sensor either through motion sensors or indirectly (e.g., through lights, water usage using micro-water turbines, etc.). The second goal of this *static* network is to act as a backbone network for the mobile nodes.

The location of emplaced nodes may be carefully chosen. However, without the loss of generality, we consider the deployment to be random. The size of the deployment area in an AAL scenario, as expected, is less than many outdoor deployments (e.g., agricultural fields). Both the number of nodes and the number of hops may be limited depending on the actual size of deployment.

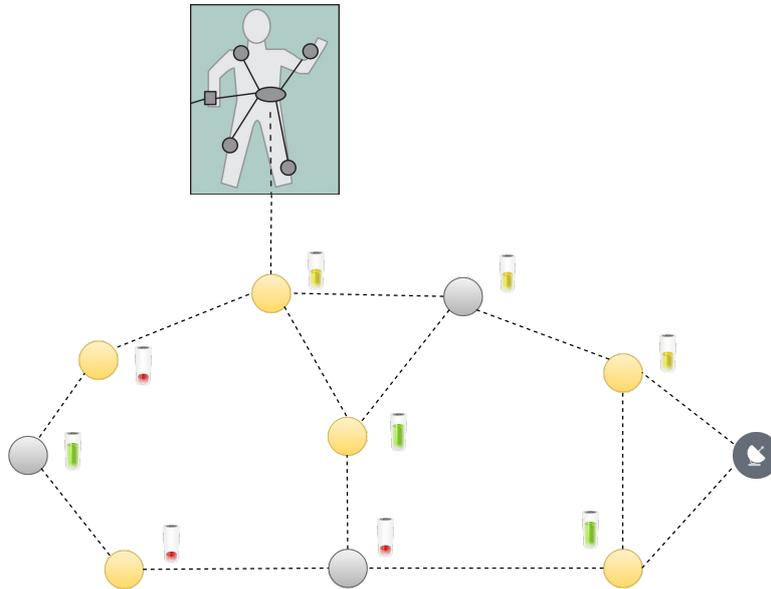


Figure 3.1: Network architecture composed by different emplaced nodes and wearables. Different colors indicate nodes being powered by different harvesters.

The mobile nodes can either be part of a wireless body area network or an individual wearable sensor that has to log data at the sink for further processing. They provide a variety of information either relating to physiological parameters, such as pulse rate, or activity data, such as from accelerometers, informing about the users movements. The position of the nodes varies as the user moves within the house. Therefore, the mobile nodes would look to send their data to one of the emplaced nodes for further routing or to the sink directly when possible.

## 3.2 Energy Model

To make the nodes operate perpetually, we consider all the nodes to be powered by harvesting energy. In this section, we shall describe the harvesters, storage elements and consumption model.

### 3.2.1 Energy Sources and Harvesters

We can have different harvesting sources to power the nodes. As mentioned in Chap. 1, in indoor deployments, a wide range of harvesting sources is available to power the nodes light, movement, water, vibrations, air and heat.

Emplaced nodes with photovoltaic (PV) harvester can make use of sun-

light that irradiates into the building through windows and the indoor lighting systems. Even though these harvesters have multiple light sources, they together provide less power than the outdoor equivalent since the node may not receive direct sunlight on its PV panel. Furthermore, some nodes will only have a fraction of reflected light from indoor lighting sources due to their placement.

Fig. 3.2 shows a trace of power harvested from PV in one of the nodes placed in a bookshelf in Columbia University, New York. Contrary to outdoor PV harvesters, indoors the energy need not be periodic. Furthermore, it is highly dependent on its location or events taking place around it, i.e. different patterns on a weekday and weekend.

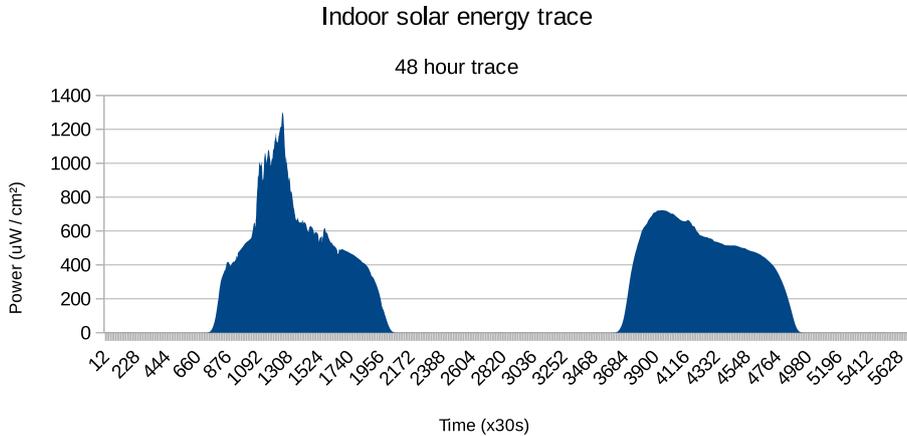


Figure 3.2: Two days indoor solar energy data, obtained from [16].

Other harvesters that can potentially be used in this scenario are water flow harvesters, which scavenge energy from water taps, and showers using micro-turbines [33]; air flow harvesters placed in the ventilation or AC ducts; *Peltier* harvesters that harness the residual temperature from heaters, and piezoelectric harvesters which scavenge energy from the moving parts like fridges, and washing machines [34]. Due to the relatively high energy harvested from light sources, it is one of the best candidates for supporting the emplaced nodes.

The mobile nodes are mainly powered by human motion by placing harvesters in shoes and textiles [24]. Inertial kinetic energy harvesters convert human motion into energy and can generate energy enough to send a few packets to a low-power node. Fig. 3.3 shows a trace of power generated from a shoe-implant harvester [40]. We see that the energy is harvested in small quantities and in bursts.

This implies that a mobile node cannot be made to operate always even with energy-efficient techniques such as duty-cycling. Such a node will turn

on when the energy is being harvested and will die soon after the user stops his activity.

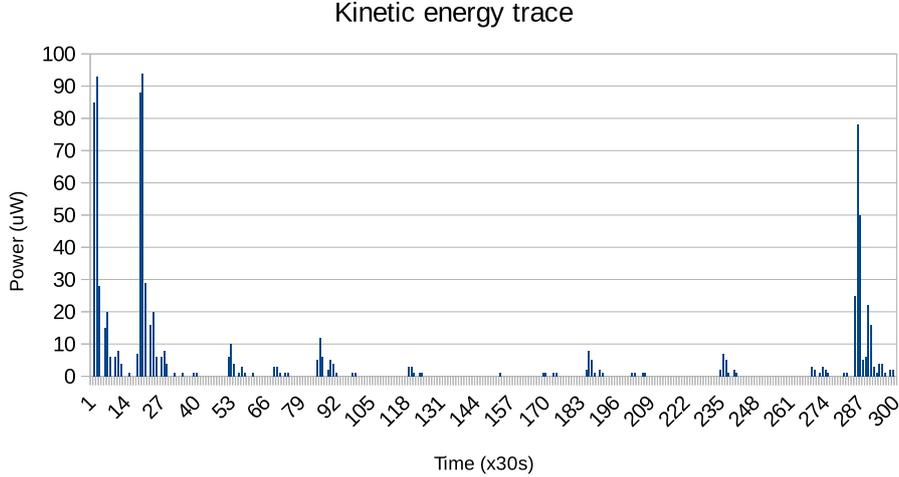


Figure 3.3: Kinetic energy data, obtained from [17].

### 3.2.2 Storage Element

We consider super-capacitors as the storage elements on the nodes. A super-capacitor has high energy density and theoretically unlimited charge-discharge cycles, unlike rechargeable batteries. The remaining energy in the supercapacitor can be estimated by  $E \approx \frac{1}{2}CV^2$ , where  $C$  is the capacitance of the supercapacitor and  $V$  is the voltage. Remaining energy is an important factor in our protocol as we will see in Chap.[4]. Other types of storage elements such as rechargeable batteries may be used, however, estimating their charge in real-time is an issue. Sophisticated techniques such as [41] are being developed and can be utilized in the model when available. Nodes can typically operate above a certain voltage, by limiting the battery capacity further. We neglect leakage of the super-capacitors in this work.

### 3.2.3 Energy Consumption Model

We consider the energy-neutral model [20] stated as follows: suppose the harvested power from the energy source is  $P_s(t)$  at time  $t$ , and the energy being consumed at time  $t$  is  $P_c(t)$ , the following equation should be satisfied:

$$\int_0^T P_c(t)dt \leq \int_0^T P_s(t)dt + B_0 \quad \forall T \in [0, \infty) \quad (3.1)$$

where  $B_0$  is the initial energy stored in the energy buffer.

As an implication of this model, a node wakes up when it has the energy to operate for a minimum period. Therefore, the nodes wake up asynchronously. Nodes try to maintain a certain duty cycle if the energy in the buffer supports the operation.

### **3.3 Assumptions**

We make two simple assumptions in this work: (a) we assume the network of emplaced nodes is initially connected, (b) we consider the radio channels to be symmetric, i.e., if node  $u$  can send a message to node  $v$ , then  $v$  can send a message to  $u$ , and (c) the mobile nodes always move within the range of at least one emplaced node.



## Chapter 4

# Harvesting Energy Aware Routing Protocol with Adaptive Duty Cycling (HEAR-DC)

There are several well-known routing protocols for sensor networks in the literature (e.g., CTP [15]). Most of these protocols have overheads for link-quality assessments and route maintenance. Opportunistic Routing (OR) based routing protocols avoid these overheads to a great extent [37]. ORW [14] has been proposed for battery powered sensor networks and ORiNoCo [37] has been adapted for EH-WSNs.

The amount of energy in the buffer on an EH node that uses the above mentioned OR protocols is influenced by four factors:

1. the rate of sending and receiving packets,
2. the rate of harvesting energy,
3. the rate of beaconing of the low-power MAC, and
4. duty cycle of the node.

The first two factors can only be passively measured by a sensor node. The second two factors can be controlled, and these factors influence the node outage<sup>1</sup>, End-To-End delay, and packet losses. In this chapter, we describe the design of our routing protocol, Harvesting Energy Aware Routing protocol with adaptive Duty Cycling (HEAR-DC), that is based on OR, and uses harvesting and residual energy levels actively to enable reliable

---

<sup>1</sup>A node outage is said to occur when it runs out of energy to participate in network activity.

data collection. We provide an overview of HEAR-DC in Sec. 4.1. In the sections that follow, we describe the communication process in Sec. 4.2, then we describe our novel gradient (routing metric) in detail in Sec. 4.3, and finally the bootstrapping and steady-state phases of the protocol in Sec. 4.5.1 and Sec. 4.6 respectively. Finally, in Sec. 4.6.1 we introduce adaptations to HEAR for including mobile nodes and operating when the packets are generated sparsely.

## 4.1 Overview of HEAR-DC

There are four main components that make HEAR-DC<sup>2</sup> tailor-made for EH-WSNs.

1. *Transmitter-Initiated MAC with Opportunistic Reception and Duplicate Avoidance*: HEAR-DC uses a variant of Box-MAC-2, an asynchronous, transmitter-initiated low-power listening MAC with high energy-efficiency. Furthermore, we enhance the protocol with the following tweaks: (a) when a node has a packet to transmit but finds another on-going transmission, it opportunistically listens to the packet if that packet can be forwarded. This can reduce the end-to-end delay. (b) If a node finds out that a packet in its queue is being forwarded by its neighboring node, then it does not forward the packet to prevent duplication.
2. *Gradient*: An ‘active’ gradient or routing metric that includes the residual energy and harvesting rate of a node and is used so that the packets are routed through nodes with high energy levels. This offers more reliability, i.e., lower packet losses.
3. *Adaptive Duty Cycling*: Nodes adapt their duty cycles according to the energy levels present in their super-capacitors. This makes the high energy nodes attract more load towards themselves which can again influence the reliability and end-to-end delay.
4. *Mobile Nodes and Sparse Data*: Mobile nodes run a light-weight version of the protocol, i.e., they act just like data sources. They look to forward packet to any nearby emplaced node or the sink. Since events in a residential setting are rather sparse, the network must be able to send this data. Most OR protocols include aging of their routing metric, leading to larger end-to-end delays and less reliable routes. We propose to overcome this effect by introducing dummy packets with very low rates when there is no traffic for a considerable duration.

---

<sup>2</sup>The version of our protocol without adaptive duty cycling is called HEAR.

## 4.2 Transmitter-Initiated MAC

The communication mechanism that allows our protocol to collect information is based on Opportunistic Routing. This means that all information exchanges are based on *anycast* communication primitives. There are no rigid transfer structures (e.g., routing tables) set. Therefore, any node can forward a packet as long as the gradient of the receiver is lower than the sender. An example is shown in Fig. 4.1, wherein the packet of node 1 is received and forwarded by nodes 2 and 3 in the direction of the sink. The

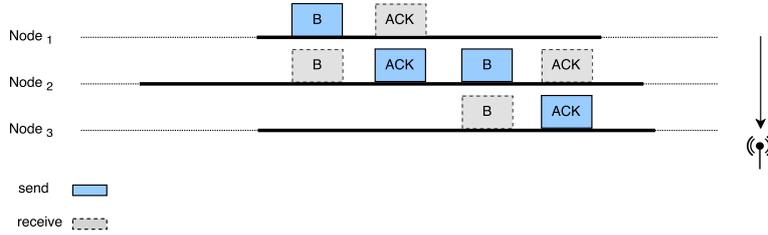


Figure 4.1: Anycast based forwarding.

medium access control employed is transmitter initiated and is a variant of Box-MAC-2 [27]. Here we shall describe our method and not contrast with Box-MAC-2.

The communication between two nodes will always be initiated by a node that has a packet to send in its queue (sender). This node first does a clear channel assessment (CCA) for the duration of a  $t_{inter-strobe}$  to avoid any collisions with on-going transmissions. If the CCA indicates the medium is busy, then an opportunistic reception mode is invoked (this will be described in Sec. 4.6). If the medium is free, then the node begins to send beacons, which we also refer to as strobes. After every strobe, the node waits for an ACK from one of its forwarders with a lower gradient. The beaconing is repeated until a receiver node successfully acknowledges the packet or the sender has reached the maximum transmission duration marked by  $t_{op}$ .

If a node receives a beacon and is a suitable forwarder, then the node will send an ACK. However, before sending the acknowledge packet to the sender, the node employs a collision avoidance mechanism, i.e., it waits for a guard time  $t_{guard} = rnd([0, T_{guard_{max}}])$ . The goal of holding the transmission back is to give time to other possible forwarders to acknowledge the packet. By setting a randomized time to reply, the chances of both forwarders answering at the same time instant are reduced. Furthermore, if the node hears ACK from another forwarder, then the packet is not ACKed by this node and dropped from its queue to avoid duplication. This mechanism is shown in Fig. 4.2.

The timings of strobing are indicated in Fig. 4.3.  $t_{strobe}$  is the time for actual air-time of the packet.  $t_{inter-strobe}$  is the inter beacon period.  $t_{rx-tx}$

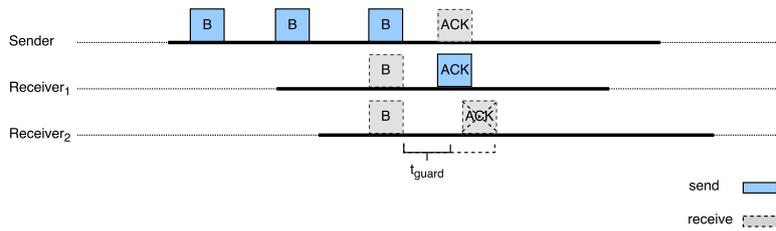


Figure 4.2: Collision prevention mechanism. Receiver 2 drops the packet after overhearing receiver 1 acknowledges it first.

is the time required to change the radio from receive mode to transmit mode, and  $t_{tx-rx}$  is the time required to change the radio from transmit mode to receive mode. These numbers can be easily determined and used to calculate the number of strobcs that can be done with a particular ON period of the node.

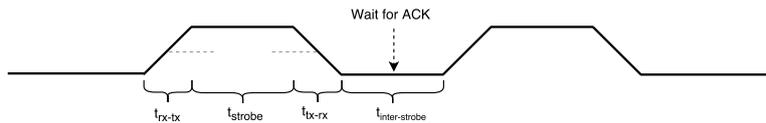


Figure 4.3: Strobe timings.

We use a common frame format for all packets - beacons and acknowledgments - since we assume the packet length to be small and to keep it simple. This structure is shown in Fig. 4.4. By modifying the ‘TYPE’ field, a node can indicate if the frame is a strobe or ACK. The ‘GRAD’ field contains the node’s gradient value. This value is also piggy-backed by the receiver acknowledging the packet. All the listening nodes will update this value, which will be described in detail in the following section. The ‘DATA’ field contains the actual data. ‘TTL’ is used for statistical purposes (to keep track of packet path). Finally, the ‘DST’ field is used either to send the packet into the sink’s direction or to mark which node a particular ACK is aimed at.

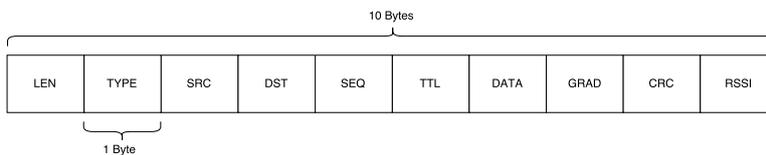


Figure 4.4: Packet field structure.

### 4.3 Gradient

The core element on our protocol is the novel gradient, which is tailor-made to operate in EH-WSNs. The gradient is a routing metric that allows each node to decide if a packet is traveling in the right direction.

The connotation of what is the ‘best’ route in EH-WSNs is debatable. This is mainly because of energy variations in the nodes due to energy harvesting. Using hop-count as gradient gives the shortest route, but packets may not reach the sink. EDC metric of ORW does not take the energy variations into consideration, also leading to packet losses. Therefore, in our context, we define the most reliable route (i.e., most number of packets being delivered) between the sink and a node as the best route. Instead of computing the best route may require gathering knowledge about the energy levels, harvesting rate and the number of packets in its queue at a central location, we shall build a route online using the gradient.

In our design, we try to shift the traffic to the nodes with higher energy values. The purpose in doing so is to boost the chances of a packet reaching the sink successfully, by hopping over reliable nodes. The key parameters that affect this metric value are (a) position in the network; (b) residual energy; and (c) energy harvesting rate.

Using these parameters, we construct our novel gradient as given in Eqn. 4.1.

$$gradient_n = MIN(\overline{grad}_n + \rho_{max}/\rho + \sigma, GRAD_{max}), \quad (4.1)$$

where  $\overline{grad}_n$  is the average gradient of the neighboring nodes that forwarded the packets of node  $n$ .  $\rho_{max}/\rho$  is the inverse of the normalized value of the current duty cycle of the node, and  $\sigma$  indicates the node’s harvesting state. And  $GRAD_{max}$  is a constant defining the maximum value the gradient can have.

The gradient of the sink is initialized to zero. Therefore, the first hop nodes will have a very low average gradient. This value increases continuously as the distance from the sink increases. However, nodes with higher energy values will have slightly lower gradient due to energy-related parameters (second and third parameters in Eqn. 4.1). This makes the nodes with higher energy levels more attractive than the lower energy level nodes.

A node harvesting energy can buffer the energy and be operable for several slots, making the node more reliable than a node with an equal amount of residual energy. Furthermore, if low energy level nodes surround this node, then the route is not reliable, and the packets are likely to be lost or delayed. This aspect is incorporated due to the average gradient, hindering nodes from choosing this route. We call this an ‘active’ metric since energy parameters of a node are made known its neighbors as described in the previous section, and therefore can make them part of the forwarding decision for a packet. This is in contrast to ORiNoCo. Wherein the energy related parameters are not disseminated, and high energy node attracts packet only

because of its adaptive duty cycling mechanism. In the situation of a high energy node being surrounded by low energy nodes, the packet will mostly be lost in ORiNoCo.

An example of the slope created by the gradient dissemination is shown in Fig. 4.5 exemplifying a situation where a longer route presents a lower accumulated gradient due to different residual energy. The forwarding cost of each node depends on the current state of a particular node, and the path to the sink is composed by the addition of those forwarding costs. In some cases, a node would forward a packet through a worse route if the receiver node awakes before other better nodes. This case can be observed in the node right of the origin. However, after the gradient has stabilized a node will pick only those routes that have lower gradient than its own.

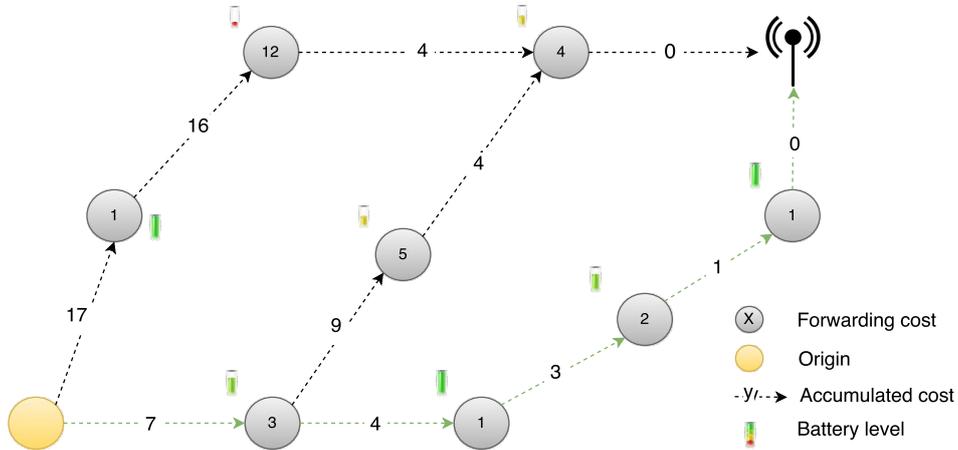


Figure 4.5: Routing with gradient. Dashed lines mark the accumulated cost through that specific path and the node value its forwarding cost.

The parameter  $\rho$  represents the duty cycle of a node. In case the node employs an adaptive duty cycle that is proportional to its residual energy, this value  $\rho$  gives the indication of residual energy in the gradient. Each node computes its DC at the beginning of each operation cycle and then normalizes this DC dividing it by the maximum fixed DC. By using the inverse of this value (i.e.,  $\rho_{max}/\rho$ ), nodes with better power capabilities will present a lower gradient value, and thus, be more attractive to its neighbors. In case the node employs a fixed duty cycle, this ratio is constant over time and acts a fixed forwarding cost.

The remaining parameter,  $\sigma$ , maps the harvesting state of a node into its metric. As  $\rho_{max}/\rho$  represents the long-term energy fluctuations of the node,  $\sigma$  indicates the short-term energy capabilities of the node. The method to compute  $\sigma$  aims to account for the harvest variations of each node by encapsulating an average of recently collected values from a particular node into a node harvesting state (NHS). Its goal is to represent the future energy

tendency of a node. This NHS value is not a static value, but is dependent on the time and the harvester type used by the node. With such value, a node will be able to modify its behavior to adapt to its future energy state, making a node’s metric more or less appealing to others depending on a snapshot of its harvesting history. Thus, nodes that are seeing their harvesting rate (HR) being reduced will redirect part of its traffic to neighboring nodes. Contrarily, if they start harvesting at higher rates, they will expose themselves to the networks as better forwarders.

We compute NHS as  $NHS = \frac{\int_t^{t+T} tP_c dt}{\int_t^{t+T} tP_s dt}$ . The outcome of this operation is either a value bigger than one when consumption is greater than harvesting, or a number between zero and one when a node has a higher energy input than the quantity it is using. We call it state *LOW* in former case and state *HIGH* in the latter case. Both *LOW* and *HIGH* are assigned a value that acts as forwarding cost (e.g., *LOW* is given 3 and *HIGH* is given 1). This value is what we define as  $\sigma$ , which is a step-like, natural projection of the value NHS. Therefore, a node with a *HIGH* state will have a better gradient than a node with a *LOW* state. With this differentiation, we shift traffic from nodes with worse harvesting parameters to those with better projection.

*Hysteresis.* A node’s NHS state can oscillate depending on the amount of energy being harvested. Therefore, to reduce oscillations due to rapid power fluctuations a hysteresis mechanism has been added to it. Instead of changing  $\sigma$  the same moment a node reaches a certain value, depending on the direction of the change, it will compare its value with a higher or lower threshold, reducing the noise a node can experience when exposed to burst, fluctuating energy sources.

With this method, nodes that just briefly start harvesting more energy will not endanger its operation state by fluctuating to a higher  $\sigma$  and will not attract much traffic. The same approach applies when comparing with a lower state. Nodes that had a transitional energy fading won’t automatically jump to a lower  $\sigma$ .

*Aging.* A node’s gradient not updated after certain duration may no longer hold due to the energy variations. Therefore, an aging mechanism has been incorporated to degrade the node’s gradient. When a node uses the gradient value of a forwarder node, the sender node notes the amount of time used to establish the connection with that forwarder as *rendezvous-time* ( $RV_t$ ). After the average gradient is computed, the  $RV_t$  is used to set an aging prevention window for that entry. This window defines an aging prevention time for each of those entries.

Therefore, each entry has a different age prevention time before it starts to age, substituting first those connections of worse quality and maintaining a better gradient overall. This age prevention time is defined as a natural number of wake cycles before the metric starts the degradation process.

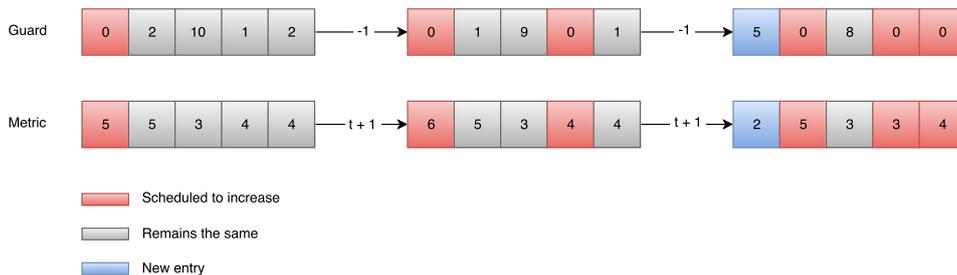


Figure 4.6: Aging and age prevention mechanism. Marked in red are the entries that are currently degrading its value. In gray color is set the age prevention number of iterations left.

With this technique, a node that finds that its neighboring nodes present worse gradient values than its own, will eventually degrade its gradient and reestablish the connection with the network. In Fig. 4.6 we can see an example of the aging and age prevention mechanism. Marked in red those entries that have reach age prevention value zero and will start to degrade and in gray those that still have iterations left.

Extreme situations occur when a section of the nodes has not forwarded any packet for a long duration. The gradient in every node ages and the network has to re-initialize. To avoid such a situation, each node generates a dummy packet with very low probability and forward it to update gradient entries.

## 4.4 Adaptive Duty Cycle

The duty cycle (DC) of a node is the ratio between the duration of time it is awake and duration of time it is in the sleep state. It is possible to adapt the duty cycle of a node proportional to its residual energy level. Adaptive duty cycling are advantageous for the following reasons: use the surplus of energy from the high harvesting nodes to attract more traffic and reduce end-to-end delay by keeping nodes awake for longer and therefore, increasing the chances to meet other nodes.

We implement the adaptive duty cycle method as proposed in [38]. In this method, the duty cycle duration is modeled as a control system. The objective of the control system is to minimize the distance between the current battery level to a set point, and the control variable is the duty cycle that is varied depending on the distance. This system can be modeled as a discrete, first order, linear dynamic system, and the problem can be formulated as a linear-quadratic (LQ) tracking problem. The model of the system is presented in Eqn. 4.2.

$$y_{t+1} = ay_t + bu_t + cw_t + w_{t+1}, \quad (4.2)$$

where  $y$  is the duty cycle output value to correct the deviation,  $u$  is the control variable,  $w$  is a mean zero input noise and  $a, b, c \in \mathfrak{R}$  are discrete step coefficients. The objective of the controller is to maintain the average value of  $|y_t - y^*|^2$  as small as possible over infinite time horizon as given in Eqn. 4.3. Here  $y^*$  is the desired set value.

$$\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=1}^N (y_t - y^*)^2 \quad (4.3)$$

We use iteratively the optimal control law 4.4 that minimizes Eq.4.3 to obtain the appropriate duty cycle output value for a specific node.

$$u_t = \frac{y^* - (a + c)y_t + cy^*}{b} \quad (4.4)$$

Solving for  $u_t$  in Eqn. 4.2 is not straightforward because the coefficients used in the process are not known a priori. In order to estimate the coefficients and solve for  $u_t$  as in Eqn. 4.4, a gradient descent technique [22] is proposed. This mechanism results in the definition of a parameter vector and an iterative approach to computing those values online. We refer the reader to [38] for the algorithm.

We tested the DC controller in a stand-alone code with an input of 24 hours of real solar data from Columbia University. Fig. 4.7 shows how the duty cycle adapts to the residual energy in the node.

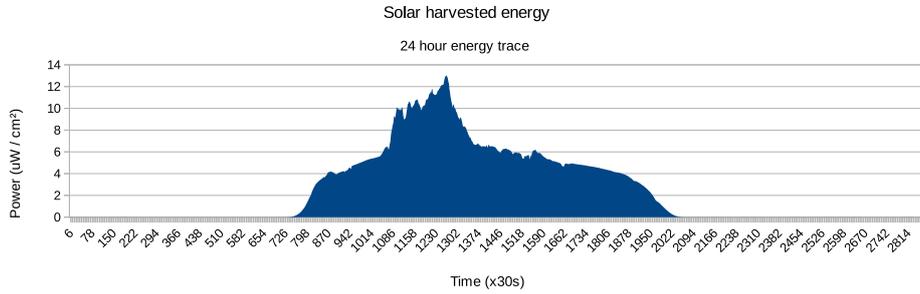
We set the duty cycle to be within  $[\rho_{min}, \rho_{max}]$ . By including this adaptive DC mechanism into our system, we aim to boost the duty cycle of each node to the maximum possible, so that each node will increase its ON duration for as long as its energy capabilities allow it. This will keep those nodes with better energy level to operate for longer durations and therefore, attract a bigger part of the network traffic to them. We shall call the version of HEAR that incorporates adaptive DC as HEAR-DC.

## 4.5 Operation of HEAR

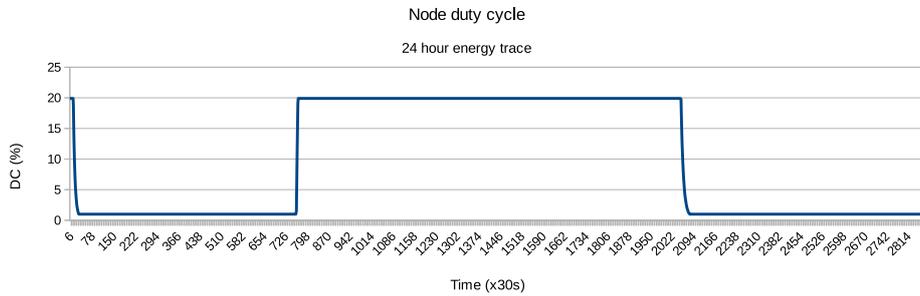
In this section, we shall describe the operation of our routing protocol HEAR.

### 4.5.1 Bootstrapping

In the beginning, each node sets its gradient to the maximum. With this value constant over the network, no data collection is possible because no node knows their position in the topology. Each node generates some dummy packets at this phase to compute its gradient to the sink. At this instance,



(a) Solar energy harvested.



(b) Adaptive duty cycle variations.

Figure 4.7: Adaptive DC controller output to EH input.

each node, when awake, will broadcast its payload. However, because every node has the same gradient, no exchange will take place. This phenomenon applies to all nodes except those nodes located within the communication range of the sink.

The 1-hop away nodes will receive an acknowledgment from the sink, which has a gradient 0. After these nodes update their own metric with respect to the sink, they will have a lower gradient value than the rest of the network. This process repeats for the second hop nodes and continues to update the rest of the network. The computation of gradient has been explained in Sec. 4.3.

The bootstrapping phase is said to end when every node has a gradient value smaller than its initial value, and is more or less steady for a duration of time. After this, the node can switch to the steady state operation phase. In our implementation, we do not explicitly mark the phase boundaries. However, a node does not forward a packet if its gradient is maximum.

## 4.6 Steady State Operation

During the steady state operation phase of a node, the node repeats a set of operations each time it wakes up. The first step is to estimate the amount of remaining energy and saves it into memory to compute the harvesting rate for future iterations. Then it computes the harvesting rate and updates the  $\sigma$  value of the gradient. Next, the node computes the duty cycle it can operate in. For HEAR, we set it  $\rho_{min}$  always, while HEAR-DC calculates this value using the method described previously. The duty cycle will set the operation duration,  $t_{op}$  which defines the maximum amount of time a node can stay awake before going to sleep.

After the required parameters have been defined, the node computes its gradient. It then enters the main loop, which can be observed in Fig. 4.8. The node will remain in this loop until the node sleeps (i.e.,  $t_{op}$  is not reached). The first action taken by the node is to check if its queue contains packets to forward or not. If there are packets in the queue waiting to be forwarded, the node enters a transmission phase. Otherwise, it would start listening for incoming transmissions.

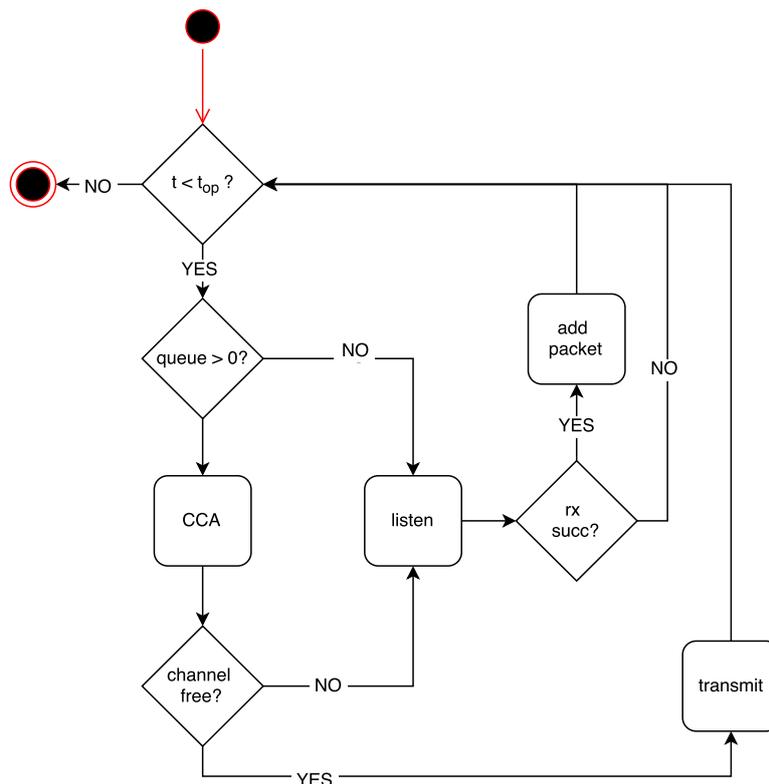


Figure 4.8: Main operation loop diagram.

In the transmission phase, the node probes the channel for currently on-

going communications (CCA). In this step, two scenarios can happen. If the channel is free, the node can proceed to strobing. However, if the channel is occupied, then the node will perform *opportunistic reception*. That is, the node will try to receive the on-going transmission. In case the node is a suitable forwarder for the received packet, then the node will try to acknowledge the transmission and add the packet to its queue. During the opportunistic reception, if the packet is not successfully received, then the node would wait for the channel to become available. The amount of time the node can wait for the channel to become free is equivalent to the inter-beacon time  $t_{inter-strobe}$ .

In case the node starts the beaconing process, the node stops beaconing in two conditions, whichever happens first. Either  $t_{op}$  reaches the maximum, or the packet is ACKed successfully. However, if the node has received an ACK and some ON duration is still available from the time set by  $t_{op}$ , it loops again to find if there are packets in its queue.

As stated above, if a node does not have queued packets to transmit, it will enter a listening phase. In this phase, the node waits for incoming transmissions. In the favorable case that the node overhears a communication, and it can provide routing progress, then node acknowledges the transmission and add the packet to the queue. Similarly to the sending phase, the listening phase has two exit conditions. First, the main condition imposed by the timer  $t_{op}$  being over, and secondly, successfully receiving and acknowledging a packet transmission. After either condition is met, the node will, in the case of having remaining available time, go through the main loop again.

#### 4.6.1 Adaptations

We propose two adaptations to HEAR for making the network operate well even when the packets are generated sparsely and accommodating mobile nodes. We describe them in this section.

##### **Sparse Data**

In a residential setting, at times there are no events to be reported for long durations. For example, during night times there are usually no events to report. The gradient, as described in Sec. 4.3, ages over time. Over a sufficiently long duration with no packets in the network, the gradient can reach the maximum value. This would necessitate a bootstrapping phase again, which is a costly process. Therefore, nodes generate dummy packets with a low probability once the metric in a specific node starts experiencing certain degradation. With this technique, when the network works under a reduced traffic, instead of undergoing a general gradient degradation due to aging, each node participates actively in maintaining that gradient.

## **Mobile Nodes**

The system has been dimensioned to be able to operate under low energy parameters and sparse traffic. These features allow us to include mobile nodes in the network without much difficulty. Mobile nodes maintain a non-initialized metric for their whole operation duration. They opportunistically transmit their data to any emplaced node in their vicinity. Moreover, because mobile nodes cannot forward packets from other nodes due to their non-initialized metric, they use all their energy in transmitting their packets whenever the channel is free.



## Chapter 5

# Trace-based Energy Harvesting Simulations

In this work, we have developed a module for Cooja that allows for simulating energy-harvesting on a sensor node, and hence the network, by using real data traces. This module is generic and straightforward to use, and can easily enable close to real-world EH-WSNs protocol evaluation. In this chapter, we describe this module in detail before moving on to evaluating HEAR.

We begin with listing the advantages of using real data as the energy input for simulations in Sec. 5.1. Next we describe our module in detail in Sec. 5.2 and 5.3. Finally, in Sec. 5.4 we summarize the implications of taking this approach and present some open questions about trace-based simulations.

### 5.1 Advantages of Trace-Based Simulations for EH-WSNs

Most protocol evaluations for EH-WSNs today are being done by simulations [4], while only a few of them are deployed in real-world with harvesters. This is done since real-world evaluation has several disadvantages: (a) each node should have a harvester built, which implies high cost to deploy; (b) it takes a long time to evaluate in all conditions (for e.g., one year for a solar harvester to cover all seasons); (c) a dense network is required to cover many locations as energy harvesting varies over locations; (d) node and software failures must be monitored and handled quickly; (e) data collection from a large network could be challenging in real-time, and (f) experiment repeatability is challenging.

Furthermore, most simulations use stochastic distributions for energy arrivals, which is not the reality. Stochastic distributions fail to capture bursts of energy arrival, correlated energy arrivals for one node and among neighboring nodes. Moreover, implementing a stochastic distribution on a real

node is further approximated due to limited floating-point computation capabilities. A data trace-based simulation, on the other hand, is much closer to reality.

Considering all these factors, we designed a simple battery module that provides trace-based energy arrival. The simulations thereof have the capabilities of (a) being fast, (b) evaluating in conditions close to reality, and (c) allowing repeatability of experiments. Since Contiki is a popular OS for WSNs and Cooja is its simulator, we wrote the EH module for Cooja.

## 5.2 Data Traces of Energy Arrival

In this section, we shall list a few datasets for a variety of harvesters and deployments that we can make use of in our EH-module.

1. Outdoor solar data trace is the simplest to obtain since meteorological department across the world save the solar irradiance ( $\text{W}/\text{m}^2$ ) values. Smart-grid initiatives further fuel data collection, initiating other sources (such as CONFRRM [10] in the USA) to collect solar data as well.
2. Indoor PV harvesting data trace in five different setups have been collected by Columbia University, New York and made it available online [16]. The five setups are (a) student's office, south-facing, 6th floor, located on a windowsill. (b) Same as (a) but placed on bookshelf far from a window. (c) Departmental conference room, large windows and unobstructed view. (d) Student office; corner window facing South and West, extensive shading used.(e) Student's office; setup on a windowsill with window usually open, receives unattenuated reflected outdoor light.
3. A shoe-implant harvesting data has also been collected by Columbia University, New York and also has been made it available [8]. This data has been collected by five people over several days each with a particular profile. The participants were asked to carry the sensing unit over a period of 25 days, and they used different commute and transportation means: foot and train. This data allowed to generate traces over a 24-hour period. The dominant motion frequency of the collected traces falls between the range of 1.92-2.8 Hz, which corresponds to human walking. This data is then converted into energy following the process described in [18].

At this time, the number of such traces is limited but we expect it to increase as more and more deployments of EH-WSNs will be done.

## 5.3 Energy Harvesting Module

### 5.3.1 Design and Implementation

In this section, we describe the design process and the implementation details of the Energy harvesting Module. An application is designed to use energy traces on Contiki's Cooja simulator.

1. The EH-model is used as an external module to Contiki's stack. As showing in Fig. 5.1. The HEAR application can invoke and use the EH-Module. Initializing the stack is easily accomplished by importing the source code and call the core Contiki structures in the node's main code. Contiki applications are built upon the Contiki Netstack. The Contiki Netstack can be observed in blue in Fig. 5.1. In the implementation of the HEAR protocol, we overwrite and merge the different stack modules in a way we can control at low level the node's behavior. The HEAR protocol is composed of two main blocks: HEAR application (gray) and the energy harvesting module (yellow).

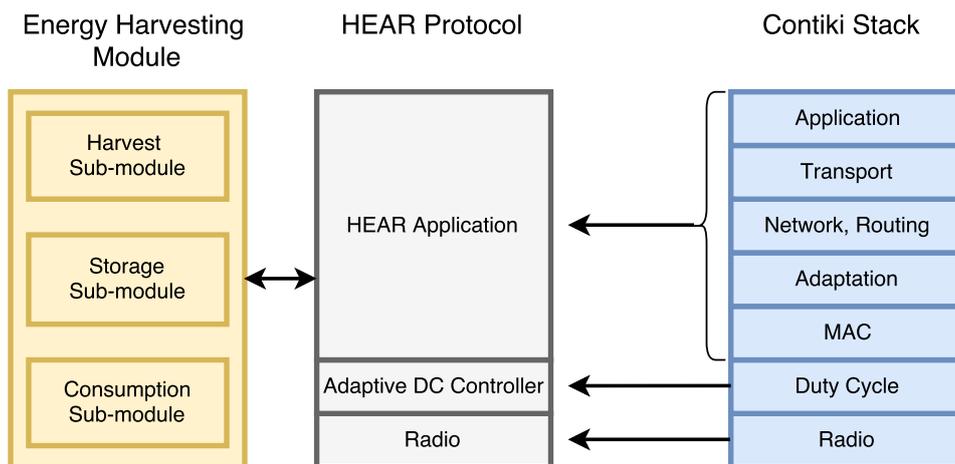


Figure 5.1: Contiki Netstack based on [9] and HEAR block diagram.

2. The energy harvesting module works as a separate program from the main node code and is organized in three sub-modules: *harvesting*, *storage*, and *consumption*. A block diagram of their dependencies can be observed in Fig. 5.2.
  - *Harvesting*: This sub-module is in charge of managing the node's input energy. The first action is to load values from a trace file into memory. Once these values have been used, the harvester module will reload the next energy trace values into memory. To obtain the energy values the harvesting module needs access to

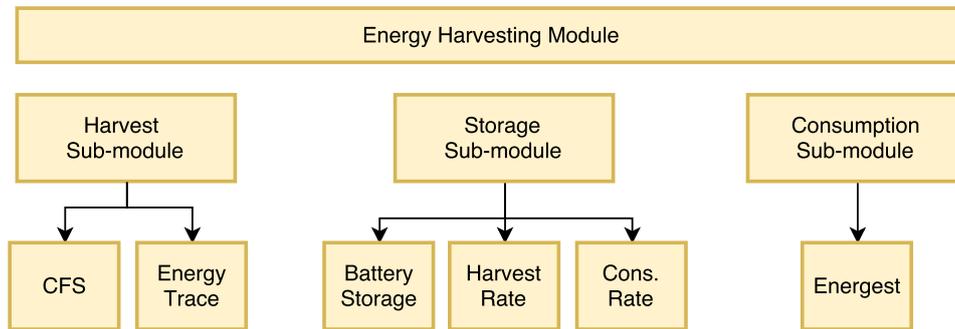


Figure 5.2: Energy harvesting module block diagram.

the energy file uploaded into the node. To do so, it uses the Coffee File System of Contiki environment. The module will provide periodically a certain amount of energy to the node and enable trace-based simulation.

- *Storage*: The storage sub-module manages the access to the shared energy variables. This sub-module is the link between the harvesting module, the consumption module, and the HEAR application. All the energy information is stored in several variables like the remaining energy, harvesting rate, consumption rate, and so on. The storage module ensures that different modules are not accessing the same variable at once with semaphores as well as validates that a node is not overflowing the supercapacitor storage capacity or depleting it below empty.
  - *Consumption*: The last sub-module uses the *Energest* (Contiki's energy estimation mechanism) information to calculate the amount of energy used by each node component, i.e. radio transmitting, radio receiving, CPU, and so on. This module receives the number of clock ticks from each element. These ticks increase when a particular element has been turned ON. All the different tick values are periodically collected by the consumption sub-module which computes the total consumption value depending on the source of each tick, i.e. the radio or CPU. Finally, the energy value is removed from the remaining battery and the energy statistics updated. Moreover, the system can isolate the manager thread and remove its consumption from the total, accounting only for the node operation consumption.
3. To use the energy harvesting module to simulate with real datasets the user need to perform several actions:
    - *Sampling frequency*: The user needs to configure how often energy is harvested by the HEAR application. This is the frequency

with which the file will be read and the energy is ‘harvester’. Depending on the data granularity of the file, the energy harvesting module will be able to access it at a different rate. In our simulations, the harvesting rate has been set to sample each 10 *ms*.

- *Energy file*: The selection of the file each node will use is defined in a Cooja simulation script. To upload such file it can either be selected manually via the Cooja’s CFS user interface or scripted with *Rhino-Javascript*. The following snipped exemplifies the file upload script:

```
m = sim.getMote(i);
FILEPATH = "/home/user/energy_traces/cfs_file.txt"
int = m.getInterfaces();
fs = m.getFilesystem();
success = fs.insertFile(FILEPATH);
```

- Once the file it is uploaded successfully into each node, then the node energy harvesting module has to initialize the file system and set the initial trace pointer. The address of the pointer can be randomized to add variation to the simulation. This allows several nodes to use the same energy file and still present different behavior. The interaction with the Coffee File System is done via the *CFS* library through the commands: *cfs\_open*, *cfs\_read*, *cfs\_seek* and so on.
- To get power values from the uploaded energy file, the energy harvesting module initially accesses the file and reads several values. These values are stored in memory for faster access. After all the values have been used the energy harvesting module will update them with the next values from the dataset. In case the energy harvesting module reaches the end of the file, it will reinitialize the pointer and start over automatically.

All the above sub-modules are encapsulated into a Contiki app. This code application can be added into a Contiki node code by importing at compilation time and then initialized as a separate proto-thread on the primary node function by calling *energytrace\_start()*. Because is a separate thread the user can set different sampling periods and management parameters without affecting the node performance.

### 5.3.2 Features

Following are some of the features of this EH module:

1. *Different trace files*. Every node can have a different trace file determined before simulations.

2. *Randomization.* Instead of loading different files for all nodes, files can be made to share among nodes. However, each node can start at a different location in the file, randomizing the amount of energy being harvested.
3. *Fine-grained harvesting.* Since the EH module runs as a separate process, it is possible to harvest energy every clock tick of the node. However, such fine-grained harvesting is not necessary since the energy harvested is too small and also increases the file length unnecessarily.
4. *Event triggering.* If events are said to occur when there is a significant change in harvested energy, then this module supports such event triggering.

## 5.4 Limitations and Open Questions

As stated above, using real data traces as the energy input for our simulations has several advantages, i.e. simulations closer to actual conditions, reduced computation load, and so on. However, it has its limitations and some differences from real implementations. We list the limitations, open questions and possible solutions in this context.

1. The reduced available onboard memory limits us to use small files that contain only part of the real data trace. This implies that simulating even a single day is currently not possible. As a consequence, the total simulation time is equal to the file size. In the current implementation, it is approximately 30 minutes. One possible solution is to automate the simulations with an external script. That is, a full day trace is split into 30 minutes' files, and the script starts simulations with the next 30 minutes files each time.
2. In the current implementation, leakage of the super-capacitor is not accounted for. It can be incorporated easily by using a suitable model. Furthermore, any battery model can be integrated easily into the module.
3. On several accounts, the trace-based simulations is only close enough to reality but does not completely behave as it happens in real-life. The first one: typical energy trace from a MET department collects data every five minutes. To use such data traces in EH-WSNs, the energy must be interpolated. One extension we have implemented is to add a zero-mean 'noise' to have some randomization of harvested energy.

4. The second one: several approximations are done with respect to hardware. The efficiency of the harvester and its degradation over time is not accounted for.
5. The third one is that while randomizing nodes to use the same file but starting from different locations allows us to create variability, it does not depict reality as well. There may be correlations in energy harvesting patterns depending on nodes' location. Care must be taken while selecting files for each node.

We can conclude that using trace-based simulations enables us to approximate reality better. However, better tools and trace management might be necessary to reduce the gap between actual deployments and simulations.



## Chapter 6

# Performance Evaluation

In this chapter, we present the results obtained from simulations performed on Cooja, the network simulator of Contiki OS. With these simulations, we compare the performance of ORW, HEAR and HEAR-DC protocols in several scenarios. The structure of this chapter is the following: Sec. 6.1 explains the methodology used to simulate, simulation parameters, and metrics. In Sec. 6.2 we describe the different scenarios used to test the various aspects of the protocols. In Sec. 6.3 which we present the results.

### 6.1 Methodology

We use Cooja, the network simulator of Contiki OS for our simulations along our EH module to enable trace-based simulations. We compare the performance of HEAR and HEAR-DC against ORW in various scenarios. We make use of the indoor energy datasets [11, 1] from Columbia University for the emplaced nodes, and shoe-implant datasets for the mobile nodes. For each data point in the results, we perform several 20-30 minute simulations before averaging the results.

For all simulations, we consider TMote Sky [35] as the wireless sensor node. The energy consumption of all operations is computed by using *Energyest* and the numbers mentioned in the datasheet [36]. All nodes have a 350 mF super-capacitors, which translates to a usable energy of 500 mJ (Note that a node can operate only when the minimum voltage provided by a super-capacitor is 2.7 V with a maximum of 3.5 V provided by the super-capacitor). For HEAR and ORW, the duty-cycle is set to 1% i.e., 1 ms in 1 s. Each node wakes up asynchronously within 1 s when they have enough energy to operate. For HEAR-DC, we set  $\rho_{min} = 1\%$  and  $\rho_{max} = 20\%$ . Unless mentioned, the desired buffer set-point for HEAR-DC is set to 35% of the buffer capacity. The payload is fixed to 10 B in all simulations with the frame format as mentioned in Sec. 4.2. Furthermore, the radio model of Cooja is chosen to be Multipath Radio Model, which provides variations in

packet reception albeit limited.

To create simulation diversity and obtain results statistically reliable, we randomized several parameters on our simulation process. These parameters affect various aspects of the node operation cycle as well as the network in general. These variations are as follows:

- Topology, node and sink position
- Energy harvesting input and initial time
- Mobile node displacement patterns

Topology variations affect the overall performance of a protocol significantly. This is due to the varying energy levels as well. For instance, an extreme case is created when a node having poor energy levels is also the only connecting node between two parts of the network. The position of the sink can also generate a wide span of different scenarios ranging from a centered collection deployment to a narrow linear configuration full of bottlenecks. We randomize topology in each simulation run to gather an average performance of the protocol. Mobile nodes, when used, will also influence the performance since they ‘leech’ onto emplaced nodes. All the generated topologies had a minimum of 2 hops and a maximum of 7 hops, with an average of 5 hops. We chose these numbers since residential areas due to extension dimensions will have around 3 to 5 hops.

The primary metrics for comparison are the reliability and latency, i.e., the average number of packets lost (here, the lack of reliability) and the average end-to-end delay. To better understand the working of the protocols, we also look into other performance parameters such as the mean number of packets created, average remaining energy at the end of simulations, and the average number of hops.

## 6.2 Scenarios

To study the performance of the protocols under different settings and parameters, we have designed four major test-sets as shown in Table 6.1. In every set, we vary a key parameter in order to study this variation.

Test-set	Varying Parameter	Range	$E_0$
High Energy	Num. of nodes	10,20,30,40	95%
Low Energy	Num. of nodes	10,20,30,40	5%
Traffic	Inter packet time	15s,45s,90s,180s, and Event	55%
Mobile nodes	Num. of mobile node	17%, 33%, 50%	55%

Table 6.1: Summary of test scenarios

The first test set studies how the three protocols, ORW, HEAR, and HEAR-DC perform with a different number of nodes deployed in the same network. The number of nodes deployed has been set to the values of 10, 20, 30, and 40 nodes. We consider that 40 nodes will already create a highly dense network in a residential setting. All the nodes will have a high initial battery and will be harvesting high energy levels as well. This is in contrast to the second test set where the nodes will have a small initial battery and will be harvesting low energy levels. In these scenarios, a packet is generated every 90s on an average by every node. We expect to observe the different behaviors how nodes act under high energy inputs and low or no energy harvesting periods. We do not evaluate an average case in between these extreme scenarios since its performance will be in-between these two cases.

The third scenario plays with different packet generation intervals or traffic intensities in the network. Each node will have an average initial battery. However, we set the energy input files such there are some nodes which harvest high energy and some which may not harvest any energy at all. Testing each protocol under different traffic loads which will give us insights on how they perform under high traffic, their capacity to rapidly deliver packets to the sink, or under infrequent packet generation periods, where methods to maintain the metric and aging take a leading role. We also test the three protocols when the packets are generated based on events when the harvesting rate changes, which may generate aperiodic traffic.

All the above test sets are to evaluate the emplaced nodes since they are the ones who route the packets. The mobile nodes as said before will just transfer their load to the emplaced nodes. To evaluate how the network behaves with the additional load of mobile nodes, we introduce them in the fourth test-set. The mobile nodes are powered by inertial harvesters and have low and sporadic energy input. These nodes move around the topology adding traffic to different areas depending on the simulation instance. This experiment studies how the protocols cope with various percentage of mobile nodes over the total number of nodes.

## 6.3 Results

This section presents the results of simulations in all the four scenarios.

### 6.3.1 High Energy Scenario

The protocols use OR as their base methodology. Thus, they all are influenced by the node density. Therefore, the studying of the network with sparse and dense topologies is especially interesting because they give us insight into the performance each protocol.

In the first scenario, there is no bottleneck on energy as all nodes will have high energy arrivals. In this best case scenario, we observe that average

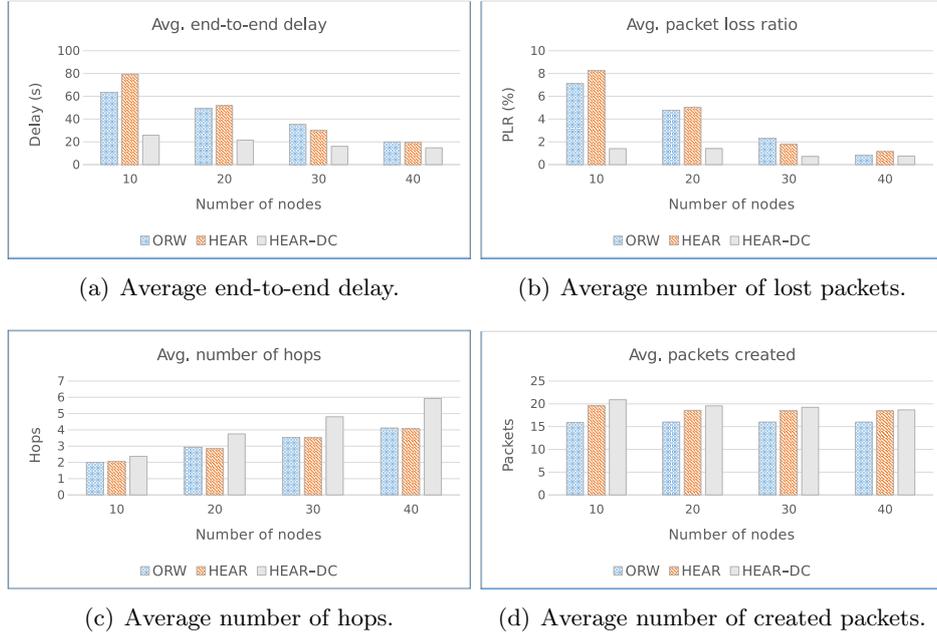


Figure 6.1: Performance comparison depending on number of nodes deployed under high energy environment.

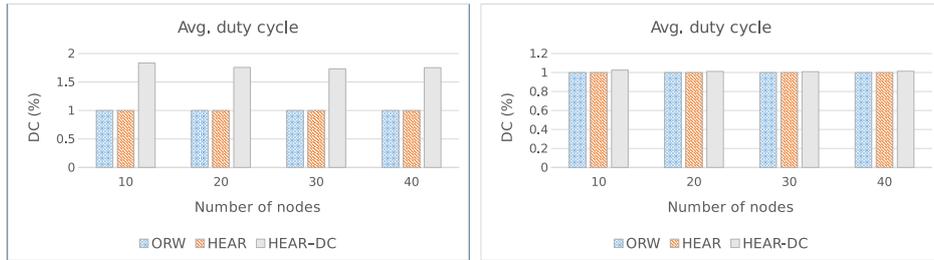
end-to-end delay and average number of packets lost decreases as density increases, as shown in Fig. 6.1. This pattern is shared by all the protocols in different magnitudes. This tendency is induced mainly by the growth on available potential parents to forward. Due to the presence of more, the average amount of time a transmitting node has to wait to find a suitable forwarder is reduced with the growing number of nodes in the network. We observe that HEAR-DC outperforms ORW in all cases. With 10 nodes, HEAR-DC achieves a latency reduction of 64% with 92.75% reduction packet losses as compared to ORW. In the high density case, the gain is much lower since ORW has enough nodes to compensate for the extra DC of nodes in the HEAR-DC protocol. In this case, HEAR-DC achieves latency reduction of just 10%.

We observe that the performance of HEAR is worse than ORW when the number of nodes is less. HEAR has the same duty cycle as ORW, i.e., 1%. However, due to the gradient of HEAR, certain nodes attract much more traffic but cannot send all of them due to limited DC. As the density increases, the chances of finding a forwarder becomes higher; therefore, it starts to perform better.

From Fig. 6.1(c) we can observe that our protocol, in instances, forces the nodes to select longer routes than ORW. This increases the number of hops but comes with the advantage of selecting nodes which present a more

reliable energy pattern than the other protocols. Our protocol (HEAR-DC in particular) redirects traffic to those nodes with a better short term (harvesting rate) and long term (adaptive DC) energy capabilities, which translates into nodes with lower chances to face battery depletion and with better probability to forward. The impact of this is clearly visible in the number of packets lost for different protocols.

In Fig. 6.2(a) we can observe how a small duty cycle difference between ORW and HEAR against HEAR-DC reduced the packet losses up to 92% in certain cases. We can extract from this fact that by slightly increasing the ON time of the nodes when extended to several nodes in the network, the chances to transmit within reduced delay grow rapidly. Secondly, in Fig. 6.2(b) can be easily seen that when the deployment faces a general energy scarcity, HEAR-DC will match the DC from HEAR and ORW, setting it to the minimum 1%.



(a) Average duty cycle under high energy environment. (b) Average duty cycle under low energy environment.

Figure 6.2: Average duty cycle comparison between high and low energy.

The last point to note is the varying number of packets created in simulations (see Fig. 6.1(d)). ORW does not create any dummy packets. However, our protocols do to actively maintain the gradient. The probability of generating a dummy packet by a node when the gradient ages is set to 15%. Despite the extra overheads, we see that HEAR-DC outperforms ORW.

### 6.3.2 Low Energy Scenario

In this scenario, the nodes start with low energy and also harvest low energy levels or sometimes do not harvest at all, which makes it the worst case of evaluation. The trends in end-to-end latency and number of packets lost are similar to the high energy case for ORW and HEAR-DC, i.e., HEAR-DC outperforms ORW in most of the cases. This is shown in Fig. 6.3(a) and Fig. 6.3(b) respectively. Even when the energy is low, and the network is sparse, HEAR-DC achieves a latency reduction of 37% and packet loss reduction of 48% compared to ORW. Furthermore, when the network is dense, the gain in latency reduction is 35% while there is no benefit concerning

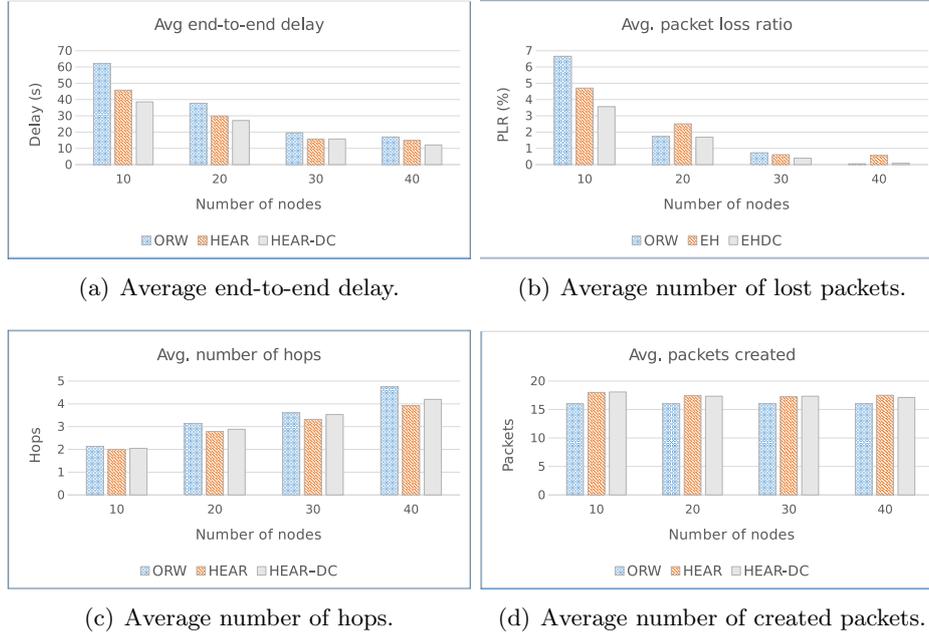


Figure 6.3: Performance comparison depending on number of nodes deployed under low energy environment.

packet losses.

Another observation is that HEAR performs better than ORW in most cases (comparable in the 20 node case as well). In the case of ORW, under the low energy regime, it may select nodes with higher link quality and closer to the sink (lower EDC), however, they may run out of energy leading to a bad performance. With HEAR and HEAR-DC, the energy level is given a high preference in route selection, leading to lower delays and lower packet losses. The performance of HEAR saturates at 30 nodes, and a higher density does not result in better results. This is due to the limited DC of the nodes.

The number of hops is more for ORW as shown in Fig. 6.3(c). This occurs due to the incorrect updates of EDC. Since many nodes die and come up again, EDC of the nodes is no longer valid, which gets propagated. Furthermore, even the nodes alive will not have an updated EDC due to an outage of many neighboring nodes. These make packets take incorrect deviations, leading to longer routes and higher packet losses.

The effects of our gradient together with adaptive DC can be seen in this scenario. Fig. 6.3.2 presents the number of node death occurrences in this scenario for various protocols. While this number is more or less the same for all protocols, the performance as shown in Fig. 6.3 is much better.

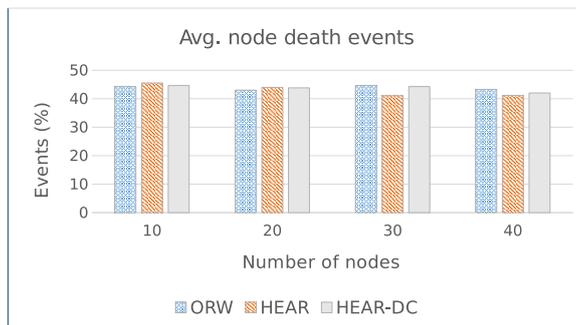


Figure 6.4: Node death occurrences under low energy.

### 6.3.3 Protocol Evaluation under Variable Traffic

In this section, we examine the influence of traffic load variations over the same fixed network, in this case, 20 nodes. We have five different packet generation patterns or intervals. First, we set all nodes to create packets periodically at various interval times. To avoid strict synchronization we randomized around the same mean the moment to generate the new data. Therefore, the resulting packet generation intervals fall around the following values:  $E[15s]$ ,  $E[45s]$ ,  $E[90s]$ ,  $E[180s]$  and finally we test the different protocols under event trigger packet generation. In this last scenario, the nodes will only generate packets when they experience a change in their harvesting rate beyond a certain threshold. In Fig. 6.5 we can observe the same four parameters as the previous experiments - delay 6.5(a), packets lost 6.5(b), number of hops 6.5(c) and number of packets created 6.5(d).

The end-to-end latency decreases as fewer packets are generated for all protocols. Similarly, the packet losses also reduce since the network experiences lower packet exchanges. We observe 17% lower end-to-end delay and 13% lower packet losses when the traffic is very high (15 s). However, as traffic decreases, the performance gap between ORW and HEAR-DC reduces significantly. This is mostly due to nodes having more energy buffered from harvesting. The most interesting case for evaluating our adaptation for sparse data (Sec. 4.6.1) is sending packets when an event trigger occurs. Due to our adaptation, we see that the adaptation enforces to result in much lower delay than ORW - all packets with HEAR-DC were delivered within 5 s, whereas ORW required a minimum of 19 s. Since the metric in ORW gets outdated, it begins to take some deviations as can be seen in Fig. 6.5(c).

### 6.3.4 Impact of Mobile Nodes

Lastly, we examine the impact of adding mobile nodes in the network. Mobile nodes change position throughout the simulation duration. We simulate mobile nodes with a Cooja plugin named *mobility*. This plugin allows the

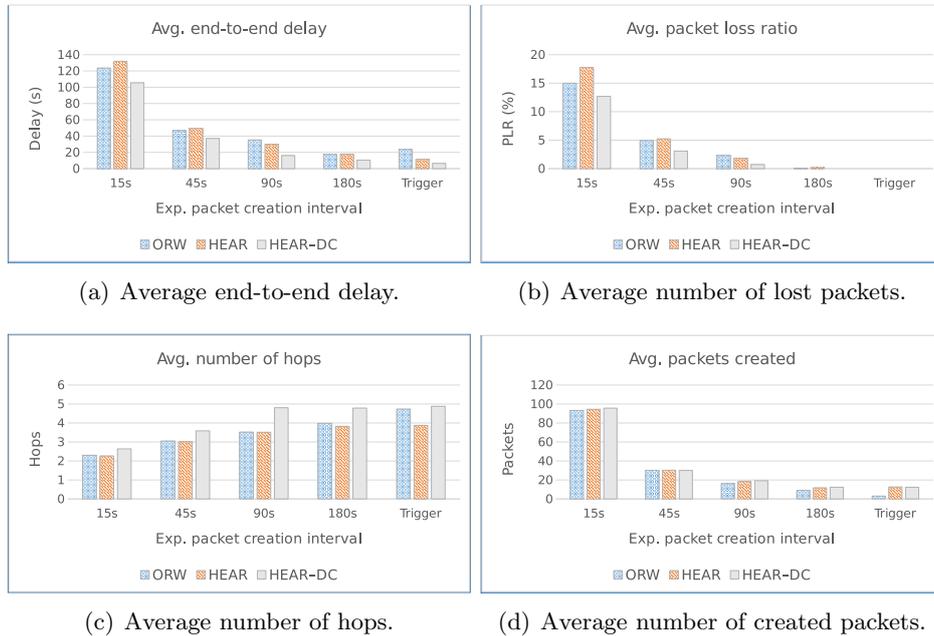


Figure 6.5: Performance comparison depending on traffic under average energy environment.

user to upload a path file that indicates at each simulation instance the position of each node. The outcome of using such plugin appears as the nodes ‘teleport’ from one location to another.

We designed three different test scenarios in which we vary the percentage of mobile nodes deployed in it. The network has a total of 30 nodes, with the number of mobile nodes being from 5, 10, and 15 nodes across the three scenarios.

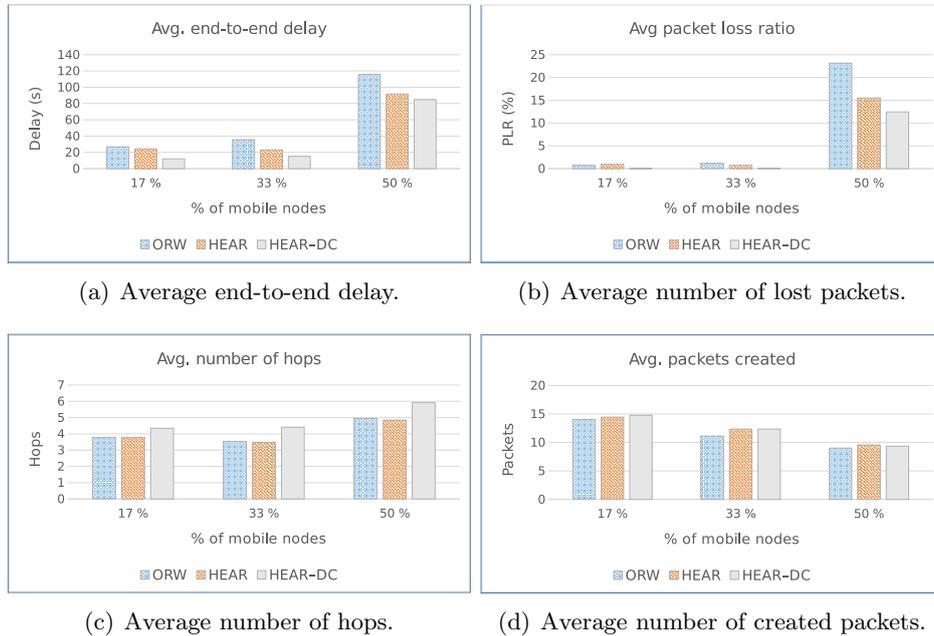


Figure 6.6: Performance comparison depending on mobile node percentage under average energy environment.

In Fig. 6.6 we see the results from this set of simulations. We observe a little difference between the protocols when the percentage of mobile nodes is 17%. A visible gain is seen in the case when 1/3 nodes are mobile - the end-to-end delays of HEAR and HEAR-DC are 45% and 53% better as compared to ORW. However, the delays slightly increase since the number of forwarding nodes reduces. This trend is more pronounced when half of the nodes are mobile. HEAR and HEAR-DC clearly outperform ORW in this case too, since ORW does not choose the routes that are more ‘reliable’.



## Chapter 7

# Conclusions and Future Work

### 7.1 Conclusions

With the proliferation of Internet of Things (IoTs), smart homes are just around the corner. Particularly, Active and Assisted Living (AAL) for people in their daily life application is of importance due their benefits. Two important requirements in the AAL scenario: (i) the ability to be available, i.e., be operational for extended periods of time; and (ii) scalable, i.e., be able to include more technologies (e.g., new on-body sensors) over time. We adopt a network architecture in which backbone nodes are statically deployed, and the mobile nodes connect to any possible emplaced node for data transfer. To make the nodes operate perpetually, we look to power all the nodes through energy-harvesting. However, it is challenging to collect data reliably with energy variations across the network. It is more so in indoor environments where the harvested energy levels are very small, typically in  $\mu\text{J}$ .

To this end, we proposed a novel routing protocol HEAR-DC which is made up of several components: (i) transmitter-initiated MAC with opportunistic reception and duplicate avoidance; (ii) an energy-harvesting aware gradient; (iii) adaptive duty cycling; and (iv) adaptations to support mobile nodes and sparse data traffic. The philosophy of this protocol is to opportunistic use of available energy and wireless channel for routing.

We developed a trace-based energy-harvesting simulation module for Cooja, which did not exist hitherto. We discussed the design, implementation, and limitations of this module. Using the same module, we evaluated our protocol against HEAR (HEAR-DC but with fixed duty cycle) and state-of-the-art opportunistic routing protocol, ORW. In the best case, i.e., when the energy levels are high, and the harvesting rates are high, we found with 10 nodes, HEAR-DC achieved a latency reduction of 64% with 92.75% lower

packet losses compared to ORW. With a higher density of nodes, performance gains were lower. In the worst case, i.e., when the energy levels are low, and the harvesting rates are low, with 10 nodes HEAR-DC achieved a substantial latency reduction of 37% and packet loss reduction of 48% compared to ORW. Due to the adaptation for sparse data traffic, the delay introduced by HEAR-DC is almost by a factor of four lower than ORW when only event-triggers were sent. Furthermore, with more than half of the network having mobile nodes that are only data sources, HEAR-DC achieved 31% lower delay and 46% lower packet loss with respect to ORW.

We observed that HEAR-DC outperforms both HEAR and ORW in all cases. HEAR outperforms ORW only in the cases when energy is around the critical mark with several nodes on the verge of outage or dead. This is clearly because ORW does not consider energy variations into account. We draw this and other inferences from the results from all scenarios.

1. ORW does not adapt to the energy conditions well but works as good as HEAR-DC when the density of the nodes is high enough. However, its performance does not significantly improve if the density is kept but energy harvesting opportunities are increased. The limitation is mainly due to the fixed 1% duty cycle. Adding adaptive DC to ORW will work well in the high energy scenario but also fail in the low energy scenario since there is little room for adaptive DC to perform. The inference is ORW, as is set by default, is not suited for EH-WSNs.
2. HEAR, on the other hand, performs badly when the energy input is high on the network since high energy nodes attract a lot of packets, but will fail to deliver them due to a fixed duty cycle of 1%. HEAR performs relatively better than ORW in the low energy case simply because the packets get queued in the high energy nodes, which have a high chance of reliable delivery. The conclusion is that energy harvesting aware gradient is important, but is not sufficient to offer good performance.
3. HEAR-DC outperforms all other protocols. Since HEAR-DC is a combination of our gradient and adaptive DC, we are interested to find out which one has more effect on the results. We see the performance of gradient only in HEAR, which works well when adaptive DC has little role to play. Only adaptive DC in this case would result in a bad performance. However, when the energy in the buffer is even slightly above the set-point, the adaptive DC makes a huge difference with respect to our metrics of delay and packet losses. In an average case scenario, both adaptive DC and gradient will work in synergy to offer better performance.
4. Most AAL networks will have low traffic and most probably an event-triggered sensing mechanism. As shown before, with ORW we will

have to endure a lot of delays due to aging, while our adaptation of dummy packets offers lower delay. The tradeoff between energy spent for dummy packet and delay for actual data packets is acceptable as the harvested energy, if unused, is useless.

5. While HEAR-DC works the best among the protocols, in real deployment a critical density of nodes must be available to have a user acceptable performance.

## 7.2 Future Work

There are several extensions that are needed to this work. The most critical one is to compare this work with ORiNoCo, which is a protocol adapted for EH-WSNs. Although we implemented it, we could not sort out all the bugs within the time-frame of the thesis.

The next step would then be to evaluate the protocols on a real testbed. While Cooja has the multipath radio model, it is only a simple model of the reality of harsh wireless network. Collisions and packet losses can affect the protocols significantly. Furthermore, this scenario would give us an opportunity to test if link-quality component of ORW will make a significant difference with respect to our gradient.

Several limitations and questions were raised in the context of our trace-based simulation module. These restrictions must be addressed to enable better evaluation of all EH-WSN protocols.



# Bibliography

- [1] *200 hours of accelerometer information recorded over 25 days from 5 participants. Columbia University.* URL: <http://crawdad.org/columbia/kinetic/20140513/>.
- [2] *AAL European Union.* URL: [http://www.aal-europe.eu/wp-content/uploads/2015/11/20151001-AAL-Strategy\\_Final.pdf](http://www.aal-europe.eu/wp-content/uploads/2015/11/20151001-AAL-Strategy_Final.pdf).
- [3] H. D. Akaydin, N. Elvin and Y. Andreopoulos. “Wake of a cylinder: A paradigm for energy harvesting with piezoelectric materials”. In: *Experiments in Fluids* 49.1 (2010), pp. 291–304. ISSN: 07234864. DOI: 10.1007/s00348-010-0871-7.
- [4] David Benedetti, Chiara Petrioli and Dora Spensa. “GreenCastalia: an energy-harvesting-enabled framework for the castalia simulator”. In: *Proceedings of the 1st International Workshop on Energy Neutral Sensing Systems*. ACM. 2013, p. 7.
- [5] Alessandro Bogliolo, Emanuele Lattanzi and Andrea Acquaviva. “Energetic sustainability of environmentally powered wireless sensor networks”. In: *Proceedings of the 3rd ACM international workshop on Performance evaluation of wireless ad hoc, sensor and ubiquitous networks - PE-WASUN '06* (2006), p. 149. DOI: 10.1145/1163610.1163636. URL: <http://portal.acm.org/citation.cfm?doid=1163610.1163636>.
- [6] Max Braun. “My Bathroom Mirror Is Smarter Than Yours”. In: (2016). URL: <https://medium.com/@maxbraun/my-bathroom-mirror-is-smarter-than-yours-94b21c6671ba#.gh6hzxanp>.
- [7] Renner Christian. “Perpetual Data Collection with Energy-Harvesting Sensor Networks”. In: *ACM Transactions on Sensor Networks* 11.1 (2014). ISSN: 15504859. DOI: 10.1145/2566675.
- [8] Mina Cong et al. *CRAWDAD dataset columbia/kinetic (v. 2014-05-13)*. Downloaded from <http://crawdad.org/columbia/kinetic/20140513/kinetic-energy.traceset:kinetic-energy>. May 2014. DOI: 10.15783/C73W29.
- [9] *Contiki Development at ANRG, University of Southern California. Contiki Netstack.* URL: [http://anrg.usc.edu/contiki/index.php/Network\\_Stack](http://anrg.usc.edu/contiki/index.php/Network_Stack).

- [10] *Cooperative Networks For Renewable Resource Measurements. Solar Energy Resource Data*. URL: [http://rredc.nrel.gov/solar/new\\_data/confrrm/](http://rredc.nrel.gov/solar/new_data/confrrm/).
- [11] *Dataset of radiant light energy measurements. Columbia University*. URL: <http://crawdad.org/columbia/enhants/20110407/>.
- [12] D. Dondi et al. “Modeling and Optimization of a Solar Energy Harvester System for Self-Powered Wireless Sensor Networks”. In: *IEEE Transactions on Industrial Electronics* 55.7 (2008), pp. 2759–2766. ISSN: 0278-0046. DOI: 10.1109/TIE.2008.924449.
- [13] Guangcheng Dong et al. “Thermal fatigue behaviour of Al<sub>2</sub>O<sub>3</sub>/Cu-DBC substrates under high temperature cyclic loading”. In: *Soldering & Surface Mount Technology* 22.November 2009 (2010), pp. 43–48. ISSN: 0954-0911. DOI: 10.1108/09540911011036280.
- [14] Euhanna Ghadimi and O Landsiedel. “Opportunistic Routing in Low Duty-Cycled Wireless Sensor Networks”. In: *ACM Transactions on Sensor Networks* (2014). DOI: 10.1145/0000000.0000000. URL: <https://people.kth.se/~%7B~%7Deuhanna/pub/oppTOSN.pdf>.
- [15] Omprakash Gnawali et al. “Collection tree protocol”. In: *Proceedings of the 7th ACM conference on embedded networked sensor systems*. ACM. 2009, pp. 1–14.
- [16] Maria Gorlatova et al. *CRAWDAD dataset columbia/enhants (v. 2011-04-07)*. Downloaded from <http://crawdad.org/columbia/enhants/20110407/energy.traceset:energy>. Apr. 2011. DOI: 10.15783/C77P47.
- [17] Maria Gorlatova et al. “Movers and Shakers: Kinetic Energy Harvesting for the Internet of Things”. In: (2013), p. 15. DOI: 10.1145/2591971.2591986. arXiv: 1307.0044. URL: <http://arxiv.org/abs/1307.0044>.
- [18] Maria Gorlatova et al. “Movers and shakers: Kinetic energy harvesting for the internet of things”. In: *ACM SIGMETRICS Performance Evaluation Review*. Vol. 42. 1. ACM. 2014, pp. 407–419.
- [19] David Hasenfratz et al. “Analysis, comparison, and optimization of routing protocols for energy harvesting wireless sensor networks”. In: *Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC), 2010 IEEE International Conference on*. IEEE. 2010, pp. 19–26.
- [20] Aman Kansal et al. “Power management in energy harvesting sensor networks”. In: *ACM Transactions on Embedded Computing Systems (TECS)* 6.4 (2007), p. 32.
- [21] K Kaushik et al. “Experimental Demonstration of Multi-Hop RF Energy Transfer”. In: *International Symposium on Personal, Indoor and Mobile Radio Communications* (2013), pp. 538–542.

- [22] Panqanamala Ramana Kumar and Pravin Varaiya. *Stochastic systems: Estimation, identification, and adaptive control*. Vol. 75. SIAM, 2015.
- [23] Emanuele Lattanzi et al. “Energetic sustainability of routing algorithms for energy-harvesting wireless sensor networks”. In: *Computer Communications* 30.14-15 (2007), pp. 2976–2986. ISSN: 01403664. DOI: 10.1016/j.comcom.2007.05.035.
- [24] Sanghyo Lee et al. “Trieboelectric energy harvester based on wearable textile platforms employing various surface morphologies”. In: *Nano Energy* 12 (2015), pp. 410–418.
- [25] Longbi Lin, Ness B. Shroff and R. Srikant. “Asymptotically optimal energy-aware routing for multihop wireless networks with renewable energy sources”. In: *IEEE/ACM Transactions on Networking* 15.5 (2007), pp. 1021–1034. ISSN: 10636692. DOI: 10.1109/TNET.2007.896173.
- [26] Paul D. Mitcheson et al. “Energy harvesting from human and machine motion for wireless electronic devices”. In: *Proceedings of the IEEE* 96.9 (2008), pp. 1457–1486. ISSN: 00189219. DOI: 10.1109/JPROC.2008.927494.
- [27] David Moss and Philip Levis. “BoX-MACs: Exploiting physical and link layer boundaries in low-power networking”. In: *Computer Systems Laboratory Stanford University* (2008), pp. 116–119.
- [28] Nikolaos A Pantazis, Stefanos A Nikolidakis and Dimitrios D Vergados. “Energy-efficient routing protocols in wireless sensor networks: A survey”. In: *IEEE Communications Surveys & Tutorials* 15.2 (2013), pp. 551–591.
- [29] R. Venkatesha Prasad et al. “Reincarnation in the Ambiance: Devices and networks with energy harvesting”. In: *IEEE Communications Surveys and Tutorials* 16.1 (2014), pp. 195–213. ISSN: 1553877X. DOI: 10.1109/SURV.2013.062613.00235.
- [30] Mohammed Rahimi et al. “Studying the feasibility of energy harvesting in a mobile sensor network”. In: *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on*. Vol. 1. IEEE, 2003, pp. 19–24.
- [31] Vijay S Rao et al. “On systems generating context triggers through energy harvesting”. In: *IEEE Communications Magazine* 52.6 (2014), pp. 70–77.
- [32] Silvia Santini and Kay Romer. “An adaptive strategy for quality-based data reduction in wireless sensor networks”. In: *Proceedings of the 3rd international conference on networked sensing systems (INSS 2006)*. 2006, pp. 29–36.

- [33] *SeedStudio [online]*. URL: <http://seedstudio.com>.
- [34] Nathan S Shenck and Joseph A Paradiso. “Energy scavenging with shoe-mounted piezoelectrics”. In: *IEEE micro* 21.3 (2001), pp. 30–42.
- [35] Tmote Sky. “Ultra low power IEEE 802.15. 4 compliant wireless sensor module”. In: *Moteiv Corporation* (2006).
- [36] *Tmote Sky datasheet*. URL: <http://www.eecs.harvard.edu/~konrad/projects/shimmer/references/tmote-sky-datasheet.pdf>.
- [37] Stefan Untersch, Christian Renner and Volker Turau. “Data-Collection Protocol”. In: (2012), pp. 1–16.
- [38] Christopher M Vigorito, Deepak Ganesan and Andrew G Barto. “Adaptive control of duty cycling in energy-harvesting wireless sensor networks”. In: *2007 4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*. IEEE. 2007, pp. 21–30.
- [39] Anthony Wood et al. “Alarm-net: Wireless sensor networks for assisted-living and residential monitoring”. In: *University of Virginia Computer Science Department Technical Report 2* (2006).
- [40] Jingjing Zhao and Zheng You. “A shoe-embedded piezoelectric energy harvester for wearable sensors”. In: *Sensors* 14.7 (2014), pp. 12497–12510.
- [41] Linfeng Zheng et al. “Novel methods for estimating lithium-ion battery state of energy and maximum available energy”. In: *Applied Energy* 178 (2016), pp. 1–8.