

Wake-up Radio-based Data Forwarding for Green Wireless Networks

Georgia Koutsandria^{a,*}, Valerio Di Valerio^a, Dora Spenza^a, Stefano Basagni^b,
Chiara Petrioli^a

^aDepartment of Computer Science, Sapienza University of Rome, Rome, Italy

^bInstitute for the Wireless Internet of Things, Northeastern University, Boston, MA, USA

Abstract

This paper presents G-WHARP, for Green Wake-up and HARvesting-based energy-Predictive forwarding, a wake-up radio-based forwarding strategy for wireless networks equipped with energy harvesting capabilities (*green wireless networks*). Following a learning-based approach, G-WHARP blends energy harvesting and wake-up radio technology to maximize energy efficiency and obtain superior network performance. Nodes autonomously decide on their forwarding availability based on a Markov Decision Process (MDP) that takes into account a variety of energy-related aspects, including energy currently available and that harvestable in the foreseeable future. Solution of the MDP is provided by a computationally light heuristic based on a simple threshold policy, thus obtaining further computational energy savings. The performance of G-WHARP is evaluated via GreenCastalia simulations, where we accurately model wake-up radios, harvestable energy, and the computational power needed to solve the MDP. Key network and system parameters are varied, including the source of harvestable energy, the network density, wake-up radio data rate and data traffic. We also compare the performance of G-WHARP to that of two state-of-the-art data forwarding strategies, namely GREENROUTES and CTP-WUR. Results show that G-WHARP limits energy expenditures while achieving low end-to-end latency and high packet delivery ratio. Particularly, it consumes up

*Corresponding author

Email address: koutsandria@di.uniroma1.it (Georgia Koutsandria)

to 34% and 59% less energy than CTP-WUR and GREENROUTES, respectively.

Keywords: Green wireless networks, wake-up radio, energy harvesting, routing, Markov Decision Process, Reinforcement Learning

1. Introduction

With 14.2 billions of connected *things* in 2019, over 41.6 billions expected by 2025, and a total spending on endpoints and services that will reach well over \$1.1 trillion by the end of 2026, the Internet of Things (IoT) is poised to
5 have a transformative impact on the way we live and on the way we work [1, 2, 3]. The vision of this “connected continuum” of objects and people, however, comes with a wide variety of challenges, especially for those IoT networks whose devices rely on some forms of depletable energy support. This has prompted research on hardware and software solutions aimed at decreasing the dependence
10 of devices from “pre-packaged” energy provision (e.g., batteries), leading to devices capable of harvesting energy from the environment, and to networks—often called *green wireless networks*—whose lifetime is virtually infinite.

Despite the promising advances of energy harvesting technologies, IoT devices are still doomed to run out of energy due to their inherent constraints on
15 resources such as storage, processing and communication, whose energy requirements often exceed what harvesting can provide. The communication circuitry of prevailing radio technology, especially, consumes relevant amount of energy even when in idle state, i.e., even when no transmissions or receptions occur. Even *duty cycling*, namely, operating with the radio in low energy consumption
20 (*sleep*) mode for pre-set amounts of time, has been shown to only mildly alleviate the problem of making IoT devices durable [4]. An effective answer to eliminate all possible forms of energy consumption that are not directly related to communication (e.g., idle listening) is provided by ultra low power radio triggering techniques, also known as *wake-up radios* [5, 6]. Wake-up radio-based
25 networks allow devices to remain in sleep mode by turning off their main radio when no communication is taking place. Devices continuously listen for a trig-

ger on their wake-up radio, namely, for a *wake-up sequence*, to activate their main radio and participate to communication tasks. Therefore, devices wake up and turn their main radio on only when data communication is requested
30 by a neighboring device. Further energy savings can be obtained by restricting the number of neighboring devices that wake up when triggered. This is obtained by allowing devices to wake up only when they receive specific wake-up sequences, which correspond to particular protocol requirements, including distance from the destination, current energy status, residual energy, etc. This
35 form of selective awakenings is called *semantic addressing* [7]. Use of low-power wake-up radio with semantic addressing has been shown to remarkably reduce the dominating energy costs of communication and idle listening of traditional radio networking [8, 9, 10, 7, 11, 12].

This paper contributes to the research on enabling *green wireless networks*
40 for long lasting IoT applications. Specifically, we introduce a wake-up radio and learning-based forwarding strategy aimed at minimizing energy consumption and end-to-end latency, while maximizing packet delivery ratio. Our solution, named G-WHARP for Green Wake-up and HARvesting-based energy-Predictive forwarding, takes full advantage of wake-up radio technologies and
45 harvestable energy to achieve great energy efficiency and corresponding superior performance. Forwarding availability decisions are driven by the energy capabilities of each device, taking into account consumed and available energy as well as the energy that can be predicted to be harvested in the near future. Key contributions of this paper include the following.

- 50 • We present G-WHARP, a data forwarding solution that smartly harnesses the joint benefits of energy harvesting and wake-up radios with semantic addressing to obtain superior network performance while minimizing the time devices are off for lack of energy. Devices determine their forwarding availability by running a Markov Decision Process (MDP) that
55 allows them to wake up based on their energy availability and on the ability to advance data closer to their destination. Wake-up semantic address-

ing is used to avoid waking up devices with no forwarding availability (as determined by the MDP), and hence eliminating the power consumption due to unnecessary idle time and communication.

- 60 • The MDP is solved at each device by running a computationally light heuristic based on a simple threshold policy. This further induces energy savings that are enough to offset the lack of optimality of the heuristic.
- We provide a detailed GreenCastalia-based [13] investigation of the performance of G-WHARP in several realistic scenarios. The performance of our protocol is first evaluated by varying key protocol and system parameters, including energy harvesting sources, network density, wake-up radio data rate and data traffic. We then compare the performance of G-WHARP to that of two state-of-the-art data forwarding solutions for green wireless networks with wake-up radio capabilities, namely, GREEN-65 ROUTES, an end-to-end energy-driven route selection and cross-layer data forwarding protocol [8], and CTP-WUR, a traditional tree-based routing solution [9]. Results show that G-WHARP clearly outperforms the other solutions. Particularly, G-WHARP consumes up to 34% and 59% less energy than CTP-WUR and GREENROUTES, respectively, allowing devices to remain operational for at least 93.7% of the simulated time. 70 With much more nodes that are up for longer time, G-WHARP obtains data packet latencies that are up to 42% and 66% lower than that of CTP-WUR and GREENROUTES, respectively. Packet delivery ratio is also positively affected: We observe that G-WHARP delivers up to 26% 75 more packets than CTP-WUR and 10% more packets of those delivered by GREENROUTES. 80

The remainder of the paper is organized as follows. In Section 2 we discuss state-of-the-art communication solutions for wake-up radio-based wireless sensor networks. Details on the scenarios considered, along with the description of G-WHARP, are presented in Section 3. Simulation results are presented 85

and discussed in Section 4, where we initially evaluate the performance of G-WHARP when varying general system parameters (Section 4.2) and we then present results by comparing G-WHARP to two state-of-the-art forwarding strategies (Section 4.3). Finally, Section 5 concludes the paper.

90 2. Related Work

Battery-related energy limitations of traditional wireless devices have always raised crucial performance concerns among researchers and practitioners in the field of wireless IoT and wireless sensor networks (WSNs). In time, this has led research towards self-sustainable networks, such as those whose devices are
95 powered by energy harvesters. The theoretically unlimited power supply from energy harvesting alleviates battery depletion, allowing network operations to last well beyond those of traditional WSNs. Over the past years, *green wireless networks* have attracted considerable interest, leading to the design of solutions at all layers of the networking stack [14, 15, 16, 17]. Despite devices can harvest
100 ambient energy and replenish their energy storage, they can still run out of energy, causing network disruption and performance degradation. Extra energy savings could be provided by wake-up radios, especially by those with semantic addressing capabilities, whose application to WSNs has prompted new design of both networking hardware and software [18, 19, 5, 20, 9, 10, 21, 22].

105 Despite the host of routing solutions proposed for energy harvesting-based WSNs and for WSNs with devices with wake-up radios, few works have been concerned with forwarding strategies reaping the benefits of the joint usage of these technologies. In the rest of this section we review these solutions, as they are the most pertinent to our work.

110 GREENROUTES is a cross-layer end-to-end energy-aware forwarding strategy [8]. Relays are chosen based on their distance from the sink and on the available residual energy along recently traveled routes. GREENROUTES implements semantic wake-up addressing to wake up only the most energy-capable devices that are also closer to the sink. It is a cross-layer protocol in that a

115 sender jointly solves the problems of relay selection and channel access by en-
gaging in an RTS/CTS handshake with its neighbors that have been woken up.
As such, packets may incur higher delivery latency because of the need of wait-
ing for the handshake to be completed before their transmission. In order to
reduce latency, GREENROUTES forwards data packets to a known and already
120 used relay, whose ID is cached for a predefined time.

GREEN-WUP is a routing protocol introduced by Petrioli et al. for net-
works with wake-up radios and energy harvesting capabilities [7]. Nodes take
advantage of semantic awakenings to wake up those neighbors with higher en-
ergy and that are (one hop) closer to the sink. Similarly to GREENROUTES,
125 awoken devices indicate their availability to forward packets through the trans-
mission of a Clear-To-Send (CTS) packet. Even though GREEN-WUP concerns
green wireless networks, it does not consider harvested energy in that semantic
addresses only encode hop count and the current energy level of a device.

A learning-based data forwarding solution for green wireless networks, called
130 WHARP for Wake-up and HARvesting-based energy-Predictive forwarding, has
been proposed in [23] by Basagni et al. WHARP is an MDP-aided forward-
ing strategy that features wake-up radios with semantic addressing. It is also
cross-layer in that it uses the RTC/CTS mechanism for channel access. As
in GREENROUTES both RTC/CTS are sent on the main radio. Nodes proac-
135 tively decide whether they will participate to the forwarder selection process
based on their residual energy, expected harvesting intake, and an estimation
of the energy consumption incurred for forwarding packets. Nodes in WHARP
solve the MDP through the Backward Value Iteration (BVI) method striving to
converge to energy optimal routes at the cost of non-negligible computational
140 energy expenditure [24].

The forwarding strategy proposed in this paper is “greener,” i.e., more energy
efficient and tailored to the needs of most WSN applications, of all these previous
solutions for wake-up radio-based green wireless networks. Using WHARP as a
“prototype,” G-WHARP jointly combines cross-layer benefits, energy-efficient
145 heuristics to solve complex learning-based machinery, optimized ID caching and

wake-up radio-enabled semantic addressing to produce superior performance and afford networks with remarkably longer lasting operations.

3. G-WHARP

This section provides the details of the G-WHARP forwarding strategy. We start by describing the green networking scenario considered in the paper (Sections 3.1). In Section 3.2 we present the MDP framework at the core of G-WHARP packet forwarding. Section 3.3 concludes the description of G-WHARP by providing details on packet forwarding operations.

3.1. Network scenario

We consider a multi-hop wireless network consisting of N devices (also called nodes) that are statically deployed in a given area of interest. Nodes are endowed with sensors that produce data according to the requirements of specific applications. They also wirelessly communicate with each other to deliver data to a network collector node, the *sink*, for processing and/or further forwarding. As the distance between a node and the sink might be exceeding their transmission range, data may have to follow multi-hop routes. All nodes but the sink harvest energy from the environment and store it in a supercapacitor. Each node is also equipped with two transceivers, namely, the *main radio* and the *wake-up radio*, operating at different frequencies. The wake-up transceiver is always on. The main radio, instead, can either be awake or asleep, which corresponds to very different levels of power consumption. The two transceivers have different sensitivities leading to ranges R_m and R_w for the main and wake-up radio, respectively. We assume that each node i knows its minimum hop distance from the sink, namely its *hop count* HC_i , computed with respect to the wake-up radio range.¹

¹ Hop count determination can be performed through a wake-up radio-based broadcast protocol initiated by the sink. The protocol is executed at the start of network operations to ensure that all nodes set up up their hop distance from the sink, and may be repeated periodically to keep the hop count updated to deal with temporary node outages. The FLOOD-WUP protocol in an example of this kind of broadcast protocols [7].

Prior to data forwarding (which always happens on the main radio), sender nodes use their wake-up radio to selectively awake specific nodes one of which will be selected as the forwarder. To this aim, each node in the network (but the sink, which is always up) chooses a set of wake-up addresses and wakes up
 175 only when its wake-up radio receives a sequence corresponding to one of the addresses in the set. A wake-up address, which is a sequence of m bits, can identify a specific node i , i.e., it represents its unique ID_i (statically assigned wake-up address), or it can be chosen to signify the node state, which depends on protocol-specific parameters, e.g., distance from the sink, current energy,
 180 memory availability, etc. (*semantic addresses*, assigned dynamically [7]). In G-WHARP each node i has always two addresses: Its ID_i and a dynamic address jointly encoding its hop distance from the sink HC_i and its availability to forward packets, which is a single bit information determined by executing learning-based algorithms (see below). The dynamic address has therefore the
 185 following format:

HC_i	Forwarding availability
$m - 1$ bits	1 bit

where the forwarding availability bit is set to either “*green*,” if the node is currently available to be selected as a forwarder, or to “*red*,” if it is not. Let
 190 us now consider a node i whose hop count HC_i is ℓ_i and that has a packet to forward. The G-WHARP routing strategy aims at waking up only those neighbors of node i that are one hop closer to the sink and that are energy-capable of packet forwarding. Consequently, node i will broadcast the following wake-up sequence:

$\ell_i - 1$	<i>green</i>
$m - 1$ bits	1 bit

Upon receiving this sequence, only nodes whose wake-up address matches the sequence broadcast by node i will wake up their main radio, and will participate

to the forwarder selection process. All other nodes, whether because they are not closer to the sink than node i or because they are not available to be forwarders (e.g., for lack of energy) will stay asleep. Finally, if and when node i needs to wake up a specific neighbor j with unique identifier ID_j , it transmits the m -bit wake-up sequence that represents ID_j .

3.2. An MDP-based framework for G-WHARP

In this section we define the formal components of the Markov Decision Process (MDP) that constitutes the pivotal step of G-WHARP data forwarding [25]. Particularly, we provide the mathematical machinery that allows each node to decide whether to make itself available to data forwarding, i.e., whether to wake up the main radio to be a potential forwarder for a data packet. Eventually, the output of the MDP-based computation is a simple flag, either *green* or *red*, that along with the node HC will determine its current dynamic wake-up address.

Notation. Nodes take forwarding availability decisions periodically, i.e., every *decision epoch* n , $0 \leq n \leq \hat{n}$, where \hat{n} is the maximum finite number of decision epochs. Once a decision is taken, it stays the same for the duration of the epoch. The set $\mathcal{B} = \{0, \dots, B_{max}\}$ contains discretized values of node energy. With b_n we indicate the amount of energy stored in the supercapacitor of a node at the start of decision epoch n . With h_n we denote the energy that is expected to be harvested in decision epoch n , determined with some form of energy predictors, e.g., AEWMA [26]. Concerning energy, we consider the following definitions (all related to the n th epoch). With e_n^s we indicate the energy consumed for sensing and for transmitting locally produced data. This value is estimated based on sensing and packet transmissions in the previous epoch. The value e_n^f denotes the energy spent to forward data packets from other nodes. We assume that e_n^f follows some specified probability distribution p^{ef} continuously estimated by each node during its operations. The expected available energy at the end of the n th epoch, when no data packets from other nodes have been forwarded, is $e_n = b_n + h_n - e_n^s$.

State space \mathcal{S} . In each epoch n , every node is in a state $s_n = b_n$ where $b_n \in \mathcal{B}$. Node i is in state $s_n = B_{max}$ if its supercapacitor is full, whereas a node with
 230 an empty supercapacitor is in state $s_n = 0$. In this case, the node shuts down (*all-off* state).

Action space \mathcal{A} . In any given state $s_n \in \mathcal{S}$ in epoch n a node can take one of two possible actions, indicating whether it is available to forward a data packet or not, i.e., indicating its current forwarding availability. Particularly,
 235 the action space \mathcal{A} is composed by the two actions a_g and a_r corresponding to positive forwarding availability (*green*) and to negative forwarding availability (*red*), respectively.

Transitions and transition probabilities. Transitions from state s_n to state s_{n+1} depend on the current state ($s_n \in \mathcal{S}$) and on the action ($a_n \in \mathcal{A}$) taken in the current decision epoch n . Formally:

$$s_{n+1} = \begin{cases} e_n & \text{if } a_n = a_r \wedge b_n + h_n > e_n^s \\ e_n - e_n^f & \text{if } a_n = a_g \wedge b_n + h_n > e_n^f + e_n^s \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

Particularly, if a node decided not to be available for forwarding ($a_n = a_r$) and it has enough energy to transmit its own data packets (i.e., $b_n + h_n > e_n^s$), then its
 240 next state s_{n+1} will be $e_n = b_n + h_n - e_n^s$. If it decided to be available ($a_n = a_g$) and if it has enough energy to transmit its own packets and to forward data packets from neighboring nodes, then its next state s_{n+1} is $e_n - e_n^f$. Finally, independently of the action taken in the n th epoch, if a node does not have enough energy to transmit/forward any data packet, then the next state will be
 245 *all-off* ($s_{n+1} = 0$).

Nodes transit from state s_n to state s_{n+1} according to a certain probability, after an action a_n has been taken. If a node at state s_n decides not to forward packets, i.e., $a_n = a_r$, then it transits to state s_{n+1} with probability $P_{s_n \rightarrow s_{n+1}}^{a_r} =$
 1. When a node chooses to forward packets, i.e., $a_n = a_g$, it transits to the next

state with the following probability:

$$P_{s_n \rightarrow s_{n+1}}^{a_g} = \begin{cases} p^{e_n^f}(e_n^f) & \text{if } b_{n+1} > 0 \\ \sum_{e_n^f=e_n}^{\infty} p^{e_n^f}(e_n^f) & \text{if } b_{n+1} = 0. \end{cases} \quad (2)$$

If a node is not *all-off* (i.e., $b_{n+1} > 0$), then the transition probability coincides with the probability $p^{e_n^f}(e_n^f)$ of consuming the energy $e_n^f < e_n$ for forwarding packets from other nodes. Otherwise, the transition probability corresponds to the probability of consuming energy for forwarding any number of packets that would exceed the available energy e_n .

Reward function. In G-WHARP nodes should be available to forward packets with the goal of maximizing their operational time, namely, maximize the time when they are not *all-off*. When nodes decide to forward packets, i.e., $a_n = a_g$, the MDP-based model in G-WHARP rewards them, while it penalizes them when they go *all-off*. More formally, when a node decides to forward packets:

$$r(s_n, a_g) = r \cdot \sum_{e_n^f=0}^{e_n} p^{e_n^f}(e_n^f) - c \cdot \sum_{e_n^f=e_n}^{\infty} p^{e_n^f}(e_n^f), \quad (3)$$

where r is the reward that a node gets if it does not run out of energy in the current decision epoch, and c is the penalty cost if it goes *all-off*. Reward and penalty costs are weighted by the probability that the energy consumption of forwarding packets is lower and higher than the available energy in a decision epoch, respectively. If a node decides not to forward packets, i.e., $a_n = a_r$, then its reward $r(s_n, a_r)$ becomes 0. Nodes that run out of energy while transmitting their own data do not get a penalty.

Deciding forwarding availability. In order for a node to evaluate how good it would be as a forwarder in a given state s_n , MDP theory states that a value function should be defined that depends on all the “ingredients” that we have defined so far, namely, s_n and a_n and the expected reward $r(s_n, a_n)$ associated to them. For each state, the optimal decision about forwarding availability, namely, the

policy that maximizes the value function, satisfies a Bellman optimality equation. Solving this equation for determining such an optimal policy can be done using a host of methods, such as Backward Value Iteration (BVI) [25]. However, these solution methods have a non-negligible computational cost, which can be as impactful on network performance as communication operations. This suggests to resort to computationally simpler heuristic solutions, which trade off optimality with energy savings.² Algorithm COMPUTEACTION summarizes the

Algorithm COMPUTEACTION(s_n, h_n)

```

1:  $e_n = b_n + h_n - e_n^s$  #Compute the available energy for packet forwarding
2:  $r(s_n, a_g) = r \cdot \sum_{e_n^f=0}^{e_n} p^{e_n^f}(e_n^f) - c \cdot \sum_{e_n^f=e_n}^{\infty} p^{e_n^f}(e_n^f)$  #Compute the reward function
3: if  $r(s_n, a_g) > 0$  then #Threshold policy
4:    $Avail_i = green$  #Positive forwarding availability
5: else
6:    $Avail_i = red$  #Negative forwarding availability

```

operations executed by node i running G-WHARP for computing its forwarding availability using a heuristic approach. The algorithm is performed at the beginning of each decision epoch n , $0 \leq n \leq \hat{n}$, when in state s_n and with energy harvesting prediction h_n . The node starts by computing the expected energy e_n using the expected harvested energy, its current energy and the energy needed for sensing and transmitting its own packets (line 1). The reward function is then computed as defined by Equation (3) (line 2). Forwarding availability is then decided according to a simple *threshold policy*, for which if the computed reward is strictly positive the node will be available to forward (*green*), and not (*red*) otherwise. The availability of node i for decision epoch n is stored in the global variable $Avail_i$ (lines 3 to 6). (Details on the rationale and proofs of the heuristic solution used in this work can be found in Appendix A.)

It is worth noting that all information needed for a node to periodically run

² In fact, we were able to demonstrate that solving the MDP heuristically saves so much computational energy that the loss in optimality is well compensated by the energy savings. The computational cost of both solutions was measured using real hardware, i.e., the MagoNode++ mote. Results show that running the heuristic solutions consume approximately 7.3 times less energy than the BVI solution.

Algorithm COMPUTE ACTION are local to the node itself. As each node runs its own energy predictor (depending on the energy source) and knows its energy consumption, G-WHARP does not incur any extra communication overhead to obtain the information required for determining its forwarding availability. For the same reason, being this information readily available to the node at the beginning of every decision epoch, there is no “convergence lag,” a problem that instead affects many reinforcement learning-based solutions for wireless networks [25].

3.3. Packet handling in G-WHARP

When a node i with hop count $HC_i > 1$ has a data packet p to transmit, it selects the best forwarder among its neighboring nodes by executing Algorithm FORWARDPACKET(p, HC_i).³ Node i broadcasts a wake-up sequence w_i

Algorithm FORWARDPACKET(p, HC_i)

- 1: BroadcastWakeUpSequence($w_i = \langle HC_i - 1 | \text{green} \rangle$) *#Broadcast wake-up sequence w_i*
 - 2: MainRadio(ON) *#Turn the main radio on*
 - 3: **wait for** GREEN packet from node j *#Wait for GREEN packets*
 - 4: TransmitPacket(p, j) *#Transmit packet p to node j*
 - 5: **wait for** ACK packet from node j *#Wait for an ACK packet*
 - 6: MainRadio(OFF) *#Turn the main radio off*
-

aimed at waking up its neighboring nodes with hop count $HC_i - 1$ and positive forwarding availability (line 1). After the transmission of w_i , node i turns its main radio on (line 2). It then awaits to receive GREEN (control) packets from the neighboring nodes it just woke up. The node j whose GREEN packet is received first is selected as the forwarder of packet p (line 3). All GREEN packets received subsequently are discarded. After p is transmitted to node j (line 4), node i awaits to receive an acknowledgment packet ACK from node j (line 5), after which it turns its main radio back off (line 6). The implementation of the

³ If its hop count $HC_i = 1$ node i sends the packet directly to the sink.

wait for statements at lines 3 and 5 include failsafe procedures based on timers as suitable thresholds, as detailed below.

305 Upon receiving a wake-up sequence w_i from node i , each neighboring node j executes Algorithm CANDIDATEFORWARDER(w_i). Node j starts by checking

Algorithm CANDIDATEFORWARDER(w_i)

```

1: if ( $w_i \downarrow_{HC} == HC_j$ ) && ( $Avail_j == green$ ) then
2:   Compute  $\delta_j$  and wait for  $\delta_j$  time units           #Compute and wait a delay  $\delta_j$ 
3:   MainRadio(ON)                                       #Turn the main radio on
4:   BroadcastPacket(GREEN)                             #Broadcast a GREEN packet
5:   wait for packet  $p$  from node  $i$                    #Wait for data packet  $p$ 
6:   TransmitPacket(ACK,  $i$ )                          #Send an ACK to node  $i$ 
7:   MainRadio(OFF)                                     #Turn main radio OFF

```

whether it could provide a positive advancement of a data packet toward the sink, i.e., if the hop count component $w_i \downarrow_{HC}$ of the wake-up sequence w_i just received equals HC_j . It also checks whether in this decision epoch it is available
310 for forwarding data packets (line 1). In the negative, node j keeps its main radio off and does not participate to the forwarder selection process. Otherwise, it computes a delay δ_j and awaits δ_j time units before turning its main radio on (lines 2 and 3).⁴ At this point it broadcasts a GREEN packet indicating that it is available as a forwarder for a data packet (line 4). A GREEN packet
315 carries information about the identity of node j so that, in case node i selects node j as the forwarder, the data packet can be sent directly to it. The function of the delay δ_j is twofold. On one side, as each node j that has woken up will compute a different δ_j with high probability, it aims at decreasing the possibility of collisions of multiple GREEN packets at node i . On the other, it is used
320 to provide node i with an indication of how suitable node j is to effectively forward packets towards the sink: The better a node j is to be a forwarder, the shorter its δ_j . (Details on the computation of δ_j are provided later in Section 3.4.) Once node j has sent the GREEN packet, it awaits to receive the packet to be

⁴ Each node handles only one forwarding request at a time. Particularly, a node that has been woken up ignores new wake-up sequences until it is done with the current forwarding.

forwarded (line 5). If packet p is received, node j transmits an acknowledgment
 325 packet ACK to node i and goes back to sleep (lines 6 and 7). It will then start the
 forwarder selection process again by executing Algorithm FORWARDPACKET(p ,
 HC $_j$). Otherwise, if packet p is sent to some other node $k \neq j$ or it is not
 received within a certain time, node j goes back to sleep, i.e., it turns its main
 radio off.

330 *An example of packet forwarding in G-WHARP.* We conclude the description
 of G-WHARP with an example of its packet forwarding mechanism. Fig. 1
 depicts a scenario where node i with hop count HC $_i = \ell > 1$ has a data packet
 to transmit and nodes j_1, j_2, j_3 and j_4 are within its wake-up radio range.
 Nodes j_1, j_2 and j_4 all have hop count HC $_{j_1} = \text{HC}_{j_2} = \text{HC}_{j_4} = \ell - 1$. Node j_3
 335 has instead hop count HC $_{j_3} = \ell + 1$.

By executing Algorithm FORWARDPACKET(p, HC_i) node i broadcast the
 wake-up sequence $w_i = \langle \text{HC}_i - 1 | \text{green} \rangle$ to its four neighbors. Having *red* for-
 warding availability node j_2 will stay asleep and will not participate to the
 forwarder selection process. Despite its positive availability, node j_3 will also
 340 stay asleep, as it would move the packet farther away from the sink (HC $_{j_3} >$
 HC $_i$). Only nodes j_1 and j_4 meet the conditions to participate in the forwarder
 selection process. Upon deciding to wake up they compute the delays δ_{j_1} and δ_{j_4} ,
 respectively. Once their delay has passed, each of them activates its main radio,
 broadcasts a GREEN packet and starts a data packet waiting timer. Node i
 345 transmits the DATA packet p to the node whose GREEN packet was received
 first, i.e., node j_1 . After reception of packet p , node j_1 acknowledges it and
 turns off its main radio. It will soon start a forwarder selection process itself, by
 executing Algorithm FORWARDPACKET(p, HC_{j_1}) to keep forwarding packet p .
 Node i goes back to sleep after receiving the ACK from node j_1 . The subsequent
 350 GREEN packet from node j_4 is ignored. As node j_4 does not receive a DATA
 packet addressed to it, it goes back to sleep after a set waiting time.

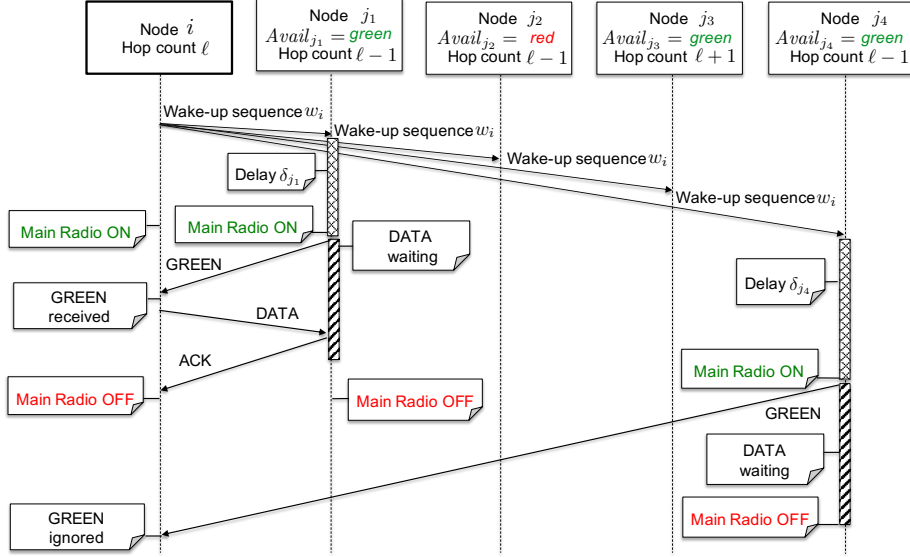


Figure 1: G-WHARP forwarding: An example.

3.4. Failsafe procedures and optimization

Timers and thresholds. For the sake of clarity, the description of algorithms FORWARDPACKET and CANDIDATEFORWARDER omits the details of failsafe procedures that are intended to take care of all those cases when a node expects to receive ℓ packet and instead it does not. Although not critical for the understanding of the operations of G-WHARP, these procedures need to be implemented, and parameters need to be chosen and tuned appropriately, which we did for the experimental evaluation of our protocol (Section 4). Particularly, in order to implement the **wait for** statement of line 3 of Algorithm FORWARDPACKET senders need to set a timer to a time t_g by which they expect to receive a GREEN packet. If a sender does not receive a GREEN packet, it repeats the process of seeking for a forwarder for a maximum of $K > 0$ times, after which it drops the packet.

For implementing the **wait for** statement of line 5 of Algorithm FORWARDPACKET a sender needs to set a timer to a time t_{ack} by which it expects to receive a packet ACK acknowledging the correct reception of packet p . If the acknowledgment is not received by t_{ack} , the sender selects a backoff time, after

which it sends packet p again to the selected forwarder for a maximum of $L > 0$
 370 times. If after L attempts the packet has not been acknowledged, the sender
 goes back to select a new forwarder.

Finally, for implementing the **wait for** statement of line 5 of Algorithm CAN-
 DIDATEFORWARDER a receiver needs to set a timer to a time t_p by which it
 expects to receive a data packet p . If a packet is not received by this time, the
 375 receiver turns its main radio off.

Calculation of the delay δ . Whenever node i sends a wake-up sequence, each
 neighboring node j that has elected to participate to the forwarder selection
 process replies with a GREEN packet after a delay δ_j computed as follows:

$$\delta_j = \left(1 - \frac{b_j}{B_{max}}\right) \cdot \delta_{max} + \delta_{rand}, \quad (4)$$

where b_j is node j currently available energy, δ_{max} is the maximum possible
 delay, and $\delta_{rand} < \delta_{max}$ is an extra small random delay used to avoid collisions of
 GREEN packets at the sender. As mentioned, the delay δ_j has the twofold aim
 of decreasing the possibility of collisions of multiple GREEN packets at node i
 380 and favors the choice of the best forwarder, namely, of the node with the highest
 energy. In other words, the higher the energy at a node, the lower its delay in
 replying to the sender, and therefore the higher its chances to be selected as a
 forwarder.

Caching IDs for optimized performance. The forwarder selection process
 385 can be time and energy consuming because of the multiple GREEN packet
 transmissions needed to find a forwarder j . In order to decrease delay and energy
 consumption, in the actual implementation of Algorithm FORWARDPACKET we
 let node i *cache* the ID of its last successful forwarder j for a predefined amount
 of time. All packets that node i needs to transmit within this time will be
 390 transmitted to node j directly, without any new forwarder selection process.
 All that node i has to do in this case is to wake node j up by using its ID_j as
 wake-up sequence and then transmit the packet on the main radio. If the data
 packet sent to node j is not successfully received, node i will re-transmit the

wake-up sequence directly to node j for at most L times. If all retransmission
395 attempts fail, node i falls back to selecting a new forwarder as described above.

4. Performance evaluation

We evaluate the performance of G-WHARP by varying general system parameters (Section 4.2) and by comparing it to the performance of two state-of-the-art forwarding strategies for wake-up radio-based green wireless networks, namely, GREENROUTES and CTP-WUR (Section 4.3). All protocols have been
400 implemented in the open-source GreenCastalia simulator [13], an extension of the OMNeT++ based Castalia simulator [27], we have developed to model green wireless networks in details. We further extended GreenCastalia to model a real wake-up radio-based system. Our extension mimics the behavior of the
405 MagoNode++ mote that supports an ultra-low-power receiver and a wake-up transmitter capable of sending wake-up sequences with semantic addressing [28]. Two different versions of the wake-up system have been designed, both using the On-Off Keying (OOK) modulation and each optimized to work at a different transmission frequency, i.e., 868MHz and 433MHz. In this section, we provide
410 results for the former type of wake-up radio. The wake-up transmitter supports different data rates to transmit wake-up sequences, i.e., $\{1, 5, 10\}$ Kbps. Our performance evaluation considers all of the three data rates supported by the MagoNode++ mote. The resource manager module of GreenCastalia is also extended to take into account the energy consumed for executing the heuristic
415 that solves the MDP used by G-WHARP (Section 3.2 and [24]). The computational energy consumption value used in this work is the outcome of real measurements using the MagoNode++ mote. In particular, we implemented both MDP solutions, namely, the heuristics and Backward Value Iteration (BVI), in TinyOS, the operating system used by the MagoNode++. Our measurements
420 confirm the lighter computational requirements of the heuristic solution, which consumes 7.3 times less energy than the BVI-based solution [24].

4.1. Simulation scenario and parameter settings

For all our experiments we consider green wireless networks with a number N of sensor nodes varying in the set $\{48, 64, 128\}$. Nodes are randomly and uniformly placed in a rectangular area of size $224 \times 56\text{m}^2$. This gives rise to networks with different densities: Sparse ($N = 48$), medium ($N = 64$), and dense ($N = 128$). The sink node is located at the lower left corner of the deployment area. Nodes perform sensing measurements generating data of 38B (data packet payload) at an inter-arrival rate from $\{5, 4, 3, 2, 1, 0.75, 0.5\}$ seconds. Payload size and data generation rates are consistent with those of several applications for green networking. The total packet size is 58B, which includes additional bytes of headers at various layers. GREEN (control) packets are 6B long. Data and control packets are transmitted on the main radio at a rate of 250Kbps, with a maximum transmission range of 60m. Wake-up sequences are 8 bit long. They are transmitted by the wake-up radio at a rate varying in the set $\{1, 5, 10\}$ Kbps. The transmission range of the wake-up radio is set to 25m, in agreement with the ranging measurements of our wake-up radio prototype [29]. (According to these measurements, at this distance, the waking up probability is higher than 90%.) Both radios implement the additive interference model to determine simultaneous transmissions from multiple nodes. All nodes but the sink are equipped with energy harvesters. They draw energy from the environment by using either solar cells (*solar nodes* in the following) or micro wind turbines (*wind nodes*). We consider real harvesting traces obtained from the National Renewable Laboratory at Oak Ridge [30] and collected in Rome, Italy, for one month during summer (solar and wind traces, respectively). Fig. 2 shows the harvesting power of the real traces used in our performance evaluation. Differences on the harvested power among the days considered are due to the weather variations, i.e., sunny vs. cloudy days. We observe that, on average, wind nodes harvest 6 times less energy than solar nodes. The harvested energy is stored in a supercapacitor with maximum operating voltage of 2.3V and capacitance of 50F. Supercapacitors are initially fully charged and their cutoff value, i.e., minimum operating voltage, is set to 1.8V according to the specifications of the

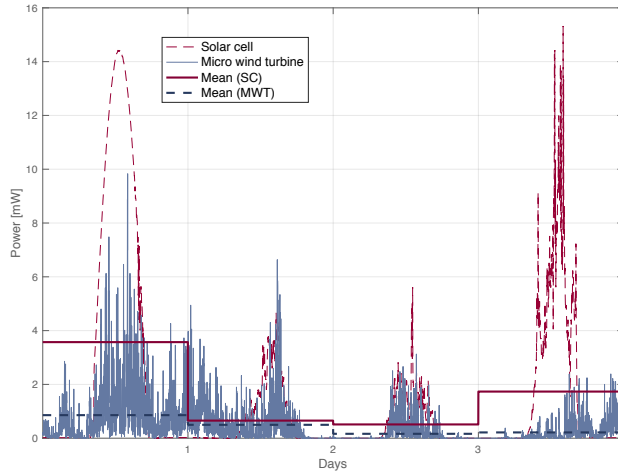


Figure 2: Sample of harvested power: Solar vs. wind.

MagoNode++ mote. The caching expiration time was set to 220s, as a result of several experiments with different values. The energy model considered in our scenarios is based on the MagoNode++ mote. All simulation parameters used
455 in our experiments are summarized in Table 1, including the power consumption of the four main components of the MagoNode++ mote, namely, its main radio, its wake-up radio, the micro-controller (MCU) in charge of executing the heuristic to solve the MDP, and the sensory component. Values for the MCU
460 are based on real measurements on the MagoNode++ mote. The total duration of the simulation was set to 4 days. Results are collected during the last two days of simulated time, as the first two days are required to reach steady-state performance.⁵ All results have been obtained by averaging the outcomes of a number of simulation runs that achieves a 95% confidence with 5% precision.

4.2. Challenging G-WHARP by varying system parameters

In this first set of experiments we investigate the performance of G-WHARP when varying four key system settings: 1) The energy type that nodes can harvest; 2) the network density; 3) the data rate of the wake-up radio, and 4)

⁵ This long transient time is required mostly by the energy predictor for providing accurate energy harvesting forecasting.

Table 1: Simulation parameters.

Notation	Definition	Value
General Parameters		
T_s	Simulation duration	4 days
-	Deployment area (m ²)	224 × 56
N	Network size (# of nodes)	48, 64, 128
$iaTime$	Data packet inter-arrival time	0.5, 0.75, 1, 2, 3, 4, 5s
-	Data packet payload size	38B
-	Data packet total size	58B
-	GREEN packet size	6B
-	Energy harvesting source	wind, heterogeneous, solar
-	Capacitance of supercapacitor	50F
-	Supercapacitor max operating voltage	2.3V
-	Supercapacitor cutoff voltage	1.8V
R_m	Main radio range	60m
-	Main radio data rate	250Kbps
R_w	Wake-up radio range	25m
-	Wake-up radio data rate	1, 5, 10 Kbps
G-WHARP Parameters		
t_g	GREEN packets waiting timer	45ms
t_p	DATA packet waiting timer	48.9ms
t_{ack}	ACK packet waiting timer	8.5ms
T_c	Expiration of cached forwarders	220s
δ_{max}	Maximum GREEN delay	35ms
δ_{rand}	Extra random GREEN delay	[0, 10ms]
-	Decision epoch length	720s
K	Max data packet retransmissions	10
L	Max data packet retransmissions using caching	2
-	Energy predictor	AEWMA [26]
Power consumption specifics		
Component	State	Value
Main radio	Tx (-2dBm)	31.2mW
	Rx	33.6mW
Wake-up radio	WUR Tx (10dBm)	90mW
	WUR Rx	1.071 μ W
MCU	Idle	0.036 μ W
	Active	54 μ W
Sensor	Active	3mW

the data packet inter-arrival time. We analyze and discuss the impact of each of
470 the varied system parameters on the performance of G-WHARP with respect
to the following key metrics: i) The average time a node spends transmitting
wake-up sequences (WUR Tx); ii) the total energy consumption spent by the
network, iii) the protocol packet delivery ratio, and iv) the end-to-end latency.
In each experiment, we vary only one of the system parameters at a time, setting
475 the others at fixed values.

1. *G-WHARP vs. energy harvesting source.* The performance of G-WHARP
for varying energy harvesting source is depicted in Fig. 3. We consider three
kinds of networks where either all nodes are equipped with solar cells, or they
are all using micro wind turbines, or half of the nodes are equipped with solar
480 cells and the remaining nodes harvest energy using micro wind turbines (hetero-
geneous harvesting scenario). The network density is set to medium (64 nodes),
the wake-up radio data rate to 5Kbps, and data average inter-arrival time to

1s. As a general trend, networks with solar nodes always achieve better performance than networks with wind nodes. The higher energy intakes of solar nodes affect their operations positively, affording them better performance. The time a node spends on transmitting wake-up sequences is always below 0.025% of the simulation time, as shown in Fig. 3a.

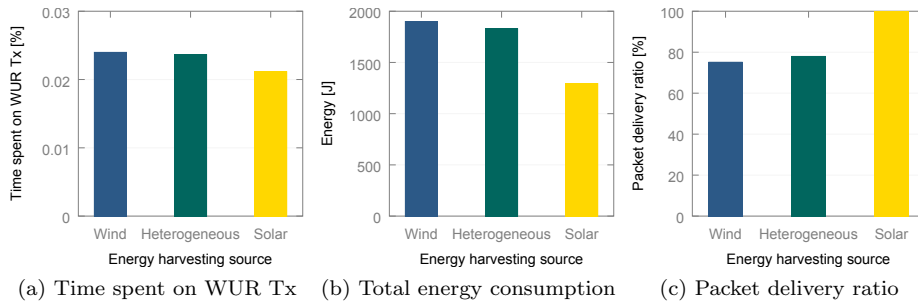


Figure 3: Performance analysis of G-WHARP for varying energy harvesting sources.

In general, the more energy a node harvests, the less it goes *all-off*. As a result, the probability of finding a next-hop forwarder decreases with increasing *all-off* time, which results to more re-transmissions of wake-up sequences and a lower packet delivery ratio. In addition, G-WHARP uses a caching mechanism to directly forward the data packet to a known (cached) next-hop forwarder (Section 3.4). However, when nodes go *all-off* more frequently, the chances of having a cached node that is *all-off* increase. Particularly, wind nodes go *all-off* approximately 1.5 times more than nodes in heterogeneous-source networks, 2% and 1.38% of the time, respectively, while solar nodes remain operational 100% of the time. This allows solar nodes to spend 14% less time on transmitting wake-up sequences than wind nodes due to the lower number of re-transmissions. As expected, when half of the nodes are wind nodes we observe that the network performs slightly better than networks with only wind nodes. While the presence of solar nodes benefits network performance, we observe minor improvements with respect to networks with only wind nodes. This is due to the low energy intake of wind nodes which impacts the time nodes go *all-off* weakening the benefit of using solar nodes. Networks with only wind nodes consume more

505 energy to successfully deliver packets to the sink than solar nodes, as the number of re-transmissions increases (Fig. 3b). Fig. 3c shows that networks with solar nodes deliver approximately 22% and 24.8% more packets to the sink than networks with heterogeneous and wind energy harvesting sources, respectively, with similar end-to-end latencies (always below 0.05 sec).

510 2. *G-WHARP vs. network density.* Fig. 4 depicts the performance of G-WHARP when network density varies. Results are depicted for heterogeneous

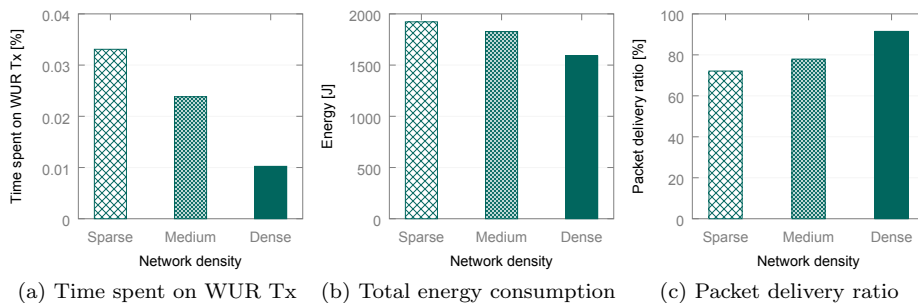


Figure 4: Performance analysis of G-WHARP for varying network density.

harvesting. All other parameters stay as in the previous experiment. As expected, the time a node spends transmitting wake-up sequences decreases with increasing density: The denser the network, the higher the number of nodes in the wake-up range of a node. Particularly, nodes in sparse networks transmit wake-up sequences for an average of 1.4 and 3.2 times longer than nodes in medium and dense networks, respectively (Fig. 4a). The total energy consumption decreases with increasing network density: As the number of nodes in the network increases, the number of data packets that a node processes decreases. When nodes in denser networks have to process fewer packets, they deal with fewer packet retransmissions and they activate their main radio less frequently, which results in lower energy consumption (Fig. 4b). For instance, sparse networks spend approximately 5% and 21% more energy than medium and dense networks, respectively. As a result, this also impacts the end-to-end latency allowing nodes in denser networks to deliver data packets to the sink 2.26% and 4.62% faster than medium and sparse networks, respectively. Finally, the packet

525

delivery ratio of a dense network is higher than that observed in medium and sparse networks with 15% and 21% more packets delivered to the sink, respectively (Fig. 4c). This, again, is because of the higher number of nodes that are available as forwarders in dense networks.

3. *G-WHARP vs. wake-up radio data rate.* Results concerning the performance of G-WHARP when varying the wake-up radio data rate are depicted in Fig. 5 (unless otherwise specified the rest of parameters remain as in previous scenarios). Nodes use their wake-up radio transmitter for longer time at lower data rates (Fig. 5a). At the lowest data rate, nodes spend approximately 5 and 12 times the time than when transmitting at 5 and 10 Kbps, respectively. This is because of the longer bit duration at lower data rates. As shown in Fig. 5b, a lower data rate implies higher energy consumption. The causes of this are multifold. First and foremost, the more a node uses its wake-up transmitter, the more energy it spends. Secondly, because of the higher energy consumption at lower data rates, nodes go *all-off* more frequently. For instance, a node

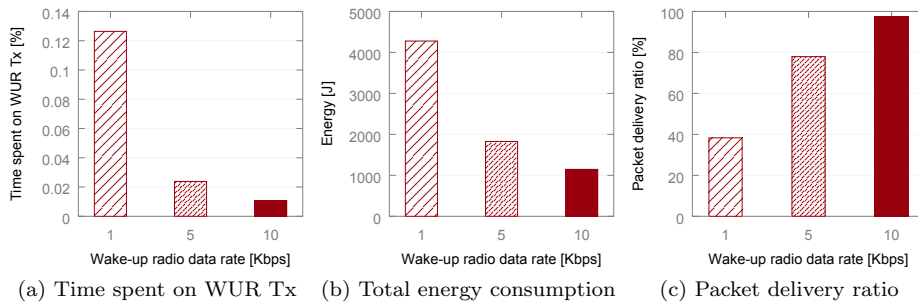


Figure 5: Performance analysis of G-WHARP for varying wake-up radio data rate.

transmitting at 1Kbps goes *all-off* 10 times more than when transmitting at 5 Kbps, while nodes transmitting at the highest data rate remain operational for approximately 99.9% of the time. This penalizes nodes transmitting at lower rates leading to a higher number of wake-up sequences re-transmissions for finding a forwarder. In fact, the energy consumption at the highest rate is significantly less than the energy spent when transmitting at medium and low data rates. In particular, transmitting at the highest wake-up data rate, i.e., 10

Kbps, requires an average of 2.3 and 4 times less energy than when transmitting
 550 at 5 and 1Kbps, respectively. As expected, the highest the wake-up data rate,
 the lower the end-to-end latency. Due to the longer time a node occupies the
 channel when transmitting at lower wake-up data rates which also results in a
 higher number of wake-up sequences re-transmissions, the end-to-end latency at
 10 Kbps is approximately 2.75 and 1.14 times lower than that at 1 and 5 Kbps,
 555 respectively. Not surprisingly, because of the higher operational time and hence
 the lower energy consumption, the higher packet delivery ratio at the highest
 data rate is particularly noticeable (Fig. 5c). We observe that at the highest
 data rate, the packet delivery ratio is 25% and 154% higher than at medium
 and low data rate, respectively.

560 4. *G-WHARP vs. packet inter-arrival time.* Fig. 6 shows the performance of
 G-WHARP for varying data packet inter-arrival time. We observe that at the
 highest traffic nodes are *all-off* for up to 10% more time than at lower traffic.
 Even caching the ID of a known good forwarder helps little, as the probability of
 having a cached node that is *all-off* increases. Our results confirm this intuition.
 565 In fact, for increasing traffic: Nodes spend more time transmitting wake-up
 sequences (Fig. 6a); the network experiences higher energy consumption (Fig. 6b),
 and the packet delivery ratio decreases (Fig. 6c). We notice however, that G-

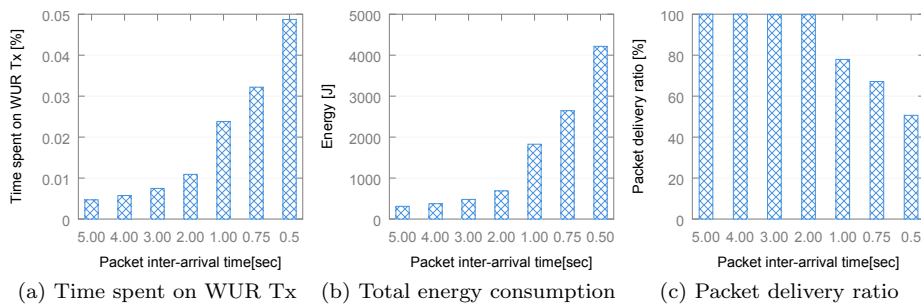


Figure 6: Performance analysis of G-WHARP for varying packet inter-arrival time.

WHARP successfully delivers all packets for inter-arrival times higher than
 1sec. On the contrary, the advantage of the caching mechanism with respect to
 570 the end-to-end latency becomes more prominent with increasing traffic. This

is because nodes eliminate the delays of the channel access by transmitting to a known forwarded more frequently with increasing traffic which compensates the higher percentage of *all-off* nodes. As an example, nodes at the highest traffic successfully deliver packets to the sink 1.41 times faster with respect to
575 the lowest traffic case.

4.3. Comparative performance evaluation

For the comparison of the performance of G-WHARP, GREENROUTES and CTP-WUR we consider the general scenario of green wireless networks of medium density with heterogeneous energy harvesting sources. We start
580 by providing a brief summary of the essential working of GREENROUTES and CTP-WUR. We then proceed to describe the scenario settings, the investigated metrics and the performance results.

4.3.1. Benchmark forwarding strategies

G-WHARP is compared to two state-of-the-art forwarding strategies for
585 green networks, namely, GREENROUTES and CTP-WUR, each being a paradigm of a different forwarding design choice. Particularly, GREENROUTES represents end-to-end energy-driven route selection protocols and CTP-WUR follows the paradigm of traditional tree-based routing. A brief description of GREENROUTES is provided in Section 2. In this section we briefly describe CTP-
590 WUR.

CTP-WUR [9] is the wake-up radio-based version of the well-known Collection Tree Protocol (CTP) [31]. Both protocols determine a tree-based topology to forward data packets from their sources to the sink (the root of the tree). The topology is maintained in time by using an adaptive beaconing mechanism
595 (control packets) for checking the availability of parent nodes. In order to offset the mismatch between the range of the wake-up radio and that of the main radio (the latter usually being at least twice as long as the former), in CTP-WUR a node forwards data packets directly to its grandparent without waking up its parent, which only acts as a wake-up relay. This reduces energy consumption

600 and end-to-end latency, since relaying nodes only use their wake-up radio, which consumes three orders of magnitude less power than the main radio.

4.3.2. Simulation settings

In addition to the parameter settings described in Section 4.1 we consider packets whose total size is as follows. G-WHARP and GREENROUTES have
605 packets of 58B, which include the bytes of the headers added at different layers. Packets for CTP-WUR are longer because they carry information needed at the MAC and network layer. As a result they are 70B long. G-WHARP, GREENROUTES, and CTP-WUR also transmit control packets whose lengths are 6B (GREEN packets), 14B (total size of RTS and CTS packets), and 25B
610 (beacons), respectively. We conducted experiments with varying wake-up radio data rate. As results at different rates show similar trends, in this paper we report only results on wake-up radios operating at the highest data rate, i.e., 10Kbps. Results are shown for increasing traffic.

4.3.3. Performance metrics

615 Performance comparison is assessed through the investigation of the following metrics.

- *Packet overhead*, defined as the fraction of the total number of control packet transmitted (in bytes) over the total number of packets successfully delivered to the sink (in bytes).
- 620 • *Time spent transmitting and receiving*, computed as the average percentage of the time a node spends transmitting/receiving on the main radio (including the percentage of time nodes spent with their main radio on for hop count determination).
- 625 • *Energy consumption*, defined as the overall energy consumed by the network (including the energy needed for hop count determination when needed).

- *Packet delivery ratio*, defined as the percentage of packets correctly delivered to the sink.
- *End-to-end latency*, defined as the time from packet generation to its delivery to the sink.

630

Similarly to the first set of experiments, all metrics are collected after the initial network setup phase, which includes hop count determination and training times for the energy predictor.

4.3.4. Performance results

635 Results are as shown in figures 7 and 8.

- *Packet overhead*. Fig. 7a depicts the control packet overhead for increasing traffic.

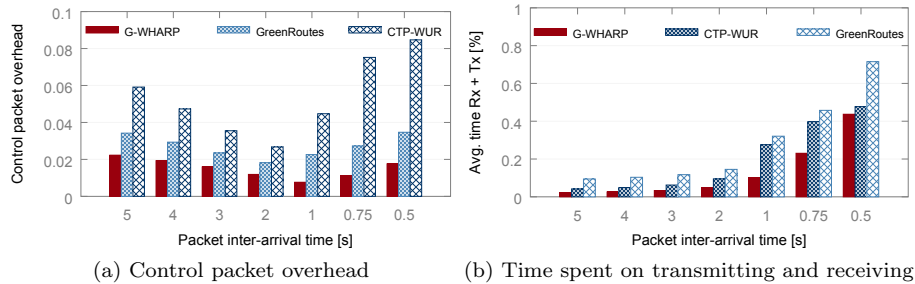


Figure 7: Control packet overhead and time spent with the main radio on.

As a general trend, the packet overhead decreases with increasing traffic up until the inter-arrival time averages to 1s for both GREENROUTES and CTP-WUR and to 0.75s for G-WHARP. This is because G-WHARP and GREENROUTES implement a caching mechanism that allows a sender node to directly transmit packets to a known forwarder (without a forwarder selection process), which results in fewer transmissions of control packets. In CTP-WUR nodes transmit beacons independently of the traffic with a rate that dynamically adapts to network changes, allowing fewer control packet transmissions with increasing traffic. However, as the traffic increases beyond the inter-arrival rate of 1s, nodes start to go *all-off*, which affects the network topology. This

640

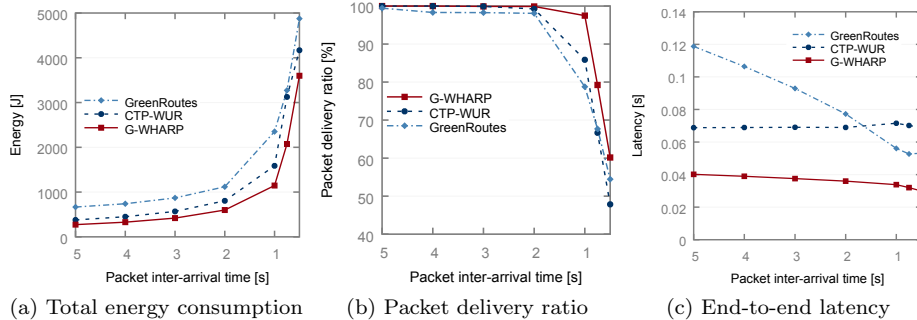
645

makes increasingly difficult to find next hop relays, intensifying control traffic and hence overhead. At the lowest traffic, G-WHARP has a packet overhead that is approximately 1.5 and 2.65 times lower than that of GREENROUTES and CTP-WUR, respectively. This depends on the different size of control packets, as well as on the design principles of each forwarding strategy. GREENROUTES uses a cross-layer mechanism à la RTS/CTS to reserve the channel and select a forwarder. G-WHARP outperforms GREENROUTES because it uses only one control packet, i.e., GREEN packet, while it takes advantage of the wake-up messages to eliminate the transmission of control packets like the RTS, and it also implements a more effective MDP-based next-hop forwarder selection mechanism. It is no surprise that CTP-WUR, which is a tree-based strategy, exhibits the worst performance, independently of traffic. This is because CTP-WUR requires the transmission of beacons, which are 25B long, to establish, repair, or maintain its tree-based topology, leading to higher control packet overhead. Furthermore, G-WHARP and GREENROUTES reap the benefit of the use of caching techniques to directly forward packets to a known forwarder, which results in fewer transmissions of control packets. At the highest traffic we observe that CTP-WUR incurs an overhead that is 4.81 times higher than that of G-WHARP. While GREENROUTES used the ID caching mechanism, it suffers a control packet overhead that is 1.97 times higher than that of G-WHARP. The causes of this are twofold. First and foremost, this is because GREENROUTES has an RTS for each CTS, while G-WHARP saves on RTSs by using the less expensive wake-up radio. Secondly, the G-WHARP reward function explicitly seeks to optimize the forwarding availability of the nodes, which corresponds to shorter *all-off* times.

- *Time spent transmitting and receiving.* As the traffic increases, nodes spend more time transmitting and receiving (Fig. 7b). Independently of traffic, G-WHARP outperforms both protocols, followed by CTP-WUR and GREENROUTES. At the lowest traffic, G-WHARP uses the main radio an average of 4.6 and 2 times less than GREENROUTES and CTP-WUR, respectively. This is because for each data packet G-WHARP requires only the subset of the

energy-capable neighbors of the sender to reply with a GREEN packet. Nodes
 680 running GREENROUTES, instead, spend more time transmitting and receiving
 because they need to transmit CTSs and RTSs, all on their main radio. Fi-
 nally, in CTP-WUR nodes periodically broadcast control packets for building
 or maintaining the tree topology using their main radio. We observe that despite
 the bigger packets, CTP-WUR uses the main radio less than GREENROUTES.
 685 This is because CTP-WUR paces the transmission of control packets according
 to topology dynamics, which are fairly low at low traffic. At the highest traffic,
 nodes in G-WHARP activate their main radio for approximately 16% and 52%
 less time than CTP-WUR and GREENROUTES, respectively. In addition to
 the reasons listed above, we observe that the higher packet overhead of GREEN-
 690 ROUTES and CTP-WUR leads to a higher number of control packet collisions
 on the main radio, which further increases the number of packet re-transmissions
 and overhead. As a result, nodes running GREENROUTES or CTP-WUR spend
 more time with their main radio on.

- *Energy consumption.* The average energy consumed by the network is shown in Fig. 8a.



695 Figure 8: Total energy consumption, packet delivery ratio, and end-to-end latency.

Clearly, energy consumption increases with traffic. Independently of traffic,
 G-WHARP always outperforms the other approaches, followed by CTP-WUR
 and GREENROUTES. Particularly, G-WHARP pays out up to 2.4 and 1.4 times
 less energy than GREENROUTES and CTP-WUR, respectively. This result
 700 is consistent with the time nodes spend transmitting and receiving (Fig. 7b),

the main radio being the major culprit of energy consumption. At the lowest traffic G-WHARP is 59% and 28% more energy efficient than GREENROUTES and CTP-WUR, respectively. As the traffic increases nodes transmit more control and data packets, which requires more energy. At the highest traffic G-
705 WHARP consumes 26% and 14% less energy than GREENROUTES and CTP-WUR, respectively.

- *Packet delivery ratio.* Fig. 8b depicts the average packet delivery ratio of the three forwarding strategies. As a general trend, the packet delivery ratio decreases with increasing traffic. This is because of the higher number of in-
710 terference and hence of re-transmissions. In addition, as traffic increases, the performance is detrimentally affected by the higher number of nodes that go *all-off*. All protocols perform well at lower traffic by successfully delivering almost all data packets to the sink. In fact, only GREENROUTES suffers some packet loss. This is because of the nature of its caching mechanism: A sender
715 that has a next-hop relay in its cache will directly forward the packet to that cached node. As such, if the cached node is no longer available, the sender node will drop the packet after a set number of attempts. The caching mechanism implemented by G-WHARP is not beset by this problem, as a node performs a new forwarder selection if transmissions to a cached node are not successful.
720 At the highest traffic, G-WHARP delivers approximately 10% and 26% more packets to the sink than GREENROUTES and CTP-WUR, respectively. The tree-based topology of CTP-WUR dictates that a node has only one possible forwarder to forward a data packet (its grandparent). When nodes are *all-off* for a longer time, sender nodes cannot always find their preset grandparent on.
725 When the grandparent of a sender node is not reachable for a set number of times, the sender attempts to forward the packet to its parent. If this transmission also fails, the packet is dropped. On the other hand, G-WHARP and GREENROUTES have more candidate forwarders for their packets, which allows them to deliver a higher number of packets.

- *End-to-end latency.* Fig. 8c shows the average end-to-end latency incurred by packets correctly delivered to the sink. As the traffic increases G-WHARP

and GREENROUTES show decreasing latency, while CTP-WUR shows a negligibly increasing one. In the case of G-WHARP and GREENROUTES latency decreases because nodes take increasing advantage of caching, which eliminates the delays of the channel access handshake: The more the packets, the more they are delivered directly through caching, the less the handshake-induced latency. Latency instead remains largely independent of traffic for CTP-WUR, because of the simple tree-based mechanism for determining routes, and the relay of wake-up sequences that reduces route length. Independently of traffic, G-WHARP consistently delivers packets faster. At the lowest traffic, G-WHARP delivers a packet to the sink 2.9 and 1.7 times faster than GREENROUTES and CTP-WUR, respectively. At the highest, G-WHARP packets are successfully delivered to the sink an average of 1.98 and 2.2 times faster than CTP-WUR and GREENROUTES, respectively. G-WHARP outperforms GREENROUTES because of its “lighter” handshake. It is faster of CTP-WUR mainly because of caching. In general, CTP-WUR delivers packets faster than GREENROUTES, because of the cross-layer nature of GREENROUTES, which requires the RTS/CTS handshake. However, we observe that GREENROUTES is faster than CTP-WUR when the traffic inter-arrival time exceeds 2s. This is because at high traffic nodes go *all-off* more frequently and CTP-WUR sender nodes might not find their set grandparent or parent available. If this is the case, the CTP-WUR tree needs to be re-built, which takes time. Instead, sender nodes in GREENROUTES select one relay among multiple nodes, and even if some of them could be *all-off*, the chances of finding at least one available are higher.

We conclude the comparison of G-WHARP, GREENROUTES and CTP-WUR by showing a per-node perspective of the impact on performance of the different forwarding paradigms. Fig. 9 shows a sample network topology. Circles represent nodes that harvest energy through solar panels, while triangles represent nodes that harvest energy through small wind turbines. The sink is depicted as a star at the bottom left corner of the deployment area. Figures 9a, 9c, and 9e depict nodes whose size is proportional to the time they use

their wake-up radio for transmitting and whose color indicates the time their main radio is on. The figures on the right column, namely, figures 9b, 9d, and 9f, show nodes whose size is proportional to the time their main radio is on and whose color indicates the node energy consumption. In general, the smaller the size and the lighter the color, the better.

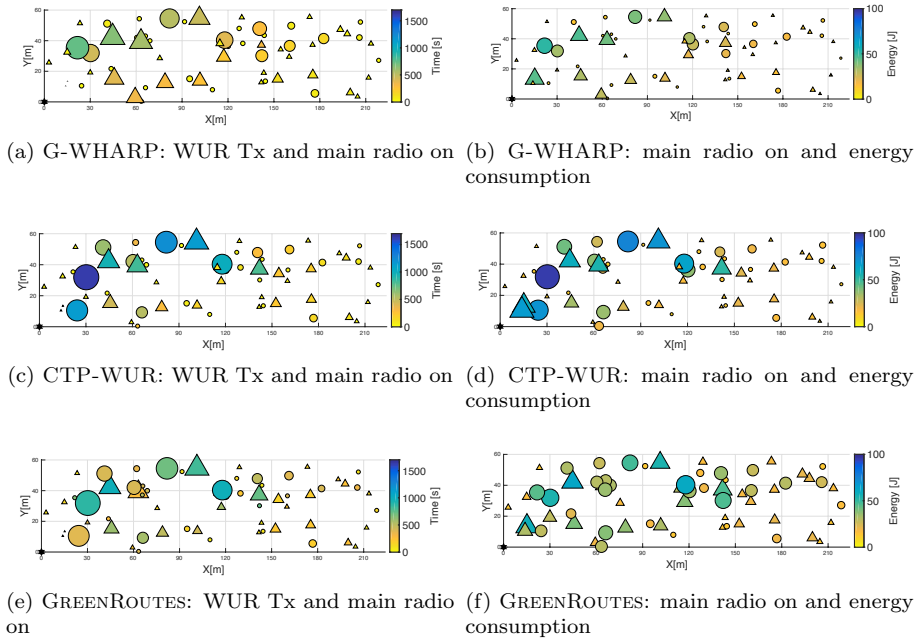


Figure 9: Per node snapshots: G-WHARP vs. CTP-WUR vs. GREENROUTES.

Nodes that are closer to the sink have bigger sizes and darker color because of the higher traffic they manage (“funnel effect,” typical of wireless sensor networking). We also observe that, in general, nodes running G-WHARP have smaller sizes and show lighter colors. From figures 9a, 9c, and 9e we observe that the G-WHARP forwarder selection mechanism distributes traffic in a more balanced way as its MDP-based proactive selection mechanism is solely based on node energy. Not surprisingly, the less a node uses its main radio, the less energy it consumes (figures 9b, 9d, and 9f). Most of the nodes running G-WHARP incur low energy consumption (Fig. 9b), which allows them to be operational for a longer time. In CTP-WUR nodes that are chosen as parents

have bigger size since they use their wake-up radio for transmitting wake-up messages more (Fig. 9c). The higher the time a node spends on communication, the higher its energy consumption (Fig. 9d). Similarly, nodes that are colored in the shades of darker colors correspond to nodes that are more frequently chosen as grandparents. These are the nodes that activate their main radio more frequently. A similar pattern is observed in Fig. 9e for GREENROUTES since nodes can be repeatedly selected as next-hop relays due to caching.

5. Conclusion

This paper concerns green wireless networks, namely wireless networks where nodes are endowed with wake-up radio capabilities and energy harvesters. We present an MDP-aided forwarding strategy, named G-WHARP, where nodes autonomously and proactively decide whether they are available for data forwarding based on their current and harvestable energy. A threshold policy is used for solving the MDP, which is also taken under consideration in the performance evaluation. Through a diverse set of simulation-based scenarios, we show that G-WHARP always outperforms state-of-the-art forwarding solutions by allowing nodes to remain operational for a longer time while consuming less energy. Our results clearly show that the smart exploitation of wake-up radios and energy harvesting technologies leads to superior network performance.

Appendix A.

In the remainder, we prove that our heuristics for solving the MDP obtains optimal solutions as standard solution methods, such as BVI, while *green* decisions may be sub-optimal in decision epochs with positive rewards and energy harvesting intakes. The *optimal policy* π^* , i.e., the forwarding availability of a node (*green* when available to forward or *red* otherwise), results by solving the following MDP Bellman equations (*value functions*):

$$V_n^{\pi^*}(s) = \max_{a_n \in \mathcal{A}} \left\{ r(s_n, a_n) + \gamma \sum_{s_{n+1} \in \mathcal{S}} P_{s_n \rightarrow s_{n+1}}^{a_n} V_{n+1}^{\pi^*}(s_{n+1}) \right\}, \quad (\text{A.1})$$

where a_n ranges in the set of *actions* $\mathcal{A} = \{a_g, a_r\}$, and s_{n+1} ranges in the set of all possible states \mathcal{S} . (The discount factor $0 \leq \gamma \leq 1$ models the uncertainty
800 about the future: The farther the reward is in time, the least important it is.)

Lemma 1. *For each decision epoch $n = 1, \dots, \hat{n}$ the value function $V_n^{\pi^*}(s)$ is non decreasing in s .*

PROOF. Let us define $q_n(k|s_n, a_n)$ as the probability that state s_{n+1} in $n + 1$ exceeds $k - 1$, i.e., that the energy level of a node will be greater than or equal to a value k . Formally:

$$q_n(k|s_n, a_n) = \sum_{s_{n+1}=k}^{\infty} P_{s \rightarrow s_{n+1}}^a. \quad (\text{A.2})$$

We claim that $q_n(k|s_n, a_n)$ is non decreasing in s_n , for all $k \in \mathcal{S}$, $a_n \in \mathcal{A}$, and $n = 1, \dots, \hat{n} - 1$. We can prove this by contradiction. Let us assume that $q_n(k|s_n, a_n)$ is a decreasing function of s_n , for all k , a_n , and decision epoch n . We now consider two consecutive states s_n^+ and s_n^- , with $s_n^+ > s_n^-$. It follows that

$$q_n(k|s_n^+, a_n) - q_n(k|s_n^-, a_n) < 0. \quad (\text{A.3})$$

We denote by s_{n+1}^+ and s_{n+1}^- the states in decision epoch $n + 1$ in which the system transits from s_n^+ and s_n^- , respectively. We consider two cases, depending
805 on the action taken.

1) $a_n = a_r$. In this case we know that state transitions are deterministic and uniquely identified by h_n and e_n^s , leading to $s_{n+1}^+ > s_{n+1}^-$ with probability 1. We can define $q_n(k|s_n^+, a_r)$ as:

$$q_n(k|s_n^+, a_r) = \begin{cases} 1 & \text{if } s_{n+1}^+ \geq k, \\ 0 & \text{otherwise.} \end{cases} \quad (\text{A.4})$$

(We can do similarly for $q_n(k|s_n^-, a_r)$.) Since $s_{n+1}^+ > s_{n+1}^-$, it follows that $q_n(k|s_n^+, a_r) \geq q_n(k|s_n^-, a_r)$, contradicting our assumption.

2) $a_n = a_g$. In this case state transitions are probabilistic and we can define $q_n(k|s_n^+, a_g)$ (and, similarly, $q_n(k|s_n^-, a_g)$) as:

$$q_n(k|s_n^+, a_g) = \begin{cases} \sum_{e_n^f=0}^{e_n^+-k} p^{e_n^f}(e_n^f) & \text{if } k < e_n^+, \\ 0 & \text{otherwise.} \end{cases} \quad (\text{A.5})$$

The intuition is that states higher than k can be reached only if they are lower than the overall energy e_n available for packet forwarding. Since $e_n^+ > e_n^-$, it follows that

$$\sum_{e_n^f=0}^{e_n^+-k} p^{e_n^f}(e_n^f) > \sum_{e_n^f=0}^{e_n^--k} p^{e_n^f}(e_n^f) \quad (\text{A.6})$$

and, consequently, $q_n(k|s_n^+, a_g) \geq q_n(k|s_n^-, a_g)$, contradicting again our assumption.

810 We can conclude that $q_n(k|s_n, a_n)$ is non decreasing in s_n , for all $k \in \mathcal{S}$, $a_n \in \mathcal{A}$, and $n = 1, \dots, \hat{n} - 1$. The lemma claim follows from plugging this result into Proposition 4.7.3 of [32], which also uses that for each action a_n and epoch n the reward function $r(s_n, a_n)$ is non decreasing in s_n . This is true by construction. \diamond

815 The above lemma is key to prove the following result.

Theorem 1. *For each decision epoch n , $n = 1, \dots, \hat{n}$, and state $s_n \in \mathcal{S}$ such that $r(s_n, a_g) < 0$, action a_r is optimal.*

PROOF. We prove this theorem by contradiction. Let us assume that when the reward is negative it would be better to transmit. From equations (1) and (A.1) it follows that:

$$r(s_n, a_g) + \gamma \sum_{e_n^f=0}^{\infty} p^{e_n^f}(e_n^f) V_{n+1}^{\pi^*}(e_n - e_n^f) > \gamma V_{n+1}^{\pi^*}(e_n), \quad (\text{A.7})$$

i.e., the value function associated to a_g is higher than that associated to a_r .

820 Since $r(s_n, a_g)$ is negative, we can write:

$$\sum_{e_n^f=0}^{\infty} p^{e_n^f}(e_n^f) V_{n+1}^{\pi^*}(e_n - e_n^f) > V_{n+1}^{\pi^*}(e_n). \quad (\text{A.8})$$

However, V^{π^*} is non-decreasing by Lemma 1 and $V_{n+1}^{\pi^*}(e_n)$ cannot be lower than the weighted sum of values lower than or equal to $V^{\pi^*}(e_n)$ itself. This contradicts our assumption and ends our proof. \diamond

Theorem 1 allows our solution method to output *red* as the optimal decision, 825 i.e., as if it was computed by standard, yet computationally more expensive techniques for solving MDPs. In the following we show that our method outputs optimal *green* decisions provided that no harvesting happens in epoch n . We start with the following lemma.

Lemma 2. *If there is no harvesting, i.e., $h_n = 0$ for each $n = 1, \dots, \hat{n}$, if 830 $r(s_n, a_g) < 0$ then $V_n^{\pi^*}(s_n) = 0$.*

PROOF. We proceed by backward induction on the number of epochs. In the last decision epoch \hat{n} , $V_{\hat{n}}^{\pi^*}(s) = \max\{r(s_{\hat{n}}, a_g), r(s_{\hat{n}}, a_r)\}$. Since $r(s_{\hat{n}}, a_r) = 0$, it follows that whenever $r(s_{\hat{n}}, a_g) < 0$ it is better to drop packets, and $V_{\hat{n}}^{\pi^*}(s) = 0$. We now consider a generic decision epoch n so that $r(s_n, a_g) < 0$. From equations (1) and (A.1) we can write:

$$V_n^{\pi^*}(s) = \max\{r(s_n, a_g) + \gamma \sum_{e_n^f=0}^{\infty} p^{e_n^f}(e_n^f) V_{n+1}^{\pi^*}(e_n - e_n^f), \gamma V_{n+1}^{\pi^*}(e_n)\}. \quad (\text{A.9})$$

Since there is no harvesting, the energy level in decision epoch $n + 1$ has to be lower than the current energy, independently of the chosen action. As a result, the reward function in the next state s_{n+1} will be negative as well and, by the induction hypothesis and Lemma 1, we can rewrite Equation (A.9) as:

$$V_n^{\pi^*}(s) = \max\{r(s_n, a_g), 0\}. \quad (\text{A.10})$$

Since $r(s_n, a_g) < 0$, we have that $V_n^{\pi^*}(s) = 0$. \diamond

Theorem 2. *If there is no harvesting, i.e., $h_n = 0$, for each decision epoch n , $n = 1, \dots, \hat{n}$, and state $s_n \in \mathcal{S}$ such that $r(s_n, a_g) > 0$, action a_g is optimal.*

PROOF. We proceed by backward induction on the number of epochs. In the last decision epoch if $r(s_{\hat{n}}, a_g) > 0$ it is better to transmit packets, otherwise the reward would be 0. We now assume that, in a generic decision epoch n , $r(s_n, a_g)$ is positive but the optimal action is to drop packets. Our assumption can be expressed by the following equation (a straightforward application of the value function definition):

$$r(s_n, a_g) + \gamma \sum_{e_n^f=0}^{\infty} p^{e_n^f}(e_n^f) V_{n+1}^{\pi^*}(e_n - e_n^f) < \gamma V_{n+1}^{\pi^*}(e_n). \quad (\text{A.11})$$

Let us define $s_{n+1} = e_n$, and let us evaluate Equation (A.11) depending on the value of $r(s_{n+1}, a_f)$.

a) $r(s_{n+1}, a_f) < 0$. In this case we know by the induction hypothesis that $V_{n+1}^{\pi^*}(s_{n+1}) = 0$. Thanks to Lemma 1, Equation A.11 becomes $r(s_n, a_g) < 0$ which contradicts the assumption that $r(s_n, a_g)$ is positive.

b) $r(s_{n+1}, a_f) > 0$. In this case we can expand the second term in Equation (A.11) by exploiting the induction hypothesis as:

$$r(s_n, a_g) + \gamma \sum_{e_n^f=0}^{\infty} p^{e_n^f}(e_n^f) V_{n+1}^{\pi^*}(e_n - e_n^f) < \gamma \left(r(s_{n+1}, a_g) + \gamma \sum_{e_n^f=0}^{\infty} p^{e_n^f}(e_n^f) V_{n+2}^{\pi^*}(e_{n+1} - e_n^f) \right). \quad (\text{A.12})$$

Since there is no harvesting, the energy available in state s_n is greater than or equal to that in state s_{n+1} , which implies that $r(s_n, a_f) \geq r(s_{n+1}, a_f)$. We can simplify the above equation as:

$$\sum_{e_n^f=0}^{\infty} p^{e_n^f}(e_n^f) V_{n+1}^{\pi^*}(e_n - e_n^f) < \sum_{e_n^f=0}^{\infty} p^{e_n^f}(e_n^f) V_{n+2}^{\pi^*}(e_{n+1} - e_n^f). \quad (\text{A.13})$$

If we focus on each couple of terms $V_{n+1}^{\pi^*}(e_n - e_n^f)$ and $V_{n+2}^{\pi^*}(e_{n+1} - e_n^f)$, we know that they are both zero if $r(e_n - e_n^f, a_f)$ is negative (Lemma 2). When it is positive, we can expand $V_{n+1}^{\pi^*}(e_n - e_n^f)$ exploiting the induction hypothesis as:

$$\begin{aligned}
V_{n+1}^{\pi^*}(e_n - e_n^f) &= r(e_n - e_n^f, a_g) + \\
&\quad \gamma \sum_{e^{tx'}=0}^{\infty} p^{e_n^f}(e^{tx'}) V_{n+2}^{\pi^*}(e_{n+1} - e_n^f - e^{tx'}) > \\
&\quad V_{n+2}^{\pi^*}(e_{n+1} - e_n^f). \quad (\text{A.14})
\end{aligned}$$

This is simply the value function equation computed in state $e_n - e_n^f$ and decision epoch $n + 1$. The last inequality holds because we know by induction that if the reward is positive it is better to choose action a_g than to drop packets. This proves that Equation (A.13) cannot hold because we have:

$$V_{n+1}^{\pi^*}(e_n - e_n^f) > V_{n+2}^{\pi^*}(e_{n+1} - e_n^f). \quad (\text{A.15})$$

This contradicts our original assumption, ending our proof. \diamond

840 References

- [1] Fortune Business Insights, Internet of things market analysis - 2026 (July 2019).
- [2] G. Omale, Gartner identifies top 10 strategic iot technologies and trends, Gartner Press Release (November 2018).
- 845 [3] C. MacGillivray, D. Reinsel, Worldwide global datasphere iot device and data forecast, 2019–2023, IDC (May 2019).
- [4] A. Kozłowski, J. Sosnowski, Energy efficiency trade-off between duty-cycling and wake-up radio techniques in IoT networks, *Wireless Personal Communications* 107 (4) (2019) 1951–1971.

- 850 [5] R. Piyare, A. L. Murphy, C. Kiraly, P. Tosato, D. Brunelli, Ultra low power wake-up radios: A hardware and networking survey, *IEEE Communications Surveys & Tutorials* 19 (4) (2017) 2117–2157.
- [6] M. Ghribi, A. Meddeb, Survey and taxonomy of MAC, routing and cross layer protocols using wake-up radio, *Journal of Network and Computer Applications* (2019) 1–54.
- 855 [7] C. Petrioli, D. Spenza, P. Tommasino, A. Trifiletti, A novel wake-up receiver with addressing capability for wireless sensor nodes, in: *Proceedings of IEEE DCOSS 2014, Marina Del Rey, CA, 2014*, pp. 18–25.
- [8] S. Basagni, V. Di Valerio, G. Koutsandria, C. Petrioli, Wake-up radio-enabled routing for green wireless sensor networks, in: *Proceedings of IEEE VTC 2017 Fall, Toronto, Ontario, CA, 2017*, pp. 1–6.
- 860 [9] S. Basagni, C. Petrioli, D. Spenza, CTP-WUR: The collection tree protocol in wake-up radio WSNs for critical applications, in: *Proceedings of ICNC 2016, Kauai, HI, 2016*, pp. 1–6.
- [10] D. Spenza, M. Magno, S. Basagni, L. Benini, M. Paoli, C. Petrioli, Beyond duty cycling: Wake-up radio with selective awakenings for long-lived wireless sensing systems, in: *Proceedings of IEEE Infocom 2015, Hong Kong, China, 2015*, pp. 522–530.
- 865 [11] J. Oller, I. Demirkol, J. Casademont, J. Paradells, G. U. Gamm, L. Reindl, Performance evaluation and comparative analysis of subcarrier modulation wake-up radio systems for energy-efficient wireless sensor networks, *Sensors* 14 (1) (2013) 22–51.
- 870 [12] K. Kaushik, D. Mishra, S. De, K. R. Chowdhury, W. Heinzelman, Low-cost wake-up receiver for RF energy harvesting wireless sensor networks, *IEEE Sensors Journal* 16 (16) (2016) 6270–6278.
- 875

- [13] D. Benedetti, C. Petrioli, D. Spenza, GreenCastalia: An energy-harvesting-enabled framework for the Castalia simulator, in: Proceedings of ACM ENSSys 2013, Rome, Italy, 2013, pp. 1–6.
- [14] D. Purkovic, M. Hönsch, T. R. M. K. Meyer, An energy efficient communication protocol for low power, energy harvesting sensor modules, *IEEE Sensors Journal* 19 (2) (2019) 701–714.
- [15] K. Z. Panatik, K. Kamardin, S. A. Shariff, S. S. Yuhaniz, N. A. Ahmad, O. M. Yusop, S. Ismail, Energy harvesting in wireless sensor networks: A survey, in: Proceedings of IEEE ISTT 2016, Kuala Lumpur, Malaysia, 2016, pp. 53–58.
- [16] D. Mishra, S. De, S. Jana, S. Basagni, K. R. Chowdhury, W. Heinzelman, Smart RF energy harvesting communications: Challenges and opportunities, *IEEE Communications Magazine* 53 (4) (2015) 70–78.
- [17] S. Basagni, M. Y. Naderi, C. Petrioli, D. Spenza, Wireless sensor networks with energy harvesting, in: *Mobile Ad Hoc Networking: The Cutting Edge Directions*, John Wiley and Sons, Inc., Hoboken, NJ, 2013, Ch. 20, pp. 701–736.
- [18] A. V. Sheshashayee, S. Basagni, Multi-hop wake-up radio relaying for the collection tree protocol, in: Proceedings of IEEE VTC 2019 Fall, Honolulu, HI, 2019, pp. 1–6.
- [19] D. Ghose, L. Tello-Oquendo, F. Y. Li, V. Pla, Lightweight relay selection in multi-hop wake-up radio enabled IoT networks, in: Proceedings of IEEE GLOBECOM 2018, 2018, pp. 1–6.
- [20] T. Kumberg, M. Schink, L. M. Reindl, C. Schindelhauer, T-ROME: A simple and energy efficient tree routing protocol for low-power wake-up receivers, *Ad Hoc Networks* 59 (2017) 97–115.

- [21] N. El Hoda Djidi, A. Courtay, M. Gautier, O. Berder, Adaptive relaying for wireless sensor networks leveraging wake-up receiver, in: Proceedings of IEEE ICECS 2018, 2018, pp. 797–800.
- 905 [22] T. Kumberg, L. Reindl, M. Moharrami, C. Schindelbauer, Improving the performance of the cross-layer wake-up routing protocol T-ROME, in: Proceedings of IEEE IWCMC 2017, 2017, pp. 780–785.
- [23] S. Basagni, V. Di Valerio, G. Koutsandria, C. Petrioli, D. Spenza, WHARP: A wake-up radio and harvesting-based forwarding strategy for green wireless networks, in: Proceedings of IEEE MASS 2017, Orlando, FL, 2017, 910 pp. 257–265.
- [24] S. Basagni, V. Di Valerio, G. Koutsandria, C. Petrioli, On the impact of local computation over routing performance in green wireless networks, in: Proceedings of IEEE WoWMoM 2018, Chania, Greece, 2018, pp. 1–9.
- 915 [25] R. S. Sutton, A. G. Barto, Reinforcement Learning: An Introduction, MIT Press, Cambridge, MA, 1998.
- [26] D. K. Noh, K. Kang, Balanced energy allocation scheme for a solar-powered sensor system and its effects on network-wide performance, *Journal of Computer and System Sciences* 77 (5) (2011) 917 – 932.
- 920 [27] A. Boulis, Castalia: Revealing pitfalls in designing distributed algorithms in WSN, in: Proceedings of ACM SenSys 2007, Sydney, Australia, 2007, pp. 407–408.
- [28] M. Paoli, D. Spenza, C. Petrioli, M. Magno, L. Benini, MagoNode++: A wake-up radio-enabled wireless sensor mote for energy-neutral applications, 925 in: Proceedings of ACM/IEEE IPSN 2016, Vienna, Austria, 2016, pp. 1–2.
- [29] S. Basagni, F. Ceccarelli, C. Petrioli, N. Raman, A. V. Sheshashayee, Wake-up radio ranges: A performance study, in: Proceedings of IEEE WCNC 2019, Marrakech, Morocco, 2019, pp. 1–5.

- [30] NREL: measurement and instrumentation data center.
930 URL <https://midcdmz.nrel.gov>
- [31] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, P. Levis, Collection tree protocol, in: Proceedings of ACM SenSys 2009, Berkeley, CA, 2009, pp. 1–14.
- [32] M. L. Puterman, Markov Decision Processes: Discrete Stochastic Dynamic
935 Programming, John Wiley & Sons, 2014.