On the Effectiveness of Semantic Addressing for Wake-up Radio-enabled Wireless Sensor Networks

Abhimanyu V. Sheshashayee
Institute for the Wireless IoT
Northeastern University
Boston, MA, U.S.A.
E-mail: sheshashayee.a@northeastern.edu

Chiara Petrioli
Department of Computer, Control and Management Engineering
University of Rome "La Sapienza"
Rome, Italy
E-mail: petrioli@diag.uniroma1.it

Stefano Basagni
Institute for the Wireless IoT
Northeastern University
Boston, MA, U.S.A.
E-mail: s.basagni@northeastern.edu

***Abstract*—This paper investigates various ways of minimizing energy consumption in Wireless Sensor Networks (WSNs). We are interested in those methods and technologies that allow network nodes to drastically decrease energy consumption by turning off their primary communication circuitry (main radio), arguably the main culprit of energy depletion. We consider WSNs whose nodes operate according to pre-set duty cycles and WSNs with nodes featuring very low-power wake-up radio devices. In these scenarios we evaluate the performance of an energy-aware routing protocol, showing that when nodes wake up their neighbors based on their suitability to forward data packets (*semantic addressing*), energy consumption and network lifetime are remarkably better than when all of a sender neighbors are awoken indistinctly (*broadcast addressing*) and than when nodes duty cycle. Protocols using semantic addressing achieve network lifetimes that are $10\times$ higher than when broadcast addressing is used and three orders of magnitude better than in duty cycle-based networks. We also observe that semantic addressing keeps data latency at bay, achieving end-to-end latency similar to that in networks with nodes with the radio always on.**

I. Introduction

A Wireless Sensor Network (WSN) is comprised of numerous computational sensing devices (generically called *motes*) that are individually capable of recording measurements of their environment and of communicating those measurements by organizing into a multi-hop wireless network [1]. WSN technology has stood the test of time for decades now. It has enabled advances in what is currently referred to as the Internet of Things (IoT), as well as applications such as smart cities, vehicular networking, and smart health [2]. Motes have typically a small form factor, are battery powered and can be deployed by scattering them "randomly" across a given deployment area. For these and other reasons, in most scenarios, regular replacement of a mote batteries is inconvenient if not entirely impossible. This places a fairly strict limitation on the energy available to each mote for its operations and gravely restricts the overall time the network can effectively work (*network lifetime*). Thus, one of the more important problems that besets WSNs is that of managing energy consumption so that their operational lifetime is maximized.

Various approaches for minimizing energy consumption have been explored at all layers of a mote architecture [3], [4]. For current motes, the energy required to communicate tends to be substantially higher than that required by other operations [5]. In particular, the bulk of the energy consumed by the radio is because of *idle listening*, i.e., not for transmitting or receiving data but just waiting for them, which indicates that most of a mote energy is wasted for no useful purposes.

Such energy wastage can be attenuated by the use of *duty cycling* [6]. The mote is considered *awake* when its radio is turned on, and *asleep* when its radio is turned off. The energy consumption is orders of magnitudes lower in the asleep state than in the awake state (typically, μWatts vs. mWatts). For the majority of the time, the mote is asleep. An internal timer periodically awakens it for short intervals, during which communications happen. While this approach may suffer significantly increased data delivery times (end-to-end latency) [7], it can noticeably reduce the idle energy consumed by the radio.

Another approach to attenuate idle energy wastage—one that has seen steadily growing popularity over the past decade—is that of using *Wake-up Radio* (WuR) technology [8], [9], [10]. This approach entails endowing the mote with an ultra-low-power auxiliary radio in addition to its main radio. As with duty cycling, the mote is considered awake when its main radio is turned on, and asleep when its main radio is turned off (which is the default state). The WuR receiver is always on, so that when a mote needs to receive data it can be awoken by the receipt of a *Wake-up Sequence* (WuS) matching the mote *Wake-up Address* (WuA). Thus, the idle energy consumption of the main radio is considerably reduced without suffering the increased latency of duty cycling [7].

Minimizing the energy consumption of communication is particularly important for data collection, which is perhaps the most important function of a WSN. In time, the sensors on the mote produce data (e.g., readings of the environment) from which the mote creates data packets. These data packets need to be delivered to some collection point (the network *sink*), where they can be stored, processed and/or forwarded, as necessary. Often, this is done by routing packets through other motes (multi-hop routing) [11]. Using duty cycling does not require routing schemes to adapt to nodes being awaken or asleep, as communication takes place only when the nodes are awake. However, routing in a WuR-enabled WSN requires careful design choices to effectively take advantage of the ultra-low-power radio. Once a mote has one or more data packets to transmit, which of its neighbors should be awakened to forward those packets towards the sink? An easy solution could be to awaken *all* neighbors of the sender and then proceed to the selection of the relay using the main radio. In this case, the WuS could be a simple, generic *broadcast signal* that, once received, would trigger turning on the main radio. Albeit simple, this approach would potentially awaken many more neighbors than necessary, with detrimental consequences on energy consumption. Although these neighbors could quickly switch back to sleep, they would waste precious energy in the process of determining whether they can relay the data packets or not. For example, in a scheme where a relay is selected based on current energy levels, it would be wasteful to wake up neighboring motes with little energy. In routing schemes based on the distance of a mote from the sink, motes that are farther away than the sender should stay asleep. A more energy judicious approach should allow a sender to wake up only those neighbors that could provide effective advancement of its data packets towards the sink. This requires using WuS capable of encoding protocol-dependent relay eligibility criteria, such as available energy, distance from the sink, mote resources, network traffic, etc. Also, as these criteria can change in time, motes should be able to dynamically change which WuS can wake them up (namely, their WuAs), so that routing utilizes the most up-to-date eligibility information. For instance, a sender with three data packets to transmit would want to wake up only those neighbors that have enough resources (e.g., buffer space) to receive the three packets. Accordingly, every mote in the network sets its WuAs to a value that encodes its eligibility, such as available buffer space. Thus, a neighbor will awaken only if its eligibility (expressed by its WuAs) matches the received requirement expressed by the WuS from the sender (e.g., if the neighbor's buffer has enough space to store three data packets). Neighbors that do not presently satisfy this requirement will still receive the WuS, but will remain asleep, saving energy. If, in time, their buffer spaces change, motes will update their WuAs accordingly. This form of addressing, wherein the WuS encodes eligibility criteria and dynamic properties of motes and network topology, is called *semantic addressing* [8], [9], [12].

In this paper we investigate the effectiveness of semantic addressing in enabling energy efficient routing in WSNs. We contrast WuR-based scenarios with duty cycling-based scenarios. Networks run a simple reactive routing protocol with variants for duty cycling, WuR broadcast addressing and WuR semantic addressing.[1] We quantitatively compare routing performance across these variants. We consider metrics that best show the impact of the different forms of awakenings on network performance, namely, energy consumption, the overall network lifetime and end-to-end latency. Our GreenCastalia-based simulation results show that the energy consumption and network lifetime of WSNs with semantic addressing are consistently better than those with broadcast addressing, which in turn are remarkably better than those experienced by WSNs where nodes duty cycle. Particularly, the average network lifetime is 1718% longer in WSN with semantic addressing than in those with broadcast addressing. Compared to networks with 10% duty cycle, semantic addressing allows WSNs to last an average of 12551% longer. Versus a 100% duty cycle, lifetime is on average 126691% longer. We also observe that semantic addressing does not penalize data latency, achieving results that are comparable to those in networks where the motes are always on (100% duty cycle).

[1] Proactive routing protocols, wherein the next-hop relay towards the sink is preemptively determined, are easier to adapt to a WuR-based scenario. For such protocols, it is enough to awaken the predetermined relay node using its *unicast* address as the WuS [13], [14], [15], [16], [17], [18].

The remainder of the paper is organized as follows. Section II illustrates the considered WSN scenarios. A simple reactive routing protocol and its variants for different forms of awakenings are described in Section II-A. Section III shows the experimental evaluation of the variants of the routing protocol. Finally, the paper is concluded in Section IV.

II. WSN Scenarios

We consider WSNs comprised of N statically deployed nodes that are scattered uniformly across a $L \times W$ deployment area. One of the nodes in the network is designated as the *sink*, to which all data packets are to be routed wirelessly. Each node has a main radio that it uses to transmit and receive packets. While the main radio is turned on (awake state), it consumes power on the order of mW for transmitting, receiving, and idling. To minimize energy consumption, nodes turn off their main radio (asleep state) when they do not have packets to transmit. In asleep state a node consumes orders of magnitude less power (μW). When a data packet is generated, the node turns on its main radio to transmit it, beginning the routing process.

We consider two different WSN scenarios:

- In *duty cycling-based WSNs* nodes operate according to a $d\%$ duty cycle. They stay in the awake state for a predefined $d\%$ of the time, listening for transmissions. For the remaining $(100 - d)\%$ of the time, they are in the asleep state, with their main radio turned off.
- In *WuR-based WSNs* nodes are equipped with a WuR. The WuR consumes power on the order of mW only for transmitting. Power consumption when receiving or idling are in μW, and the WuR is never turned off. Current WuRs achieve this level of energy efficiency by compromising on transmission range and data rate, which are significantly lower than those of the main radio [19]. In WuR-based scenarios, we consider two forms of awakening capabilities:
  - With *broadcast addressing* a sender broadcasts a generic WuS, which is the same for every node. All neighbors that receive this WuS awaken.
  - With *semantic addressing* a sender broadcasts a specific WuS, indicating the characteristics (e.g., resources) a neighbor should have to be a relay. Only the neighbors with those characteristics, namely, the neighbors with WuAs that match that WuS, will awaken.

The energy consumption and lifetime of a node are heavily influenced by how it is awakened [7]. Fig. 1 depicts the energy consumption over time and the lifetime of a receiving node in a single communication link, where the transmitter sends a packet of 70 B every 2 seconds.

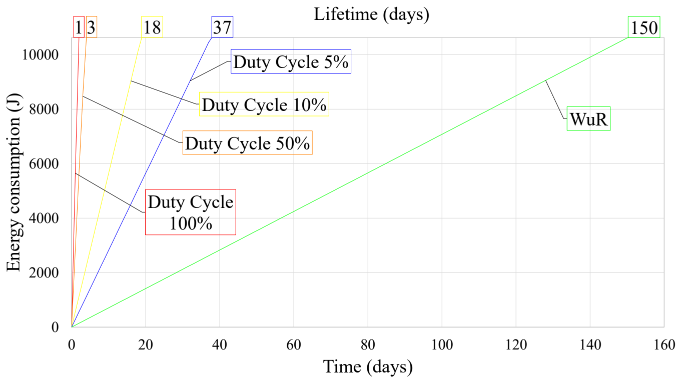

Fig. 1. Energy consumption and lifetime: WuR vs. duty cycling.

A WuR-enabled node with a new lithium-ion AA battery (10656 Joules) lasts over 150 days. A node following a duty cycle has instead a considerably shorter lifetime: It lasts 37 days if it operates with a duty cycle of 5%, 18 days if 10%, 3 days if 50%, and only 1 day if it never goes to sleep (100%).

A. A simple reactive routing protocol

We now describe a simple reactive routing protocol that nodes use to forward data packets toward the sink. This protocol is designed specifically to examine and contrast different awakenings and addressing strategies. In our description we assume that networks are connected, namely, that there is always a wireless route between a source node and the sink (according to both main radio and WuR technology). We also stipulate that nodes are made aware of their neighbors and of the number of hops to the sink (via both main radio links and WuR links) [18]. A node n that has a data packet to forward acts as follows.

1) Node n broadcasts a *Request to Send* (RTS) signal.
2) Every neighbor m that receives that signal responds with a *Clear To Send* (CTS) control packet.
3) After n has received one or more CTSs, it selects one neighbor m' to be the relay and forwards the packet to it.
4) When m' receives the data packet, it responds with an *acknowledgement* (ACK) control packet.
5) When n receives the ACK, it considers the data packet forwarded.

This process repeats until m' is the sink, at which point the data packet has been successfully routed.

Relay selection (step 3) is based on two very simple criteria: The distance h_m from the sink, which should be lower than that of n, h_n, and node m current energy level x_m. If there are multiple neighbors tied for the shortest distance to the sink, then the tie is broken by selecting the neighbor with the highest current energy level. If there is still a tie, it is broken by selecting the neighbor whose CTS was received first. To facilitate this (as well as to avoid collisions) the transmission of the CTS is done with a random jitter of j_{CTS} milliseconds.

After n sends the data packet, it awaits for the ACK from m' for up to t_{ACK} milliseconds, after which it will retry sending the data packet. After a maximum of r_{DP} retries, it returns to step 1) and starts over.

The routing protocol is adapted to work with the different WSN scenarios as follows.

- *Duty Cycling*. The RTS signal of step 1) is sent via node n main radio, and contains among other information, h_n. Every neighbor m with $h_m < h_n$ that is awake receives the RTS and responds with a CTS that contains its eligibility information, namely, x_m. Node n awaits for incoming CTSs until a timeout of t_{CTS} milliseconds. If n does not receive a CTS before the timeout expires, it retries sending the RTS up to a maximum of r_{RTS} retries, after which it drops the data packet. If n receives at least one CTS before the timeout expires, it uses the received information to select the most eligible neighbor as the relay m' and forwards the data packet to it. The remainder of this protocol is the same as the generic variant: m' responds to the data packet with an ACK, and n considers the data packet forwarded when it receives said ACK.
- *WuR Broadcast Addressing*. In this scenario, a node n that has a data packet sends the RTS signal (step 1) as a generic WuS via its WuR. Then it awakens (if not already) and awaits for incoming CTSs on its main radio. Every neighbor m that receives the WuS awakens and responds with a CTS that contains its eligibility information, including h_m and x_m, using its main radio. If n does not receive a CTS before a timeout of t_{CTS} milliseconds, it will retry sending the WuS up to a maximum of r_{WuS} times, after which it drops the data packet and returns to sleep. Otherwise, n selects the relay m' such that $h_{m'} < h_n$ and with the highest x_m, and forwards the data packet to it. Other awakened neighbors wait until a timeout of t_{DP} milliseconds, after which they return to sleep.
- *WuR Semantic Addressing*. Every node m sets its WuA to the (binary encoding of its) distance to the sink and its current energy level: $\text{WuA}_m = \text{bin}\,(h_m)^\frown \text{bin}\,(x_m)$. When node n has a data packet, it broadcasts $\text{WuS}_n = \text{bin}\,(h_n - 1)^\frown \text{bin}\,(x_{m'} - i)$ on its WuR, where $x_{m'}$ is the energy level of the previous selected relay, and i is the number of times n has tried sending the WuS. It then awaits for incoming CTSs on its main radio. Every neighbor m with $\text{WuA}_m = \text{WuS}_n$ awakens and responds with a CTS. If n does not receive a CTS before a timeout of t_{CTS} milliseconds, it increments i by one and transmits WuS_n with the new value of i. After a maximum of r_{WuS} retries, n drops the packet and goes to sleep. Otherwise, when n receives a CTS, it selects its sender as the relay m' and forwards the packet to it. If any other neighbors were awakened, they will wait until a timeout of t_{DP} milliseconds, after which they return to sleep.

III. Experimental Evaluation

We assess the effectiveness of semantic addressing by evaluating the performance of the three variants of the sample reactive protocol described in Section II-A via simulations.

The variants are implemented in the *GreenCastalia* simulator [20], an extension of *Castalia* [21]. All experiments are conducted on simulated networks of *MagoNode++ motes*, which are WuR-enabled low-power wireless sensor nodes [22]. The MagoNode++ motes serve as an ideal platform for investigating WuR characteristics and protocols [18], [19]. All simulation parameters concerning energy consumption, transmission ranges and other hardware related parameters are based on real measurements from testing campaigns of MagoNode++ in scenarios similar to those modeled here [19]. Furthermore, GreenCastalia realistically models additive interference for both main radio and WuR in order to accurately represent real-world collisions, subsequent re-transmissions, and their impact on network performance.

A. Simulation parameter settings

The values of the parameters of our simulations are listed in Table I (communication and energy), Table II (network settings), and Table III (routing protocols). We use the energy and battery models provided by GreenCastalia, which also precisely estimate the energy consumption of various mote components and operations. The data traffic generation is typical of prevailing WSN applications, with packets generated according to a Poisson distribution with parameter i_{DP}. The timeout values for the routing protocols have been selected based on extensive testing, and have been found to elicit the optimal performance for every variant.

TABLE I
COMMUNICATION AND ENERGY PARAMETERS.

PARAMETER	VALUE
Main radio:	
Transmission power	−3 dBm
Power consumption:	
Transmitter	51.9 mW
Receiver	65.4 mW
Channel data rate	250 kbps
Range	70 m
WuR:	
Transmission power	10 dBm
Power consumption:	
Transmitter	90 mW
Receiver	1.071 μW
Data rate	5 kbps
Range	20 m
Energy storage capacity	10656 J

TABLE II
NETWORK PARAMETERS.

PARAMETER [NOTATION]	VALUE
Deployment area $[L \times W]$	56×224 m^2
Network size $[N]$	$\{32, 64, 128\}$
Interval between data packets $[i_{\mathrm{DP}}]$	5 s
Data packet size	70 B
WuA and WuS size	1 B

B. Metrics

We consider metrics that allows us to assess the impact of semantic addressing on the network performance, namely:

- **Energy consumption** (Joules), defined as the energy consumed by a node in an hour.
- **Network lifetime** (hours), conservatively defined as the time until the first node "dies" by energy depletion.
- **End-to-end latency** (milliseconds), i.e., the amount of time required for a data packet to successfully travel from its source to the sink.

All metrics are collected after pruning transient time, thereby representing steady state performance. Results are the average of the outcomes of a number of simulation runs that is large enough to obtain a minimum 5% precision at confidence level 95%.

C. Results

Energy consumption. The majority of energy consumed during the operational lifetime of the network is by the main radio. Specifically, by the receiver as it idly listens for incoming packets. This means that the duty cycling variant with $d = 100\%$ consumes far more energy than any of the other variants. Since the main radio is constantly on, the idle energy consumption by the receiver is tremendous. Following suit, node duty cycling with $d = 10\%$ consume almost exactly a tenth of the energy as the 100% variant. The WuR variants consume considerably lower energy than the duty cycling ones. Again, the bulk of this energy is consumed by the main radio receiver idly listening for the short periods during which the nodes are awake. In the case of the broadcast variant, as all neighboring nodes are awakened, far more energy is consumed than by using semantic addressing, where only a fraction of the neighbors are awakened. The shorter CTS timeout for semantic addressing also reduces the energy wasted by idle listening.

Fig. 2 depicts the effects of increasing the network size N on the energy consumption of the routing protocol variants.

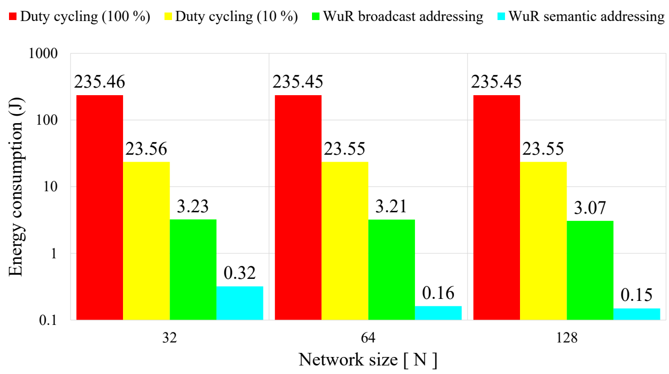

Fig. 2. Energy consumption in networks of different size.

Network size has negligible effect on the duty cycling variants as the energy consumption is due primarily to the main radio, which is periodically turned on for every node.

There is a very slight effect on the WuR broadcast variant. Energy consumption decreases by approximately 5.01% as network size quadruples. While denser networks mean that more nodes are awakened on average, the proportion between

TABLE III
PROTOCOL PARAMETERS.

PARAMETER [NOTATION]	DUTY CYCLING	WUR BROADCAST ADDRESSING	WUR SEMANTIC ADDRESSING
Duty cycle $[d]$	$\{100, 10\}\%$	—	—
RTS, ACK size	6 B	6 B	6 B
Timeout for incoming ACK $[t_{\text{ACK}}]$	15 ms	15 ms	15 ms
Maximum number of data packet retries $[r_{\text{DP}}]$	5	5	5
CTS size	7 B	7 B	6 B
CTS random jitter $[j_{\text{CTS}}]$	$[0, 70]$ ms	$[0, 25]$ ms	$[0, 7]$ ms
Timeout for incoming CTS $[t_{\text{CTS}}]$	85 ms	50 ms	30 ms
Maximum number of signal retries $[r_{\text{RTS}}, r_{\text{WuS}}]$	15	15	15
Timeout for incoming data packet $[t_{\text{DP}}]$	—	60 ms	15 ms

the number of nodes awakened and the network size decreases slightly with larger networks.

With semantic addressing, the size of the network has a considerably more significant effect. The energy consumption more than halves (approximately 53.29% decrease) as the network size quadruples. This is due to the consistently low number of nodes awakened, regardless of network size.

Network lifetime. Fig. 3 depicts the network lifetime for increasing values of the network size N.

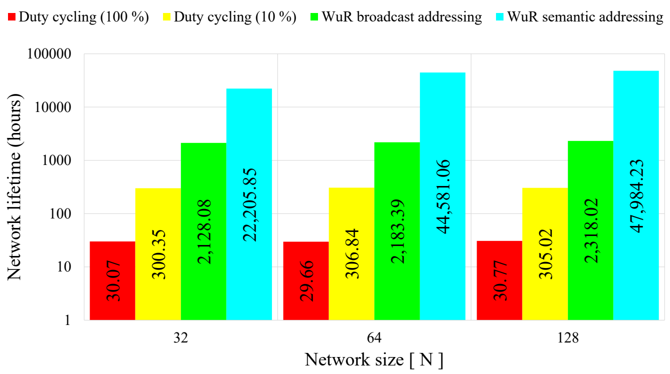

Fig. 3. Network lifetime.

Network size has negligible effect on the duty cycling variants and on the WuR broadcast variant, and leads to improved performance in WuR semantic addressing scenarios. For duty cycling, network lifetime averages 30.17 hours at $d = 100\%$ and 304.07 hours at $d = 10\%$. With broadcast addressing, network lifetime ranges between 2128 hours at $N = 32$ and 2318 hours at $N = 128$. This is approximately a 8.93% increase as network size quadruples. With semantic addressing, these values go up to 22206 hours and 47984 hours respectively, which means that lifetime more than doubles (approximately 116.09% increase) as network size quadruples.

At $N = 32$, the network lifetime for semantic addressing is approximately 1043% that of broadcast addressing, 7393% that of duty cycling with $d = 10\%$, and 73841% that of duty cycling with $d = 100\%$. At $N = 64$, these ratios are respectively 2042%, 14529%, and 150303%. At $N = 128$, the ratios further increase to 2070%, 15732%, and 155930%.

End-to-end latency. End-to-end latency is affected primarily by retries and timeouts. Retries are affected mainly by the duty cycle or by the collisions of CTS packets. In the case of duty cycling, the sender will keep retrying RTSs while the viable next-hop candidates are asleep. It could take several retries before a viable next-hop candidate finally awakens, receives the RTS, and responds with the CTS. Meanwhile, CTS collisions can also prevent the sender from becoming aware of viable next-hop candidates. The routing protocol is designed to minimize the likelihood of CTS collisions by randomizing the time at which a CTS is transmitter via the jitter. The larger the maximum jitter, the larger the corresponding timeout for the incoming CTS needs to be. Thus, the maximum jitter and timeouts are larger for the duty cycling variants than for the WuR variants. Larger timeouts results in proportionally greater latency. If, however, the timeouts were to be kept short, it would result in numerous retries, which would also increase latency. Thus, minimizing latency requires a careful selection of optimal timeouts, which can accommodate all network sizes. Another important factor affecting latency is the number of hops to the sink. The fewer the hops, the faster the data packet gets to its destination. With the duty cycling variants, all communication is done via the main radio. Whereas, with WuR variants, next-hop neighbors are awakened using the WuR. Thus, the range difference between the main radio and WuR results in less hops for the duty cycling variants and more hops for the WuR variants.

Fig. 4 depicts the effects of the network size N on the end-to-end latency.

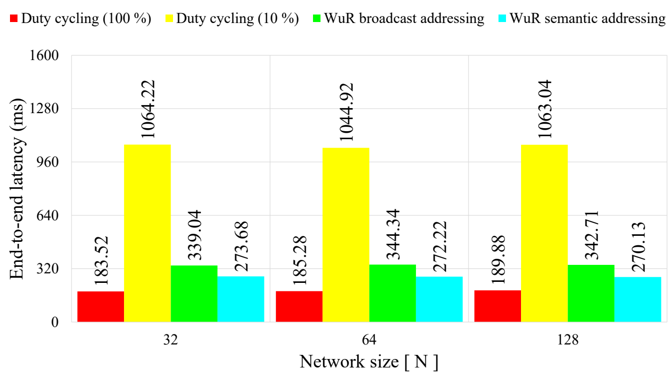

Fig. 4. End-to-end latency.

Network size has a negligible effect on end-to-end latency.

As expected, the latency for the duty cycling with $d = 100\%$ is consistently lower than all other variants. This is because of the lower number of hops to the sink combined with the lower number of retries needed. The lower number of hops is due to the use of the main radio, as explained above. The lower number of retries needed is due to the combination of the fairly large number of next-hop candidates and the routing protocol's mechanics for dealing with CTS collisions. Meanwhile, duty cycling with $d = 10\%$ consistently has the highest latency. This is due to both the higher number of retries required as well as the larger timeout of the duty cycling variants. Together, they increase the cumulative latency considerably. The WuR variants have latency that fall in-between the two duty cycling variants. The latency for the semantic addressing variant is consistently lower than that of the broadcast variant. This is mainly because a sender node does not have to wait for all incoming CTSs; rather, it can immediately respond to the first CTS it receives. While the broadcast addressing variant must wait for the full $t_{\text{CTS}} = 50$ ms timeout before it can start forwarding its data packet, the semantic addressing variant can receive a CTS and forward a data packet in as little as 22 ms.

IV. Conclusion

WuR technology allows reactive routing protocols to considerably reduce energy consumption without a significant sacrifice of latency. However, to gain the most benefit, it is useful to consider strategies that minimize the number of nodes awakened. In particular, *semantic addressing* allows nodes to awaken only the most eligible neighbors for relaying packets to the sink. This paper presents a simulation-based experimental evaluation of the effectiveness of semantic addressing used for reactive routing over WuR-enabled WSNs. We contrast the performance against that of WuR-based broadcast addressing and of duty cycle-based WSNs (for scenarios with motes that are not WuR-enabled). We conduct GreenCastalia-based simulations on networks of varied size. Our results show that semantic addressing has over 20 times the network lifetime of broadcast addressing. For duty cycling, this ratio rises to over 150 times against routing in networks with 10% duty cycle, and over 1550 times against those with a 100% duty cycle. This impressive gain in longevity is complemented by low end-to-end latency that is similar to scenarios where the mote main radio is always on.

Acknowledgments

Chiara Petrioli was supported in part by the Italian MUR-funded PON project SMARTOUR.

References

[1] H. M. A. Fahmy, *Wireless Sensor Networks: Concepts, Applications, Experimentation and Analysis*, ser. Signals and Communication Technology. Springer Science+Business Media Singapore, 2018.

[2] B. Nayak, S. K. Pani, T. Choudhury, S. Satpathy, and S. N. Mohanty, Eds., *Wireless Sensor Networks and the Internet of Things: Future Directions and Applications*. Apple Academic Press, September 2021.

[3] M. Tahir and R. Farrell, "Optimal communication-computation tradeoff for wireless multimedia sensor network lifetime maximization," in *Proceedings of IEEE WCNC 2009*, Budapest, Hungary, April 5–8 2009, pp. 1–6.

[4] A. Cammarano, D. Spenza, and C. Petrioli, "Energy-harvesting WSNs for structural health monitoring of underground train tunnels," in *Proceedings of IEEE INFOCOM Workshops*, Turin, Italy, April 14–19 2013, pp. 75–76.

[5] S. Basagni, V. Di Valerio, G. Koutsandria, and C. Petrioli, "On the impact of local computation over routing performance in green wireless networks," in *Proceedings of IEEE WoWMoM 2018*, Chania, Greece, October 12–15 2018, pp. 1–9.

[6] R. C. Carrano, D. Passos, L. C. S. Magalhaes, and C. V. N. Albuquerque, "Survey and taxonomy of duty cycling mechanisms in wireless sensor networks," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 181–194, June 2014.

[7] S. Basagni, F. Ceccarelli, F. Gattuso, and C. Petrioli, "Demo abstract: Abating LPL-induced latency with wake-up radio technology," in *Proceedings of ACM IoT DI 2017*, Pittsburg, PA, April 18–21 2017, pp. 1–2.

[8] C. Petrioli, D. Spenza, P. Tommasino, and A. Trifiletti, "A novel wake-up receiver with addressing capability for wireless sensor nodes," in *Proceedings of IEEE DCOSS 2014*, Marina Del Rey, CA, May 26–28 2014, pp. 18–25.

[9] D. Spenza, M. Magno, S. Basagni, L. Benini, M. Paoli, and 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, April 26–30 2015, pp. 522–530.

[10] R. Piyare, A. L. Murphy, C. Kiraly, P. Tosato, and D. Brunelli, "Ultra low power wake-up radios: A hardware and networking survey," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2117–2157, July 24 2017.

[11] C. Nakas, D. Kandris, and G. Visvardis, "Energy efficient routing in wireless sensor networks: A comprehensive survey," *Algorithms*, vol. 13, no. 3, pp. 1–65, 2020.

[12] G. Koutsandria, V. Di Valerio, D. Spenza, S. Basagni, and C. Petrioli, "Wake-up radio-based data forwarding for green wireless networks," *Computer Communications*, vol. 160, pp. 172–185, July 2020.

[13] T. Stathopoulos, M. Lukac, D. McIntire, J. Heidemann, D. Estrin, and W. Kaiser, "End-to-end routing for dual-radio sensor networks," in *Proceedings of IEEE INFOCOM 2007*, Anchorage, AK, May 6–12 2007, pp. 2252–2260.

[14] S. Basagni, C. Petrioli, and D. Spenza, "CTP-WUR: The collection tree protocol in wake-up radio WSNs for critical applications," in *Proceedings of IEEE ICNC 2016*, Kauai, HI, February 15–18 2016, pp. 1–6.

[15] M. Iwata, S. Tang, and S. Obana, "Sink-based centralized transmission scheduling by using asymmetric communication and wake-up radio," in *Proceedings of IEEE WCNC 2017*, San Francisco, CA, March 19–22 2017, pp. 1–6.

[16] T. Kumberg, M. Schink, L. Reindl, and C. Schindelhauer, "T-ROME: A simple and energy efficient tree routing protocol for low-power wake-up receivers," *Ad Hoc Networks*, vol. 59, pp. 97–115, May 1 2017.

[17] T. Kumberg, L. Reindl, M. Moharrami, and C. Schindelhauer, "Improving the performance of the cross-layer wake-up routing protocol T-ROME," in *Proceedings of IWCMC 2017*, Valencia, Spain, June 26–30 2017, pp. 780–785.

[18] A. V. Sheshashayee and S. Basagni, "Multi-hop wake-up radio relaying for the Collection Tree Protocol," in *Proceedings of IEEE VTC 2019 Fall*, Honolulu, HI, September 22–25 2019, pp. 1–6.

[19] S. Basagni, F. Ceccarelli, C. Petrioli, N. Raman, and A. V. Sheshashayee, "Wake-up radio ranges: A performance study," in *Proceedings of IEEE WCNC 2019*, Marrakech, Morocco, April 15–19 2019, pp. 1–5.

[20] D. Benedetti, C. Petrioli, and D. Spenza, "GreenCastalia: An energy-harvesting-enabled framework for the Castalia simulator," in *Proceedings of ACM ENSSys 2013*, Rome, Italy, November 11–14 2013, pp. 1–6.

[21] A. Boulis, "Castalia: Revealing pitfalls in designing distributed algorithms in WSN," in *Proceedings of ACM SenSys 2007*, Sydney, Australia, November 6–9 2007, pp. 407–408.

[22] M. Paoli, D. Spenza, C. Petrioli, M. Magno, and L. Benini, "MagoNode++: A wake-up radio-enabled wireless sensor mote for energy-neutral applications," in *Proceedings of ACM/IEEE IPSN 2016*, Vienna, Austria, April 11–14 2016, pp. 1–2.