CTP-WUR: The Collection Tree Protocol in Wake-up Radio WSNs for Critical Applications

Stefano Basagni
Department of Electrical and Computer Engineering
Northeastern University
E-mail: basagni@ece.neu.edu

Chiara Petrioli, Dora Spenza
Department of Computer Science
University of Rome “La Sapienza”
E-mail: {petrioli, spenza}@di.uniroma1.it

***Abstract*—Allowing the nodes of a wireless sensor network (WSN) to turn their radio off periodically noticeably increases network lifetime. *Duty cycling*, however, does not eliminate idle listening, comes at the price of longer latencies and obtains lifetimes that are still insufficient for many critical applications. Using a wake-up receiver (WUR) allows actual communications on the main radio only for transmission or reception, virtually eliminating node idling. However, the range of current WUR prototypes is still significantly shorter than that of the main radio, which can challenge the use of existing WSN protocols in WUR-based networks. In this paper we present an approach to mitigate this limitation of wake-up-based networks. In particular, we show that the Collection Tree Protocol (CTP), a standard protocol for data gathering in WSNs, suitably redefined to work on WUR-endowed nodes, achieves lifetimes of several decades. This constitutes a remarkable improvement over duty cycle-based solutions, where CTP makes the network lasts only a handful of months. At the same time, our WUR-based approach obtains data latencies comparable to those obtained by keeping the main radio always on.**

I. Introduction

A well-established technique for saving energy in a wireless sensor networks (WSN) is to allow its nodes to turn their radio off (“sleep mode”) and on (“awake mode”) according to a pre-set duty cycle. In fact, most commercially available sensor nodes feature low power modes with consumptions of few tens of μWs, i.e., three order of magnitudes less than when the radio is on. Use of duty cycle-based media access control (MAC) protocols has been shown to be useful in prolonging the lifetime of a node, and therefore that of the whole WSN. However, depending on the time spent by a node in sleep mode, duty cycling MACs impose considerable data latencies: Data delivery is beset by delays of tens of seconds, or of minutes, which can be unacceptable for many applications. More important, the betterment in network lifetime enabled by duty cycling is largely insufficient to support critical applications such as the structural health monitoring of a bridge, of underground tunnels [1] or of other key infrastructures. In these applications the sensor nodes are often embedded inside the structure itself (e.g., in the concrete of a bridge deck), and as such they require the network to last decades rather than months or a few years. For these reasons, recent research has been focusing on the design of WSN nodes that include wake-up radios, i.e., on-board circuitry used to wake up neighboring nodes consuming power in the order of a few μWs. When a node has packets to transmit, it uses its wake-up transmitter to wake up its neighbors. Data communication then happens by using the node main radio. This combined use of wake-up and main radios virtually eliminates the need for duty cycling, and therefore idle power consumption, obtaining network lifetimes of several decades, which is unheard of for WSNs. Wake-up radios have thus the potential to profoundly change the way WSNs operate, addressing the latency vs. energy consumption tradeoff by enabling low latency data collection at minimum energy cost.

Research on wake-up radios has so far mainly focused on ways of designing wake-up circuitry with energy requirements of a few μWs [2], [3], [4]. Some of the designs currently available are range-based: All nodes within communication range wake up whenever a wake-up request is transmitted. Other designs enable node to wake up specific neighbors based on their address. Exploiting this latter feature, advances in the design of wake-up receivers (WURs) are enabling the development of novel paradigms for wake-up radio-based networks. For example, *semantic addressing* allows the design of selective awakening-aware protocols in which the WUR address assigned to a node is not just its ID, but rather a wake-up sequence having a specific semantic meaning. Recent works have shown how semantic addressing can be used to naturally express complex relay selection policies [2], and to implement efficient broadcasting and harvesting-aware routing primitives to significantly improve system performance [5].

In this paper we contribute to the research on “idle-free” WSNs by presenting a wake-up radio-based solution for critical applications that extends the well-known Collection Tree Protocol (CTP) [6] for Wireless Sensor Networks. In particular:

- We propose CTP-WUR, an extension of the *de facto* standard in WSN data collection routing—the Collection Tree Protocol (CTP) [6]—for wake-up radio-based networks. CTP-WUR is based on the idea of enabling the relay of WUR requests to extend the achievable wake-up range. This improves the performance of using short-range WURs in practical application scenarios, by reducing the number of hops needed to relay data packets.
- We compare the performance of CTP-WUR and CTP with duty cycle in application scenarios for long-lasting WSNs while varying critical parameters such as duty

cycle and data traffic. Simulation are performed by using GreenCastalia [7], an open-source extension to the Castalia simulator [8] that we have developed to accurately model energy-related aspects of WSNs. Our GreenCastalia model of a WUR is based on extensive experimentation on a WUR prototype. Results show that CTP-WUR achieves network lifetimes several decades longer than those of CTP with duty cycle, enabling very low latency data collection at minimum energy cost.

The remainder of the paper is organized as follows. Section II summarizes previous work on wake-up radio-assisted protocols for WSNs. In Section III we describe the architecture of a WSN node with a wake-up radio. A brief description of CTP is presented in Section IV. In Section V, we introduce CTP-WUR, an extension of CTP for WUR-based networks. The comparative performance evaluation of CTP-WUR and CTP with duty cycle is shown in Section VI. Section VII concludes the paper.

II. Related Works

Research on wake-up radios has so far mostly focused on the design of wake-up circuitry with energy requirements comparable to those of the main radio in sleep mode. Several WUR prototypes, with varying performance in terms of sensitivity, maximum reachable wake-up distance, latency and power consumption, have been recently presented [3], [4], [9], [10]. Despite the increasing choice of wake-up architectures, however, little attention has been devoted to the design of communication protocols that natively support radio-triggering features. The first works on WSNs that explicitly consider the presence of an extra radio for wake-up purposes were concerned with energy-saving topology management, and are suitable for WSN applications that are delay tolerant [11], [12], [13]. Some protocols employing addressed-based wake-up have been proposed, in which however nodes are awaken based solely on their static ID. For example, addressed-based wake-up is described by Marinkovic et al. for a protocol for single-hop WBANs that works with a wake-up receiver [14]. While this protocol works well in a small, single-hop network like a WBAN, it lacks in flexibility to work for more general WSNs applications. A WUR-based clustering protocol is presented by Blanckenstein et al. in [15], which uses dynamic wake-up addressing to wake up nodes depending on the data they have sensed. Their approach clusters nodes with similar data readings and elects a cluster head to forward the data to their destination. The concept of semantic WUR addressing, in which a pool of multiple WUR addresses is assigned to a node and dynamically updated based on its status, have been recently proposed by Petrioli et al. in [5]. A dedicated WUR-enabled communication stack exploiting selective wake-ups and dynamic address assignment is shown to enhance system performance. The use of semantic addressing to implement complex relay selection policies is demonstrated by Spenza et al. in [2], though the design of a cross-layer solution exploiting selective awakenings for data gathering in sensing systems. An analytical and empirical comparison of the achievable lifetimes of WUR-based systems against Low Power Listening techniques is performed by Prinn et al. [16]. They quantify the benefits of using a wake-up receiver in terms of energy consumption through in-lab power measurements, but do not evaluate other relevant metrics, such as latency. As for the use of a wake-up radio in multi-hop WSNs, Chen et al. recently demonstrated a multi-hop wake-up sensor network using passive RFID-based WURs [17]. A simple protocol is defined to determine which nodes should rebroadcast WUR requests to their neighbors. WUR addressing is not supported in the current implementation.

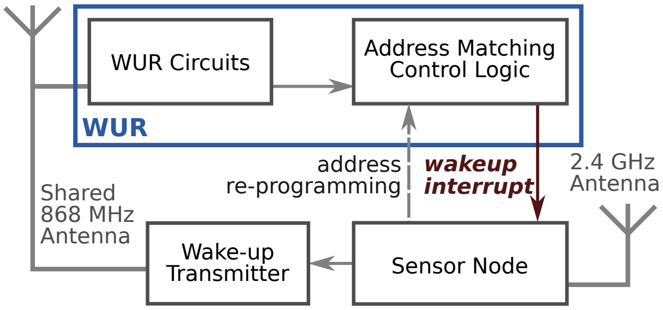


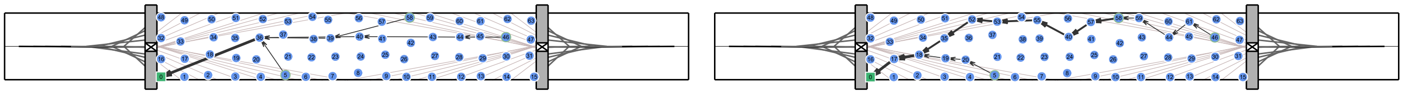
Fig. 1. The architecture of a WUR node with an out-of-band WUR.

III. Node Architecture

Fig. 1 depicts the main building blocks of a wake-up radio mote. Each node is equipped with a main transceiver operating in the 2.4GHz ISM band and with an out-of-band WUR working in the 868MHz ISM band. Wake-up sequences (i.e., addresses) are transmitted by using a dedicated transmitter that operates in the same band of the WUR possibly sharing the same antenna (via a low-power antenna switch). Addressing is performed by an ultra-low power microcontroller (MCU) that wakes up the main node by generating an interrupt whenever a valid address is detected. Multiple WUR addresses can be assigned to the node and dynamically managed over time. Nodes receiving a wake-up signal, even if they are in sleep mode (i.e., with their main radio and MCU off), can determine if they are intended receivers by using their ultra-low power WUR MCU to check for a matching address in their address pool. Serial communication between the main MCU and the WUR MCU enables dynamic address reprogramming. The architecture depicted in Fig. 1 is based on that of a recent WUR prototype of ours that allows inexpensive integration on existing WSN motes [2]. It supports multiple WUR addresses and dynamic address reprogramming, key features for the design of protocols based on selective addressing.

IV. CTP

The Collection Tree Protocol (CTP) [6] is the *de facto* standard in data collection for WSNs, being widely used in a variety of deployments and testbed. CTP is a distance-vector tree-based collection protocol: Nodes build and maintain a minimum-cost tree rooted at the network data collection point (the sink). Each node in the network maintains an estimate



(a) Main radio topology

(b) WUR topology

Fig. 2. CTP routes in a multi-hop WSNs when using a) the main radio topology and b) the WUR topology. The square node (in green) is the network sink.

of the cost of its route to the sink, expressed in terms of Expected Transmission Count (ETX). In particular, the sink advertise a cost of zero, while each node computes its cost as the cost of its next hop (i.e., its *parent*) plus the cost of its link to the parent. Routing information are exchanged by broadcasting control beacons, which contain the transmitter local cost estimate. Adaptive beaconing (based on the Trickle algorithm [18]) is employed to reduce the frequency of beacons in the long run. For topology maintenance, datapath validation is also used. Since each data packet contains the transmitter local cost estimate, loops or inconsistencies in the topology can be detected by comparing the transmitter cost with that of its next hop: if a transmitter advertised cost is no greater than that of the receiver, topology information are assumed to be stale and topology repair is started.

V. CTP-WUR

A critical aspect in the design of WURs is the tradeoff between receiver sensitivity (and therefore wake-up range), and power consumption. The power consumption of current WUR prototypes comes at the cost of reduced communication ranges. Typical distances at which a node can be successfully awaken vary from 11m [19] (passive wake-up systems) to 45m [2] (semi-active WURs). While a wake-up range of up to a few tens of meters is reasonable for many applications, the network topology induced by the wake-up radio (WUR topology) can be very different than that induced by the main radio. This brings forward new challenges for well-know protocols, such as CTP, in WUR-based networks. In particular, routing of packets in the WUR topology happens through paths that can be significantly longer that those made up of links of the main radio. This affects latency detrimentally, which, together with network lifetime is one of the most stringent requirements of critical applications. Fig. 2 depicts routes obtained by CTP run over the main radio topology (a) and on the WUR topology (b) (see Section VI-A for the simulation settings). In this example, the average route length on the WUR topology is more than twice than that on the main radio topology, which basically doubles data latency. To mitigate this fundamental limitation of wake-up-based networks, we propose a backward-compatible extension of CTP, named CTP-WUR, that obtains reduced end-to-end latency by extending the achievable wake-up range. The key idea of CTP-WUR is to allow the relaying of WUR requests. CTP-WUR works on top of a channel access protocol designed to exploit the flexible addressing features of the architecture described in Section III. Our proposed extensions works as follows. Each node in the network is assigned a set of wake-up addresses: A unique WUR address, a unique WUR relay address and a broadcast WUR address that is shared by all nodes in the network. Actual packet transmission is preceded by a wake-up phase in which the intended receivers are awaken based on their WUR addresses. Handling of broadcast packets and unicast packets in CTP-WUR is described below.

A. Broadcast packets

When a broadcast packet (e.g., a CTP beacon) should be transmitted, the sender node awakens its whole neighborhood by sending a wake-up packet addressed to the WUR broadcast address. Nodes receiving this packet turn their main radio on and set it to receive mode (RX). After transmitting the wake-up packet, the sender node turns its own main radio on and awaits for a given (platform dependent) amount of time to allow the main radio of the receivers to enter RX. After transmitting the broadcast packet on the main radio, the sender goes back to sleep.

B. Unicast packets

CTP builds and maintains a minimum-cost tree to the sink such that each node in the network (but the sink) has a parent node through which data traffic is routed. Due to the shorter range of the wake-up radio, a node and its CTP parent, which are on each other main radio communication range, are generally not connected in the WUR topology. Since a node cannot directly wake up its parent, data forwarding in a wake-up-based WSN must happen through an intermediate data relay. An example in shown in Fig.3. Nodes A, B and C are all on each other main radio communication range. However, node A and node C are not connected in the WUR topology. In a traditional approach, to forward data to node C node A should use node B as an intermediate data relay. In other words, A should wake up node B (by sending it a WUR request) and then transmit the data packet to B (on the main radio). In turn, node B should wake up node C (WUR-based communication) and then transmit the data packet to C (main radio).

To reduce the route length of data packets, CTP-WUR enables relaying of WUR requests. In particular, whenever a node (e.g., node A of Fig. 3) has a unicast data packet to send, it tries to forward it directly to the parent of its parent (node C), rather than to its parent (node B). If the attempt succeeds, relaying of the data packet through the intermediate relay hop (node B) is avoided, thus reducing both latency and energy consumption. To do so, node A sends a special WUR request to node B, which can awake node C on its behalf, *without the need of waking up its main radio*. Relay of WUR packets are requested by means of a dedicated WUR address, which contains the unique address of the node and

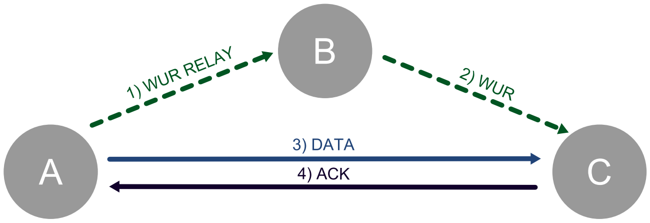


Fig. 3. Example of data forwarding using CTP-WUR.

an additional flag to indicate that the WUR request should be forwarded to the receiver own parent. The relay of a WUR packet is performed by the WUR transmitter. After forwarding the WUR request, node B sets an internal timer to avoid starting a new data transmission before the packet forwarding from A to C is expected to complete. When the WUR request is received by node C, its WUR generates an interrupt to wake up the node. The main radio of node C is then turned on and set to RX. If no packet is received after a predefined time, e.g., because of interference or of a false wake-up positive, node C goes back to sleep. The sender node transmits the outgoing packet on the main radio, and awaits for the receiver acknowledgment before going back to sleep.

Depending on the network topology, it may happen that data packets sent by the sender node cannot be received by the parent of its parent, as it may be out of range. To handle this case, each node tries to send data packets directly to the parent of its parent for a number T of times. If all attempts fail, the node sets the parent of its parent as not reachable, and it uses only its parent for data forwarding.

It is worth noting that the information about the parent of its current parent can be obtained by each node with no additional communication overhead, as the parent field is already included in the CTP routing frames [20].

VI. Performance evaluation

We show the benefits of WUR-based communications and of WUR request relaying by comparing the performance of CTP-WUR and CTP with duty cycle, termed CTP LPL (as duty cycling is implemented through Low Power Listening). We implemented both versions of CTP in GreenCastalia [7], a freely available extension to the Castalia simulator [8].

A. Simulation scenarios and settings

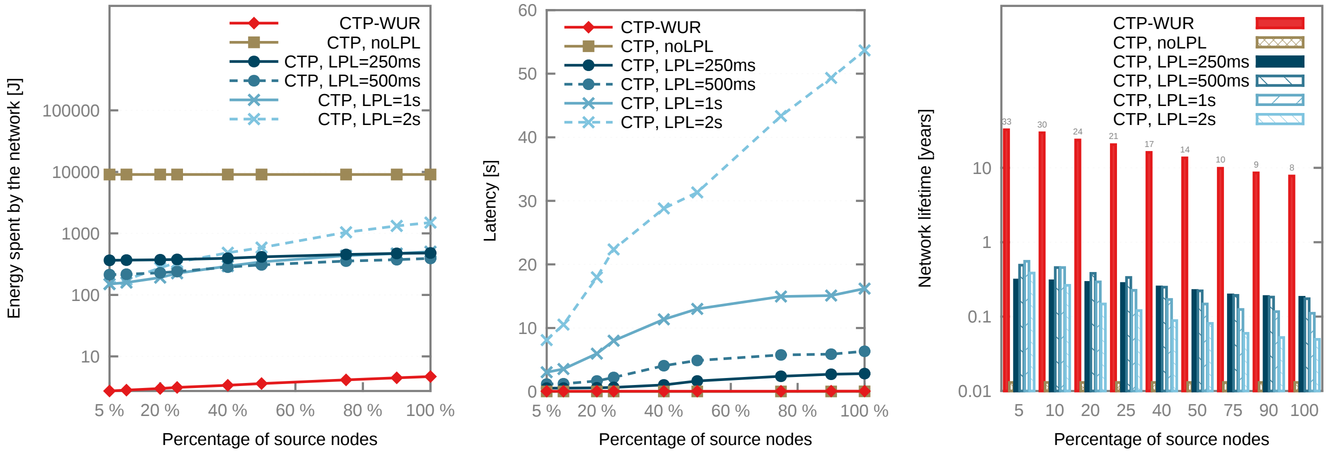
We consider WSNs with 64 nodes embedded in the reinforced concrete deck of a bridge. As a reference, we use the Leonard P. Zakim Bunker Hill Memorial Bridge, one of the widest cable-stayed bridges in the world. The bridge has been in use in Boston since 2003. The area monitored by our networks concerns the main span of the bridge, which is 227m long and 56m large (highlighted in the figure). The network sink is located at one of the corners of the main span. Nodes are positioned according to a randomized grid deployment: Nodes are laid down as a 16×4 grid where the actual location of each node is randomly displaced from the precise grid point by 10%. An example is depicted in Fig. 2. The sink is (the square) in the lower left corner.[1]

The network is deployed for monitoring the temperature and humidity in the deck. Source nodes, which are equipped with on-board Sensirion SHT1x sensors, periodically perform temperature and humidity measurements and generate a data packet to report sensor readings to the sink. Based on the sensor datasheet specifications, we set the power consumption of sensing to 3mW, and the time needed for the measurement to complete to 171ms [21]. In our simulations, we set the packet generation interval to 300s, and varied the percentage p of source nodes between 5% and 100% (excluding the sink). The size of each data packet is set to 70B to include application payload (sensing measurements) and headers and trailers added by the lower layers. The channel data rate is 250Kbps. Routing in the network is performed by using CTP. We ported to GreenCastalia the publicly available implementation of CTP [22], and extended the CTP-compliant MAC module provided in the same package to support both WUR and LPL-enabled communications. In our simulations we used the default parameter setting of CTP, and set LPL parameters based on the TinyOS 2.1 implementation of BoX-MAC-2 [23]. Nodes using LPL follow asynchronous wake-up schedules, performing periodic receive checks every $l$ms, $l \in \{0, 250, 500, 1000, 2000\}$. In other words, nodes sleep for 250ms, 500ms, 1s or 2s between successive checks for channel activity. Setting $l = 0$ corresponds to the case with no LPL, i.e., with the main radio always on. We implemented CTP-WUR by extending CTP so that nodes maintain updated information about the parent of their current parent, as detailed in Section V.

We use the energy model of the TelosB mote [24], which we extend to account for the power consumption of both the WUR (including its integrated MCU) and the WUR transmitter (the low-power CC1101 transceiver from Texas Instruments). Based on this model, each node operates in one of five base power modes, depending on the state of their microcontrollers and transceivers. Simulations take into account the power consumption of the node in each state, as well as the transition times and the average power consumption of each of the node components during state switching. Mimicking the TinyOS implementation, the MCU of the nodes is put into the LPM3 low power sleep state whenever the radio is off and the node is not performing sampling. In this state, the nominal current consumption of the main node is 5.1μA [24].

The default GreenCastalia settings were used for channel and radio models. The transmission power of a node main transceiver is set to -3dBm for energy conservation, and its transmission range is 70m. The average path loss between nodes in the network is estimated by using the lognormal shadowing model. Packet collisions are determined based on the additive interference model, according to which simultaneous transmissions from multiple nodes are calculated as

[1] Simulations considering a different deployment, where nodes are randomly and uniformly scattered throughout the deck, produce results with trends that are very similar to those shown here.



(a) Total energy consumed by the network

(b) Packet latency

(c) Estimated lifetime of the network in years

Fig. 4. Performance comparison of CTP-WUR and CTP LPL for increasing fraction of source nodes.

interference at the receiver, by linearly adding their effect. Packet reception probability for each link is computed based on SINR and packet size. WUR sequences are sent at 5Kbps, and consists of 1B of data. The WUR is modeled according to the specifications of our WUR prototype and its parameters are set to values obtained by actual measurements [2]. In particular, the power consumption of the WUR is set to 1.276μW. Its sensitivity is set to -55dBm, which obtains a maximum wake-up range of 45m when transmitting at $+10$dBm. Based on experimental results, we model the wake-up probability as a function of the power received by the WUR and of the data rate used to transmit wake-up sequences.

B. Performance metrics

The performance of CTP-WUR and of CTP LPL is evaluated with respect to the following metrics.

1) The total energy consumed by all network nodes, but the sink.
2) The average data delivery latency, computed as the mean of the time it takes to successfully deliver a data packet to the sink.
3) The packet delivery ratio, i.e., the fraction of generated data packets that is successfully delivered to the sink.
4) The expected lifetime of the network, conservatively defined as the time when the first node dies.

Results have been obtained by averaging outcomes from 400 simulation runs on connected topologies, each lasting 3600s. This obtains us results with a 95% confidence interval and a 5% precision. All metrics are collected after an initial transitory time (steady-state performance).

C. Performance results

1) Energy consumption: Fig. 4(a) shows the energy consumed by CTP-WUR and CTP LPL for increasing percentages of source nodes. CTP-WUR shows remarkable performance, reducing the energy consumption of the network by approximately three orders of magnitude with respect to CTP without LPL. Even when nodes operates at very low duty cycle, i.e., $l = 2$s, CTP-WUR consumes from around 55x to 280x less energy than CTP LPL, depending on generated traffic.

2) End-to-end latency: The average end-to-end packet latency is shown in Fig. 4(b). Predictably, CTP with no LPL achieves the best performance, incurring average latencies in the range $20 - 30$ms (depending on traffic). Duty-cycling, as expected, significantly affects latency, thus making CTP LPL unfit for time-critical WSN applications. In particular, using CTP LPL with l set to 250ms, 500ms and 1s results in average latencies of 1.44s, 3.70s and 10.12s, respectively. When $l = 2$s, CTP LPL incurs latencies of up to 50s. CTP-WUR, instead, consistently obtains latencies lower than 65ms in all scenarios. This is due to its effective WUR relay mechanism, which reduces the average route length w.r.t. the case in which forwarding of WUR requests is not implemented.

3) Packet delivery ratio: Independently of traffic, CTP-WUR consistently obtains a packet delivery ratio of 99.99% or higher, similar to that of CTP without LPL. The performance of CTP with LPL is sensitive to increasing traffic. In particular, the percentage of packets delivered by CTP LPL falls below 99% when $p = 50\%$ and $l = 2$s, and below 95% when $p = 100\%$ and $l = 2$s.

4) Network lifetime: Our final set of experiments concerns the expected WSN lifetime. We obtain a conservative estimation of the network lifetime by computing the expected time when the first node in the network depletes its energy. In our calculation we assume that the traffic patterns observed during one-hour simulations are representative of the traffic experienced by each node in the long-term. For this reason, the energy spent during the initial setup phase is not considered in lifetime estimations. Simulated nodes are powered by two 1.5V AA alkaline batteries whose capacity is 2500mAh [25]. Results are shown in Fig. 4(c), which reports the estimated number of years for which a network using CTP-WUR and CTP LPL is expected to operate under different fraction of source nodes. The remarkably low energy consumption of

CTP-WUR allows to achieve expected network lifetimes of several decades. In comparison, a network strictly following a nominal duty cycle of 1% in a scenario where no data packet is generated or transmitted would last for less than 2 years. CTP LPL is expected to achieve a network lifetime of less than one year: For example, the expected typical lifetime of a network running CTP LPL with $l = \{250, 1000\}$ is of 112 and 202 days, respectively, when $p = 5\%$.

The impressive lifetime achieved by CTP-WUR can be extended even further by coupling WUR capabilities with a quasi-passive sensing strategy for asynchronous wake-up by events from multiple sensors [26]. In such a scenario, the MCU of a node is only woken up by external interrupt sources, i.e., by the WUR or by a quasi-passive sensor, and it can be kept at the lowest power mode (e.g., LPM4 in case of the TI MSP430 F1611 microcontroller of TelosB motes). Simulations results (not described here) show that this strategy would extend the expected lifetime of a network using CTP-WUR to up to 41 years when the percentage of source nodes is 5%.

VII. Conclusion

The paper shows the advantage of using wake-up radio-based motes in wireless sensor networking for critical applications, i.e., those requiring lifetimes of many years and low data delivery latencies. By combining wake-up capabilities with selective addressing and relaying of WUR requests for longer wake-up ranges, we show that protocols for WSNs routing such as CTP can achieve lifetimes of decades while keeping data latencies comparable to those in networks whose nodes always keep the main radio on. In scenarios where WSNs are used for structural health monitoring, we have shown that WUR-based solutions outperform solutions based on Low Power Listening also with respect to metrics such as energy consumption (reduced by a factor of 54 or more) and packet delivery ratio, which is consistently 99.99% or higher irrespective of data traffic.

Acknowledgment

This work was partly supported by the "Social Museum & Smart Tourism" project (MIUR CTN01_00034_23154) and by the National Science Foundation award "GENIUS: Green sEnsor Networks for aIr qUality Support" (NSF CNS 1143681).

References

[1] D. Porcarelli, D. Spenza, D. Brunelli, A. Cammarano, C. Petrioli, and L. Benini, "Adaptive Rectifier Driven by Power Intake Predictors for Wind Energy Harvesting Sensor Networks," *IEEE Journal of Emerging and Selected Topics in Power Electronics*, vol. 3, no. 2, pp. 471–482, 2014.

[2] 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.

[3] J. Oller, I. Demirkol, J. Casademont, J. Paradells, G. U. Gamm, and L. Reindl, "Performance evaluation and comparative analysis of subcarrier modulation wake-up radio systems for energy-efficient wireless sensor networks," *Sensors*, vol. 14, no. 1, pp. 22–51, 2013.

[4] S. J. Marinkovic and E. M. Popovici, "Nano-power wireless wake-up receiver with serial peripheral interface," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 8, pp. 1641–1647, 2011.

[5] 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, USA, May 2014, pp. 18–25.

[6] O. Gnawali, R. Fonseca, K. Jamieson, M. Kazandjieva, D. Moss, and P. Levis, "CTP: An efficient, robust, and reliable collection tree protocol for wireless sensor networks," *ACM Transactions on Sensor Networks*, vol. 10, no. 1, pp. 16:1–16:49, December 2013.

[7] D. Benedetti, C. Petrioli, and D. Spenza, "GreenCastalia: An Energy-harvesting-enabled Framework for the Castalia Simulator," in *Proceedings of ACM ENSSys 2013*, 2013, pp. 7:1–7:6.

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

[9] V. Jelicic, M. Magno, D. Brunelli, V. Bilas, and L. Benini, "Analytic comparison of wake-up receivers for wsns and benefits over the wake-on radio scheme," in *Proceedings of ACM PM2HW2N 2012*, 2012, pp. 99–106.

[10] I. Demirkol, C. Ersoy, and E. Onur, "Wake-up receivers for wireless sensor networks: Benefits and challenges," *IEEE Wireless Communications*, vol. 16, no. 4, pp. 88–96, August 2009.

[11] C. Schurgers, V. Tsiatsis, and M. B. Srivastava, "STEM: Topology management for energy efficient sensor networks," in *Proceedings of the IEEE Aerospace Conference 2002*, Big Sky, MT, March 9–16 2002, pp. 3:1099–3:1108.

[12] C. Schurgers, V. Tsiatsis, S. Ganeriwal, and M. B. Srivastava, "Topology management for sensor networks: Exploiting latency and density," in *Proceedings of ACM MobiHoc 2002*, Lausanne, Switzerland, June 9–11 2002, pp. 135–145.

[13] T. Stathopoulos, M. Lukac, D. Mclntire, J. Heidemann, D. Estrin, and W. J. 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. J. Marinkovic and E. M. Popovici, "Power efficient networking using a novel wake-up radio," in *Proceedings of PervasiveHealth 2011*, Dublin, Ireland, May 23–26 2011, pp. 139–143.

[15] J. Blanckenstein, J. Klaue, and H. Karl, "Energy efficient clustering using a wake-up receiver," in *Proceedings of European Wireless 2012*, Poznan, Poland, April 18–20 2012, pp. 1–8.

[16] M. Prinn, L. Moore, M. Hayes, and B. O'Flynn, "Comparing low power listening techniques with wake-up receiver technology," in *Proceedings of SMART 2014*, Paris, France, July 20–24 2014.

[17] L. Chen, J. Warner, W. Heinzelman, and I. Demirkol, "MH-REACH-Mote: Supporting Multi-hop Passive Radio Wake-up for Wireless Sensor Network," in *Proceedings of IEEE ICC 2015*, Apr 2015.

[18] P. Levis, N. Patel, D. Culler, and S. Shenker, "Trickle: A self-regulating algorithm for code propagation and maintenance in wireless sensor networks," in *Proceedings of NSDI 2004*, 2004.

[19] L. Chen, S. Cool, H. Ba, W. Heinzelman, I. Demirkol, U. Muncuk, K. R. Chowdhury, and S. Basagni, "Range extension of passive wake-up radio systems through energy harvesting," in *Proceedings of IEEE ICC 2013, Ad Hoc and Sensor Networking Symposium*, Budapest, Hungary, June 9–13 2013, pp. 142–147.

[20] R. Fonseca, O. Gnawali, K. Jamieson, S. Kim, P. Levis, and A. Woo, "TEP 123: The Collection Tree Protocol (CTP)," 2006.

[21] Sensirion AG, "SHT1x Datasheet: Humidity and Temperature Sensor IC," 2011.

[22] U. Colesanti and S. Santini, "The Collection Tree Protocol for the Castalia wireless sensor networks simulator," Department of Computer Science, ETH Zurich, Tech. Rep. 729, June 2011.

[23] D. Moss and P. Levis, "BoX-MACs: Exploiting physical and link layer boundaries in low-power networking," Stanford University, Stanford, CA, Tech. Rep. SING-08-00, 2008.

[24] Crossbow Technology, "TelosB mote platform datasheet," 2004, document Part Number: 6020-0094-01 Rev B.

[25] I. Buchmann, *Batteries in a portable world: A handbook on rechargeable batteries for non-engineers*, 2nd ed. Cadex Electronics, Inc., May 2001.

[26] M. Malinowski, M. Moskwa, M. Feldmeier, M. Laibowitz, and J. A. Paradiso, "CargoNet: A low-cost micropower sensor node exploiting quasi-passive wakeup for adaptive asynchronous monitoring of exceptional events," in *Proceedings of ACM SenSys 2007*, 2007, pp. 145–159.