Network-Harmonized Scheduling for Multi-Application Sensor Networks

Vikram Gupta[†‡], Nuno Pereira[†], Shashank Gaur[†], Eduardo Tovar[†], Ragunathan (Raj) Rajkumar[‡]
[†]CISTER/INESC-TEC Research Center, ISEP, Polytechnic Institute of Porto, Portugal
[‡]Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, USA
vikramg@ece.cmu.edu, nap@isep.ipp.pt, sgaur@isep.ipp.pt, emt@isep.ipp.pt, raj@ece.cmu.edu

***Abstract*—The support for multiple concurrent applications is an important enabler for promoting the use of sensor networks as an infrastructure technology, where multiple users can deploy their applications independently. In such a scenario, different applications on a node may transmit packets at distinct periods, causing the node to change from sleep to active state more often, which negatively impacts the energy consumption of the whole network. In this paper, we propose to batch the transmissions together by defining a *harmonizing period* to align the transmissions from multiple applications at periodic boundaries. This harmonizing period is then leveraged to design a protocol that coordinates the transmissions across nodes and provides real-time guarantees in a multi-hop network. This protocol, which we call *Network-Harmonized Scheduling (NHS)*, takes advantage of the periodicity introduced to assign offsets to nodes at different hop-levels such that collisions are always avoided, and deterministic behavior is enforced. NHS is a light-weight and distributed protocol that does not require any global state-keeping mechanism. We implemented NHS for the Contiki operating system and show that it can achieve comparable duty-cycle to an ideal TDMA approach.**

Keywords*—*Wireless Sensor Networks; Multi-hop Communications; Routing; Medium-Access Control

I. INTRODUCTION

Wireless Sensor Networks (WSN) are evolving into an important infrastructure technology, particularly with applications like smart buildings ([1]), data-center monitoring ([2], [3]), industrial sensing ([4]) and structural-health monitoring ([5]). For a more widespread adoption of WSN in our everyday life, it is crucial to allow several independent users to deploy their applications for accomplishing diverse goals, thus converting the WSN into an infrastructure where multiple applications are deployed. Several hardware developments also support this infrastructure view of WSN. Platforms based in newer System-on-Chip (SoC) solutions (like the Atmel ATMEGA 128RFA1 [6]) provide better performance (for processing and wireless communications), and consume a similar or even lower amount of power than the previous generations of WSN hardware. Additionally, an increasingly large number of different kinds of sensors can be integrated in a single WSN at a reasonable cost to help users to employ one or more sensors for creating independent applications.

The key motivation behind supporting multiple applications is that a sensor networking infrastructure can be used more efficiently both in terms of resource-usage and cost effectiveness. Many solutions have been provided in the past to support and optimize multiple applications from several perspectives such as in-network programming [7], strategic application deployment [8], redundancy elimination [9] and operating systems [10], [11], [12]. However, there is still scope for optimizing the network behavior when multiple applications are executed on sensor nodes. It is often the case that applications release packets independently in the network, which can lead to excess energy consumption due to factors like increase in the number of packets, more frequent radio-switching and extra contention at the Medium Access Control (MAC) layer. The energy consumed in transmitting a packet from a source node to a destination node depends on many aspects, and with common MAC approaches, a packet may undergo contention at several points in a multi-hop network. This jeopardizes the deterministic behavior of the network and makes it very difficult to provide any timing guarantees. In this work, we present the *Network-Harmonized Scheduling (NHS)* approach, in which radio transmissions from multiple applications are coordinated across a multi-hop network by *harmonizing* them around periodic boundaries, while obviating the need for explicit MAC and routing layers. NHS is a simple and effective approach that is inspired from Rate-Harmonized Scheduling (RHS) [13] and applied to the context of multi-hop networking. By using NHS, it is possible to provide real-time performance guarantees in a multihop wireless network without requiring any central coordination.

Even if applications release one or more packets in a periodic manner, the overall packet transmission by a sensor node may no longer be periodic because of the mismatching periods of different applications. In such a case, the total number of packets released by a sensor node grows proportionally with respect to the number of deployed applications. All the packets may suffer contention at different hops in the network if the underlying MAC layer is based on a carrier-sense mechanism. Therefore, multiple applications releasing packets independently may increase the overall resource consumption in the network. Moreover, the latency suffered by packets in a dense multi-hop network may become prohibitively large and non-deterministic. To overcome these issues, NHS aligns packet releases from different applications around periodic boundaries at each node, and leverages this periodic behavior to harmonize the transmissions at the network level.

NHS includes a light-weight protocol that groups periodic batched transmissions from different devices, such that the nodes can turn on their radios when other devices transmit. We first describe the protocol assuming that all nodes lie in a single broadcast domain. We further develop the protocol to support multi-hop scenarios, where we harmonize packet transmissions in a periodic manner without any global state maintenance. One of the major advantages of this protocol is that it includes an implicit link-layer mechanism, and from its multi-hop operation, it can be inferred that dedicated route maintenance is also not required. Moreover, the protocol provides deterministic bounds on the end-to-end latency for packet delivery, and design parameters can be chosen such that the packet deadlines can be met for real-time applications. This characteristic emphasizes the usefulness of NHS in cyber-physical applications.

In this paper, we analyze the performance of NHS with respect to parameters such as end-to-end latency, maximum utilization of the channel and average power consumption. The implementation of NHS is simple ($\sim$ 400 lines of code), and does not require modification of application semantics. Applications only need to declare their period of operation and the maximum number of packets they may transmit in every round. Each node only maintains information about its neighbors, but can still achieve a performance (in radio duty-cycle) similar to that of Time-Division Multiple Access (TDMA). NHS requires only a few cycles to converge to a stable schedule, and does not need additional information exchange if the network remains static. Our approach can also optionally provide a contention slot for supporting mobile and intermittently connected devices.

The main contributions of this paper can be summarized as follows:

- A harmonizing task is designed for sensor nodes that *batches* together the transmissions from multiple applications by ensuring that packets are released into the network only at periodic boundaries.
- A protocol is proposed that coordinates packet transmissions in a network around periodic boundaries. For multi-hop networks, this protocol works in a distributed manner and pipelines the transmissions from successive hops to make sure that no collisions occur at any node.
- NHS is implemented[1] for the Contiki operating system, and we show through experiments that the protocol is suitable for real-time applications by providing deterministic bounds on parameters such as the end-to-end latency, channel utilization and radio duty-cycle.

The remainder of this paper is organized as follows. We describe the state-of-the-art in the next section, and contrast NHS with other research-works that also aim at optimizing the network operation. In Section III, we outline the assumptions made in our approach and propose the model of the applications and the sensor network. The process of batching the packet transmissions using RHS (from which NHS is inspired) is explained in Section IV. The protocol to harmonize data from multiple nodes in a single broadcast domain is proposed in Section V, which is followed by the description of the NHS approach for generic multi-hop networks. In Section VII we discuss various design parameters, Section VIII provides the details about the implementation of NHS for Contiki and Section IX contains the results from experimental evaluation. The paper then concludes with a description of future work.

II. RELATED WORK

Reducing the radio-usage for data delivery in wireless sensor networks is a well-researched area. The solutions cover various aspects ranging from link-layer protocols to network flooding and distributed TDMA solutions. Network resource-consumption in case of multiple applications is a new challenge and not many works directly address this. Low-Power Wireless Bus (LWB) [15] is a flooding mechanism where data from one or more *initiators* is flooded across the network. Data from each source is received by every other device, so it emulates a bus-like behavior even in the case of multi-hop networks. LWB is based on Glossy [16], a flooding and synchronization approach that leverages the property that multiple near-simultaneous and identical receptions at a node do not interfere, and these packets can be demodulated with a high degree of success. However, the flooding of packets, even if they are directed to a small subset of nodes, may lead to a high degree of redundant transmissions. Moreover, in the case of multiple applications, the resulting overhead can become prohibitively large.

There are many approaches that aim at reducing the amount of time that a node has the radio in the ON state by defining a periodic wake-up scheme. Typically, these approaches are distinguished between synchronous and asynchronous. In synchronous approaches, nodes agree on a common sleep/wakeup schedule [17], [18] to save energy. However, in such schemes, nodes may be forced to maintain several such schedules corresponding to each of their neighbors. Asynchronous approaches are based on channel polling [19], [20]. In these protocols, nodes periodically wake up and try to sense the channel, and if the channel is active, then the nodes stay awake to receive the packet transmissions.

Similar to our goal of batching data from multiple applications, the Unified Broadcast (UB) [21] approach transmits data from various services running on a sensor node. In Unified Broadcast, data from various services is transmitted when the number of accumulated packets reaches a certain threshold. This approach is limited to sending data together from multiple services, and therefore it is only valid for broadcast messages. In the case of multiple applications requiring many-to-one communication, an additional protocol is required. This work on unified broadcasts has shown experimentally that it is still possible to preserve correctness of a set of representative WSN protocols (such as FTSP [22], Trickle [23] or CTP [24]) when packets are delayed to minimize network resource usage. In the case of Unified Broadcast, the periods of the applications are implicitly detected when a second packet of the same protocol is requested to be transmitted. As we will see, in this work, we explicitly take the period of the application into account to derive the harmonizing period of communication.

The extreme of sleep/wakeup schemes is Time Division Multiple Access (TDMA), where nodes only wakeup at the scheduled transmit/receive time instants, at a cost of tight synchronization and no flexibility to changes. TDMA for multi-hop networks is typically achieved using 2-distance graph-coloring algorithms. Such protocols require much more information about the network topology, and typically a central coordinator, as it is done in RT-Link [25] or Distributed TDMA [26]. In contrast, our approach aims to achieve TDMA-like efficiency without global state-maintenance. An interesting

---

[1] The implementation of the Network Harmonized Scheduling protocol can be obtained at [14]

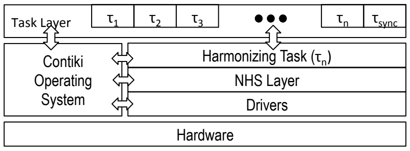


Fig. 1: Architecture of Network-Harmonized Scheduling (NHS) protocol

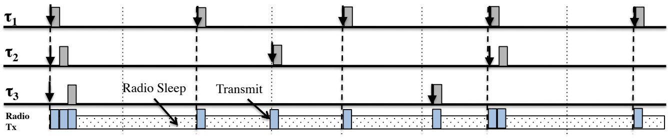
perspective to create a TDMA schedule and thus minimize the probability of collision such that nodes transmit independently in non-overlapping time slots is proposed in Desync [27]. The Desync approach is conceptually closest to ours and forms a round robin TDMA schedule for reducing the power consumption. This scheme, unlike NHS, is limited to single broadcast domain, and it is not easily extendable to multi-hop communication scenarios.

III. MODEL AND ASSUMPTIONS

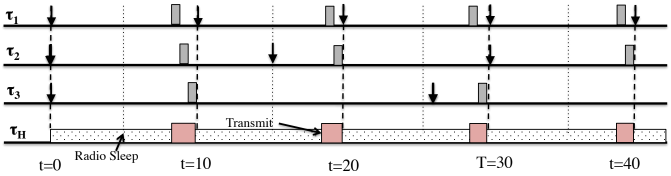
In this work, we assume that several network-wide applications execute concurrently on a sensor network, and each application has a corresponding node-level task that releases periodic jobs on each sensor node. The set of all the tasks on a sensor node is represented by Γ. We assume that there are n tasks in Γ, and the i^{th} task is denoted by τ_i, where $i \in \{1, 2, ..., n\}$. The tasks execute on top of the Network Harmonized Scheduling layer as shown in Figure 1. Every task releases an infinite number of jobs with a period of T_i. Without the loss of generality, we sort the tasks in an ascending order of their periods, and then assign serial indexes to the tasks. Hence, it follows that $T_1 \leq T_2 \leq \ldots T_n$. Every job may sample the sensors, process the data from sensors or incoming packets, and release one or more packets towards a destination node. Assuming that the maximum number of packets a job of the i^{th} task can transmit is p_i, the time consumed by the packets for transmission is $w_i = p_i \cdot \delta$, where δ is the duration corresponding to one packet. As an example, the value of δ for IEEE 802.15.4 packets of size 128 bytes transmitted at 256 Kbps is $4ms$.

One of the tasks deployed on each sensor node can implement a simple clock-synchronization service, τ_{sync}, that executes periodically. The management and exchange of MAC-level time-stamps and correction of clock-drift is the responsibility of this clock-synchronization task which sits above the Network-Harmonized Scheduling protocol layer. As we will see in the next section, the harmonizing period (period of network operation) is chosen to be at-least as small as the period of the most frequent task, allowing the synchronization task to comfortably operate at the desired frequency. As it will be evident from the description of the protocol, each node only maintains a schedule according to its immediate neighbors, NHS requires that a child node is synchronized with its parent.

The nodes are assumed to have unique id's, and the number of nodes in the network is assumed to be bounded by N, and a multi-hop operation may be required to communicate from a root node to another node in the network, or vice-versa. Let h_{max} be the maximum number of hops required for connecting a root to all other nodes in the network, and Q denotes the maximum degree of connectivity in the network.



(a) Packet transmission from different tasks, if scheduled using Rate-Monotonic Scheduling. The task with a shorter period has higher priority. The number of independent packet transmissions over a time-window is the sum of number of packets from each task.



(b) The transmission schedule after harmonizing the transmissions. The harmonizing task (with a period of $T_H = T_1 = 10$ time-units) makes sure that the packets are dispatched in batches, never more often than the harmonizing period.

Fig. 2: A task set with three tasks τ_1, τ_2, τ_3, with periods of 10, 15 and 26 time-units respectively, scheduled by Rate-Monotonic Scheduling. Block arrows show the time-instants when the packets are released by different jobs of the tasks.

IV. RATE HARMONIZED SCHEDULING FOR PACKETS

Rate-harmonized scheduling (RHS) [13] is a policy that optimizes the execution of tasks on a uni-processor system such that the job executions of all the tasks are aligned near the period boundaries of the task with shortest period (most-frequent task). RHS saves power by removing inefficient switching in processor states, namely *active, idle,* and *sleep*, based on the observation that the power consumption in the sleep state is orders of magnitude lower than that in the idle state, but going to and coming out of the sleep state takes longer time. RHS makes sure that the task executions are harmonized and aligned in time, and the processor can optimally go to deep-sleep states more often and for longer time-spans. We adapt RHS in this work to align packet transmissions by different periodic applications on a sensor node, such that that overhead of radio switching can be avoided, and the packets are released into the network in a periodic manner.

With multiple tasks releasing packets at every T_i time units, the transmission pattern can be irregular as shown with an example in Figure 2a. The packets in the example are transmitted using the well-known Rate-Monotonic Scheduling (RMS) approach. On the other hand, the packets from various tasks are batched together with harmonizing period $T_H = T_1$ as shown in Figure 2b. RHS is implemented using a simple queueing mechanism, where every job of all the tasks submits packets to a harmonizing task, τ_H, instead of directly copying them into the radio-buffer. τ_H then transmits all the packets in its queue with a period of operation equal to a *harmonizing period*, T_H. As the packets from a node are transmitted in a contiguous manner (back-to-back) as a *batch*, the number of radio switchings is reduced significantly.

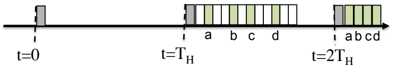


Fig. 3: Aligning packet transmissions in a broadcast domain around periodic boundaries

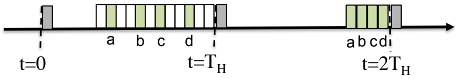


Fig. 4: Aligning packet transmissions before the scheduled transmission by the root node, optimized for collection of data from all the nodes.

An important distinction in scheduling of packets is that preemptions are not possible once the packets transmission begins, whereas, most task-scheduling approaches on a processor, however, allow preemption.

Harmonizing using the above mechanism converts intermittent packet transmissions from a sensor node into periodic batch-releases, and this periodicity is a fundamental building-block for further optimizations as described in the subsequent sections.

V. SINGLE BROADCAST DOMAIN

Once the packet transmissions from multiple tasks are batched around the harmonizing period, we can design a distributed and online protocol to align packets transmitted by multiple nodes in a single broadcast domain. Let us assume that the nodes have unique id's and, for simplicity, the root (or a cluster head) is already known among the nodes. The goal is to create a scheme where the transmissions from all the nodes can be gathered at the root in a periodically regular manner. The protocol works as follows. A simplified operation with a root node and 4 nodes with id's a, b, c and d is shown in Figure 3.

- Initially, all the nodes turn on their radios and listen to any incoming packets. The root transmits a beacon to initiate the protocol, with its node-id and the harmonizing period.
- All the nodes listen to this beacon, and take note of the root-id and the period of transmission. The nodes locally create a schedule by assigning transmission slots as a monotonic function of node id, such that each slot of transmission is unique to each node. This implies that the nodes with lower id transmit earlier, and those with higher id transmit later.
- Until the next period boundary, all the nodes listen to the medium after transmitting a chosen slot. As the node id's are assumed to be unique, it is guaranteed that there is no collision in any slot in the network.
- All the nodes listen to the medium during this period and they learn about the other nodes in the broadcast domain, and their respective slots. It is possible that there may be several empty slots in the schedule, and the nodes can then independently compress the schedule by removing the empty slots in the next cycle as shown in the second cycle of Figure 3 and Figure 4.
- From the next round onwards, the packets are transmitted in a compressed schedule such that the root only wakes-up periodically for a duration equal to the total time required by all the slots. All the other nodes only wakeup close to their slot in the schedule to transmit their data.

In this example, the protocol is designed for nodes to choose slots after the scheduled transmission by the root node in the next cycle. However, the protocol can also be modified such that the nodes transmit *before* the next transmission by the root as shown in Figure 4. This is more beneficial in data-collection oriented applications, because the root can receive the data and then react to the data in the same cycle. The difference in these two approaches is more pronounced in the multi-hop scenario (discussed in the next section), where the first approach is better suited for flooding and the second for many-to-one communication.

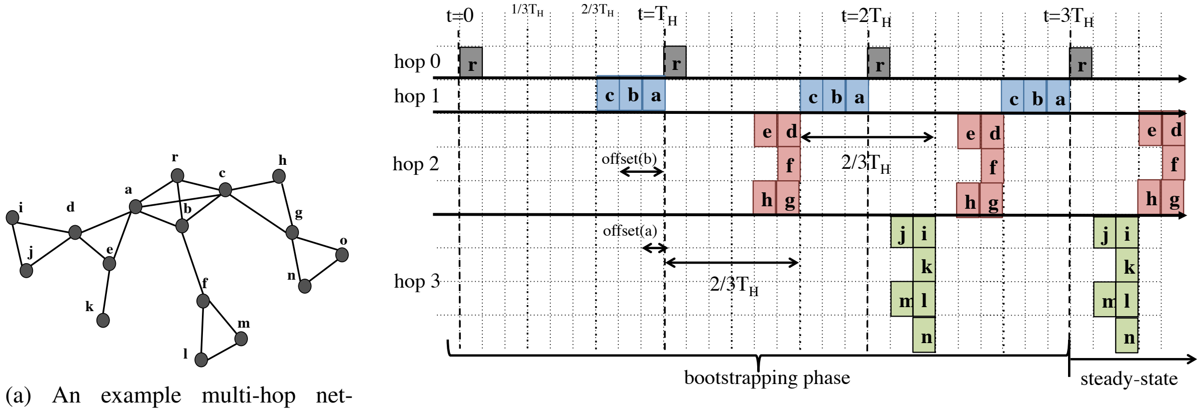
This protocol helps to remove the overhead of contention and random back-off as in the case of carrier-sense MAC protocols, and achieves TDMA-like timing efficiency without the need to maintain a global schedule. The width of each slot, denoted by σ, has to be large enough to accommodate batched packets, but short enough so that a Harmonizing Period can accommodate packets from all the nodes; that is:

$$\delta \sum_{i=1}^{n} p_i \leq \sigma \leq \frac{T_H}{N} \tag{1}$$

VI. HARMONIZATION IN A MULTI-HOP NETWORK

We now extend the above approach to a multi-hop scenario where all the nodes have to send their data to a sink in the network. The protocol described in the previous section is easily applicable to multi-hop topologies with minor modifications, so that the possibility of collisions and packet-drops due to the hidden-terminal problem can be eliminated. The root initiates the protocol by broadcasting a *trigger* beacon, which is received by the neighbors of the root. Similar to the case of a single broadcast domain, the children nodes listen to the beacon, then choose slots as a function of their id's and then compress the schedule. The nodes at the second hop-level should not transmit until the schedule has been compressed, because the nodes at this level cannot listen to all the transmissions at the next level closer to the root. Each node in the network has information only about its 1-hop neighbors and its peers (siblings under the same parent).

The main goal of this protocol is to enable scheduled transmissions in a distributed manner, without requiring global knowledge of the network-topology and without explicit time-synchronization. The working principle behind the protocol is to ensure more than 2-hop distance in simultaneous transmissions in the network. Assigning slots to transmissions in a TDMA-based network is typically accomplished by applying distance-two vertex coloring graph. To maximize the throughput, the problem is equivalent to choosing the minimum number of colors [28]. In our approach, we achieve the required 2-hop distance by dividing each harmonizing period into three equal slices and nodes at consecutive hop-levels transmit only



(a) An example multi-hop network, with 15 nodes, and a root-node (r). Solid lines depict bi-directional wireless links.

(b) Timeline of transmissions from nodes at different hops, showing the working of the NHS protocol. The listening schedule is not explicitly shown, but the nodes can listen when the children and the parent nodes transmit.

Fig. 5: An example of a multi-hop topology, and the corresponding timeline for illustrating the NHS protocol with in a network of thirteen nodes.

in non-overlapping slices. The number of slices can be chosen to be greater or equal to 3, and we call it *cadence-factor,* ω.

The nodes at each hop choose slots to avoid collision using the approach described in Section V for the single broadcast domain case. In addition to the data to be transmitted, each node transmits a *NHS-tuple* as a part of the packet header. The NHS-tuple, denoted by λ, consists of η, the number of hops the transmitter is from the root, and ϕ, its offset in terms of the number of slots from the boundary of its section:

$$\lambda \equiv < \eta, \phi >$$

There can be cases of collisions in a multihop scenario due to hidden-terminal problems, because some nodes may occupy the same slot as a hidden terminal when the schedule is being compressed. This is avoided by ensuring that the root (or a parent) sends a message containing the list of successful receptions during the bootstrapping phase. This way, every node becomes aware of other hidden-terminals, and avoids choosing overlapping slots. If a node at receives a packet from more than one parent node in the bootstrapping phase, then it chooses a parent which is closest to it, by considering the signal strength of the received packets. This way the network topology is generated by setting up parent-child links. If the topology changes, and a node does not receive a packet from its parent for a preset number of cycles, it choses a different parent within its neighborhood. By design, NHS is more suitable and reliable for static and nomadic sensor deployments than mobile and intermittent networks.

To illustrate the operation of this protocol, let us consider the example topology shown in Figure 5a. We assume that data collection is the more important goal of the network, and the protocol is programmed such that the nodes transmit *before* the transmission of the parent node in the next cycle, similar to the operation shown in Figure 4. The network consists of a root node r and several other nodes, namely $\{a, b, \ldots, o\}$. At the start, all the nodes turn on their radios and wait to receive packets. The root node r broadcasts a *trigger* beacon at $t = 0$, and nodes a, b and c receive this beacon. A simplified timeline of operation of NHS corresponding to this example in shown in Figure 5b, and the labeled slots imply transmission by a node. Nodes a, b and c transmit at the period boundary before the slot for the root. The transmissions are aligned around this boundary as explained in the previous section.

For the sake of simplicity, the step of choosing non-overlapping slots is deliberately omitted in this example by assuming that the node id's are consecutive within a broadcast domain.

At the next period boundary, nodes lying within the first hop create a compressed schedule, and the transmissions from a, b and c are received by their respective children. Based on the λ values transmitted by them, their children nodes now also participate in the same protocol to find a local compressed schedule. The transmissions by the children nodes in the next cycle are carried out with respect to a future relative reference time T_s, which is estimated as follows:

$$T_s = \frac{2}{3} T_H + \phi$$

The offset values are shown for nodes a and b in Figure 5b. The factor of $\frac{2}{3}$ is the key in dividing the harmonizing period into three sections, and making sure that the nodes in successive hops transmit *earlier* than their parents by one-third of the period.

Similarly, the children of the nodes in the second-hop, i.e. $\{i, j, ...o\}$ also transmit according to their respective T_s values and their transmissions do not overlap with any transmission in the first or the second hop. Using T_s, the nodes in the fourth hop would have chosen to transmit simultaneously with the transmission from the first hop, and so on. Simultaneous transmissions with NHS are guaranteed to have a hop-distance of three, making sure that no collision occurs at any receiving node. Designing a protocol with a *cadence-factor,* ω, of less than three can result in collisions at the receivers. On the other hand, if ω is chosen to be greater than 3, data from deeper hops can reach the root within one harmonizing period. The choice of ω provides a tradeoff between the latency suffered by a packet to reach from a leaf-node to the root and the maximum number of children a node can have. In general, the reference time T_s can be calculated with respect to ω as follows:

$$T_s = (\frac{\omega - 1}{\omega})T_H + \phi \tag{2}$$

In the steady state, the network operation is harmonized with respect to the cadence-factor (ω) and the harmonizing period (T_H). Transmissions by any node in the network is conducted only once every harmonizing period, but different hops are offset in a cyclic manner to avoid collisions. This approach improves the end-to-end latency in a fashion similar to that of pipelining. The nodes at the j^{th} hop now listen only in the windows where the nodes in the $(j+1)^{th}$ or $(j-1)^{th}$ hop are going to transmit. By listening to the nodes in the previous hop-level, the protocol ensures that the communication from the root node to leaf-nodes is also possible. It can be observed that the data from ω number of hops can reach the root within one cycle, once the protocol starts to operate in the steady state. The packets in the opposite direction can reach the root in a number of cycles equal to the number of hops. The network operation can be configured to be more responsive in either direction. If data collection is the main goal, then the described design is appropriate, otherwise the children nodes transmit just *after* the next transmission by their parent to enable fast delivery of data from the root to the leave in case of flooding applications. This asymmetric operation is practical for most data-collection applications, the reverse data channel can be used for network maintenance, acknowledgements, and other similar functions.

The NHS protocol is distributed by design and maintains very little state. The primary benefit from this approach is that the transmissions are harmonized around periodic boundaries, and packets do not suffer from contention. The radios on the nodes only need to be turned on in a periodic manner for a small time-span, which considerably reduces the radio switching overhead. The proposed protocol does not aim to achieve a high throughput, since the goal is not to maximize the number of possible simultaneous transmissions in the network. That optimization problem requires global knowledge and has been solved in the past using graph-coloring approaches [25].

VII. REAL-TIME PERFORMANCE

So far, we have described the operation of the NHS protocol based on parameters such as harmonizing period (T_H), cadence-factor (ω) and node degree (Q). The choice of these parameters can directly or indirectly impact the resource consumption and real-time characteristics of the network. The following sections introduce an analysis that allows to reason about the trade-offs between the end-to-end latency and parameters: T_H, ω, and Q.

Observing that NHS parameters can be selected such that packets are always delivered before the next transmission, enables pre-runtime guarantees on end-to-end packet delivery deadlines, as presented in Section VII-B.

A. End-to-End Latency

The worst-case end-to-end latency that may be suffered by a packet from the time it is released by an application to the time it is delivered at the root node consists of the delays occurred due to the packet-batching and the latency in the multi-hop network. Based on the design of our approach for batching the packets from multiple applications, worst-case delay a packet can suffer at a node is equal to the harmonizing period, T_H. The worst-case happens when the packet is released by the application just after the end of a T_H cycle, and it can only be released into the network near the end of the next period boundary, as evident from Figure 2b.

At the j^{th} hop, the maximum number of slots that can be occupied for transmissions is equal to the maximum number of children a node can have. Also, the worst-case network latency suffered by a packet from a node in the j^{th} hop to the root is

$$L_j = \frac{(j-1)T_H}{\omega} + Q\sigma \leq \frac{jT_H}{\omega} \tag{3}$$

Each node turns on its radio to listen to its children and its parent and then to forward the data in the next slot. The total worst-case latency (including the delay at the node) suffered by a packet released by a node in the j^{th} hop-level can be given as:

$$L_{total} \leq T_H + \lceil \frac{j}{\omega} \rceil T_H$$

We define *delivery-factor* φ, as

$$\varphi = (1 + \lceil \frac{h_{max}}{\omega} \rceil)$$

Hence, the worst-case latency for a packet from the h_{max} hop level is:

$$L_{max} = \varphi \cdot T_H \tag{4}$$

If the harmonizing period is chosen to be less than half of the period of the most frequent task (i.e., $T_H < T_1/2$), and if ω can be chosen to be greater than the maximum number of hops in the network, NHS can guarantee that a packet from any node in the network can always be delivered to the root before the next packet is released by that node, and this determinism allows NHS to provide real-time performance guarantees to applications as we will see in the next subsection.

B. End-to-End Deadlines

The Network-Harmonized Scheduling protocol can provide deterministic end-to-end latency that can be leveraged to meet the packet delivery deadlines specified by the applications. We assume that the applications specify relative deadlines given by D_i, where $i = 1, 2, ...n$, considered from the point of release of the packet. For ensuring that all the packets meet their delivery deadlines, the maximum latency should be less than the minimum deadline, such that:

$$min(D_1, D_2, ...D_n) \geq L_{max} \tag{5}$$

which implies:

$$D_{min} \geq \varphi \cdot T_H \tag{6}$$

The above equation shows that a packet originating within ω number of hops can reach the root node within two cycles of the harmonizing period. Conversely, the harmonizing period can be selected such that the deadlines are always met.

TABLE I: Various fields in the NHS packet header.

Field	Description
ID	ID of the transmitter
Slot	Offset (ϕ), transmitter's slot with respect to its parent
Parent	ID of transmitter's parent
Hopcnt	Hop count (η), Transmitter's Hop level
Cycle	Current cycle in number of harmonizing periods
N_child	Number of transmitter's children
Child_k	ID of the k^{th} child
NHS Data	Data from all the deployed applications, delineated by application ID and length.

The choice of the *cadence-factor*, ω, is also important in determining the latency suffered by the packets over a multi-hop path, as given by (3). Increase in ω also results in narrowing the offset in the transmissions from successive hops; hence, it may not be possible to increase ω beyond the point that the transmissions from nodes at a hop may not fit inside a time-window of $t_\omega = T_H/\omega$. The number of slots in a time-window of t_ω can be calculated as:

$$n_\omega = t_\omega/\sigma = \frac{T_H}{\sigma\omega}$$

Assuming that the maximum node degree is Q, the number of slots in each t_ω should at least be equal to Q to accommodate all transmissions from all the nodes:

$$n_\omega \geq Q \Rightarrow \frac{T_H}{\sigma\omega} \geq Q \tag{7}$$

By eliminating T_H from (6) and (7), we can deduce that:

$$\frac{D_{min}}{\varphi} \geq Q\sigma\omega \tag{8}$$

We can now find a suitable value for the maximum number of children a node can have such that the minimum end-to-end deadline is met.

$$Q \leq \frac{D_{min}}{\varphi\sigma\omega} \tag{9}$$

As the packets are batched with the same harmonizing period, we can say that if the minimum deadline is met, the larger deadlines will also be met. Equation 9 provides an upper limit on the network size, such that the real-time requirements of applications are met.

VIII. IMPLEMENTATION

We implemented the Network Harmonized Scheduling for the Contiki operating system, such that it replaces the Radio Duty-Cycling (RDC) and the Medium Access Control (MAC) layers of the Contiki network-stack. The core of the NHS implementation is a simple state-machine as shown in Figure 7. The state machine is the same for all the nodes except the root node. The protocol starts with all the nodes in the network waiting to receive a packet with their radios on. Whenever a node receives its first packet, it registers the id of the incoming packet as its parent node, and continues to listen for incoming packets for another harmonizing period so that it can listen to its peers. The packet header in the NHS implementation is shown in Figure 6, which contains several fields as described in Table I.

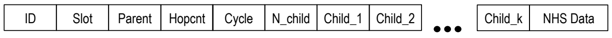


Fig. 6: The packet header in NHS implementation

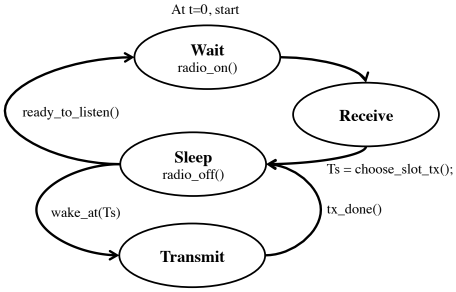


Fig. 7: State machine showing the core of implementation of the NHS protocol at each node.

If a node receives a packet from a parent, it calculates the next relative time-reference to transmit based on the received λ-tuple using the function `choose_slot_tx()`. This function calculates the reference time with the help of Equation 2, and conducts the slot selection algorithm among the neighbors of this node as previously described in Section V. Then, the node goes to the sleep state, and wakes up at the time-reference Ts to transmit its data. Once the transmission is finished, the node goes to sleep immediately. The time instants for the nodes to wake up to listen is calculated using the function `ready_to_listen()`, that wakes up the nodes only when either parent hop or children hop transmit. The implementation of the Network Harmonized Scheduling protocol for Contiki can be obtained at [14].

IX. EXPERIMENTAL EVALUATION

We implemented the Network-Harmonized Scheduling protocol for the Contiki [10] operating system for TMote Sky sensor nodes. In order to evaluate the performance of NHS with respect to energy efficiency, we compared the average radio duty-cycle achieved with NHS against an ideal TDMA approach. For estimating the duty-cycle of each node with the ideal TDMA, we assume that each node only turns on its radio at the exact instants to listen to its parent, its children and to transmit in its allocated slot. All other overheads of radio-switching and clock-synchronization are assumed to be negligible in the ideal TDMA estimation. The duty-cycle for each node in case of ideal TDMA for a given topology is calculated offline by considering the network graph.

We measured the average radio duty-cycle per node for two different topologies, with varying values of harmonizing period. Topology 1 is a linear-topology as shown in Figure 8, where 8 nodes are arranged in a linear topology with a maximum number of hops equal to 6. Topology 2 is a multi-hop tree topology shown in Figure 9. The values for duty-cycle are obtained after running the network for a duration of 600 harmonizing periods. The experiments are conducted for a large number of cycles to amortize the radio on-time in the bootstrapping phase, as the radio remains on continuously for the first few cycles. The results of the experiment are shown in Figure 10 and Figure 11. It can be observed from the results that the average duty-cycle in NHS is within 15% as compared to the ideal TDMA case. This overhead appears partly because the radio remains on for the first few cycles to identify the parents, peers and children, and partly because of the time consumed in the switching of the radio. It can be observed that the relative overhead compared to the ideal is larger for small values of harmonizing periods, which is about 30% for the Topology 1 and about 12% for Topology 2, with a period of 1 sec. The overhead is larger for the linear topology because each node has only one parent and one child and the radio switching overhead accrues for each reception. However, for the tree topology a node receives several packets from its children back-to-back, thus reducing the switching overhead.

A. Real-Time Performance Evaluation

We also simulated the operation of the Network-Harmonized Scheduling protocol to evaluate its performance and validate the analytical results obtained in the previous section. The network topology for simulation consists of a set of nodes spread randomly with a uniform distribution in a 2-dimensional field of size $100m \times 100m$. A given number of nodes, N are spread uniformly over the field. The number of nodes in each broadcast domain are automatically selected during the procession of the protocol. Corresponding to the number of nodes in the broadcast domain at a given hop level, successive hops are generated based on the nodes that lie within the radio transmission range. The simulation is a time-driven execution, where each node is autonomously assigned a slot to transmit.

In this evaluation, we configured the transmission power of the nodes such that a receiver within a certain radius (in meters) can receive the packets successfully with a probability of 100%. Various external factors such as interference from other devices and multi-path can result in unexpected packet-loss, but our evaluation focuses on the performance limits of the protocol and its overall behavior under perfect packet reception. The simulation environment is chosen to highlight the advantages of the NHS protocol with respect to its real-time characteristics, and impact of network capacity on deadlines.

We observed the impact of the network size and selection of the harmonizing period on the real-time behavior of our protocol. The results are shown in Figure 12 and Figure 13. Firstly, we measured the effect of the choice of harmonizing period on the number of packets that miss the minimum deadline. The deadline is chosen corresponding to the harmonizing period as given by (6). The harmonizing period is defined according to the minimum possible period (and deadline) to show the performance limits of the NHS protocol. Longer deadlines are bound to provide better performance in terms of deadline misses by allowing the choice of larger T_H. Smaller harmonizing periods will require more packets to be transmitted within a smaller window, thus more deadlines will be missed. In Figure 12, we show the decrease in deadline misses as the harmonizing period increases, for different values of the radio-range. The total number of nodes, N was fixed to 100 for these experiments, and we assume that each node releases a packet every harmonizing period. Note that, with larger radio range, more nodes are covered within one broadcast domain and more slots are required at each hop. Hence, a short harmonizing period may not be sufficient to accommodate all the transmissions. For example, a period of $1000ms$ or larger is enough to guarantee that all network deadlines are met, if the radius of coverage by each node is equal to or less than $30m$.

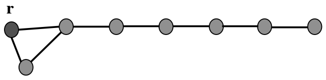


Fig. 8: An example linear topology with 8 nodes.

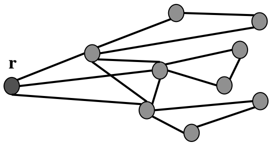


Fig. 9: An example tree-like topology with 10 nodes.

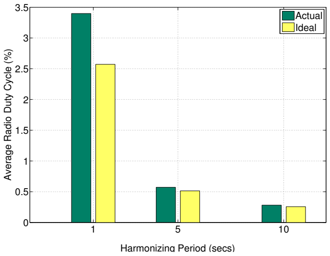


Fig. 10: Average radio duty-cycle for the linear topology in Figure 8.

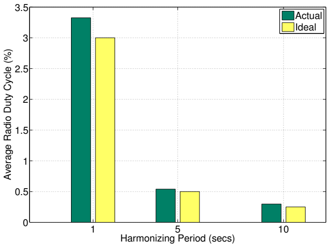


Fig. 11: Average radio duty-cycle for 10 nodes in a multi-hop graph shown in Figure 9.

The following experiment studied the impact on deadline misses as the number of nodes in the network increases, for different radio ranges, and a given harmonizing period of $1000ms$. As the number of nodes increases, more packets need to be transmitted in each slot, thus the number of deadline misses also increases, as shown in Figure 13. This figure also shows that, with an harmonizing period of $1000ms$, up to 100 nodes are supported with no deadline misses.

Energy consumption of a sensor node is directly dependent on the duty-cycle of the radio, and we measured the average duty-cycle over all the nodes for different values of the harmonizing period. If the harmonizing period can be made large while meeting the deadlines, then increasing the harmonizing period improves the duty-cycle. Note that the pre-runtime guarantees offered by NHS (presented in section VII) enable us check if our selection of the harmonizing period will allow us to meet all deadlines. One of the key advantages of NHS lies in the fact that the nodes are autonomously assigned slots in the bootstrapping phase, and then the nodes do not need to listen to activity other than in the slots of its neighbors. A node only transmits in one slot per harmonizing period, and keeps the radio on during the transmission by its neighbors, which is always less than Q number of slots. The results for average radio duty-cycle over all the nodes in the network with varying network size are shown in Figure 14. The values are averaged over 20 iterations, and the error bars show the range of deviation in the duty-cycle. With a network size of 100 nodes, NHS can achieve about 0.50% duty-cycle at a period of 60 secs. The average duty-cycle remains below 2% for periods greater than 20 secs with a network size up to 200 nodes.

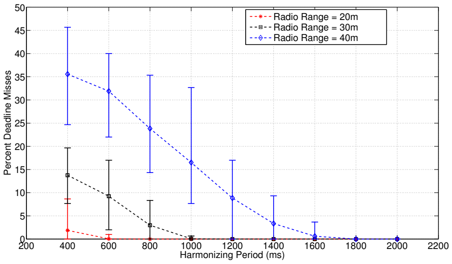


Fig. 12: Impact on deadline misses with increase in T_H with different radio ranges, averaged over 20 iterations

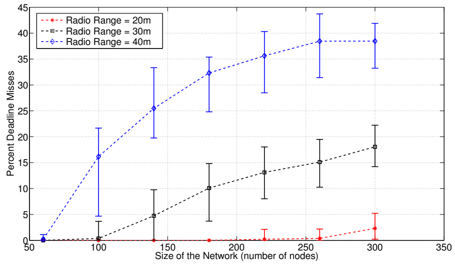


Fig. 13: Average deadline misses over 20 iterations with respect to the size of the network. The error bars are also shown.

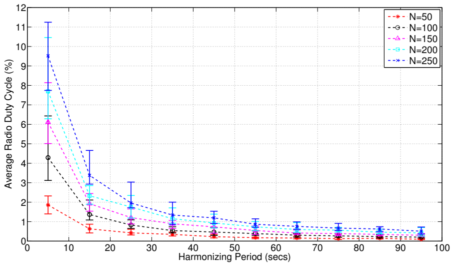


Fig. 14: Average radio-duty cycle over all the nodes after 20 iterations with the increase in the harmonizing period.

X. LIMITATIONS AND FUTURE WORK

Despite its simple implementation and distributed design, there are certain limitations that may hamper the application of Network-Harmonized Scheduling (NHS) in some scenarios. NHS provides static schedules to the nodes in the network, hence the support for mobile nodes is not available in the described version of the protocol. It is, however, possible to provide contention slots at each hop to allow mobile or nomadic nodes to participate in the network.

As future work, we would like to further reduce the energy consumption during the bootstrapping phase by employing Carrier Sense techniques with Low-Power Listening (LPL) instead of keeping the radios ON constantly until the reception of a packet. Finally, we would like to modify the protocol to divide the harmonizing period into sections of exponentially decreasing widths instead of fixed widths, since the packets get accumulated in an exponential manner while forwarding the data from leaf nodes to the root node.

XI. CONCLUSIONS

We proposed the Network Harmonized Scheduling (NHS) protocol for distributed coordination of packet transmissions in a multi-hop network. The concept of NHS is inspired from the Rate-Harmonized Scheduling approach where the executions of various tasks is aligned around a period boundary for saving power by enabling the processor to go into deep sleep states more often. We use a similar approach to *batch* packets from multiple applications together around periodic boundaries, which makes the transmissions periodic. This periodic behavior is leveraged to create a network protocol that obviates the need for an explicit medium-access, and pipelines the packet transmissions over a multi-hop network. Our work shows that it is possible, and beneficial at the same time, to coordinate network access across multiple hops in a simple manner, without global state maintenance. This approach results in deterministic network operation, and allows pre-runtime delay guarantees to be derived for the protocol.

ACKNOWLEDGEMENTS

Support for this research was provided by the Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) through the Carnegie Mellon Portugal Program and by ERDF (European Regional Development Fund) through COMPETE (Operational Programme 'Thematic Factors of Competitiveness'), within projects Ref. FCOMP-01-0124-FEDER-037281 (CISTER), FCOMP-01-0124-FEDER-020312 (SMARTSKIN) and FCOMP-01-0124-FEDER-028990 (PATTERN)

REFERENCES

[1] M. Chan, D. Estve, C. Escriba, and E. Campo, “A review of smart homes present state and future challenges,” *Computer Methods and Programs in Biomedicine*, vol. 91, no. 1, pp. 55 – 81, 2008.

[2] C.-J. M. Liang, J. Liu, L. Luo, A. Terzis, and F. Zhao, “Racnet: a high-fidelity data center sensing network,” in *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, ser. SenSys ’09. New York, NY, USA: ACM, 2009, pp. 15–28. [Online]. Available: http://doi.acm.org/10.1145/1644038.1644041

[3] N. Pereira, S. Tennina, and E. Tovar, “Building a microscope for the data center,” in *Wireless Algorithms, Systems, and Applications*, ser. Lecture Notes in Computer Science, X. Wang, R. Zheng, T. Jing, and K. Xing, Eds. Springer Berlin Heidelberg, 2012, vol. 7405, pp. 619–630. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-31869-6_54

[4] V. Gungor and G. Hancke, “Industrial wireless sensor networks: Challenges, design principles, and technical approaches,” *Industrial Electronics, IEEE Transactions on*, vol. 56, no. 10, pp. 4258–4265, 2009.

[5] S. Kim, S. Pakzad, D. Culler, J. Demmel, G. Fenves, S. Glaser, and M. Turon, “Wireless sensor networks for structural health monitoring,” in *Proceedings of the 4th international conference on Embedded networked sensor systems*, ser. SenSys ’06. New York, NY, USA: ACM, 2006, pp. 427–428. [Online]. Available: http://doi.acm.org/10.1145/1182807.1182889

[6] Atmel, “Atmega 128rfa1 data sheet,” 2011.

[7] Y. Yu, L. J. Rittle, V. Bhandari, and J. B. LeBrun, “Supporting concurrent applications in wireless sensor networks,” in *the 4th international conference on Embedded networked sensor systems*, ser. SenSys ’06. Boulder, Colorado, USA: ACM, 2006, pp. 139–152.

[8] S. Bhattacharya, A. Saifullah, C. Lu, and G.-C. Roman, “Multi-application deployment in shared sensor networks based on quality of monitoring,” *Real-Time and Embedded Technology and Applications Symposium, IEEE*, vol. 0, pp. 259–268, 2010.

[9] V. Gupta, E. Tovar, K. Lakshmanan, and R. Rajkumar, “Inter-application redundancy elimination in wireless sensor networks with compiler-assisted scheduling,” in *Industrial Embedded Systems (SIES), 2012 7th IEEE International Symposium on*. IEEE, 2012, pp. 112–119.

[10] A. Dunkels, B. Gronvall, and T. Voigt, “Contiki - a lightweight and flexible operating system for tiny networked sensors,” in *the 29th IEEE International Conference on Local Computer Networks*, ser. LCN ’04. Washington, DC, USA: IEEE Computer Society, 2004, pp. 455–462.

[11] C. Dixon, R. Mahajan, S. Agarwal, A. J. Brush, B. Lee, S. Saroiu, and V. Bahl, “An operating system for the home,” in *Proceedings of the 9th USENIX conference on Networked systems design and implementation*, ser. NSDI’12, 2012.

[12] B. Porter, U. Roedig, and G. Coulson, "Type-safe updating for modular WSN software," in *Proceedings of the 7th IEEE International Conference on Distributed Computing in Sensor Systems*, ser. DCOSS '11. IEEE, 2011.

[13] A. Rowe, K. Lakshmanan, H. Zhu, and R. Rajkumar, "Rate-harmonized scheduling for saving energy," in *RTSS '08: Proceedings of the 2008 Real-Time Systems Symposium*. Barcelona, Spain: IEEE Computer Society, 2008, pp. 113–122.

[14] NHS, "http://users.ece.cmu.edu/˜vikramg/nhs.htm," 2013.

[15] F. Ferrari, M. Zimmerling, L. Mottola, and L. Thiele, "Low-power wireless bus," in *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*, ser. SenSys '12. New York, NY, USA: ACM, 2012, pp. 1–14. [Online]. Available: http://doi.acm.org/10.1145/2426656.2426658

[16] F. Ferrari, M. Zimmerling, L. Thiele, and O. Saukh, "Efficient Network Flooding and Time Synchronization with Glossy," in *Proceedings of the 10th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, Chicago, IL, USA, April 2011, pp. 73–84.

[17] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient mac protocol for wireless sensor networks," in *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 3, 2002, pp. 1567–1576 vol.3.

[18] T. van Dam and K. Langendoen, "An adaptive energy-efficient mac protocol for wireless sensor networks," in *Proceedings of the 1st international conference on Embedded networked sensor systems*, ser. SenSys '03. New York, NY, USA: ACM, 2003, pp. 171–180. [Online]. Available: http://doi.acm.org/10.1145/958491.958512

[19] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks," in *Proceedings of the 2nd international conference on Embedded networked sensor systems*, ser. SenSys '04. New York, NY, USA: ACM, 2004, pp. 95–107. [Online]. Available: http://doi.acm.org/10.1145/1031495.1031508

[20] M. Buettner, G. V. Yee, E. Anderson, and R. Han, "X-mac: a short preamble mac protocol for duty-cycled wireless sensor networks," in *Proceedings of the 4th international conference on Embedded networked sensor systems*, ser. SenSys '06. New York, NY, USA: ACM, 2006, pp. 307–320. [Online]. Available: http://doi.acm.org/10.1145/1182807.1182838

[21] M. Hansen, R. Jurdak, and B. Kusy, "Unified broadcast in sensor networks," in *Information Processing in Sensor Networks (IPSN), 2011 10th International Conference on*, 2011, pp. 306–317.

[22] M. Maroti and B. Kusy and G. Simon and A. Ledeczi, "The flooding time synchronization protocol," *Proc. ACM Sensys*, 2004.

[23] 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 the 1st conference on Symposium on Networked Systems Design and Implementation - Volume 1*, ser. NSDI'04. Berkeley, CA, USA: USENIX Association, 2004, pp. 2–2. [Online]. Available: http://dl.acm.org/citation.cfm?id=1251175.1251177

[24] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis, "Collection tree protocol," in *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, ser. SenSys '09. New York, NY, USA: ACM, 2009, pp. 1–14. [Online]. Available: http://doi.acm.org/10.1145/1644038.1644040

[25] Rowe A., Mangharam R., and Rajkumar R., "RT-Link: A Time-Synchronized Link Protocol for Energy-Constrained Multi-hop Wireless Networks," *SECON*, 2006.

[26] T. Herman and S. Tixeuil, "A distributed tdma slot assignment algorithm for wireless sensor networks," *CoRR*, vol. cs.DC/0405042, 2004.

[27] J. Degesys, I. Rose, A. Patel, and R. Nagpal, "Desync: self-organizing desynchronization and tdma on wireless sensor networks," in *Proceedings of the 6th international conference on Information processing in sensor networks*, ser. IPSN '07. New York, NY, USA: ACM, 2007, pp. 11–20. [Online]. Available: http://doi.acm.org/10.1145/1236360.1236363

[28] S. Ramanathan, "A unified framework and algorithm for channel assignment in wireless networks," *Wirel. Netw.*, vol. 5, no. 2, pp. 81–94, Mar. 1999. [Online]. Available: http://dx.doi.org/10.1023/A:1019126406181