Into the SMOG: The Stepping Stone to Centralized WSN Control

Pablo Corbalán, Ramona Marfievici, Victor Cionca, Donna O'Shea, and Dirk Pesch
Nimbus Centre for Embedded Systems Research
Cork Institute of Technology, Cork, Ireland
{firstname.lastname}@cit.ie

***Abstract*—Previous research has shown that centralized network control in Wireless Sensor Networks (WSNs) can lead to improved network lifetime, benefit reliability, help to diagnose and localize network failures, assist network recovery, and lead to optimal routing and transmission scheduling. A stepping stone to centralized network control is to build and maintain a complete network topology model that scales and reacts to the network dynamics that occur in low-power wireless networks. We propose SMOG as a mechanism to build and maintain a centralized full network topology model using probabilistic data structures. Extensive analysis of the proposed approach in both simulation and two testbeds shows that SMOG can build a complete model of a WSN of over 100 nodes with 98% accuracy in less than four minutes. Our approach also offers fast recovery from heavy network interference, recovering model accuracy to 98% in less than two and a half minutes.**

I. INTRODUCTION

Centralized network control in Wireless Sensor Networks (WSNs) has many advantages in terms of improved network lifetime, reliability, and delay [1]. Centralized control can also determine, localize, and diagnose network failures, and help with network recovery [2]. It has also been applied to routing in WSNs [3], [4], [5]. Standards for industrial WSNs such as ISA100.11a [6] and WirelessHART [7] take advantage of centralization to build Time-Slotted-Channel-Hopping (TSCH) schedules, and disseminate routes in the network that can meet strict industrial application requirements.

A first step towards centralized network control is to build and maintain a complete network topology model that scales and reacts to the common network dynamics that occur in low-power wireless networks. Previous approaches such as [1], [4] build partial network models, therefore limiting the amount of network-wide decisions that can be taken. On the other hand, WirelessHART appears to have scalability issues still to be addressed [8]. These limitations encouraged us to study and analyze this problem by creating a mechanism to build and maintain an accurate and complete centralized network topology model that scales, is reactive, and adds low overhead.

Here, we propose SMOG, a mechanism to build and maintain a centralized full network topology model using probabilistic data structures, specifically, Bloom filters. The network model captures in a binary matrix the network's connectivity map. Nodes use a suitable protocol to discover and maintain neighborhood information, i.e., the nodes belonging to their neighborhood, which is reported using the probabilistic data structure to populate the model at a central point, e.g., base station or sink. Three different modes of operation are considered for reporting neighborhood information. We analyzed the possible inconsistencies that can occur between the physical network topology and the model due to the probabilistic behavior of wireless links as well as due to the use of the probabilistic data structures. We have studied the implications of these inconsistencies on the accuracy of the model and have extensively evaluated SMOG's behavior in simulations, focusing on scalability, and in two different WSN testbeds, focusing on SMOG's responsiveness under realistic channel conditions. Our analysis shows that SMOG is able to build a complete model of a dynamic WSN with 100 nodes (Indriya testbed [9]) from scratch in less than 4 minutes with over 98% accuracy. Furthermore, we analyzed the impact of interference on maintaining the network model accuracy. We created severe interference in FlockLab [10] to introduce network changes and found that SMOG can recover up to 98% accuracy in less than 2.5 min. We believe that our evaluation results greatly assist in better understanding the possibilities and limitations of building and maintaining complete and accurate centralized network topology models for network-wide optimization and flow scheduling.

The remainder of this paper is organized as follows. Section II introduces SMOG, its requirements, and its dependencies along with the design of SMOG and its core mechanisms. Section III presents and discusses the results of our extensive evaluation in simulation and testbed experiments. We review related work in Section IV, provide a discussion and outlook in Section V, and conclude in Section VI.

II. SMOG DESIGN

SMOG is a mechanism to build and maintain a complete and accurate centralized network topology model using probabilistic data structures. SMOG collects neighborhood information from every node in the network at a central entity, e.g., base station or sink, which processes this information and updates the model. To be useful for centralized network control, the network model created using SMOG needs to *accurately* reflect the underlying network topology. Also, SMOG should be *reactive* to network changes and should be able to reliably and rapidly detect these changes and update the model. It needs to be *scalable* to work with large networks and is designed for low-power wireless networks that need to be *energy-efficient*

978-1-5090-2833-7/16 $31.00 © 2016 IEEE
DOI 10.1109/MASS.2016.29

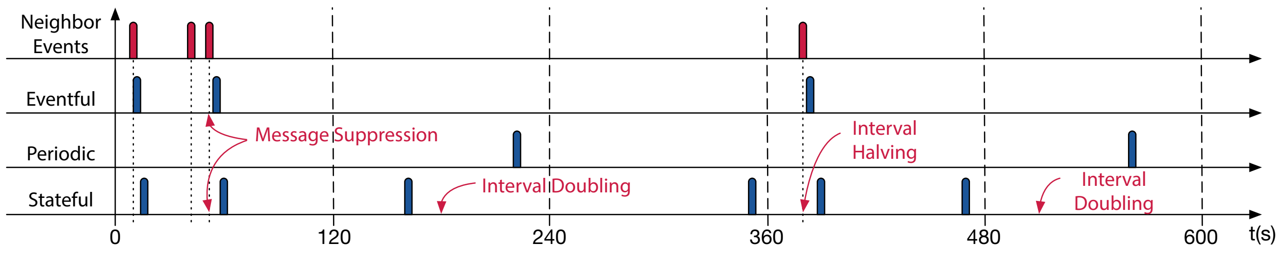


Fig. 1. Modes of Operation (MOPS). *Eventful* and *stateful* react to neighbor events, while *periodic* is unresponsive to them.

for long battery life. Hence, SMOG needs to add low overhead in terms of energy consumption.

SMOG relies on underlying network protocols that build routes to the central entity to report SMOG messages, and enable nodes to discover and maintain their neighbors. In our case, we use RPL [11] to discover and maintain neighbors and build routes to the central entity. We use IPv6 NDP [12] to detect neighbor unreachability. Moreover, we use RPL information to improve the accuracy of our model. However, the main ideas behind SMOG are independent of RPL and NDP. Hence, SMOG should be easy to adapt to other constrained wireless network technologies that require a mechanism to build and maintain an accurate centralized network model.

Next, we describe the design of SMOG. We begin with a description of the SMOG model and its core mechanisms. We then introduce the possible inconsistencies and the metric we define to assess the accuracy of the model.

A. Model

Given a WSN, let $\mathcal{D}_R = (V_R, E_R)$ be the directed graph that represents its physical connectivity, where V_R is its non-empty set of vertices representing the nodes, and E_R its set of edges representing the directed links. As we wanted to account for the presence of link asymmetries, we opted for a directed graph. Network protocols perceive the one-hop (local) neighborhood of each node. Let $\mathcal{D}_l = (V_l, E_l)$ be the directed graph that represents the logical network topology built by composing these local neighborhoods. According to [13], at a certain point in time, $\mathcal{D}_l$ may not be consistent with $\mathcal{D}_R$.

Our focus in SMOG is to collect at a central entity the logical network topology ($\mathcal{D}_l$) to build and maintain an up-to-date and accurate network model, denoted by $\mathcal{D}_s = (V_s, E_s)$. Ideally, $\mathcal{D}_s = \mathcal{D}_l$, but packet loss and delays can introduce differences between these two graphs.

B. Mechanisms

This section describes the core mechanisms of SMOG, that is: *1)* its Modes of Operation (MOPS), *2)* Bloom filters (BF), and *3)* a False Positive (FP) Discovery Mechanism.

*1) Modes of Operation (*MOPS*):* Dynamic network conditions generate neighbor change events. Any such change must be detected and reliably reported to the central entity for updating the network model, as this affects the accuracy of the model and might have an impact on higher layer applications that depend on SMOG. Hence, reliability and reactivity are two requirements for the design of SMOG. To this end, we define three modes of operation for reporting neighborhood information: $i)$ *eventful*: an event triggers a message 1 to 5 s after it occurred. If another event occurs before the message is sent, the previous message is suppressed and a new SMOG message is scheduled. $ii)$ *periodic*: a message is sent during the second half of a static interval of 5 min. $iii)$ *stateful*: an event triggers the same behavior as in *eventful* but with larger intervals (10 to 15 s). In the absence of events, SMOG messages are sent during the second half of a dynamic interval (2 to 20 min). To limit overhead, if no events occur the interval doubles, eventually reaching the maximum of 20 min. When neighbor events occur, the interval halves, eventually reaching the minimum of 2 min. The randomized transmission included in all MOPS is used to avoid collisions. Figure 1 further illustrates how each MOP reacts to neighbor events. The choice between *eventful*, *periodic*, and *stateful* should be left to the user of SMOG, balancing the network's dynamicity against the requirements of higher layer applications.

2) Bloom Filters: Explicitly reporting neighbor changes has disadvantages. First, losing a report on a recent event results in the central model not reflecting the neighbor change unless an extra mechanism, e.g., end-to-end ACKs, is used. Secondly, reporting the whole neighborhood of a node can become prohibitive for dense networks. In a 6LoWPAN network, a single node with 40 neighbors would have to report 40 IPv6 or IEEE 802.15.4 MAC addresses. On the other hand, we can envisage a solution in which a node reports only a subset of the neighborhood, but this results in having partial network models. SMOG overcomes these disadvantages by using Bloom filters (BFs) to compress the whole neighborhood information of a node into a single bit array. As a consequence, nodes report BFs instead of neighbor addresses or identifiers. BFs are space-efficient probabilistic data structures that can be used for set insertion and membership query [14], [15]. A BF is an m-bit array. To insert an element into a BF, we apply k hash functions to the element, obtaining the k positions to be set in the array. To check if the BF contains an element, we check if the k bits obtained by hashing that element are set in the array. BFs determine if an element either *is definitely not* in the set or *may be* in the set. Therefore, BFs produce False Positives (FP), which makes the SMOG model probabilistic.

For specific m, k, and n elements inserted in the BF, the probability of having a FP is defined as in [14]:

$$p = \left(1 - \left(1 - \frac{1}{m}\right)^{kn}\right)^{k} \tag{1}$$

We analyze the feasibility of using BFs with SMOG by checking their false positive rate with two configurations: $i)$ $m = 128$ bits, $k = 4$, and $ii)$ $m = 256$ bits, $k = 8$. The value of k was selected to minimize p, given m and for $n = 20$. To obtain k hash functions, we split a single Shift-add-XOR (SAX) hash function into k hashes, as in [15]. In our test, we insert IEEE 802.15.4 MAC addresses into a BF and we compare the FP rate obtained with SAX to the theoretical bound obtained from Eq. (1). Figure 2 shows a close fit between real and theoretical FP rate. This makes BFs a good solution to compress the entire local neighborhood information of a node into a single bit array that fits in a packet.

In SMOG, nodes start with an empty BF, and then add MAC addresses to it as neighbors are discovered. When a neighbor is removed, the whole BF is recomputed as BFs do not support delete operations. Then, nodes report BFs together with their total number of neighbors, their preferred parent MAC address, and the RPL Rank, i.e., distance to the RPL root using a cost function [11]. Upon message receiving, the sink computes the neighbors of the sender by checking the MAC addresses that are already in the network model ($\mathcal{D}_s$) against the BF. This results in a set of *possible* neighbors. Before updating this set in the network model, the False Positive Discovery Mechanism is activated.

3) False Positive Discovery Mechanism: Because of the FPs produced by BFs, network-wide decisions or flow scheduling relying on the network model can be wrong. To overcome this, we designed a false positive discovery mechanism to detect and remove FPs. Let $\mathcal{S}_b$ be the set of possible neighbors of a node, computed as explained in the previous section (II-B2), with $s_b = |\mathcal{S}_b|$ elements. Let $s = |\mathcal{S}|$ be the number of neighbors reported by the same node. Then, if $s_b > s$, the difference $s_b - s$ constitutes the number of FPs. Knowing the number of FPs, the next step is to discover them. For this, we define three rules: $i)$ *parent*: the preferred parent reported is the only certain neighbor, $ii)$ *neighbor semi-reciprocity*: if node $\mathcal{A}$ is a neighbor of $\mathcal{B}$, $\mathcal{B}$ *may be* a neighbor of $\mathcal{A}$, $iii)$ *rank*: the RPL rank [11] distance between two nodes must be $\leq 2 \times MinHopRankIncrease$.

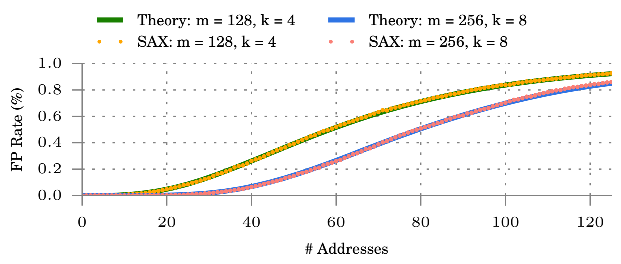
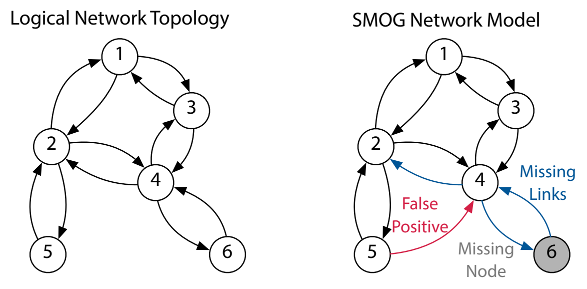


Fig. 2. Bloom Filter False Positive Rate.



	1	2	3	4	5	6
1	0	1	1	0	0	0
2	1	0	0	1	1	0
3	1	0	0	1	0	0
4	0	1	1	0	0	1
5	0	1	0	0	0	0
6	0	0	0	1	0	0

	1	2	3	4	5	6
1	0	1	1	0	0	-
2	1	0	0	1	1	-
3	1	0	0	1	0	-
4	0	0	1	0	0	-
5	0	1	0	1	0	-
6	-	-	-	-	-	-

Fig. 3. Inconsistencies.

These rules are applied in sequence to the set of possible neighbors ($\mathcal{S}_b$), generating three subsets $\mathcal{S}_p$, $\mathcal{S}_{sr}$, and $\mathcal{S}_r$. The neighbor semi-reciprocity rule is not a strict rule, because links can be asymmetrical and the respective neighbors can also have FPs. The other two rules are strict, unless rank values are outdated. The set of neighbors of a node in the model ($\mathcal{S}_M$) is computed as $\mathcal{S}_p \cup \mathcal{S}_{sr} \cup \mathcal{S}_r$. If the union set still contains FPs, we reduce $\mathcal{S}_M$ to $\mathcal{S}_p \cup \mathcal{S}_r$. It must be noted that this mechanism is not guaranteed to eliminate all FPs. In fact, if $|\mathcal{S}_M| < s$, it means that real links are missing or have been removed. However, making central decisions using links that are not available (FP) will always fail, whereas not considering available links will at worst achieve sub-optimality.

C. Model Accuracy

We describe the possible inconsistencies that might arise between $\mathcal{D}_s$, the network model built by SMOG, and $\mathcal{D}_l$, the logical network, and their impact on the model accuracy. We define: *1)* the types of inconsistencies, and *2)* the model accuracy metric as a function of the inconsistencies.

1) Inconsistencies: Differences between $\mathcal{D}_l$ and $\mathcal{D}_s$ constitute the inconsistencies. These can arise from several reasons, e.g., packet delay or loss. We classify the possible inconsistencies as follows: $i)$ *Missing Node*: a node that exists in $\mathcal{D}_l$ and is not contained in $\mathcal{D}_s$, $ii)$ *Missing Link*: a link that exists in $\mathcal{D}_l$ and is not contained in $\mathcal{D}_s$, and $iii)$ *False Positive Link*: a link that exists in $\mathcal{D}_s$ and is not contained in $\mathcal{D}_l$. Figure 3 shows the impact of the possible types of inconsistencies on the $\mathcal{D}_s$ adjacency matrix. A *Missing Node* inconsistency affects $2|V_l| - 1$ elements, while a *Missing Link* or a *False Positive Link* only affects one element of the matrix.

2) Model Accuracy Definition: To be useful, SMOG should build and maintain a network model ($\mathcal{D}_s$) that accurately represents the logical network topology ($\mathcal{D}_l$). The inconsistencies that appear over time affect the accuracy of the model. To appropriately understand the accuracy of the model achieved by SMOG, we define the Model Accuracy metric, $M_a(t)$, which reflects the total amount of differences between $\mathcal{D}_s$ and $\mathcal{D}_l$ at time t. Denoting $s_{ij}(t)$ and $l_{ij}(t)$, with $i, j \in \mathcal{N}$, $i \neq j$, the elements of the adjacency matrices of $\mathcal{D}_s$ and $\mathcal{D}_l$, respectively, at time t, and $n = |\mathcal{N}|$ the total number of nodes of $\mathcal{D}_l$, we can define $M_a(t) \in [0, 1]$ as:

$$M_a(t) = 1 - \frac{\sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}} |l_{ij}(t) - s_{ij}(t)|}{n^2 - n} \tag{2}$$

Denoting $\mathcal{I}_{mn}(t)$, $\mathcal{I}_{ml}(t)$, and $\mathcal{I}_{fp}(t)$ the number of missing node, missing link, and FP inconsistencies, respectively, at time t, Eq. (2) can be redefined as:

$$M_a(t) = 1 - \frac{\mathcal{E}_{mn}(t) + \mathcal{I}_{ml}(t) + \mathcal{I}_{fp}(t)}{n^2 - n} \tag{3}$$

Where $\mathcal{E}_{mn}(t) = n^2 - (n - \mathcal{I}_{mn}(t))^2 - \mathcal{I}_{mn}(t) - \mathcal{I}^*_{ml}(t)$ penalizes the effect of having missing node inconsistencies without taken into account neither the main diagonal elements nor the missing links associated to missing nodes ($\mathcal{I}^*_{ml}(t)$).

According to Eq. (2), the absence or presence of a link has the same impact on the model accuracy. As a consequence, for a network with graph density, $\rho = 0.2$, 80% of the elements of its adjacency matrix are zero. This potentially allows SMOG to reach $M_a(t) \geq 80\%$.

III. EVALUATION

In this section, we present: *A)* the key metrics we looked at to evaluate SMOG, *B)* our system integration, *C)* a simulation-based evaluation to analyze the impact of network size and density on SMOG, and *D)* a testbed-based evaluation, focusing on effectiveness under realistic channel conditions.

A. Metrics

We consider three metrics to assess the performance of SMOG: *i)* *model accuracy*, $M_a(t)$, as defined in Section II-C2, *ii)* *end-to-end packet delivery ratio (PDR)* per MOP, i.e., ratio of SMOG messages received at the sink over those collectively sent by the nodes, and *iii)* *radio duty cycle*, i.e., portion of time spent with the radio on. The first metric accounts for the accuracy of SMOG in building and maintaining the central model, while the second provides a direct assessment of the reliability of each MOP in reporting neighborhood information. The third metric measures the energy consumption necessary to build and maintain the model and ensure its reliability. To further understand the overhead added by SMOG, we compare the duty cycle per MOP to a baseline obtained with a Null application running on top of RPL and NDP. Moreover, we looked at the *number of inconsistencies*, i.e., number of differences between the logical network and SMOG model adjacency matrices, and the *reactivity*, i.e., inconsistency duration.

B. System Integration

We used Contiki's network stack with 6LoWPAN, RPL, NDP, and ContikiMAC with a wake-up interval of 125 ms. SMOG sent its messages on top of UDP. Neighbor reachability was confirmed with Link Layer ACKs to reduce NDP traffic. Neighbor lifetime is 10 min and we set the maximum number of neighbors to 40 to avoid further neighbor changes due to neighbor cache limitations. The MOPS and Bloom filters used are as described in Section II. The duty cycle baseline application uses exactly the same parameters in 6LoWPAN, RPL, NDP, and ContikiMAC for a fair overhead comparison. We log every neighbor change to compute offline the logical network topology ($\mathcal{D}_l$) and compare it to the SMOG network model ($\mathcal{D}_s$) over time. Thus, $\mathcal{I}_{mn}(t)$, $\mathcal{I}_{ml}(t)$, $\mathcal{I}_{fp}(t)$, and $M_a(t)$ can be obtained at any point in time.

C. Simulation-based Evaluation

To carry out a fine-grained analysis of SMOG's behavior in a controlled environment, we ran a set of experiments in Cooja, focusing on scalability under favorable radio conditions to isolate the impact of the network size and density on SMOG.

1) Simulation Settings: We emulated square grid network topologies with sizes ranging from 4 to 121 nodes, and different network densities, from 3.64 to 16.16 on average, for the 121-node network. Note that changing the network size or density in a square grid network changes the network diameter. The sink, i.e., RPL root, was placed in a corner to maximize the network diameter. In our simulations, we resorted to the Unit Disk Graph Medium (UDGM) radio model provided by Cooja (commonly used in the literature) to isolate the impact of the network size and density on SMOG. We used two BF configurations: *i)* $m = 128$ bits and $k = 4$ hash functions, and *ii)* $m = 256$ bits and $k = 8$ hash functions. The size of the BF and the number of hash functions have an impact of the FP rate as shown in Section II-B2 and as a consequence, can affect the model accuracy achieved by SMOG. The values of k set for the BFs minimize the FP rate for $n = 20$ neighbors. All experiments were carried out over 20 minutes. Each configuration was tested five times for statistical relevance. We present results from 390 experiments with the MOPS and 65 experiments to compute the duty cycle baseline.

2) Network Size: We evaluated the impact of the network size on SMOG by changing the network size from 4 to 121 nodes. Figure 4 shows the average model accuracy at $t = 1200$ s after the experiment was started and the network became stable, the duty cycle, and the end-to-end PDR per MOP with a BF of size $m = 256$ bits. We observe that *periodic* and *stateful* achieve $M_a(t = 1200s) = 100\%$ with all network sizes, while *eventful*'s accuracy decreases with increasing network size to 92.88%. Due to the stable radio conditions, in simulations, all neighbor events occurred at the beginning of the experiments while the network was being built. The resulting congestion reduced *eventful*'s end-to-end PDR to 80.97%, causing loss of model update messages. There were no more events after this period, so the model could not be updated, resulting in lower accuracy for the *eventful* MOP. The message reporting periodicity of *periodic* and *stateful* helped these MOPS achieve very high PDR ($\geq 99.81\%$ and $\geq 99.83\%$, respectively) and therefore, achieve such high accuracy. On the other hand, the stable radio conditions and

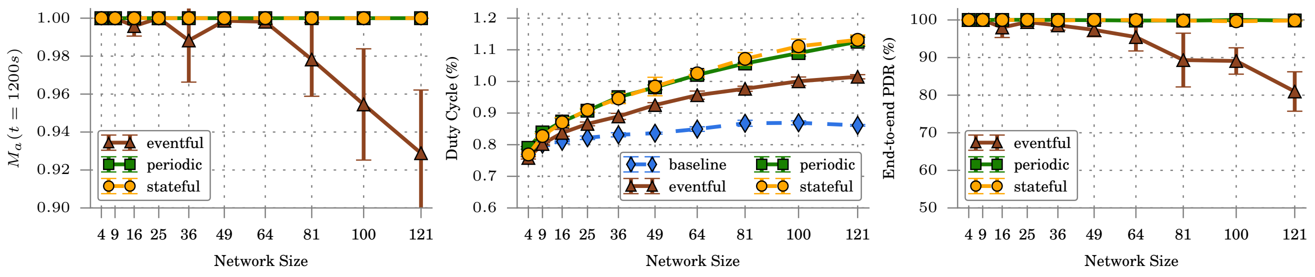


Fig. 4. Network Size Experiment: Model accuracy at $t = 1200$ s (left), duty cycle (centre), and end-to-end PDR (right) with Bloom filter size $m = 256$ bits and $k = 8$ hash functions.

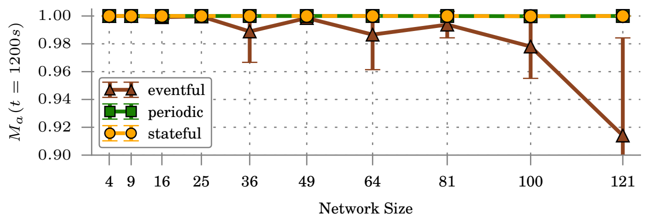


Fig. 5. Network Size Experiment: Model accuracy at $t = 1200$ s with Bloom filter size $m = 128$ bits and $k = 4$ hash functions.

the lack of neighbor events, resulted in a duty cycle overhead for *eventful* of 17.83% over the baseline for the network size of 121 nodes. *Periodic* and *stateful* had a higher overhead due to the periodic messages sent by both MOPS ($\leq 30.65\%$ and $\leq 31.35\%$, respectively).

Figure 5 shows the average model accuracy at $t = 1200$ s with the BF configuration of size $m = 128$ bits. The duty cycle and end-to-end PDR with both BF configurations are very similar, and their results for the BF configuration of size $m = 128$ bits are not shown. With this configuration, the model accuracy achieved by *periodic* and *stateful* could not reach 100% for the network size of 100 nodes due to 1 to 3 missing link inconsistencies per experiment.

With this particular setup, a FP appears when SMOG computes the neighbors of a node. The FP Discovery Mechanism detects, before adding the false positive link, that there is a FP and in its effort to avoid having FP inconsistencies, removes 1 to 3 neighbors from the model (depending on RPL rank values), creating missing link inconsistencies. This can be explained by the increased FP rate with the BF configuration of size $m = 128$. On the other hand, *eventful* decreased its accuracy to 91.4% with the 121-node network.

The low network density in these experiments, ranging from 2 to 3.64 on average, decreased the chances of having false positives and therefore enabled to achieve $M_a(t = 1200s) \geq 99\%$ with both BF configurations with *periodic* and *stateful*.

3) Network Density: We evaluated the impact of the network density with a fixed network size of 121 nodes by changing the radio range of the nodes, resulting in network densities ranging from 3.64 to 16.16 on average. Figure 6 shows the impact of the network density on the key metrics of SMOG with the BF configuration of size $m = 256$ bits. Similar to the previous network size experiment, *periodic* and *stateful* reach $M_a(t = 1200s) \geq 99.8\%$ with all network densities tested, while *eventful*'s accuracy also decreases, due to the increased congestion and higher number of events. The *eventful* accuracy reaches a minimum of 69.4% with an average density of 6.94 neighbors. At this specific density the network diameter was larger than at higher densities, and the increased congestion led to the lowest *eventful* end-to-end PDR of 62.84%. For the highest network density experiment, due to the higher number of neighbor events to report, *eventful* had a higher overhead than the other MOPS of up to 12.67%. *Stateful*, despite being responsive to neighbor events, has a very similar overhead to *periodic*. This can be explained by the message suppression mechanism of *stateful*.

Figure 7 shows the average model accuracy at $t = 1200$ s with the BF configuration of size $m = 128$ bits. With this configuration, the accuracy with *periodic* and *stateful* decreases to 96.6% and 96.9% at the highest density tested. The higher density increased the appearance of FP inconsistencies with that BF configuration. Additionally, the FP discovery mechanism in its effort to remove these inconsistencies, produced missing link inconsistencies. Note that this is a consequence of SMOG's priority to remove FPs. On the other hand, with the BF configuration of size $m = 256$ bits, FPs were rare and consequently, *periodic* and *stateful* had no difficulties in achieving such high accuracy. These results reiterate the importance of selecting an appropriate BF configuration depending on the network characteristics as discussed in Section II-B2.

Overall, the simulation experiments prove SMOG's capability to build a complete and accurate centralized network model that scales, at least, up to 121 nodes with different network densities with *periodic* and *stateful*. The results show scalability issues with *eventful* under the simulated network conditions. Moreover, PDR and FP rate proved to be the main reasons for decreased accuracy.

D. Testbed-based Evaluation

Next, we present an evaluation of SMOG in two different WSNs testbeds, focusing on effectiveness under realistic network conditions and dynamics, and the impact of interference.

1) Testbed Settings: We evaluate the behavior of SMOG in Indriya [9] and FlockLab [10]. Indriya features a 100 TelosB node deployment in a three-floor office building, and FlockLab a 31 TelosB node deployment in a single floor of a university building with three nodes outdoors. In both testbeds, we set node #1 as the RPL Root and sink; we used Bloom filters of size $m = 256$ bits and with $k = 8$ hash functions; we used channel 26 to avoid cross-technology interference, e.g.,

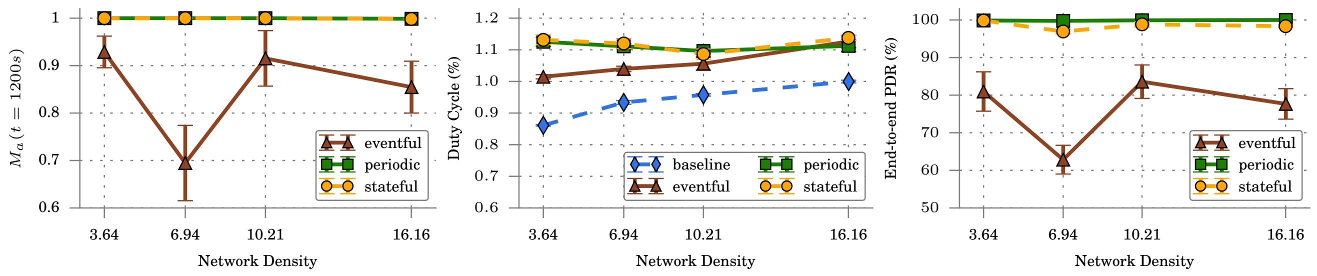


Fig. 6. Network Density Experiment: Model accuracy at $t = 1200$ s (left), duty cycle (centre), and end-to-end PDR (right) with Bloom filter size $m = 256$ bits and $k = 8$ hash functions.

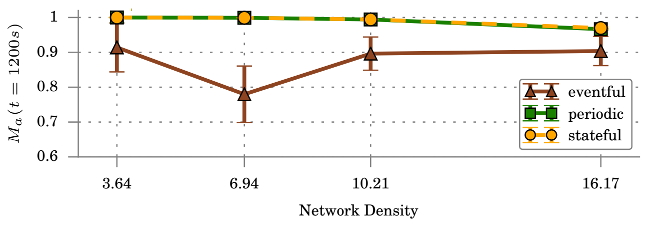
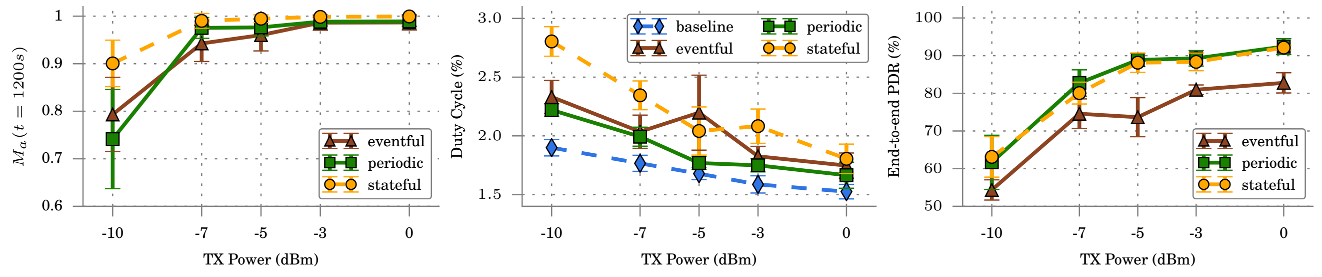


Fig. 7. Network Density Experiment: Model accuracy at $t = 1200$ s with Bloom filter size $m = 128$ bits and $k = 4$ hash functions.

WiFi. The results we report in this section correspond to 153 experiments with the MOPS and 51 experiments to compute the duty cycle baseline.

2) Network Dynamics: To analyze the impact of network dynamics we ran a set of experiments with five transmission (TX) powers, from -10 dBm to 0 dBm, resulting in networks with different link qualities and of different densities, ranging from 6.55 to 11.27 on average in Indriya and from 3.73 to 6.71 in FlockLab. All experiments lasted for 20 min. The testbed results shown are averages with standard deviations. Figure 8 shows the model accuracy at $t = 1200s$, duty cycle, and end-to-end PDR in Indriya (top) and FlockLab (bottom). We observe different trends compared to simulations, arguably due to different network and channel conditions. Firstly, all three MOPS achieve $M_a(t = 1200s) \geq 98.64\%$ in Indriya and $M_a(t = 1200s) \geq 99.22\%$ in FlockLab with TX Power ≥ -3 dBm. In the testbeds, dynamic channel conditions produced neighbor events after network convergence, increasing *eventful*'s accuracy up to 98.68% in Indriya and 99.78% in FlockLab. Also, these neighbor events led to a higher duty cycle in *eventful* and *stateful* w.r.t. the baseline, 15.06% and 31.26% respectively in Indriya, and 13.15% and 16.54% in FlockLab at -3 dBm. *Periodic*, unresponsive to neighbor events, had the lowest duty cycle overhead ($\leq$ 10.21% in Indriya and $\leq$ 6.93% in FlockLab with TX Power ≥ -3 dBm). Secondly, we also observe that all MOPS, despite not achieving very high end-to-end PDR ($\leq$ 92.5% PDR in Indriya and $\leq$ 96.33% in FlockLab) manage to solve most of the inconsistencies and achieve high accuracy. This observation can be explained with the argument that using BFs, a node can easily communicate its entire neighborhood. This means that even if a neighbor change could not be successfully reported to the central entity, it is likely to be reported later on. At -10 dBm, the poor results obtained by all MOPS can be explained by the low connectivity in both testbeds.

Figure 9 shows the relationship between the number of inconsistencies and the model accuracy over time in FlockLab with a TX Power of 0 dBm. At the beginning of an experiment, SMOG starts without having any knowledge of the network. During the first minutes, nodes start joining the RPL network and discovering their neighbors, which leads to missing node and missing link inconsistencies. This results in temporary peaks of very high number of inconsistencies and troughs of accuracy. *Eventful* and *stateful*, due to their responsiveness to neighbor events, solve most of these inconsistencies very fast, obtaining more than 97% and 96% accuracy, respectively in less than three minutes. On the other hand, *periodic* takes longer to solve its inconsistencies due to its unresponsiveness to neighbor events achieving $M_a(t) \geq 97\%$ with $t \geq 15$ min. At 0 dBm in Indriya, *stateful* achieves $M_a(t) > 98\%$ in less than four minutes, while *eventful* takes more than 14 min to achieve such accuracy, arguably due to high packet loss at the beginning of the experiments while RPL creates network contention. *Periodic* achieves $M_a(t) > 97\%$ with $t \geq 14$ min. Figure 9 also shows that SMOG's performance is consistent across experiments. We show this by plotting the standard deviation for the number of inconsistencies and model accuracy over time for all the MOPS across five experiments. When we looked at the data collected during the experiments, we observed that decreasing the TX power comes with an increase in the standard deviation with respect to the one observed in Figure 9. We argue that this effect is due to links getting closer to the transitional region [16] and therefore producing more unstable conditions in the network. Figure 10 shows the histogram of the duration of inconsistencies for each MOP, which represents their reactivity to network changes. *Eventful* is the most reactive, followed closely by *stateful*. *Periodic*, unresponsive to neighbor events, presents peaks at $(5x - 1.25)$ min, $\forall x \in \mathbb{Z}^+$, as per definition of the MOP. The second lower peak is the result of the inconsistencies that could not be resolved with the first report, but were solved with the second. Overall, the testbed experiments demonstrate SMOG's effectiveness under real channel conditions with different densities and topologies. The experiments in Indriya also demonstrate that SMOG scales up to 100 nodes with realistic channel conditions and the network features tested. The results also show a trade-off between energy consumption and reactivity. Network dynamics produce neighbor events after network convergence, which helps *eventful* achieve high accuracy, and resolve its scalability issues observed in simula-


(a) Indriya (100 nodes).

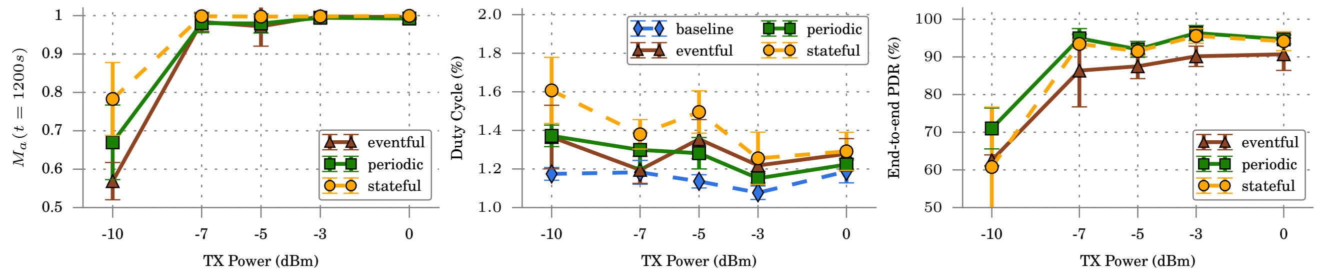

(b) FlockLab (31 nodes).

Fig. 8. Testbed Experiments: Model accuracy at $t = 1200\ s$ (left), duty cycle (centre), and end-to-end PDR (right) with Bloom filter size $m = 256$ bits and $k = 8$ hash functions.

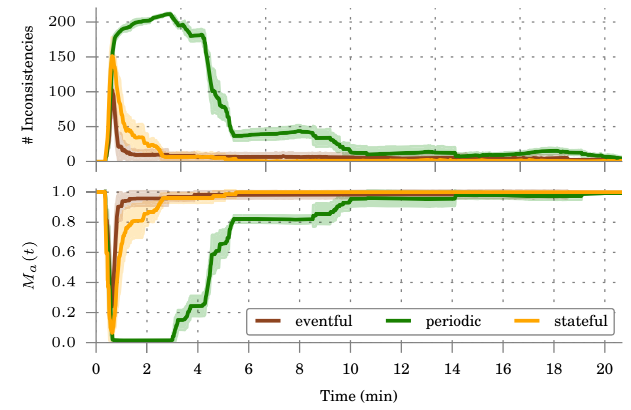


Fig. 9. FlockLab Experiment: Mean and Standard Deviation of the number of inconsistencies and model accuracy over time per MOP at 0 dBm.

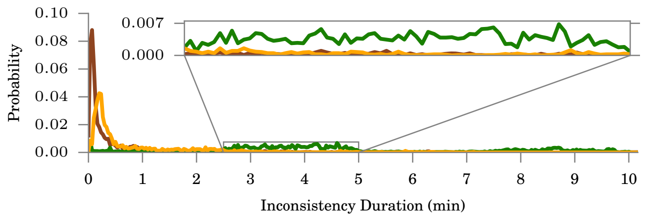


Fig. 10. Indriya Experiment: Inconsistency Duration histogram at −3 dBm. Only solved inconsistencies are considered.

tions. These neighbor events increase the duty cycle of *eventful* and *stateful*, which pay an energy price to be more reactive than *periodic*.

3) Impact of Interference: We investigated SMOG's behavior under controlled interference in FlockLab. To this end, we emulated a WiFi file transfer interference pattern using JamLab [17]. We set two nodes as interferers, node #8 and node #22, which block communication and produce neighbor changes. Node #8 is located very close to the RPL Root, and therefore makes it difficult to successfully report SMOG messages. Node #22 is located in a dense area of FlockLab, directly interfering with 41.37% of the network. The interference generated by both nodes directly affects 17 nodes, i.e., 58.62% of the network. We interleave periods of 20 min without interference with periods of 15 min with interference. Because the neighbor lifetime expiration used by NDP is 10 min, we generated interference for 15 min in order to increase the chances of neighbor removals. All nodes, including interferers, use 0 dBm as transmission power and channel 26 to avoid other sources of interference. The experiments ran for 90 min, including two periods of interference.

Figure 11 shows the impact of interference on the number of missing link ($\mathcal{I}_{ml}(t)$) and false positive inconsistencies ($\mathcal{I}_{fp}(t)$), model accuracy ($M_a(t)$), duty cycle, and end-to-end PDR. During the interference periods, the PDR dramatically decreases to 9.83%, 16.51%, and 20.63% on average for *eventful*, *periodic*, and *stateful* respectively, arguably due to the single-channel feature of ContikiMAC [18]. The high packet loss triggers more ContikiMAC retransmissions, and an increase in the duty cycle. However, we observe that the MOPS' duty cycle overhead w.r.t. to the baseline remains relatively low. The packet loss during interference also makes NDP operation difficult and nodes suffer to confirm reachability to their neighbors. This produces neighbor removals and creates false positive inconsistencies in the network model, consequently reducing the model accuracy. However, we observe that even during the interference periods $M_a(t) > 90\%$ for all MOPS. SMOG recovers rapidly after interference ceases. *Stateful* recovers from 90.89% accuracy observed during the second period of interference to more than 98% in less than two and a half minutes. On the other hand, *eventful* and

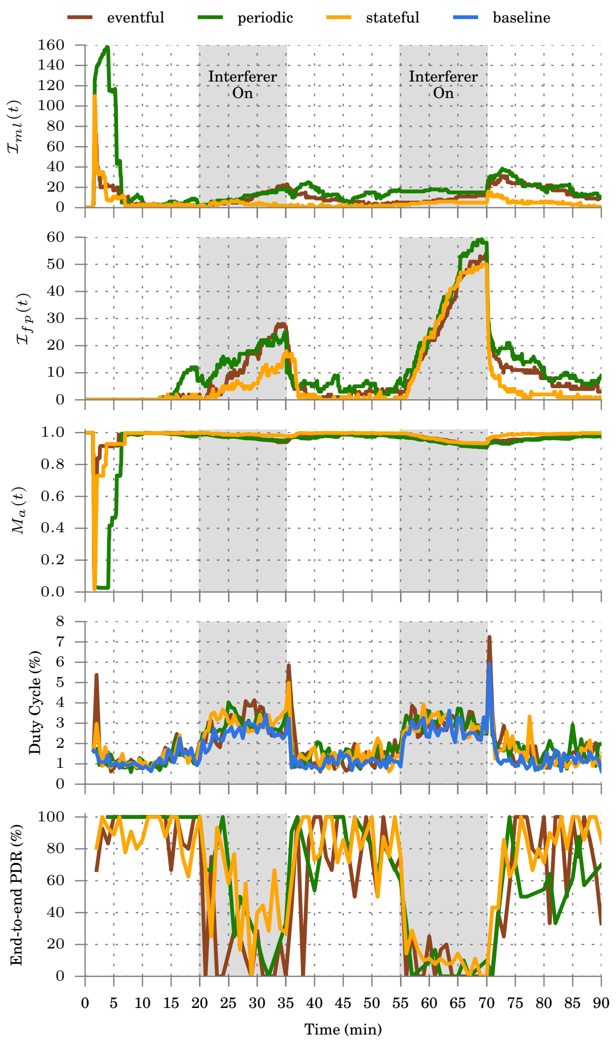


Fig. 11. FlockLab Experiment: Impact of interference on SMOG.

periodic take more than 15 min to recover from 91.87% and 90.89% up to more than 98% and 97.7% accuracy respectively. The higher amount of messages sent by *stateful*, due to its behavior when neighbor events occur allows this MOP to recover faster and better than the other two. After the second peak of interference, we also observe a temporary increase of the missing link inconsistencies, especially with *eventful* and *periodic*, that is the result of the FP discovery mechanism that removes existing links from the model as explained in Section II-B3. We argue that the difference between the two peaks in terms of FP inconsistencies could be related to RPL creating a sub-optimal topology after the first period of interference.

Overall, we observe that interference has a significant impact on SMOG, producing numerous neighbor changes that lead to inconsistencies that cannot be resolved during the interference periods due to packet loss. However, SMOG with *stateful* recovers fast after the interference ceases. Nevertheless, we argue that the impact of interference on SMOG could be highly reduced by using multi-channel MAC layer protocols, like MiCMAC [19], which have already been shown to increase the resilience to interference. We plan to extend our experiments and to investigate how SMOG performs using a channel hopping MAC.

IV. Related Work

The goal of supporting centralized routing, decoupling of control and data plane, and of computing network-wide communication schedules has led several researchers to build and maintain a global view of the network. In CentRoute [3], one of the first solutions for centralized routing in WSNs, a node includes a link estimate in the join request. Upon receiving membership requests, the sink merges all entries into a global membership view. Then, the nodes *periodically* push updates towards the sink. Koala [5] decouples control and data plane at the node level, by implementing a network-wide routing control plane. Koala's decisions to select the routes are driven by the information that the network's nodes collect. Once awake, each node collects its neighborhood: identities of neighbors and the quality of links, i.e., $RSSI$. To accelerate the neighborhood collection process, nodes send periodic beacons which are also acknowledged, generating bidirectional link info. The nodes select the beaconing interval from an exponential distribution and suppress their transmissions if they received a beacon before their timer expires. Hydro [4], the initial routing protocol for the TinyOS 6LoWPAN stack, has its collection part based on directed acyclic graphs created proactively. On demand routes are constructed by the DODAG root that maintains a global view of the network topology based on link state reports coming from the network nodes. Rather than attempt to maintain a complete global view, Hydro makes the trade-off for reduced control traffic by maintaining a view of a *limited* subset of node's neighbors, normally only its best (in terms of link quality). Moreover, Hydro piggybacks the link state information on every outgoing data traffic packet. This results in obvious limitations in scalability and adaptability to varying quality of links. pTunes [1] rests upon a centralized approach. To reason about network-wide performance, it periodically collects at a central entity reports from each node that contain local routing and network state information. For this it uses Glossy [20] network floods: following the initial flood by the sink, each of the other nodes initiates a flood in turn, within exclusive slots. Nevertheless, pTunes builds only the Contiki Collect tree. Sympathy [2] uses connectivity metrics gathered from nodes, for detecting and localizing routing failures. The routing table and the set of neighbors is collected from each node. Passive collection is done using a plug-in to snoop the routing control packets broadcasted by nodes. Active collection, from distant nodes whose broadcasts were not heard by the sink, is done using a TinyOS module which periodically collects metrics from different stack layers. Active metric collection is triggered by a TinyOS timer and all metrics time out after a period. WirelessHART [7] and ISA-101.11a [6], both provide services for creating complex data flows that reliably interconnect factory devices. Apart from deterministic MACs, the reliability of multi-hop delivery is increased by using directed acyclic graphs with redundant paths from source to destination. Both data link and routing layers of the network nodes are controlled by a dedicated device, the Network Manager, in a centralized manner, based on information about the global network state.

All these related works, however, do not aim to characterize the efficiency of building a complete central network model nor its accuracy, which instead is the goal of this paper.

V. Discussion and outlook

The complete central network model built and maintained by SMOG can directly inform flow scheduling decisions and network-wide control and optimization. However, in order to exploit the network model, the conditions at the physical layer that enable the communication between the nodes in the network (e.g., in terms of $RSSI$ or PDR) have to be known. In this respect, SMOG can be easily equipped with mechanisms to acquire and report the quality of the links. For example, SMOG could be configured to report only highly reliable neighbors (i.e., $PDR > 90\%$). However, this results in having a view of a limited subset of nodes' neighbors instead of a global view. Also, SMOG could use different Bloom filters to represent subsets of neighbors with different link qualities (e.g., perfect, intermediate, and poor). This is already on our agenda and it will enable us to enhance the network model with a qualitative perspective of the network.

Despite the encouraging results, further work is required to explore the impact of false positives on mechanisms that make use of SMOG. For instance, a central routing protocol might decide to use a false positive link. This would result in data not being successfully delivered, higher energy consumption due to an increased number of retransmissions, and possibly side effects on neighboring nodes. We plan to revisit SMOG's design and devise new solutions that leverage multiple BFs to report different neighborhood subsets.

Finally, a practical use of the SMOG network model would be its integration in a system where network-wide decisions (e.g., flow rules) are taken based on QoS application requirements and the current network state. These decisions would then need to be disseminated throughout the network for the nodes to act accordingly. Changes in the network state (e.g., a neighbor removal or a node failure) would trigger model updates that might cause a recomputation of the network decisions previously taken. In some cases, this might be followed by a new dissemination process to update or insert new rules in order to adapt to the new network state.

VI. Conclusions

We presented SMOG, a mechanism to build and maintain a complete and accurate network topology model. At its core, SMOG uses Bloom filters to represent neighborhood information and offers multiple modes of operation to report this information to a central entity. Our extensive evaluation, in simulations and two testbeds, clearly shows that SMOG can build and maintain a model that scales and reacts to common network dynamics.

To the best of our knowledge, our study is the first that looks at the factors that impact the effectiveness of building and maintaining a central model. Furthermore, our analysis indicates promising results for accuracy and reactivity that can be used for central decision making. As such, we consider SMOG to be the next step on the stairway towards centralized WSN control.

Acknowledgments

We would like to thank the National University of Singapore and ETH Zurich for providing access to the Indriya and FlockLab testbeds respectively. We also thank Timofei Istomin for helping with testbed experiments and Carlo Alberto Boano for providing the JamLab tool for the interference experiments. This work was funded by Science Foundation Ireland (SFI) under grant 13/IA/1885.

References

[1] M. Zimmerling, F. Ferrari, L. Mottola, T. Voigt, and L. Thiele, "pTunes: Runtime Parameter Adaptation for Low-power MAC Protocols," in *Proc. of IPSN*, 2012.

[2] N. Ramanathan, K. Chang, R. Kapur, L. Girod, E. Kohler, and D. Estrin, "Sympathy for the Sensor Network Debugger," in *Proc. of SenSys*, 2005.

[3] A. Stathopoulos, "Exploiting Heterogeneity for Routing in Wireless Sensor Networks," Ph.D. dissertation, UCLA, 2006.

[4] S. Dawson-Haggerty, A. Tavakoli, and D. Culler, "Hydro: A Hybrid Routing Protocol for Low-Power and Lossy Networks," in *Proc. of SmartGridComm*, 2010.

[5] R. Musaloiu-E., C.-J. M. Liang, and A. Terzis, "Koala: Ultra-Low Power Data Retrieval in Wireless Sensor Networks," in *Proc. of IPSN*, 2008.

[6] International Society of Automation, "Wireless Systems for Industrial Automation: Process Control and Related Applications, ISA-100.11a-2011 Standard," 2011.

[7] HART Communication Foundation, "HART Field Communication Protocol Specification, Revision 7.5," 2013.

[8] C. Lu, A. Saifullah, B. Li, M. Sha, H. Gonzalez, D. Gunatilaka, C. Wu, L. Nie, and Y. Chen, "Real-Time Wireless Sensor-Actuator Networks for Industrial Cyber-Physical Systems," *Special Issue on Industrial Cyber-Physical Systems*, vol. 104, no. 5, 2016.

[9] M. Doddavenkatappa, M. C. Chan, and A. Ananda, "Indriya: A Low-Cost, 3D Wireless Sensor Network Testbed," in *Proc. of TridentCom*, 2011.

[10] R. Lim, F. Ferrari, M. Zimmerling, C. Walser, P. Sommer, and J. Beutel, "FlockLab: A Testbed for Distributed, Synchronized Tracing and Profiling of Wireless Embedded Systems," in *Proc. of IPSN*, 2013.

[11] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, and R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks," RFC 6550, March 2012.

[12] T. Narten, E. Nordmark, W. Simpson, and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)," RFC 4861, September 2007.

[13] A. Jhumka and L. Mottola, "On Consistent Neighborhood Views in Wireless Sensor Networks," in *Proc. of SRDS*, 2009.

[14] B. H. Bloom, "Space/Time Trade-offs in Hash Coding with Allowable Errors," *Commun. ACM*, vol. 13, no. 7, 1970.

[15] S. Duquennoy, O. Landsiedel, and T. Voigt, "Let the Tree Bloom: Scalable Opportunistic Routing with ORPL," in *Proc. of SenSys*, 2013.

[16] M. Zuniga and B. Krishnamachari, "An Analysis of Unreliability and Asymmetry in Low-power Wireless Links," *ACM Trans. Sen. Netw.*, vol. 3, no. 2, 2007.

[17] C. A. Boano, T. Voigt, C. Noda, K. Roemer, and M. Zuniga, "JamLab: Augmenting Sensornet Testbeds with Realistic and Controlled Interference Generation," in *Proc. of IPSN*, 2011.

[18] A. Dunkels, "The ContikiMAC Radio Duty Cycling Protocol," Swedish Institute of Computer Science, Tech. Rep. T2011:13, December 2011.

[19] B. A. Nahas, S. Duquennoy, V. Iyer, and T. Voigt, "Low-Power Listening Goes Multi-Channel," in *Proc. of DCOSS*, 2014.

[20] F. Ferrari, M. Zimmerling, L. Thiele, and O. Saukh, "Efficient Network Flooding and Time Synchronization with Glossy," in *Proc. of IPSN*, 2011.