Does Wireless Sensor Network Scale? A Measurement Study on GreenOrbs

Yunhao Liu[1,2], Yuan He[1,2], Mo Li[7], Jiliang Wang[1], Kebin Liu[1,2], Lufeng Mo[3,4], Wei Dong[5], Zheng Yang[1,2], Min Xi[3], Jizhong Zhao[3], Xiang-Yang Li[2,6]
1. CSE Department, HKUST; 2. TNLIST, School of Software, Tsinghua University;
3. CS Department, Xi'an Jiaotong University; 4. Zhejiang Agriculture and Forestry University;
5. CS College, Zhejiang University; 6. CS Department, IIT, USA; 7. SCE, NTU, Singapore

Abstract—In spite of the remarkable efforts the community put to build the sensor systems, an essential question still remains unclear at the system level, motivating us to explore the answer from a point of real-world deployment view. Does the wireless sensor network really scale? We present findings from a large scale operating sensor network system, GreenOrbs, with up to 330 nodes deployed in the forest. We instrument such an operating network throughout the protocol stack and present observations across layers in the network. Based on our findings from the system measurement, we propose and make initial efforts to validate three conjectures that give potential guidelines for future designs of large scale sensor networks. (1) A small portion of nodes bottlenecks the entire network, and most of the existing network indicators may not accurately capture them. (2) The network dynamics mainly come from the inherent concurrency of network operations instead of environment changes. (3) The environment, although the dynamics are not as significant as we assumed, has an unpredictable impact on the sensor network. We suggest that an event-based routing structure can be trained optimal and thus better adapt to the wild environment when building a large scale sensor network.

I. Introduction

Recent advances in low-power wireless technologies have enabled us to make use of wireless sensor networks, a new class of networked systems. Researchers have envisioned a wide variety of applications, from environment monitoring [1], scientific observation [2], to emergency detection [3], field surveillance [4], structure monitoring [5], and etc. In those applications, hundreds or even thousands of sensor nodes are assumed to be deployed in the target field. Beside many algorithmic studies that focus on designing efficient schemes or protocols to coordinate the large scale sensor network, there are also systematic studies that make efforts in optimizing sensor network behaviors in practice, which are usually tested on lab-scale testbeds or small scale deployments. An essential question, however, remains unclear at the system level, motivating us to explore from a point of real world deployment view:

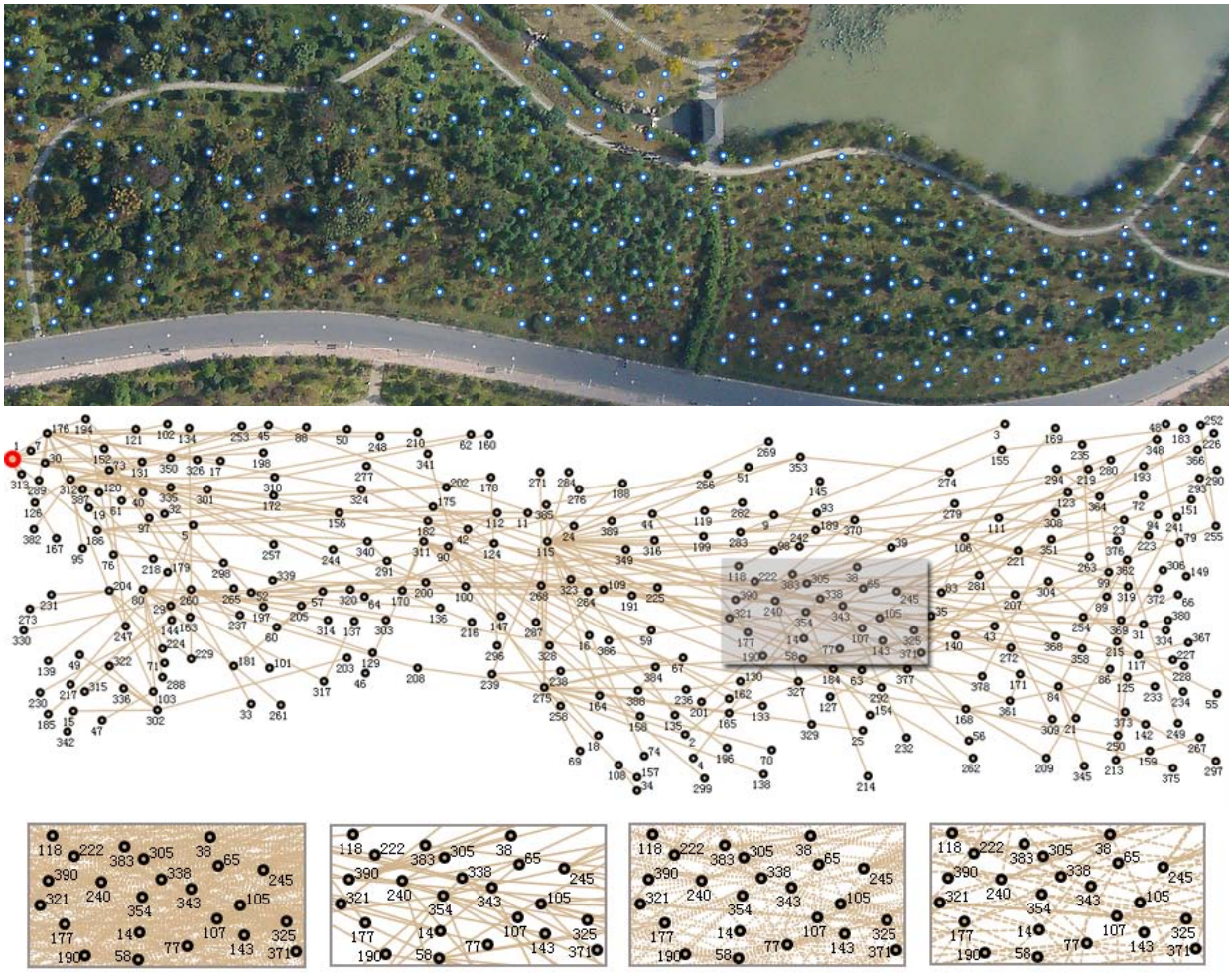
Does the wireless sensor network really scale to contain hundreds or even thousands of nodes that cooperatively work without depleting the limited physical resources, just as it was expected?

There have been several large-scale sensor network deployments reported during the past years, including VigilNet for field surveillance [4], Motelab that provides an indoor testbed [6], SensorScope for weather monitoring in the wild [7], and Trio which enables a large-scale solar-powered sensor network [8]. Those deployments, however, are often highly optimized for specific application needs and not fully leveraged as platforms for consistently observing general network behaviors.

In this work, we conduct a measurement study on GreenOrbs, which is a consistently operating sensor net-work system deployed for the aim of forest monitoring. With up to 330 nodes deployed in the wild, GreenOrbs provides us an excellent platform for observing sensor network behaviors at scale. Figure 1 plots the real topology of the sensor network. The sink is deployed at the upper left corner. Each sensor node is depicted according to its 2-D geographical location. We plot all the communicational links through which data packets are delivered. Although such lined links together with the dotted nodes compose a network-wide topology which traditional algorithm or protocol designers used to play with, we highlight a subgraph within the network and exhibit that the concept of "topology" does vary according to the perspective we look at it. Figure 1(a) exhibits a much denser topology if we take all reachable pairs of nodes into account. Figure 1(b) exhibits the topology with which the network delivers data back. Figure 1(c) exhibits a topology if we select all good links that have RSSI (Received Signal Strength Index) [9] beyond a threshold. Figure 1(d) exhibits a topology if we select those good links with high LQI (Link Quality Indicator) [10] when data packets are transmitted. If we consider different conditions or calibrating criterion, there will be more different types of "topologies", and such "topologies" vary from time to time. Indeed, like many large-scale distributed systems have exhibited, in sensor networks, there grow numerous dynamic behaviors with the concurrent and interactive operations inside the system. Such dynamic behaviors can hardly be fully considered before the system is deployed in the field brimming of unpredictable and unexpected operating conditions.

In this work, we conduct the measurement study on the operating sensor network system deployed in the wild, trying to summarize the critical factors that limit the system scale out of the dynamics on the surface. We instrument such an operating network throughout the protocol stack. We vary the system settings, e.g., the network scale, traffic generation, transmission power level, and test the system behaviors under a variety of conditions.

We present findings across different layers that the system works on. At the physical layer, we present measurements on radio signal strength impacted by wild environment. At the link layer, we measure packet drop/reception, link quality, transmitting rate over the entire network and how they are affected by a variety of system settings. At upper layers, we



(a) (b) (c) (d)

Fig. 1: An overlook on the sensor network deployment and the real topology. (a) The topology of reachable links; (b) The topology of data delivery link chosen by the upper layer routing protocol; (c) The topology of good quality links with RSSI thresholding; (d) The topology of good quality links with LQI thresholding when transmitting data packets.

present observations on routing dynamic, traffic distribution, end-to-end delivery, topological features, and etc.

Our study reveals that traditional opinions on the "hot area" around sink and the instability of links may not be the major concern for large-scale sensor network systems. The physical resources in such networks may have been underestimated and severely underutilized. There is an urgent need to improve current methods in company with those emerging critical factors when the network scales. Our experimental results also suggest us several guidelines that we should carefully consider in designing future protocols for large-scale sensor networks. In particular, the designers should take special care of the phenomena raised from the inherent contention and concurrency of numerous nodes when the sensor network scales, which might be underestimated in existing design concept continued from traditional Wi-LAN or MANET protocol design.

The rest of the paper is organized as follows. In Section II, we describe related work in sensor network deployment and measurement experiences, as well as existing work towards making sensor networks scalable. In Section III, we introduce the background of GreenOrbs, some details of the system implementation, and the measurement methodology. We present our major observations in Section IV. In Section V, we give a comprehensive discussion on how the network is bottlenecked and give guidelines in mitigating such effect. We conclude this work in Section VI.

II. Related Work

In this section we summarize the efforts of research community in building large-scale sensor networks and corresponding measurement studies.

A number of practical network deployments have been reported during the last decade [1], [2], [7], [11]. Although the findings from the above deployments are important, the measurements at this scale, usually tens of sensors, can hardly reveal some network behaviors, such as routing dynamics and topology evolution, which exist only in large-scale networks. Researchers have designed and developed indoor medium-scale testbed such as MoteLab [6] and Kansei [12]. Those indoor testbeds, however, do not fully capture the nuances in realistic environments.

Deploying sensor networks at scale is important because each order of magnitude increase in network size ushers in a new set of unforeseen challenges. VigilNet [4] is designed to support long-term military surveillance using a sensor network consisting of 200 nodes. ExScal [13] is an attempt to deploy a sensor network at "extreme" scale. The system consists of about 1000 sensor nodes and 200 backbone nodes, covering 1300*300 square meters. Dutta et al. [8] report a network deployment "Trio" of 557 solar-powered motes for multi-target tracking. SenseScope [7], [14] is a real-world deployment that took place on a rock glacier, consisting of about 100 sensor nodes.

Most of above mentioned systems are not clearly proper for network measurement due to the following two reasons. First, those systems are organized hierarchically. Such hierarchical architecture inherently alleviates the negative impact induced by large-scales, thus hardly reflecting the performance of general and homogeneous ad-hoc sensor networks. Second, no single system has integrated large-scale (e.g., hundreds of nodes) and long-lifetime (e.g., one year) into a cohesive whole. In other words, those deployments have achieved either scale or lifetime, but usually not both.

In the context of wireless sensor networks, a number of empirical studies present network measurement results in many aspects, with emphasis on understanding the complex and non-ideal behavior of low power wireless communications. Link quality is one of the most important indicators for wireless communication and thus attracts many research efforts [15]–[20]. Srinivasan et al. [9] conclude from measurements on MicaZ motes with CC2420 radios that RSSI is a good estimate of link quality. The authors in [21] study the transition region and quantify its influence. Studies such as [22], [23] emphasize the temporal performance dynamics of wireless links and provide important findings about such phenomenon. The study of beta-factor [24] presents a comprehensive study to quantify and characterize link burstiness. The authors in [25]–[28] investigate radio interference and point out the inaccuracy of range-based interference model [25].

The results presented in those empirical studies are basically obtained from single-purpose and short-lived testbeds of tens of sensors. In contrast, the measurements of GreenOrbs are fairly comprehensive, from low-level radio signal strength and link quality to high-level routing and data traffic issues.

III. GreenOrbs Overview

A. System and Applications

GreenOrbs aims at all-year-round ecological surveillance in the forest, collecting various sensory data, such as temperature, humidity, illumination, and content of carbon dioxide. The collected information can be utilized to support various forestry applications, detailed as follows.

Canopy closure estimates. Canopy closure is defined as the percentage of ground area vertically shaded by overhead foliage. It is a widely-used significant forestry indicator but the traditional measurement techniques have either poor accuracy or prohibitive cost. Based on the readings of illuminance sensors and Monte Carlo Theory, GreenOrbs realizes accurate and efficient canopy closure estimates of vast forest. Using the similar method, another forestry indicator called Leaf Area Index can also be measured by GreenOrbs with sensors deployed in the three-dimensional space.

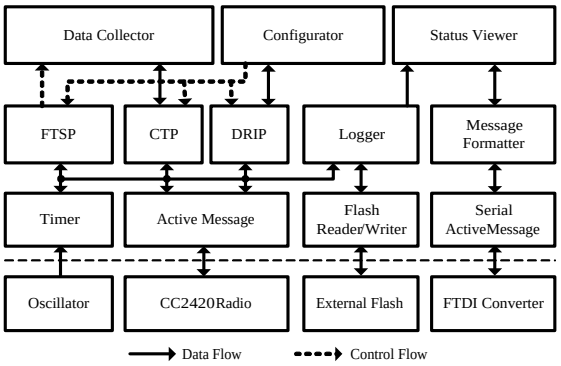


Fig. 2: The diagram of software modules.

Research on biodiversity. The sensor readings of temperature, humidity, illuminance, and carbon dioxide, precisely characterize the forest microclimate. Those data, which quantify the biological activity and multispecies competition, can be utilized to support research on biodiversity.

Carbon sequestration. To maximize the utility of forest carbon sequestration, the capacity of carbon sequestration of different tree species need to be accurately measured. This can be realized with carbon dioxide sensors in the three-dimensional forest space. By comparing the sensor readings at different heights, the amount of carbon dioxide a tree canopy absorbs can be continuously monitored.

Fire risk prediction. Using the sensor data in the forest, namely temperature and humidity, GreenOrbs continuously monitors the environmental, supporting fine-grained real-time fire risk prediction.

GreenOrbs employs the TelosB mote with a MSP430 processor and CC2420 transceiver. The manufactory cost of a GreenOrbs node is 50 US dollars.

The software on the GreenOrbs nodes is developed on the basis of TinyOS 2.1. Figure 2 depicts the design diagram of the software modules. The system mainly carries bi-directional data streams. The mainstream is multi-hop data collection from the ordinary nodes to the sink. The *Data Collector* component based on CTP [29] is employed for this purpose. The rest transmissions are the configuration packets sent from the sink to the ordinary nodes. Hence *Configurator* component based on DRIP [30] is devised to achieve efficient data disseminations. Meanwhile, the FTSP protocol [31] plays the functions of network-wide synchronization, so as to enable the globally synchronized duty cycles. The *Logger* component is in charge of data access (read and write) to the measurement serial flash. The *Status Viewer* component merges and encapsulates the data from the sensors, network, and flash, according to the preconfigured message formats. Such encapsulated messages are transmitted via the serial communication port.

The first GreenOrbs deployment was carried out in July 2008. Ever since then, GreenOrbs has experienced a num-ber of deployments at different places, with different scales, and for different durations.

B. Data Set

The data set used for analysis, evaluation, and experiments in this paper mainly comes from the operational period of

TABLE I: Configurations Used in the Data Set

Trace No.	Network Scale	Power level	Data Rate (pkts/hour)	Duration (hour)	Duty cycle
1	100	15	3	60	No
2	200	15	3	25	No
3	330	15	3	300	No
4	330	15	12	24	No
5	330	15	18	100	No
6	330	15	27	30	No
7	330	15	54	3	No
8	330	15	108	3	No
9	330	31	12	1	No
10	330	21	12	1	No
11	330	15	12	1	No
12	330	8	12	1	No
13	330	15	3	150	8%
14	330	15	60	12	8%

GreenOrbs in December 2009. It contains data of 29 consecutive days, counts 2,540,000 data packets. In order to conduct comprehensive observation on the large-scale sensor network system, during the abovementioned period we have regulated the nodes with different combinations of operational parameters. The detailed configurations of the data set are shown in Table I.

Back End Data Set. The back end data set refers to the entire data set collected at the sink via multi-hop routing, denoted by D_{sink}. D_{sink} is made up of three categories of traces as follows. (1) Routing trace, denoted by $T_{routing}$ and encapsulated as packet of type 41. It mainly records the routing path of a packet, namely the sequence of relaying nodes between the source and the sink. The sensor readings, such as temperature, humidity, illuminance and carbon dioxide, are included in $T_{routing}$ as well. (2) Link trace, denoted by T_{link} and encapsulated as packet of type 42. It includes the list of neighbor node IDs. For each neighbor node, the RSSI, LQI, and ETX (Estimated Transmission Counts) [32] are included in T_{link} as well. (3) Node statistics trace, denoted by T_{stats} and encapsulated as packet of type 45. T_{stats} is a large set of statistical information on each node, including the cumulative time of radio power on, the cumulative number of transmitted and received packets, the cumulative number of packet drops (due to receive pool overflow, d transmit queue overflow, and transmit timeout), the cumulative number of transmissions that are not ACKed, retransmissions, received duplicate packets, and the parent changes with the CTP.

Due to the packet losses and various failures in wireless sensor networks, the back end data set is far from sufficient for characterizing the GreenOrbs system at a full scale. Thus we introduce three out-band measurement techniques, namely overhearing, beaconing, and local logging.

Overhearing. We deploy multiple sniffers in the network to overhear the network traffic. A sniffer is a TelosB mote, which passively listens without sending out any packets. In our early attempt we let the sniffers store all the overheard data in their serial flash. The 1Mbytes flash on TelosB mote was soon found too limited for durative overhearing, so we connect the sniffers to stable and powerful devices, e.g. a laptop, to record all the overheard data. The locations of sniffers are carefully selected so that the combined communication ranges of the sniffers cover the entire network. The data from sniffers are denoted by $D_{sniffer}$.

Beaconing. In many scenarios, we find a number of nodes never successfully report data to the sink, making us fail to find out the cause by using D_{sink} only. Therefore, in some of the experiments, we let each node actively broadcast beacons periodically. The content of the beacon is similar to that in T_{stats} (packet type 45). The broadcast beacons are overheard by the nearby sniffers and stored in $D_{sniffer}$. The neighbor nodes heard the beacon from a node can also use it to update T_{link}.

Local logging. Other than the networking information, the fine-grained local events on the nodes are equally important for us to understand their behavior and interactions. As a necessary complement, every node locally logs events such as transmissions, retransmissions, ACKs of packet receptions. Each event is recorded with six bytes, where two bytes denote the event type and the other four bytes denote the timestamp of an event. The data set of local logging is denoted by D_{log}. Since the deployment is still in operation, we do not collect all the nodes back to read their logs. D_{log} is currently used as a backup data set for diagnosis on some faulty nodes.

C. Measures and Derivations

Yield. We use *yield* [2] to measure the quantity of the collected data. The *network yield* measures the quantity of the entire network while *node yield* measures the quantity of an individual node. Specifically, the *node yield* is calculated by

$$Yield_i = \frac{\text{\# of data pkts received by the sink from } i \text{ during } w}{\text{\# of data pkts sent by } i \text{ during } w}$$

where i is the node ID, and w is a measurement period. The *network yield* is calculated by

$$Yield = \frac{\text{\# of data pkts received by the sink during } w}{\text{\# of data pkts sent by all nodes during } w}$$

Packet Reception Ratio / Loss Ratio. We use *packet reception ratio* (*PRR*) to measure the quality of a link. Throughout this paper, we use two-way link *PRR*, i.e., we consider a successful transmission only if the sender receives an ACK.

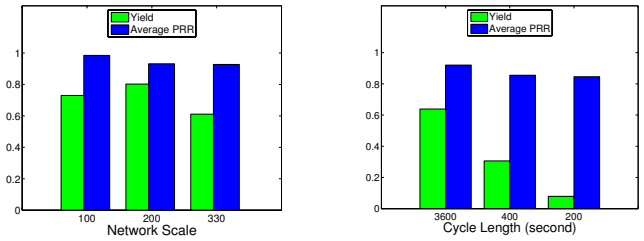
$$PRR = \frac{\text{\# of ACKed data pkts}}{\text{\# of sent data pkts}}$$

The packet loss ratio is $PLR = 1 - PRR$.

Packet Delivery Ratio (*PDR*). *PDR* is defined as the ratio of the amount of packets received by the destination to those sent by the source. Since the transmissions are reinforced with retransmissions, *PDR* can be higher than link *PRR* in practice.

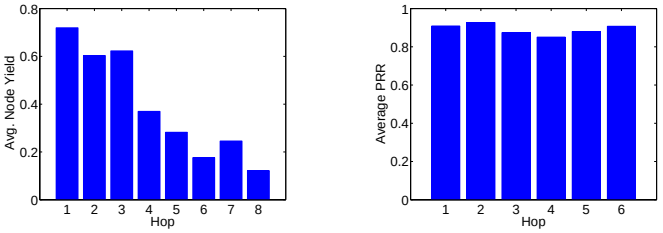
End-to-end delay. The *end-to-end delay* of a packet is the time difference between the sending time at the source node and the reception time at the sink. We stamp each data packets when it is first transmitted from the source node and when it is received at the sink. The FTSP protocol is used to ensure time synchronization.

Correlation Coefficient. Correlation coefficient is a statistical measure of association between two variables, e.g. the ETX value and the packet delivery ratio. The range of correlation coefficient is [-1, 1]. The sign denotes whether two variables are positively or negatively related and the absolute value corresponds to their correlation strength. For example, the correlation coefficient equals to 1 when two variables are in positive linear relationship, -1 in the case of a negative



(a) (b)

Fig. 3: *Network yield* and *PRR* v.s. (a) different network scales, and (b) different traffic loads.



(a) Packet reception ratio (b) Link PRR

Fig. 4: System performance for different categories of nodes.

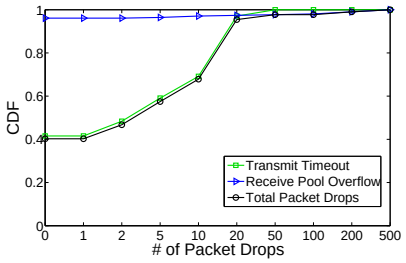


Fig. 5: CDF of the nodes with different numbers of packet drop occurrences.

linear relationship, and 0 when two variables are completely independent.

IV. Basic Observations

In this section, we present a set of basic observations on the operation of the system. Our observations range from the high level system performance down to the detailed behaviors at the link level. From those basic observations, we summarize the network characteristics and explore the reasons that bottleneck the performance when the network scales.

A. Network Characteristics

The *network yield*, the ratio of packets successfully received at the sink side to the total number of packets generated by all the nodes, is a primary metric that evaluates the system performance. It provides us a global indication on how complete the network-wide data are collected. Another metric is link PRR that estimates the percentage of successfully ACKed packets over all the transmissions plus retransmissions, giving us a microscopic indication on how the transmissions perform on the links.

Figure 3(a) exhibits the system performance when the network scales from 100 nodes to 200 nodes, and then to 330 nodes. During the measurement, the data generation rate at each node is three packets per hour. There is not apparent trend of changes on the *network yield*, partially because the traffic inserted into the network is relatively low. On the other hand, the average link *PRR* across all the links does not exhibit apparent difference when the network scales.

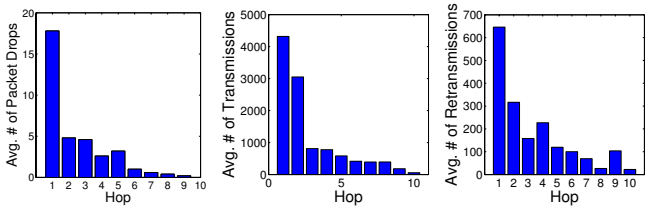
We then measure the same metrics while exerting different traffic load over the network, keeping the network scale as 330. Letting each node generate three packets per cycle, we increase the traffic load in a stepwise manner by shortening the cycle lengths, namely 3600, 400, and 200 seconds. As Figure 3(b) shows, the increasing traffic load severely degrades the system performance. As depicted in Figure 3(b), the *network yield* rapidly drops from over 60% to less than 10%.

A natural question raised from the above observation is: whether the degradation in terms of *network yield* is due to the throughput bottleneck around the sink? Indeed, the "hot area" around the sink has recently been widely reported in a number of literatures. The research communities also propose a variety of protocols to mitigate such a problem [33], [34]. However, if we carefully analyze the data provided by Figure 3(b), we notice that the highest network throughput occurs when the cycle length is set at 400 seconds. The average packet size in GreenOrbs is 100 bytes. The goodput of data reception from the network can be calculated by: $3 \times 330 packets \times 100 bytes \times 8 \times 31\%/400s = 0.61$Kbps. Such a goodput is far less than 250Kbps (the upper bound data rate of TelosB mote) that the sink can provide as a gateway of the networked nodes. This huge gap clearly suggests that the network is far from being bottlenecked before the sink bandwidth is used up. Hence, the follow-up question is: Now that the area around the sink receives relatively high traffic load and severe transmission contentions, is it the place where a large portion of the packet losses occur?

In Figure 4, we present a close look at the system behaviors, using Trace No. 6. We categorize the sensor nodes in the network according to their hop counts to the sink. Note that a node sometimes switches its parent, resulting in dynamic routing paths to the sink of different hop counts. In the statistics, we use a precise granularity to categorize the nodes with such behavior. The packets sent from the same node with different hop counts are separately counted into different categories. Figure 4(a) depicts the packet reception ratio from the nodes of different hop distances to the sink. There is a clear trend that the nodes farther from the sink have a lower *PDR* to the sink. Nevertheless, Figure 4(b) depicts the link *PRR* according to links' hop distances to the sink. There are apparent differences among all the links. This is direct evidence, which reveals that the area around the sink is not the rendezvous of packet losses. Otherwise, the packet reception ratios of different categories of nodes should not deviate in the manner of Figure 4(a). All the packets are likely to be equally dropped around the sink, due to the contention or congestion.

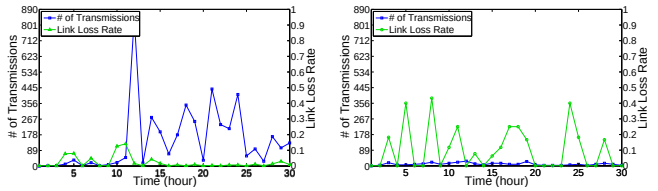
To investigate the cause of packet losses, we further classify the packet losses into three categories:

- Transmit_Timeout: the packet is (re)transmitted 30 times and dropped due to not receiving the ACK signal. Such packet drops are mainly due to the poor quality of the wireless channels or severe collisions during wireless



(a) Packet drops (b) Traffic load (c) Retransmissions

Fig. 6: Node performance of different categories.



(a) (b)

Fig. 8: The traffic and PRR on two typical links.

transmission.

- Receive_Pool_Overflow: the packet is successfully received at the receiver end but immediately dropped due to the forwarding queue overflow. This type of packet drop is mainly caused by the excessively heavy data congestion at the receiver.
- Send_Queue_Overflow: the packet fails to be inserted into the forwarding queue, mainly due to the mismatch between sensor processing capability and the high rate of packet arrival.

We examine all packet losses in Trace No. 5. Among all packet losses, Transmit_Timeout accounts for 61.08% and Receive_Pool_Overflow accounts for the rest 38.92%. No Send_Queue_Overflow is detected.

We further investigate the distribution of packet drop occurrences among the nodes, as shown in Figure 5. The packet drops due to Transmit_Timeout are evenly distributed across different intensities. Nearly 90% nodes have less than 20 times of Transmit_Timeout and no node has more than 50 times of Transmit_Timeout. Surprisingly, we find that over 95% nodes do not have any Receive_Pool_Overflow drop. All the Receive_Pool_Overflow drops (38.92% of all packet drops) occur on less than 5% nodes. Such a finding implies that there exist a very small portion of nodes in the network which play critical roles, taking excessively high traffic load, and responsible for the major portion of packet losses.

B. Investigating Critical Nodes/Links

We take a deep look into the network and investigate the node level behavior. Figure 6 exhibits the individual node performance according to their hop distances to the sink. An intuitive impression is that the nodes near the sink take more traffic load and hence have apparently poorer performance. However, we still cannot conclude that the critical nodes mainly lie near the sink, as Figure 6 only gives us the aggregated performance of many nodes.

Those critical nodes need to be individually identified within the network. For this purpose, Figure 7 plots all the 330 nodes. In total, eight snapshots of eight consecutive operational periods are included. Each node is colored according to the traffic load it takes. A deeper color indicates higher incoming traffic load at a node. The figure sequence clearly shows that there exist a very small portion of nodes that take excessive traffic. It is worth noticing that they are distributed across the entire deployment area instead of concentrated near the sink (the black node in the figures). Further we index the nodes according to their traffic load and find that less than 10% critical nodes commit 80% traffic load and thus 61.06% of the packet loss. They act as bottlenecks of the system and further suggest that such a set of critical nodes are relatively stable.

C. Looking into the Links

As our network-wide statistics suggest, there exist a small portion of critical nodes that bottleneck the performance of the entire network. According to our statistics on different categories of the packet drops, both Transmit_Timeout and Receive_Pool_Overflow contribute a large portion, implying that both congestion and link losses are possible causes that degrade the network performance. We are interested in the reason behind such a phenomenon. A question yet we want to answer is whether the existence of such critical nodes is mainly due to the poor quality of wireless communication, severe congestion or contention accompanied with the unbalanced traffic overhead. To answer this question, we further take a look into the link behavior.

Figure 8 shows the observation on two typical links. We find that the link loss rate fluctuates with time and it seems independent from the traffic load. An immediate guess is that such link dynamics may come from the environmental dynamics. Recall that our system is indeed deployed in the wild. To further explore the link loss fluctuation, we adjust the transmission power of the nodes. Intuitively, as the transmission power is increased, the received signal strength will be strengthened and the link *PRR* will be improved. The level of transmission power is respectively set at 8, 15, 21 and 31 (Traces No. 9–12). In CC2420, they correspond to the sending power of slightly above -15dBm, -7dBm, around -4dBm, and near 0dBm.

The observational results, however, still exhibit consistent fluctuations on many links. The system performance under different settings of transmission power is shown in Figure 9. As the transmission power is regulated, the *network yield* does not change much, remaining at 35%–50%. A higher transmission power does not help to stabilize the link quality, nor does it result in a better *network yield*.

Similar results also hold for the impact of power on end to end delay. We also find that the highest power does not necessary mean the shortest delay.

With such observations we have to carefully reconsider the way we used to view the wireless links in sensor networks. Are they inherently unpredictable with fluctuating quality? If so, are the link fluctuations due to the unpredictable environmental dynamics? Otherwise, assuming the wireless links as indeed good medium for data communications, do the current designs and protocols simply fail to make the best use of them?

V. Who Moved Our Cheese?

As we have experienced from our basic observations, the network cannot unlimitedly scale due to the physical resource

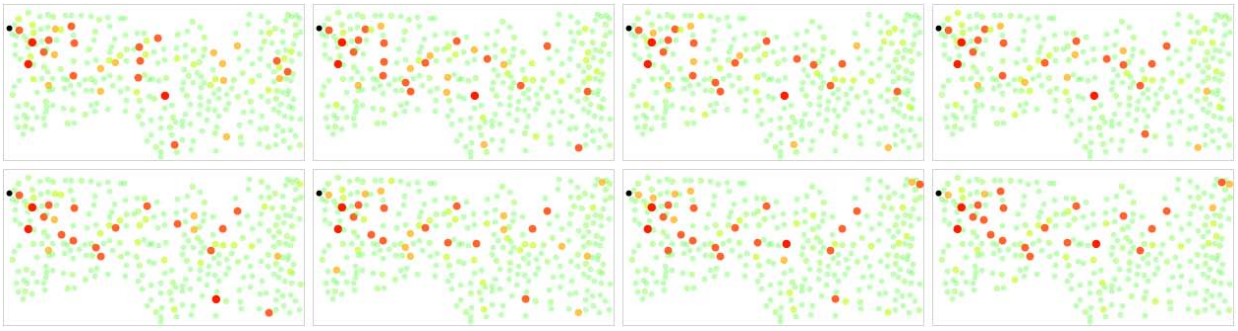


Fig. 7: The traffic distribution over the network.

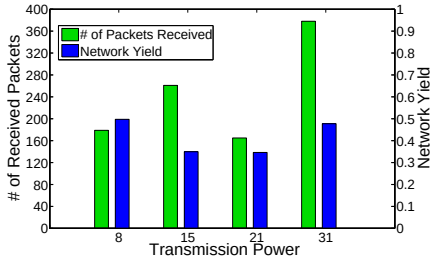


Fig. 9: The system performance under different settings of transmission power.

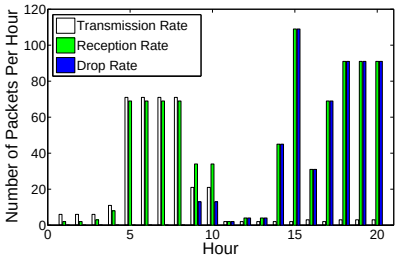


Fig. 10: The packet reception, forwarding, and drop at node 225 within 60 hours.

constraint. In this section, we summarize from our basic observations and try to explore the major factors that limit the system scale. What is the dominant resource that is at the first depleted when the network workload scales? Is such resource appropriately used? Where are the places of resource depletion that bottleneck the entire network? How should existing protocols be improved to adapt to large-scale sensor network characteristics? Bearing those questions in mind, we proactively look into our data traces and conduct a new set of experiments.

A. The Last Straw that Breaks the Camel's Back

As previously shown in Section IV, when the size of the network scales and the traffic load increases, the overall system performance drops, especially after the scale exceeds a limit.

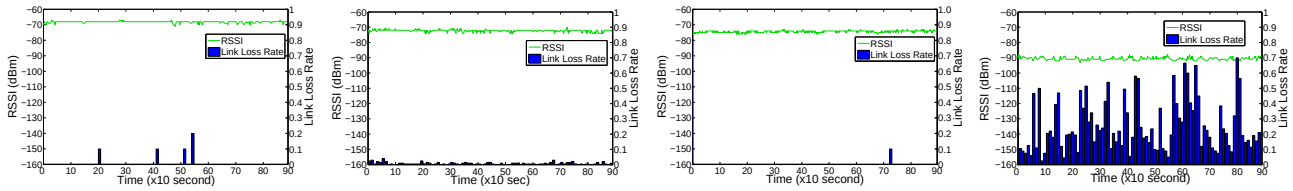
Differing from previous studies, our measurement results suggest that the "hot area" problem around the sink does not play a major role in degrading the system performance. Instead, we observe a set of critical nodes that are distributed across the network. Those critical nodes receive excessive incoming traffic, with fluctuating link loss rates and account for a large portion of packet drops. Current data collection protocols, like CTP with ETX as the routing metric, however, do not seem to successfully handle those cases in time. The routing structure often overreacts to the ETX increases, leading the network traffic concentrated from one area to another, creating "hot" spots from time to time.

In Figure 10, we post a 60 hour statistics on the data forwarding behaviors of a particular node (node 225). In the first half, it exhibits satisfactory performance, forwarding almost all of the incoming packets successfully. Starting some intermediate time point around the 30th hour, this node happens to drop all the incoming data packets while still successfully sending its own data packet. This abnormal behavior is likely related to a software bug that leads to locked memory of the forwarding queue in CTP with special concurrent operations. The real problem is that, even when such a node drops all the incoming packets it receives, it is still selected as the parent in the routing tables of many nodes for the rest of time. Such a phenomenon is largely due to the fact that the ETX indicator does not capture packet drops on a forwarding node. The ETX measured at node 225 is always good and broadcasted to its neighboring nodes, consistently absorbing the traffic and dropping them. Against such a problem, an aggregated indicator is urged, which should reflect both link quality and node's forwarding quality.

Thus our first conjecture is that: the bottlenecks in a large-scale sensor network does not necessarily lie in the "hot area" around the sink. It is likely that some of the intermediate nodes bottleneck the entire network while the existing widely used indicators may not accurately capture them.

B. How Dynamic Is the Environment?

According to our observations, the No_ACK_drop contributes the largest portion of packet drops. In fact, many existing works have reported the possibility of environmental dynamics that affect link quality. To validate our guess, we conduct an independent set of experiments. We place two sensor nodes in the same environment where GreenOrbs is deployed and measure the link quality between them under different settings. The two nodes are placed 20m and 50m apart, respectively. We let one node send data packets and the other receive. Each packet contains 100 bytes payload. The sending rate is set at 1Hz and then 20Hz. We measure the



(a) 1Hz and 20m (b) 20Hz and 20m (c) 1Hz and 50m (d) 20Hz and 50m

Fig. 11: The RSSI and link loss rate measured at an independent pair of nodes.

RSSI and link loss rate at the receiver. Each set of experiment is conducted three rounds at different time of a day, and each round lasts for 1000 seconds.

As Figure 11 suggests, the RSSI, which is a major indicator that measures the quality of propagated signal is relatively stable through most of the time and settings. The fluctuation of RSSI is mostly composed of a series of sparkling burrs. That is quite possible from the interference from nearby 802.11b AP signals, as reported in [23]. Only in the experiment of 50m distance and 20Hz sending rate, the RSSI varies around -90dBm, and there appear some observable link losses. This is mainly because the CC2420 transceiver has a receiving sensitivity at around -93dBm to -87dBm [10].

The above measurement results indicate that the signal propagation in the wild is not as dynamic as we imagined. Recall that in our observations on the link performance, the link loss rate fluctuates far more intensively (see Figure 8) and there is not an apparent correlation between the link loss rate and the traffic overhead on that link. However, as our network-wide statistics suggest, the high link loss usually occur at those nodes within or near high traffic regions. Such observations suggest that the fluctuating link loss is due to the collisions of concurrently transmitted packets in those regions. Such contentions are not effectively detected by the CSMA mechanism, resulting in improper concurrency. Considering the dense deployment of sensor networks, the traditional wireless "hidden terminal" problem might be far more popular than else where like 802.11 AP network or MANET where the network is usually of a small density.

Thus our second conjecture is that: most of the wireless links used in sensor networks are physically stable. The dynamics of sensor networks do not mainly come from the external environment but the internal network operations. The inherent concurrency of operations among different nodes should be further investigated and considered in designing scalable network protocols.

C. Adaptive Routing Design

While our measurement results reflect that the environment introduces very limited dynamics to the network, the impact from the deployment environment itself is non-negligible.

A general manner of sensor nodes deployment is to place them uniformly across the field, aiming to provide a uniform networking infrastructure. As explicitly shown in Figure 1, however, the resulting networking infrastructure of GreenOrbs does not match the expectation. Some nodes have excessive neighbors and forward much more data than others. Some of them become critical nodes later, bottlenecking the network performance. We fail to achieve logical uniformity from geological uniformity, largely due to the inherent irregularity of the deployment environment. The bumpy floor in the wild, woods standing in between, slope of the hill, and etc., all environment factors make the signal propagation irregular. Only after the networking characteristics are thoroughly studied after deployment, we are able to provide customized schedule in the routing layer that optimizes the system performance.

While current dynamic routing approaches aim to be adaptive to the network dynamics, they usually lack tailored optimization in adapting to the surrounding environment. Besides, according to our observations, the environment impacts are relatively stable, providing us adequate room in designing comparatively stable while highly optimized routing protocols.

Figure 12 exhibits our preliminary attempt in support our argument. During the system operation, we let the network first run with CTP routing for 120 minutes. Then we fix the routing tables for another 120 minutes, letting each node forward packets to a fixed parent node. We do not observe apparent difference between the performance of *network yield* in the two working periods.

We believe, with careful consideration on the actual network structure under the practical environment and an intelligent learning process, it is very possible that a highly optimized static routing structure outperforms existing dynamic routing approaches in a large-scale sensor network. Moreover, a static routing structure can be made adaptive to the environment changes on an event-triggered basis. The routing structure will only be reconstructed when sharp events happen like intensive weather changes, large relief variations, a broad area of sensor damages, and etc, and after adequate knowledge about the new environment is learnt.

Thus our third conjecture is that: the environment, while with less dynamics than we expected, has an unpredictable impact on the sensor network system running under it. Current dynamic routing approaches usually lack adaption to the surrounding environment without adequately learning its unique characteristics. We suggest that an event-based static routing structure may have better performance in operating a large-scale sensor network in the wild environment.

VI. Conclusions

In this work we conduct a measurement study on a large-scale operating sensor network system, GreenOrbs, with up to 330 nodes deployed in the wild. We aim to comprehensively understand how the sensor network performs when it scales to contain hundreds or even thousands of nodes. We instrument such an operating network throughout the protocol stack. The contribution of this work is twofold.

First, to the best of our knowledge, we are the first to conduct a long term and large-scale measurement study on an operating sensor network in the wild. We present observations

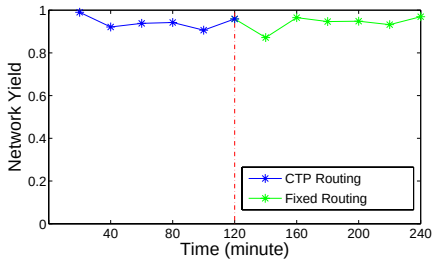


Fig. 12: The performance comparison between original CTP routing and that when the routing table is fixed.

across a variety of layers in the network that provide research community empirical experiences on how practical problems affect when the sensor network scales.

Second, based on our basic findings from the system measurement, we further propose and initially attempt to validate three conjectures that provide guidelines for future algorithm and protocol designs with larger scale sensor networks. In summary, (1) we think it might be very possible that some of the intermediate nodes bottleneck the entire network, and most of currently used indicators may not accurately capture them; (2) most of the wireless links in large scale sensor networks are physically stable. The dynamics mainly come from the inherent concurrency of network operations which should be further investigated and considered in designing scalable network protocols; (3) the environment, while with insignificant dynamics, has an unpredictable impact on the sensor network under it. We suggest that an event based routing structure can be trained optimized and thus better adapt to the wild environment when building a large-scale sensor network.

Acknowledgments

This work is supported in part by NSFC/RGC Joint Research Scheme N_HKUST602/08, National Basic Research Program of China (973 Program) under Grants 2010CB328000 and No. 2011CB302705, COE_SUG/RSS_20 Aug2010_13/14 in Nanyang Technological University of Singapore, the NSFC under Grant No. 60803152, NSF CNS-0832120, NSF CNS-1035894, National Natural Science Foundation of China under Grant No. 60828003, program for Zhejiang Provincial Key Innovative Research Team, and program for Zhejiang Provincial Overseas High-Level Talents (One-hundred Talents Program).

References

[1] G. Tolle, J. Polastre, R. Szewczyk, D. Culler, N. Turner, K. Tu, S. Burgess, T. Dawson, P. Buonadonna, D. Gay, and W. Hong, "A Macroscope in the Redwoods," in *Proc. of ACM SenSys*, 2005.

[2] G. Werner-Allen, K. Lorincz, J. Johnson, J. Lees, and M. Welsh, "Fidelity and Yield in a Volcano Monitoring Sensor Network," in *Proc. of OSDI*, 2006.

[3] M. Li and Y. Liu, "Underground Coal Mine Monitoring with Wireless Sensor Networks," *TOSN*, vol. 5, no. 2, pp. 1–29, 2009.

[4] T. He, P. Vicaire, T. Yan, Q. Cao, G. Zhou, L. Gu, L. Luo, R. Stoleru, J. A. Stankovic, and T. F. Abdelzaher, "Achieving Long-Term Surveillance in VigilNet," *TOSN*, vol. 5, no. 1, pp. 1–39, 2009.

[5] N. Xu, S. Rangwala, K. K. Chintalapudi, D. Ganesan, A. Broad, R. Govindan, and D. Estrin, "A Wireless Sensor Network For Structural Monitoring," in *Proc. of ACM SenSys*, 2004.

[6] G. Werner-Allen, P. Swieskowski, and M. Welsh, "MoteLab: A Wireless Sensor Network Testbed," in *Proc. of ACM/IEEE IPSN*, 2005.

[7] G. Barrenetxea, F. Ingelrest, G. Schaefer, M. Vetterli, O. Couach, and M. Parlange, "SensorScope: Out-of-the-Box Environmental Monitoring," in *Proc. of ACM/IEEE IPSN*, 2008.

[8] P. Dutta, J. Hui, J. Jeong, S. Kim, C. Sharp, J. Taneja, G. Tolle, K. Whitehouse, and D. Culler, "Trio: Enabling Sustainable and Scalable Outdoor Wireless Sensor Network Deployments," in *Proc. of ACM/IEEE IPSN*, 2006.

[9] K. Srinivasan and P. Levis, "RSSI is Under Appreciated," in *Proc. of EmNets*, 2006.

[10] *CC2420 data sheet: http://focus.ti.com/lit/ds/symlink/cc2420.pdf*.

[11] T. He, P. Vicaire, T. Yan, Q. Cao, G. Zhou, L. Gu, L. Luo, R. Stoleru, J. A. Stankovic, and T. F. Abdelzaher, "Achieving Long-Term Surveillance in VigilNet," in *Proc. of IEEE INFOCOM*, 2006.

[12] E. Ertin, A. Arora, R. Ramnath, M. Nesterenko, V. Naik, S. Bapat, V. Kulathumani, M. Sridharan, H. Zhang, and H. Cao, "Kansei: A Testbed for Sensing at Scale," in *Proc. of ACM/IEEE IPSN*, 2006.

[13] A. Arora, R. Ramnath, E. Ertin, and et al., "ExScal: Elements of an Extreme Scale Wireless Sensor Network," in *Proc. of IEEE RTCSA*, 2005.

[14] G. Barrenetxea, F. Ingelrest, G. Schaefer, and M. Vetterli, "The Hitchhiker's Guide to Successful Wireless Sensor Network Deployments," in *Proc. of ACM SenSys*, 2008.

[15] A. Woo, T. Tong, and D. Culler, "Taming the Underlying Challenges of Reliable Multihop Routing in Sensor Networks," in *Proc. of ACM SenSys*, 2003.

[16] J. Zhao and R. Govindan, "Understanding Packet Delivery Performance in Dense Wireless Sensor Networks," in *Proc. of ACM SenSys*, 2003.

[17] D. Ganesan, D. Estrin, A. Woo, and D. Culler, "Complex Behavior at Scale: An Experimental Study of Low-Power Wireless Sensor Networks," UCLA and UC Berkeley, Tech. Rep., 2002.

[18] S. Lin, G. Zhou, K. Whitehouse, Y. Wu, J. A. Stankovic, and T. He, "Towards Stable Network Performance in Wireless Sensor Networks," in *Proc. of IEEE RTSS*, 2009.

[19] T. Liu, A. Kamthe, L. Jiang, and A. Cerpa, "Performance Evaluation of Link Quality Estimation Metrics for Static Multihop Wireless Sensor Networks," in *Proc. of IEEE SECON*, 2009.

[20] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris, "Link-level Measurements from an 802.11b Mesh Network," in *Proc. of ACM SIGCOMM*, 2004.

[21] M. Zuniga and B. Krishnamachari, "Analyzing the Transitional Region in Low Power Wireless Links," in *Proc. of IEEE SECON*, 2004.

[22] A. Cerpa, J. L. Wong, M. Potkonjak, and D. Estrin, "Temporal Properties of Low-Power Wireless Links: Modeling and Implications on Multi-Hop Routing," in *Proc. of ACM MobiHoc*, 2005.

[23] K. Srinivasan, P. Dutta, A. Tavakoli, and P. Levis, "Understanding the Causes of Packet Delivery Success and Failure in Dense Wireless Sensor Networks," Stanford University and UC Berkeley, Tech. Rep., 2006.

[24] K. Srinivasan, M. A. Kazandjieva, S. Agarwal, and P. Levis, "The Beta Factor: Measuring Wireless Link Burstiness," in *Proc. of ACM SenSys*, 2008.

[25] G. Zhou, T. He, J. A. Stankovic, and T. Abdelzaher, "RID: Radio Interference Detection in Wireless Sensor Networks," in *Proc. of IEEE INFOCOM*, 2005.

[26] R. Maheshwari, S. Jain, and S. R. Das, "A Measurement Study of Interference Modeling and Scheduling in Low-power Wireless Networks," in *Proc. of ACM SenSys*, 2008.

[27] R. Gummadi, D. Wetherall, B. Greenstein, and S. Seshan, "Understanding and Mitigating the Impact of RF Interference on 802.11Networks," in *Proc. of ACM SIGCOMM*, 2007.

[28] L. Qiu, Y. Zhang, F. Wang, M. K. Han, and R. Mahajan, "A General Model of Wireless Interference," in *Proc. of ACM MobiCom*, 2007.

[29] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis, "Collection Tree Protocol," in *Proc. of ACM SenSys*, 2009.

[30] G. Tolle and D. Culler, "Design of an Application-Cooperative Management System for Wireless Sensor Networks," in *Proc. of EWSN*, 2005.

[31] M. Maróti, B. Kusy, G. Simon, and Ákos Lédeczi, "FTSP: The Flooding Time Synchronization Protocol," in *Proc. of ACM SenSys*, 2004.

[32] D. S. J. D. Couto, D. Aguayo, J. Bicket, and R. Morris, "A High-throughput Path Metric for Multi-hop Wireless Routing," in *Proc. of ACM MobiCom*, 2003.

[33] S. Olariu and I. Stojmenovic, "Design Guidelines for Maximizing Lifetime and Avoiding Energy Holes in Sensor Networks with Uniform Distribution and Uniform Reporting," in *Proc. of IEEE INFOCOM*, 2006.

[34] X. Wu, G. Chen, and S. K. Das, "Avoiding Energy Holes in Wireless Sensor Networks with Nonuniform Node Distribution," *TPDS*, vol. 19, no. 5, pp. 710–720, 2008.