Scalability of Time Synchronized Wireless Sensor Networking

Samuel Zats, Richard Su, Thomas Watteyne*, Kristofer S.J. Pister
BSAC, University of California, Berkeley, USA
szats@berkeley.edu, {yukuwan,watteyne,pister}@eecs.berkeley.edu
*now with Dust Networks

***Abstract*—Existing commercial wireless sensor network solutions use Time-Synchronized Channel Hopping (TSCH) to achieve an end-to-end reliability higher than 99.9% and industry-accepted network lifetime (5-10 years on batteries). In these types of networks, once nodes synchronize, they follow a schedule which determines the time and frequency of the channel that is used to transmit and receive. Standards such as WirelessHART apply this technique. This paper addresses the ability to scale such a scheduling approach for specific applications.**

Specifically, this paper demonstrates the ability to create a successful schedule for a network consisting of 10,000 nodes within a $0.1km^2$ area, an equivalent density to one million nodes, deployed in a $10km^2$ area. Each node reports a sampled measurement every $10s$. Such environmental requirements are common for industrial plants, where individual elements are equipped with various sensors (e.g., vibration, pressure, temperature, flow, tank level and corrosion).

Given these typical network densities and empirical propagation models, we develop a targeted open-source end-to-end network and packet simulator to model mote position, connectivity and routing to create a schedule which yields collision-free network operation. We show that such a schedule can be built, and determine the minimal number of sink nodes needed in such a network. The schedule is verified by packet flow simulation to assess expected packet reliability, delay and power consumption.

I. INTRODUCTION

Wireless sensor networks can be deployed for fractions of the installation and maintenance costs required by wired solutions. These fully wireless solutions face challenges from reliability and lifetime. Reliability is challenged by the unpredictable nature of wireless, lifetime by the fact that nodes are mostly dependent on battery power.

Time Synchronized Channel Hopping (TSCH) is a mediim access technique that has been able to address these challenges. By synchronizing, nodes aggressively reduce the fraction of time their radio is actively on (radio duty cycles below 1% are commonplace) and thereby increase their lifetime (5-10 years on a single AA battery). Channel hopping, a technique by which a pair of neighbor nodes continuously changes the radio frequency used for communication, can overcome external interference and multi-path fading. This yields highly reliable communication, with achievable end-to-end reliability higher than 99.999% [1]. TSCH lies at the foundation of standards such as WirelessHART [2], ISA100.11a [3] and IEEE802.15.4e [4]. Numerous companies sell readily-available products based on these standards, which have become de-facto in industrial automation.

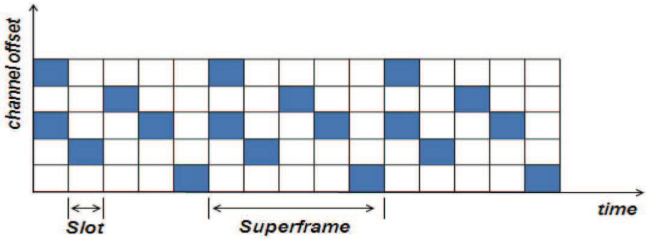

Fig. 1. A representation of a TSCH superframe with 5 slots and 5 channels.

Most networks currently deployed are relatively small, with no more than a few hundred motes [5]. New applications and opportunities have started to emerge. One example is the oil refinery, in which miles of piping are equipped with hundreds of thousands of temperature, pressure, level and corrosion sensors, which are deployed in a relatively small geographical area. The question is whether traditional TSCH networks can scale and be successfully employed in these applications.

Specifically, we consider a network of one million nodes randomly dispersed in a $10km^2$ area, with each node reporting every $10s$. The industry expects a target minimal lifetime of 7 years for a mote powered by a set of 2200 mAh batteries, and a minimal end-to-end reliability of 99.9%. Two issues makes this question non-trivial to answer:

- Is it physically possible to build a schedule that accommodates for such a high density of nodes (1 node per 10 m^2) and one that ensures collision-free operation? This question is detailed in Section I-A.
- How many access points are needed to drain the 100,000 packets per second the network generates? This question is detailed in Section I-B.

A. Collision-free Scheduling

Once synchronized, nodes in a TSCH network follow a time schedule. Time is organized into $10ms$ time slots. L slots form a superframe which indefinitely repeats over time. Fig. 1 depicts a superframe with $L = 5$ (in real deployments, a superframe of tens to hundreds of slots is commonplace). WirelessHART, ISA100.11a and IEEE802.15.4e use IEEE802.15.4-compliant radios, capable of transmitting on one of the 16 orthogonal frequency channels [6]. When following a TSCH schedule, a node knows on which slot to transmit/receive to

978-1-61284-972-0/11/$26.00 ©2011 IEEE

which neighbor, and on which channel.

Every pair of neighbor nodes can be assigned to one or more cells in the superframe. The role of the scheduler is to ensure that pairs of nodes that are within radio range never communicate on the same timeslot and channel, thereby ensuring collision-free communication. The same cell (a time slot and a channel) can be assigned to two pairs of nodes only when they do not interfere. Most commercial solutions use a central scheduler; designing a distributed version remains an open question.

B. Bandwidth Constraints

In a typical network, one or more nodes called access points (APs) collect the data from nearby sensor nodes. With $10ms$ slots, one access point can collect at most 100 packets per second (one packet per slot), which is less than the 100,000 packets generated by the nodes in the network. This paper determines how many APs are needed in our network to ensure low latency, given this bandwidth constraint. The goal is to minimize the number of APs as they are typically higher cost and require wired installations.

C. Simulating at a 1% Scale

For this study, we develop a simulator to model the network[1]. While numerous simulators exist, we develop one targeted for high-density high-volume applications based on WirelessHART. As simulating one million nodes is too computationally demanding, we scale the problem down to 1%, resulting in a wireless network of 10,000 motes within square area of side $316m$. The simulation environment contains a network simulator and a packet simulator.

The network simulator builds the network by positioning the nodes, deciding which nodes are connected, establishing the multi-hop routes and assigning links to cells in a TSCH schedule. The packet simulator simulates the multi-hop transmission of sensor sampled data on that network, according to the TSCH superframe schedule constructed by the network simulator.

The remainder of this paper is organized as follows. Section II details the network simulation. After presenting the simulator, it analyzes the connectivity and hop-count of the obtained network. Section III focuses on packet simulation. It presents the packet simulator, which is then used to determine the reliability, latency and lifetime of the network. Section IV concludes this paper.

II. NETWORK SIMULATION

A. Network Simulator

The Network Simulator models a network of randomly placed motes, determines connectivity, creates routes, and a schedule of its links. It consists of the following steps.

a) Positioning Nodes: APs and motes are randomly positioned throughout the rectangular area (x and y coordinates uniformly chosen). This is representative of our target environment as different processes in industrial automation require greatly varying sensor node densities.

b) Connecting Nodes: determines whether each pair of nodes can communicate based on the distance between and probability of transmission. We use the Friis Transmission Model [7] with an additional randomly-selected path loss constant uniformly chosen between and 0 dB and 40dB. This model accounts for multi-path and interference encountered in typical indoor deployments [1]. The module creates a connectivity N-by-N matrix (with N the number of nodes in the network) which indicates whether or not a pair of nodes can successfully communicate; a 1 indicates nodes can communicate, a 0 indicates nodes can not communicate.

c) Multi-hop Routing: routes are constructed using the upstream algorithm of RPL [8], with a metric which introduces load-balancing between APs. In RPL, every node is assigned a rank and every link a cost. The rank of each AP is set to the number of nodes which send data to it. The routing protocol builds a gradient field from the rank of each node. The further a node from the APs, the larger its rank. Similarly, the more nodes transmit data to an AP, the higher its rank.

We initially assign a rank of 0 to all APs, and a rank of infinity (indicating disconnected state) to all motes. Every node which is part of the network (only the APs at the start) periodically broadcasts an advertisement to its neighbors, indicating its rank. By listening for advertisements, a node builds a list of neighbors, their ranks, and the cost of the link to each neighbor. For each of its neighbors, a node sums the rank of that neighbor with the cost of the link to that neighbor. We call this sum a neighbor's *potential rank*. It then elects the neighbor which yields the lowest potential rank as its routing parent, and use that potential rank as its own rank. Nodes continuously listen for advertisements and evaluate the cost of links to neighbor nodes. As the topology changes, nodes update their ranks and continuously adapt.

Load-balanced least-cost algorithm. We use the upstream routing algorithm of RPL and choose a metric which balances AP load balancing with the minimal number of transmission. The cost of a link is its Estimated Transmission Count (ETX) [9], defined as the inverse of the packet delivery ratio (PDR) of that link. With a PDR of 50%, only half the packets sent are received by the next hop. This means that, on average, a node has to send twice; the ETX of the link is 2. The routing algorithm executes as follows:

1) Assign each AP Node with a cost of 0, sensor node with a cost of infinity.
2) Select a mote to be the Initial Node based on farthest physical distance.
3) Using the connectivity matrix, look up all of the initial node's neighbors and calculate the new cost by the formula in equation (1).
4) Select the node with the least cost.
5) Continue steps 3 and 4, replacing cost assignments with

---

[1] This open-source simulator is available at `http://wsn.eecs.berkeley.edu`.

lower costs until an AP destination is found.
6) Continue steps 2-5 for all potential nodes.

This algorithm outputs a routing table which assigns a parent to each node.

Cost Function. In order to base routings on cost, a cost function is defined as (1):

$$Cost_x = Cost_{current} + ETX_{link} + (\lambda * Load_{AP}). \quad (1)$$

In this equation, the new cost is the sum of current cost plus the ETX of the link, and the AP's current load (sum of all nodes reporting to it) by the Load Factor, λ. λ is calculated by the user-inputted factor, ranging from [0,15] and divided by 200. 200 is selected from the target of 50 APs, resulting in an ideal load balancing of 200 nodes. Since the costs barely exceed 15, the user-inputted load balancing factor is no greater than 15. We assume the PDR of all links which are declared connected in the connectivity matrix to be 80% [10]. This results in a link ETX of 1.25, i.e on average 1.25 transmissions are attempted for each packet successfully delivered.

d) Scheduling: assigns cells in the TSCH schedule according to the multi-hop routes identified by the routing algorithm.

TSCH Scheduling. a schedule consists of a superframe which repeats over time. Nodes are time synchronized and follow the schedule, which indicates on which timeslot and on which channel to transmit/receive to which neighbor node. With 15 available channels[2], this schedule guarantees that 15 communications can happen simultaneously, in a given radio space, without collision and interference.

As per the IEEE802.15.4e standard [4], we simulate time slots of $10ms$. Assuming a IEEE802.15.4 physical layer, this duration allows for both packet transmission and a return acknowledgment. The simulation assumes motes report data every $10s$. We use a $3\times$ provisioning factor, so 3 cells are provisioned in the schedule for each transmission. The extra 2 cells can be used in case the first transmission fails as well as for downstream traffic. The simple way to implement this provisioning factor is to use a superframe of $3.33s$, in which every node gets one opportunity to generate a data packet. This schedule consists of 333 slots. With 15 channels, each superframe consists of 4995 cells.

With 10,000 motes in our 1% simulation, we need to schedule a 10,000 multi-hop paths per superframe. As our superframe contains only 4995 cells, the scheduling algorithm needs to spatially reuse every cell at least twice, depending on average number of hops. That is, schedule the same cell no less than two links to simultaneously transmit, but are not in radio range one from another.

Layered scheduling algorithm. The scheduler assigns links to cells by time first (row-major) layers to evenly distribute transmission over the duration of the superframe. We call a "layer" the number of assignments in a single cell. To avoid interference, a cell is assigned to an additional link if and only if none of the cell's previously scheduled nodes are connected to either of the nodes in the new link, as per the connectivity matrix.

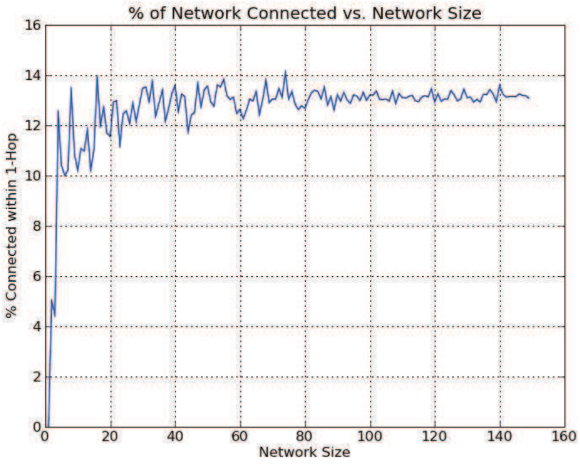

Fig. 2. Average node degree as a function of network size.

The algorithm executes as follows:

1) Loop through all nodes to identify the multi-hop path from each node to one of the APs;
2) Order all paths by decreasing hop count;
3) Assign entire paths one at time, filling first layer of an entire channel for the full period before starting at initial time on following channel;
4) Proceed to schedule links, checking that nodes are available and do not interfere within cell assignments;
5) Continue uniform distribution of links per cell through row-major assignment until all links are scheduled.

B. Resulting Connectivity

Fig. 2 shows the fraction of nodes in a network that an average node is connected to, as a function of the network size. That is, in a network with 140 nodes, each node is connected to 13% of the nodes within a single hop.

We run a series of 100 trials with 10,000 motes randomly placed in a $0.1km^2$ square. Fig. 3 shows the number of nodes that are reachable with every additional meter under the existing connectivity model. Given the incremental number of nodes peaks at 54 meters, we can adjust the transmitting power to minimize power consumption.

The number of gained motes is highest at $54m$. At this distance, our connectivity model produces a 22% likelihood that the nodes are connected. Increasing power to assign nodes that are further apart increases the total potential assignment for a given node but the probability that a single link will drop and never connect increases. Thus, the curve for creating incremental links between two nodes creates a Gaussian-like function with a sharp incline at the lower distances as more

---

[2] Strictly speaking, the IEEE802.15.4 standard [6] supports 16 channels at 2.4GHz. Regulations for the last channel (2.480GHz) vary from country to country, so in practice, commercial TSCH networks operate on 15 channels.

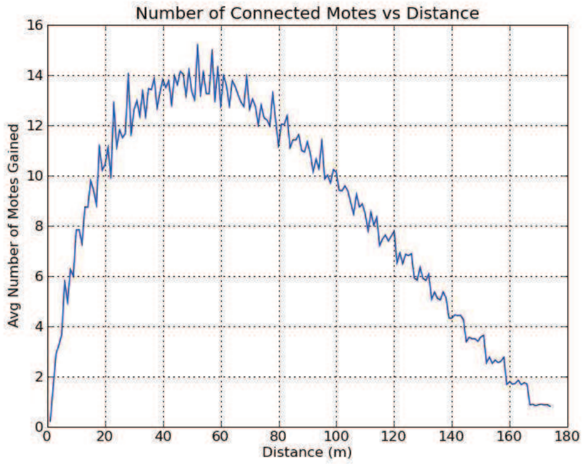

Fig. 3. The average number of connected motes gained per meter as a function of distance.

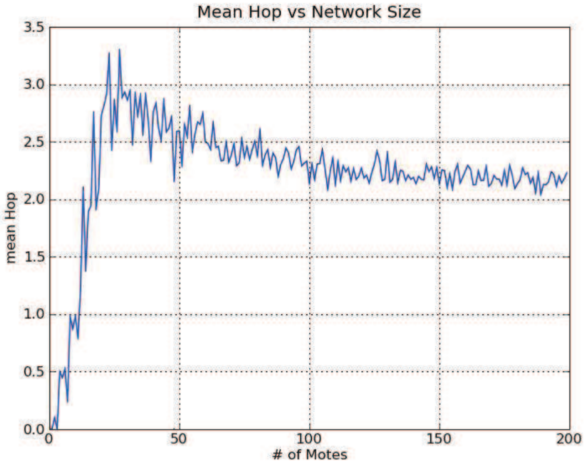

Fig. 4. The average number of hops versus number of motes per AP.

motes are located within the range and the transmission is not yet challenged by distance.

C. Resulting Hop Count

Fig. 4 shows the number of hops between two randomly chosen nodes in the network, as function of network size. This is obtained by averaging over 10 independent runs. In the event two nodes are not connected, they are not considered for the average hop count.

The results suggest that the mean hop fluctuates greatly for small networks. Since we disregard nodes that are disconnected from the network, we see the large run up to roughly 30 motes, reaching as high as 3.3 hops. At that point, the entire network is connected and all nodes are able to reach others in the network. With 120 nodes, the network's mean hop count stabilizes at approximately 2.3 hops per node.

The metric used in our routing algorithm seeks to minimize number of transmissions, while balancing the load among the APs. Fig. 5 is obtained by simulating 10,000 nodes deployed in $0.1km^2$, with $\lambda = 0$. By assigning λ to 0, we only consider the best case conditions with no account for load balancing. APs may become overwhelmed with unbalanced assignments. This allows us to evaluate only link cost and minimize hop count.

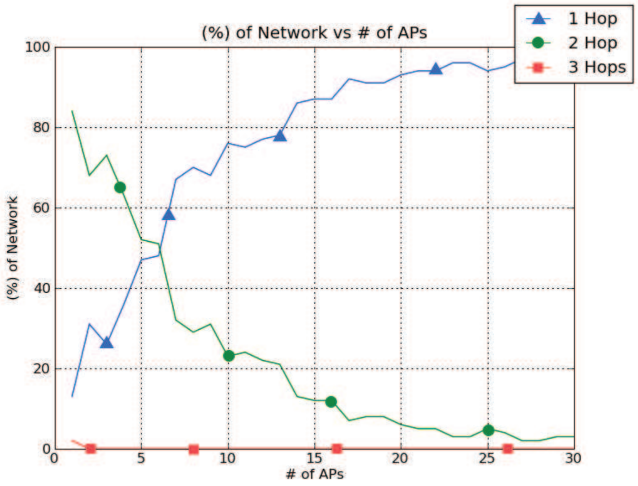

Fig. 5. Fraction of motes at one, two, and three hops from an AP vs. number of APs.

We vary the number of APs, and plot what fraction of the 10,000 nodes is 1 hop away from an AP, 2 and 3 hops away. With 15 APs, 87% of the nodes are a single hop away from an AP, 13% are two hops away. Moreover, 54% of the nodes are connected to an AP, if there are more than 7 of them. Note that, having a large portion of nodes a single hop away from an AP is good from an energy point of view; it does not prevent multi-hop.

III. PACKET SIMULATION

The network simulator positions the nodes, determines the connectivity graph, decides on multi-hop routes from every node to AP, and builds a schedule by populating cells in the TSCH schedule using a 3× over-provisioning factor. The packet simulator injects packets into the network using the resulting schedule. We obtain packet-level statistics such as network reliability, end-to-end latency and average power consumption.

Wireless motes have a limited amount of RAM memory, which means they can only buffer a finite number of packets. It is typical to allocate buffer space for 10 packets. In the IEEE802.15.4 standard, the maximum packet length is 128 bytes, so the allocated buffer space is $1.25kB$. In our simulation, we model a mote's buffer as a First-In-First-Out (FIFO) queue. Each mote generates a packet every $10s$. A mote keeps a packet in its buffer until it receives an acknowledgment from its next-hop neighbor, indicating successful reception. We assume the firmware running on the motes is correct, i.e. motes never reset. Under these circumstances, a packet is lost only when a node's queue is already full when it generates a new

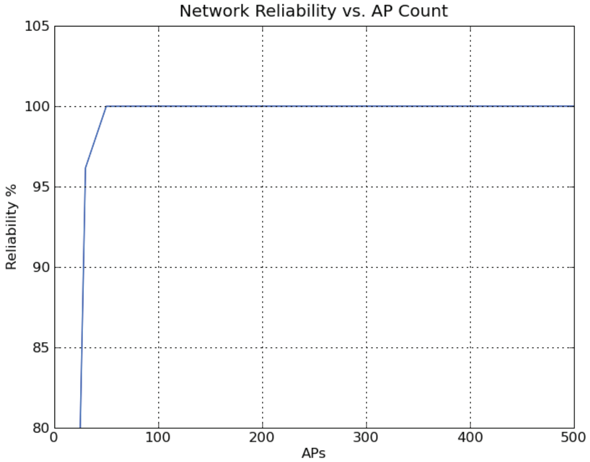

Fig. 6. Network reliability as a function of the number of APs.

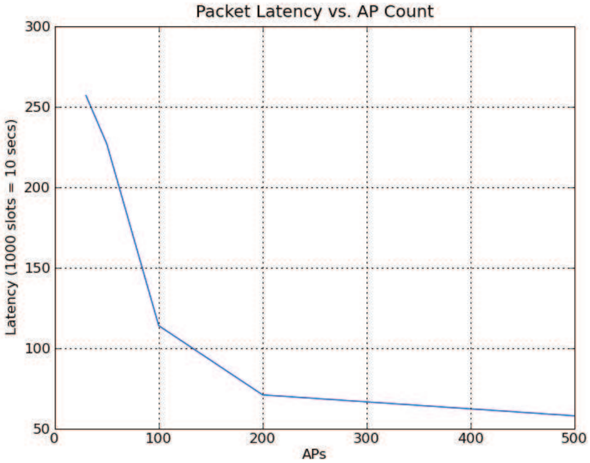

Fig. 7. Packet latencies as a function of the number of APs.

packet.

Simulating packets flowing consists of the following steps. At each time slot:

1) Each mote checks whether it needs to generate a packet. If so, it enqueues a new packet into its buffer. If it is full, it drops the packets, and the corresponding statistics are updated;
2) Each mote reads the schedule to find out whether it has to transmit or receive during that time slot;
3) For each sending mote, if the next-hop's buffer is full, abandon the transmission;
4) For the remaining links, flip a coin weighted by the PDR value of the link. If the transmission is successful, move the packet to the queue of the next hop.

A. Network Reliability

We call network reliability (NR) the ratio of successfully received packets to the overall number of generated packets, as defined in (2). In practice, the simulator keeps track of the dropped packets.

$$NR = ReceivedPackets/(\frac{SimulationTime}{GenerationInterval} * Motes). \tag{2}$$

Fig. 6 shows the network reliability obtained by simulating 300 superframes. While the network has reliability performance issues at low AP counts, the reliability reaches 96% when using 25 APs and over 99.999% when using 50 APs. The network meets our 99.9% reliability goal when using at least 50 APs.

B. End-to-end Latency

We define end-to-end latency as the time elapsed between the instant a mote creates a data packet to report, and the time the packet reliably reaches an AP. If the packet is lost en-route, it is not included for end-to-end latency statistics, but is reflected in Network Reliability. Lower end-to-end latency is important in industrial automation, since the sensor network performs real-time monitoring.

$$PacketLatency = APArrivalTime - GenerationTime \tag{3}$$

In the simulator, each packet is time-stamped both when it is created and when it reaches an AP. Fig. 7 shows the end-to-end latency in time slots (each time slot is $10ms$ long), as a function of the number of APs in the network. These results were obtained by simulating 300 superframes.

End-to-end latency decreases with the number of APs. With more APs in the network, each mote is on average closer to an AP. The routing algorithm causes a node to send data to the AP which is topologically closest; i.e. with more APs the average number of hops of path is smaller. Fig. 7 shows how, from an end-to-end latency of $2.5s$ with 25 APs, the latency drops below $500ms$ when using 130 APs or more.

C. Current Consumption

In each slot, a node can be idle (radio off), listening, transmitting, or receiving. Each of these types of slots requires the radio to be on some portion of the slot, during which it draws some charge from the battery. Table I is built by using the power consumption characteristics of the Texas Instruments CC2520 IEEE802.15.4-compliant radio [11]. By counting the number of idle, listening, transmitting and receiving slots, one can determine the energy consumed by a mote. The CC2520 consume $< 1uA$ in sleep mode [11]; strictly speaking, an idle slot requires $< 0.010\mu C$ of charge, which we round in Table I.

We calculate the current consumption by summing the simulated actions and averaging over the time period to arrive at the current distribution in Fig. 8. Performed for a network of 50 APs and 10,000 motes, the figure represents the average current consumption in micro-amperes (μA) after simulating packets by the proposed schedule from the network simulator. With almost 80% of nodes at $10\mu A$ consumption, this correlates to successfully transmitting a single packet every ten

Type	Radio-on time	Charge drawn
Idle	$0ms$ in Tx, $0ms$ in Rx	$T_i = 0\mu C$
Transmit packet	$3ms$ in Tx, $1ms$ in Rx	$T_x = 100\mu C$
Listen for packet	$0ms$ in Tx, $1ms$ in Rx	$R_i = 25\mu C$
Receive packet	$1ms$ in Tx, $3ms$ in Rx	$R_x = 75\mu C$

TABLE I
CHARGE CONSUMED FOR SINGLE TIMESLOT BASED ON NODE ACTIVITY.

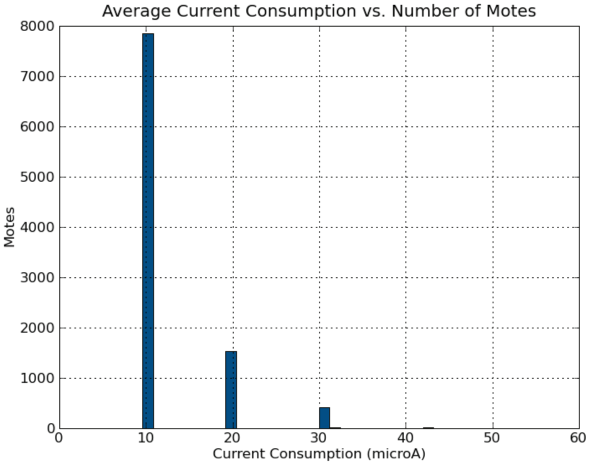

Fig. 8. The distribution of power consumption for the simulated network.

seconds. Subsequently, the greater consumption occurs when nodes require two attempts to transmit, $20\mu A$, as well as those that not only transmit but also must forward a packet from a child, $30\mu A$.

Assuming that each mote contains a set of 2200mAh AA batteries and the worst case scenario of $30\mu A$ for packet transmission, the lifetime of the network of nodes is $2200.10^{-3}/30.10^{-6} = 73.10^3 h$, or 8.4 years. This allows for 16% overhead for additional consumption due to costs for the radio to power up, sleep, and transmit extraneous alert packets that have not been simulated. Thus, this meets our goal of achieving a 7 year lifetime.

D. Scaling Network

We have simulated a network of 10,000 motes deployed in a square area with sides $316m$, and have identified that this network meets the 99.9% reliability and 5-10 years lifetime targets, when using 50 APs. The network's expected latency averages $2.25s$. By tiling 100 of these networks, we can conclude that the same results hold for a one million mote network deployed in $10km^2$, using 5,000 APs.

IV. CONCLUSION

This paper presents a simulator for Time Synchronized Channel Hopping networks which we use to explore the challenges of deploying a large wireless sensor network in industrial automation. We focus on a million motes deployed in an area of $10km^2$. Due to computing limitations, we simulate a 1% downscaled network, while keeping the same density. The simulated network consists of 10000 nodes deployed in a square area of side $316m$. The simulation is divided in network simulation (deployment, connectivity, routing and scheduling) and packet simulation (simulation the flow of packets on the resulting simulator).

We use the upstream routing algorithm of the RPL protocol, together with a routing metric which optimizes both the number of transmissions, and the load-balancing between different access points. The scheduler populates the 4995 cells in a TSCH schedule by assigning each to a link of neighbor nodes, in multiple passes. This schedule evenly distributes links and guarantees collision-free operation.

We show that a million mote sensor network can be deployed in $10km^2$, provided 5,000 APs are used. In this case, the end-to-end latency is $2.25s$, with end-to-end packet delivery radio above 99.9% and a lifetime of 8.4 years on a set of 2200mAh AA batteries. Further enhancements to the work includes both adding greater detail on downstream communications as well as developing redundant paths to ensure reliability.

REFERENCES

[1] K. Pister and L. Doherty, "TSMP: Time Synchronized Mesh Protocol," in *Proceedings of the IASTED International Symposium*, November 2008.
[2] *HART Field Communication Protocol Specifications, Revision 7.1, DDL Specifications*, HART Communication Foundation Std., 2008.
[3] ISA, *ISA-100.11a-2009: Wireless Systems for Industrial Automation: Process Control and Related Applications*, International Society of Automation Std., 11 September 2009.
[4] *IEEE P802.15.4e/D0.01 Draft Standard for Information technology Telecommunications and information exchange between systems Local and metropolitan area networks Specific requirements Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs) Amendment 1: Add MAC enhancements for industrial applications and CWPAN*, IEEE P802.15 Working Group for Wireless Personal Area Networks (WPANs) Std. IEEE Std 802.15.4e, Rev. D0.01/r3, 13 September 2009.
[5] R. Adler, P. Buonadonna, J. Chhabra, F. M., K. L., N. Kushalnagar, L. Nachman, and M. Yarvis, "Design and Deployment of Industrial Sensor Networks: Experiences from the North Sea and a Semiconductor Plant," in *Conference on Embedded Networked Sensor Systems (SenSys)*. San Diego, California, USA: ACM, 2-4 November 2005.
[6] *IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements. Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)*, IEEE Std., Rev. 2006, 8 September 2006.
[7] S. Saunders and A. Aragon-Zavala, *Antennas and Propagation for Wireless Communication Systems, 2nd Ed.*, S. Saunders and A. Aragon-Zavala, Eds. Wiley, 2007.
[8] T. Winter, P. Thubert, A. Brandt, T. Clausen, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, and J.-P. Vasseur, "RPL: IPv6 Routing Protocol for Low power and Lossy Networks," IETF ROLL, IETF Internet-Draft, 2011, draft-ietf-roll-rpl-19 [work in progress].
[9] P. Thubert, "RPL Objective Function 0," IETF ROLL, IETF Internet-Draft, 2011, draft-ietf-roll-of0-10 [work in progress].
[10] S. Lanzisera and A. M. Mehta, "Reducing Average Power in Wireless Sensor Networks Through Data Rate Adaptation," in *Proc. IEEE International Conference on Communication*, June 2009.
[11] *CC2520 Datasheet - 2.4 GHZ IEEE 802.15.4/ZigBee RF Transceiver*, Texas Instruments, December 2007, SWRS068.