Cross-Layer Lifetime Optimization for Practical Industrial Wireless Networks: A Petroleum Refinery Case Study

Michael J. Herrmann and Geoffrey G. Messier

***Abstract*—This paper studies how to maximize the lifetime of ISA100.11a and Wireless HART (WiHART) compatible sensor networks for a petroleum refinery scenario. When accounting for the energy consumption of a typical refinery process sensor and the node micro-processor, only a relatively small percentage of battery energy on average is spent on wireless communication. However, this paper will demonstrate that optimizing network operation can still considerably extend network lifetime. The longest lifetimes are achieved using a new network optimization approach that accounts for the frame structure of ISA100.11a/WiHART. Results are generated using a full network protocol stack simulation that incorporates three different network optimization approaches and includes the energy consumption of the wireless transceiver, sensor and micro-processor.**

***Index Terms*—industrial wireless network, petroleum refinery, network optimization, ISA100.11a**

I. Introduction

Wireless sensor networks (WSNs) have a broad application space [1], [2]. However, the chemical and petroleum refining industry is often overlooked in the wireless network research community as an application area. This is an important omission since refinery sites are extremely rich with sensors and actuators [3] and the cost of installing wired interconnect for communication and power within refineries is very high [4]. This is due to safety requirements that require all wires to be metal shielded. In addition, the wireless highway addressable remote transducer (WiHART) and ISA100.11a wireless instrumentation control standards are mature and well integrated into the specialty sensors and actuators utilized by the petroleum industry [5], [6].

This paper presents an investigation of industrial wireless sensor network lifetime maximization that has practical value for the industrial networking community for two reasons. First, we present cross layer network optimization and scheduling algorithms that can can be used directly by current WiHART and ISA100.11a networks without any modification of the standards. These algorithms attempt to find the power levels and traffic flows in the network that maximize the battery life of nodes in the network by minimizing energy consumption. Our second practical contribution is to evaluate our techniques using a full network protocol stack simulation that includes not just the energy consumption of the wireless transceiver but the microprocessor and sensor as well. Accounting for the total energy consumption of the nodes allows us to provide new insight into how a network optimization must be structured to have an impact on battery lifetime.

The vast majority of WSN studies account only for the energy consumption of the wireless transceiver, often claiming that the transceiver is the largest source of energy consumption. While this may be true for some applications, it certainly is not for the petroleum refinery case study we consider and is likely not true for any industrial network where nodes must use their batteries to power sensors/actuators via the HART current loop interface [7]. We will show that an average WiHART or ISA100.11a sensor node in an petroleum refinery spends approximately 9% of its energy on wireless communications in a typical sampling cycle. With such a small percentage of the energy budget at its disposal, this brings into question whether or not any network optimization algorithm could have a meaningful impact on node battery lifetime.

Addressing this question is the motivation for this paper. Our results will demonstrate that cross-layer network optimization does indeed have a significant impact on node lifetime in a refinery setting. However, the gain achieved by network optimization is closely tied to the definition of network lifetime. Understanding the impact and sensitivities of techniques for extending battery life is an important topic for the petroleum industry since refinery operators expect sensors to operate for at least a year before a battery change [8]. While the battery sizes used by the industry are a considerable 246.2 kJ [9], this battery must power the transceiver, microprocessor and sensor/actuator.

Much of the existing work in cross layer sensor network design assumes a random access medium access control (MAC) layer and/or presents a heuristic that allows nodes working in a distributed fashion to approach optimal network performance [10], [11]. However, the preferred mode of operation for both the WiHART and ISA100.11a standards is to utilize a time division multiple access (TDMA) MAC under the control of a centralized scheduler [6], [12]. The TDMA MAC allows for more deterministic network behavior which is important for control applications. Also, a centralized scheduler can make use of an optimization to find the network schedule then disseminate that schedule to the nodes in the network.

While the number of nodes in a refinery is very large, the sensors are divided into control/actuation domains specific to the different stages in the refining process [8], [13]. Each of these domains are served by a dedicated wired access point that can act as a gateway for the wireless nodes in its domain. This reduces the number of sensors per access point to groups small enough to communicate using a flat, peer-to-peer mesh

network structure rather than having to resort to using a cluster based routing approach more suited to larger networks [14]. The maximum number of wireless nodes that would participate in an ISA100.11a or WiHART network is typically restricted to be in the range of 50 - 100 [6].

The network optimization approach presented in [15] is a very good fit for the petroleum network design problem described above. It assumes a small sized flat mesh network and centrally scheduled TDMA MAC. However, [15] also assumes that traffic flows can be divided at the bit level which is not a practical assumption for WiHART and ISA100.11a. Both of these standards are based on an 802.15.4 physical layer and assume the smallest transmission unit is a frame typically transmitted in a 10 ms timeslot [6]. While the results in [15] can simply be rounded to the nearest frame size, we will demonstrate that this leads to a large number of infeasible schedule solutions. The new optimization approach we present that accounts for frame sizes directly in the optimization significantly increases both the percentage of feasible solutions and the network lifetime itself. To the authors' knowledge, this work is unique since the cross layer wireless network design studies performed to date that are specifically adapted for the WiHART and ISA100.11a protocols have been restricted to latency minimization only [12], [16].

To summarize, our contribution is to present the first cross-layer WSN network lifetime optimization that is fully compatible with existing WiHART and ISA100.11a standards. We are also the first to evaluate our technique by accounting for the complete energy budget of a node used for petroleum refinery sensing. These results will shed light on the value of network optimization for industrial applications and the importance of accounting for the limitation of existing standards when designing these optimizers. Our results are verified using a complete ISA100.11a network protocol stack simulation implemented using the ns3 network simulation package [17].

Section II describes the the petroleum refinery monitoring application in more detail and how it influences our assumed network topology. Section III describes the hardware of the sensor nodes, node energy consumption and the structure of the network protocol stack. Section IV presents the algorithms used to optimize and schedule network traffic, including a detailed discussion of how the optimizer results are mapped to the scheduling and routing mechanisms described in the ISA100.11a standard. Finally, Section V presents simulation results evaluating the effectiveness of the algorithms presented in Section IV and concluding remarks are made in Section VI.

In the following, if $\mathtt{X}$ is used to indicate a set or multi-set, then the nth element of that set is represented by $\mathtt{X}(n)$. The closed interval $[a, b]$ represents the set of natural numbers starting at a and ending with b. If $\mathtt{X}$ is an ordered set or multi-set, then it is assumed that the operation $\mathtt{X} = x \cup \mathtt{X}$ inserts an element at the beginning of the set and $\mathtt{X} = \mathtt{X} \cup x$ inserts an element at the end of the set. The notation $\times$ is used to indicate a duplication of elements. For example, $(i, j) \times 3 = \{(i, j), (i, j), (i, j)\}$. Finally, if $x = (a, b)$ then $x_0 = a$ and $x_1 = b$.

II. Monitoring a Petroleum Refinery

Most refinery control systems are relatively stable. Once the refining process converges, actuators (valves, pumps, etc.) require little adjustment [18]. However, the control system may still be required to quickly react to a process disturbance or emergency. As a result, the sensors will continue to sample the process at the control system time constant which is typically on the order of 2 s for most chemical processes [8]. This means that the vast majority of battery energy over the lifetime of a node will be spent on collecting sensor data and sending that data to the access point sink. Thus, in this paper, we will neglect the flow of actuator traffic through the network and focus on sensing only.

The sensing environment in this study will be based on the region of the Shell Canada gas refinery investigated as part of the wireless propagation work in [19]. During the propagation study, measurements were collected within the outdoor "deep-cut" process area which measures approximately 90 m $\times$ 60 m. This area contains a dense concentration of metallic buildings, vessels, distillation stacks, tanks, condensers and reboilers that ranged between 5 m and 30 m in height. The deep-cut region is the type of process area that would be assigned a dedicated WiHART or ISA100.11a mesh network [13].

One of the process units in the deep-cut area is a de-propanizer (deprop) stack, used to separate propane and butane from natural gas. A deprop stack requires a minimum of 5 sensors [18] that are separated by at least 3 m. The deep-cut area can be divided into 10 process units of approximately the same complexity to the deprop stack which means the sensor density for the entire area is approximately 0.0093 nodes/m^2 (50 sensors in the 90 m $\times$ 60 m process area).

III. Network Node and Protocol Model

This section describes the network as implemented using the ns3 simulation framework. The network model assumes a set of N nodes communicating with peer-to-peer wireless links. Nodes are identified by the integers $[0, N]$ where N is the number of sensor nodes and node 0 is the sink. The wireless link and physical layer are described in Section III-A, medium access control (MAC) in Section III-B and routing in Section III-C. Finally, the energy consumption of the network nodes is described in Section III-D.

A. Physical Layer

The channel model for radio propagation between nodes is the standard log-distance path loss model with log-normal shadowing [20]. The path loss exponent and shadowing standard deviation are assumed to be 2.91 dB and 4.58 dB, respectively. It is shown in [19] that these parameters are appropriate for non line-of-sight propagation in an outdoor refinery environment. Small scale channel fading will be very slow or even stationary due to the static nature of the network nodes and the lack of moving objects in the refinery environment. It has also been shown that the coherence bandwidth of the refinery channel is smaller than the frequency hopping range of an ISA100.11a device [19]. Thus, it is assumed that

this frequency hopping is able to compensate for any small scale variation. As a result, small scale channel effects are not modeled in this study.

The 802.15.4 physical layer mode used by the ISA100.11a device is offset quadrature phase shift keying (OQPSK) spread spectrum operating at 2.45 GHz [21]. The signal has a 2 MHz bandwidth due to a chip rate of 2 Mchip/s and a data rate of 250 kbit/s which means that the spread spectrum provides a linear processing gain of 8. The power of the thermal noise corrupting the signal is -110.8 dBm and the probability of frame error is given by $1-(1-P_b)^{N_b}$, where N_b is the number of data and overhead bits in the frame and P_b is the probability of bit error for OQPSK calculated using the received signal to noise ratio. It is assumed the data payload would not exceed 40 bytes since a data frame will typically carry only a single sensor sample plus a small amount of overhead used by the refinery instrumentation application.

The ISA100.11a standard provides a mechanism to feed back received signal power to the transmitter in order to facilitate transmit power control [22]. Therefore, the transmit powers of the nodes in the simulation are adjusted so that received power is maintained at a target level of -101 dBm, which is the power level required to maintain a 1% frame error rate on a typical 802.15.4 transceiver chip [23].

An automatic repeat request (ARQ) retransmission scheme is not implemented in the simulation. Since errors cannot be predicted, a buffer time would be required in the TDMA schedule to allow the retransmissions to occur. This would significantly reduce node sleep time and degrade the energy efficiency of the network. Not implementing ARQ is also an acceptable design choice since it has been shown in [24] that stability in a networked control system can still be achieved in a "best-effort" style network that does not utilize frame retransmissions.

B. Medium Access Control

The MAC layer implements TDMA using the superframe structure described in [22] which consists of L timeslots of duration T_s. WiHART networks utilize a fixed T_s = 10 ms which is also cited in [22] as a typical value for ISA100.11a . However, ISA100.11a allows T_s to be adjusted. The data packets in this study consist of 40 bytes of data payload and an additional 29 bytes of overhead [21], [22]. This gives a total frame length of N_b = 552 bits which corresponds to a frame duration of 2.2 ms. Therefore, the minimum value of T_s that can be used by ISA100.11a is 4.5 ms which equals the 2.2 ms frame duration plus 2.3 ms which is specified in [22] as the required amount of idle time at the start of a slot.

The TDMA schedule during the superframe is represented by the ordered multiset $\mathtt{Schd}$ where $|\mathtt{Schd}| \leq L$ and the lth element of the set is the 2-tuple $\mathtt{Schd}(l) = (i, j)$ which indicates that nodes i and j are scheduled to transmit and receive during the lth timeslot, respectively. The contents of $\mathtt{Schd}$ are determined by the algorithms in Section IV.

In general, a TDMA scheme may reuse a timeslot such that two nodes may simultaneously transmit if they are separated by a large enough distance. However, due to the relatively small coverage area described in Section II, we adopt the same small network assumption as [15] where we do not allow timeslot reuse since the large majority of nodes are within range of each other and the benefit of utilizing timeslot reuse would be very limited.

C. Routing

Multi-hop routing in the network simulation is implemented using the source routing mechanism described in [22]. Source routing requires that a node embed an address list in each packet it generates. This list contains the addresses of all the nodes along the multi-hop path to the sink. The packet is sent by the originating node to the first address in the list. When a relaying node receives the packet, it removes its own address from the list in the packet header and forwards the packet to the next node in the list.

The address list inserted by node n into its packet header is represented by the ordered set $\mathtt{A}_n$. The contents of $\mathtt{A}_n, n \in [1, N]$ are determined jointly with the TDMA schedule by the central scheduler using the algorithms in Section IV.

D. Energy Consumption

The main components of a sensor node are the sensor, the processor and the wireless transceiver. The period of a single *operational cycle* of a node is equal to the sampling time of the control process, T_c. An operational cycle consists of the nodes each collecting a single sample and then sending those samples to the sink over the mesh network. At the start of each operational cycle, the processor wakes up and initiates the sensing operation. Once sensing is complete, the processor puts the sensor to sleep and wakes the transceiver to both transmit the sensor data and potentially to act as a relay for other sensors. After the node completes all of its transmission and reception obligations in the TDMA schedule, the processor and transceiver return to sleep mode until the start of the next cycle. The processor and transceiver will also sleep during slots in the TDMA schedule where the node is idle.

The Rosemount 3051 pressure transmitter is taken as representative of the sensors typically used in a refinery environment [25]. Its power consumption is 27 mW which is the mid-range power consumption when the sensor is operating in low power mode. This power is consumed for the entire response time of the sensor which is 100 ms for most pressure ranges.

The radio transceiver (TRX) modeled by the simulation is the Atmel RF233 [23]. It uses a 3 V supply and consumes 0.02 μA of current in sleep mode and 300 μA of current in TRX off mode. Current consumption when the receiver is on is 11.8 mA regardless of whether the TRX is actively receiving data. For a given transmit power, current consumption is determined using a best fit linear interpolation of the current consumption versus transmit power values given in [23] with maximum current draw being 13.8 mA at a transmit power of 4 dBm. The processor keeps the TRX in sleep mode except

for the time slots where it is scheduled to either transmit or receive.

The processor is assumed to be the Atmel ATXmega32D2 which is the family of processor commonly paired with the RF233 [26]. The 32 MHz version of the processor has a supply voltage of 3 V and consumes 7.8 mA in active mode and 2.6 μA in sleep mode.

Note that [23], [25], [26] indicate that the times required to transition between active and sleep modes for the processor, sensor and transceiver are all small relative to the sensing time and slot duration, T_s. Thus, they do not have a significant impact on either energy consumption or the timing of the TDMA schedule. This allows us to assume in the simulation that the transitions between sleep and active modes occur instantly.

Finally, it is possible to use the current and voltage values in this section to make an approximate average energy budget for a typical petroleum refinery sensor node. Assume the node operates as a single hop relay in addition to sending its own data (one receive packet and two transmit packets), superframe timeslots are 10 ms and that the processor is active throughout any sensing and communications. In a single operational cycle, the node will spend 2700 μJ on sensing (43%), 3042 μJ on processing (48%), 178 μJ on transmission (3%) and 354 μJ on reception (6%). Reception is higher than transmission since the transceiver will be in the TRX off state for 5.6 ms of the 10 ms when sending a packet but remains on for the entire 10 ms when receiving.

IV. Network Scheduling Approaches

The purpose of the algorithms in this section is to determine the TDMA transmission schedule $\texttt{Schd}$ and multi-hop path address lists $\texttt{A}_n, n \in [1, N]$ described in Sections III-B and III-C, respectively. The multi-hop paths are determined using three different methods: minimum hop, bit level optimization and packet level optimization. These techniques are discussed in Sections IV-A, IV-B and IV-C, respectively. The optimizations in Sections IV-B and IV-C both seek to maximize network lifetime which is defined as the time until the first node runs out of energy. **As discussed in Section I, the focus of this paper is on moderately sized networks ($\sim$100 nodes or less). As a result, all the optimization problems presented in this section can be solved in an acceptable amount of time without having to resort to sub-optimal heuristics.**

All three path finding methods create a directed graph $G = (V, E)$ representing transmissions through the wireless mesh where the vertices, V, in the graph represent wireless nodes and the edge between nodes i and j, $ij \in E$, represents a communication link from i to j. The weight of that edge, denoted w_{ij}, equals the number of packets or bits that travel through that link in a single operational cycle and an edge $ij \in E$ if and only if $w_{ij} \neq 0$. Once the graph and its edge weights have been determined using the methods in Sections IV-A to IV-C, Section IV-D describes how they are translated into the schedule, $\texttt{Schd}$ and routing address lists, $\texttt{A}_n, n \in [1, N]$. In this section, the set of nodes that transmit to node v is $\texttt{In}_v = \{j | jv \in E\}$ and the set of nodes that v transmits to is $\texttt{Out}_v = \{j | vj \in E\}$. Finally, $\texttt{Rng}_v$ denotes the set of nodes that can receive a wireless transmission from node v.

A. Minimum Hop Routing

This minimum hop routing algorithm is meant to represent the default network configuration for an ISA100.11a network. **The purpose of the algorithm is to find the network routes that minimize the number of hops required by each packet to reach the sink.** While the standard [22] describes routing mechanisms that support multi-hop mesh routing, including the source routing discussed in Section III-C, it does not provide any guidance on how the specific multi-hop paths should be determined. However, [8] suggests that a minimum hop approach to mesh networking is a logical default choice for industrial networks.

Algorithm 1 determines the edge weights for the minimum hop routing graph. The ordered sets $\texttt{Dst} \in \mathbb{N}$ and $\texttt{Pwr} \in \mathbb{R}$ store the number of hops of each node from the sink and the current transmit power of each node, respectively. The elements of the ordered set $\texttt{Fin} \in \{0, 1\}$ are equal to 1 if the corresponding node has been processed as a parent by Algorithm 1, otherwise they are zero. The function FoundNewParent(j, p) returns true if $\texttt{Dst}(j) > \texttt{Dst}(p) + 1$ or if $\texttt{Dst}(j) = \texttt{Dst}(p) + 1$ and $P(j, p) < \texttt{Pwr}(j)$ where $P(j, p)$ denotes the power node j uses to transmit to node p as determined by the power control algorithm described in Section III-A.

Algorithm 1 begins by calling FindMinHopPaths(0) with the initial settings $w_{ij} = 0$ for $i, j \in [0, N]$, $\texttt{Fin}(0) = 1$, $\texttt{Fin}(n) = 0$ for $n \in [1, N]$, $\texttt{Dst}(0) = 0$, $\texttt{Dst}(n) = \infty$ for $n \in [1, N]$ and $\texttt{Pwr}(n) = \infty$ for $n \in [0, N]$.

Algorithm 1 FindMinHopPaths(p)

```
Nxt = {}
for j = 0 ... N do
    if (j ∈ Rng_p) and FoundNewParent(j, p) then
        if Fin(j) = 0 then
            Nxt = Nxt ∪ j
            Fin(j) = 1
        Dst(j) = Dst(p) + 1
        Pwr(j) = P(j, p)
        c = j, n = p
        while c ≠ 0 do
            w_cn = w_cn + 1
            c = n,  n = Out_c(0)
for j ∈ Nxt do
    FindMinHopPaths(j)
```

B. Bit Level Optimized Routing

Bit level optimization (BLO) is the technique presented in [15] for minimizing average energy consumption that we modify in this section to maximize network lifetime. The optimization takes the perspective of being performed over a single superframe of duration LT_s where maximizing network

lifetime is equivalent to minimizing the maximum energy, $E_{\max}$, consumed by a node during the superframe. The variables under optimization are $E_{\max}$ and w_{ij} where w_{ij} is in units of bits. Since all variables being optimized are allowed to be real numbers, the optimization is convex and its cost equation is

$$\min\ E_{\max} \tag{1a}$$

$$\text{s.t.} \sum_{i=1}^{N}\sum_{j=0}^{N} w_{ij}/R_L \leq LT_s \tag{1b}$$

$$\sum_{j=0}^{N} w_{ij} - \sum_{j=1}^{N} w_{ji} = R_D T_c \quad (1 \leq i \leq N) \tag{1c}$$

$$0 \leq w_{ij} \leq \begin{cases} \infty, & j \in \texttt{Rng}_i \\ 0, & \text{otherwise} \end{cases} \quad (1 \leq i \leq N) \tag{1d}$$

$$\sum_{j=0}^{N} E_{T,i} w_{ij} + \sum_{j=1}^{N} E_{R,i} w_{ji} \leq E_{\max} \quad (1 \leq i \leq N) \tag{1e}$$

$$0 < E_{\max} \leq E_I \tag{1f}$$

where R_L is link throughput in bits/sec, R_D is the data generation rate of a sensor node, $E_{T,i}$ is the transmit energy per bit cost for node i and $E_{R,i}$ is the receive energy per bit cost for node i. The values of $E_{T,i}$ and $E_{R,i}$ are calculated using the information in Section III-D. The value of R_D is calculated to include both sensor data and overhead. Constraint (1b) ensures all node data transmissions will fit within a superframe, (1c) and (1d) are the flow constraints, (1e) ensures energy consumption is less than the maximum and (1f) ensures that the max energy consumed during the superframe does not exceed the initial energy allocated to the nodes, E_I.

Once the optimal values of w_{ij} are determined, they are converted to units of frames by rounding up to the nearest frame size according to $w_{ij} = \lceil w_{ij}/N_b \rceil$. While this will always guarantee each node will have a viable path through the network, we will demonstrate in Section V that the unused time slots very quickly cause the TDMA schedule generated from BLO to exceed the maximum number of slots in the superframe since constraint (1b) does not account for rounding up to the nearest frame size. This limits the size of network where BLO can be applied. The fact that the optimization does not directly operate at the frame level also means that it will not achieve the maximum lifetime solution in many cases. This degradation will also be demonstrated in Section V.

C. Frame Level Optimized Routing

Frame level optimization (FLO) is a mixed integer optimization that maximizes network lifetime by calculating the optimal integer values of w_{ij} in units of frames rather than bits. This has two benefits. First, it more accurately reflects the operating conditions of the real network which improves the quality of the solutions. The second benefit is that it tends to result in a more compact TDMA schedule than BLO which allows this method to be used for larger network sizes. The cost equation is written as

$$\min\ E_{\max} \tag{2a}$$

$$\text{s.t.} \sum_{i=1}^{N}\sum_{j=0}^{N} w_{ij} N_b/R_L \leq LT_s \tag{2b}$$

$$\sum_{j=0}^{N} w_{ij} - \sum_{j=1}^{N} w_{ji} = R_D T_c/N_b \quad (1 \leq i \leq N) \tag{2c}$$

$$0 \leq w_{ij} \leq \begin{cases} \infty, & j \in \texttt{Rng}_i \\ 0, & \text{otherwise} \end{cases} \quad (1 \leq i \leq N) \tag{2d}$$

$$\sum_{j=0}^{N} E_{T,i} N_b w_{ij} + \sum_{j=1}^{N} E_{R,i} N_b w_{ji} \leq E_{\max} \quad (1 \leq i \leq N) \tag{2e}$$

$$0 < E_{\max} \leq E_I \tag{2f}$$

$$w_{ij} \in \mathbb{N} \tag{2g}$$

where the constraints (2b) through (2f) serve the same purposes as discussed for (1). The only additional constraint is (2g) which enforces that the edge weights must be integer multiples of a frame.

D. TDMA Scheduling and Source Route Generation

Once the edge weights w_{ij} indicating frame traffic flow between nodes are determined, they are translated into the TDMA schedule, `Schd`, by Algorithm 2. Based on the scheduler in [15], Algorithm 2 schedules transmissions first for the nodes directly connected to the sink and then works outwards towards the network edge. Since new schedule entries are always placed at the front of the schedule, this ensures that all leaf nodes on the network edge transmit first so that relay nodes closer to the sink receive edge node data before they are scheduled to transmit. Specifically, the algorithm begins by placing all nodes with a direct connection to the sink in set Q and scheduling them. Then, it loops through all elements $q \in Q$. If q is a leaf and has no inbound links, its outbound links have been scheduled and it can be removed. If q is not a leaf but all of its outbound links have been scheduled, the nodes connected to its inbound links are added to set Q_0 to be scheduled in the next round of the algorithm and q is removed. Finally, once all the elements of Q have been processed and removed, the next level of the tree is processed by setting $Q = Q_0$.

The purpose of Algorithm 3 is to utilize `Schd` to create the source routing lists described in Section III-C. For each node l, the algorithm traces the multi-hop path in `Schd` from that node to the sink and adds node addresses to $\texttt{A}_l$ for each hop along that path.

V. Results

Section II describes the network topology base case where 50 nodes cover a 60 m $\times$ 90 m area. To investigate the effect

Algorithm 2 TDMASCHEDULER

```
Schd = {}
Q = In_0
for q ∈ Q do
    Schd = {(q, 0) × w_q0} ∪ Schd
repeat
    Q_0 = {}
    while |Q| ≠ 0 do
        for q ∈ Q do
            if |In_q| = 0 then
                Q = Q \ q
            else if (q, j) ∈ Schd ∀ j ∈ Out_q then
                Q_0 = In_q
                for r ∈ In_q do
                    Schd = {(r, q) × w_rq} ∪ Schd
                Q = Q \ q
    Q = Q_0
until |Q_0| = 0
```

Algorithm 3 FINDROUTELISTS

```
for l = 0 ... L − 1 do
    c = Schd(l)_0, n = Schd(l)_1
    if |A_c| = 0 then
        while c ≠ 0 do
            A_l = A_l ∪ n
            c = n
            for r = l + 1 ... L − 1 do
                if Schd(r)_0 = n then
                    n = Schd(r)_1
                    break
```

of varying this topology, it is assumed that network operators will seek to increase the size of the wireless network coverage area over time but that the density of the nodes, which is tied to the instrumentation needs of the refinery process, would remain constant. Thus, as the number of nodes is increased in the simulation from 50 to 100, the coverage area is increased proportionally. The height of the coverage area remains 60 m but the width is set to $90N/50$ m, where N is the number of nodes being simulated. The sink is located on the edge of the coverage area at coordinate (0 m, 30 m). For a given value of N, 200 random topologies are generated and simulated in ns3 where nodes are uniformly distributed over the coverage area with the restriction that the minimum separation between them is 3 m, as described in Section II.

The ns3 simulation implements the ISA100.11a protocol stack as described in Section III. Two values of slot length are used: T_s = 10 ms and T_s=4.5 ms. For an operational cycle time of T_c=2 sec, these slot lengths yield superframe schedules with L = 200 slots and L = 444 slots, respectively. Regarding energy consumption, each of the 200 simulations for a given value of N are conducted with an initial energy of 3.0 J which resulted in simulations long enough to record a statistically significant number of operational cycles. To use the simulated network lifetime values to predict lifetime for the 246.2 kJ batteries typically used for petroleum refinery sensor nodes, the simulated lifetimes were scaled by a factor of 246,200/3.

While the minimum hop algorithm described in Section IV-A always resulted in a viable TDMA schedule, the BLO and FLO algorithms described in Sections IV-B and IV-C were prone to failure. In the case of BLO, the majority of failures were caused by producing a schedule that exceeded the maximum number of superframe slots, L. While FLO would produce schedules that fit within the superframe, its solutions sometimes resulted in nodes without a viable link to the sink. The failure rates of the two algorithms are shown in Fig. 1. Clearly, BLO results in a unacceptable number of failures for even a modest number of network nodes. Running BLO for T_s = 4.5 ms (L = 444) improves this failure rate but only slightly for $N = 60$. The failure rates for FLO are more modest but still reach a significant level for larger networks.

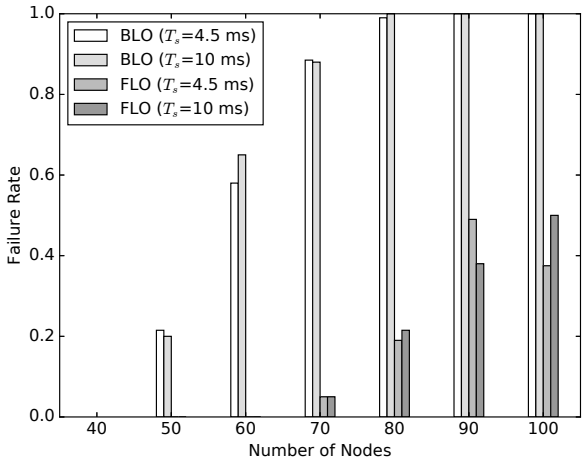

Figure 1. Optimizer failure rate.

Average network lifetime is shown in Fig. 2. Note that BLO results are not shown for the data points where the technique resulted in a failure rate of 100% in Fig. 1. For all three routing strategies, using a 4.5 ms slot time provides a clear improvement over the 10 ms slot time since the shorter slots reduce the amount of time nodes are in receive mode. We also observe FLO resulting in very long network lifetimes even for a large number of nodes. In contrast, the lifetime achieved using minimum hop routing degrades steadily with increasing network size since the minimum hop algorithm tends to overload a small number of nodes near the sink.

In Section III-D is was shown that an average node may only spend 9% of its energy on wireless communication and yet Fig. 2 shows that proper optimization of wireless energy can more than double network lifetime. The key to understanding this apparent disconnect is to note that the definition of network lifetime used in this paper is dependent on maximum energy consumption, not average consumption. This is illustrated in Fig. 3 which shows both average and maximum node energy consumption per 2 sec operational

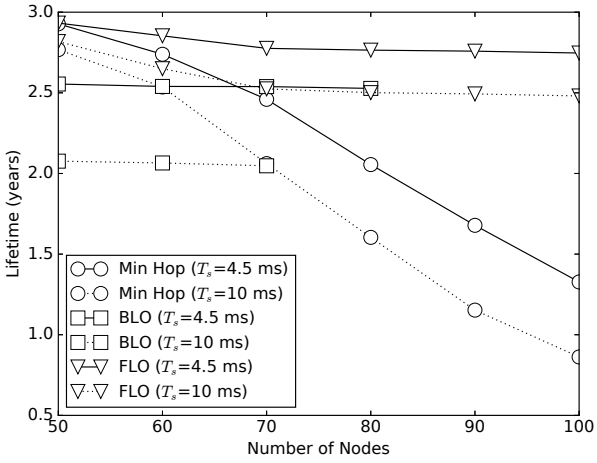

Figure 2. Network lifetime.

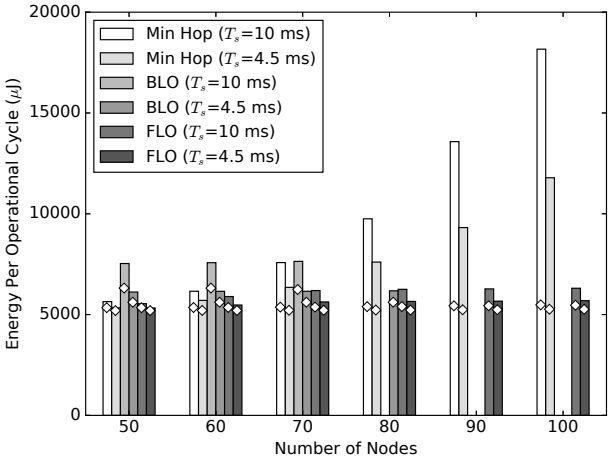

Figure 3. Average (diamonds) and maximum (bars) node energy consumption.

cycle. The maximum energy consumption bars are generated by identifying the node in each of the 200 simulation iterations that consumes the maximum energy per cycle and then averaging those 200 maximum values together. The average energy consumption diamonds are simply the energy consumption per cycle averaged across all nodes and all iterations.

We observe in Fig. 3 that the routing algorithm we choose does indeed have very little impact on the average energy the nodes consume but that BLO and FLO are much better at controlling the maximum node energy consumption since that is the objective of their cost equations. Since overall network lifetime is equal to the lifetime of the node consuming the most energy, the dramatic difference in maximum energy consumption for larger networks in Fig. 3 accounts for the very large improvement in lifetime that FLO offers in Fig. 2.

It is therefore expected that the advantage of using FLO or minimum hop would disappear for network designs optimized to minimize average node energy consumption. However, optimizing for average energy consumption is only possible for applications that can tolerate node failures. For petroleum refinery operations where safety is a critical issue, node failures are unacceptable and the lifetime metric we adopt in this paper is the most appropriate.

Finally, the benefit of using network optimization to extend network lifetime is illustrated in a slightly different way in Fig. 4. This figure shows the average fraction of energy remaining in node batteries after the first node dies. While it has been shown in [27] that uniform energy depletion across the network is not a requirement for maximizing lifetime, it is intuitive that longer lived networks should do a better job of extracting the energy collectively stored by the batteries of all nodes in that network. Fig. 4 demonstrates that this is the case since we observe FLO harvesting a considerably greater portion of the energy from the network than minimum hop, especially for larger network sizes.

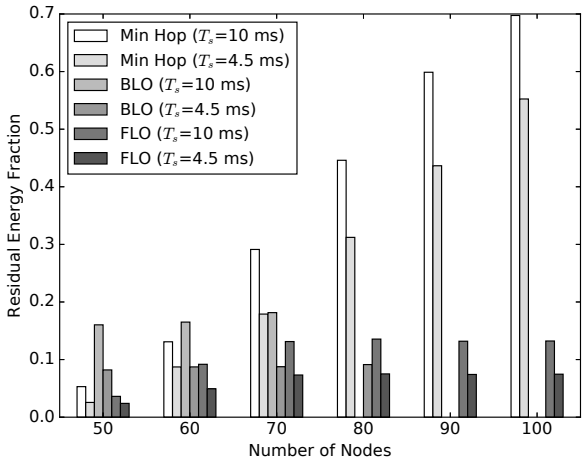

Figure 4. Average residual energy per node.

VI. Conclusion

This paper has presented a wireless sensor network study where the parameters of the network are based on a petroleum refinery instrumentation scenario. In this scenario, the sensor and processor require considerably more energy than the operation of the wireless network. While it might be expected that this would reduce the benefit of optimizing the wireless network, this paper demonstrates that optimization can still considerably increase network lifetime when the definition for lifetime is dependent on maximum node energy consumption rather than average consumption.

Care must be taken when selecting the algorithm for determining the network routes. This paper demonstrates the importance of using an optimization that accounts for the fixed frame size constraint of current industrial wireless standards. Results show this approach achieves network lifetimes that far exceed minimum hop routing and is considerably more robust than

optimization performed at the bit level, especially for large network sizes. That said, even frame based optimization does experience a non-negligible failure rate when solving routes for large network sizes. Under these conditions, minimum hop may still prove a viable choice if routing robustness is more critical than network lifetime.

REFERENCES

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

[2] K. Romer and F. Mattern, "The design space of wireless sensor networks", *IEEE Wireless Communications*, vol. 11, no. 6, pp. 54–61, 2004.

[3] R. Young, "Petroleum refining process control and real-time optimization", *IEEE Control Systems Magazine*, vol. 26, no. 6, pp. 73–83, 2006.

[4] I. Johnstone, J. Nicholson, B. Shehzad, and J. Slipp, "Experiences from a wireless sensor network deployment in a petroleum environment", in *Proceedings of the 2007 International Conference on Wireless Communications and Mobile Computing*, ser. IWCMC '07, Honolulu, Hawaii, USA: ACM, 2007, pp. 382–387.

[5] A. N. Kim, F. Hekland, S. Petersen, and P. Doyle, "When HART goes wireless: Understanding and implementing the WirelessHART standard", in *IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2008)*, IEEE, 2008, pp. 899–907.

[6] S. Petersen and S. Carlsen, "WirelessHART versus ISA100.11a: The format war hits the factory floor", *IEEE Industrial Electronics Magazine*, vol. 5, no. 4, pp. 23–34, 2011.

[7] R. Zurawski, Ed., *Industrial Communication Technology Handbook*. CRC Press, 2014.

[8] T. Blevins, D. Chen, M. Nixon, and W. Wojsznis, *Wireless Control Foundation: Continuous and Discrete Control for the Process Industry*. International Society of Automation, 2015.

[9] Pepperl + Fuchs, *W-BAT-B1-Li lithium thionyl chloride battery*, 2014.

[10] J. Zuo, C. Dong, S. X. Ng, L. L. Yang, and L. Hanzo, "Cross-layer aided energy-efficient routing design for ad hoc networks", *IEEE Communication Survey & Tutorials*, vol. 17, no. 3, pp. 1214 –1238, 2015.

[11] I. Al-Anbagi, M. Erol-Kantarci, and H. T. Mouftah, "A survey on cross-layer quality-of-service approaches in WSNs for delay and reliability-aware applications", *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 525–552, 2016.

[12] F. Dobslaw, T. Zhang, and M. Gidlund, "QoS-aware cross-layer configuration for industrial wireless sensor networks", *IEEE Transactions on Industrial Informatics*, vol. 12, no. 5, pp. 1679–1691, 2016.

[13] *System engineering guidelines IEC 62591 WirelessHART*, Emerson Process Management, 00809-0100-6129, Rev AA, 2015.

[14] L. Shi and A. O. Fapojuwo, "TDMA scheduling with optimized energy efficiency and minimum delay in clustered wireless sensor networks", *IEEE Transactions on Mobile Computing*, vol. 9, no. 7, pp. 927–940, 2010.

[15] S. Cui, R. Madan, A. J. Goldsmith, and S. Lall, "Cross-layer energy and delay optimization in small-scale sensor networks", *IEEE Transactions on Wireless Communications*, vol. 6, no. 10, 2007.

[16] W. Shen, T. Zhang, F. Barac, and M. Gidlund, "PriorityMAC: A priority-enhanced mac protocol for critical traffic in industrial wireless sensor and actuator networks", *IEEE Transactions on Industrial Informatics*, vol. 10, no. 1, pp. 824–835, 2014.

[17] ns-3 Project, *The NS-3 network simulator manual*, Oct. 4, 2016.

[18] M. King, *Process Control*. Wiley, 2011.

[19] M. Gaafar and G. G Messier, "Petroleum refinery multi-antenna propagation measurements", *IEEE Antennas and Wireless Propagation Letters*, vol. 15, pp. 1365 –1368, 2016.

[20] T. S. Rappaport, *Wireless Communications, Principles and Practice*, 2nd ed. Prentice Hall, 2002.

[21] IEEE Computer Society, *IEEE std 802.15.4 - 2006*, The Institute of Electrical and Electronics Engineers, Inc. New York, USA, 2006.

[22] American National Standard, *ANSI/ISA-100.11a-2011 wireless systems for industrial automation: Process control and related applications*, 2011.

[23] Atmel Corporation, *AT86RF233: Low power, 2.4GHz transceiver for ZigBee, RF4CE, IEEE 802.15.4, 6LoWPAN, and ISM applications*, AT86RF230, Atmel, 2014.

[24] O. C. Imer, S. Yüksel, and T. Başar, "Optimal control of LTI systems over unreliable communication links", *Automatica*, vol. 42, no. 9, pp. 1429–1439, 2006.

[25] *Rosemount 3051 pressure transmitter*, 00813-0100-4001, Rev SA, Rosemount/Emerson, 2014.

[26] *8/16-bit Atmel AVR XMEGA D3 microcontroller*, ATxmega64D3, Atmel, 2015.

[27] S. Olariu and I. Stojmenovic, "Design guidelines for maximizing lifetime and avoiding energy holes in sensor networks with uniform distribution and uniform reporting", in *Proceedings of the 25th IEEE International Conference on Computer Communications (INFOCOM 2006)*, 2006, pp. 1–12.