LoRa Synchronized Energy-Efficient LPWAN

Boyang Zhang, Vidya Srinivas, Yan Zhang, and Robert P. Dick
Dept. of Electrical Engineering and Computer Science
University of Michigan
Ann Arbor, Michigan, U.S.A.

***Abstract*—This work addresses the need for low-power wide-area networks (LPWANs) that remain energy efficient even when scaled to cover large geographic areas without power delivery infrastructure. LoRaWAN is a widely used LPWAN that requires a star (or star-of-stars) topology in which only leaf nodes are energy efficient enough to be powered by compact and inexpensive batteries. We describe the design of LoRa Synchronized Energy-Efficient LPWAN (SEEL), a novel protocol for multi-hop LoRa networks that allows battery-powered nodes to forward messages to difficult-to-access locations. SEEL performs link-quality-aware, dynamic formation of a tree topology network enabling nodes to deliver their packets to a central gateway while minimizing node energy consumption. We evaluate SEEL via a month-long deployment in an outdoor, real-world environment covering 9.0 km^2 and analyze its delivery rates and energy efficiency; we find SEEL nodes, on average, function 6.6× as long as always-on nodes would in the same deployment setting. We conduct a follow-up deployment of SEEL to compare its dynamic network formation protocol with a static network formation protocol; the SEEL network drops 16% fewer node-to-node packets than would a static topology network, even in near-ideal circumstances for a static topology network.**

***Index Terms*—LoRa, Low-Power, LPWAN, Multi-Hop, Time Synchronization, Tree Topology, Wireless Sensor Networks**

I. Introduction

Wireless Internet-of-Things (IoT) systems are proliferating. The number of IoT-connected devices has grown from 8.6 billion in 2019 to 11.3 billion in 2021 and is projected to reach 30 billion in 2030 [1]. Many of these systems require low-power wide-area networks (LPWANs) that are scalable and energy efficient. LPWANs have many applications, including agricultural monitoring and smart city management.

Some LPWAN applications require high coverage area, low power consumption, and the ability to place network devices in locations without power infrastructure and high-speed internet access. These applications benefit from multi-hop communication, which uses transitive links to a gateway node. In this configuration, the gateway has access to a practically unlimited energy source (such as grid AC power or a large battery) and possibly the internet, while all other nodes are powered by compact, inexpensive batteries. Such multi-hop networks require careful design if they are to be reliable, scalable, and energy efficient.

In this work, we describe and evaluate a novel protocol called LoRa Synchronized Energy-Efficient LPWAN (SEEL). SEEL uses LoRa PHY (physical layer) technology: a long-range, low-power radio modulation technique [2]. SEEL wirelessly and transitively connects a single wall-powered gateway node to numerous battery-powered sensing and communication nodes in a dynamically adapting tree structure. We describe the SEEL protocol and share insights from a one-month deployment covering 9.0 km^2. We analyze network performance, energy consumption, and network lifetime.

II. Related Work and Contributions

LoRa wide-area network (LoRaWAN) is a popular LPWAN that uses LoRa PHY and is promoted by the LoRa Alliance [3]. Standard LoRaWANs are constructed using a star or star-of-stars topology [4]. Its network span is limited to two hops by its star-of-stars topology and hopping requires always-awake, energy-intensive intermediate gateways.

Several researchers have considered using LoRa PHY in multi-hop applications without energy constraints. Dias et al. [5] describe an augmented version of LoRaWAN in which end nodes transmit data to multiple intermediate nodes that relay the data to gateways. Lee et al. [6] describe a LoRa mesh network where the gateway node maintains path information and can query a child node's data through its parent path. While the work described in this paragraph expands upon the capabilities of LoRaWAN, in contrast with our work, it does not consider energy efficiency.

A few topologies have been proposed for protocols using LoRa PHY in energy-constrained environments. Misbahuddin et al. [7] introduce a tree-based LoRa network where nodes are placed in concentric rings surrounding the gateway node. The authors simulate their network and find that a single-hop approach uses ten times as much energy as their multi-hop approach. Tehrani et al. [8] describe an energy-efficient LoRa tree topology protocol and compare it against a star topology protocol and LoRaWAN. They show their design is more energy efficient due to their optimization of energy-relevant, node-level LoRa parameters. Their approach does not focus on packet delivery and uses a static tree topology. They evaluated their protocol through simulation and a 35-node real-world deployment spanning 0.01 km^2. These prior approaches do not restructure the network in response to variations in node-to-node link connectivity. SEEL dynamically adapts its network topology to variations in the RF environment so nodes that would otherwise be dropped can associate with new parents, improving message delivery rates.

Tran et al. [9] describe a tree-based LoRa protocol (Two-Hop RT-LoRa) that adjusts its topology dynamically; their protocol does not consider energy constraints and limits communication

ICC 2023 - IEEE International Conference on Communications | 978-1-5386-7462-8/23/$31.00 USG | DOI: 10.1109/ICC45041.2023.10279767

to two hops. In comparison, SEEL considers energy constraints and has no maximum node hop limit. The topology in the Two-Hop RT-LoRa is reformed only when nodes join or have poor communication. SEEL's dynamic topology frequently adapts to environmental conditions, determining appropriate routes immediately before message transmission. However, nodes in the Two-Hop RT-LoRa protocol can be assigned transmission slots after nodes are deployed; in contrast, SEEL nodes must have their transmission slots manually configured beforehand. Thus, Two-Hop RT-LoRa network is better suited to deployments where manual slot assignment is difficult because node locations are unknown beforehand or change, while the SEEL network is better suited to large-area deployments where power delivery infrastructure is limited. Tran et al. test their protocol in simulation and in a 40-node real-world deployment spanning an area of $0.175\,\mathrm{km}^2$. SEEL is evaluated in a 13-node, $9.0\,\mathrm{km}^2$ deployment, which better approximates a scenario where multi-hop networks would be most useful.

Our main contribution is a novel dynamic network construction and adaptation protocol. This protocol

1) adapts the network topology to the current conditions of the wireless communication environment,
2) tolerates node movement and failures,
3) is optimized to minimize energy consumption, and
4) supports time synchronization among nodes, which enables higher energy efficiency and supports record keeping in periodic sensing applications.

The open-source implementation of SEEL for the Arduino platform is available for free use by researchers and educators (see Conclusions).

Many LPWAN protocol designs are evaluated through simulations or small-scale deployments. We conduct a month-long deployment of our SEEL network in a $9.0\,\mathrm{km}^2$ outdoor environment. Our deployment uses economical hardware in a harsh natural environment that exposes problems theoretical models may overlook. This deployment enabled us to draw several conclusions. For example, we describe and experimentally evaluate a power model for predicting node lifetimes and discuss discrepancies with the real-world deployment. We also compare the node-to-node packet delivery rate of our dynamic topology network with that of a static topology network; dynamically adapting the topology reduces dropped packets by 16%.

III. System Design

The SEEL protocol may be used with a wide range of platforms and physical layers; however, it is optimized for long-range, low data-rate LPWANs. SEEL's parameters allow tuning for particular hardware and environmental characteristics.

A SEEL network is composed of one gateway node (GNode) and one or more sensor nodes (SNodes). The term "node" used in the following sections refers to both GNodes and SNodes. The GNode acts as the sink for data messages and initiates network formation and time synchronization. The GNode stores or uploads received data messages, and is not energy constrained. SNodes are the sources of data messages and also forward data and protocol messages. Nodes send data using the TDMA collision avoidance scheme [10]. The following subsections describe the components used in SEEL.

A. Message Types

SEEL has four types of messages with priorities decreasing in the following order: *Broadcast*, *Acknowledge*, *Data*, and *ID_Check*. *Broadcast* initiates network formation, *Acknowledge* confirms packet delivery, *Data* carries user-generated information, and *ID_Check* allows SNodes to join the network.

Broadcast messages contain time synchronization information and Over-the-Air (OTA) parameters that allow reconfiguration of already-deployed SNodes. *Broadcast* messages are processed first to ensure that network synchronization and parameter configuration are completed before other messages are received. They include OTA parameters such as SNode awake and sleep time and information about network restarts, hop counts, and parent selection.

B. Dynamic Tree Topology

There are temporal variations in RF communication channels [11]. These changes may be due to transient interference, changes in the positions and chemical compositions of RF barriers (such as foliage), weather, or other environmental factors. We collected node-to-node Received Signal Strength Indication (RSSI) measurements for five days. During the measurement period, multiple node-to-node links experienced temporary RSSI shifts lasting from several hours to several days. Figure 1 shows measurement data for one link. The RSSI variations presented suggest that dynamic topology formation may be beneficial because it allows adaptation to varying environmental conditions. The resulting improvements in communication reliability may improve network performance and energy efficiency by better supporting time synchronization, enabling more accurate (and therefore longer) SNode sleep durations.

1) Network Formation: SEEL periodically forms a tree-structured network, once per "cycle". In SEEL, the associations among parents and children dynamically change over time, e.g., to adapt to the loss of nodes. Dynamic adaptation may be useful in networks with either stationary or mobile nodes since node-to-node link strength is affected by both environmental conditions and motion. Each cycle, SEEL and user-defined functions are called. First, the GNode transmits a *Broadcast* that is forwarded downstream to every transitively connected SNode; each node transmits one *Broadcast* per cycle. Receivers of *Broadcast* messages set the senders as their parents if certain conditions are met (Section III-B2). Each cycle, an SNode may have only one parent, but multiple children. After the network is formed, SNodes send *Data* and *ID_Check* messages upstream. Messages that do not reach the GNode during a cycle are saved locally and continue moving toward the GNode in later cycles.

2) Parent Selection: When an SNode receives a *Broadcast*, it determines whether the sender is a suitable parent. The SNode sets the first non-blacklisted node (Section III-B5) it receives

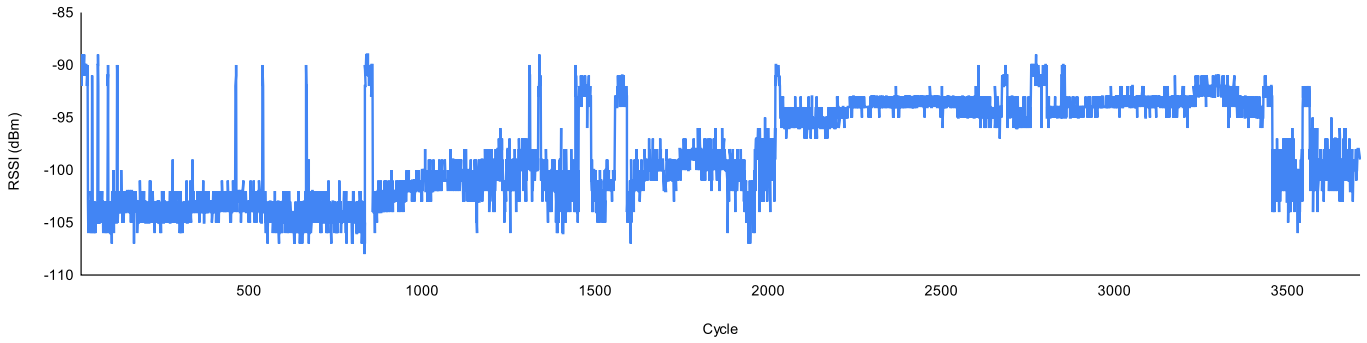

Fig. 1. Measurements illustrating node-to-node RSSI changes during 5 days with measurements every 2 minutes. RSSI is typically between -105 dBm and -100 dBm. There are four 50 cycle periods near cycles 850, 1,450, 1,550, and 3,550 where RSSI is between -95 dBm and -90 dBm. There is one 1,450 cycle period near cycle 2000 with RSSI around -95 dBm.

a *Broadcast* from as its parent. For subsequent *Broadcast* messages, the SNode evaluates suitable parents based on their hop counts and path RSSIs. The path RSSI, the minimum RSSI along a path to the GNode, is passed down in *Broadcast* messages. When an SNode selects a parent, it assigns itself its parent's hop count plus one. Potential parents must have hop counts less than or equal to the receiver node's current hop count to prevent loops. If this condition is met, the receiver compares the sender's path RSSI with its current parent's, selecting the parent with the highest path RSSI.

3) Node Entry: SNodes must apply to join a SEEL network each time they are power-cycled. When an uninitialized SNode receives a *Broadcast*, it sends an *ID_Check* and awaits a response; *ID_Check* messages are forwarded toward the GNode. SNodes waiting for an *ID_Check* response do not send *Data* messages. An SNode continues to send one *ID_Check* per cycle until it receives a transitive response.

4) Message Acknowledgement: When a parent node receives a *Data* or *ID_Check* from its child, it replies with an *Acknowledge* message. A child node retains the *Data* or *ID_Check* in its queue until it receives an *Acknowledge* from its parent. This prevents loss of *Data* and *ID_Check* messages due to transmission errors.

5) Node Blacklist: If the parent of an SNode does not respond during a cycle, the SNode blacklists that parent, preventing its selection in future cycles. However, SNodes can still receive network parameters from a blacklisted node's *Broadcast*. If an SNode receives messages from only blacklisted parents while it is awake, it clears its blacklist, allowing any node to be its parent the next cycle.

C. Time Synchronization

During network formation, SNodes are time synchronized. The GNode embeds its internal time in the *Broadcast* it transmits every cycle. When downstream SNodes receive the *Broadcast*, they set their own system times to the time in the *Broadcast*. To compensate for transmission delay, SNodes increment the time in the *Broadcast* based on their packet Time-on-Air (ToA: the duration a LoRa transmission takes to complete) estimate before transmitting the *Broadcast*.

D. Duplicate Message Filtering

Nodes may receive multiple instances of the same message due to multi-path effects (signal reflections, deflections, scattering, etc.). Therefore, each node maintains a packet sequence number that is incremented on every message transmission. Nodes contain queues of received messages. Incoming messages with the same sender node ID and sequence number as queued messages are considered duplicates and ignored.

E. Timer Adjustment

SNodes may remain in low-power sleep mode longer or shorter than desired due to hardware timer inaccuracies. In our deployments, we used the Arduino watchdog timer which can deviate from its nominal rate by up to 10% depending on process variation, voltage, temperature, etc.; this causes timer drift. SEEL's sleep adjustment algorithm compensates by keeping a local estimate of the true sleep duration. SNodes record the durations between waking up and receiving a *Broadcast*, the wakeup-to-*Broadcast* (WTB) times. They then subtract the actual and requested sleep durations to proactively adjust future requested sleep durations, thereby compensating for drift.

When an SNode misses a *Broadcast*, it shortens the sleep duration to reduce the likelihood of missing the next *Broadcast*. After three consecutive misses (an adjustable parameter), the SNode remains awake until it receives a *Broadcast*. This long awake period imposes a large energy toll. To reduce the frequency of such expensive events, the user may indicate a slack parameter to wake up SNodes earlier. Battery lifespan is non-monotonic in this parameter: it trades off the high but rare cost of remaining awake for an entire cycle against the low but frequent cost of increased wake duration. The optimal value depends on the predictability of timer drift.

IV. Experimental Design

We deployed a SEEL network in a 9.0 km^2, outdoor environment for evaluation. We aimed to answer these questions.

- How many packets arrive at the GNode? Why are packets lost?
- How accurately does a state-based power model estimate battery-dependent network lifespan for this application?

- Does dynamic adaptation of the network topology enable better packet delivery rates than a static topology?

Our deployment consisted of 1 GNode and 13 SNodes. The GNode was an off-the-shelf Dragino LG01-N single-channel LoRa IoT gateway. Each SNode consisted of an Arduino Pro Mini (3.3V 8MHz) and a HopeRF RFM95W LoRa transceiver, and was powered by a CR123A 3.0 V battery.

We deployed the network in a exurban area in Lima and Chelsea, Michigan for 40 days; the deployed nodes covered an area of 1.7 km×5.3 km, or 9.0 km^2. Distance was measured as the Euclidean distance between 2-D GPS coordinates; neglecting elevation never produced greater than 1% error.

To conserve battery energy and accelerate deployment, the network was deployed in two phases. Initially, we configured the network with a 3-minute awake time and a 7-minute sleep time, which allowed us to frequently check the signal strengths of deployed nodes. Deployment started near the GNode. This allowed us to measure signal strengths and verify connections during deployment so adjusting node positions was less time consuming. After the initial deployment, we reconfigured the network to use a 3-minute awake time and 57-minute asleep time, which are more typical of distributed periodic sensing applications.

V. Evaluation

This section presents and explains data gathered during network operation. Some questions required a different deployment setup to answer, so we deployed a secondary small-scale SEEL network. The 9.0 km^2 deployment is henceforth referred to as the 'primary' deployment and the small-scale deployment is referred to as the 'secondary' deployment. Both deployments used the same hardware, but SNode IDs are inconsistent across them. During the primary deployment, SNode 7 almost immediately suffered a permanent hardware failure, so its data are omitted from energy analysis and network performance statistics; however, its data are included in topology analysis since some data were collected before the failure, and they give insight on packet routing for SNode 7.

A. Network Topology

Figure 2 shows an overview of the network layout and the frequency of different parent-child connections during the primary deployment. It shows that nodes typically associate with several parents. Parent selection is based on RSSI, so we expect nodes to connect to physically close nodes because distance is inversely correlated with RSSI. However, since we are using path RSSI as the parenting measure, some nodes taking physically longer routes to the GNode to avoid low-RSSI paths. Figure 3 shows node-to-node connection counts and their average immediate RSSIs. Nodes 15 and 20, which are near the GNode, send most of their messages to the GNode and act as parents for many other nodes. In contrast, node 11, which is also physically near the GNode, only occasionally sends to the GNode because it often identifies other parent with higher path RSSIs. These data indicate that SEEL's topology formation is heavily influenced by its parenting heuristic and that the parent with highest path RSSI varies over time.

TABLE I
Primary Deployment Packet Delivery Rates to GNode

Node ID	PDR	Node ID	PDR	Node ID	PDR
6	0.948	13	0.856	17	0.762
10	0.914	14	0.848	18	0.956
11	0.887	15	0.872	19	0.850
12	0.785	16	0.854	20	0.762

TABLE II
Parameters (Prm.) Used in Power Estimation

Prm.	Definition	Val.	Prm.	Definition	Val.
BW	bandwidth (kHz)	250	DE	low data rate on	0
SF	spreading factor	12	n_{pr}	# preamble symbols	8
CR	coding rate (index)	4	T_a	awake time (s)	180
PL	payload length (B)	24	T_s	sleep time (s)	3420
CRC	CRC check on	1	V_b	battery voltage (V)	3.0
IH	implicit header on	1			

Values in reference to model in Loh et al. [13].

B. Network Performance

SNodes (omitting data from SNode 7) in our primary deployment delivered an average of 660 data messages out of an average of 761 data messages generated; an average of 101 data messages were lost per SNode (13.3%). Individual SNode PDRs are listed in Table I. We used a message queue size of 7 due to SNode memory constraints; dropped packets were due to SNodes' message queues overflowing. This overflow problem can be reduced by increasing SNode memory to outlast periods of poor transmission reliability, deploying more SNodes closer to the GNode to handle the increased volume of messages there (fat trees [12]), or using a more sophisticated parenting heuristic to load balance SNode communication.

C. Energy

In this section, we use a state-based power model to estimate SNode battery lifespans and compare these results with measured lifespans. We model three power-consumption states: receive, transmit, and sleep. In the receive state (rec), an SNode waits to receive messages. In the transmit state (tx), an SNode transmits messages. In the sleep state (slp), an SNode goes into a low-power sleep. Other SNode activities, such as processing messages, are also included in the rec state since they have similar power consumptions. Due to the timer drift described in Section III-E, we record the WTB time.

We present general equations for SNode energy use per cycle in joules E_{cyc} and SNode lifetime in cycles L_{cyc}. In the following equations V_b refers to battery voltage in volts and E_b refers to battery energy in joules. T refers to time in seconds and I refers to current in amperes; each of these variables may have a subscript indicating its cause (e.g., its associated state).

$$E_{cyc} = (T_{tx} \cdot I_{tx} + T_{rec} \cdot I_{rec} + T_{slp} \cdot I_{slp}) \cdot V_b \text{ and} \quad (1)$$

$$L_{cyc} = (E_b - E_{mb})/E_{cyc}. \quad (2)$$

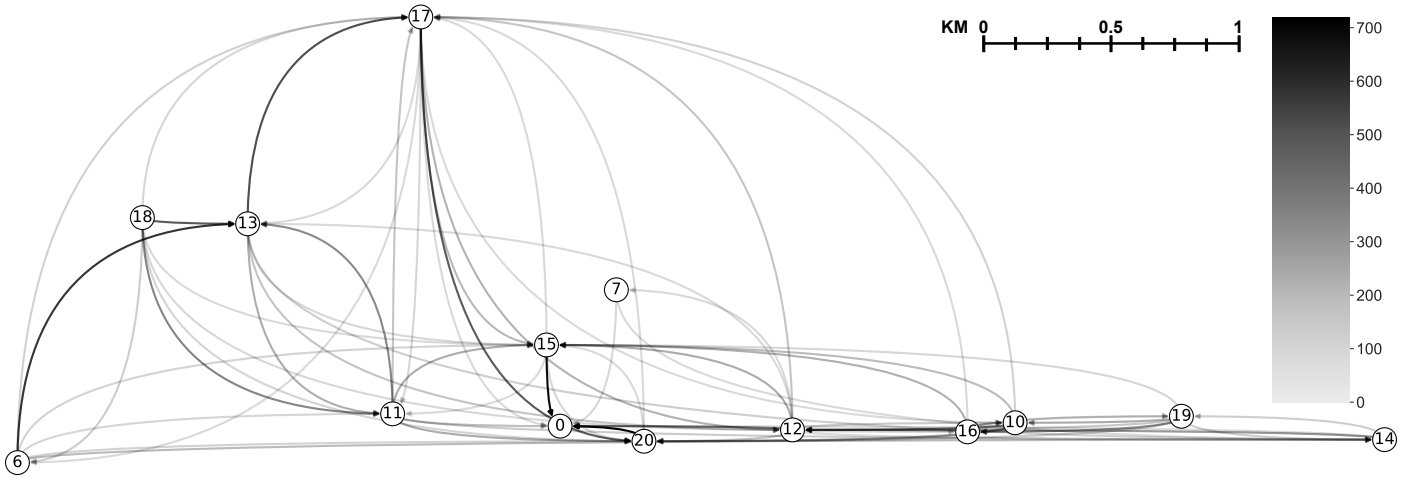

Fig. 2. Network topology diagram showing the number of parent-child connections during the primary deployment. Nodes on the diagram represent SEEL nodes where node '0' is the GNode. Arc shades indicate the number of connections between the two nodes they connect. Node placements are based on their scaled-down GPS coordinates. Exact values for node-to-node connections are listed in Figure 3.

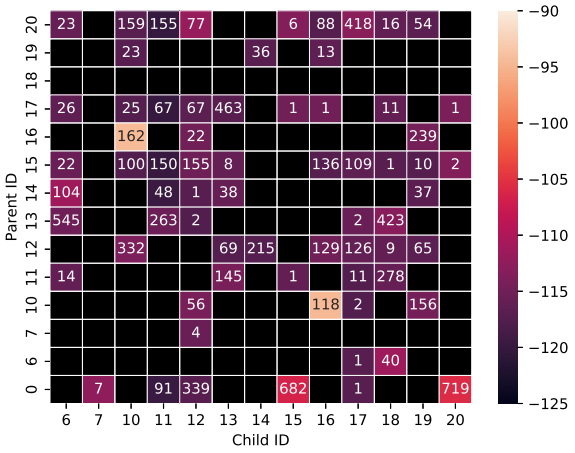

Fig. 3. Heatmap quantifying parent-child links formed during the primary deployment. For each cell, the color represents the average RSSI of received messages by the child and the number in the cell represents the number of times the parent-child link was made. A cell without a number implies no connections.

1) SNode Lifetime Estimation: We calculate the transmission time T_{trans} following Loh et al. [13]. We use the parameters in Table II in our calculations, yielding a transmission time estimate of 0.987 s, the packet's ToA.

To estimate the lifetime of an SNode, we must first calculate its average power consumption. We measured the power consumption for each of its states, as summarized in Table III, which contains the computed mean and standard deviation over five SNodes, based on measurements of state-dependent power consumptions of individual SNodes.

Our SNode lifetime model accounts for the following factors: average WTB time in seconds (T_{wtb}), average transmissions per cycle (n_{tx}), and missed *Broadcast* messages (mb).

$$T_{tx} = T_{trans} \cdot n_{tx}, \tag{3}$$

$$T_{rec} = T_a - T_{tx} + T_{wtb}, \text{ and} \tag{4}$$

$$T_{slp} = T_s - T_{wtb}. \tag{5}$$

The energy expended due to missed *Broadcast* messages (E_{mb}) is computed using the equations below. mb is a list of missed *Broadcast* messages for the SNode, where $mb[x]$ is the number of times an SNode missed x *Broadcast* messages in a row. S is the user-defined scalar for extended awake time when multiple *Broadcast* messages are missed in a row. M is the number of missed *Broadcast* messages before the SNode stays awake indefinitely waiting for a *Broadcast*.

$$T_{mb} = \sum_{i=1}^{M-1} \left[T_a \cdot S^i \cdot mb[i]\right] + \sum_{j=M}^{\infty} mb[j] \cdot (T_a + T_s) \text{ and} \tag{6}$$

$$E_{mb} = T_{mb} \cdot I_{rec} \cdot V_b. \tag{7}$$

Figure 4 depicts the experimental SNode lifetime in our deployment as a percentage of estimated SNode lifetime. The estimated SNode lifespans had a root mean squared error (RMSE) of 488.0 hours relative to the measured lifespans, and a mean average error (MAE) of 420.6 hours. Notably, there was a systemic overestimation of 59.7%, which when calibrated for with a single multiplicative factor, reduced RMSE and MAE to 191.8 hours and 168.9 hours, i.e., 39.3% of the RMSE was due to systemic causes that affected all SNodes such as regional temperature that can change battery performance.

The remaining 60.7% of the RMSE was due to non-systemic variation such as localized events (e.g., interference events), process variation of on-board components (e.g., watchdog timers), and environmental variations (e.g., shading changing local temperatures and surrounding objects acting as ground planes). The RMSE might be further reduced by calibrating to variations in the power consumptions of individual SNodes or considering temperature-dependent battery effects. However,

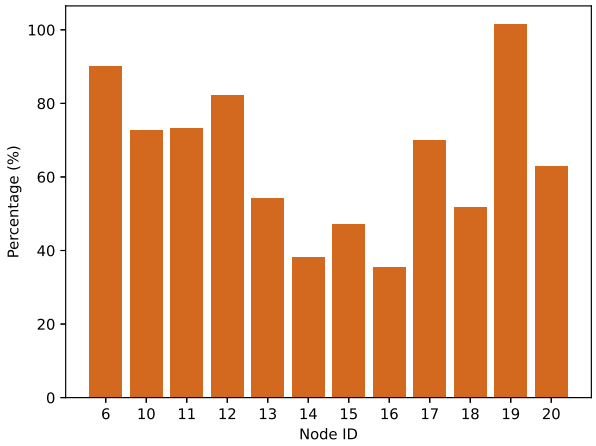

Fig. 4. Measured SNode lifetimes compared to uncalibrated model-based estimates. Each bar represents measured SNode lifetime as a percentage of the estimate.

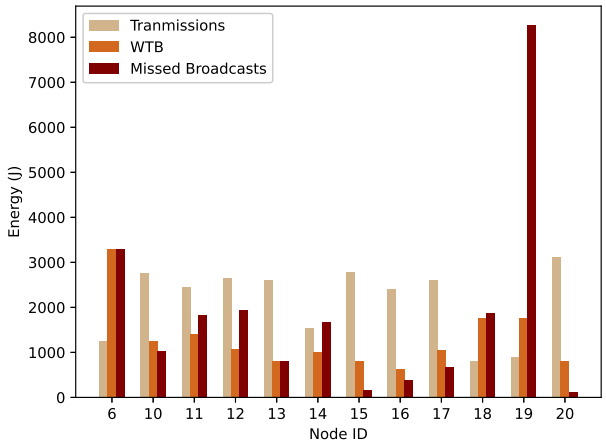

Fig. 5. For each SNode, the energy expended on overall transmissions, overall WTB, and missed *Broadcast* messages.

TABLE III
MEASURED STATE-DEPENDENT POWER CONSUMPTIONS

State	**Power Consumption** $(\mu \pm \sigma)$
Receive	$0.04519 \pm 0.0004938\,\mathrm{W}$
Transmit	$0.3807 \pm 0.008905\,\mathrm{W}$
Sleep	$1.624 \cdot 10^{-5} \pm 2.126 \cdot 10^{-6}\,\mathrm{W}$

the non-systemic calibration values would only be useful for a particular set of SNodes, i.e., our SNodes will have different process variation parameters than those of other researchers and developers. Although we believe our general modeling approach will be of use to others interested in estimating SNode battery lifespans, our results may have been influenced by circumstances specific to our deployment, such as temperature patterns, so the model parameters should be verified before use, especially in dramatically different deployment scenarios.

We determine the amounts of energy spent on different tasks during our primary deployment. This information may be useful for focusing energy optimization efforts on the most relevant tasks. We present energy expended in transmission, WTB, and missed *Broadcast* messages over the duration of our deployment, as shown in Figure 5. To reduce transmission energy, node-to-node links should be empirically validated such that there are few message delivery failures, thus reducing the number of repeated transmissions. WTB and missed *Broadcast* messages are interdependent; higher WTB reduces the probability of missing a *Broadcast*, so WTB parameters should be tuned to minimize the sum of WTB and missed *Broadcast* energy.

2) SNode Lifetime Comparison: Always-On SNodes: Without time-synchronization and power management features similar to those of SEEL, nodes in multi-hop LoRa networks would need to remain on to receive transmissions. We therefore compare battery lifespans of SEEL SNodes and sleepless SNodes. Using Equation 2 with the same number of transmissions as a primary deployment SNode, we find that the sleepless SNode has a lifespan of 99 hours. In our primary deployment, time-synchronized and sleep-capable SNodes have lifespans 6.6× as long, on average. These results indicate that time synchronization and sleep were beneficial in our primary deployment.

D. Topology Evaluation

We conducted a secondary, eight-SNode, short-range (single room), one-day, but heavily instrumented deployment to compare node-to-node link qualities in a static network and SEEL, which supports dynamic, per-cycle network formation. This deployment had much higher data rates and shorter sleep cycles (4 minutes vs. 1 hour) than our primary deployment and gathered additional data, enabling trace-driven network simulation of the PDRs resulting from different speculative network topologies. The node arrangement represented a nearly ideal case for static network topologies, since all nodes were in the same room; all SNodes were able to communicate directly with the GNode and there were few sources of environmental variation over the short deployment. For comparison to SEEL, we selected a base-case static topology by using the most frequently formed (77.6%) topology during this deployment, which turned out to be the star topology (where all SNodes were directly connected to the GNode). This is unsurprising, as all nodes were placed within one-hop range of the GNode. It is also the topology used by conventional LoRaWAN. Nine topologies were encountered. The secondary deployment facilitated data collection for a trace-driven simulation in a nearly ideal scenario for a single-hop static topology. Unlike the primary deployment, it was not typical for SEEL, which is designed for multi-hop, dynamic topology networks.

Table IV shows node-to-node PDR for each SNode averaged over all cycles of the secondary deployment with 90% confidence intervals. SNode 8 had low PDR to the GNode, producing fewer samples and resulting in a wider confidence interval. The dynamic topology results in 16.0% fewer (node-

TABLE IV
SECONDARY DEPLOYMENT STATIC VS. DYNAMIC TOPOLOGY NODE-TO-NODE PDR WITH 90% CONFIDENCE INTERVALS

Node ID	Dynamic PDR	Static PDR
1	0.97134 ± 0.00831	0.96418 ± 0.00849
2	0.97477 ± 0.00485	0.96926 ± 0.00550
4	0.96856 ± 0.00624	0.96431 ± 0.00669
6	0.96336 ± 0.00800	0.95918 ± 0.00796
8	0.94053 ± 0.03585	0.93086 ± 0.03289
10	0.96716 ± 0.01055	0.96288 ± 0.01009
14	0.97116 ± 0.00601	0.97164 ± 0.00615
16	0.97073 ± 0.00692	0.95574 ± 0.00797
AGR*	0.96914 ± 0.00275	0.96328 ± 0.00285

*Aggregate node-to-node packet delivery rate.

to-node) dropped packets than the static topology. Examining individual node-to-node links at 90% confidence showed the dynamic topology generally performed better than the static topology. However, using 95% confidence intervals would have resulted in overlap of the static and dynamic intervals. We conclude that the dynamic network probably outperforms the static in a near-best-case setup for the static network although it is possible (but unlikely) that the observed differences were due to random variation.

One would expect a static topology to perform well in our secondary deployment conditions; there are no node dependencies (for hopping) in the frequently formed star topology. However, even in an environment suitable for a static topology, the dynamic topology has higher node-to-node PDR. We believe the benefits of a dynamic network would be greater in a larger (i.e., multi-hop) and longer duration deployment with more wireless environment variation, i.e., a typical wide-area distributed sensing deployment represented by our primary deployment.

VI. CONCLUSIONS

This paper described SEEL, a novel LoRa PHY based LPWAN protocol that enables multi-hop networking via a dynamically adapting tree topology and maintains high energy efficiency via synchronized sleeping. We evaluated the performance and energy efficiency of SEEL in a month-long, large-scale deployment covering 9.0 km^2. SNodes in our primary deployment functioned 6.6× as long as always-on nodes would have. We conducted a secondary deployment and found that SEEL's dynamic topology dropped 16% fewer packets than a static topology despite nearly ideal conditions for a static topology. We provide an open-source implementation of SEEL for the Arduino platform at https://github.com/SEEL-Group/SEEL and invite others to use and improve SEEL.

ACKNOWLEDGEMENTS

This work was supported in part by the University of Michigan Electrical Engineering and Computer Science Department. We would like to thank Brian Oo, Pranav Karra, Kevin Tactac, Albert Anwar, Matthew Smith, Peter Mulliner, Kevin Vandegrift, Kristen Vandegrift, and Jim Krichbaum for their assistance in testing the SEEL protocol.

REFERENCES

[1] L. S. Vailshery, "Number of IoT connected devices worldwide 2019-2021, with forecasts to 2030," Aug. 2022, https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/.

[2] U. Noreen, A. Bounceur, and L. Clavier, "A study of LoRa low power and wide area network technology," in *Proc. Int. Conf. on Advanced Technologies for Signal and Image Processing*, 2017, pp. 1–6.

[3] J. Haxhibeqiri, E. De Poorter, I. Moerman, and J. Hoebeke, "A survey of LoRaWAN for IoT: from technology to application," *Sensors*, vol. 18, no. 11, p. 3995, 2018.

[4] J. Silva, J. J. P. C. Rodrigues, A. M. Alberti, P. Solic, and A. L. L. Aquino, "LoRaWAN – low power WAN protocol for Internet of Things: a review and opportunities," in *Proc. Int. Multidisciplinary Conf. on Computer and Energy Science*, 2017, pp. 1–6.

[5] J. Dias and A. Grilo, "LoRaWAN multi-hop uplink extension," *Procedia Computer Science*, vol. 130, pp. 424–431, 2018.

[6] H. C. Lee and K.-H. Ke, "Monitoring of large-area IoT sensors using a LoRa wireless mesh network system: Design and evaluation," *IEEE Trans. on Instrumentation and Measurement*, vol. 67, pp. 1–11, Mar. 2018.

[7] M. Misbahuddin, M. S. Iqbal, D. F. Budiman, G. W. Wiriasto, and L. A. S. I. Akbar, "EAM-LoRaNet: Energy aware multi-hop LoRa network for Internet of Things," *KINETIK: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*, vol. 7, no. 1, pp. 81–90, 2022.

[8] Y. H. Tehrani, A. Amini, and S. M. Atarodi, "A tree-structured LoRa network for energy efficiency," *IEEE Internet of Things Journal*, vol. 8, no. 7, pp. 6002–6011, 2021.

[9] H. P. Tran, W.-S. Jung, D.-S. Yoo, and H. Oh, "Design and implementation of a multi-hop real-time LoRa protocol for dynamic LoRa networks," *Sensors*, vol. 22, no. 9, 2022.

[10] S. C. Ergen and P. Varaiya, "TDMA scheduling algorithms for wireless sensor networks," *Wireless Networks*, vol. 16, no. 4, pp. 985–997, 2009.

[11] K. Srinivasan, P. Dutta, A. Tavakoli, and P. Levis, "Understanding the causes of packet delivery success and failure in dense wireless sensor networks," in *Proc. Int. Conf. on Embedded Networked Sensor Systems*, 2006, pp. 419—420.

[12] Y. Wu, J. A. Stankovic, T. He, and S. Lin, "Realistic and efficient multi-channel communications in wireless sensor networks," in *Proc. Int. Conf. on Computer Communications*, 2008, pp. 1193–1201.

[13] F. Loh, N. Mehling, and T. Hoßfeld, "Towards LoRaWAN without data loss: studying the performance of different channel access approaches," *Sensors*, vol. 22, no. 2, 2022.