LoRaLitE: LoRa protocol for Energy-Limited environments

Lukasz Sergiusz Michalik*, Loic Guegan*, Issam Raïs*, Otto Anshus*, John Markus Bjørndalen*

* Department of Computer Science, UiT The Arctic University of Norway
email: { lukasz.s.michalik, loic.guegan, issam.rais, otto.anshus, john.markus.bjorndalen }@uit.no

***Abstract*—To do in-situ observations of the arctic tundra, many small sensor nodes are used. Reporting the data to remote back-ends is hard because backhaul networks are scarce, and a node cannot expect to see one. Also, there are typically no humans physically close to the nodes to fetch the data and to replace batteries. Consequently, the nodes need a highly available, energy-efficient long range network.**

While LoRaWAN provides energy-efficient communication for nodes, it does so at the cost of needing an always-on gateway with a high energy consumption. The gateway is also a single point of failure.

Instead we propose LoRaLitE for Energy-Limited environments where the gateway enters sleep phases to reduce energy consumption. The sleep phases of the nodes are coordinated accordingly. For high availability, any end-node with a backhaul-network can be elected to become the gateway.

We conducted a series of simulation experiments to document the performance behavior of both LoRaWAN and LoRaLitE. The results show that the LoRaLitE gateway spends from 10 to 10,000 times less energy than a LoRaWAN gateway. A LoRaLitE node spends from 13% to 42% more energy than a LoRaWAN node. The achievable bandwidth for LoRaLitE is only insignificantly smaller than for LoRaWAN.

A LoRaWAN gateway needs a relatively large, heavy battery. If the gateway fails, assuming a new gateway can be elected, several nodes must have similar large batteries to be able to take over as the gateway. This is impractical to achieve for more than a few nodes because the nodes become more expensive, harder to camouflage, and impractical to deploy.

A LoRaLitE gateway on the other hand only needs a battery like any other node. Even if all nodes need larger batteries than for the LoRaWAN case, the increase is in practical terms insignificant.

We conclude that LoRaLitE makes it realistic to use LoRa for nodes on the arctic tundra, providing for insignificant increased energy usage at the nodes, insignificant reduced bandwidth, but with significant increase in gateway availability. LoRaLitE also provides for an easier deployment in practice because all nodes can be kept small and light.

***Index Terms*—LoRa, network protocol, IoT, Cyber Physical System, Arctic Tundra**

I. Introduction

The conceptual models used by the ecologists from the Climate-ecological Observatory for Arctic Tundra (COAT) (http://www.coat.no) to predict the impact of the climate change on the Arctic Tundra ecosystem, require performing in-situ observations of the tundra with the use of sensor nodes.

However, the arctic tundra is large, has harsh weather conditions, is potentially dangerous for humans, and expensive to visit. There are laws and regulations restricting the size and type of installations which can be put there. Network and energy infrastructures are as the common case limited or not existing.

All of this work together to force observation nodes to be small, light-weight, unobtrusive, relying on small batteries instead of solar panels and wind mills.

Nodes need access to a back-haul data network to report data. However, accessing cellular data networks has in practice proven difficult even when the coverage indicated by the coverage maps was good [1]. Applying larger radio antennas on the nodes to improve signal quality and strength is restricted by laws and regulations, or not possible because of damage from wind, snow and ice. The nodes also have a limited amount of energy limiting how much effort they can put into searching for networks, and boosting signal strength. Providing access to a backhaul network is thus a challenge.

Long-range transmissions and high receive sensitivity is offered by LoRa [2] technology. It employs a sub-GHz-band signal capable of penetrating obstacles to some degree. This improves the reachability of nodes covered by snow, ice, mud, earth and rocks.

LoRaWAN is a software communication protocol stack and system architecture applying LoRa. It is commonly used for sensor networks [3]. In LoRaWAN, nodes talk to one or more gateways over LoRa, and the gateways communicate with a remote LoRaWAN network server over a backhaul network using IP.

The end-nodes sleep for long periods of time, and wake up to communicate either according to some schedule or when they need to in order to transmit messages. Nodes then go back to sleep again to conserve energy. This is very energy efficient. Consequently, the nodes can use smaller batteries and still operate for years.

Gateways on the other hand are always-on, listening simultaneously on multiple LoRa channels in order to be able to receive transmission at any given moment. This is costly in terms of energy for the gateways. To operate for years the gateways need to be connected to a power grid or large and heavy batteries. Because a power grid is not available on the arctic tundra, batteries must be used. However, for a typical multi-channel LoRa concentrator which can be used in LoRaWAN gateway, at least 12,6 kWh is needed per year [4]. With current battery technologies this is equivalent to a battery weighing about 100kg. Even a single gateway becomes impractical to deploy by field personnel having to hike far into remote observation sites.

A further issue with LoRaWAN is that the gateways need a back-haul network to report to the network server. However, as explained, back-haul networks are scarce on the arctic tundra. Therefore multiple gateways must be used to increase the probability of having enough gateways to communicate with every node and also have back-haul network access. This adds to the challenge of using LoRaWAN for the arctic tundra because multiple gateways are needed, and each needs a battery of 100 kg.

The aforementioned characteristics of LoRaWAN make it inadequate to be used for nodes on the arctic tundra. Even if the nodes use little energy, the gateways are very energy costly. Location of the gateways also become critical because they need a back-haul network to be able to relay data from the nodes to the back-end network server.

A network better suited for use on the arctic tundra should provide for small batteries both for the gateways and the nodes. Also, gateways should not be fixed to the initial location at deployment because the reachability of a back-haul network can change over time depending on factors including failures and weather. The bandwidth should be comparable to the LoRaWAN bandwidth, but at significantly lower gateway energy usage.

To this end, we propose the Energy-Limited LoRa (LoRaLitE) protocol. Both the nodes and the gateways sleep and wake up on a schedule, determined by the gateways, to communicate. Consequently only smaller batteries are needed. Any node in the network (*child* nodes) can in principle be elected to become a gateway (*parent* node) if a gateway fails or looses access to a back-haul network. Gateways are elected from the set of nodes which sees a back-haul network, or which with a certain probability can expect to do so within a time range.

In practical terms both the nodes and the gateways become identical, and in particular they all have smaller batteries. While the node energy cost increases a bit, it increases much less than the decrease in gateway energy cost. Consequently, while the weight of node batteries increases, say, 200 grams, the weight of the gateway batteries decreases 99 kg.

This has several advantages. Any node can become a gateway, spreading the energy cost among the nodes seeing a back-haul network. The operational lifetime limited by a single battery charge increases. Bandwidth is not reduced. If gateways fail, other nodes can take over, increasing the probability of seeing a back-haul network and being able to communicate with all nodes. The smaller batteries make it practical to deploy the nodes without needing special considerations for large and heavy gateways. This aids scaling up the number of nodes to be deployed.

The means to achieve this is through introducing coordinated communication between the nodes. The parent node reduces power consumption by turning off the radios and suspending own operations when it is not involved in communication. The child nodes communication is coordinated by the parent. They wake up and listen for instructions according to a pre-arranged routine. They do not transmit unless they receive instructions from the parent node.

This paper focuses on the lightweight, high-availability LoRaLitE protocol. The parent node election, dynamic network creation, and fault tolerance are the topics of future work.

In summary, this paper provides the following contributions:

- it presents LoRaLitE protocol eliminating the dependence on external electrical grid and enabling fully battery-supported operations,
- it documents the architecture, the design, and the implementation of the proposed solution,
- it defines the characteristics of a network suitable for the resource-constrained arctic tundra,
- it describes the coordinated communication solution that is used as the means of reducing the energy expenditure, and it addresses the synchronization of nodes in the presence of clock drift,
- it evaluates the performance and energy usage of LoRaLitE protocol network, both theoretically and through simulations, and compares it with LoRaWAN.

This paper is organized as follows. Section II presents related work. Section III and Section IV describe the proposed LoRaLitE architecture and design respectively. Section V introduces LoRaLitE simulator as the proof of concept implementation of the protocol. Section VI and Section VII offers the experiments settings followed by an evaluation of the received results. Finally, Section VIII provides our conclusions.

II. Related work

Multiple different medium access protocols were proposed for LoRaWAN to increase network scalability and data collection performance. The simplest approach, called Listen Before Talk (LBT), is described in [5]. The authors proposed a mechanism where the end-node has to listen for a random amount of time to detect whether a given LoRa channel is occupied by another device before transmitting a prepared packet. Carrier-sense Multiple Access (CSMA) [6] aims to improve network throughput by using different channels and multiple spreading factors simultaneously. Time-Division Multiple Access (TDMA) scheme [7] [8] uses transmission slots allocation to avoid collisions in the network. TDMA scheme is interesting because it provides a way to organize LoRa communication in an organized fashion and was the inspiration for our current research. However, the aforementioned approaches require the network to operate with an always-on gateway. In contrast, the LoRaLitE network does not need any node to be always active.

On-demand data collection using short-range Wake-Up Receiver (WuR) and LoRa radio was proposed in [9] [7]. Instead of random LoRa communication intervals, the authors proposed a system where end-nodes are woken up on demand to transmit data. The gateway sends requests over LoRa to the always-on cluster head which wakes up neighborhood end-nodes via short-range WuR. Then each woken up end-node sends data directly to the gateway over LoRa. However, the proposed approach further increases the energy consumption of the network due to the introduction of the additional always-on cluster head. Moreover, the presented WuR has a maximum range of 50 meters which is far shorter than the typical distance between nodes deployed in the Arctic Tundra [1].

Device to Device (D2D) communication protocol for LoRaWAN network was presented in [10]. The authors proposed to use a network server to arrange direct communication between nodes in the LoRaWAN network where data from one end-node is needed by another end-node. This approach reduces the load for the LoRaWAN gateway and backhaul network due to data being directly exchanged between interested parties. However, the proposed solution depends on the availability of a backhaul network whose reach is limited in the Arctic region. In contrast, LoRaLitE was designed around that limitation. The network can operate without backhaul network access and act when the access is sporadically available.

The existing solutions are not suited for nodes on the Arctic Tundra because they require a dedicated, always-on gateway and the availability of a backhaul network. On the other hand, any LoRaLitE node with the highest probability of having backhaul network access can act as a gateway (parent node) for the neighborhood nodes. Moreover, a backhaul network is not required for internal communication in a LoRaLitE network. LoRaLitE parents are only turned on when communicating with child nodes.

III. System architecture

This section presents the system architecture of the observation node and the LoRaLitE functionality.

A. Observation Node

An observation node is composed of multiple independent but interconnected components. In this paper we consider a node consisting of two components: the observation component, and the connectivity component. The former is responsible for observing and collecting information about the environment. The latter is responsible for establishing connectivity between nodes within reach of each other. The connectivity is based upon short and long range wireless network technologies.

B. LoRaLitE Functionality

The network of nodes forms a star topology with two types of nodes: a parent node and several child. The potential parent node is already selected by all nodes in the neighborhood. The parent has access to a backhaul network and uses it to deliver data collected from child nodes. The parent node uses network commands to coordinate child nodes communication. A child node can communicate only when it is allowed to do so by a received network command.

The LoRaLitE network can be in one of two defined states: *warmup-state* and *data-oriented-state*. The network is considered to be in the *warmup-state* when it is being formed for the first time by the nodes. This state is characterized by frequent changes in the number of nodes in the network as well as which node is the parent node. The communication in the network is mainly related to node discovery and parent selection. Nodes try to synchronize their clocks so that they can be active at the same time to communicate. The *warmup-state* can last multiple days before the network is considered to be stable enough to phase into the *data-oriented-state*.

In the *data-oriented state* the parent node is focused on requesting data from child nodes. It sends network commands to all nodes within reach and waits long enough to receive all responses. Even though the status of all nodes in the network is already established, changes in the network can happen occasionally due to node failures. A different node can be elected as the parent node if, for instance, the current one does not have access to a backhaul network for a specified amount of time (typically days) and there is another node in the network which can replace it.

In this paper, we only consider the *data-oriented-state*. The dynamic network creation, the transition between network states, and network management including the process of selecting the parent node and the process of a node joining the network is a part of further research.

The connectivity component for long range communication has to support three main functions: command, sleep, and calculate next network event time. *Command* is used by the parent node to orchestrate communication in the network. The parent node sends commands with a given known interval. *Sleep* lowers the energy usage of the component while the network is idle, usually in between network commands issued by the parent node. *Calculate next network event time* function let the node calculate the next network activity time thus providing the input for the *Sleep* duration. All nodes need to support these functions. The main difference is that the parent nodes send commands, while child nodes respond to them.

IV. Design

The design for the LoRaLitE prototype builds upon the LoRa network technology. It takes physical radio properties into consideration and focuses on the configuration that provides the maximum possible communication range.

A. Observation Node

A node comprises several separate functional parts including an energy-efficient microcontroller with enough internal resources to run the LoRaLitE functionalities. The microcontroller can be connected to other parts of the node by wire or wirelessly. Within a node, the microcontroller and the other parts of the node need to have a common time reference as well as a way to exchange data. To provide a shared time an external real-time clock is used. To exchange data a crash tolerant filesystem over a shared SD card is used. The microcontroller is equipped with a LoRa radio for creating the local long-range network. An example diagram of a node consisting of a computer responsible for reading the sensors and a microcontroller responsible for connectivity over LoRa is depicted in Figure 1.

In the rest of the paper, we focus on the connectivity component, ignoring the other components of the node.

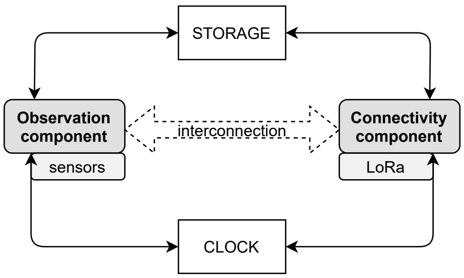


Fig. 1. Diagram of a simple node.

B. LoRaLitE Medium Access Control

LoRaLitE uses Time Division Multiple Access (TDMA) scheme [7] for the Medium Access Control (MAC) in the local LoRa network. The parent node coordinates and instructs child nodes by sending out specific commands as broadcast messages.

There are two types of commands: *Requiring Response* (RR) and *Not Requiring Response* (NRR). In the case of the RR type of command, the parent assigns response time slots to the child nodes by including a list of node IDs in the payload, requesting a response from each node in the same order. The i^{th} child node's response slot is given by $i(ToA+T_{rg})$, where ToA is the Time on Air calculated for a configured response size, and T_{rg} is a response guard time used as a margin to avoid collision between transmitting nodes. The response guard time is configurable with a millisecond resolution.

There are three commands the parent node sends:

- *beacon* (B) - NRR type of command used by the parent node to notify child nodes about the interval between commands. Child nodes use it to calculate the sleep duration and the next wake up time.
- *discovery* (D) - RR type of command issued by the parent node to update its list of child nodes in the network. If a child node does not respond to this command a given number of times, it is removed from the network. The parent node will not query that child node again unless it rejoins the network. In a future version of the LoRaLitE design, this command will be expanded to also let the parent node discover nodes joining the network.
- *collect* (C) - RR type of command used by parent node to collect data from child nodes.

LoRaLitE uses a packet format depicted in Figure 2. On the physical level, the LoRa packet consists of only three standard fields: preamble, phy_payload, and crc. On the MAC level the packet is composed of multiple fields described below:

- *N_ID*: a unique number between 0 and 254 which identifies the node in the network that sent the packet. This number is statically assigned to each node at the network configuration time.
- *Packet_NR*: packet sequence number assigned by the sender of the packet. Each node keeps track of the sequence number of the last received message in order to identify lost transmission.
- *CMD*: a parent specifies the command (B, D, C), and child nodes specify the response type: discovery response (DR) and collect response (CR).
- *Nr_of_Ret*: States how many times the given command with the same *Payload* will be repeated by the parent node. The value of that field decreases until it reaches 0 which indicates that the parent node finished repeating the same command.
- *Payload*: holds the payload for the different command and response packets:
  - *Beacon*: INTERVAL indicates the interval between commands issued by the parent. NEW_INTERVAL holds the same value as the INTERVAL unless the parent node decides to change the commands interval. In that case, the parent node will repeat the same *Beacon* command multiple times to ensure that child nodes will receive it. The *Nr_of_Ret* field indicates the number of repetitions left before the interval changes.
  - *Discovery, Collect*: holds a list of N_IDs for child nodes that were selected by the parent node to respond. The lists are typically rotated between commands to balance energy consumption between child nodes. This way, any child node gets a fair chance of responding in the first slot and then suspending immediately. Alternatively, a range can be specified to select a larger set of child nodes.
  - *Discovery Response*: a child node sends back its RSSI value logged at the time of receiving a *Discovery* request from the parent node.
  - *Collect Response*: holds data sent by the child node. This is typically meta-data about the node, or collected sensor data.

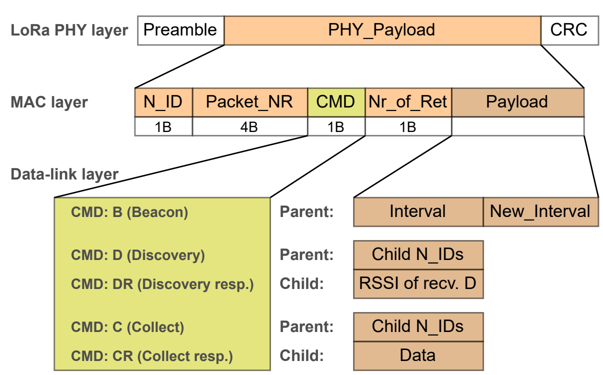


Fig. 2. LoRaLitE packet format.

C. Calculating next network-event time

To suspend and resume operations while still ensuring synchronized communication, both parent and child nodes calculate the time of the next network event (parent node command) before entering sleep-mode. This is done based on the time of the current network event and the *send interval* time communicated by the parent to the child node. The nodes then go to sleep, and wake up shortly before the expected time of next synchronization, accounting for the duration of the wake-up phase, depicted in blue in Figure 3.

Clock drift compensation is only done in the child nodes. To do this, they wake up earlier to compensate for a slower clock than the parent node, and stay up longer to compensate for a potentially faster clock than the parent. The prolonged time listening for parent node commands is called the *guard time* T_g. The maximum clock drift CD_{max} that can occur in a given send interval is a product of the send interval length SI and the accuracy of the real time clock RTC_{acc}, given in parts per million, $CD_{max} = SI \cdot RTC_{acc}$.

Two different worst-case scenarios can occur involving opposite maximum clock drifts at the parent and child nodes, as depicted in Figure 3. In both of those scenarios the synchronization points expected by the parent and child nodes are separated by $2CD_{max}$. In the first scenario, where the parent-node clock is accelerated, and the child-node clock is delayed, the child needs to wake up earlier by $2CD_{max}$ with respect to when it would wake up if there was no risk of clock drifts involved. This allows the child to be ready for receiving the transmission from the parent on time. In the other scenario, where the child is accelerated and the parent is delayed, the transmission from the parent occurs $2CD_{max}$ later then expected by the child. To compensate for both scenarios, the child needs to wake up $2CD_{max}$ earlier than the target time and stay up for a period of $4CD_{max}$.

Furthermore, to detect any incoming transmission, a LoRaLitE node has to listen for at least 5 symbols of the preamble. The minimum guard time T_g required for ensuring communication synchronization in any of those extreme clock-drift scenarios is thus calculated as $T_g = 4CD_{max} + PDT$, where PDT stands for a preamble detection time. During the wake-up time the child is not in the active listening phase, and thus the wake-up time is not considered in the total guard time.

The minimum guard time for different send intervals and clock accuracy is plotted in the inset in Figure 6. Simulations of the two worst-case clock-drift scenarios confirm that adhering to this minimum guard time is sufficient for ensuring communication synchronization.

V. Implementation

To aid in exploring and documenting the performance characteristics of LoRaLitE we developed an event-driven simulator [11]. The simulator consists of modules that simulate the LoRa physical layer, the LoRaLitE MAC layer, and node behavior related to the LoRaLitE protocol such as sleep, wake-up, and radio state change.

A. LoRaLitE

The LoRaLitE simulator provides a way to configure the LoRa physical layer parameters such as frequency band, spreading factor (SF), bandwidth (BW), coding rate (CR), and transmission power. The configurable parameters for the LoRaLitE protocol include: command interval, guard time, child node response guard time, discovery and collect commands windows size, and packet payload size used by child nodes to generate a response to the parent requests. The simulator uses the Log-distance path loss model to determine whether a given LoRa transmission is successful. However, in this paper we do not consider scenarios with packet loss.

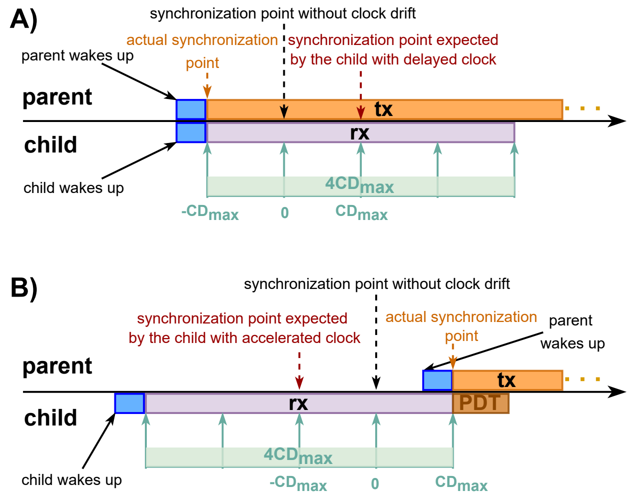


Fig. 3. Worst-case scenarios for parent and child nodes synchronization: A) parent-node clock accelerated, child-node clock delayed; B) parent-node clock delayed, child-node clock accelerated.

B. Energy model

The simulator calculates the energy consumption of every node in the network by recording time spent by a node and a radio module in a given state. The three defined states of a node are Idle, In Operation, and Sleep. The Idle state indicates that the node waits for any input from the connected peripherals. Currently, the only external input is provided by the radio module. When the node transmits or receives data it remains in the In Operation state. The node goes to the Sleep state either after receiving a transmission or after transmitting data as described with more details in Section IV-C. The LoRa radio module remains in one of the three defined states such as Receive (Rx), Transmit (Tx), or Off. We assume that the radio is in the Off state when the node goes to Sleep.

The energy consumption E of a node depends on the power consumption P of the node and varies over time t. For a given period of the simulation T, the energy consumption is given by $E(T) = \int_0^T P(t)\,dt$, as proposed in ECOFEN model [12].

VI. Simulation Settings

A. LoRa PHY

We have configured the LoRa physical layer to provide the highest receiver sensitivity by setting SF to 12, BW to 125 kHz, CR to 4/8, and the transmitting power to +14 dBm.

We used the 868 MHz frequency band with 1% duty cycle limitation.

B. Energy characteristics

For the energy model we applied energy characteristics of an Ambiq Micro Apollo 3 MCU (active-state: 2.5 mW, sleep-state: 5 µW) to a LoRaLitE node instance and Semtech sx1262 LoRa radio (RX: 21.6 mW, TX: 226 mW). In the instances where the LoRaWAN network was simulated the class A end-node used the same aforementioned energy characteristics of the LoRaLitE node while the always-on gateway was composed of WiMOD iC880A LoRa concentrator (RX: 1.45 W, TX: 1.5 W) with 4 active channels. We have also evaluated a single channel LoRaWAN gateway with the energy characteristics of a node with Apollo 3 MCU and SX1262 LoRa radio.

C. LoRaWAN settings

In the simulations involving LoRaWAN protocol, we assumed that the network was already established and the class A end-node was sending packets with a fixed interval selected for the scenarios. Upon sending a packet, the node opened two 304 ms long receive windows RX1 and RX2 in accordance with the LoRaWAN specification. In our simulation, the LoRaWAN gateway was only listening for transmissions from end-nodes without transmitting anything in return. To avoid packet collisions, each class A end-node schedule was generated with a 2-second separation from any other transmission in the network. This represents an optimistic best-case scenario for LoRaWAN with no lost packets due to collisions.

D. LoRaLitE settings

For the Collect command responses and LoRaWAN data transmissions we used 51 B packets. LoRaLitE parent node used receive windows big enough to query all child nodes in a single request. In all scenarios, the parent node schedule included one synchronization command and one discovery command per day with the remaining slots filled by data collection requests. The command interval (T_d) varied between scenarios. In the case of the Energy consumption at maximum transmission rate scenario VII-A the interval was set to the minimum allowed value of 329 s to comply with 1% duty cycle for the child nodes transmitting 51 B of data to the parent node. For the second scenario, the command interval was set to the longest possible length that allowed each child node to deliver a specified amount of data over 365 days of simulated time. Child nodes were configured to use the guard time (T_g) according to the considerations in Section IV-C and 50 ms long response guard time (T_{rg}).

E. Network

We configured the simulator to create from 1 to 60 nodes for each scenario to analyze the performance and the energy usage of the system. For every scenario, we simulated 365 days of the system's operation.

VII. Evaluation

In this section, we present the evaluation of the LoRaLitE protocol in various simulated scenarios. In addition, we compared the protocol to LoRaWAN to show the difference in the network energy usage when the network is governed by either the always-on LoRaWAN gateway or the duty-cycled LoRaLitE parent node.

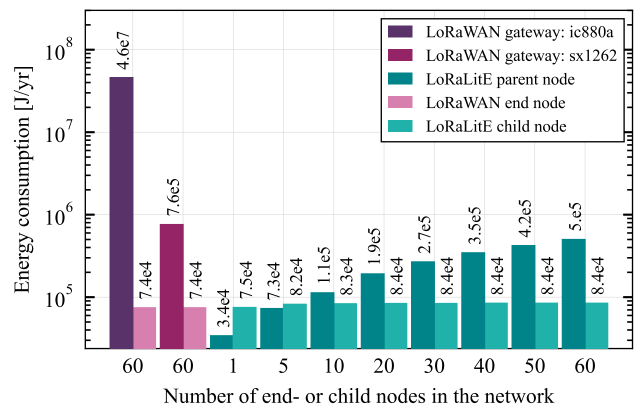


Fig. 4. Energy usage of LoRaWAN and LoRaLitE nodes at the maximum transmission rate over 365 days.

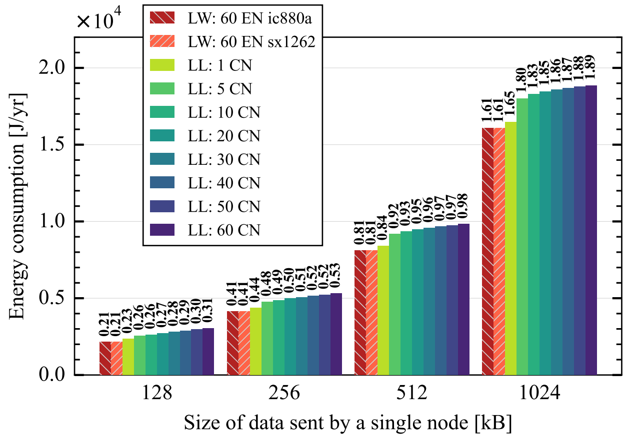
A. Energy consumption at maximum transmission rate

We quantify the maximum amount of data each child node is able to deliver within 365 days of the simulated time to the parent node in the LoRaLitE network when the command interval was set to the minimal value of 329 s allowed by regulations (1% duty cycle). We investigated how the number of nodes in the network affects the energy consumption of the parent and child node in this data-intensive scenario. We compared the results to a corresponding LoRaWAN network where each end-node transmits data with a 329 s interval.

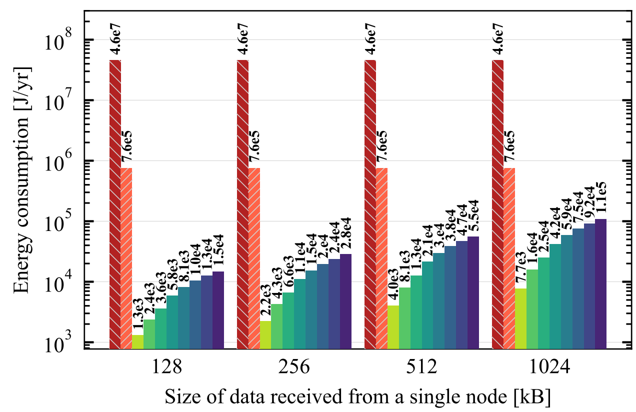
A single end-node (EN) in the LoRaWAN network sent 4.77 MB of data in 365 days of simulated time while a single child node (CN) sent 4.73 MB of data in the corresponding amount of time. The amount of data sent by a single child node in the LoRaLitE network was 0.77% smaller than in the LoRaWAN network with the same number of end-nodes. The difference can be explained by the overhead of sending a beacon and a discovery once per day.

The energy consumption for LoRaWAN and LoRaLitE nodes is depicted in Figure 4. The energy consumption of the LoRaWAN gateway, marked as GW on the Figure, was constant in all simulation runs, regardless of the number of LoRaWAN end-nodes in the network. The always-on LoRaWAN gateway used 45.72 MJ and 760.65 kJ of energy while equipped with WiMOD ic880a and Semtech sx1262 LoRa radios respectively. Each LoRaWAN end-node used 74.37 kJ of energy in the simulated collision-free environment.

For the LoRaLitE network, the energy consumption is a function of the number of child nodes under parent node (PN) control. When the LoRaLitE network was small and



(a) LoRaWAN end-node / LoRaLitE child node sending data.



(b) LoRaWAN gateway / LoRaLitE parent node receiving data.

Fig. 5. Energy consumption at LoRaWAN (LW) and LoRaLitE (LL) nodes for various amounts of data transferred in a period of 365 days, and different numbers of nodes constituting the network. The legend indicates the number of end-nodes (EN) / child nodes (CN) in the network and applies to both panels.

consisted of up to 5 child nodes the energy consumption of the parent node did not exceed the energy used by a single child node. However, as the number of child nodes increased in the network, the parent node used approximately 77 kJ more for each 10 additional child nodes in the network. At the same time, a single child node used between 75.07 kJ and 84.35 kJ of energy for the network consisting of between 1 and 60 child nodes. Figure 4 shows that the energy consumption of the child node did not increase much for a network with more than 10 child nodes.

The experiment results present a trade-off in energy consumption of the LoRaLitE network compared to LoRaWAN: parent node energy consumption is reduced with some additional energy consumption for the child node. The energy consumption of the LoRaWAN gateway is higher than the worst case for the LoRaLitE parent node: 91.63 times more for the 4-channel LoRaWAN gateway, and 1.52 times for the 1-channel gateway. With one child node, the difference is larger: 1339.34 times more and 22.28 times more respectively.

On the other hand, the child node used about 1.13 times more energy than the end-node in a similar size LoRaWAN network because the child nodes had to wait for the parent node commands and response slots.

B. Energy consumption in a range of communication scenarios

To measure the energy consumption with a range of more realistic communication scenarios, we used 4 scenarios where child nodes need to send an amount of data per year, ranging from 128 kB to 1024 kB. This corresponds to a range from 8 to 57 collect response messages per day from each child node. The length of the command interval in the LoRaLitE network, and the delay in data transmission in the LoRaWAN network were configured to the maximum possible value where the parent node and the LoRaWAN gateway could collect a required amount of data from every single node. The intervals configured for the scenario are presented in Figure 6.

Figure 5a depicts the energy used by LoRaLitE child nodes and LoRaWAN end-nodes for a network consisting of 1 to 60 nodes. When the total amount of transmitted data was relatively small, at 128 kB, a LoRaLitE child node used up to 1.42 times more energy than the LoRaWAN end-node. However, the difference in energy usage decreased to 1.17 times more when child nodes sent more data.

The main reason behind that is the data send interval presented in Figure 6. When the total amount of data transmitted from a node is small the difference in the data send interval is much bigger between LoRaLitE and LoRaWAN networks than when the nodes need to send more data. With less transferred data, the overhead from the beacon and discovery messages

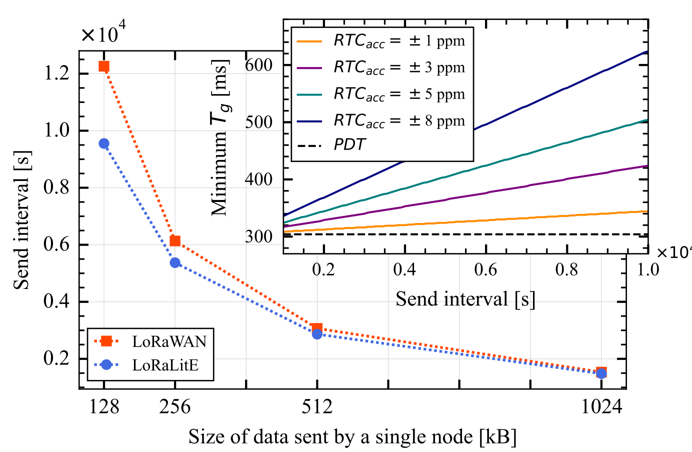


Fig. 6. Outer plot: Maximum send interval for delivering selected amount of data over 365 days. Inset: Minimum guard time T_g for different send intervals and clock accuracy. Constant preamble detection time marked with a dashed line.

becomes relatively larger.

The energy used by the LoRaLitE parent node and the LoRaWAN gateway is presented in Figure 5b. The numbers for any variant of the LoRaWAN gateway are the same as in the experiment in section VII-A since each of these gateways was always-on and listening for transmissions from end-nodes. For the 4-channel LoRaWAN gateway, the energy consumption is between 420 and 35007 times the energy consumption of the LoRaLitE parent node, depending on the amount of data and the number of child nodes. For the 1-channel LoRaWAN gateway, the energy consumption ranges from 7 and 582 times the energy consumption of the LoRaLitE parent node.

C. LoRaLitE in a practical use case

To put the proposed solution into perspective, we relate it here to the case of CO_2 observation nodes deployed in the Arctic tundra [1]. Ten separate nodes were involved, each trying to reach a 4G backhaul network once per day. No local network or gateway was present. As it resulted, only a small subset of the nodes was able to communicate at all, and for a period of one year there was no means of establishing whether the remaining nodes are operational, configured correctly and gathering data. With a LoRaLitE network, any of the nodes with a successful 4G connection could be elected as a parent, gathering status messages and a subset of observation data from the other nodes, and forwarding them to the backhaul network. That would open for a possibility of a relatively quick identification of any malfunctioning nodes and an 'on-call' service.

The energy cost of introducing LoRaLitE network protocol for the CO_2 nodes can be estimated based on Figure 5. For the case of 10 child nodes sending 128 kB of data each, the energy consumption at the child node is 2,633 joules per year, and at the parent node it is 3,572 joules per year. Each of the CO_2 nodes in [1] was equipped with 12 AAA lithium batteries, corresponding to the total theoretical maximum capacity of 226,800 joules per node. Executing the 128-kB-case of LoRaLitE protocol would thus consume 1.2% and 1.6% of the specified battery capacity for the child and parent nodes respectively. This indicates that LoRaLitE network could provide communication for the inaccessible CO_2 nodes on the Arctic Tundra without significantly adding to their energy requirements.

VIII. Conclusions

This paper presents the architecture of a modular observation node and a new LoRa protocol suitable for Arctic Tundra. LoRaLitE leverages coordinated long range communication and TDMA scheme to create a network that does not need a dedicated always-on gateway. Any node with access to a backhaul network can become a parent node. It uses network commands to instruct child nodes about how and when the communication happens. In case of a parent node failure, any of the child nodes can take over as a new parent.

This brings the energy consumption of the LoRaLitE parent down by a factor between 7 and 35007 times compared to a LoRaWAN gateway, depending on the gateway and the use case. For the child nodes, there is an increase in the energy consumed of around 1.13 to 1.42 compared to LoRaWAN end-nodes. Compared to CO_2 observation units we have previously deployed in the field, a LoRaLitE network would use between 1.2% and 1.6% of the energy budget for child and parent nodes. This makes LoRaLitE practical to use in the Arctic Tundra.

Future works will focus on the remaining functionality required for the LoRaLitE to be deployed in the field such as the dynamic network creation, transition between network states, and network management including the process of selecting the parent node and the process of a node joining the network.

Acknowledgments

This work is supported by Department of Computer Science (IFI) at the Arctic University of Norway (UiT), and the Distributed Arctic Observatory (DAO) project being supported by the Research Council of Norway (RCN) IKTPluss program, project number 270672.

References

[1] M. J. Murphy, Ø. Tveito, E. F. Kleiven, I. Raïs, E. M. Soininen, J. M. Bjørndalen, and O. Anshus, "Experiences building and deploying wireless sensor nodes for the arctic tundra," in *2021 IEEE/ACM 21st International Symposium on Cluster, Cloud and Internet Computing (CCGrid)*, May 2021, pp. 376–385.

[2] L. Alliance, "White paper: A technical overview of lora and lorawan," *The LoRa Alliance: San Ramon, CA, USA*, pp. 7–11, 2015.

[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, 2018. [Online]. Available: https://www.mdpi.com/1424-8220/18/11/3995

[4] *WiMOD iC880A DATASHEET*, IMST GmbH, 6 2018, rev. 1.1.

[5] J. Ortín, M. Cesana, and A. Redondi, "How do aloha and listen before talk coexist in lorawan?" in *2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2018, pp. 1–7.

[6] T. To and A. Duda, "Simulation of lora in ns-3: Improving lora performance with csma," in *2018 IEEE International Conference on Communications (ICC)*, 2018, pp. 1–7.

[7] R. Piyare, A. L. Murphy, M. Magno, and L. Benini, "On-demand tdma for energy efficient data collection with lora and wake-up receiver," in *2018 14th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2018, pp. 1–4. [Online]. Available: https://doi.org/10.1109/WiMOB.2018.8589151

[8] L. Chasserat, N. Accettura, and P. Berthou, "Short: Achieving energy efficiency in dense lorawans through tdma," in *2020 IEEE 21st International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*, 2020, pp. 26–29.

[9] F. A. Aoudia, M. Gautier, M. Magno, M. L. Gentil, O. Berder, and L. Benini, "Long-short range communication network leveraging lora™ and wake-up receiver," *Microprocessors and Microsystems*, vol. 56, pp. 184 – 192, 2018. [Online]. Available: https://doi.org/10.1016/j.micpro.2017.12.004

[10] K. Mikhaylov, J. Petäjäjärvi, J. Haapola, and A. Pouttu, "D2d communications in lorawan low power wide area network: From idea to empirical validation," in *2017 IEEE International Conference on Communications Workshops (ICC Workshops)*, 2017, pp. 737–742.

[11] Loralite simulator. [Online]. Available: https://github.com/arcticobservatory/loralite/

[12] A.-C. Orgerie, L. Lefèvre, I. Guérin-Lassous, and D. M. Lopez Pacheco, "Ecofen: An end-to-end energy cost model and simulator for evaluating power consumption in large-scale networks," in *2011 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*, 2011, pp. 1–6.