2022 22nd International Symposium on Cluster, Cloud and Internet Computing (CCGrid)

Impact of loosely coupled data dissemination policies for resource challenged environments

Issam Raïs*, Loic Guegan*, Otto Anshus*,
Department of Computer Science, UiT The Arctic University of Norway, Tromso, Norway*
Corresponding authors: issam.rais@uit.no, loic.guegan@uit.no

2022 22nd International Symposium on Cluster, Cloud and Internet Computing (CCGrid) | 978-1-6654-9956-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/CCGRID54584.2022.00062

***Abstract*—A Cyber-Physical System (CPS) deployed to a resource-constrained environment can face multiple challenges like (i) no or limited network coverage, (ii) no or limited possibility of energy replenishment, (iii) no or limited physical access by humans, (iv) nodes must cope with environmental factors including avalanches, low temperatures, snow, ice, water and wild animals. Devices being part of such a CPS must be battery powered and be energy efficient to achieve long life-time. However, the CPS still has to disseminate data to increase resiliency, safely keep results or update nodes. A trade-off between energy spent and data dissemination needs to be found, ideally by minimizing the energy usage and maximizing the relevant data dissemination performance metrics. In this paper, we evaluate and discuss the efficiency (in energy, time and number of successful distributions) of multiple data distribution policies by mean of flow-level simulations. We report on the trade-off between (i) successful data dissemination and (ii) energy and uptime overheads resulting from the usage of loosely coupled policies. To fully explore the scope of possibilities, we simulate a wide range of scenarios extracted from real measurements and previous deployments. Characteristics of CPS devices developed by the Distributed Arctic Observatory (DAO) are used as simulation platforms. Results show that an efficient policy in a given scenario can perform worse in another scenario. We also show that simple policies, especially when combined, can help in minimizing the energy consumed by most of the devices composing the CPS and maximizing the relevant dissemination performance metrics.**

***Index Terms*—CPS, data dissemination, energy efficiency, tundra, monitoring;**

I. INTRODUCTION

Recent literature shows that the number of Cyber-Physical Systems (CPS), wireless sensor networks (WSN), Internet of things (IoT), edge and extreme edge deployments explode in the last couple of years for multiple areas such as monitoring the environment [1], health care [2], crowd-sensing [3], [4], military [5], agriculture [6], gas-monitoring [7] and many others [8]–[11]. Low-Power Wide-Area Network (LPWAN) technologies have gained in popularity, making it possible and accessible to use and monitor larger areas especially the ones in scarce network coverage environments. Choosing these technologies imply having a wide coverage but low bandwidth and low energy overheads during communication phases [12].

We are interested in monitoring the Arctic tundra, one of the most sensitive eco-system to climate change. It is a large area with presently too few large-scale deployments of systems made of too few observation sites [13]. Gathering, processing and reporting of observations are limited by the availability of sufficient energy, and a data network with enough bandwidth and latency. The opportunities provided are consequently constrained by critical resources: energy and data networks.

The Distributed Arctic Observatory (DAO) project at the University of Tromsø, the Arctic University of Norway, is the use case of this paper. The project develops a CPS of devices called Observation Nodes (ONs) for the Arctic tundra. The DAO system observes the tundra and reports the observations.

As nodes are deployed in an isolated environment, we assume that the nodes can only exchange with neighbours and are supposed to save their energy. Thus, data dissemination must be carefully studied to reduce the energy overhead but still maximize the number of successful disseminations.

In this paper, we evaluate how loosely coupled dissemination policies can help when used in resource-scarce environments such as the one imposed by the Arctic tundra. The goal is to limit the energy overhead while increasing the number of successful disseminations when using LPWAN technologies. We focus on policies that do not impose a strict coordination between nodes (i.e loosely coupled). This is because (i) full coordination (i.e waiting for everyone to be up and running, ready and available, schedule current and future tasks and do it regularly) would be very costly both in time and energy; (ii) instruments deployed in the field are conservatively using their energy budget, as they need to survive during very long period of times.

The contributions of this paper are the following ones:

- Document and evaluate the effect of loosely coupled data dissemination policies in scarce-resource deployments;
- Quantify the impact of these policies on energy and uptime through simulation of previous deployments;
- Underline a range of possible trade-offs between energy overhead and successful distributions under various scenarios;
- Applying loosely coupled policies for data dissemination on a unique use-case: the Distributed Arctic Observatory (DAO) project.

The remaining of this paper is structured as follows. Section II presents the use case of this paper: the Arctic tundra, the DAO project, previous Arctic tundra deployment and their characteristics. Section III presents the related work. Section IV presents the experimental setup and details the policies, the metrics, the simulation tool and scenarios. Section V presents the results of simulated scenarios on explored metrics.

978-1-6654-9956-9/22/$31.00 ©2022 IEEE
DOI 10.1109/CCGrid54584.2022.00062

Section VI presents the lessons learned from the simulation campaigns. Finally VII concludes this work.

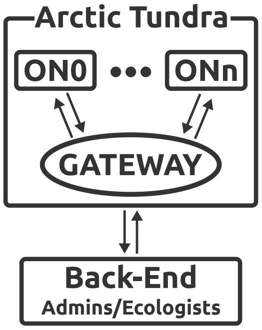


Fig. 1. Overview of the system imposed by the Arctic Tundra's characteristics. Back-end hosts a set of services [14]. Its connectivity to Observation Nodes (ON) deployed at the Arctic Tundra is sparse and unexpected. The wireless gateway in the topology is only used for 1:1 communications between Observation Nodes forming a star topology.

II. MOTIVATING USE-CASE: THE DAO PROJECT

This section presents the use-case of this work: the DAO project. First, the Arctic tundra and the difficulties to monitor it are covered. Then, the needs and the challenges for a distributed observatory are exposed. Finally, a current deployment and the importance of data dissemination are described.

A. The Arctic tundra, a complicated eco-system

The Arctic tundra is a very large, remote, hard to reach, and potentially dangerous eco-system. By observing its flora, fauna and environmental parameters, changes can be identified and tracked. Presently, much less than 1% of the Arctic tundra is monitored. However, it is the most sensitive eco-system to climate change [13]. Therefore, to accurately detect climate change, larger observations of the Arctic tundra are needed.

The Climate-Ecological Observatory for Arctic Tundra (COAT[1]) initiative is tasked with observing the Norwegian Arctic tundra, detect and explain climate related changes to advise the public and the authorities. First, the state of the Arctic tundra is determined based on measurements of the flora, fauna, weather, and the atmosphere to create multiple data sets. Second, the data sets are processed to detect interesting events, like the species of animals captured in images, creating multiple new data sets. Third, the new data sets are analyzed to extract significant information, like the number of foxes and eagles detected at the different monitored sites. These insights are then used as input to climate models. Finally, based on previous results, human understanding and decision making take place [13]. A ground-based observation system can observe large areas, do measurements at any time and rapidly react to local events both above and below ground, snow and ice, and do measurements at very high resolutions. Data can be reported back at any time, regularly, or on-demand. Significant processing and storage resources can be added to the devices to enable edge computing. The DAO project focuses on such ground-based observation approaches.

B. Towards a Distributed Arctic Observatory (DAO)

There are three major obstacles to consider when building an observation system for the Arctic tundra: (i) The lack of roads and associated infrastructure implies the impossibility to realistically visit by humans more than a very limited number of sites in order to fetch data, supply energy, or do repairs and updates; (ii) The limited or non-existing availability of a back-haul data network for doing automated reporting of data; (iii) The lack of energy working against using devices with advanced functionalities and still get a long operational lifetime. A distributed Arctic observatory system must carefully manage two fundamental resources: energy and wireless data networks. Devices are working on a limited energy budget delivered from batteries. As it is a complicated scenario, with bad weather and no long sun exposition during winter, swapping batteries by humans and regular energy harvesting are not plausible solutions. In addition, a set of functionalities are needed by the devices, including autonomous operations to save energy while still striving to observe and report.

While a back-haul network cannot be expected to be available as the common case, a device can have multiple local networks enabling communication with neighbours. Using a multi-hop approach, data can be reported through multiple units and finally to one or more units having access to back-haul networks or which are located to be reachable by humans or drones [15]. However, using the radio is energy-expensive. One approach to reduce transmission related energy consumption is to reduce the number of bits to exchange between devices, but such leverage is only applicable if the data can still be used to get close to the same analytic precision [14].

In this paper we focus on delivering data from one node to neighbours in the context of nodes deployed to and isolated on the Arctic tundra (i.e not accessible by a back-haul network as a common case), without multi-hopping nor modifying the data, as shown in Figure 1. Such a focus is interesting for multiple different cases.

C. Data dissemination, a crucial need

COAT ecologists presently use several approaches to observe the Arctic tundra [16], [17]. Tens to a few hundreds of small dedicated instruments are typically deployed according to where interesting events are expected. These instruments are deployed for multiple purposes, including to capture images animals. For hard to reach installations, it can take up to 6-12 months before humans visit the site to fetch the data. These deployments are usually done in small clusters, with 10 to 15 instruments per cluster. Each instrument is separated by at least hundreds of meters, to kilometers. Disseminating data from nodes to their neighbours, in such a deployment context could be crucial in multiple cases.

a) Important results backup: Deployed nodes can do local computation on local observations. It can be crucial to duplicate the results from these computations, due to the high

---

[1] https://www.coat.no/en/

probability of crash of deployed units (e.g through flooding, hardware failure). As a direct implication, to keep the data safe and reduce the chance of loosing results, we want to disseminate important results to as many neighbours as possible. For example, in [14] we reduce the size of captured pictures to reduce the number of bytes to be transmitted to a remote CNN deep learning application. Both the full sized as well as the reduced sized photos should (for some deployments) be disseminated inside a neighborhood for safe keeping purposes, until the data can be reported.

b) Update dissemination: Few to no nodes are expected to have a connection to a back-haul network (which would be sporadic and unreliable). As it is complicated and expensive to physically access the Arctic tundra, updates (e.g configuration files, executables, packets or other newer content for a receiver) need to be delivered from the back-end (when possible and needed). Updates can come from users of the system such as an ecologist or an administrator, as shown in Figure 1. When a node finally gets an update, we can expect it to disseminate it to its neighbours. As it is the only one getting the data from the back-end, it is the only node that can be trusted to have a valid version of the update files.

In both cases, the size of the disseminated data is not expected to be very high, due the wireless technologies limitations, energy and computing capabilities available. These constraints related to batteries and energy consumption are tackled in our previous work [14], [15], [18].

III. RELATED WORK

This section presents the related work concerning the network technologies usable in the Arctic tundra and the data dissemination policies, with a focus on energy efficient ones.

A. Network technologies

When choosing a network technology, the architect must have a systematic approach starting by looking at 3 main characteristics: (i) throughput, (ii) range (iii) energy efficiency requirements. Choosing a network topology turns out to be a trade of between these 3 dimensions. When a technology has a high throughput (e.g WiFi, Bluetooth), it has a low maximum range. When a technology has a high energy efficiency (e.g LoRa), it has a low throughput [12], [19], [20].

We noticed that very few network technologies allow for having peer to peer connections and wide range coverage. On top of our knowledge, only DASH7 Alliance [21], [22] proposes a wide range coverage and peer-to-peer possibility. Most of the LPWAN technologies (including LoRA and NbIoT) rely on a star topology, with a dedicated gateway as the center of the star topology [12], [19], [20], [23].

As previously stated, for our use-case, it is crucial to cover large areas. Nodes are usually separated by couple hundreds of meters. They are also supposed to be energy efficient, to survive for almost a year. They cannot have a heavy set-ups (antennas, batteries) because they are physically carried by humans and deployed in protected environments. In the few areas where they are, the monitored areas are scarcely covered by cellular towers. The only relevant and possible choice is to use Low Power Wide Area Network (LPWAN) technologies, that includes LoRa and NbIoT, with the hypothesis that a local-gateway is available for a given deployment to create an isolated star topology, depicted in Figure 1.

B. Large scale deployments and literature hypothesis

Large scale deployments can be understood in two dimensions: (i) number of devices or (ii) area covered. Such deployments can be found in different domains such as Wireless Sensor Networks (WSN), Internet of Things (IoT) with edge deployments (or so called extreme-edge deployments).

The hypothesis of WSN are usually linked to the fact that (i) nodes are only monitoring their environment to send data back to a centralized point, (ii) network coverage is not excellent, forcing them to connect through ad-hoc technologies [5], [24].

For edge related deployments, hypothesis are the following ones (i) connection to back-end and good coverage with usually multiple network technologies are expected, (ii) a strong connection with the cloud is expected (for services usage such as computation and data gathering) [10], [25].

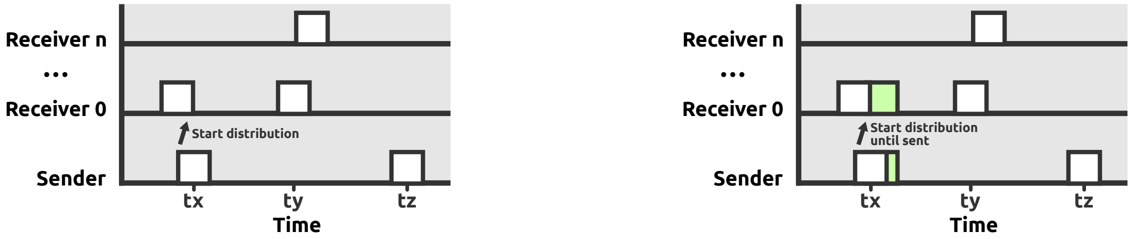
In our use-case, we are in the middle of these two literature. We are large scale in the area covered. We want to be large scale in the number of devices but we are limited with the regulations in terms of deployment [15]. We want to have back-end connections to deliver the data to the scientist, but we do not have good coverage (in the most optimistic cases) to allow every node to have such a characteristic. We want to have an ad-hoc connection through neighbour nodes but unlike WSN deployments, we have very few nodes. Furthermore, they are deployed under snow and rocks, separated from each other by a couple of hundred meters as a minimal distance, avoiding the ad-hoc capabilities that technologies used in WSN literature (e.g WiFi, Bluetooth) could offer.

C. Data dissemination in large scale deployment

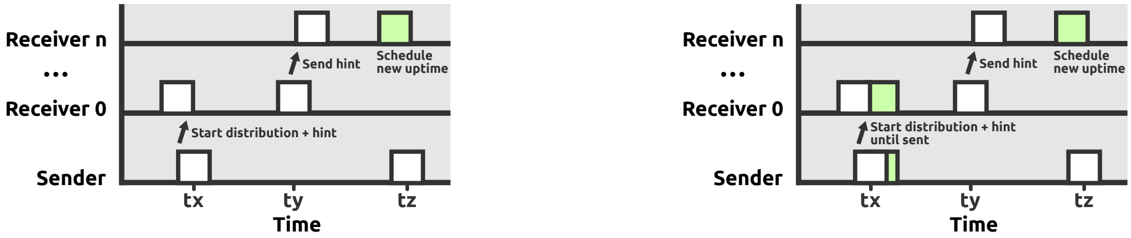
Multiple energy efficient policies can be found in the literature for large scale deployments. We focus on 4 types of contributions.

Authors in [26] targeted the connectivity for mobile nodes and energy conservation in WSN. They propose energy efficient protocols for data dissemination in dense sensor environment, where failure of multiple sensing devices is not a problem for the overall sensing. Energy efficient choices on data dissemination are done in function of the observations needs. It is not our case, as we do not want to disturb the scheduled and unexpected observations. Thus, such solution could not be used in our context. First, we need to quantify the impact of one instrument that needs to disseminate data to its neighbours on both its own and the overall energy budget.

Solutions that deal with reducing redundant transmission to be energy efficient, like in [27], usually comes with the hypothesis that sensors are part of a grid. In the case of a deployment in a scarce-ressource environment such as the Arctic Tundra, it won't be beneficial to have such a representation as the nodes are (i) few in numbers, (ii) far from each other and



(a) Baseline (b) Extended



(c) Hints (d) Combination Hints and Extended

Fig. 2. Sender and receivers lifetime, with impact of proposed policies on observation nodes' uptimes and communication. Messages, uptimes and added uptimes are represented as arrows, gray and green rectangles, respectively.

most importantly (iii) must implement shutdown policies and thus be OFF most of the time.

Works like [28] are providing policies to deal with nodes that fail on the field. These type of contributions are effective for a limited number of failures, which is expected as authors don't expect to see all nodes failing in a deployment. For the Arctic tundra, we are in the opposite case. We expect all nodes to not be available most of the time, because of independent shutdown policies embedded on each node, trying to live as long as possible. Node suddenly shutting down unexpectedly is equivalent to a node failing, for a neighbour.

A resource limited environment such as the Arctic tundra imposes conditions where it is complicated to evaluate when available ideas to disseminate data in an energy efficient way have a positive impact, as chosen hypothesis cannot match our realities. Quantification of loosely coupled policies costs (here in energy and time) from calibrated values extracted from the literature under plausible hypothesis such as this work provides is essential to map realities to have answers to build upon.

IV. EXPERIMENTAL SETUP

This section presents the evaluated policies and the metrics used to evaluate them. A simulator description developed to experiment with communication related energy consumption is depicted, along with the simulated parameters and scenarios.

A. Policies

In this paper, we want to compare multiple loosely coupled policies for data dissemination in the context of our use-case. This section describes the chosen policies and their relevance in our context. Figure 2 presents a graphical representation of the following policies. Exchanged messages, original uptimes and modified uptimes are depicted with arrows, white and green rectangles, respectively. Undrawn time periods represents OFF periods of Observation Nodes.

Baseline represents the devices waking up randomly. We are simulating a set of devices in resource limited environment with randomly picked uptimes. The devices are OFF most of the time to save energy. We simulate a short wake-up, once every hour, to model a device that must wake-up to do observations. The chosen duration represents the time needed to boot, monitor the environment and finally go back to sleep. Depending on the bandwidth and size of distribution, the data distribution will either be a success or a failure.

Figure 2(a) presents the *Baseline* policy on a given example with three Observation Nodes. Only one uptime of the sender overlaps with the one of the receivers. A distribution from $Sender$ to $Receiver0$ starts around time t_x.

Baseline is essential to evaluate the impact of the use-case on relevant metrics when no policy is activated. Thus, the following policies are compared to *Baseline*.

Extended implies that when an exchange starts (i.e when the sender overlaps with a receiver and starts communication), the duration of the uptime for both sender and receiver are extended, until the exchange finishes.

Figure 2(b) presents *Extended* policy with the same Observation Nodes and uptimes as Figure 2(a). Here, we consider that the overlap is not enough to have a successful distribution. This policy extends the uptime of both sender and receiver Observation Node until the distribution is successful.

Extended is essential to evaluate how much we can leverage the overlap between sender and receivers to maximize the successful distributions.

Hints implies that receivers share hints they received from the sender, when their uptimes overlap. A hint is given by the sender at the start of a delivery by adding the time-stamp (only a few bits) of its next uptime. Even if the hints are very small size-wise, we include them in the simulated energy overhead. When a node has a hint and do not have a successful delivery yet, it adds to its schedule a new uptime, starting at the hint timestamp.

Figure 2(c) presents *Hints* policy with the same Observation Nodes and uptimes as Figure 2(a). The data are prefixed with a hint about the next uptime of the sender. Around t_x, $Sender$ starts a distribution with a hint about its next uptime (t_z) to $Receiver0$. Then, the hint gets distributed from $Receiver0$ to $ReceiverN$, thanks to an overlap at time t_y, independently from the $Sender$. As a consequence, $ReceiverN$ creates a new uptime around t_z, the given hint. This policy is essential to evaluate how much we can leverage the overlap between receivers, independently from the sender.

Combination Hints and Extended implies that both *Extended* and *Hints* policies are activated.

Figure 2(d) presents *Combination Hints and Extended* policy with the same Observation Nodes and uptimes as Figure 2(a). As it is a combination of both *Extended* and *Hints* policies, the combined effect of their respective impacts can studied. Around time t_x, the uptime of both $Sender$ and $Receiver0$ are extended, to have a successful distribution. At time t_y, a hint previously received from the $Sender$, is delivered from $Receiver0$ to $ReceiverN$. This hint is used by $ReceiverN$ to schedule a new uptime around t_z, to overlap with $Sender$. This policy is essential to evaluate the impact of leveraging both policies on the relevant metrics.

B. Metrics

The energy overhead, $\%eOvhd(p)$, represents the relative energy overhead for a given policy p compared to the *Baseline* policy. It is computed for the sender and the receivers. For readability, it is displayed as a percentage.

$$\%eOvhd(p) = \frac{energyConsumed_p * 100}{energyConsumed_{Baseline}} - 100 \qquad (1)$$

$energyConsumed_p$ and $energyConsumed_{Baseline}$ represent the energy consumed (in Joules) during the complete simulated scenarios of a policy p and $Baseline$, respectively.

The uptime overhead $upOvhd(p)$ represents the uptime added by using policy p compared to the $Baseline$.

$$upOvhd(p) = AccUptime_p - AccUptime_{Baseline} \qquad (2)$$

The accumulated uptime $AccUptime_p$ represents the sum of all uptimes, during the simulation of policy p in a given scenario. It is expressed in seconds.

The policy efficiency $eff(p)$ represents the energy consumption (in Joule) per number of delivery success.

$$eff(p) = energyConsumed_p/\#Succ_p \qquad (3)$$

With $\#Succ_p$ that represents the number of data delivery success for the policy p. The policy efficiency metric helps to evaluate the energy efficiency of each policy where a low value means better energy efficiency.

TABLE I
SUMMARY OF SIMULATION PARAMETERS

Bandwidth (Ltnc)	LoRa	50kbps (0s) [29], [30]
	NbIoT	200kbps (0s) [29]
Energy states	P_{idle}	0.4W [31]
	LoRa	+0.16W (+32mA at 5V) [32]
	NbIoT	+0.65W (+130mA at 5V) [32]
Uptime	Long	3 min/hour
	Short	1 min/hour
Data size	1MB	
# Receivers	12	

C. Simulation

To evaluate the different policies, we propose to use flow-level network simulations. This approach has several benefits. First, it allows to save time compared to real experimentations. Indeed, we were able to simulate more than 8 years of uptimes for a set of Observation Nodes (i.e 1 sender and 12 receivers). Then, simulation offers reproducibility which is crucial to compare the policies based on the same initial conditions. Hence, by using real parameters from the literature, simulation offers accurate results that can be analyzed. Simulation campaigns are done to show how much energy, uptime overhead and successful update distributions can be expected by using a given policy instead of *Baseline*.

Simulation aim and metric computation:

The aim is to simulate: (i) 24 hours of sparse random uptimes (one each hour, for a given duration) for both sender and 12 neighbours potential receivers, (ii) a sender that is the only owner of the data, tries to successfully deliver a distribution to each 12 neighbours, (iii) 12 neighbours that randomly wake up once every hour to do observations and listen to potential messages, (iv) following a given policy chosen at start of simulation for both receivers and sender.

The metrics will be presented as averages of 200 runs. Each run will have a different uptime distribution for sender and receivers. Each set of distribution (receiver and senders) is run for each defined policy. For each simulation, receivers value for each metric will be an average of all 12 receivers.

Network and energy simulation:

In this work, the simulations are implemented using the SimGrid simulation framework [33], [34]. SimGrid is a flow-level network simulator which allows for efficient simulation of distributed applications by mean of strongly validated models. Network performance are express in terms of bandwidth and latency. Besides implementing our policies and simulating the network, SimGrid offers energy models that we can instantiate with energy parameters from the literature to predict the energy consumption of the Observation Nodes.

D. Simulation parameters:

Due to the characteristics of our use-case, LPWAN technologies is the only usable family of network technologies to achieve node to node communications, in the Arctic tundra.

Thus, we assume that each simulated deployment has an already deployed local wireless gateway to form a star topology to create local neighbourhoods, as depicted in Figure 1.

The values of the simulation parameters are displayed Table I. It is important to note that we considered the latency to be 0 sec in our scenarios. Indeed, despite the fact that NbIoT or LoRa may reach 10s of latency under dense scenarios [29], [32], we make the hypothesis that there is only one sender under an isolated environment (the Arctic tundra). To this end, we assume that there is no concurrency to access the wireless medium and thus, no latency except for the signal propagation speed in the air that we neglect.

For the idle state, we simulate a Raspberry Pi Zero [31]. This device has the advantage of having characteristics between a regular Raspberry Pi and a micro-controller based device. The worst case scenario for communication energy consumption would be that receiving and sending consume the same amount of energy (as receiving typically consumes less energy). Thus, when a device communicates (send or receive), we add $0.16W$ and $0.65W$ to P_{idle} to simulate a communication phase using $LoRa$ or $NbIoT$, respectively [31], [32].

We are simulating how an Observation Node located at the Arctic Tundra can randomly wake up, without coordination with others, to observe an event, monitor the observed event and go back to sleep to save energy. We consider a long uptime and short uptime to be equal to 3 and 1 minutes respectively. Uptimes are randomly picked, one every hour. 3 minutes is considered to be long because it is enough for all scenarios to have a successful delivery, if overlap starts at the beginning of both uptimes. We simulate one day (24 hours) on each run.

In both network technologies cases, and as previously discussed, the size of the disseminated data is not expected to be very high. In our case, we will simulate 1 MB as the size of the expected distribution. Due to the low bandwidth of possible network technologies (here LoRa and NbIoT), 1MB could already be a worst case scenario.

V. EVALUATION

In this section, we present the results of simulation for previously described scenarios and parameters. We simulate 200 random uptime distributions, on which we apply the 4 described policies. From these runs, we measure each studied metric from the simulator. For each scenario, we display two types of bar chart. One related to the energy metrics and the other to the accumulated uptime. Regarding the energy metrics bar charts, four values are shown. First, the average energy consumption, then $eOvhd(p)$ is in parenthesis, $eff(p)$ in curly brackets and $\#Succ_p$ in square brackets. For the accumulated uptime bar charts, two values are shown. First, the accumulated uptime and then $upOvhd(p)$ is in parenthesis. All these metrics are computed as an average of the 200 runs for the senders and the receivers. The standard deviations are represented with error bars. Results for 1 minute and 3 minutes uptimes are depicted in Figure 3 and Figure 4, respectively.

A. Scenario 1: Short uptime duration, LoRa

The first studied scenario comprises: (i) LoRa as the chosen network technology and (ii) an uptime duration of 1 minute. The simulation results are visible under the *LoRa* part on Figure 3. Since we chose a file size of 1 MB, 1 minute is not enough to have a successful delivery.

As expected, *Baseline* does not successfully deliver any file. This explains why the efficiency metric is not available (inf) for this scenario. But no success doesn't mean no overlaps and no tries. When overlaps exist between the sender and a receiver, the sender tries to make a delivery. This phenomenon can be seen with the non-null standard deviation for $Energy$ on both sender and receivers.

Similarly, *Hints* does not successfully deliver any file, as it does not change the uptime duration. $Hints$ adds new uptimes to the receivers. For these reasons, *Hints* is only an overhead when compared to *Baseline* in this context, with no benefits when it comes to number of successful deliveries.

Extended successfully delivers an average of 6 receivers over the 12 expected. It is expensive in terms of energy consumption for the sender, with $+62.7\%$ of energy overhead (when compared to *Baseline*). But the energy consumed by the receivers only have an average overhead of $+5.3\%$. This policy adds, in average, 11 min 25 sec (685.6 sec) and 58 sec to the senders and receiver's accumulated uptimes, respectively.

Combination Hints and Extended successfully delivers an average of 6.5 receivers. It is expensive for the sender as it adds $+67.8\%$ of energy overhead when compared to *Baseline*. For the receivers, it is more expensive than *Extended*, with an overhead of 6.6%. This policy adds, in average, 12 min 10 sec (730.5 sec) and 1 min 11 sec (71.6 sec) to the sender and the receiver's accumulated uptimes, respectively.

Thus, in such a context (where bandwidth and uptime duration are not enough to deliver the chosen size), choosing *Hints* or *Baseline* would have been a mistake as they only add overhead. A policy using *Extended* is necessary to have successful deliveries. *Combined Hints and Extended* is useful to reach most of the receivers, with an important uptime overhead especially for the sender.

B. Scenario 2: Short uptime duration, NbIoT

The second studied scenario comprises: (i) NbIoT as the chosen network technology and (ii) an uptime duration of 1 minute. The simulation results are visible under the *NbIoT* part of Figure 3. Thanks to the bandwidth of NbIoT, 1 min is enough to transmit a file of 1 MB.

Even if the scenario allows successful deliveries very few successful deliveries are witnessed on *Baseline* (2.4). This is due to the sparse and independent distribution of uptimes leading to few overlaps between sender and receivers.

With *Extended*, more receivers get the data successfully (6.4, in average). An energy overhead of 6.4% and -0.4% is measured for sender and receivers, respectively. Hence, due to less transmission failures, receivers perform better than baseline in terms of energy consumption. This policy adds,

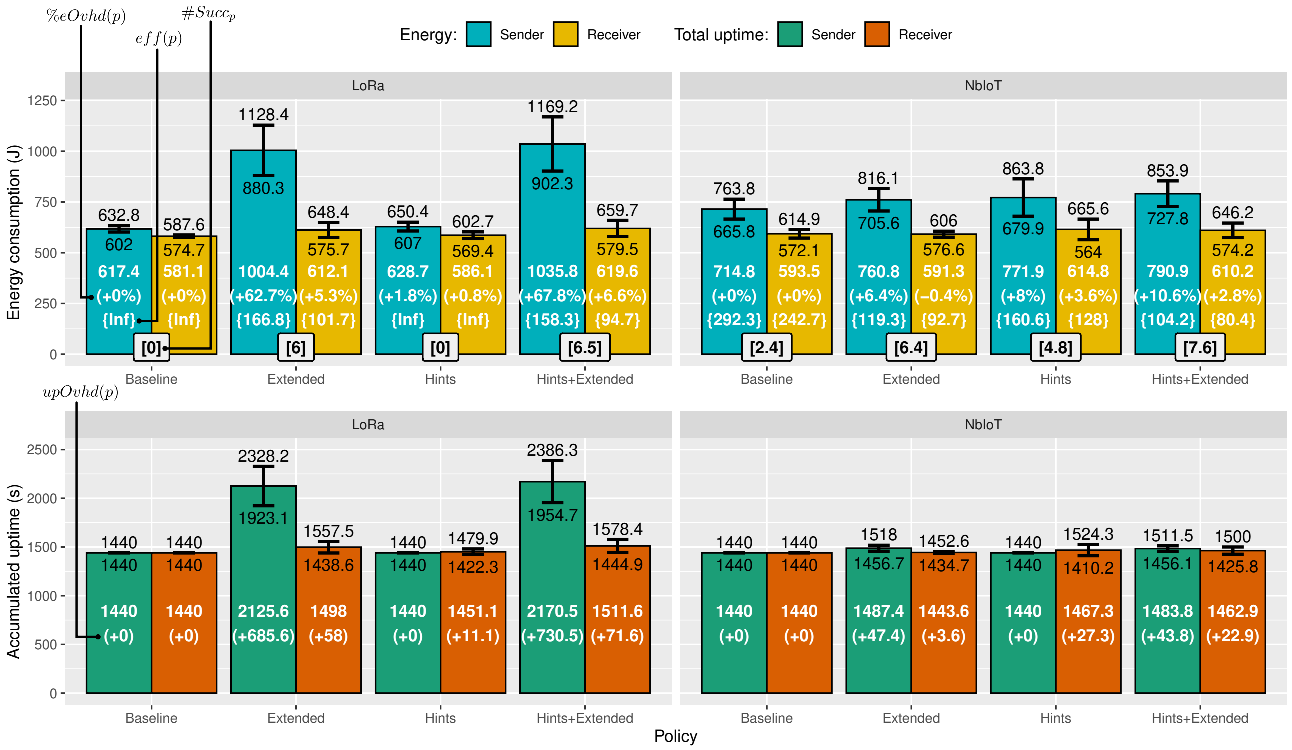


Fig. 3. Simulation results for 1 minutes uptime (average of 200 seeds with standard deviations as error bars). The upper bar charts display energy metrics, the other the uptime metrics. Each energy bar charts has four values: (i) the average energy consumption, (ii) $eOvhd(p)$ in parenthesis, (iii) $eff(p)$ in curly brackets and (iv) $\#Succ_p$ in square brackets. For the uptime bar charts, two values are shown: (i) accumulated uptime and (ii) $upOvhd(p)$ is in parenthesis.

in average, 47 sec and 4 sec to the sender and receivers accumulated uptimes, respectively.

Now that the uptime duration permits of having successful distribution, the *Hints* policy has 4.8 data delivery success. It has an energy overhead of $+8\%$ for the sender and $+3.6\%$ for the receiver. This policy do not adds accumulated uptime for the sender but adds 27s for the receiver.

Combination Hints and Extended reaches 7.6 successful deliveries (closer to the *Extended* policy). Although, it is more expensive compare to the *Hints* policy for the sender with an energy overhead of $+10.6\%$ and cheaper for the receivers, with 2.8% of energy overhead. This policy adds, in average, 44 sec and 23 sec to the accumulated uptime of the sender and receivers, respectively.

Thus, for this scenario and for all policies, we notice that all senders have a bigger overhead in terms of time and energy than the receivers. *Combination Hints and Extended* is the best trade-off to maximize the number of successful deliveries and minimize the overhead of energy consumed by the receivers, at the expenses of the sender. Moreover, this policy has the best energy efficiency which range from 104.2 to 158.3 Joules per delivery success for the sender and from 80.4 to 94.7 Joules per delivery success for the receiver.

C. Scenario 3: Long uptime duration, LoRa

The third studied scenario comprises: (i) LoRa as the chosen network technology and (ii) an uptime duration on the field equal to 3 minutes. The simulation results are shown under the *LoRa* part of Figure 4. In this context, with 3 minutes uptime, it is enough to have successful distributions of 1MB files.

Even if the scenario allows successful deliveries, very few success are witnessed on *Baseline* (2.2). Again this is due to the sparse and independent distribution of uptimes leading to few overlaps between sender and receivers.

Extended is, in such a context, very good concerning successful deliveries, with an average of 10.9. An energy overhead of $+8.3\%$ and 0.1% are shown for sender and receivers, respectively. This policy adds, in average, 8 min 9 sec and 39 sec to the sender and receivers accumulated uptimes, respectively.

Hints delivers 11.2 distributions successfully, with and energy overhead of $+4.9\%$ and $+17.5\%$ for sender and receivers, respectively. This policy adds, in average, 9 min 16 sec to the receivers accumulated uptime. No added uptime is measured at the sender. This policy do not perform well regarding the receivers energy efficiency with 185.6 Joules per delivery success. Indeed, since the *Hints* policy adds new uptimes and hint forwarding, it induces more energy consumption.

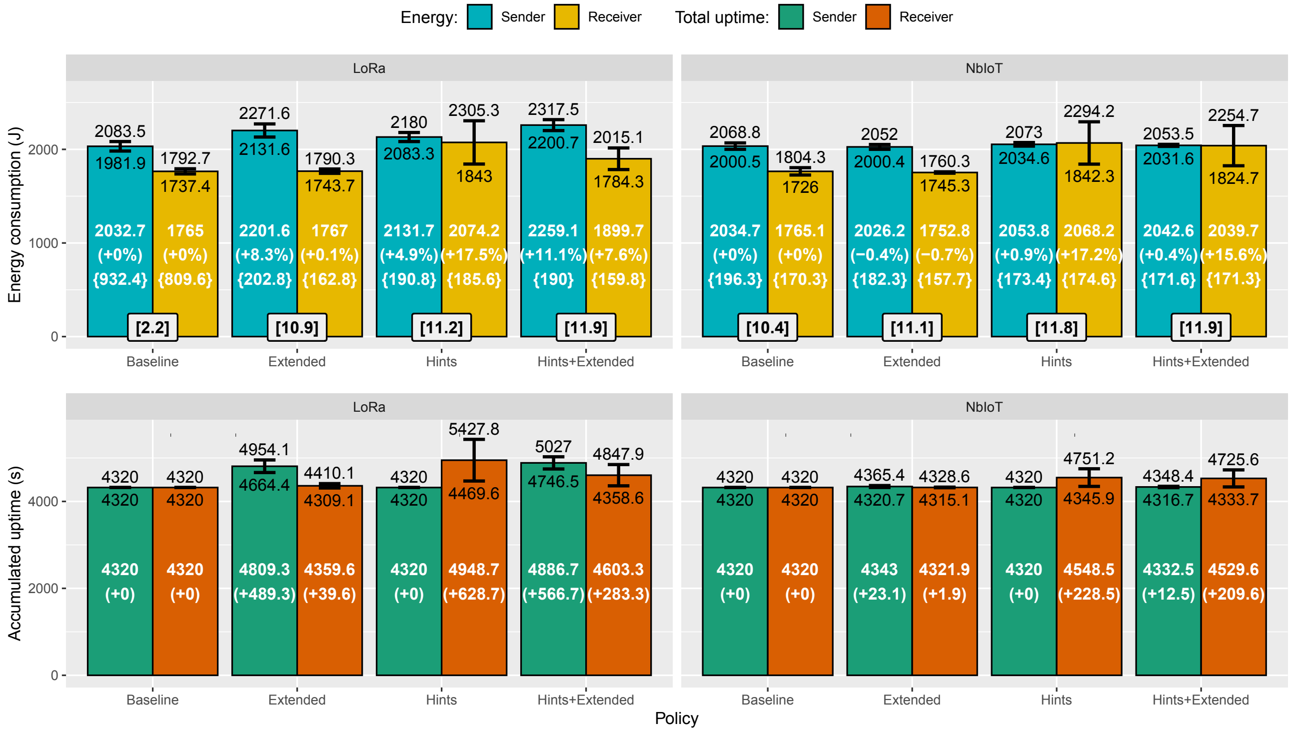


Fig. 4. Simulation results for 3 minutes uptime (average of 200 seeds with standard deviations as error bars). The upper bar charts display energy metrics, the other the uptime metrics. Each energy bar charts has four values: (i) the average energy consumption, (ii) $eOvhd(p)$ in parenthesis, (iii) $eff(p)$ in curly brackets and (iv) $\#Succ_p$ in square brackets. For the uptime bar charts, two values are shown: (i) accumulated uptime and (ii) $upOvhd(p)$ is in parenthesis.

Combination Hints and Extended reaches almost all deliveries, with an average of 11.9. An energy overhead of $+11.1\%$ and $+7.6\%$ is measured for sender and receivers, respectively. This policy adds, in average, 9 min 26 sec and 4 min 43 sec to the accumulated uptime of the sender and the receivers.

For this scenario, *Extended* is the best compromise between high deliveries and reduced energy consumption for the receivers. *Hints* reduces the energy consumption of sender, at the expenses of the receivers. *Combination Hints and Extended* is the best compromise to reach all deliveries and low receiver overheads, at the expenses of the sender.

D. Scenario 4: Long uptime duration, NbIoT

The fourth studied scenario comprises: (i) NbIoT as the chosen network technology and (ii) an uptime duration equal to 3 minutes. The simulation results are displayed in the *NbIoT* part of Figure 4. As a reminder, we chose a fixed file size of 1MB. In such a context, 3 minutes is enough to have a successful distribution.

Such a context allows *Baseline* to reach 10.4 successful deliveries. Thus, the impact of the sparse and independent distribution of uptimes is not as strong as in previous scenarios.

Extended successfully delivers 11.1 for an energy overhead of -0.4% and -0.7% for sender and receivers, respectively. Notice that the overheads are negative, meaning that in average we reduced the energy consumption because of less transmission failures. This policy adds, in average, 23 sec and 2 sec to the accumulated uptime. Notice that greater accumulated uptime does not necessarily mean greater energy consumption. Here, *Extended* is slightly more efficient than *Baseline* and reduces the energy consumption of the Observation Nodes.

Hints reaches 11.8 successful deliveries, with an energy overhead of $+0.9\%$ and $+17\%$ for the sender and receivers. This policy adds, in average, 3 min 48 sec to the accumulated uptime of the receivers. No uptime overhead is measured at the sender side. We note that the use of hint has a clear impact on the receiver energy efficiency. Indeed, this policy performs the worse with an average efficiency of 174.6 Joules per delivery success for the receivers.

Combination Hints and Extended also reaches almost all deliveries with an average of 11.9. An energy overhead of $+0.4\%$ and $+15.6\%$ is measured for sender and receivers, respectively. This policy adds, in average, 12 sec and 3 min 29 sec to the accumulated uptime of the sender and receiver, respectively.

Thus, for this scenario and when a policy is activated, the successful deliveries all hovers around the 12 (i.e between 10.4 and 11.9). Globally when a policy is activated, a lower overhead is measured when compared to *Baseline* (between -0.4% and $+0.9\%$ for the sender and between 0.7% and $+17\%$ for the receivers). The best compromise to reduce the energy overhead at the receiver side and maximize the number of successful deliveries is either choosing *Combination Hints and Extended* or *Extended*. Similarly to the short uptime scenario, *Combination Hints and Extended* policy offers the best energy efficiency ranging from 171.6 to 190 Joules per delivery success for the sender and from 159.8 to 171.3 Joules per delivery success for the receiver.

VI. LESSONS LEARNED

This section presents the lessons learned from the results of this simulation campaign.

A. Choosing a policy, under several scenarios

From the 4 previously studied scenarios, we can noticed common behaviours. Regarding energy consumption, *Extended* is expensive for the sender but has low overhead for the receivers. *Hints* adds a non negligible amount of accumulated uptimes to the receivers, which does not always translate into a bigger energy overhead for the receivers compared to the sender. *Combination Hints and Extended* always has an energy overhead for the sender close to the one measured on *Extended*. For the receivers, *Combination Hints and Extended* has an overhead closer to the one seen on *Extended*, for a number of successful deliveries closer to *Hints*.

In a Cyber-Physical System like ours, where most of the nodes are independent and deployed in a scarce resources environment, we want the energy consumed by an Observation Node to depend on itself first. When a node asks the group for help, it should have the largest energy overhead. It would not be fair to consume the groups' energy to absorb the impact of its own actions (except maybe in very critical cases). In such a context, we should aim for maximizing the number of successful deliveries and reducing the overhead of energy consumption for the receivers. *Combination Hints and Extended* seems like the best compromise.

In fact, as seen on previous simulated scenarios, this policy permits to achieve good number of data deliveries even on short uptime scenarios. In addition, this policy allows to be very close to the number of deliveries given by *Hints* (when *Hints* outperforms all others) with the minimum of impact on the receivers energy overhead.

B. Choosing a network technology

These experiments permit to compare the impact of choosing either *LoRa* or *NbIoT* when a node aims at disseminating data to its neighbours. Except for the baseline (in both chosen uptime duration), the average energy consumption of the senders is almost always lower for *NbIoT*. Similar trends can be observed for the energy consumed by the receivers, except for *"1 min - Hints+Extended"*, *"1 min - Hints"* and *"3 min - Hints"* scenarios, where Lora is negligibly better than *nbIot*.

Concerning the number of successful deliveries, when policies are not activated, *NbIoT* is obviously better. When policies are activated, both *LoRa* and *NbIoT* are comparable and within the standard deviation, except when the uptime is not enough for *LoRa* to have any successful delivery (i.e "1 min - Hints"). For slightly energy efficient receivers and successful deliveries, choosing *NbIoT* seems to be the right choice.

This paper does not investigate about the uptime duration. We assume that the duration of an uptime, for an Observation Node, is static and includes constraints such as boot-up times or sensors that needs to warm-up. There is room for improvement in this dimension, especially in the policies involving *Hints*. Indeed, in this paper, we simply added an uptime with the same duration as the one set for the experiment. By doing so, the energy overhead for the receivers for *Hints* and *Combination Hints and Extended* would be even lower (which is already an argument for choosing these policies).

The energy efficiency depends on (i) the network technology, (ii) the consumption of the nodes (idle and during wireless communications), (iii) the size of the data to transmit, (iv) the current bandwidth between two nodes. We explored (i) - (ii) in our simulation, and fixed (iii)-(iv) with realistic values for our use-case. From the presented results, it is not obvious what are the good choices for these parameters. Thus, our next future work includes a model that determined what policy should be chosen for given values for these parameters, to be energy efficient and maximize successful deliveries.

VII. CONCLUSION

Connected devices working from batteries are flourishing everywhere around us. Reducing the energy consumed during communication periods is crucial. It is even more crucial when it comes to large scale battery based deployments done in scarce resources environments such as the Arctic Tundra. The DAO-CPS project is in this specific case. We propose to quantify the energy and time overhead for data dissemination in this context. We study 3 loosely coupled policies that we compare to a baseline, where no policy is activated.

We simulate an existing deployment with randomly picked uptimes, that allows nodes to wake-up randomly every hour, for a very short duration (1 and 3 minutes) and potentially communicate. We simulate communication through plausible network technologies, LoRa and NbIoT. One node needs to disseminate its data to its neighbours. We compare the number of successful distributions achieved by each policies over their respective overheads, in energy and time.

Evaluation shows that the best choice concerning the policy depends on the characteristic of the environment. When the uptime is too small for the size of the delivery and bandwidth to be sent, *Hints* and *Baseline* policies are very bad in terms of successful deliveries. When the uptime and bandwidth is enough to have a successful delivery a trade-off exists between *Extended*, *Hints* and *Combination Hints and Extended*, depending on which overhead is prioritized (sender or receivers). When the uptime is more than enough to have a successful delivery under the chosen bandwidth, the policies still help to acheive more successful deliveries, for low overheads. An overall good choice, in all cases, stays the *Combination Hints and Extended* one, which usually has a slightly higher overhead than *Extended* policy for the sender but a lower overhead than *Hints* policy for the receivers, for a very good number of successful delivery.

As a future work, we plan to extract a model that will dynamical help a node to choose a policy according to current or predicted environmental characteristics. Such a model could be embedded in instruments used in real life deployments.

ACKNOWLEDGMENT

The DAO project is supported by the Research Council of Norway (RCN) IKTPluss program, project number 270672. Thank you very much to the COAT ecologists, UiT.

REFERENCES

[1] I. Rodero and M. Parashar, "Data cyber-infrastructure for end-to-end science: Experiences from the nsf ocean observatories initiative," *Computing in Science & Engineering*, 2019.

[2] T. Adame, A. Bel, A. Carreras, J. Melia-Segui, M. Oliver, and R. Pous, "Cuidats: An rfid–wsn hybrid monitoring system for smart health care environments," *Future Generation Computer Systems*, vol. 78, pp. 602–615, 2018.

[3] L. Liu, W. Liu, Y. Zheng, H. Ma, and C. Zhang, "Third-eye: a mobilephone-enabled crowdsensing system for air quality monitoring," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, no. 1, pp. 1–26, 2018.

[4] X. Sun and E. J. Coyle, "Quantization, channel compensation, and optimal energy allocation for estimation in sensor networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 8, no. 2, pp. 1–25, 2012.

[5] F. T. Jaigirdar and M. M. Islam, "A new cost-effective approach for battlefield surveillance in wireless sensor networks," in *2016 International Conference on Networking Systems and Security (NSysS)*. IEEE, 2016, pp. 1–6.

[6] C.-R. Rad, O. Hancu, I.-A. Takacs, and G. Olteanu, "Smart monitoring of potato crop: a cyber-physical system architecture model in the field of precision agriculture," *Agriculture and Agricultural Science Procedia*, vol. 6, pp. 73–79, 2015.

[7] V. Jelicic, M. Magno, D. Brunelli, G. Paci, and L. Benini, "Context-adaptive multimodal wireless sensor network for energy-efficient gas monitoring," *IEEE Sensors journal*, vol. 13, no. 1, pp. 328–338, 2012.

[8] H. Demirkan, "A smart healthcare systems framework," *It Professional*, vol. 15, no. 5, pp. 38–45, 2013.

[9] A. Vulimiri, C. Curino, B. Godfrey, K. Karanasos, and G. Varghese, "Wanalytics: Analytics for a geo-distributed data-intensive world." in *CIDR*, 2015.

[10] A. M. Rahmani *et al.*, "Exploiting smart e-health gateways at the edge of healthcare internet-of-things: A fog computing approach," *Future Generation Computer Systems*, vol. 78, pp. 641–658, 2018.

[11] S. Sakib, M. M. Fouda, Z. M. Fadlullah, and N. Nasser, "Migrating intelligence from cloud to ultra-edge smart iot sensor based on deep learning: An arrhythmia monitoring use-case," in *2020 International Wireless Communications and Mobile Computing (IWCMC)*. IEEE, 2020, pp. 595–600.

[12] K. Mekki, E. Bajic, F. Chaxel, and F. Meyer, "A comparative study of lpwan technologies for large-scale iot deployment," *ICT express*, vol. 5, no. 1, pp. 1–7, 2019.

[13] R. A. Ims, J. U. Jepsen, A. Stien, and N. G. Yoccoz, "Science plan for coat: climate-ecological observatory for arctic tundra," *Fram Centre report series*, vol. 1, pp. 1–177, 2013.

[14] I. Raïs, O. Anshus, J. M. Bjørndalen, D. Balouek-Thomert, and M. Parashar, "Trading data size and cnn confidence score for energy efficient cps node communications," in *2020 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID)*. IEEE, 2020, pp. 469–478.

[15] I. Raïs, J. M. Bjørndalen, P. H. Ha, K.-A. Jensen, L. S. Michalik, H. Mjøen, Ø. Tveito, and O. Anshus, "Uavs as a leverage to provide energy and network for cyber-physical observation units on the arctic tundra," in *2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS)*. IEEE, 2019, pp. 625–632.

[16] P. H. HAa, R. A. IMSb, J. U. JEPSEN, S. T. KILLENGREEN, E. F. KLEIVEN, E. M. SOININEN, N. G. YOCCOZ, and A. HORSCH, "Building a sensor system for a large scale arctic observatory," *Communicating Process Architectures 2015 & 2016: WoTUG-37 & WoTUG-38*, vol. 69, p. 445, 2018.

[17] E. M. Soininen, I. Jensvoll, S. T. Killengreen, and R. A. Ims, "Under the snow: a new camera trap opens the white box of subnivean ecology," *Remote Sensing in Ecology and Conservation*, vol. 1, no. 1, pp. 29–38, 2015.

[18] M. J. Murphy, O. Tveito, E. F. Kleiven, I. Rais, E. M. Soininen, J. M. Bjorndalen, 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)*. Melbourne, Australia: IEEE, May 2021, pp. 376–385. [Online]. Available: https://ieeexplore.ieee.org/document/9499357/

[19] Q. M. Qadir, T. A. Rashid, N. K. Al-Salihi, B. Ismael, A. A. Kist, and Z. Zhang, "Low power wide area networks: a survey of enabling technologies, applications and interoperability needs," *IEEE Access*, vol. 6, pp. 77 454–77 473, 2018.

[20] H. Wang and A. O. Fapojuwo, "A survey of enabling technologies of low power and long range machine-to-machine communications," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2621–2639, 2017.

[21] M. Weyn, G. Ergeerts, R. Berkvens, B. Wojciechowski, and Y. Tabakov, "Dash7 alliance protocol 1.0: Low-power, mid-range sensor and actuator communication," in *2015 IEEE Conference on Standards for Communications and Networking (CSCN)*. IEEE, 2015, pp. 54–59.

[22] M. Weyn, G. Ergeerts, L. Wante, C. Vercauteren, and P. Hellinckx, "Survey of the dash7 alliance protocol for 433 mhz wireless sensor communication," *International Journal of Distributed Sensor Networks*, vol. 9, no. 12, p. 870430, 2013.

[23] J. Finnegan and S. Brown, "A comparative survey of lpwa networking," *arXiv preprint arXiv:1802.04222*, 2018.

[24] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer networks*, vol. 52, no. 12, pp. 2292–2330, 2008.

[25] R.-A. Cherrueau, A. Lebre, D. Pertin, F. Wuhib, and J. M. Soares, "Edge computing resource management system: a critical building block! initiating the debate via openstack," in {*USENIX*} *Workshop on Hot Topics in Edge Computing (HotEdge 18)*, 2018.

[26] Z. Zhou, X. Xang, X. Wang, and J. Pan, "An energy-efficient data-dissemination protocol in wireless sensor networks," in *2006 International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM'06)*. IEEE, 2006, pp. 10–pp.

[27] H. Sabbineni and K. Chakrabarty, "Location-aided flooding: an energy-efficient data dissemination protocol for wireless-sensor networks," *IEEE transactions on computers*, vol. 54, no. 1, pp. 36–46, 2005.

[28] A. Antonopoulos and C. Verikoukis, "Multi-player game theoretic mac strategies for energy efficient data dissemination," *IEEE Transactions on Wireless Communications*, vol. 13, no. 2, pp. 592–603, 2013.

[29] K. Mekki, E. Bajic, F. Chaxel, and F. Meyer, "Overview of cellular lpwan technologies for iot deployment: Sigfox, lorawan, and nb-iot," in *2018 ieee international conference on pervasive computing and communications workshops (percom workshops)*. IEEE, 2018, pp. 197–202.

[30] loraAlliance, "What is the lorawan® specification ? (2020)," https://lora-alliance.org/about-lorawan, 2020.

[31] J. Geerling, "Power consumption benchmarks, raspberry pi," https://www.pidramble.com/wiki/benchmarks/power-consumption, 2020.

[32] R. S. Sinha, Y. Wei, and S.-H. Hwang, "A survey on lpwa technology: Lora and nb-iot," *Ict Express*, vol. 3, no. 1, pp. 14–21, 2017.

[33] H. Casanova and L. Marchal, "A network model for simulation of grid application," 2002.

[34] P. Velho and A. Legrand, "Accuracy study and improvement of network simulation in the simgrid framework," in *Proceedings of the 2nd International Conference on Simulation Tools and Techniques*. ICST (Institute for Computer Sciences, Social-Informatics and …, 2009, p. 13.