Impact of loosely coupled data dissemination policies for resource challenged environements

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

***Abstract*—A Cyber-Physical System (CPS) deployed in a scarce resources environment can face multiple constraints like (i) no network coverage, (ii) no possibility of energy replenishment, (iii) no possible human interactions, (iv) high chances of failures due to environmental factors like snow, floods or wild animals. Devices being part of such a CPS must be battery powered and very energy efficient to achieve long life-time. Although, the CPS still has to disseminate data, for example, to increase resiliency, safe keep results or update nodes. A trade-off between energy spent and data dissemination needs to be found, ideally by minimizing the energy overhead while maximizing the dissemination.**

In this paper we evaluate and discuss the efficiency (in energy, time and number of successful distributions) of multiple data distribution policies. We report on the trade-off between (i) successful data dissemination and (ii) energy and up-time overheads implied by 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 experimental platforms.

Results show that a performant policy in a given scenario can be terrible 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 maximize the number of dissemination,

***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 often limited by the availability of sufficient energy. The reporting is also limited by the availability of a data network with sufficient bandwidth and latency. The opportunities provided are consequently limited by the availability of 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. Although, they are also 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 objective is to limit the energy overhead while increasing the number of successful disseminations when using LPWAN technologies. We focus on policies that don't 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 applicable on scarce-resource deployments;
- Quantify the impact of chosen policies on energy and up-time 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 deployement and their characteristics. Section III presents the related work. Section IV presents the experimental setup that includes a description of the policies, the metrics, the simulation tool and scenarios. Section V presents the results of simulated scenarios on explored metrics. Section VI presents the lessons learned from the simulation campaigns. Finally VII concludes this work.

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]. Consequently, to accurately detect climate change, larger observations of the arctic tundra are needed.

The COAT 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 creates 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 and instruments 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, for such a deployment and use-case 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 probability of crash of deployed units (e.g through flooding, hardware failure). As a direct implication, we want to disseminate important results to as many neighbours as possible, to keep the data safe and reduce the chance of loosing results. 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, if it ever happens). Although, as it is complicated and expensive to physically access the arctic tundra, updates (e.g configuration files, executables, packets

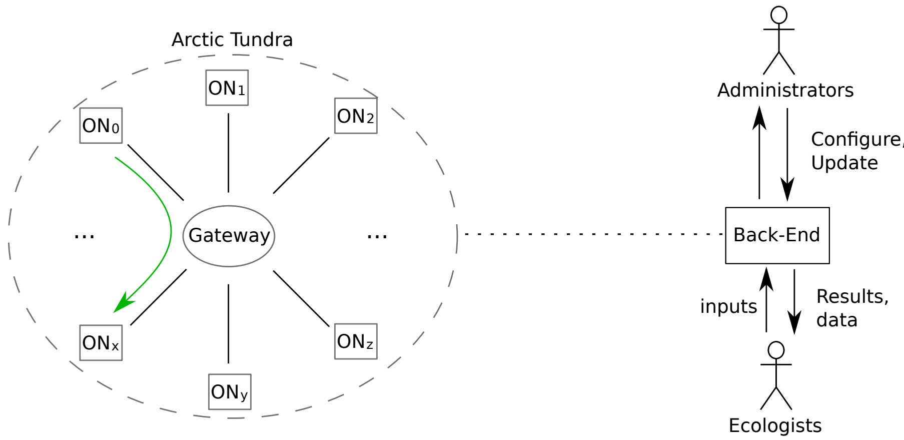


Fig. 1: Overview of the system imposed by the Arctic Tundra's characteristics. Back-end hosts a set of services [14]. Its connectivity to Observation Nodes (ON) deployed at the Arctic Tundra is sparse and unexpected. In this paper, we focus on data dissemination inside the isolated system deployed at the Arctic Tundra, forming a star topology. The wireless gateway at the middle of the topology is only used to do 1:1 communications between Observation Nodes (green arrow).

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 to previously exposed limitation in network technology choices, energy and computing capabilities available.

III. Related work

This section presents the related work concerning the network technologies available and 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 and finally (iii) energy efficiency needed. 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], [18], [19].

We noticed that very few network technologies propose a possibility of having peer to peer connections and wide range coverage. On top of our knowledge, only DASH7 Alliance [20], [21] proposes a wide range coverage and peer-to-peer possibility. Most of the LPWAN technologies (including LoRA and NbIoT) relay on a star topology, with a dedicated gateway as the center of the star topology [12], [18], [19], [22].

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 can't have a heavy set-ups (antennas, batteries) because they are physically carried by humans and deployed in protected environments. The monitored areas are scarcely covered by cellular towers, in the few areas where they are. 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 multiple literature 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) that network coverage is not excellent, forcing them to connect through ad-hoc technologies [5], [23].

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], [24].

With 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 like for most edge deployments, but we don't 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 type of contributions.

Authors in [25] targeted the connectivity for mobile nodes and energy conservation in wireless sensor networks. They propose energy efficient protocols for data dissemination for heavily sensored 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 don't want to disturb the scheduled and unexpected observations. Thus, such solution could not be used in our context. We first 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 [26], usually comes with the hypothesis that sensors are part of a grid. In the case of a deployment in a scarce-ressource environement 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 most importantly (iii) must implement shutdown policies and thus be OFF most of the time.

Works like [27] 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 (like our arctic tundra use-case) 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 can't 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. Then, a description of the simulator 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 up-times and modified up-times are depicted with green arrows, gray and green rectangles, respectively. Uncolored time periods represents OFF periods of Observation Nodes.

Baseline represents the devices waking up randomly, with no activated policy. We are simulating a set of devices in resource limited environment with randomly picked up-times. The devices are OFF most of the time to save energy. We simulate a wake-up once every hour (for a short duration) to simulate 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.

Figure 2(a) presents *Baseline* policy on a given example of three Observation Nodes, one sender and two receivers. There is only one up-time where the sender overlaps with 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. Following policies are compared to *Baseline*. Depending on the bandwidth and size of distribution, the distribution will either be a success or a failure.

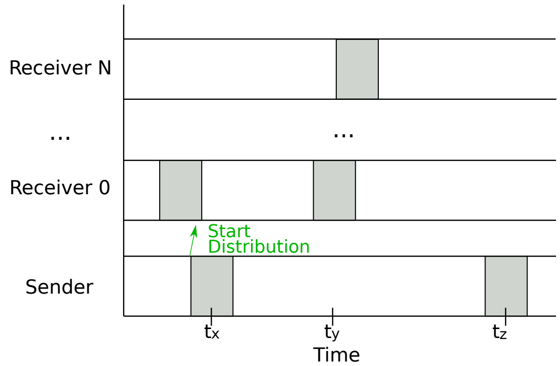
Extended implies that when an exchange starts (i.e when the sender overlaps with a receiver and starts communication), the duration of the up-time for both sender and receiver are extended, until the exchange finishes.

Figure 2(b) presents *Extended* policy with the same Observation Nodes and up-times as Figure 2(a). Here, we consider that the overlap is not enough to have a successful distribution. This policy extends the up-time of the Observation Node until the distribution is successful. Notice that both sender and receivers can have their up-times extended.

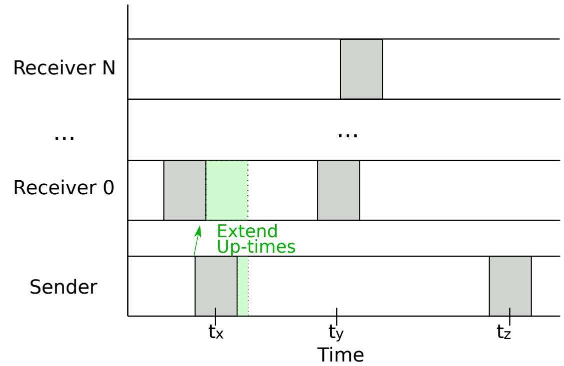
Extended is essential to evaluate how much we can leverage the overlap between sender and receivers' randomly chosen up-times to maximize the successful distributions.

Hint implies that receivers share hints they have received about the sender, when their up-times overlap. A hint is given, by the sender, at the start of a delivery by adding the timestamp (thus only a few bits) of its next up-time. Even if the hints are very small size-wise (both server to receiver and receiver to receiver), we include them in the simulated energy overhead. When a node has a hint and has not had a successful delivery yet, it adds a new up-time to its schedule, starting at the hinted timestamp.

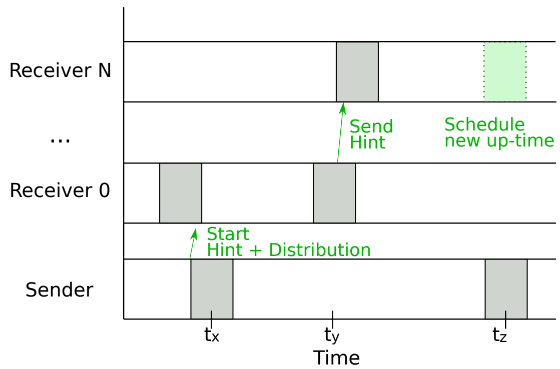
Figure 2(c) presents *Hints* policy with the same Observation Nodes and up-times as Figure 2(a). A distribution is slightly modified to first deliver the hint about the next up-time of the sender. Around t_x, $Sender$ starts a distribution with a hint about its next up-time (t_z) to $Receiver0$. The hint gets then distributed from $Receiver0$ to $ReceiverN$, thanks to an overlap at time t_y, independently from the $Sender$. As



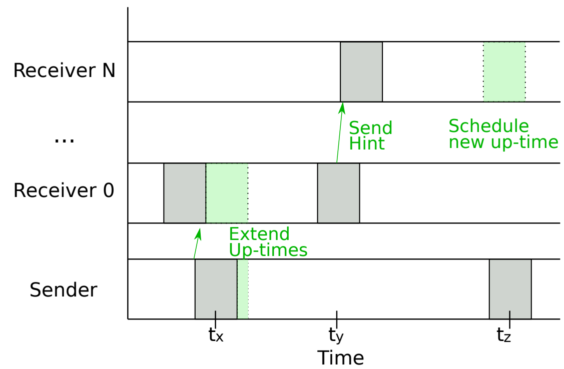
(a) Baseline



(b) Extended



(c) Hints



(d) Combination Hints and Extended

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

a consequence, $ReceiverN$ creates a new up-time 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 *Hint* policies are activated.

Figure 2(d) presents *Combination Hints and Extended* policy with the same Observation Nodes and up-times as Figure 2(a). As it is a combination of both *Extended* and *Hints* policies, their previously discussed impacts can be seen when using this policy. Around time t_x, the up-time of both $Sender$ and $Receiver0$ are extended, to have a successful distribution. At time t_y, a hint, previously received from $Sender$'s distribution, is delivered from $Receiver0$ to $ReceiverN$. This hint is used by $ReceiverN$ to schedule a new up-time around t_z, to overlap with $Sender$.

This policy is essential to evaluate the impact of leveraging both policies on relevant metrics, on both sender and receivers.

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.

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

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

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

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

The accumulated up-time $AccUptime_p$ represents the sum of all up-times, during the simulation of policy p in a given scenario. It is expressed in seconds.

Per simulation, we also measure the number of successful distributions, noted $\#Succ$.

C. Simulation

A simulator is implemented in order to evaluate the discussed policies in a wide variety of contexts, which would have been extremely time consuming to achieve in real life. For example, accumulated, we simulate more than 4 years of up-times for a set of Observation Nodes (i.e sender and 12

TABLE I: Summary of simulation parameters

Bandwidth (Ltnc)	LoRa	50 kbps (0 sec) [28], [29]
	NbIoT	200 kbps (10 sec) [28]
Energy State	P_{idle}	0.4W [30]
	LoRa	+0.16W (+32mA at 5V) [31]
	NbIoT	+0.65W (+130mA at 5V) [31]
Up-time	Long	3min /hour
	Short	1min /hour
Size data	1 MBytes	
# Receivers	12	

receivers). Simulation campaigns are done to show how much energy and up-time overhead and finally 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 up-times (one each hour, for a given duration) for both sender and 12 neighbours potential receivers, (ii) a sender that 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 sender is the only owner of the data and is the only one trying to have distributions.

The discussed metrics will be presented as averages of 100 runs. Each run will have a different up-time distribution for both sender and each receivers. Each set of distribution (receiver and senders) is run for each defined policy. For each simulation, receivers value (noted $Rcvrs$) for each metric will be an average of all 12 receivers. Each average value for a given metric will be followed by its standard deviation (in parenthesis).

Network and energy simulation:

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.

As shown in [32], [33], the time (T) to transfer data (S Bits) on one network link having bandwidth (bw) and latency ($Ltnc$) as characteristics, can be computed as :

$$T = Ltnc + \frac{S}{bw} \tag{3}$$

The simulator assumes that: (i) the bandwidth and latency don't change during the simulation, (ii) there is no congestion nor packet loss, (iii) there is no concurrent communication.

The energy consumed during a specific phase (either idle, send or receive) for a given node is equal to:

$$Energy \mathrel{+}= T_{State} * P_{State} \tag{4}$$

where P_{State} and T_{State} represents the power and time spent in a given state, respectively. Each scenario have its own values for power states. To increase accuracy, we simulate every second of the simulated scenarios.

D. Simulation parameters:

The values of the simulation parameters are displayed Table I. To accurately simulate the energy consumption when communicating, we also simulate the latency measured when using both network technologies. LoRa is claimed to be "insensible to latency" [28], we set it to 0 seconds. NbIoT is measured to have a latency "under 10 seconds" [28], [31]. To conservatively simulate the energy consumed, we set the latency of nbIoT to its maximum seen in the literature: 10 seconds.

For the idle state, we simulate a Raspberry Pi Zero [30]. 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. 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 [30], [31].

We are simulating how an Observation Node located at the Arctic Tundra can randomly wake up to observe an event (without coordination with others), monitor the observed event and go back to sleep to save energy. We consider a long up-time and short up-time to be equal to 3 and 1 minutes, respectively. Up-times 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 up-times. 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 100 random up-time distributions, on which we apply the 4 described policies. From these runs, we measure each studied metric from the simulator. Each chosen metric will be exposed as an average and standard deviation (in parenthesis) of these 100 runs. To ease the comparison, each metric will have its overhead to the $baseline$ (as a percentage for the energy efficiency and as an amount of seconds for the accumulated up-time). Results for 1 minute and 3 minutes up-times are displayed in Table II and Table III, respectively.

A. Scenario 1: Short up-time duration, LoRa

The first studied scenario comprises: (i) LoRa as the chosen network technology and (ii) an up-time duration on the field equal to 1 minute. The simulation results are diplayed under *LoRa* column in Table II. As a reminder, we chose a fixed size

TABLE II: Simulation results with 1 minute uptime and 1MB file size

	LoRa				NbIoT			
	Energy(J)	$\%eOvhd$	AccUptime (sec)	$upOvhd$	Energy(J)	$\%eOvhd$	AccUptime (sec)	$upOvhd$
Baseline								
Sndr	619.73 (14.02)	+0.00 %	1440.00 (0.00)	+0.00	749.69 (58.05)	+0.00 %	1440.00 (0.00)	+0.00
Rcvrs	579.84 (1.29)	+0.00 %	1440.00 (0.00)	+0.00	591.07 (5.42)	+0.00 %	1440.00 (0.00)	+0.00
# Succ.	0.0 (0.0)				1.27 (1.12)			
Extended								
Sndr	1207.61 (159.87)	+94.86%	2564.53 (284.49)	+1124.53	946.39 (90.43)	+26.24 %	1788.32 (85.03)	+348.32
Rcvrs	627.35 (15.06)	+8.19 %	1523.52 (24.85)	+83.52	599.13 (6.66)	+1.36 %	1447.54 (3.94)	+7.54
# Succ.	7.70 (2.00)				7.16 (1.80)			
Hints								
Sndr	753.42 (49.26)	+21.57 %	1440.00 (0.00)	+0.00	1007.06 (120.37)	+34.33 %	1440.00 (0.00)	+0.00
Rcvrs	957.64 (126.08)	+65.15 %	2354.94 (304.74)	+914.94	702.59 (51.75)	+18.87 %	1666.51 (108.88)	+226.51
# Succ.	0.0 (0.0)				9.72 (3.45)			
Combination Hints and Extended								
Sndr	1397.98 (196.58)	+125.58 %	2904.50 (350.94)	+1464.50	1066.66 (123.75)	+42.28 %	1901.47 (116.41)	+461.47
Rcvrs	663.58 (37.67)	+14.44 %	1602.39 (83.98)	+162.39	623.30 (24.64)	+5.45 %	1492.53 (49.26)	+52.53
# Succ.	9.80 (2.20)				9.41 (2.36)			

of file to deliver: 1MB. Although, under these circumstances, 1 minute is not enough to have a successful delivery.

As expected, *Baseline* doesn't successfully deliver any file. 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. No communications between sender and receivers would have given the same energy consumption for every run and thus a null standard deviation.

Hints also doesn't successfully deliver any file, as it also doesn't change the up-time duration. $Hints$ adds new up-times to overlap the next one from the sender. 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 7.70 expected by the 12 receivers (i.e more than half). Although, it is expensive for the server from an energy consumption perspective, with $+94.86\%$ of energy overhead (when compared to *Baseline*). But the energy consumed by the receivers only have an average overhead of $+8.19\%$. This policy adds, in average, 1 min 23 sec (83.52 seconds) and 18 min 44 sec (1124.53 seconds) to the receivers and sender's accumulated up-times, respectively.

Combination Hints and Extended successfully delivers an average of 9.80 receivers. Although, it is expensive for the sender as it adds $+125.58\%$ of energy overhead when compared to *Baseline*. For the receivers, it is more expensive than *Extended* but less expensive than *Hints*, with an overhead of 14.44%. This policy adds, in average, 2 min 42 sec (162.39 seconds) and 24 min 24 sec (1464.50 seconds) to the receivers and sender's accumulated up-times, respectively.

Thus, in such a context (where bandwidth and up-time 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 overhead especially to the sender that sees its up-time doubled.

B. Scenario 2: Short up-time duration, NbIoT

The second studied scenario comprises: (i) NbIoT as the chosen network technology and (ii) an up-time duration on the field equal to 1 minute. The simulation results are displayed under *NbIoT* column in Table II. As a reminder, we chose a fixed size of file to deliver: 1MB. Thanks to the bandwidth of NbIoT, 1 minute is enough to have a successful distribution.

Even if the scenario allows successful deliveries very few successful deliveries are witnessed on *Baseline*, 1.27, due to the sparse and independent distribution of up-times leading to few overlaps between sender and receivers.

With *Extended*, more receivers get their distribution successfully (7.16, in average). An energy overhead of $+26.24\%$ and 1.36% is measured for sender and receivers, respectively. This policy adds, in average, 5 min 48 seconds and 7 seconds to the sender and receivers accumulated up-times, respectively.

Now that the up-time duration is enough to have a successful distribution, *Hints* outperforms *Extended*, with 9.72 successful distribution. Although, an overhead of $+34.33\%$ and $+18.87\%$ for sender and receiver, respectively. This policy adds, in average, 3 min 46 seconds to the accumulated up-time of the receivers. No overhead accumulated up-time is witnessed at the sender.

Combination Hints and Extended reaches 9.41 successful deliveries (close to *Hints*). Although, it is expensive for the sender with an energy overhead of $+42.28\%$ but cheap for the receivers, with 5.45% of energy overhead. This policy adds, in average, 7 min 41 sec and 52 sec to the accumulated up-time of the sender and receivers, respectively.

Thus, for this scenario and for all policies, we notice that all senders will always have a bigger overhead than the receivers. *Combination Hints and Extended* is the best trade-off when we want to maximize the number of successful deliveries and minimize the overhead of energy consumed by the receivers, at the expenses of the sender.

TABLE III: Simulation results with 3 minutes up-time and 1MB file size

	LoRa				**NbIoT**			
	Energy(J)	$\%eOvhd$	AccUptime (sec)	$upOvhd$	Energy(J)	$\%eOvhd$	AccUptime (sec)	$upOvhd$
Baseline								
Sndr	2026.23 (48.60)	+0.00 %	4320.00 (0.00)	+0.00	2117.65 (50.93)	+0.00 %	4320.00 (0.00)	+0.00
Rcvrs	1755.92 (5.74)	+0.00 %	4320.00 (0.00)	+0.00	1761.78 (4.84)	+0.00 %	4320.00 (0.00)	+0.00
# Succ.	1.91 (1.40)				9.90 (1.33)			
Extended								
Sndr	2692.46 (74.21)	+32.88 %	6038.78 (132.57)	+1718.78	2311.52 (48.45)	+9.15 %	4868.78 (45.67)	+548.78
Rcvrs	1772.35 (8.64)	+0.94 %	4364.27 (17.17)	+44.27	1760.92 (3.39)	-0.05 %	4323.71 (3.24)	+3.71
# Succ.	11.45 (0.70)				11.23 (0.87)			
Hints								
Sndr	2094.09 (16.87)	+3.35 %	4320.00 (0.00)	+0.00	2180.07 (27.51)	+2.95 %	4320.00 (0.00)	+0.00
Rcvrs	2169.50 (40.86)	+23.55 %	5344.86 (100.48)	+1024.86	1868.54 (35.11)	+6.06 %	4576.83 (85.28)	+256.83
# Succ.	12.00 (0.00)				11.92 (0.37)			
Combination Hints and Extended								
Sndr	2757.79 (58.99)	+36.10 %	6155.50 (105.30)	+1835.50	2351.14 (28.68)	+11.03 %	4906.10 (27.12)	+586.10
Rcvrs	1867.34 (50.95)	+6.35 %	4600.09 (128.28)	+280.09	1822.44 (31.08)	+3.44 %	4473.32 (77.84)	+153.32
# Succ.	11.94 (0.24)				11.88 (0.41)			

C. Scenario 3: Long up-time duration, LoRa

The third studied scenario comprises: (i) LoRa as the chosen network technology and (ii) an up-time duration on the field equal to 3 minutes. The simulation results are displayed under *LoRa* column in Table III. As a reminder, we chose a fixed size of file to deliver: 1MB. In such a context, 3 minutes is enough to have a successful distribution.

Even if the scenario allows successful deliveries very few success are witnessed on *Baseline*, 1.91, again due to the sparse and independent distribution of up-times leading to few overlaps between sender and receivers.

Extended is, in such a context, very good concerning successful deliveries, with an average of 11.45. An energy overhead of $+32.88\%$ and 0.94% are witnessed for sender and receivers, respectively. This policy adds, in average, 28 min 38 seconds and 44 seconds to the sender and receivers accumulated up-times, respectively.

Hints delivers all 12.00 distributions successfully, with and energy overhead of $+3.35\%$ and $+23.55\%$ for sender and receivers, respectively. This policy adds, in average, 17 min 04 to the accumulated up-time of the receivers. No added up-time is measured at the sender.

Combination Hints and Extended reaches almost all deliveries, with an average of 11.94. An energy overhead of $+36.10\%$ and $+6.35\%$ is measured for sender and receivers, respectively. This policy adds, in average, 30 min 35 sec and 4 min 40 sec to the accumulated up-time of the sender and receivers, respectively.

Thus, for this scenario, *Extended* is the best compromise between high deliveries and reduced energy consumption for the receivers. *Hints* is the best compromise to reduce 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 clients overheads, at the expenses of the sender.

D. Scenario 4: Long up-time duration, NbIoT

The fourth studied scenario comprises: (i) NbIoT as the chosen network technology and (ii) an up-time duration equal to 3 minutes. The simulation results are displayed under *NbIoT* column in Table III. As a reminder, we chose a fixed size of file to deliver: 1MB. In such a context, 3 minutes is much longer that expected to have a successful distribution.

Such a context allows *Baseline* to reach 9.90 successful deliveries. Thus, the impact of the sparse and independent distribution of up-times is not as strong as in previous scenarios.

Extended successfully delivers 11.23 for an energy overhead of $+9.15\%$ and -0.05% for sender and receivers, respectively. Notice that the overhead for the receiver is negative, meaning that we reduced in average the consumption of the receivers. This policy adds, in average, 9 min 8 sec and 3 sec. Notice that an added accumulated up-time doesn't necessarily mean added energy consumption. Here, *Extended* is slightly more efficient than *Baseline* and thus reduces the energy consumption of the receivers.

Hints reaches 11.92 successful deliveries, with an energy overhead of $+2.95\%$ and $+6.06\%$ for the sender and receivers. This policy adds, in average, 4 min 16 sec to the accumulated up-time of the receivers. No up-time overhead is measured at the sender side.

Combination Hints and Extended also reaches almost all deliveries with an average of 11.88. An energy overhead of $+11.03\%$ and $+3.44\%$ is measured for sender and receivers, respectively. This policy adds, in average, 9 min 46 sec and 2 min 32 sec to the accumulated up-time of the sender and receiver, respectively.

Thus, for this scenario and when a policy is activated, the successful deliveries all hovers around the 12, the number of receivers. Globally when a policy is activated, a low overhead is measured when compared to *Baseline* (between $+11.03\%$ and $+2.95\%$ for the sender and between 6.06% and -0.05% 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*. When the objective is to reduce the energy overhead of the sender the choice should either be disabling all policies (i.e choosing *Baseline*) or either choose *Hints* for a higher number of successful deliveries, at the expenses of the receivers.

VI. Lessons learned

This section presents the lessons learned from the simulations of multiple policies for the various scenarios covered. We also present the lessons learned from the comparison of simulation results under chosen network technologies: *LoRa* and *NbIoT*.

A. Choosing a policy, under several scenarios

From the 4 previously studied scenarios, we noticed and can extract common behaviours. *Extended* is expensive for the sender but with low overhead for the receivers, from an energy consumption point of view. *Hints* adds a non negligible amount of accumulated up-times to the receivers, which doesn't always translate into a bigger energy overhead for the receivers when 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 wouldn't be fair to consume the groups' energy to amortize 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 be very close to the number of deliveries achieved by *Hints* (when *Hints* outperforms all others) with an energy overhead for the receivers close to *Extended*, and with an energy overhead for the sender lower than the one measured while using *Hints*.

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 up-time duration), the average energy consumption of receivers is always lower for *NbIoT*. Same goes for the energy consumed by the sender, except for *"3minutes - Hints"* scenario, 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 up-time is not enough for *LoRa* to have any successful delivery (i.e "1 minute - Hint").

For slightly energy efficient receivers and successful deliveries, choosing *NbIoT* seems to be the right choice.

A dimension that was not explored in this paper is the duration of the up-time. We assume that the duration of an up-time, for an Observation Node, is a static choice (due to 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 up-time 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 choice of the network technology, (ii) the consumption of the nodes (idle and during usage of a given network technology), (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 include a model that would answer 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. This need 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 such a context. We study 3 loosely coupled policies that we compare to a baseline, where no policy is activated.

We simulate an already existing deployment with randomly picked up-times, allowing nodes to wake-up randomly every hour, for a very short duration (1 and 3 minutes). 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 up-time 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 up-time 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 up-time 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 *Hints* policy for the sender and a lower overhead than *Extended* policy for the receivers, for a very good number of successful delivery.

As a future work, we plan on extracting a model that will dynamical help a node to choose the policy in function of 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.

[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.

[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.

[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] 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.

[19] 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.

[20] 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.

[21] 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.

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

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

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

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

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

[31] 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.

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

[33] 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.