Journal of Parallel and Distributed Computing 197 (2025) 105013

Contents lists available at ScienceDirect

Journal of Parallel and Distributed Computing

journal homepage: www.elsevier.com/locate/jpdc

A large-scale study of the impact of node behavior on loosely coupled data dissemination: The case of the distributed Arctic observatory

Loïc Guégan *, Issam Raïs *, Otto Anshus

Department of Computer Science, UiT The Arctic University of Norway, Tromsø, Norway

ARTICLE INFO

Keywords:
CPS
Data dissemination
Energy efficiency
Scalability
Networks
Tundra
Monitoring

ABSTRACT

A Cyber-Physical System (CPS) deployed in remote and resource-constrained environments faces multiple challenges. It has, no or limited: network coverage, possibility of energy replenishment, physical access by humans.

Cyber-physical nodes deployed to observe and interact with the Arctic tundra face these challenges. They are subject to environmental factors such as avalanches, low temperatures, snow, ice, water and wild animals. Without energy supply infrastructures and humans available, nodes must achieve long operational lifetime from a single battery charge. They must be extremely energy-efficient. To reduce energy costs and increase their energy efficiency, cyber-physical nodes sleep most of the time, and avoid to communicate when they are unreachable.

But, a CPS needs to disseminate data between the nodes for multiple purposes including data reporting to a back-end service, resilient operations, safe-keeping of observational data, and propagating nodes updates. Loosely-coupled data dissemination policies offer this possibility [1]. Although, investigations should be made on their applicability to large-scale CPS.

In this paper, we evaluate and discuss the efficiency in energy, time and number of successful delivery of four data dissemination policies proposed in [1]. This evaluation is based on flow-level simulations. We study small and large-scale CPS, and evaluate the effects of the number of nodes and the size of the disseminated data on the nodes energy consumption and the dissemination's delivery success. To mitigate negative effects raised on large-scale CPS and large disseminated data sizes, different strategies are proposed and evaluated. We show that energy saving strategies do not always imply energy efficiency, and better data dissemination often comes at a cost. This last result highlights the importance of simulation prior to real CPS deployments in constrained environments.

1. Introduction

Cyber-Physical Systems (CPS), Wireless Sensors Network (WSN) and the Internet Of Things (IoT) are applied in various domains [2] such as Environmental [3], Flora and Fauna monitoring [4], Habitat monitoring [5], Health Care [6], Military [7], Industry and Urban management [8]. To communicate, they rely on various wireless technologies (such as LoRa, Wi-Fi, Nb-IoT etc.) to ensure connectivity among the nodes and a potential back-haul network. Wireless communications offer great flexibility in terms of deployment since CPS are meant to reach a degree of autonomy, directing the research towards the optimization of communications and node operations.

The Arctic tundra is a particularly harsh environment and pushes existing monitoring solutions to their limits. Monitoring the Arctic tundra requires to deploy energy efficient nodes, expected to operate for long time periods. The Arctic tundra offers little to no coverage by cellular networks coupled to extreme weather conditions. The success of this monitoring is driven by the energy efficiency of nodes and the availability of data when and where needed.

The Distributed Arctic Observatory (DAO) at The Arctic University of Norway (UiT), is the context of this paper. The DAO project relies on Computer Science research to address the challenges raised by the monitoring of the Arctic tundra. Sensing nodes, called Observation Nodes (ON), are deployed in-situ. ON are expected to operate for months and even years. To ensure the availability of data collected by these nodes, an efficient data dissemination mechanism is required.

In [1], four loosely coupled data dissemination policies are proposed in the context of the DAO project. Simulations are carried out and the results reveal a great data dissemination capability on each of these policies. Reasonable amount of energy is consumed by the nodes for rel-

* Corresponding authors.
E-mail addresses: loic.guegan@uit.no (L. Guégan), issam.rais@uit.no (I. Raïs).

https://doi.org/10.1016/j.jpdc.2024.105013
Received 10 January 2024; Received in revised form 25 October 2024; Accepted 16 November 2024

Available online 22 November 2024

atively small networks (up to 12 nodes). With the aim of extending this work, measuring the applicability of such policies on denser networks is important and can enable their usage in other contexts. Similarly, the disseminated data size can vary significantly depending on the context, it is also crucial to quantify its impact on the system. In addition, several energy saving and data dissemination strategies can be used to improve the performance of these policies in terms of energy consumption and data dissemination.

In this paper, we propose a study of these four data dissemination policies on large-scale deployments (up to 100 of nodes) using flow-level simulations. We also investigate the impact of various disseminated data sizes, ranging from 1 KB up to 1 GB. Several strategies that aim at improving each policy in terms of energy and data dissemination efficiency are also proposed.

The contributions of this paper are:

- An analysis of the applicability of loosely coupled data dissemination policies for large-scale deployments and various disseminated data sizes
- The evaluation of four energy saving and data dissemination optimization strategies
- A detailed numerical comparison between new proposed strategies and our previous work [1]
- A discussion on the data dissemination when applied to CPS in harsh environment

The paper is organized as follows. Section 2 details the challenges raised by the DAO project. Section 3 presents the related work. Section 4 presents the strategies for energy saving and data dissemination improvement along with the experimental setup used for their evaluation. The Section 5 details the simulation results for the scalability experiments that study the number of nodes. Section 6 presents the results for the data size scalability study. The Section 7 presents the evaluation of the strategies and a comparison to our previous results [1]. A discussion on the simulation results is proposed in Section 8. Finally, Section 9 concludes the work.

2. 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.

2.1. The Arctic tundra, a complicated eco-system

As depicted by Fig. 1, the Arctic tundra is a large, remote, hard to reach, and potentially dangerous area. It is divided into three regions (High, Low and Sub Arctic) that are characterized by their unique ecosystems. By observing the Arctic tundra flora, fauna and environmental parameters, changes can be identified and tracked. Presently, less than 1% of the Arctic tundra is monitored. However, it is one of the most sensitive eco-system to climate change [9]. Therefore, to detect accurately climate changes, larger observations of the Arctic tundra are needed.

The Climate-Ecological Observatory for Arctic tundra (COAT[1]) initiative is responsible for observing the Norwegian Arctic tundra, detecting and explain climate related changes to advise the public and the authorities.

To do so, the state of the Arctic tundra is determined based on measurements of the flora, fauna, weather, and the atmosphere. From these measurements, several layers of data sets are generated. For example,

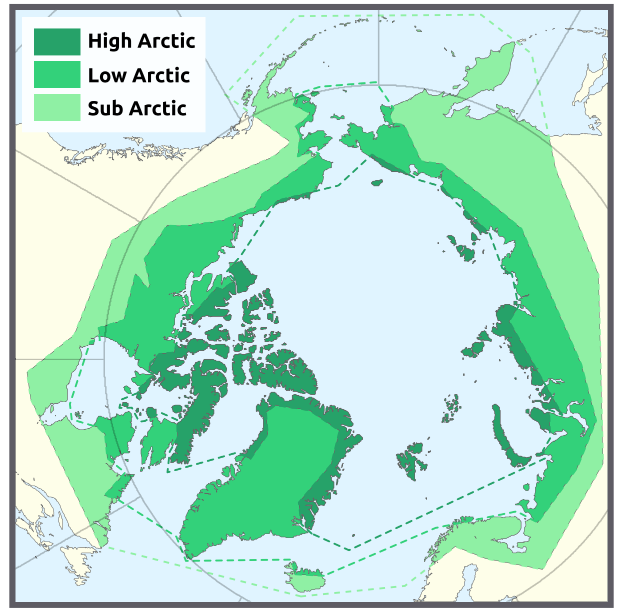


Fig. 1. North pole circumpolar area representing the Arctic tundra. It is divided into three regions: High, Low and Sub Arctic.

species of captured animals can be detected from a first data set of images, creating a new data set. This new data set is analyzed to extract significant information such as the number of foxes and eagles detected at the different monitored sites. These insights are used as input to climate models. Finally, based on the results history of climate models, human understanding and decision making take place [9].

A ground-based observation system can monitor large areas, do high resolution measurements at any time and promptly react to local events above and below ground. Data might be reported back at any time, regularly, or on-demand. To enable edge computing, significant processing and storage resources can be added to the nodes. The DAO project focuses on such ground-based observation approaches.

2.2. Towards a distributed Arctic observatory (DAO)

There are major obstacles to consider when building an observation system for the Arctic tundra. In this environment, energy is a scarce resource, especially in winter where the sun does not rise. Deep snow makes wind-based energy replenishment difficult to achieve. The lack of roads and associated infrastructures makes it impossible to visit deployment sites more than a very limited amount to fetch data, supply energy, do repairs and updates. Availability of a back-haul network to perform automated reporting of data can be limited or non-existent. Thus, it is challenging to have sufficient energy supply for nodes with advanced functionalities while still getting long operational lifetimes.

A distributed Arctic observatory system must manage carefully two fundamental resources: energy and wireless data networks. Nodes are working on a limited energy budget provided by batteries. As it is a complicated scenario (harsh weather, short periods of sun exposition during winter), swapping batteries by humans and regular energy harvesting are not plausible solutions. Nodes can also fail after deployment because of harsh weather conditions (snow, ice, low temperatures) that could damage the hardware, or simply prevent nodes from communicating. Consequently, nodes must implement a set of functionalities, 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 node can have multiple local networks enabling communication with neighbors. Using a multi-hop approach, data can be reported through multiple units and finally to one that have access

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

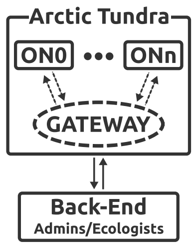


Fig. 2. Overview of the system imposed by the Arctic tundra characteristics. The back-end hosts a set of services [11]. Its connectivity to Observation Nodes (ON) deployed at the Arctic tundra is sparse and unexpected. The wireless gateway in the topology is only used for 1:1 communications between Observation Nodes forming a star topology.

to the back-haul network or is located on sites reachable by humans or drones [10]. However, using the radio is energy-expensive. One approach to reduce transmissions is to reduce the number of bits exchanged between nodes. But such leverage applies only if the data can still provide similar analytic precision [11].

In this paper we focus on delivering data from one node to neighbors in the context of nodes deployed and isolated on the Arctic tundra (i.e. not accessible by a back-haul network as a common case). This work does not consider multi-hopping nor modifying the data, as shown in Fig. 2.

2.3. Data dissemination, a crucial need

Presently, COAT ecologists use several approaches and instruments to observe the Arctic tundra [12,13]. Typically, tens to a few hundreds of small dedicated instruments are deployed according to the expected location of interesting events. These instruments are deployed for multiple purposes, including images capture of animals. For hard to reach installations, it takes up to 6-12 months before humans visit the site and fetch the data. These deployments are usually done in small clusters of 10 to 15 instruments. Each instrument is separated from hundreds of meters to kilometers. In such deployment context, disseminating data from nodes to their neighbors is crucial in many cases.

a) Backing up important results Deployed nodes can do local computation on local observations. Due to the high probability that deployed units crash (e.g. through flooding, hardware failure) it can be important to duplicate the results from these computations. Important results must be disseminated to as many neighbors as possible, to keep the data safe and reduce the chance of loosing results. For example, in [11] we ease the dissemination of the data by reducing the size of captured pictures to reduce the number of transmitted bytes to a remote CNN deep learning application. For some deployments, both the full sized as well as the reduced sized photos should be disseminated inside a neighborhood for safe keeping purposes, until the data can be reported.

b) Disseminate updates Few to no nodes are expected to have connection with a back-haul network as it would be sporadic and unreliable. Since physically accessing to the Arctic tundra is complicated and expensive, updates (e.g. configuration files, executable, packets or other newer content for a receiver) must be delivered by the back-end. Updates can come from users of the system such as ecologists or administrators, as shown in Fig. 2. When a node finally gets an update, we can expect it to disseminate it to its neighbors. 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 all cases, the size of the disseminated data is not expected to be very high, due the wireless technologies limitations, energy and the availability of computing resources. The constraints related to batteries and energy consumption are tackled in our previous work [4,10,11].

3. Related work

3.1. Wireless technologies

Choosing the right wireless technology depends on the use case. In our work, the use case is the Distributed Arctic Observatory, and we are focusing on three main criteria: 1) the energy consumption 2) the communication range 3) the bandwidth. The Arctic tundra is a particularly hostile environment. Nodes are expected to survive for almost a year without humans intervention. Having an energy efficient wireless technology allows to reduce nodes deployment campaigns and having longer sensing periods. Performing long range wireless communications is crucial and allows to cover larger areas for the scientific measurements. Finally, using a wireless technology with sufficient bandwidth allows to generate finer grained data and improves the quality of measurements. Consequently, we selected wireless technologies that provide good compromise between these three criteria. Such technologies are part of the Low-Power Wide Area Networks (LPWAN) category.

The DASH7 Alliance (D7A) is an open source wireless solution for WSN [14,15]. Working at 433 MHz, D7A allows to achieve long range communications up to 2 km [16]. It can be used with the D7A protocol allowing for star-based network topology and device-to-device communications. Hence, it offers great flexibility on the network architecture. Moreover, D7A has good energy performance compared to others LPWAN technologies [17] while providing up to 200kbps of applicative bandwidth [16]. This makes it suitable for use in the Arctic tundra.

LoRa is another well known wireless LPWAN technology. In networks that uses LoRa, gateways are deployed to offer communication relay among the nodes and potentially a back-haul network. Depending on the physical layer configuration, LoRa provides long range wireless communications that can reach up to 5-6 km distances [18]. LoRa can deliver up to 50kbps [19] while being energy efficient. This makes LoRa a good candidate for our use-case.

Next, Narrow-Band IoT (Nb-IoT) is an interesting technology to consider for our use case. It is a derivation of the Long-Term Evolution (LTE) that use the existing LTE infrastructure as a communication gateway [19]. Since the Arctic tundra is scarcely covered by cellular tower as mention in [4], Nb-IoT is suitable to our context. Similarly to LoRa, Nb-IoT uses a star topology with a gateway that relay the data. In addition, Nb-IoT has been designed for low-power and long range wireless communications. Despite being more energy consuming than LoRa, it provides more bandwidth (up to 200kbps [20]).

Several other wireless technologies are available in the literature [21,22]. To the best of our knowledge, none of them are relevant enough for our use-case. Consequently, we choose LoRa and Nb-IoT to conduct our study as they are use in DAO prototypes and deployments [10]. These technologies offer a good compromise among the three criteria exposed for the DAO project.

3.2. Data dissemination

The literature provides multiple data dissemination policies that aim at reducing the energy consumed by the nodes. In this section, we turn our focus on three of them.

Achieving energy frugal data dissemination for mobile nodes in wireless sensor networks has been proposed by basing the dissemination on a grid structure [23]. However, this approach requires coordination to maintain the grid structure and the coupling in time between nodes. This implies having to do more communications, leading to higher energy usage, and a faster drain of the battery. In our case, maintaining such an overlay network would lead to too many communications and drain nodes battery life. Thus, such approach could not be used in our context.

Solutions that deal with reducing redundant transmissions to be energy efficient, like in [24], usually comes with the hypothesis that sensors are part of a virtual grid and maintain a node list. In the case of a deployment in a scarce-resource environment such as the Arctic tundra, it will not be beneficial to have such a representation as the nodes must implement shutdown policies and be OFF most of the time. Thus, nodes are mostly unreachable leading to an obsolete virtual grid and node list.

Works such as [25] are providing policies to handle nodes that fail on the field. These types of contributions are effective for a limited number of failures. However, such failures are common in our use-case and the work do not account for scenarios where all nodes are failing in a deployment. For the Arctic tundra, we are in the opposite case. Most of the time, we expect a significant part of nodes to be unavailable. This is due to independent shutdown policies embedded on each node, trying to last as long as possible. Node suddenly shutting down unexpectedly is equivalent to a node failing, for a neighboring node.

A resource limited environment such as the Arctic tundra imposes conditions where it is complicated to evaluate if a solution to disseminate data has a positive impact on the energy consumption. Contributions covered in this section use hypothesis that do not match with the requirements of our use case. Quantifying the costs (in time and energy) of loosely coupled policies from calibrated values extracted from the literature under plausible hypothesis such as this work is essential. It allows to establish a relation between researchers works and reality, and having answers to build upon.

In a resource limited environment such as the Arctic tundra, existing energy efficient data dissemination cannot be used. Very few assumptions can be made about neighboring nodes and most of the hypothesis used in existing works do not match with this reality. Structure-less based schemes [26] are currently the best approach to disseminate the data in the use-case of the DAO, as propose in [1]. They provide mechanisms to disseminate data without having to maintain a data structure that would require more communications, synchronicity, and higher node availability. This work extends [1] with the aim of improving both energy and data dissemination efficiency.

4. Experimental setup

This section presents a brief summary of the different data dissemination policies used in the contribution. Then, strategies that aim to improve CPS performance in terms of energy consumption and data delivery success are presented. Finally, the simulation setup along with the metrics used for the analysis are detailed.

4.1. Dissemination policies

Our previous paper [1], presents four different loosely-coupled data dissemination policies suitable for our use case.

Baseline - This policy corresponds to a sender node that wakes up randomly every hour and tries to send its data to receivers. Since we are in a resource limited environment, sender and receivers nodes are OFF most of the time. They wake up randomly for a duration called *uptime*. During this uptime, overlap between the sender and one or more receiver can occur. In that case, the sender tries to send its data to the receivers. Having such policy allows to have a baseline for comparison with more complex policies. Fig. 3(a) depicts the *Baseline* policy where a sender start to send its data around time t_x to a receiver.

Extended - In some baseline scenarios, the uptime duration is not long enough to allow the data to be sent entirely. The communication is then interrupted when the node shuts down. This reduces the data delivery success. *Extended* policy introduces extra time at the end of every uptime to allow the data to be sent properly. It aims at a higher data delivery success. Fig. 3(b) details this policy where a sender and a receiver get their uptime extended.

Hints - This policy is an enhancement compare to the *Baseline* policy. It introduces an additional timestamp that is added to the data. This timestamp, notifies the receiver about the next time at which the sender is expected to wake up. This hint can be forwarded by the receivers during overlap with other receivers uptime to maximize the number of overlap between sender and receivers. Note that the sender occasionally sends separate hint to receivers in addition to normal data communications. From these overlaps, sender can transmit the data to receivers, increasing the delivery success. Fig. 3(c) explains the *Hints* policy where a hint is delivered along with the data to a receiver that forwards it.

Hints and Extended - This policy is a combination of the *Extended* and *Hints* policies. Both combined policies behave exactly the same as they were introduced. The objective is to further improve the delivery success by merging the impact of both policies. The *Hints and Extended* policy allows to measure the benefits of combining policies in terms of various metrics detailed later in the paper. Fig. 3(d) depicts a given scenario where *Hints* and *Extended* policies are used together.

The lessons learned from our previous work [1], show that, these policies can be greatly improved regarding data delivery and energy performance. The following sub-section introduces several strategies with the ambition to improve the dissemination of the data on these two axes.

4.2. Strategies

Policies presented in [1], provide a solution to disseminate data in loosely coupled networks and strive to mitigate the energy consumed. To further improve the efficiency of these data dissemination policies, this section introduces three distinct strategies. The first two aim at optimizing the existing dissemination policies by exploiting simple energy saving ideas. The third one proposes a mechanism for communications aggregations to leverage the data delivery success and reduce the energy consumption.

Strategy 1: Shutdown on receive In the current version of our data dissemination policies presented in Section 4, receivers wake up with the hope of receiving data from the sender. If a communication occurs and the data are successfully received, the receiver keeps on being awake for the complete duration of its uptime. It allows to communicate with other potential receivers, and being part of the hint forwarding mechanism to improve the efficiency of the hints dissemination.

However, this approach has a cost for the receiver in terms of energy consumption. Since the receiver already owns the data, the *Shutdown on receive* strategy turns off the receiver as soon as the data are received. Fig. 4(a) depicts a scenario where this strategy is applied. The energy saved depends on the uptime duration left after the data distribution ends (green area). This duration is impacted by: 1) The time at which the communication starts 2) The communication duration 3) The uptime duration. Consequently, having short communications on large uptime scenarios can lead to significant energy saving if the nodes are able to shut down.

But, since the receiver is part of the hint forwarding mechanism during the remaining time period (after the data delivery), studying the impact of such strategy on the energy efficiency and the data delivery success is required.

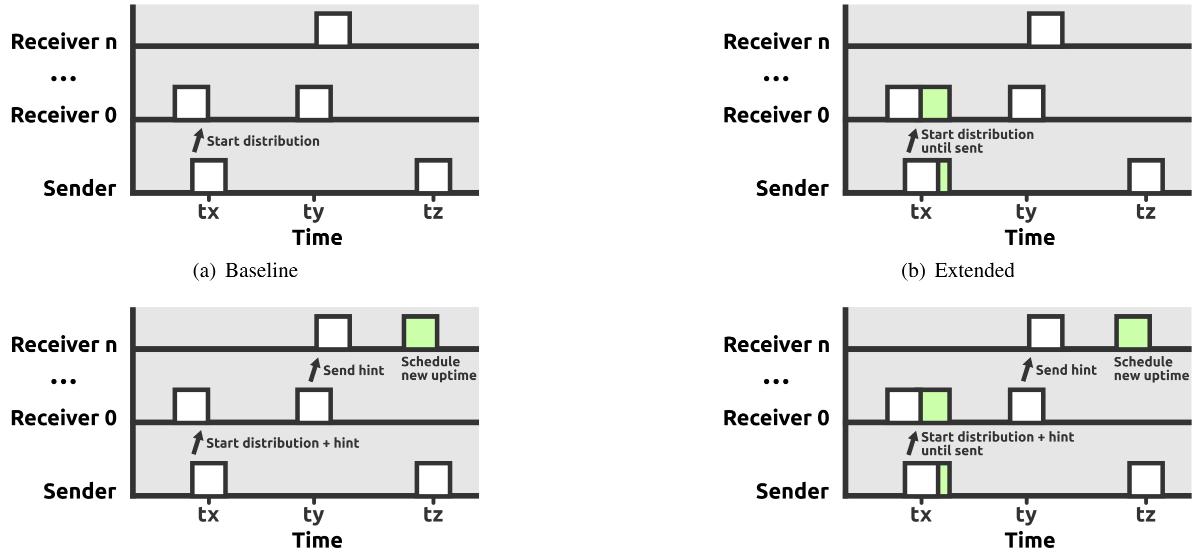
Strategy 2: Unschedule on receive

Among the four studied policies, two of them use the hint forwarding mechanism. This mechanism increases the likelihood of uptime overlaps between sender and receivers. It dynamically schedules a new wake-up time on the receivers to get an uptime overlap with the sender on the next uptime.

In many situations, nodes could receive a hint from another node just before receiving the data (on the same uptime slot). As receivers propagate hints (see Fig. 3(c) and 3(d)), this new uptime slot is also used by the receivers to propagate hints. Hence, increasing the amount of overlap between sender and receivers uptime.

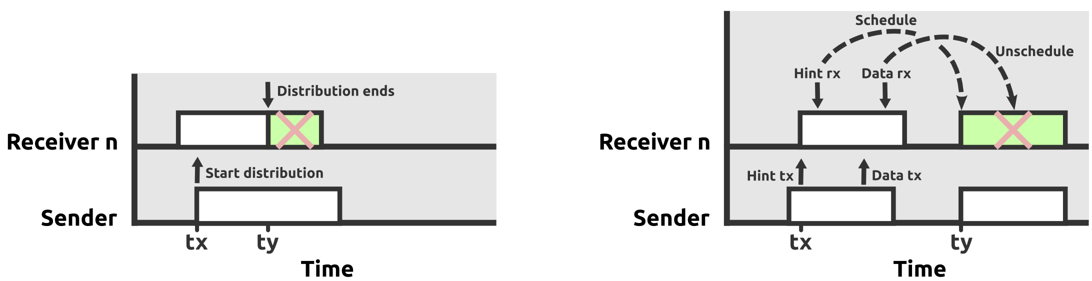
One approach to save energy in these situations is to unschedule the new uptime on nodes that already received the data. This strategy assumes that, the sender disseminates the same data on a given day. Receivers that own the data would unschedule the new uptime that occurs the same day.

This strategy, called *Unschedule on receive*, is presented on Fig. 4(b). Removing these additional uptimes impacts the hint forwarding mech-



(a) Baseline (b) Extended (c) Hints (d) Combination Hints and Extended

Fig. 3. 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.



(a) Shutdown on receive (b) Unschedule on receive

Fig. 4. Sequence diagram of the two first energy saving strategies. Green time slots represent additional wake up times due to the receive of hints.

anism, and could lead to lower overall energy and data delivery performance. Hence, a study of the *Unschedule on receive* strategy is required.

Strategy 3: Far hint The *Hints* and *Hints and Extended* policies, from [1], rely on the hint forwarding mechanism to increase the likelihood of having uptime overlaps between sender and receivers. A hint contains a timestamp that notifies the receivers about the next sender uptime, and it is forwarded by the receivers. However, the duration between the first transmission of a given hint by the sender, and the timestamp contained in this hint may not be large enough to ensure proper hint dissemination. Small hint duration reduces its chances of propagation within the network. On the contrary, large hint duration leads to better hint propagation.

The *Far Hint* strategy proposes to extend this duration to highlight its impact on the energy and data delivery performance. Increasing this duration could lead to more communications among the nodes and increase their energy consumption. As a result, even if an increase of data delivery success is noticeable, simulations should be conducted to quantify the impact on the nodes energy consumption.

4.3. Simulation setup

The simulations conducted in this work have three purposes. First, evaluating the scalability of *Baseline*, *Extended*, *Hints* and *Hints and Extended* in terms of data dissemination efficiency and nodes energy consumption while varying the number of nodes in the system. Second, providing a similar evaluation of the policies scalability when varying the amount of data disseminated. Third, quantifying the impact of the following strategies: *Shutdown on receive*, *Unschedule on receive* and *Far Hint* on the data dissemination efficiency and nodes energy consumption.

Our experiments are based on the network simulator developed in [1]. This simulator uses flow-level network models provided by the SimGrid simulation framework. Flow-level models allow to achieve computationally efficient simulations by making use of abstract network models. In addition, SimGrid provides strongly validated models that ensure accurate predictions. Currently, this simulator implements the four data dissemination policies discussed in this paper. This simulator is extended to implement the three strategies presented in this work. With this approach, strategies can be applied directly on top of the dissemination policies, allowing to preserve the exact same simulation environment and initial conditions. All the experiments presented in this paper can be reproduced, and are available online [27].

The simulation parameters are detailed in Table 1. They are extracted from the literature and based on our previous deployments [4]. Compared to our previous study, the hint duration has increased from one to three hours [1]. Receivers are notified about the next sender uptime located three hours away from the current one. We choose a hint duration of three hours to represent several wake up times, where nodes are going to sense their environment and potentially communicate. We choose a greater duration than [1], to quantify its impact on the dissemination of the data and the nodes energy consumption.

For the strategies evaluation, each simulated scenario uses 1 sender and 12 receivers. To reproduce the use case presented in Fig. 2, each node is considered to be reachable by each other, forming a clique. The

Table 1
Simulation parameters.

Bandwidth (Latency)	LoRa Nb-IoT	50kbps (0 s) [19,20] 200kbps (0 s) [20]
Power states	P_{idle} LoRa Nb-IoT	0.4 W [28] +0.16 W (+32 mA at 5 V) [29] +0.65 W (+130 mA at 5 V) [29]
Uptime	Short Long	1 min/hour 3 min/hour
Data size	Fixed Varied	1 MB 1 KB to 1 GB
Nodes	Fixed Varied	1 Sender, 12 Receivers 1 Sender, 12 to 100 Receivers
Far hint duration	3 hours	
Simulated time duration	24 hours	

sender owns 1 MB of data that should be transmitted to the 12 receivers. This amount of data is fixed for the strategies evaluation scenarios. To communicate, each node wakes up once every hour for a duration called uptime. This uptime lasts $60s$ or $180s$ depending on the simulation inputs. The total simulated time for each run is 24 hours. Each run is performed 200 times with a different randomly generated nodes schedule. Hence, all the studied metrics will be averaged over these 200 runs.

The scalability study for the number of nodes uses similar parameters. Except that, the number of nodes are varied from 12 to 100 and 20 runs are performed for each scenario. For this evaluation, all the studied metrics are averaged over these 20 runs.

Parameters from Table 1 are also used to study the impact of the disseminated data size. The amount of data disseminated by the sender varies from 1 KB up to 1 GB. For each data size, all the studied metrics are averaged over 200 runs.

4.4. 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 reasons, 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 policy p and $Baseline$, respectively. An $\%eOvhd(p)$ of 0% for a given policy p indicates that no additional energy is consumed compared to the *Baseline* one, thus the "-100".

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

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

The accumulated uptime $AccUptime_p$ represents the sum of all nodes 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 (noted J/S).

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

With $energyConsumed_p$ representing the average energy consumption of the sender node or the receiver nodes, $\#Succ_p$ that represents the number of data delivery success for the policy p. The lower $eff(p)$ is, the more energy efficient the policy p is on the given sender or receiver nodes. Energy consumption for sender and receivers are reported separately to be able to identify imbalances, for a given or across different scenarios.

5. Scalability: number of nodes

The scalability results for the four data dissemination policies introduced in [1] are presented on Fig. 5. These results show the impact on the energy consumption and the delivery success of the senders and receivers while varying the number of nodes. The colored backgrounds show the standard deviation over the 20 runs for the given number of node.

5.1. Scenario using $60s$ uptime

The results for $60s$ uptime are presented on Figs. 5(a) 5(b) and 5(c). These results show a strong correlation between the energy consumption and delivery success of the senders visible on Figs. 5(a) and 5(c). This is due to the sender being the only node that propagate the data. Therefore, the sender drives the delivery success while impacting its own energy consumption. On Fig. 5(c), *Baseline* and *Hints* with LoRa are overlapping with a constant delivery success of 0. Under $60s$ uptime, LoRa does not provide enough bandwidth to disseminate 1 MB of data.

The results from Fig. 5(c) for the LoRa wireless technology show that, the *Baseline* and *Hints* policies are not able to disseminate the data. The *Hints* policy induces an additional cost on the receivers energy consumption due to the hint forwarding mechanism. But, because of the use of the *Extended* policy, the *Extended* and *Hints and Extended* policies allow to disseminate data and provide better delivery success in this scenario.

Fig. 5(a) shows that using the LoRa, the sender exhibits a logarithmic grow of the energy consumption on both *Hints* and *Hints and Extended* policies. With the *Extended*, the energy consumption of the sender quickly reaches high values compared to Nb-IoT. In short uptime scenarios, using a wireless technology with a higher bandwidth and a slightly higher energy consumption, could potentially save energy on the sender node.

Despite using Nb-IoT, the delivery success of the *Hints* policy has a bottleneck around 25 nodes. Although the *Hints* policy improves the likelihood of uptime overlap between senders and receivers, the fact that nodes cannot extend their uptime adds limitations to the dissemination performance. This bottleneck is also visible on the sender energy consumption for Nb-IoT. On the receiver side, the energy consumption is affected by the hint forwarding mechanism. Hence, Nb-IoT increases significantly the energy consumption of the receivers with the *Hints* and *Hints and Extended* policies. The lower energy consumption of LoRa allows to mitigate this effect on the receivers.

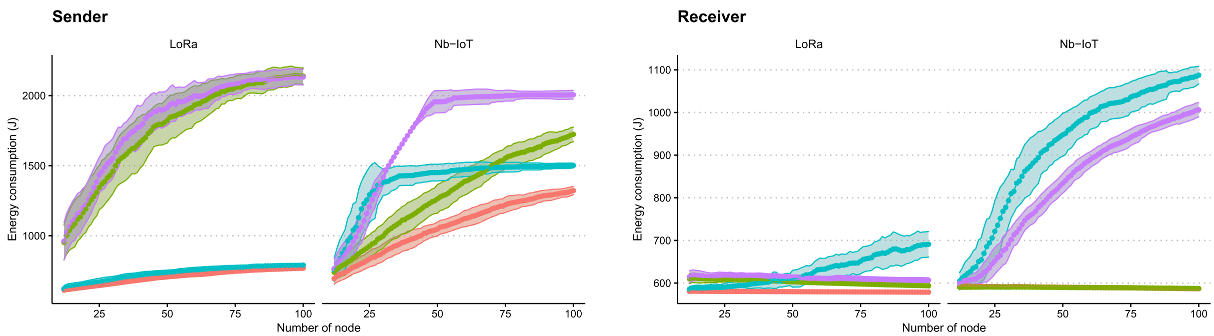
Using a large number of nodes, the *Extended* policy with Nb-IoT offers the best trade off between energy consumption and delivery success. It has a limited impact on the sender and receiver energy consumption. It is able to provide up to 39 delivery successes with 100 nodes.

5.2. Scenario using $180s$ uptime

The $180s$ uptime results are depicted on Figs. 5(d), 5(e) and 5(f). Similarly to previous results, a strong correlation between the energy consumption of the nodes and the delivery success is visible.

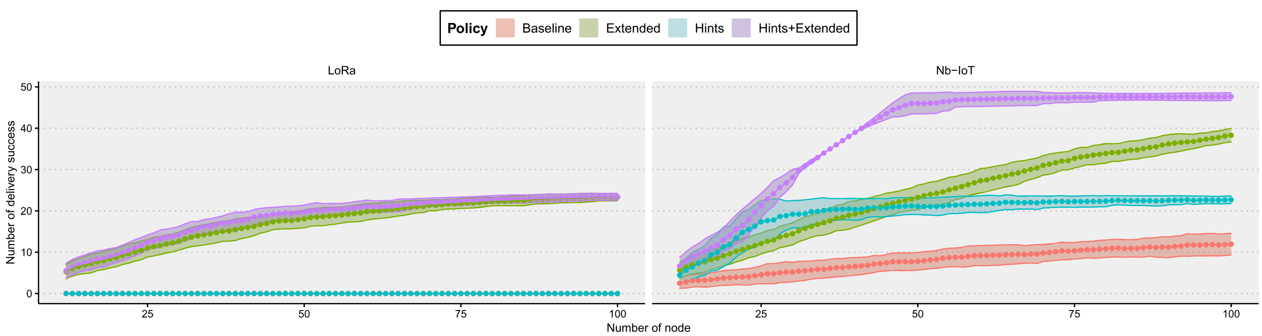
On Fig. 5(f) the results for the LoRa wireless technology show a performance bottleneck around 50 nodes with the *Hints and Extended*. In this case, the delivery success does not increase any further. This bottleneck shows that LoRa does not provide enough bandwidth to disseminate the data to all receivers. To further increase the delivery success with LoRa and a data size of 1 MB, increasing the uptime duration is the only leverage. This is also visible on the sender energy consumption. But, the energy consumption of the receivers keep increasing with a logarithmic trend due to the *Hints* policy. Hence, for this scenario with LoRa, the only policy to offer a good compromise between delivery success and energy consumption is the *Extended*.

Regarding Nb-IoT, the results on the delivery success show that the *Baseline* policy scales better than LoRa. The bandwidth of Nb-IoT can

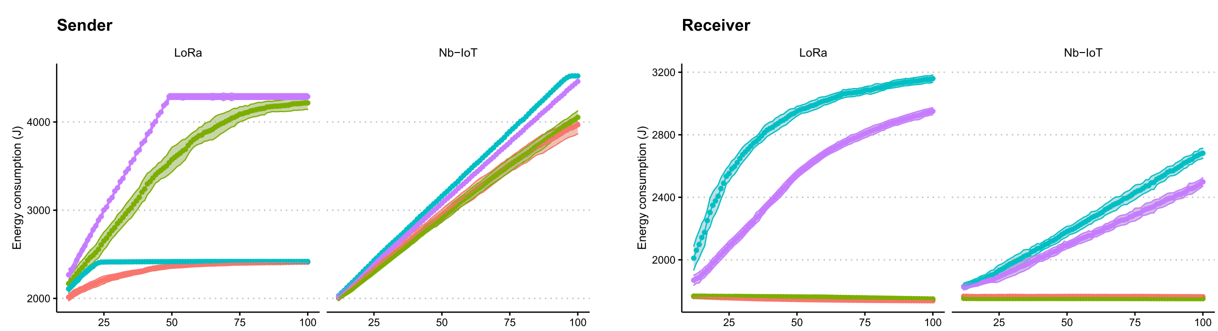


(a) Sender energy consumption in Joules for 60s uptime

(b) Receiver energy consumption in Joules for 60s uptime

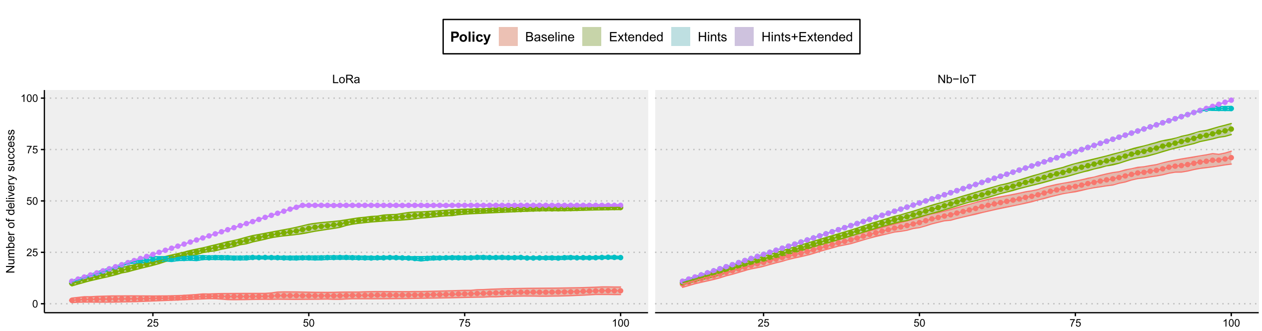


(c) Delivery success for 60s uptime



(d) Sender energy consumption in Joules for 180s uptime

(e) Receiver energy consumption in Joules for 180s uptime



(f) Delivery success for 180s uptime

Fig. 5. Scalability results obtained by varying the number of nodes from 12 to 100. Each run of n nodes is composed of 1 sender and $n-1$ receivers. Standard deviation of each curve over 20 runs is represented with their respective background color.

be leverage to improve the data dissemination. Moreover, the average energy consumption of the senders and receivers follows a linear increase up to 100 nodes. In this case, Nb-IoT provides enough bandwidth to disseminate the data to most receivers. Despite being more energy demanding, Nb-IoT combined to our four policies, offers great performance that allows to disseminate data in dense scenarios while mitigating the impact on the energy consumption to LoRa.

The *Hints* and *Hints and Extended* policies allow to disseminate the data to most of the receivers despite introducing a slightly higher energy consumption on the sender compared to the *Baseline* and *Extended* policies. But, their impact on the receiver energy consumption is significant. The *Baseline* policy performs well on dense scenarios. Still, the *Extended* policy achieves better delivery success with an energy consumption similar to *Baseline* and a narrow standard deviation.

5.3. Summary

These scalability results reveal interesting phenomenons on dense network scenarios. Having a too few wireless bandwidth leads to a low delivery success and a higher energy consumption since more uptimes are required for the data to reach the receivers. In dense networks, policies that use additional communications to improve the delivery success (such as the *Hints* policy) increase significantly the energy consumption of the receivers. This is particularly true on the receivers that are involved in the hints dissemination. One way to mitigate this effect is to use different wireless technology for sending the data and the hints. Finally, on large-scale deployments, using rather simple policies such as *Extended* allows to save a lot of energy compare to more complex one such as *Hints* and *Hints and Extended* and offer decent data dissemination performance. On the other hand, the *Hints* policy is useful on lower-scale deployment (below 25 nodes in the $180s$ uptime scenario) and allows to achieve better delivery success while consuming less energy than the *Extended* policy. The *Hints and Extended* policy allows to maximize the delivery success when the energy consumption is not a critical resource.

Limiting the impact of a policy on the energy consumption is crucial. For energy consumption reasons, policies that work perfectly on relatively small-scale deployments may not be used on large-scale one (as seen with the *Hints* policy). Several factors such as the size of the disseminated data can have a major impact on the policies efficiency and must be studied. It is also important to provide strategies that strive to limit the energy consumption of small-scale deployments. This could translate into bigger energy saving on denser networks. The remaining of this work evaluates the impact of the disseminated data size and strategies that could help in this direction.

6. Scalability: data size

The scalability results for the four data dissemination policies introduced in [1] are presented on Fig. 6. These results show the impact on the energy consumption and the delivery success of the sender and receivers while varying the size of the data disseminated by the sender. The colored backgrounds show the standard deviation over 200 runs for a given data size. For clarity, each policy gets a different line shape format to highlight overlaps between results.

6.1. Scenario using 60 s uptime

The results for $60s$ uptime are presented on Figs. 6(a) 6(b) and 6(c). It shows a negative relationship between the energy consumption of the nodes and the delivery success. As the data size increases, communications duration gets longer, leading to fewer amount of communications per day. Data gets disseminated to fewer receivers.

The energy consumed with the *Extended* and *Hints and Extended* policies gets higher for larger data size. In these cases, sender and receivers communicate for a longer duration due to their extended uptime duration. It increases significantly the energy consumption of the nodes. This increase is particularly important on the sender node since it is involved in most communications. Consequently, for large data size, it is critical to ensure that the nodes energy budget is met when allowing nodes to extend their uptime duration.

The energy consumed with the *Baseline* and *Hints* policies increases as data size increases. It is not visible due to the difference in scale between policies results. With these two policies, communications duration is bounded by the uptime duration combined to the limited bandwidth. For these reasons and for larger data sizes, the energy consumption does not increase further compared to scenarios with extended uptime duration (*Extended* and *Hints and Extended*) and similar bandwidth.

On the wireless technologies perspective, LoRa has less bandwidth compared to Nb-IoT. A bottleneck is reached around 500 KB where the bandwidth of LoRa becomes a limitation. Under $60s$ uptime, the data cannot reach the receivers. This similar bottleneck is introduced in Section 5.1. Nb-IoT offers more bandwidth which shifts this bottleneck to 2 MB (Fig. 6(c)). The Nb-IoT bandwidth reduces the communications duration and provides higher delivery success. For the energy consumption, LoRa and Nb-IoT consume roughly the same for large data size using the *Extended* and *Hints and Extended* policies. However, this is not the case for the *Baseline* and *Hints*. Even though it is not visible on Figs. 6(a) and 6(b), Nb-IoT consumes approximately 30% more energy with these policies with large data size. For a fine grain exploration of the data, the entire data set and the analysis scripts are available online [27].

6.2. Scenario using 180 s uptime

Results for $180s$ uptime are presented on Figs. 6(d) 6(e) and 6(f).

Overall, the energy consumed by the nodes increases with $180s$ uptime and for both, LoRa and Nb-IoT. Compared to the $60s$ case, the energy consumption trends are similar and a negative relationship with the delivery success is visible. Both wireless technology consume roughly the same for the *Extended* and *Hints and Extended* policies. However, for the *Baseline* and *Hints*, Nb-IoT consumes approximately 75% more energy with large data size (not visible on Figs. 6(d) and 6(e) but can be explored online [27]).

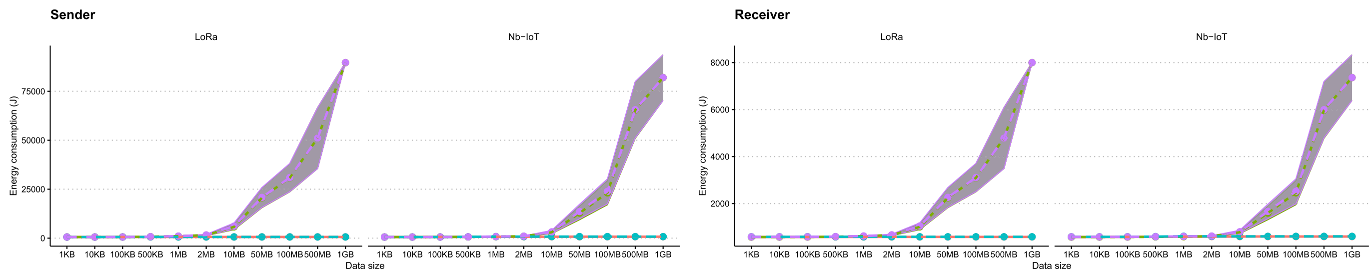
A $180s$ uptime duration allows to achieve longer communications with the *Baseline* and *Hints* policies. For LoRa the delivery success bottleneck for the *Baseline* and *Hints* policies shown on Fig. 6(f) shifted from 500 KB to 2 MB. For Nb-IoT, this bottleneck arises at 10 MB. Uptime duration is thus an important leverage to consider in this context to increase the delivery success with large data sizes.

Finally, larger uptime duration provides a delivery success that is more deterministic. Overall, the standard deviation is reduced with $180s$ uptime duration compared to $60s$.

6.3. Summary

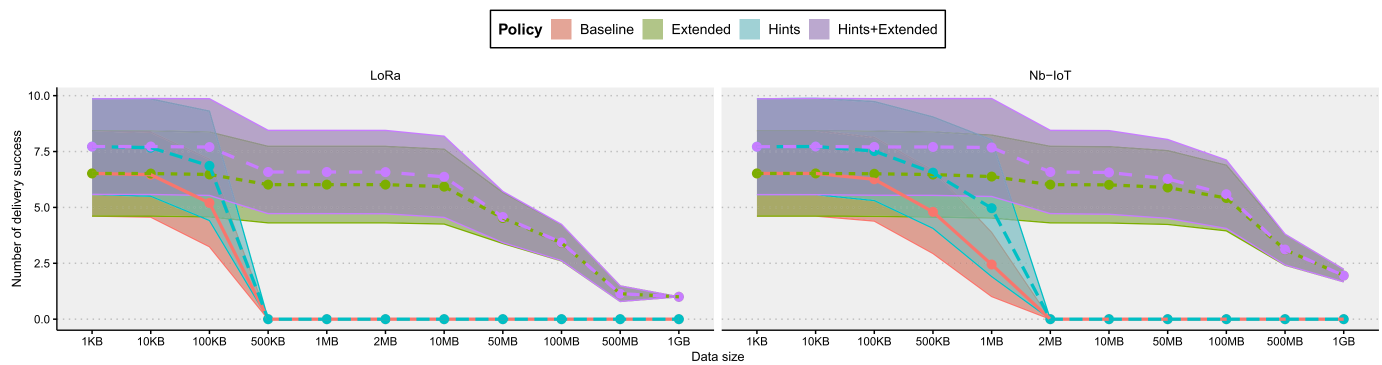
These scalability results illustrate important leverages to consider when disseminating large data size in context like the DAO. First, the performance of wireless technology in terms of bandwidth and energy consumption has a major impact on the delivery success of the policies and the energy consumption of the nodes. In these cases, trading energy consumption for higher bandwidth can be a good solution to increase the delivery success. Second, the uptime duration is a leverage that can improve significantly the delivery success of certain policies such as *Baseline* and *Hints*. Overall, having longer uptime duration provides higher delivery success. Third, using the correct policy can also be a crucial leverage to meet an energy budget and a delivery success target. Works as [30] provide early results on predicting which policy must be used in this use-case.

Overall, the scalability experiments show an energy consumed by nodes that can be drastically increased, on large-scale CPS and for large disseminated data size. Thus, providing energy saving strategies is a crit-

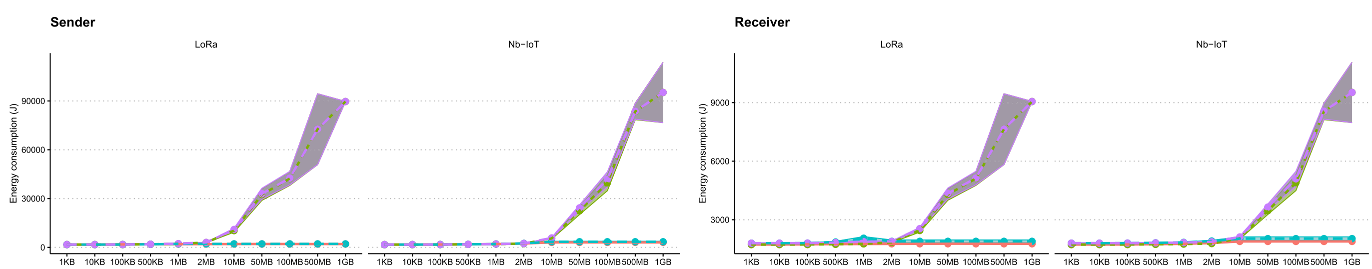


(a) Sender energy consumption in Joules for 60s uptime

(b) Receiver energy consumption in Joules for 60s uptime

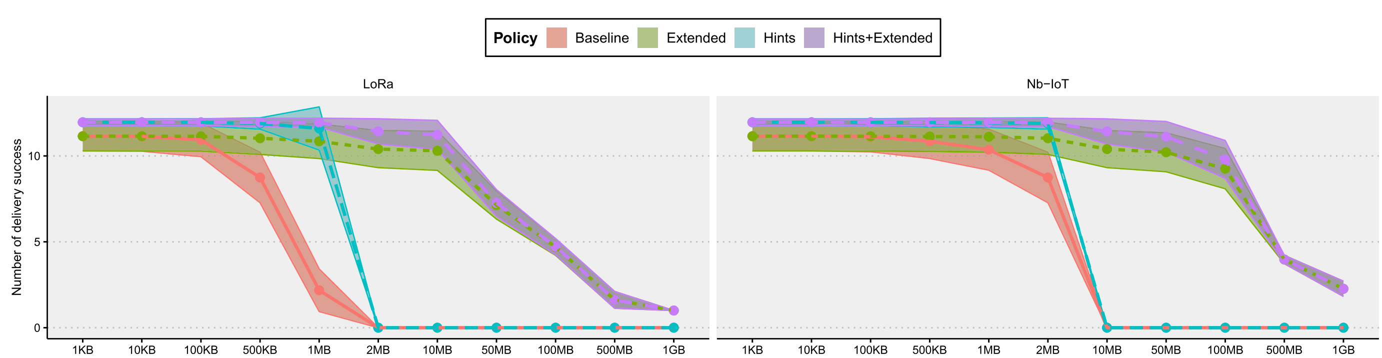


(c) Delivery success for 60s uptime



(d) Sender energy consumption in Joules for 180s uptime

(e) Receiver energy consumption in Joules for 180s uptime



(f) Delivery success for 180s uptime

Fig. 6. Scalability results obtained by varying the data size disseminated by the sender from 1 KB to 1 GB. Each run comprises 13 nodes, 1 sender and 12 receivers. Standard deviation of each curve over 200 runs is represented with their respective background color.

Table 2
Energy consumption standard deviations (Std).

Strategy	Table	Min Std	Max Std	Median Std
Shutdown on receive	3	$6J$	$203J$	$41J$
Unschedule on receive	4	$6J$	$231J$	$39J$
Far Hint	5	$6J$	$301J$	$46J$
Combined	6	$6J$	$223J$	$43J$

ical need. The following section evaluates the energy saving strategies proposed in this work.

7. Strategies evaluation

This section analyzes the strategies presented in Section 4. Results for each metric are presented into tables, and correspond to an average over 200 runs conducted with different node schedule. Both wireless technologies and uptime duration are covered. The tables provide comparison between our previous results presented in [1] using color signed numbers. Green indicates positives impacts and red negatives impacts. To highlight the stability of the energy consumption results, a summary about the energy consumption standard deviations for all results is reported on Table 2. Note that the median standard deviations are all low compared to the actual energy values presented in the tables. Thus, the standard deviation for each individual energy consumption results in Tables 3, 4, 5, 6 is omitted for clarity and readability. To further explore these standard deviations, the datasets are available online [27].

7.1. Strategy 1: shutdown on receive

The Table 3 shows the results for the *Shutdown on receive* strategy for both $60s$ and $180s$ uptime duration and using LoRa and Nb-IoT.

Using *Shutdown on receive* with LoRa and an uptime of $60s$ has no impact compared to our previous results. This policy does not introduce any change on the energy consumption, the delivery success nor the accumulated uptime. With an uptime of $60s$ the performance for LoRa does not allow to transmit 1 MB of data. In this scenario, nodes essentially rely on the extended policy to allows for data transmission. Although the *Shutdown on receive* shutdown the receiver when the data are received, since nodes never receive the data, this strategy has no impact on the simulation outcomes.

Overall, the results using Nb-IoT and $60s$ show a small improvement on the energy consumption of the receivers with up to 7.21 J saved on the receive side for the *Hints and Extended* policy. This energy saving means that Nb-IoT provides faster data transmission that leads to early shutdown of the receivers. Despite the energy saved, this scenario provides less energy efficiency overall. The average delivery success of each policy is reduced at worst by 0.26 which is too high to benefit from the energy saved.

For $180s$ uptime using LoRa, results show improvement of the receivers energy consumption with up to 45.18 J saved with the *Shutdown on receive* strategy and the *Hints* policy. However, compare to our previous results, the *Hints* policy gives worse energy efficiency (+2.35J/S) with a significant drop in the data delivery success (−0.38). It shows that the *Hints* policy is very sensitive to the *Shutdown on receive* strategy has the hint forwarding mechanism is impacted. The sender is also affected (+6.84J/S in energy efficiency for the *Hints* policy) since more hints are sent. Still, the *Hints and Extended* policy is less sensible to this strategy and has a slightly better energy efficiency (−1.14J/S).

Results for the $180s$ uptime and Nb-IoT show a clear improvement of the receiver energy consumption. Up to 130.97 J is saved with the *Hints* policy. Despite a minor reduction of the delivery success (−0.06) for the *Hints* and *Extended* policy, the energy efficiency has improved up to $-10.29J/S$ for the receivers with a small increase on the energy efficiency on the sender side (up to $+0.83J/S$).

Results demonstrate that the *Shutdown on receive* strategy is not energy efficient on low uptime scenarios. In such scenarios, the energy saved by shutting down the nodes is so small, that a slightly lower data delivery success reduces the energy efficiency of the system. On long uptime scenarios, the energy efficiency is improved in most cases. The duration of the data transmission should also be taken into account. If this duration exceeds the uptime duration (scenarios with the extended policy), no energy can be saved with this strategy. The time left between the reception of the data and the end of the uptime should be considered to ensure a good energy efficiency.

The *Shutdown on receive* strategy has a side effect. The sender energy efficiency is equal or worst (up to $+6.84J/S$) in all studied cases. Nodes that previously rely on receivers to get hints are now more likely to communicate directly with the sender, leading to a higher energy consumption for the sender. But, in most cases, the energy efficiency improvement on the receiver side is much higher and balances this drawback.

7.2. Strategy 2: unschedule on receive

Table 4 shows the results for the *Unschedule on receive* strategy. The results for this strategy show that only on the *Hints* and *Hints and Extended* policies get impacted. Since the unscheduled uptimes are solely used by the hint forwarding mechanism, only the policies that uses hints are impacted.

With LoRa and $60s$ uptime, the *Hints* policy is not impacted. No data are delivered thus no unscheduled uptimes. Results for *Hints and Extended* show that not enough energy is saved by senders and receivers to achieve better energy efficiency.

Results for Nb-IoT with $60s$ uptime show an improvement on the nodes energy consumption with up to −5.01 J saved on the *Hints and Extended* policy. The energy improvement is also visible on the sender side since less hints are forwarded to the receivers due to fewer amount of uptime (unscheduled). The energy overhead is also lower with up to −0.84 on the *Hints and Extended* policy compared the previous results. However, these improvements are not sufficient to be energy efficient. This is due to the lower data delivery success that range between −0.14 and −0.15 for both *Hints* and *Hints and Extended* policies.

With LoRa and $180s$ uptime a greater energy is saved. This is particularly visible on the *Hints and Extended* policy that saves in average 28.34 J on the receivers and 6.92 J on the senders. The delivery success is slightly higher with the *Hints* policy. This is caused by the receivers unscheduled time slots that allow the sender to reach other potential receivers and deliver either hints or data, leading to a higher delivery success. Even if this increase is very small on the *Hints* policy (+0.01) and the sender is consuming more energy (+1.69 J), this allows better energy efficiency on both sender and receiver.

Results for Nb-IoT and $180s$ uptime show improvements in the energy consumption, the energy overhead and the energy efficiency. Both impacted policies have a lower delivery success (at least −0.04) leading to worse sender energy efficiency. But, the energy saved on the receiver nodes allows for a more energy efficient CPS. For example, the *Hints and Extended* policy has a worse sender energy efficiency (+0.92J/S) still, the receiver energy efficiency has improved with −5.22J/S.

To summarize, the *Unschedule on receive* strategy impacts the policies that use extra uptimes to propagate hints namely the *Hints* and *Hints and Extended* policies. In theory, these extra uptimes can drastically increase the energy consumption of receiver nodes. However, the *Unschedule on receive* strategy shows a limited impact on the results compare to *Shutdown on receive*. This means that, scenarios with extra uptimes are rare in that case. Overall, this strategy has a positive impact on the energy consumption of the senders and the receivers. But, on low uptime scenarios ($60s$), this impact is not sufficient to compensate a lower delivery success.

Table 3
Simulation results using the *Shutdown on receive* strategy. Comparison between our previous results [1] is in color. Green indicates improvements, red shows regressions and blue indicates no change.

Uptime	Scenario	$\#Succ_p$	Energy Consumption (J)		$eOvhd(p)$ (%)		$eff(p)$ (J/S)	
			Sender	Receiver	Sender	Receiver	Sender	Receiver
LoRa								
60	baseline	0 =	617.37 =	581.14 =	0 =	0 =		
	extended	6.02 =	1004.36 =	612.06 =	+62.68 =	+5.32 =	166.84 =	101.67 =
	hint	0 =	628.74 =	586.07 =	+1.84 =	+0.85 =		
	hintandextended	6.54 =	1035.76 =	619.62 =	+67.77 =	+6.62 =	158.25 =	94.67 =
180	baseline	2.18 =	2032.69 =	1763.96 -1.05	0 =	0 =	932.43 =	809.16 -0.48
	extended	10.86 =	2201.59 =	1764.07 -2.93	+8.31 =	+0.01 -0.11	202.82 =	162.51 -0.27
	hint	10.8 -0.38	2133.02 +1.35	2028.98 -45.18	+4.94 +0.07	+15.02 -2.49	197.59 +6.84	187.96 +2.35
	hintandextended	11.85 -0.04	2251.64 -7.45	1879.74 -19.96	+10.77 -0.37	+6.56 -1.07	190.01 +0.01	158.63 -1.14
Nb-IoT								
60	baseline	2.44 =	714.79 =	592.29 -1.22	0 =	0 =	292.35 =	242.25 -0.5
	extended	6.38 =	760.83 =	588.96 -2.32	+6.44 =	-0.56 -0.19	119.25 =	92.31 -0.36
	hint	4.69 -0.11	777.89 +6.03	608.42 -6.39	+8.83 +0.84	+2.72 -0.86	165.86 +5.23	129.73 +1.77
	hintandextended	7.32 -0.26	785.91 -4.95	602.98 -7.21	+9.95 -0.69	+1.8 -1.01	107.29 +3.09	82.32 +1.92
180	baseline	10.37 =	2034.67 =	1729.62 -35.49	0 =	0 =	196.3 =	166.87 -3.42
	extended	11.12 =	2026.21 =	1717.55 -35.28	-0.42 =	-0.7 =	182.3 =	154.53 -3.17
	hint	11.79 -0.06	2054.06 +0.27	1937.27 -130.97	+0.95 +0.01	+12.01 -5.17	174.22 +0.83	164.31 -10.29
	hintandextended	11.85 -0.06	2041.5 -1.05	1916.28 -123.44	+0.34 -0.05	+10.79 -4.77	172.35 +0.78	161.78 -9.55

Table 4
Simulation results using the *Unschedule on receive* strategy. Comparison between our previous results [1] is in color. Green indicates improvements, red shows regressions and blue indicates no change.

Uptime	Scenario	$\#Succ_p$	Energy Consumption (J)		$eOvhd(p)$ (%)		$eff(p)$ (J/S)	
			Sender	Receiver	Sender	Receiver	Sender	Receiver
LoRa								
60	baseline	0 =	617.37 =	581.14 =	0 =	0 =		
	extended	6.02 =	1004.36 =	612.06 =	+62.68 =	+5.32 =	166.84 =	101.67 =
	hint	0 =	628.74 =	586.07 =	+1.84 =	+0.85 =		
	hintandextended	6.46 -0.08	1031.35 -4.41	617.27 -2.34	+67.05 -0.71	+6.22 -0.4	159.53 +1.28	95.48 +0.81
180	baseline	2.18 =	2032.69 =	1765.01 =	0 =	0 =	932.43 =	809.64 =
	extended	10.86 =	2201.59 =	1767 =	+8.31 =	+0.11 =	202.82 =	162.78 =
	hint	11.19 +0.01	2133.37 +1.69	2073.37 -0.79	+4.95 +0.08	+17.47 -0.04	190.73 -0.02	185.37 -0.24
	hintandextended	11.85 -0.04	2252.17 -6.92	1871.36 -28.34	+10.8 -0.34	+6.03 -1.61	190.14 +0.14	157.99 -1.79
Nb-IoT								
60	baseline	2.44 =	714.79 =	593.52 =	0 =	0 =	292.35 =	242.75 =
	extended	6.38 =	760.83 =	591.28 =	+6.44 =	-0.38 =	119.25 =	92.68 =
	hint	4.66 -0.14	768.19 -3.67	612.71 -2.1	+7.47 -0.51	+3.23 -0.35	164.67 +4.03	131.34 +3.39
	hintandextended	7.44 -0.15	786.55 -4.31	605.19 -5.01	+10.04 -0.6	+1.97 -0.84	105.72 +1.52	81.34 +0.95
180	baseline	10.37 =	2034.67 =	1765.11 =	0 =	0 =	196.3 =	170.3 =
	extended	11.12 =	2026.21 =	1752.83 =	-0.42 =	-0.7 =	182.3 =	157.7 =
	hint	11.8 -0.04	2053.65 -0.13	1994.3 -73.95	+0.93 -0.01	+12.98 -4.19	174.04 +0.65	169.01 -5.6
	hintandextended	11.84 -0.07	2041.39 -1.16	1965.96 -73.76	+0.33 -0.06	+11.38 -4.18	172.49 +0.92	166.11 -5.22

7.3. Strategy 3: Far Hint

The results for the *Far Hint* strategy are presented in the Table 5. As this strategy impacts policies that use hints, only the *Hints* and *Hints and Extended* policies will be covered.

The results for the LoRa wireless technology and $60s$ uptime show an impact only on the *Hints and Extended* policy. Since data cannot be sent successfully in this scenario with the *Hints* policy, the simulation is most likely to be the same and only hint can be exchange among the nodes. The *Hints and Extended* policy shows great improvements in terms of energy efficiency. Even if the energy consumption and the energy overhead compare to the *Baseline* is worse (+97.74 J on the energy consumption of the receivers and +15.83 on the energy overhead of the senders), the number of delivery success has increase significantly with +1.56. Therefore, this strategy is a good approach to improve data dissemination on low delivery success scenarios.

Similarly, the results for the Nb-IoT wireless technology using $60s$ uptimes show a significant improvement in the delivery success with up to +2.92 of increase. Both impacted policies suffer from an increase in the energy consumption of the sender (up to +84.31 J for the *Hints* policy) and the receiver (up to +51.62 J for the *Hints and Extended* policy) leading to worse energy overhead. It is the consequence of three factors. First, hint is located further away in the future, thus hints are propagated for a longer time (more communications), which increases the energy consumption. Second, since the delivery success increases, there is more data transmission, leading to more energy consumed. Finally, since hints are further away in the future, the *Shutdown on receive* strategy tends to delay the moment were all receivers receive the data ($\#Succ_p = 12$). Consequently, senders and receivers take more time to reach 12 successful deliveries and thus consume more energy. Nonetheless, the energy efficiency of the system is greatly improved. Taking into account all the scenarios, the average energy efficiency has improved with -49.73J/S and -41.8J/S for the sender and the receiver respectively.

The results for the LoRa wireless technology and $180s$ uptime are balanced. The delivery success of the *Hints* and the *Hints and Extended* policies are already good. Thus, it is more difficult to perform better in terms of energy efficiency with this strategy. As explain before, the *Far*

Table 5
Simulation results using the *Far Hint* strategy. Comparison between our previous results [1] is in color. Green indicates improvements, red shows regressions and blue indicates no change.

Uptime	Scenario	$\#Succ_p$	Energy Consumption (J)		$eOvhd(p)$ (%)		$eff(p)$ (J/S)	
			Sender	Receiver	Sender	Receiver	Sender	Receiver
LoRa								
60	baseline	0 =	617.37 =	581.14 =	0 =	0 =		
	extended	6.02 =	1004.36 =	612.06 =	+62.68 =	+5.32 =	166.84 =	101.67 =
	hint	0 =	628.74 =	586.07 =	+1.84 =	+0.85 =		
	hintandextended	8.11 +1.56	1133.5 +97.74	642.2 +22.58	+83.6 +15.83	+10.51 +3.89	139.85 -18.4	79.24 -15.44
180	baseline	2.18 =	2032.69 =	1765.01 =	0 =	0 =	932.43 =	809.64 =
	extended	10.86 =	2201.59 =	1767 =	+8.31 =	+0.11 =	202.82 =	162.78 =
	hint	11.68 +0.5	2137.52 +5.85	2140.98 +66.82	+5.16 +0.29	+21.3 +3.79	183.01 -7.75	183.3 -2.3
	hintandextended	11.99 +0.1	2254.61 -4.48	1985.64 +85.94	+10.92 -0.22	+12.5 +4.87	187.96 -2.04	165.54 +5.77
Nb-IoT								
60	baseline	2.44 =	714.79 =	593.52 =	0 =	0 =	292.35 =	242.75 =
	extended	6.38 =	760.83 =	591.28 =	+6.44 =	-0.38 =	119.25 =	92.68 =
	hint	7.72 +2.92	856.17 +84.31	665.08 +50.27	+19.78 +11.8	+12.06 +8.47	110.9 -49.73	86.15 -41.8
	hintandextended	10.22 +2.63	864.57 +73.72	661.81 +51.62	+20.96 +10.31	+11.51 +8.7	84.55 -19.64	64.72 -15.67
180	baseline	10.37 =	2034.67 =	1765.11 =	0 =	0 =	196.3 =	170.3 =
	extended	11.12 =	2026.21 =	1752.83 =	-0.42 =	-0.7 =	182.3 =	157.7 =
	hint	11.98 +0.14	2057.18 +3.39	2259.92 +191.67	+1.11 +0.17	+28.03 +10.86	171.65 -1.74	188.56 +13.95
	hintandextended	11.99 +0.09	2044.79 +2.24	2237.9 +198.18	+0.5 +0.11	+26.79 +11.23	170.54 -1.03	186.65 +15.31

Hint strategy adds delay in scenario that reach 12 successful deliveries. The slight +0.5 delivery success improvement for the *Hints* policy makes it more energy efficient on the sender and the receiver.

Using Nb-IoT with an uptime of 180s leads to worse energy efficiency. Without any strategy enabled, these policies perform well in these scenarios with more than 10 delivery success. The *Far Hint* strategy is an overhead in such case and has a negative impact on the energy consumption. This strategy adds around 200 J to the receiver energy consumption corresponding to more than two complete idle time slots leading to a worse receiver energy efficiency.

The key feature of the *Far Hint* strategy is its ability to improve the delivery success of all policies. The additional time used to propagate the hint allows to reach more receivers, leading to a better data deliveries. The main down side of this strategy is its impact on the node energy consumption that can increase significantly. Since hints timestamps are further away in the future, the *Far Hint* strategy is delaying the data delivery that impact the scenario with high delivery success. Hence, this strategy performs better in terms of energy efficiency on scenario with low delivery success.

7.4. Strategy 4: Combined

Results for the *Combined* strategy are presented in Table 6. Using the LoRa wireless technology and 60s uptime, the results show an improvement for the *Hints and Extended* policy on the delivery success with +1.51, the energy efficiency of the sender (−17.88J/S) and receiver (−15.27J/S). As expected, *Hints and Extended* are the only impacted policy. The *Shutdown on receive* and *Far Hint* strategies are the only ones to affect this scenario (negatively and positively). Since, *Combined* combines the effects of both, it performs slightly worse compared to *Far Hint* alone. Still, the energy performance has improved.

The results for the Nb-IoT wireless technology and the 60s uptime show an increase of the delivery success. Using *Shutdown on receive* allows to save energy on the *Baseline* and *Extended* policies by reducing the energy consumption of the receivers. Then, the *Combined* strategy increases the delivery success of the *Hints* and *Hints and Extended* policies up to +2.23 and improves their energy efficiency. Still, the *Combined* strategy offers less improvement on these policies compare to the *Far Hint* strategy. For example, the energy efficiency for the *Combined* strategy improves by −26.21J/S for the *Hints* policy where the *Far Hint* policy provides an improvement of −49.73J/S (cf. Table 5).

The results for the LoRa wireless technology and the 180s uptime show better energy efficiency for most cases except for the *Hints and Extended* policy. Even if a small increase of +0.08 in the delivery success is noticeable, it is not enough to compensate for the expense of the sender (+21.51 J) and the receiver (+26.79 J). Since the strategies are combined, it mitigates the negative effects that the *Far Hint* strategy has on the *Hints and Extended* policy.

Finally, the results for the Nb-IoT wireless technology and the 180s uptime show an improvement with all policies. Despite a high delivery success for each policy, the energy consumption of the receiver for the *Baseline* and *Extended* policies has decrease up to 3.9 J. Combining strategies allow to benefit from their individual effects.

To summarize, combining strategies allows to merge the effects of each individual strategy. It allows to leverage more scenarios. In this case, most of the scenarios improved in terms of delivery success and energy efficiency. In some scenarios such as the Nb-IoT/Hints/60s, individual policy may perform better in terms of delivery success, energy overhead and energy efficiency. Combining strategies is a promising idea to maximize the performance of the system. Carefully selecting the strategies to combine is important, as it can strongly impact the performance outcomes

8. Discussion

This section discusses the conclusions and future directions of the work. A summary of the results, including the ones from previous work [1] is given. Table 7 summarizes the results of policies and strategies studies and, Table 8 the ones of the scalability studies.

8.1. Impact of the number of nodes

In the DAO context (i.e., having small cliques of nodes isolated from each other usually of size 10-15 [1]), 100 reachable nodes is large scale. The scalability evaluation for the number of nodes shows that it is challenging to achieve similar performance in small and large networks. Despite an increase of the uptime overlap likelihood with the number of node, reaching all the receivers is still challenging. The more receivers there is, the longer it takes to disseminate data to all of them, the more energy is consumed by the CPS. If delivering data to the same proportion of nodes on small and large networks is critical, policies such as *Hints* and *Hints and Extended* must be used as they provide additional uptimes to disseminate the data.

The results also show that policies relying on hints lead to an increase of the receiver energy consumption. This phenomenon is even more significant on dense networks. Hence, for large-scale CPS deployments, the

Table 6
Simulation results using the *Combined* strategy. Comparison between our previous results [1] is in color. Green indicates improvements, red shows regressions and blue indicates no change.

Uptime	Scenario	$\#Succ_p$	Energy Consumption (J)		$eOvhd(p)$ (%)		$eff(p)$ (J/S)	
			Sender	Receiver	Sender	Receiver	Sender	Receiver
LoRa								
60	baseline	0 =	617.37 =	581.14 =	0 =	0 =		
	extended	6.02 =	1004.36 =	612.06 =	+62.68 =	+5.32 =	166.84 =	101.67 =
	hint	0 =	628.74 =	586.07 =	+1.84 =	+0.85 =		
	hintandextended	8.05 +1.51	1130.67 +94.91	639.55 +19.93	+83.14 +15.37	+10.05 +3.43	140.37 -17.88	79.4 -15.27
180	baseline	2.18 =	2032.69 =	1764.97 -0.04	0 =	0 =	932.43 =	809.62 -0.02
	extended	10.86 =	2201.59 =	1766.76 -0.24	+8.31 =	+0.1 -0.01	202.82 =	162.76 -0.02
	hint	11.57 +0.39	2136.67 +5	2096.91 +22.75	+5.12 +0.25	+18.81 +1.29	184.67 -6.08	181.24 -4.37
	hintandextended	11.97 +0.08	2280.6 +21.51	1926.49 +26.79	+12.2 +1.06	+9.15 +1.52	190.53 +0.53	160.94 +1.17
Nb-IoT								
60	baseline	2.44 =	714.79 =	593.47 -0.05	0 =	0 =	292.35 =	242.73 -0.02
	extended	6.38 =	760.83 =	591.14 -0.14	+6.44 =	-0.39 -0.02	119.25 =	92.65 -0.02
	hint	6.11 +1.31	821.32 +49.46	627.96 +13.15	+14.9 +6.92	+5.81 +2.22	134.42 -26.21	102.78 -25.18
	hintandextended	9.82 +2.23	858.74 +67.89	635.36 +25.17	+20.14 +9.5	+7.06 +4.25	87.45 -16.75	64.7 -15.69
180	baseline	10.37 =	2034.67 =	1761.97 -3.14	0 =	0 =	196.3 =	169.99 -0.3
	extended	11.12 =	2026.21 =	1748.93 -3.9	-0.42 =	-0.74 -0.04	182.3 =	157.35 -0.35
	hint	11.96 +0.12	2055.21 +1.43	2028.09 -40.16	+1.01 +0.07	+15.1 -2.07	171.84 -1.55	169.57 -5.04
	hintandextended	11.97 +0.07	2044.19 +1.64	2019.51 -20.2	+0.47 +0.08	+14.62 -0.94	170.71 -0.87	168.64 -2.69

Table 7
Summary of result's trends for policies, strategies and wireless technologies.

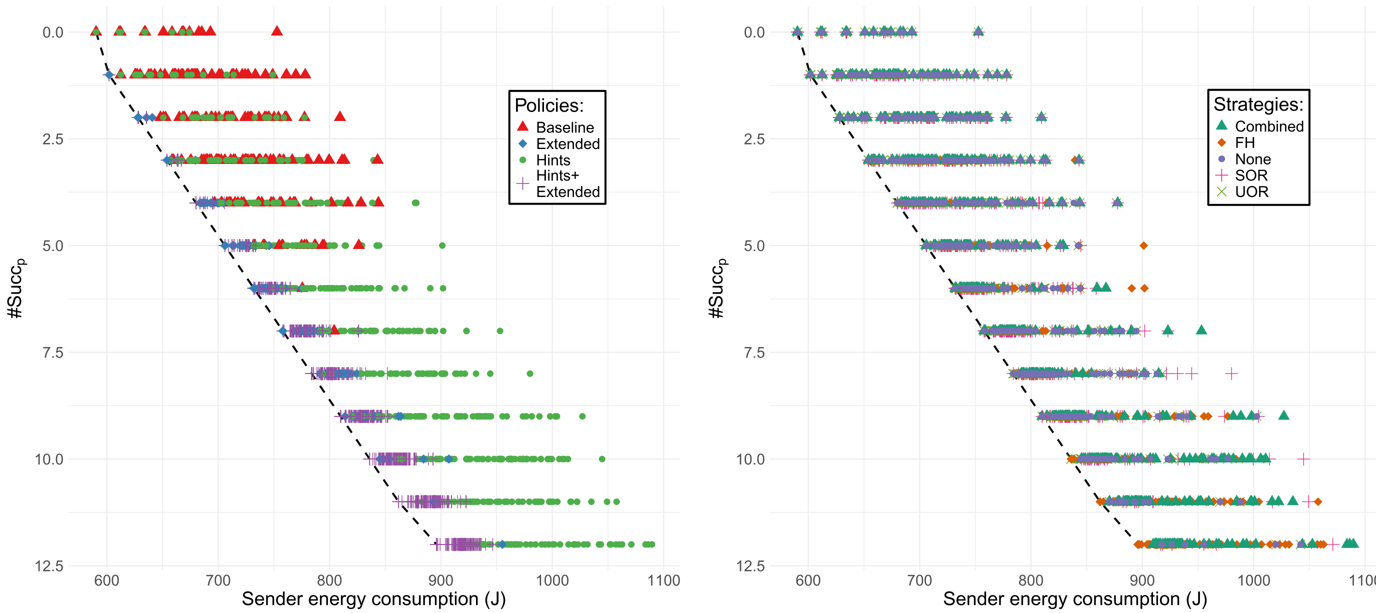
		Energy		
		Sender	Receiver	$\#Succ$
Policies	*Baseline*	=	=	=
	Extended	--	=	++
	Hints	-	-	++
	Hints and Extended	---	-	+++
Strategies	*Shutdown on receive*	-	++	-
	Unschedule on receive	+	+	-
	Far Hint	--	--	+
	Combined	---	=	+
Wireless	LoRa	=	=	=
	NbIoT	++	=	+++

+ good, ++ very good, +++ excellent, = fair, - bad, -- very bad, --- worst Policies comparisons use *Baseline* as reference. Wireless technologies comparisons use LoRa as reference.Strategies comparisons depict overall results trends.

Table 8
Summary of scalability result's trends for policies and wireless technologies.

		Many nodes			Large data sizes		
		Energy			Energy		
		Sender	Receiver	$\#Succ$	Sender	Receiver	$\#Succ$
Policies	*Baseline*	=	=	=	=	=	=
	Extended	--	=	++	---	---	++
	Hints	-	---	+	-	-	+
	Hints and Extended	---	--	+++	---	---	+++
Wireless	LoRa	=	=	=	=	=	=
	NbIoT	=	--	+++	-	-	++

+ good, ++ very good, +++ excellent, = fair, - bad, -- very bad, --- worst Policies comparisons use *Baseline* as reference. Wireless technologies comparisons use LoRa as reference.Strategies comparisons depict overall results trends.



(a) Policies are highlighted

(b) Strategies are highlighted

Fig. 7. Comparison between the average sender node's energy consumption and the average delivery success for each policy and strategies, using Nb-IoT with 60 s uptime. Pareto-front is highlighted, by a dashed line.

Extended policy is the best compromise between data dissemination and energy consumption.

Multiple solutions can be used to mitigate the impact of the policies on the receivers energy consumption. First, using a technology such as LoRa to disseminate hints and Nb-IoT to transfer the data. Further studies must be conducted on this regard. Second, our evaluation does not investigate scenarios where part of the nodes uses different policies. This approach of using heterogeneous policies may reduce significantly the energy consumed by the receivers while maintaining good dissemination performance. Performing such extended studied, with an increased number of scenarios to investigate, is a future work. A model that chooses the correct policy to use on the fly can also be contributed as a future work.

The strong correlation between the delivery success and the sender energy consumption reveals that, allowing a subset of receivers to forward the data (such as most data dissemination solutions) may help to avoid draining the sender's node battery. This approach could help to reduce the amount of data transmitted on the sender side. Experimentation is required to test these hypotheses on loosely coupled CPS.

8.2. Impact of the data size

The data size scalability experiments demonstrate that the wireless technology, uptime duration and policies play a major role for having high delivery success and energy efficient nodes. More studies are required to provide trade-offs between these different parameters for a given data size.

Depending on the use case, the size of the disseminated data can vary significantly for a given system. In that case, dynamically changing the policies, uptime duration and wireless technology of the nodes according to the size of the disseminated data could ensure higher delivery success and lower energy consumption of the nodes. Further experiments are required on this regard.

This work shows the importance of reducing and compressing data in constraints environment contexts like the DAO. Works such as [11] provide valuable results on this leverage.

8.3. Energy consumption and delivery success trade-off

The strategies evaluation results show that, improvements in the delivery success often lead to higher energy consumption. A trade-off between both metrics must be found. Fig. 7(a) and Fig. 7(b), detail these existing trade-offs obtained section 7 using Nb-IoT and 60 s uptime. The figures show a parallel between the average energy consumed by the sender and the delivery success, achieved by each data dissemination policy and strategy. All configurations on the Pareto-front are highlighted and linked together with a dashed line. Among these configurations, 18 are from *Far Hint*, 15 from *Shutdown on receive*, 18 from *Unschedule on receive*, 13 from *Combined* and 17 when no strategy is used. Concerning the policies, 5 are from *Baseline*, 5 from *Hints*, 55 *Extended* and 16 from *Hints and Extended*.

As expected, the *Extended* policy offers a good trade-off between the energy consumption and the delivery success. However, choosing one of the strategies to balance the energy consumption and delivery success depends on the objective trade-off. In addition, different wireless technology and uptime duration, lead to different configurations on the pareto-front. These results also demonstrate that, better dissemination is often associated with a higher energy consumption.

8.4. Predicting the impact of a strategy

The analysis of the results reveal that, predicting the impact of a strategy on the simulation outcomes is difficult. A strategy that appear to save energy can leads to lower delivery success and in turn, leverage the energy efficiency. Other factors such as the wireless technology and the uptime duration have a non-negligible impact on the simulation results. Performing simulations and real deployments is important to have a full understanding of an energy saving strategy. Having a strategy that saves energy is not sufficient to conclude that it is more energy efficient.

8.5. Choosing the correct strategy combination

In this work, we choose to combine all the strategies to study collective impact. Although, carefully choosing the strategies to combine is a better approach. Results show that, among the strategies, *Unschedule on receive* strategy is the one that provides least significant improvements.

Removing this strategy from the *Combined* one could lead to better overall improvements. But, this work aims to be as general as possible and provides new research insights and directions.

8.6. Strategies adaptability

Another concern that needs to be addressed is the adaptability of the proposed strategies to various changes in network conditions and node behavior over time. After deployment, nodes reachability and network performance can vary. Clock drifts, energy shortage and node failures are phenomenons that affect nodes behavior during their operation. These uncertainties can affect the performance of the strategies in terms of energy consumption and delivery success. This work investigates the best that can be leveraged from these strategies in a stable network scenario, but with nodes being autonomously turned On (for a short amount of time) and Off. To further evaluate the strategies under more variable conditions, additional simulations must be performed along with the deployment of prototypes.

8.7. Far hint timestamp

Choosing the correct duration to use with the *Far Hint* strategy is not trivial. Far hint uses a hint duration of three hours. Receivers are informed of the next sender uptime, located three hours away from the current one. But, using different hint duration may produce different results. A too short hint duration may result in small to no improvements in terms of delivery success and energy consumption. The results show that, having larger hint duration induces more communications (due to the hint forwarding mechanism) causing higher energy consumption for both senders and receivers. But, long hint duration can significantly increase the delivery success, since hints have more time to propagate. Meanwhile, on scenarios that converge quickly toward a complete data dissemination (12 successful data deliveries), a long hint duration just results in higher energy consumption. The take-away message is that, choosing the correct hint duration depends on the use case, and whether trading energy consumption for delivery success can be afforded.

In real scenario, the density of the network should be taken into account. In this work, we use classical flooding to forward timestamps. In very dense networks such as dense Wireless Sensors Network, this could lead to broadcast storm effects [26]. Our case assumes that we are using less dense networks with sporadic data transmissions. The fact that hints forwarding stops as soon as the duration expires, allows to limit those effects and mitigate re-transmissions.

9. Conclusion

The Arctic tundra is a very hostile environment. Deploying nodes and ensuring proper power supply can be difficult, specially on hard weather conditions. It is important to provide energy efficient solutions to disseminate data to neighboring nodes and remotely located servers. In this work, we investigate such dissemination policies on small and large-scale networks. We quantify the impact of the size of the disseminated data. In addition, we propose several strategies to optimize data dissemination and energy saving. This study is conducted using flow-level network simulations.

The scalability study for the number of nodes reveals that the *Extended* policy is able to handle large-scale networks and consume a reasonable amount of energy. Other polices such as the *Hints and Extended* have a significant impact on the receiver energy consumption but are more efficient at data dissemination. The scalability study for the size of the disseminated data highlights the importance of the wireless technology and the uptime duration. Policies that extend the nodes uptime duration (e.g.: *Extended* and *Hints and Extended*) can drastically increase the nodes energy consumption.

This work also evaluates the effects of the proposed strategies with each data dissemination policies. A direct comparison to the previous results details in [1] is exposed. The results reveal that, predicting the effects of a given strategy is very difficult and experimentation must be conducted prior to real CPS deployments. Strategies such as *Shutdown on receive* and *Unschedule on receive* appear to save energy. However, they impact the data delivery performance which may lead to lower energy efficiency. Similarly, the *Far Hint* strategy has counter intuitive effects since it increases nodes energy consumption but on the overall improves the energy efficiency. Finally, the *Combined* strategy shows that combining strategies have hardly predictable effects. Consequently, providing simulation tools to study such trade-offs is important and enable the development of more efficient Cyber-Physical Systems.

As a future work, experiments on dense networks with heterogeneous policies should be done to reduce the energy consumption of receiver nodes. Regarding the use of multiple strategies, the effects of different strategies combinations must be studied. The goal would be to optimize the energy and the dissemination performance of the system. Measuring the impact of energy saving strategies on the scalability study is envisioned. Investigating the effects of other parameters could help in the comprehension of such data dissemination approach. For example, the impact of environmental variables such as weather conditions (e.g.: temperatures, rain, snow) must be modeled and integrated to the simulations. This will provide results that account for deployment challenges such as node failures. Finally, test-bed experiments are planned [31] for a having a transition from simulation to prototyping and a real-world deployment.

CRediT authorship contribution statement

Loïc Guégan: Conceptualization, and most major roles (methodology software, validation analysis etc.). **Issam Raïs:** Conceptualization, Supervision, Writing – review & editing. **Otto Anshus:** Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

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.

Data availability

No data was used for the research described in the article.

References

[1] I. Raïs, L. Guégan, O. Anshus, Impact of loosely coupled data dissemination policies for resource challenged environments, in: 2022 22nd IEEE International Symposium on Cluster, Cloud and Internet Computing (CCGrid), 2022, pp. 524–533.

[2] D. Kandris, C. Nakas, D. Vomvas, G. Koulouras, Applications of wireless sensor networks: an up-to-date survey, Appl. Syst. Innov. 3 (1) (Feb. 2020) 14, https://www.mdpi.com/2571-5577/3/1/14.

[3] E.A. Kadir, H. Irie, S.L. Rosa, Modeling of wireless sensor networks for detection land and forest fire hotspot, in: 2019 International Conference on Electronics, Information, and Communication (ICEIC), IEEE, Auckland, New Zealand, Jan. 2019, pp. 1–5, https://ieeexplore.ieee.org/document/8706364/.

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

[5] J.J. Lahoz-Monfort, M.J.L. Magrath, A comprehensive overview of technologies for species and habitat monitoring and conservation, BioScience 71 (10) (Oct. 2021).

[6] R. Dinesh, R. Marimuthu, A survey about WSN and IoT based health care applications and ADPLL contribution for health care systems, in: 2019 IEEE 10th International Conference on Awareness Science and Technology (iCAST), IEEE, Morioka, Japan, Oct. 2019, pp. 1–8, https://ieeexplore.ieee.org/document/8923167/.

[7] A. Ali, Y.K. Jadoon, S.A. Changazi, M. Qasim, Military operations: wireless sensor networks based applications to reinforce future battlefield command system, in: 2020 IEEE 23rd International Multitopic Conference (INMIC), IEEE, Nov. 2020, pp. 1–6, https://ieeexplore.ieee.org/document/9318168/.

[8] P. Shrivastava, EPO: an optimization technique for urban traffic management while limiting the pollution using WSN, Int. J. Commun. Syst. 33 (5) (Mar. 2020) e4246, https://onlinelibrary.wiley.com/doi/10.1002/dac.4246.

[9] R.A. Ims, J.U. Jepsen, S. Audun, S.G. Yoccoz, Science plan for COAT: climate-ecological observatory for Arctic tundra, https://www.coat.no/portals/coat/SciencePlanCOAT_web.pdf?ver=2017-10-26-130928-570, 2013.

[10] I. Rais, J.M. Bjorndalen, P. Hoai Ha, K.-A. Jensen, L.S. Michalik, H. Mjoen, O. Tveito, 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, Santorini Island, Greece, May 2019, pp. 625–632, https://ieeexplore.ieee.org/document/8804827/.

[11] I. Rais, O. Anshus, J.M. Bjorndalen, D. Balouek-Thomert, 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, Melbourne, Australia, May 2020, pp. 469–478, https://ieeexplore.ieee.org/document/9139716/.

[12] J.M. Bjørndalen, O.J. Anshus, T. Brox-Larsen, P.H. Ha, R.A. Ims, J.U. Jepsen, S.T. Killengreen, E.F. Kleiven, E.M. Soininen, N.G. Yoccoz, A. Horsch, Building a Sensor System for a Large Scale Arctic Observatory, 2016, p. 6.

[13] E.M. Soininen, I. Jensvoll, S.T. Killengreen, R.A. Ims, Under the snow: a new camera trap opens the white box of subnivean ecology, Remote Sens. Ecol. Conserv. 1 (1) (Oct. 2015) 29–38, https://onlinelibrary.wiley.com/doi/10.1002/rse2.2.

[14] M. Weyn, G. Ergeerts, R. Berkvens, B. Wojciechowski, 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, Tokyo, Japan, Oct. 2015, pp. 54–59, http://ieeexplore.ieee.org/document/7390420/.

[15] M. Weyn, G. Ergeerts, L. Wante, C. Vercauteren, P. Hellinckx, Survey of the DASH7 alliance protocol for 433 MHz wireless sensor communication, Int. J. Distrib. Sens. Netw. 9 (12) (Dec. 2013) 870430, http://journals.sagepub.com/doi/10.1155/2013/870430.

[16] W. Ayoub, A.E. Samhat, F. Nouvel, M. Mroue, J.-C. Prevotet, Internet of mobile things: overview of LoRaWAN, DASH7, and NB-IoT in LPWANs standards and supported mobility, IEEE Commun. Surv. Tutor. 21 (2) (2019) 1561–1581, https://ieeexplore.ieee.org/document/8502812/.

[17] R.K. Singh, P.P. Puluckul, R. Berkvens, M. Weyn, Energy consumption analysis of LPWAN technologies and lifetime estimation for IoT application, Sensors 20 (17) (Aug. 2020) 4794, https://www.mdpi.com/1424-8220/20/17/4794.

[18] R. Sanchez-Iborra, J. Sanchez-Gomez, J. Ballesta-Viñas, M.-D. Cano, A. Skarmeta, Performance evaluation of LoRa considering scenario conditions, Sensors 18 (3) (Mar. 2018) 772, http://www.mdpi.com/1424-8220/18/3/772.

[19] B. Foubert, N. Mitton, Long-range wireless radio technologies: a survey, Future Internet 12 (1) (Jan. 2020) 13, https://www.mdpi.com/1999-5903/12/1/13.

[20] K. Mekki, E. Bajic, F. Chaxel, 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, Athens, Mar. 2018, pp. 197–202, https://ieeexplore.ieee.org/document/8480255/.

[21] A. Seferagić, J. Famaey, E. De Poorter, J. Hoebeke, Survey on wireless technology trade-offs for the industrial Internet of things, Sensors 20 (2) (Jan. 2020).

[22] D.K. Singh, R. Sobti, Wireless communication technologies for Internet of things and precision agriculture: a review, in: 2021 6th International Conference on Signal Processing, Computing and Control (ISPCC), IEEE, Solan, India, Oct. 2021.

[23] Zehua Zhou, Xiaojing Xiang, Xin Wang, An energy-efficient data-dissemination protocol inWireless sensor networks, in: 2006 International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM'06). Buffalo-Niagara Falls, IEEE, NY, USA, 2006, pp. 13–22, http://ieeexplore.ieee.org/document/1648440/.

[24] H. Sabbineni, K. Chakrabarty, Location-aided flooding: an energy-efficient data dissemination protocol for wireless-sensor networks, IEEE Trans. Comput. 54 (1) (Jan. 2005) 36–46, http://ieeexplore.ieee.org/document/1362638/.

[25] A. Antonopoulos, C. Verikoukis, Multi-player game theoretic MAC strategies for energy efficient data dissemination, IEEE Trans. Wirel. Commun. 13 (2) (Feb. 2014) 592–603, http://ieeexplore.ieee.org/document/6684543/.

[26] X.-L. Zheng, M. Wan, A survey on data dissemination in wireless sensor networks, J. Comput. Sci. Technol. 29 (3) (May 2014) 470–486, http://link.springer.com/10.1007/s11390-014-1443-8.

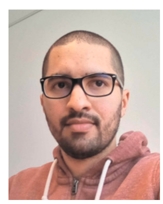
[27] Loïc Guégan, Experiments GitLab repository for repeatability, https://gitlab.com/manzerbredes/loosely-coupled-dss-extended, 2024.

[28] J. Geerling, Raspberry pi power consumption benchmarks, https://www.pidramble.com/wiki/benchmarks/power-consumption.

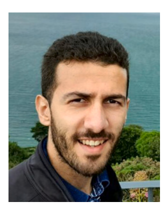
[29] R.S. Sinha, Y. Wei, S.-H. Hwang, A survey on LPWA technology: LoRa and NB-IoT, ICT Express 3 (1) (Mar. 2017) 14–21, https://linkinghub.elsevier.com/retrieve/pii/S2405959517300061.

[30] L. Guegan, I. Raïs, O. Anshus, (POSTER) towards data dissemination policy prediction for constrained environments using analytics, in: 2023 19th International Conference on Distributed Computing in Smart Systems and the Internet of Things (DCOSS-IoT), IEEE, Pafos, Cyprus, Jun. 2023.

[31] L. Guégan, S. Tofaily, I. Raïs, Design and evaluation of single-board computer based power monitoring for IoT and edge systems, in: The 19th IEEE International Conference on Green Computing and Communications (GreenCom 2023), IEEE, 2023.



Loïc Guégan received his PhD in Computer Science from École Normale Supérieure (Rennes, France) in January 2021. He is an associate professor at "UiT, the Arctic University of Norway", since 2024. He is part of the Cyber-Physical System (CPS) group. His research interests are energy efficiency, distributed systems, network simulation, IoT, cloud and edge systems.



Issam Raïs received his PhD in Computer Science from École NormaleSupérieure (Lyon, France) in September 2018. He is an associate professor at "UiT, the Arctic University of Norway", since 2021. He leads the "Cyber Physical System (CPS)" group, since 2023. His focus includes energy efficiency, systems, distributed systems, high performance computing, cloud, IoT and edge computing. He has co-authored more than 30 articles published in peer-reviewed journals and conferences.

Otto Anshus is a Professor at the Department of Computer Science, University of Tromsø (UiT) The Arctic University of Norway, and a part-time Professor at the Department of Informatics, University of Oslo, Norway. His current primary research interests are distributed and parallel Cyber- Physical Systems.