IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received September 19, 2018, accepted November 2, 2018, date of publication November 13, 2018,
date of current version December 18, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2881088

Enabling the Internet of Arctic Things With Freely-Drifting Small-Satellite Swarms

DAVID PALMA[1] AND ROGER BIRKELAND[2]
[1]Department of Information Security and Communication Technology, Norwegian University of Science and Technology, 7491 Trondheim, Norway
[2]Department of Electronic Systems, Norwegian University of Science and Technology, 7491 Trondheim, Norway

Corresponding author: David Palma (david.palma@ntnu.no)

The work of D. Palma was supported by the European Union's Horizon 2020 Research and Innovation Programme through the Marie Skłodowska-Curie under Grant 699924, SINet.

ABSTRACT The widespread deployment of Internet-capable devices, also known as the Internet of Things (IoT), reaches even the most remote areas of the planet, including the Arctic. However, and despite the vast scientific and economic interest in this area, communication infrastructures are scarce. Nowadays, existing options rely on solutions such as Iridium, which can be limited and too costly. This paper proposes and evaluates an alternative to existing solutions, using small satellites deployed as a freely-drifting swarm. By combining these simpler and more affordable satellites with standard protocols, we show how IoT can be supported in the Arctic. Networking protocols and link characteristics are emulated for 3 different satellite orbits and 4 ground nodes. The impact of different protocols and communication conditions is assessed over a period of 49 days and a cross-layer routing approach proposed. The obtained results reveal that a communication overhead bellow 27 % can be achieved and that the implemented satellite-aware route selection allows reducing the end-to-end time of a request up to 93 min on average. This confirms that freely-drifting small-satellite swarms may enable the Internet of Things even in the most remote areas.

INDEX TERMS Satellite communication, Internet of Things, software defined networking, Arctic, small satellites, swarm.

I. INTRODUCTION

Activity in the Arctic region is increasing [1], [2] and several bodies such as the European Union (EU), NASA and the Arctic Council expect this to continue [3]. Activities range across fishing, mining, shipping and securing environmental situation awareness. However, due to the lack of land-based infrastructures and satellite coverage (e.g. satellites in a Geostationary Earth Orbit (GEO) are not reachable north of 81° latitude), information and communication technologies for supporting these activities are scarce.

A project by the European Space Agency (ESA), entitled ArticCOM [4], concluded that there is a communication gap in this area and listed future projects expected to cover parts of non-European Arctic. However, several of these have been cancelled or delayed. This report further acknowledges that no planned systems for the European Arctic existed. Nonetheless, this has changed with projects such as the Norwegian Highly Elliptical Orbit (HEO) initiative, the Canadian Telesat, proposed mega constellations from SpaceX (StarLink) and OneWeb, aiming at providing worldwide broad-band coverage, including in the Arctic.

Science missions currently rely on costly systems such as Iridium [5] or manned missions for collecting nodes and retrieving their data. Alternatively, a hierarchical network could be used with different levels of communication between sensor nodes, unmanned vehicles (UVs) or small satellite nodes [6]. Small Satellites, or smallsats, can be deployed in freely-drifting swarms, which do not require thrusters and allow increasing communication coverage at a reduced cost when compared against traditional satellites [7], [8]. In this work, a freely-drifting swarm is defined as a set of such smallsats in one system, without any station keeping capabilities. This constrasts with a fixed constellation or a swarm of satellites where member satellites perform station keeping and may interact with each-other.

Small satellite constellations are currently under development, however they require more costly equipment and mostly aim at covering densely populated regions of the Earth (e.g. Sky and Space Global [9]). Another limitation is the envisaged throughput of only a few kilobytes per day (e.g. Astrocast [10] and HeliosWire [11]).

The irregular presence of vehicles and the intermittent nature of satellite links requires a robust and flexible IoT setup. This motivated several works to focus on the principles of Delay Tolerant Networking (DTN) [12], [13] or even on the combination of IoT with DTN protocols. For example, the Constrained Application Protocol (CoAP) [14], due to its suitability for IoT constrained nodes, has been combined with the Bundle Protocol (BP) [15] in order to support intermittent connectivity. Moreover, this heterogeneity demands a convergence layer for enabling seamless interoperability between distinct communication technologies.

A cornerstone of the Internet is the Internet Protocol (IP), currently on version 6 (IPv6) [16], which provides a way of identifying nodes and allows data to be sent and received across different networks. IPv6 can be seen as the required convergence layer between different technologies, including satellite-based communications [17], providing seamless interoperability with existing systems. Additionally, the IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN) has been considered as an appropriate solution for constrained link-layers [18], as expected in remote locations due to the lack of resources and infrastructures.

This paper takes into account communication needs in the Arctic and evaluates the feasibility of a network solution supported by a smallsat drifting swarm. Specifically, the following contributions are provided:

- Emulation of a smallsat IoT network for the Arctic, combining both IPv6 and 6LoWPAN with CoAP;
- Analysis of the different phases of a 3-satellite freely-drifting swarm and the impact on communications;
- Proposal of a satellite-aware routing approach.

The Internet of Arctic Things (IoAT) is presented in Section II, introducing the envisaged architecture, explaining the inner-workings of a freely-drifting swarm and detailing the proposed network solution. Section III presents the defined evaluation methodology followed by the obtained performance results in Section IV. Finally, Section V provides an overview of the main conclusions of this work.

II. INTERNET OF ARCTIC THINGS WITH SMALLSATS

Smallsats stand out from larger satellites by dint of their simplicity and low-cost design. Multiple satellites can be deployed so that they form a distributed system [19]–[21]. The satellites can be deployed either as a constellation, which implies the use of more sophisticated and expensive platforms with propulsion or drag management for station keeping, or as a freely-drifting swarm using simpler platforms.

By giving the satellites a varying spring load in the deployment pod, they will in turn be deployed with a small velocity difference relative to each other. This results in a varying distribution of the satellites within the swarm throughout time, from now one referred to as swarm phases, leading to variable network coverage and performance. The following subsections discuss these aspects and a possible architecture.

A. ARCHITECTURE OVERVIEW

The Internet of Arctic Things networking proposal presented in this work considers 3 distinct types of nodes:

- **Ground Station (GS):** A gateway to traditional Internet services, located at higher latitudes (e.g. Vardø, Norway);
- **Border Router (BR):** A smallsat acting as relay node or data-mule between a GS and a Sensor Node (SN);
- **Sensor Node (SN):** A resource-constrained Internet-capable device collecting data in the Arctic region.

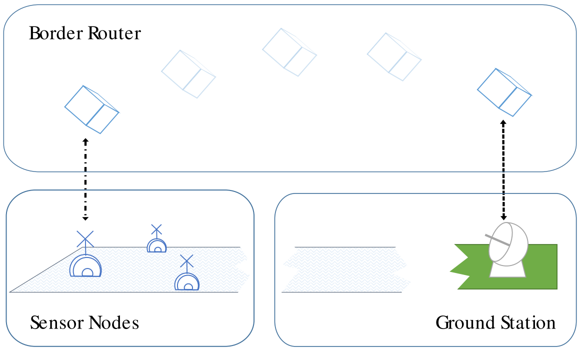


Figure 1. System architecture with different link characteristics between satellites and ground nodes.

Figure 1 shows the system architecture, and how the different network nodes interacts. In order to reduce their complexity, smallsats are not expected to communicate amongst themselves. However, if UVs are to be included, they may also be considered BRs and communicate with smallsats. This could occur when a UV is not in range with a GS and relies on smallsats to act as BRs, even though this is outside the scope of this paper.

Despite being resource-constrained devices, SNs may communicate with other SNs and benefit from data aggregation mechanism to reduce overhead. This is particularly important as the number of SNs increases, further motivating the use of standardised IP-based protocols in order to guarantee interoperability and access to other existing features (e.g. encryption).

B. FREELY-DRIFTING SMALLSATS SWARM

In addition to the lower cost of smallsats when compared to other larger Geostationary Earth Orbit (GEO) satellites, they use a Low Earth Orbit (LEO) which is advantageous when considering low-power communications. This is a direct consequence of the distance between a ground node and a satellite, which can be ten times shorter than when considering GEO satellites. Resorting to GEO satellites would incur much larger propagation losses and delays, requiring higher transmission power and larger antennas. Moreover, GEO satellites are not reachable by ground nodes north of 81° latitude.

The cost of a smallsat node can be kept low due to their simple design and use of commercial-off-the-shelf (COTS) components. In addition, due to their small form-factor smallsats can be launched as a secondary payload to other missions. This allows a "ride-sharing" approach with larger commercial missions, therefore avoiding dedicated launches and further reducing the overall cost of the solution [17]. A limitation of this approach is that the smallsat mission does not control the final orbital parameters, save for choosing which launch to book a ride on. The mission designer must then choose a fitting orbit for the given satellite mission. For example, polar orbits are often used for earth observation missions and by definition they cover the Arctic areas, being therefore suited this kind of mission.

In this paper we consider 3 smallsats deployed from the same upper stage on a common launch, which places them in the same orbital plane. Even without exactly-timed deployments or thrusters, this freely-drifting swarm is able to achieve the same performance as a constellation composed by 2 smallsats [7]. Satellites without station keeping capabilities will be smaller and cheaper, operational requirements become more relaxed and the cost savings can be used to launch more space crafts. Deployment strategies and how to choose reasonable and realistic velocity differences are discussed in [1], [7], [21], and [22].

By giving the 3 satellites small velocity differences, they enter slightly different orbits, with distinct orbital periods. Due to this, the smallsats will drift relatively to each other, resulting in a freely-drifting swarm with varying and evolving phases, as seen in Figure 2. Hence, the properties of a network supported such swarm will constantly change.

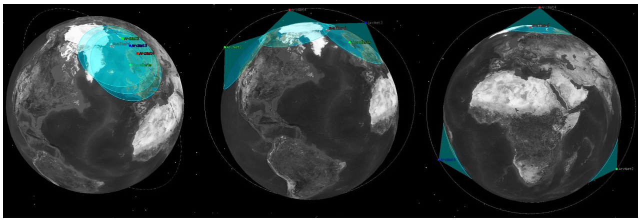


Figure 2. **Swarm phases: *overlapping satellites* (left), *trailing* (centre) and *uniform distribution* (right).**

The considered freely-drifting swarm with 3 smallsats will transition through 3 different phases as illustrated by Figure 2. The first phase corresponds to 3 *overlapping satellites*, or a variation where only 2 satellites overlap with the third being diametrically opposite to them. Another phase assumes a *trailing* or scattering configuration, where the satellites either diverge or converge towards one of the other phases. Finally, a *uniform distribution* of the satellites around the planet.

Bearing in mind the different possible phases, the best possible coverage with respect to the re-visit time is achieved with a *uniform distribution*, resulting in comparable gaps between each smallsat pass. On the other hand, for the *trailing* phase, short re-visit intervals are followed by a larger one, while for *overlapping satellites* the interval between passes is the greatest, resulting in large periods without coverage. Nevertheless, when considering the total coverage time for a node placed north of Svalbard (KSAT – 78°13′48.0″N 15°23′24.0″E), our simulations show that *overlapping satellites* only account for 2.8 % of the covered time and that the 2 satellites overlap phase only occurs 10.6 % of the time. It can therefore be argued that the penalty of letting the swarm drift freely is quite small, considering that a satellite with orbit-manoeuvre capabilities (i.e. thrusters) will have a significantly higher cost.

In addition to the variability added by changing phases, it is also important to consider the dynamics between Earth's rotation and the satellites' orbital plane (c.f. Section III-A). The orbital plane is inclined with respect to Earth's rotational axis and Earth rotates within it. This means that the satellites will not pass directly over-head of a given ground node in every orbit revolution. In fact, the satellites' ground track will move along the surface of Earth in each pass, therefore affecting the duration of each pass throughout a day, being less noticeable by nodes at higher latitudes since they are closer to the axis of rotation. For example, ground nodes placed as far north as mainland Svalbard observe all the passes from a polar orbiting satellite, while nodes further south miss some passes in a day.

An IoT network in the Arctic must take into account the dynamics between orbital planes and Earth's rotation, as well as the characteristics of the described freely-drifting swarm, specially when transmitting data between multiple ground nodes that may exist. Since all satellites are capable of eventually reaching all these nodes, a naive networking approach would select the first arriving satellite as a next-hop. However, the desired destination may not be aligned with this satellite's ground track at all given times. Alternatively, selecting a later arriving satellite with a more suitable ground track (i.e. closer to the destination), may provide a shorter delay between the source and destination nodes.

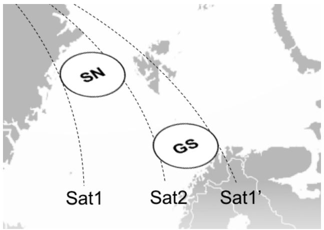


Figure 3. **Ground tracks for *Sat*1 and *Sat*2, with $t_{Sat1} < t_{Sat2} < t_{Sat1'}$ (background adapted from Wikimedia Commons / CC BY 3.0).**

This is illustrated by Figure 3, where *Sat1* becomes visible to the sensor node *SN* before *Sat2*, but that requires one more orbit revolution until its ground track (*Sat*1′) is aligned with the ground station *GS*. Conversely, while *Sat2* only becomes visible to *SN* later, due to its better track alignment, it is also visible to *GS* in the same revolution. This allows significantly reducing the required time for a message to be relayed from SN to GS.

C. NETWORKING AND COMMUNICATION

Diversity in the IoT increases not only the number of networking possibilities, but also the number of challenges and requirements to be met, such as interoperability. Focusing on the Arctic and maritime operations, different activities may require monitoring of simple weather parameters (e.g. temperature, wind speed), or highly complex data (e.g. hyperspectral images). This leads to several heterogeneous nodes and communication technologies being found in such scenarios [23].

The use of standardised Internet protocols is the best way of guaranteeing interoperability between different nodes and technologies. We rely on IPv6 addressing and on its lightweight version of 6LoWPAN to support this. In particular, we consider the use of full IPv6 addresses for communication technologies and nodes with higher availability of resources, such as the links between GSs and satellites, which will typically have more energy and higher-gain antennas than sensor nodes. Even though 6LoWPAN was developed in the context of IEEE 802.15.4 [24], it has also been considered in the context of other communication technologies [18]. Using it for constrained satellite and sensor-node links would allow benefiting from the existing adaptation layer [25] and compression mechanisms [26], reducing networking overhead.

In order to support other communication links that may exist, even between the same BR and SN, a Software-defined Networking (SDN) solution was used on the satellites. By adding or removing flow rules issued by the Ground Station, our nodes are capable of dynamically changing an IPv6 address into a 6LoWPAN one, from global to unique link-local addresses and by selecting the corresponding network interface. This allows not only the change between communication technologies but also to the establishment of priority between flows, among other features.

Typical satellite-based networking solutions select DTN routing protocols to solve the issue of intermittent connectivity and rely on opportunistic or predictable establishment of communication links (e.g. PRoPHET [27]). However, these solutions typically introduce abstractions such as an overlay of links and networks resulting from the Bundle Protocol or Convergence Layers [28], not considering the specifics of the domain in question and resulting in unnecessary overhead. In our SDN-based approach, routing overhead between nodes is prevented by having routes defined by the GS or by a local controller, which can automatically be added or removed when appropriate. Moreover, these can be updated if new nodes are deployed or if any changes are deemed necessary.

In the Internet many applications follow a client/server representational state transfer (REST) architectural style. Similarly, in IoT and with constrained devices in mind, CoAP was designed to be RESTful while also keeping overhead to a minimum. CoAP messages require a header of only 4 B [14] and the User Datagram Protocol (UDP) is used instead of the Transmission Control Protocol (TCP), with additional mechanisms such as confirmable messages being optional but also possible. The design of CoAP was also conceived so that seamless interoperability with other Internet services could be provided. In particular, CoAP defines the concept of CoAP proxy, where a node can be used to forward request/responses or even to convert Hypertext Transfer Protocol (HTTP) requests into lightweight CoAP messages and vice-versa.

By using CoAP as an application layer protocol responsible for handling data transfers, GSs or SNs can issue requests to any node in the network, specifying BRs as proxy nodes. However, CoAP proxies were not designed to support proxying as typically found in satellite nodes, which can act as DTN-capable nodes. This has already been addressed by previous works in the literature [15] and can be achieved by slightly adapting the protocol without breaking its compatibility with standard implementations (c.f. Section III-B).

Another important networking aspect concerns the selection of the most appropriate next-hop. The simplest approach consists of selecting the first BR available, especially since we consider that each BR is capable of reaching all SNs. If more than one BR is available at a given instant, the typical approach would be to select the one with the lowest hop count to the destination, but they are all equal.

Regardless of having one or more BRs available, as discussed in the previous sub-section, a naive approach can lead to selecting a BR out of alignment with the desired destination node. Bearing that in mind, we propose a smarter approach where, depending on the source and destination ground nodes, where the fastest satellite to reach them both is selected. This exploits knowledge about the domain, namely available satellite orbits and nodes' positions, and can either be pre-calculated or periodically updated by making use of the proposed SDN routing approach.

III. METHODOLOGY

The proposed Internet of Arctic Things architecture, supported by a freely-drifting swarm of smallsats, is depicted in Figure 1. In order to realistically evaluate its feasibility, a combination of simulation and emulation techniques was used. The dynamics of the swarm were simulated, serving as input for the network emulator that ran real networking protocols over emulated links created and destroyed according to the BRs' coverage of each node.

The scope of this paper is to evaluate higher-layer network protocols. Therefore, lower-layer protocols for the radio links and media access strategies are not included. In this setup, the BR is polling each node, hence controlling most of the access to frequency resources. Nonetheless, a suitable media access scheme should be implemented.

The SNs are assumed to be able to run a light-weight operating system (e.g. Linux or Contiki), enabling the use of the IoT network stack. In addition, positioning device shall be available providing also accurate timing. With these capabilities and the knowledge about the satellite's orbits, which can be regularly updated, the SNs can calculate the start time of a pass by use of an orbit propagator. This approach will have a negligible communication and processing cost and allow nodes to save energy by turning off the radio until the desired satellite pass is near.

A. FREELY-DRIFTING SWARM SIMULATION

The evaluated freely-drifting swarm was based on realistic satellite orbits from the Two-Line-Element (TLE) [29] set of AAUSat-3 [30], with the epoch *13 Feb 2014 12:35:42.657*. The used TLE was retrieved from the Systems Toolkit (STK) [31], and each of the 3 defined satellites had its *orbits-per-day* and eccentricity e parameters edited accordingly. An inclination of 98.6235° and a perigee height of 768 km was set to all of them. Their apogee altitudes were 771.83, 787.17 and 802.55 km. These slightly different orbits are responsible for the previously mentioned drift that results in different phases (c.f. Figure 2). For the chosen orbits, one "full cycle" of all the possible satellite phases, from which the same pattern is repeated, lasts for approximately 45 days.

The simulation of the chosen swarm depends on the selected ground nodes, for which a singular coverage perspective must be determined. Focusing on a realistic scenario in the Arctic region, the positioning of the GS chosen for this paper was Vardø, Norway, where one of northernmost mainland ground stations is currently in use. Three other ground nodes were selected, 3 SNs named GR_north, GR_south and Rossøya. Their locations, as seen in Figure 4, were based on a previous research work also addressing the Arctic region [32].

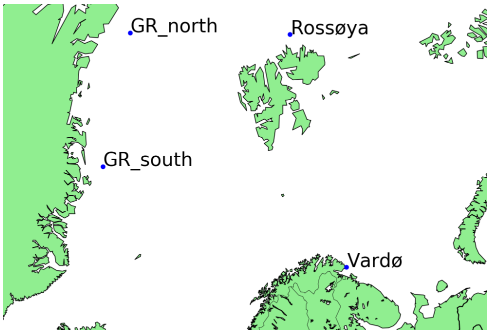


Figure 4. Placement of ground nodes in the Arctic region. Sensor Nodes: GR_north (80°52′39.0″N 11°59′53.0″W), GR_south (75°51′0.0″N 16°52′15.5″W) and Rossøya (80°49′40.8″N 20°21′0.0″E). Ground Station: Vardø (70°22′12.0″N 31°6′36.0″E).

B. NETWORK EMULATION

The evaluation of the overall networking performance was conducted through emulation, using the simulation details between each satellite and ground node as input for configuring each link. These details concern mostly the delay of each link and its availability, taking into account their ground track and distance to the ground node. The bitrate for links between the GS and BRs was set to 1 Mbit s^{-1}, based on available COTS S-band radios, while for links between BRs and SNs it was set to 20 kbit s^{-1} also based on COTS components with such bitrates already available.[1]

The used emulation tool [33], in addition to the used *qdiscs*, was adapted to mimic the constrained nature of satellite links by using network interfaces based on Linux's *nl802154* physical layer. This means that in addition to controlling the bitrate and delay of each link, the link between BRs and SNs was also limited to a maximum transmission unit of 127 B, fully integrating the links with 6LoWPAN. The entire networking stack was emulated using Ubuntu 16.04 (Linux Kernel 4.14.15-1) containers for each node, using dedicated *network namespaces* for isolating traffic between links.

Network performance was evaluated considering the overhead of the used protocols and from the *user's perspective*. The latter consists of the end-to-end response time, from the instant when a request is issued until its response is received. For this purpose, NON-confirmable CoAP requests were randomly created, following a uniform distribution between 60 s and 180 s. The destination for each request was also selected following a random uniform distribution, so that all SNs were equally used. Finally, a constant payload of 512 B per response was used, based on IoT networking where periodic small-size data transfers are expected. Nonetheless, it is worth noting that several requests and responses may be queued between satellite passes, resulting in data bursts when a new link becomes available.

The chosen CoAP implementation was CoAPthon [34], modified to support the queuing of CoAP messages whenever no route is available. This behaviour allows the support of intermittent connectivity without relying on any additional messages or overhead. Instead, an event-triggered approach was used, resorting to *IPDB-callbacks*[2] for new routes available in the system. This allows CoAP to be completely decoupled from any routing mechanisms being used.

C. SATELLITE-AWARE ROUTING

Regarding the selection of the most appropriate BR as next-hop, Figure 5 shows our satellites' ground track relatively to the used ground nodes. It consists of a combined snapshot of a relay opportunity for two of the satellites (ArcNet1 and ArcNet9) which the first satellite does not observe (ArcNet5). This illustrates one instance when the benefit of smart routing can improve the network performance, considering a request issued from Vardø to GR_north. Specifically, on this occasion, ArcNet5 is the first smallsat to reach the GS at Vardø after several hours without coverage.

A naive approach would select ArcNet5 node as a next-hop since no others would be available at that instant. However, this BR requires one more orbital revolution in order to reach GR_north and complete the communication. Instead, by either waiting for ArcNet1 or ArcNet9 to become visible, 30 min later, requests can be relayed directly to GR_north, reducing the end-to-end time in nearly 60 min.

---

[1] [Online]. For example: `gomspace.com/Shop/subsystems/communication/nanocom-ax100.aspx`

[2] [Online]. From pyroute2 netlink library. Available: `docs.pyroute2.org`

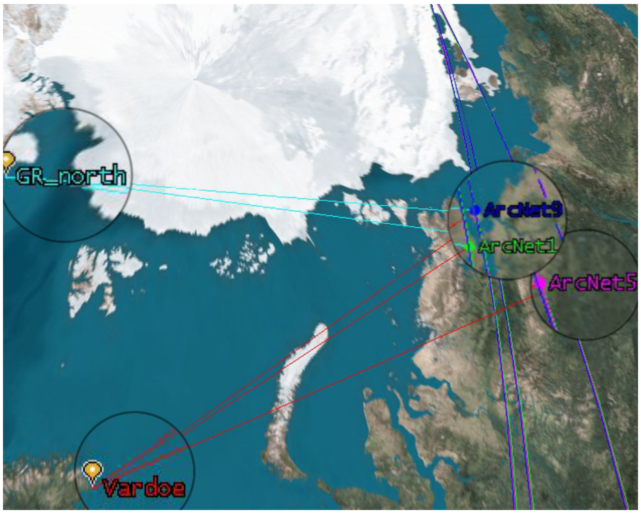


Figure 5. Example of different ground tracks and coverage: ArcNet5 is the first reaching Vardø but fails to reach GR_north. ArcNet1 and ArcNet9 reach Vardø later but act as relays for GR_north, reaching it simultaneously.

As previously mentioned, the defined network setup allows next-hop selection based on different methods. In addition to the naive approach, routes can be set by the GS or, alternatively, by a local controller in each node that determines the best next-hop based on the desired destination and current time. The *Smart*, or satellite-aware, routing method was implemented using the light-weight *pyephem* library [35], which allows the calculation of upcoming passes for a given node.

The implemented algorithm used the total end-to-end time (i.e. waiting time for a BR in each node) as its main metric, with the best next-hop minimising this value. However, additional path constraints were added, taking into account propagation and processing delays and the duration of each satellite pass. In particular, since some satellite passes may exist but be extremely short, a minimum threshold should be set in order to avoid selecting inadequate paths. In the performed evaluation, 3 flavours of smart-routing were used, *Smart5*, *Smart15* and *Smart30* respectively, with thresholds of 5, 15 and 30 s.

IV. PERFORMANCE EVALUATION

In this section we present the results obtained from emulating and simulating the described network architecture and its respective smallsat swarm. The experiment period was of 49 days, covering more than a "full cycle" of all the possible satellite phases, starting with a *trailing* phase until it returns to its initial state. This resulted in more than 32000 CoAP requests being transmitted through the evaluated network.

A. OVERHEAD

The low communication and computational impact of the chosen IoT protocols, designed to operate in resource-constrained nodes, was one of the main considerations in the proposed Internet of Arctic Things. Table 1 presents the overhead registered in the performed experiments, both for full IPv6 addresses and 6LoWPAN compressed (i.e. 16 bit) addresses. Specifically, these results correspond to the links between the GS and BRs (full IPv6) and between BRs and SNs (6LoWPAN).

TABLE 1. Overhead.

	Full IPv6	6LoWPAN
Ethernet/15.4 (L2) (%)	5.803	8.661
IPv6/6Lo (L3) (%)	16.58	10.586
ICMPv6 (L3) (%)	0.95	4.73
UDP (L4) (%)	3.316	0
CoAP (L7) (%)	7.227	2.676
Total Overhead (%)	33.876	26.653

As expected, the total overhead introduced by using full IPv6 addresses is higher than with 6LoWPAN. For example, due to the used compression mechanisms, 6LoWPAN eliminates UDP overhead by including it in its headers. However, when carefully analysing the sources of overhead for each, some noteworthy results were registered. For example, the percentage of transmitted ICMPv6 messages in 6LoWPAN is more than 4 times greater than IPv6.

By analysing all the captured traffic it was found that this resulted from a characteristic of the *nl802154* driver, which is not namespace-aware and until recently did not support knowledge about connected edges.[3] This resulted in *Neighbor Solicitation and Advertisement* messages being received by multiple nodes simultaneously, even if no link existed. Therefore, in a real scenario this overhead would be lower. Finally, since CoAP requires an extra field for specifying the desired proxy address, the overhead in the link between the GS and BRs was higher.

TABLE 2. Performance comparison – overall.

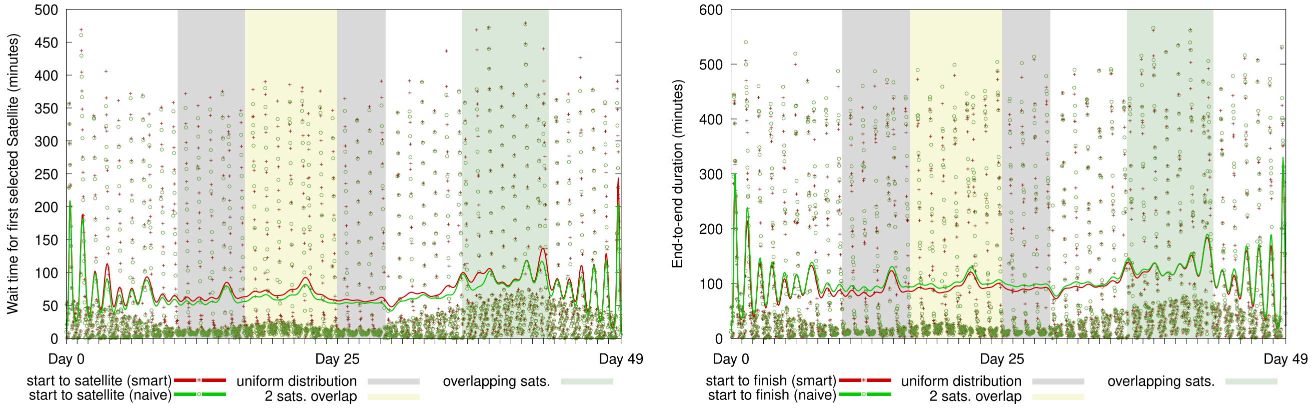
	Naive	Smart5	Smart15	Smart30
First Satellite (s)	4031	4349	4381	4427
End to end (s)	5853	5060	5113	5401
Same Choice (%)	–	85.7	85.6	85.2
Improvement (s)	–	5636	5381	4281
Total Losses (%)	2.8	5.2	5.0	3.3

B. OVERALL PERFORMANCE

The overall performance of the evaluated experiment is summarised in Table 2, comparing the average end-to-end time for all the created requests and verifying that a low-percentage of losses can be achieved, even without using CoAP confirmable requests. The obtained results also

---

[3] [Online]. Available: `https://patchwork.kernel.org/patch/10369859/`



(a) (b)

Figure 6. **Performance of *Naive vs. Smart30* routing approaches. (a) Time taken for a GS request to reach the selected BR. (b) Time taken from request to received reply at the GS.**

validate the claim for the need to employ satellite-aware routing mechanisms in nearly 15 % of the routing decisions. By analysing the row *Improvement* it is possible to see that the end-to-end time to retrieve data from a sensor node can be, on average, reduced up to 93 min.

Since a real networking stack was used, unpredictable behaviours due to congestion or delays led to the *Naive* approach being better for some requests. However, these correspond to less than 3 % of the requests and should be considered as outliers and are ignored in this evaluation.

After a careful analysis the registered outliers, they were attributed to a concurrency issue in the used software implementation of CoAP, resulting in a request to miss the expected pass and therefore taking an incorrect route. Moreover, this behaviour of the network stack is confirmed by the number of increasing losses in the less restrictive routing approach (*Smart5*), where selecting a short-lived pass results in some messages timing-out, and not necessarily being lost.

Figure 6 presents a comparison between the *Naive* and the *Smart30* routing approaches, using the green and red colours respectively. Each emulated request corresponds to a point in the plot (i.e. a circle or a cross depending on the used routing approach). Their interpolation is represented by a continuous line of the same colour of the corresponding routing approach.

Figure 6 further illustrates the impact of the different satellite phases. For example, it is possible to see that a majority of the requests takes less time when the swarm follows a *uniform distribution*. This value increases the most with *overlapping satellites* and is subject to higher variation when the swarm has a *trailing* phase (white background).

More concretely, Figure 6a shows the time taken since creating a request at the GS until it reaches the selected BR (i.e. smallsat). It shows that *Smart* routing generally takes longer to communicate with the desired BR, depicted by the red line slightly above the green one, confirming the results presented in Table 2. However, this is justified by the selection of BRs that are better aligned with the final destination.

Conversely, Figure 6b confirms the resulting improvement from this selection. It shows that the total amount of time from the request being issued until the response is received (end-to-end time) is lower for the *Smart* approach, outperforming the *Naive*. This is seen in the green line constantly being above the red one.

C. GEOGRAPHICAL IMPACT

As previously described, the location of a ground node influences the perceived satellite coverage. However, resorting to a *Smart* routing approach significantly improves performance, regardless of the nodes' positions. This means that the penalty of using a *Naive* approach should also be analysed for each different destination.

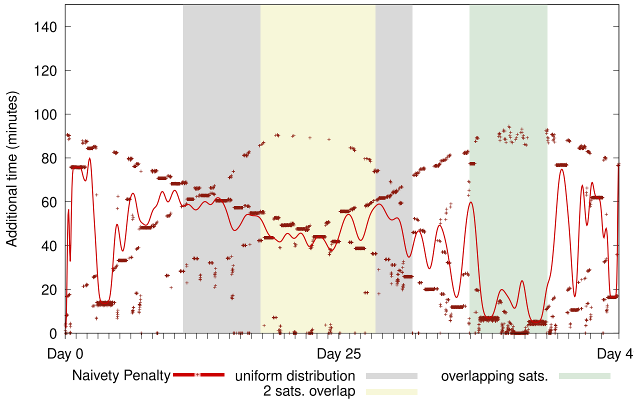


Figure 7. **Overall time penalty of *Naive* routing.**

In order to better visualise the negative impact of using a *Naive* routing approach, Figure 7 to Figure 10 illustrate the time penalty from selecting the first available BR. This penalty is determined by calculating the end-to-end time difference between the *Naive* and *Smart* approaches, per request.

The figure combines the requests issued for each of 3 destinations previously presented. It does however not include any requests where both routing approaches selected the same BR or outliers by unpredictable network behaviours.

The analysis of Figure 7 reveals that the penalty of the *Naive* approach is higher when the swarm is found in a *trailing* phase, due to the higher scattering of satellites. On the other hand, with *overlapping satellites* this penalty is less significant because the satellites' ground track is similar and fewer alternatives exist. Nonetheless, the *Naive* approach is penalised in almost all instances.

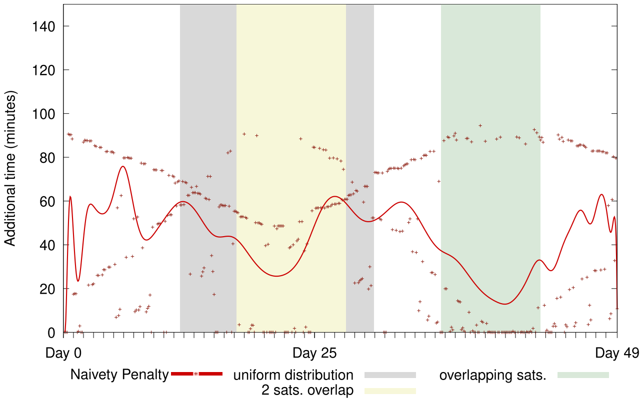


Figure 8. Time penalty of *Naive* routing at Rossøya.

Figure 8 shows the penalty when the selected destination node is located in Rossøya. Since this SN is fairly aligned with Vardø, whenever a satellite's ground track covers the GS it is also likely to reach Rossøya. In the performed evaluation, only 5 % of all the requests to this destination benefited from *Smart* routing. Nevertheless, their end-to-end delay was significantly reduced, with some improvements reaching nearly 100 min.

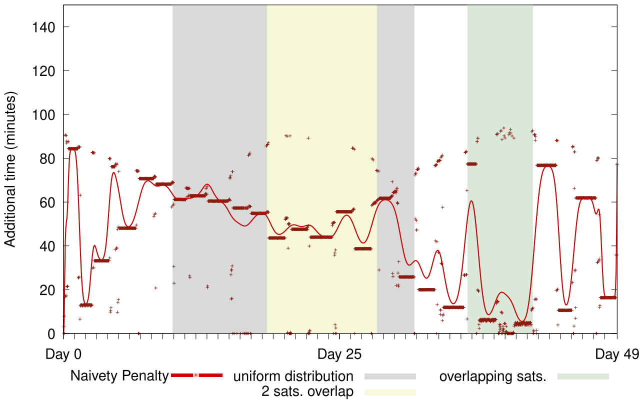


Figure 9. Time penalty of *Naive* routing at GR_north.

In different circumstances, GR_north is the farthest sensor node from the GS, leading to a higher misalignment. This results in more requests being penalised when selecting the first available BR, as seen in Figure 9. Approximately 27 % of the requests to GR_north (i.e. 9 % of the total number of requests), are negatively impacted by this.

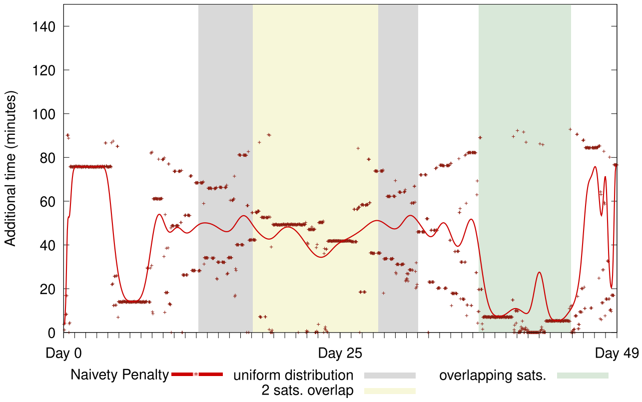


Figure 10. Time penalty of *Naive* routing at GR_south.

Finally, Figure 10 shows the impact of *Naive* routing for requests to GR_south. The number of affected requests is lower than in GR_north, with only 13 % of penalised requests. Despite this, both locations share similarities in how they are affected by different swarm phases.

V. CONCLUSION

In this paper, the concept of the Internet of Arctic Things was introduced, demonstrating how a freely drifting swarm of small satellites can be used for supporting communications in the Arctic. The different phases that such a satellite swarm can assume were analysed, as well as their impact on communications. These satellites do not require thrusters for station keeping, meaning that they are simpler and cheaper than solutions using a uniform constellation during the mission lifetime. This allows for launching more satellites within the same budget, adding redundancy to the system.

An experimental assessment was conducted, emulating real IoT-protocol implementations combined with 3 simulated satellite orbits and 4 ground nodes deployed in the Arctic. These protocols were chosen for their low computation and communication overhead. In particular, IPv6 and 6LoWPAN were used together with CoAP as the basis of the defined networking architecture.

The obtained results indicate that a low number of lost requests/replies can be achieved ($< 5\,\%$), while keeping overhead as low as 27 % when using CoAP with non-confirmable messages. This confirms that low-cost smallsats can effectively be used to provide coverage for different locations in the Arctic using COTS communication technologies and standardised networking protocols. Moreover, this can be seen as a dedicated and affordable alternative to commercial satellite solutions.

A new satellite-aware cross-layer routing approach was also tested, revealing an improvement of the end-to-end time of request, up to 93 min less, when compared against a routing approach based on simple hop-count metrics. This result confirms that future networking solutions may be *smarter*, benefiting from upper-layer knowledge about satellites' ground tracks in relation to the ground nodes' positions.

ACKNOWLEDGMENT

The authors like to thank the CAMOS project (Coastal and Arctic Maritime Operations and Surveillance) – IE, NTNU.

REFERENCES

[1] R. Birkeland and A. Hornig, "On how a cubesat swarm can improve the coverage for an arctic ground based sensor network," in *Proc. ESA Small Satell. Syst. Services*, 2016, pp. 1–13.

[2] R. Birkeland, "Proposal for a satellite communication payload for sensor networks in the arctic area," in *Proc. 65th Int. Astron. Congr.*, 2014, pp. 6862–6865.

[3] Arctic Council. (2017). *Telecommunications Infrastructure in the Arctic*. [Online]. Available: https://oaarchive.arctic-council.org/bitstream/handle/11374/1924/2017-04-28-ACS_Telecoms_REPORT_WEB-2.pdf

[4] ESA. (2011). *ESA Artes Articom*. Accessed: Apr. 2018. [Online]. Available: https://artes.esa.int/projects/arcticom

[5] Iridium Communications Inc. (2013). *Iridium Next: Advanced Technology. Iridium NEXT: Advanced Technology*. [Online]. Available: http://www.iridium.com/About/IridiumNEXT/Technology.aspx

[6] R. Birkeland, D. Palma, and A. Zolich, "Integrated smallsats and unmanned vehicles for networking in remote locations," in *Proc. 68th Int. Astron. Congr.*, 2017, pp. 6288–6295.

[7] R. Birkeland, "Freely drifting cubesat constellations for improving coverage for arctic sensor networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017, pp. 1–6.

[8] A. G. C. Guerra, F. Francisco, J. Villate, F. A. Agelet, O. Bertolami, and K. Rajan, "On small satellites for oceanography: A survey," *Acta Astronautica*, vol. 127, pp. 404–423, Oct./Nov. 2016.

[9] (2018). Sky and Space Global. [Online]. Available: https://www.skyandspace.global/

[10] (2018). Astrocast—The Nanosatellite IoT network. [Online]. Available: https://www.astrocast.com/

[11] (2018). *HeliosWire*. [Online]. Available: https://helioswire.com/

[12] L. Lambrinos, C. Djouvas, and C. Chrysostomou, "Applying delay tolerant networking routing algorithms in maritime communications," in *Proc. IEEE 14th Int. Symp. World Wireless, Mobile Multimedia Netw. (WoWMoM)*, Jun. 2013, pp. 1–6.

[13] L. Torgerson *et al.*, *Delay-Tolerant Networking Architecture*, document RFC 4838, Oct. 2015. [Online]. Available: https://rfc-editor.org/rfc/rfc4838.txt

[14] C. Bormann, K. Hartke, and Z. Shelby, *The Constrained Application Protocol (CoAP)*, document RFC 7252, Jun. 2014. [Online]. Available: https://rfc-editor.org/rfc/rfc7252.txt

[15] M. Auzias, Y. Mahéo, and F. Raimbault, "CoAP over BP for a delay-tolerant Internet of Things," in *Proc. 3rd Int. Conf. Future Internet Things Cloud*, Aug. 2015, pp. 118–123.

[16] S. Deering and R. Hinden, *Internet Protocol, Version 6 (IPv6) Specification*, document RFC 2460, Dec. 1998. [Online]. Available: https://rfc-editor.org/rfc/rfc2460.txt

[17] M. de Sanctis, E. Cianca, G. Araniti, I. Bisio, and R. Prasad, "Satellite communications supporting Internet of remote things," *IEEE Internet Things J.*, vol. 3, no. 1, pp. 113–123, Feb. 2016.

[18] Y.-G. Hong, C. Gomez, A. R. Sangi, T. Aanstoot, and S. Chakrabarti. (Mar. 2018). IPv6 Over Constrained Node Networks (6lo) Applicability & Use Cases, Internet Engineering Task Force, Internet-Draft draft-ietf-6louse-cases-04. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-ietf-6lo-use-cases-04

[19] G. Liu and S. Zhang, "A survey on formation control of small satellites," *Proc. IEEE*, vol. 106, no. 3, pp. 440–457, Mar. 2018.

[20] M. N. Sweeting, "Modern small satellites-changing the economics of space," *Proc. IEEE*, vol. 106, no. 3, pp. 343–361, Mar. 2018.

[21] N. H. Crisp, K. Smith, and P. Hollingsworth, "Launch and deployment of distributed small satellite systems," *Acta Astronautica*, vol. 114, pp. 65–78, Sep./Oct. 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S009457651500171X

[22] A. Dono, L. Plice, J. Mueting, T. Conn, and M. Ho, "Propulsion trade studies for spacecraft swarm mission design," NASA, Washington, DC, USA, Tech. Rep. ARC-E-DAA-TN51390, 2018. [Online]. Available: https://ntrs.nasa.gov/search.jsp?R=20180001729

[23] A. Zolich *et al.*, "Survey on communication and networks for autonomous marine systems," *J. Intell. Robotic Syst.*, Apr. 2018.

[24] *IEEE Standard for Local and Metropolitan Area Networks—Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)*, IEEE Standard 802.15.4-2011, Sep. 2011, pp. 1–314.

[25] G. Montenegro, J. Hui, D. Culler, and N. Kushalnagar, *Transmission of IPv6 Packets over IEEE 802.15.4 Networks*, document RFC 4944, Oct. 2015. [Online]. Available: https://rfc-editor.org/rfc/rfc4944.txt

[26] P. Thubert and J. Hui, *Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks*, document RFC 6282, Sep. 2011. [Online]. Available: https://rfc-editor.org/rfc/rfc6282.txt

[27] A. Lindgren, A. Doria, E. B. Davies, and S. Grasic, *Probabilistic Routing Protocol for Intermittently Connected Networks*, document RFC 6693, Aug. 2012. [Online]. Available: https://rfc-editor.org/rfc/rfc6693.txt

[28] L. Wood, P. Holliday, D. Floreani, and W. M. Eddy, "Sharing the dream," in *Proc. Int. Conf. Ultra Modern Telecommun. Workshops*, Oct. 2009, pp. 1–2.

[29] D. T. Kelso. (1998). *Frequently Asked Questions: Two-Line Element Set Format*, [Online]. Available: https://celestrak.com/columns/v04n03/

[30] Aalborg University. (Sep. 2014). *AAUSAT3*. [Online]. Available: http://www.space.aau.dk/aausat3/

[31] Analytical Graphics, Inc. (2016). *STK*. Accessed: Apr. 2016. [Online]. Available: http://www.agi.com/products/stk/

[32] R. Birkeland and D. Palma, "Freely drifting small-satellite swarms for sensor networks in the arctic," in *Proc. 3rd Int. Congr. Inf. Commun. Technol.*, 2018, pp. 175–190.

[33] D. Palma, A. Zolich, Y. Jiang, and T. A. Johansen, "Unmanned aerial vehicles as data mules: An experimental assessment," *IEEE Access*, vol. 5, pp. 24716–24726, 2017.

[34] G. Tanganelli, C. Vallati, and E. Mingozzi, "CoAPthon: Easy development of CoAP-based IoT applications with Python," in *Proc. IEEE 2nd World Forum Internet Things (WF-IoT)*, Dec. 2015, pp. 63–68.

[35] B. Rhodes. (2015). *Pyephem Home Page*. Accessed: Oct. 2015. [Online]. Available: http://rhodesmill.org/pyephem/

DAVID PALMA received the Ph.D. degree in information science and technology from the University of Coimbra in 2013. He was an H2020 Marie Sklodowska-Curie Post-Doctoral Fellow with the Norwegian University of Science and Technology (NTNU), Trondheim, and also a Researcher and a Project Manager both in industry and academia. He is currently an Associate Professor with the Department of Information Security and Communication Technology, NTNU. His current research interests are on cognitive IoT, networking in remote areas, routing, cloud-computing and software-defined networks, subjects on which he has authored and co-authored multiple papers in refereed conferences and journals. He has participated in several TPCs as well as in multiple international research projects (FP6/FP7/H2020).

ROGER BIRKELAND received the master's degree in electronics major in satellite communications from the Norwegian University of Science and Technology (NTNU), where he is currently pursuing the Ph.D. degree. He is studying the use of small satellites in future Arctic sensor networks. He has previously worked with embedded systems and underwater communications, before leading a student satellite project at NTNU.

• • •