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

David Palma
Department of Information Security and Communication Technology
NTNU Norwegian University of Science and Technology
Trondheim, Norway
david.palma@ntnu.no

Roger Birkeland
Department of Electronic Systems
NTNU Norwegian University of Science and Technology
Trondheim, Norway
roger.birkeland@ntnu.no

***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 such solutions, using affordable small satellites deployed as a freely-drifting swarm. By combining these simpler and more affordable satellites with IoT protocols, we show how the IoT can be supported in the Arctic. An evaluation through the emulation of 1 ground station, 3 sensor nodes and 3 satellites is presented. 3 different satellite orbits are used, resulting in a dynamic swarm with different layouts, from overlapping to uniformly spaced. The impact of these different conditions on communication is assessed over a period of 48 days using 2 distinct routing approaches. In addition, the overhead for retrieving data from the ground nodes, using IPv6, 6LoWPAN and CoAP is studied. The obtained results reveal that the proposed solution is suitable for supporting Internet communications in the most remote areas and that satellite-aware routing should be considered in such conditions.**

***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. Due to the lack of land-based infrastructures and satellite coverage, the communication infrastructure in the Arctic areas is scarce.

A project form the European Space Agency (ESA), entitled ArticCOM [4], concluded in 2011 that there is a communication gap in this area and listed some projects that were 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 the Norwegian HEO initiative, Canadian Telesat and proposed mega constellations from SpaceX (StarLink) and OneWeb, aiming at providing broad-band coverage to the Arctic.

As satellites placed in a Geostationary Earth Orbit (GEO) are not reachable north of 81° latitude, science missions currently rely on costly systems such as Iridium [5] or even on manned missions for collecting nodes and retrieving their data. Even if broad-band coverage in the Arctic is provided in the future, it may not suit the needs of IoT and sensor networks (e.g. low-energy, small-payload). Alternatively, a hierarchical network could be used with different levels of communication between sensor nodes, unmanned vehicles (UVs) or satellite nodes [6]. Focusing on satellite links, Small Satellites or smallsats, are considered as likely candidates for increasing the communication coverage in remote areas due to their reduced cost [7], [8], while UVs can be used for targeting specific areas and retrieve large amounts of data [9].

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) [10], [11] or even on the combination of IoT with DTN protocols. For example, in addition to the use of IPv6, the Constrained Application Protocol (CoAP) [12], due to its suitability for IoT constrained nodes, has been combined with the Bundle Protocol (BP) [13] 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) [14], providing a way of identifying nodes and allowing 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 [15]. Additionally, the IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN) has been considered as an appropriate solution for constrained link-layers [16], as expected to be found in remote locations.

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 network for the Arctic combining both IPv6 and 6LoWPAN with CoAP;
- Analysis of the different layouts of a 3-satellite swarm and their implications on communications;
- Proposal of a satellite-aware routing approach.

The proposed approach for 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 smallsats can be deployed either as a constellation, which implies the use of more sophisticated and expensive platforms with propulsion, or as a freely-drifting swarm using simpler platforms. This results in different swarm layouts throughout time, providing a 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.

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 satellites, the use of a Low Earth Orbit (LEO) is also 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 Geostationary Earth Orbit (GEO) satellites. Resorting to GEO satellites would incur much larger propagation losses and delays, requiring higher transmission power and larger antennas.

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, their launch and deployment can also significantly contribute to this. Most smallsats are launched as a secondary payload, ride-sharing on commercial launches for larger missions, thus avoiding dedicated launches and further reducing the overall cost of the solution [15]. A limitation of this method is that the smallsat mission does not control the final orbital parameters, save for choosing which launch to book a ride on. Nonetheless, previous works have showed that the performance of a constellation composed by 2 smallsats can be achieved by a freely-drifting swarm of 3 smallsats without needing thrusters or exactly-timed deployments [7].

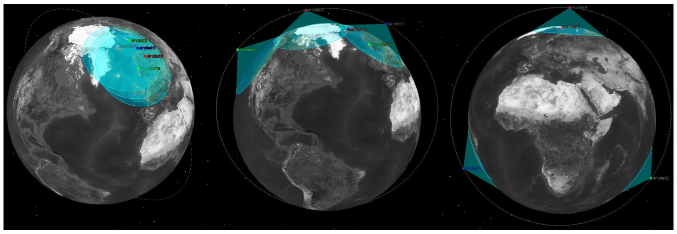

Fig. 1. Swarm Layouts (*overlapping sats.*, *trailing* and *uniform distribution*)

In this paper we consider 3 smallsats deployed from the same upper stage on a common launch. Deployment strategies and how to chose reasonable and realistic velocity differences are discussed in [1], [7]. By giving the satellites small velocity differences, they enter slightly different orbits, with different periods. Due to this difference, the smallsats will start to drift relatively to each other, resulting in a freely-drifting swarm with different possible layouts (Fig. 1). Hence, the properties of a network supported such swarm will constantly change.

The different layouts that a freely-drifting swarm with 3 smallsats can assume include a *uniform distribution* of the satellites around the planet, one with 3 *overlapping satellites*, another where 2 *satellites overlap* with the third being diametrically opposite to them and finally a *trailing* or scattering layout, where the satellites diverge or converge towards each of the other layouts. Bearing this in mind, 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, with a *trailing* layout short re-visit intervals are followed by a larger one, while with *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, our simulations show that *overlapping satellites* only account for $2.8\,\%$ of the covered time and that the 2 *satellites overlap* layout only occurs $10.6\,\%$ of the time.

In addition to the variability introduced by changing layouts, 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 tilted 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.

Taking into account the dynamics between orbital planes and Earth's rotation, as well as the characteristics of the described freely-drifting swarm, it is clear that an IoT networking solution in the Arctic must consider these properties, specially

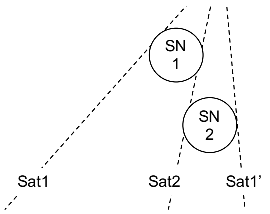

Fig. 2. Ground tracks for $Sat1$ and $Sat2$, with $t_{Sat1} < t_{Sat2} < t_{Sat1'}$

when transmitting data between multiple nodes. Since all satellites are capable of eventually reaching all ground nodes, a naive approach would be to always select the first arriving satellite as a next-hop. However, the desired destination may not be aligned with this satellite's ground track at the time. On the other hand, 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, despite the fact that it arrives later at the source.

This is illustrated by Fig. 2, where *Sat1* becomes visible to *SN1* before *Sat2*, but requires one more orbit revolution until its ground track ($Sat1'$) is aligned with *SN2*. Conversely, while *Sat2* only becomes visible to *SN1* later, due to its better track alignment, it is also visible to *SN2* in the same orbit revolution.

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 [17].

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 [18], it has also been considered in the context of other communication technologies [16]. Using it for constrained satellite and sensor-node links would allow benefiting from the existing adaptation layer [19] and compression mechanisms [20], reducing networking overhead.

In order to support other communication links that may exist, even between the same BR and SN, an SDN-inspired 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 [21]). However, these solutions typically introduce abstractions such as an overlay of links and networks resulting from the Bundle Protocol or Convergence Layers [22], 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 $4\,\mathrm{B}$ [12] 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 [13] 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, were 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 SDN-based routing approach.

III. METHODOLOGY

In order to realistically evaluate the feasibility of the proposed Internet of Arctic Things supported by a freely-

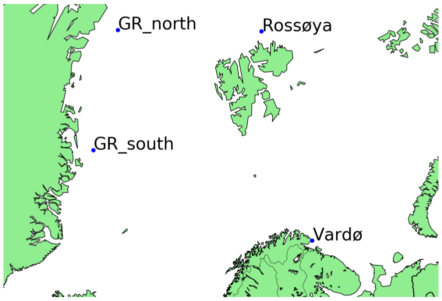

Fig. 3. Placement of ground nodes (Sensor Nodes: GR_north, GR_south and Rossøya; Ground Station: Vardø)

drifting swarm of smallsats, 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.

A. Freely-Drifting Swarm Simulation

The evaluated freely-drifting swarm was based on realistic satellite orbits from the Two-Line-Element (TLE) [23] set of AAUSat-3 [24], with the epoch *13 Feb 2014 12:35:42.657*. The used TLE was retrieved from the Systems Toolkit (STK) [25], and each of the 3 defined satellites had its *orbits-per-day* and eccentricity e parameters changed accordingly. An inclination of $98.6235°$ and a perigee height of $768\,\mathrm{km}$ was set to all of them. Their apogee altitudes were 771.83, 787.17 and $802.55\,\mathrm{km}$. These slightly different orbits are responsible for the previously mentioned drift that results in different layouts. For the chosen orbits, one "full cycle" of layouts, 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 Fig. 3, were based on a previous research work also addressing the Arctic region [26].

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 $1\,\mathrm{Mbit\,s^{-1}}$, based on available COTS S-band radios[1], while for links between BRs and SNs it was set to $20\,\mathrm{kbit\,s^{-1}}$ based on realistic bitrates also from COTS components such as from GOMSpace[2].

The used emulation tool [9], in addition to the already 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 $127\,\mathrm{B}$, 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 taking into account the overhead of the used protocols and from the *user's perspective*. The latter consists of the end-to-end response time, considering the instant from when a request is issued until its response is received. For this purpose, CoAP NON-confirmable requests were randomly created, following a random uniform distribution between $60\,\mathrm{s}$ and $180\,\mathrm{s}$. 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 $512\,\mathrm{B}$ 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 [27], 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*[3] 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, Fig. 4 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 this BR node as a next-hop since no others would be available at that instant. However, ArcNet5 requires one more orbital revolution in order to reach GR_north and complete the communication. Instead, by waiting for ArcNet1 or ArcNet9 to become visible, $30\,\mathrm{min}$ later, requests can be relayed directly to GR_north, reducing the end-to-end time in nearly $60\,\mathrm{min}$.

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

---

[1] gomspace.com/Shop/subsystems/communication/nanocom-sr2000.aspx

[2] gomspace.com/Shop/subsystems/communication/nanocom-ax100.aspx

[3] pyroute2 netlink library: http://docs.pyroute2.org

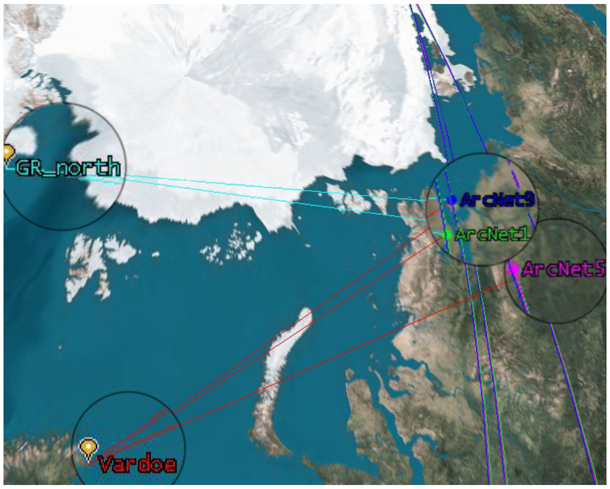

Fig. 4. Example of different ground tracks and coverage: ArcNet5 is the first reaching Vardø but fails to reach GR_north. ArcNet1 and ArcNet9 reach Vardø later but simultaneously reach GR_north, acting as relays.

was implemented using the light-weight *pyephem* library [28], which allows the calculation of upcoming passes for a given node. This implementation 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 48 days, covering more than a “full cycle” of layouts, starting with a *trailing* layout and returning to the initial state. This resulted in more than 32 000 CoAP requests being transmitted through the evaluated network.

A. Overhead

The impact of the chosen protocols was one of the main considerations in the proposed Internet of Arctic Things. In particular, one important goal was to take advantage of standard Internet protocols without resulting in prohibitive overhead. Table I 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).

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[4]. 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 I
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

B. Overall Performance

The overall performance of the evaluated experiment is summarised in Table II, 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 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 start to finish completion time can be, on average, reduced up to 93 min. However, since a real networking stack was used, unexpected behaviours due to congestion or delays led to the *Naive* approach being better for some requests, corresponding to less than 3 % when both routing approaches selected a different proxy. These correspond to outliers where a request may have missed the expected pass and therefore taken an incorrect route. Moreover, this unpredictable behaviour of the network stack is confirmed by the number of increasing losses in the less restrictive routing approach (*Smart5*), where by selecting a very short-lived pass results in some messages being lost or timing-out (limited to 12 h).

Fig. 5 shows the time taken since creating a request at the GS until it reaches the selected BR, with each request corresponding to a point in the plot for both the *Naive* and *Smart30* routing approaches. This figure further illustrates the impact of the different satellite layouts, with a majority of the requests taking less time to reach BRs when the swarm follows a *uniform distribution*, increasing the most with *overlapping satellites* and being subject to higher variation in the *trailing layout* (white background). The interpolation of the plotted

---

[4] Connected edges support: https://patchwork.kernel.org/patch/10369859/

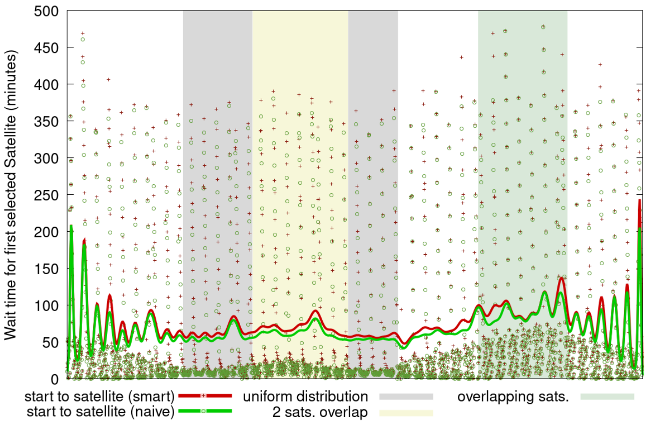

Fig. 5. Start to Satellite: *Naive vs. Smart30*

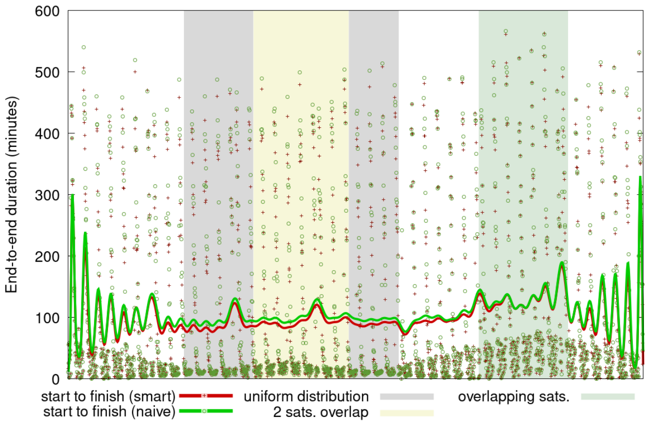

Fig. 6. Start to End: *Naive vs. Smart30*

TABLE II
PERFORMANCE COMPARISON – OVERALL

	Naive	Smart5	Smart15	Smart30
First Satellite (s)	4031	4349	4381	4427
Start to Finish (s)	5853	5060	5113	5401
Same Choice (%)	–	85.7	85.6	85.2
Improvement (s)	427	5636	5381	4281
Total Losses (%)	2.8	5.2	5.0	3.3

points – represented by a coloured line – shows that *Smart* routing generally takes longer to communicate with the desired BR. However, the selected BRs are better aligned with the final destination and, as confirmed by Fig. 6, allow achieving a lower end-to-end that outperforms the *Naive* approach.

In order to better visualise the negative impact of using a *Naive* routing approach, Fig. 7 illustrates the time penalty from selecting the first available BR, even though better alternatives may exist. This figure does not include requests where both routing approaches selected the same BR. As it can be seen, the penalty is higher when the swarm is found in a *trailing* layout, due to the higher scattering of satellites. Conversely, with *overlapping satellites* the penalty is less significant because the satellites' ground track is similar and fewer alternatives exist.

As previously described, the location of a ground node significantly influences the perceived satellite coverage. Fig. 8 shows the impact of selecting a *Naive* routing approach 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, meaning that only a few requests to this destination can benefit from *Smart* routing (approx. $5\,\%$). Nevertheless, end-to-end delay can still be significantly reduced. In different circumstances, GR_north is the farthest SN from the GS, resulting in a higher misalignment and more requests being penalised when selecting the first available BR, as seen in Fig. 9 (approx. $27\,\%$ of its requests or $9\,\%$ of the total number of requests). Finally, Fig. 8 shows that the impact of *Naive* routing in GR_south is lower than in GR_north (approx. $13\,\%$), while also sharing some similarities in how it is affected by different layouts.

V. CONCLUSION

In this paper, the concept of the Internet of Arctic Things was introduced, demonstrating how a freely-drifting small-satellite swarm can be used for supporting communications in the Arctic. The different layouts that such a satellite swarm can assume were analysed, as well as their impact on communications. An experimental assessment was conducted, emulating real IoT-protocol implementations combined with simulated satellite orbits. In particular, IPv6 and 6LoWPAN were used together with CoAP for as the basis of the defined networking architecture and a new satellite-aware routing approach was also tested. The obtained results indicate that a low number of losses can be achieved ($< 5\,\%$), while keeping overhead as low as $27\,\%$ when using CoAP. This confirms that low-cost smallsats without thrusters can effectively be deployed in order to provide coverage for different locations in the Arctic using COTS communication technologies and standardised networking protocols. The performance improvement from the proposed routing solution confirmed the benefits of taking into account satellites' ground tracks, instead of simply relying on a hop-count metric.

ACKNOWLEDGMENT

This work was partially funded by the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie Grant Agreement No. 699924. The authors also 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 *Proceedings of ESA Small Satellites Systems and Services 2016*, 2016.

[2] R. Birkeland, "Proposal for a satellite communication payload for sensor networks in the arctic area," in *Proceedings of the 65th International Astronautical Congress*, 2014.

[3] A. Council, "Telecommunications infrastructure in the arctic," 2017. [Online]. Available: oaarchive.arctic-council.org/bitstream/handle/11374/1924/2017-04-28-ACS_Telecoms_REPORT_WEB-2.pdf

[4] ESA, "Esa artes articom," 2011, accessed April 2018. [Online]. Available: https://artes.esa.int/projects/arcticom

[5] Iridium Communications Inc, "Iridium next: Advanced technology," 2013, iridium NEXT: Advanced Technology. [Online]. Available: http://www.iridium.com/About/IridiumNEXT/Technology.aspx

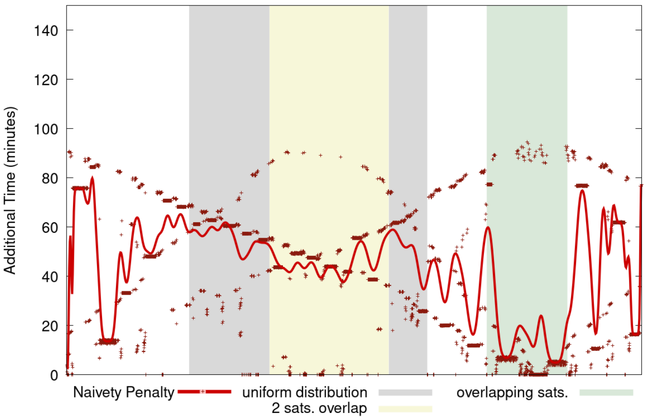

Fig. 7. Naivety Penalty Overall

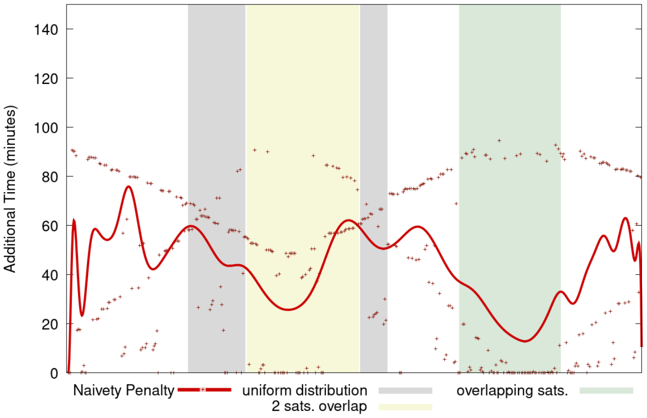

Fig. 8. Naivety Penalty for Rossøya

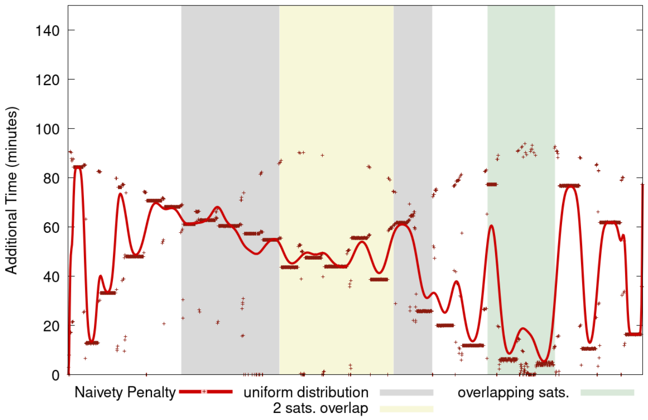

Fig. 9. Naivety Penalty for GR_north

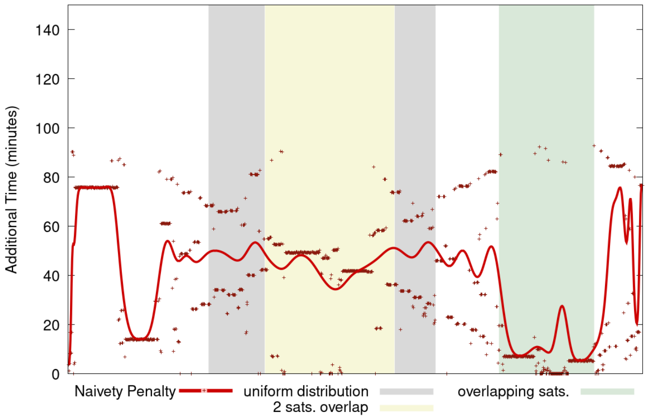

Fig. 10. Naivety Penalty for GR_south

[6] R. Birkeland, D. Palma, and A. Zolich, "Integrated smallsats and unmanned vehicles for networking in remote locations," in *Proceedings of The 68th International Astronautical Congress*, 2017.

[7] R. Birkeland, "Freely drifting cubesat constellations for improving coverage for arctic sensor networks," in *2017 IEEE International Conference on Communications (ICC)*, May 2017, pp. 1–6.

[8] A. G. 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, 2016.

[9] D. Palma, A. Zolich, Y. Jiang, and T. A. Johansen, "Unmanned aerial vehicles as data mules: An experimental assessment," *IEEE Access*, vol. 5, pp. 24 716–24 726, 2017.

[10] L. Lambrinos, C. Djouvas, and C. Chrysostomou, "Applying delay tolerant networking routing algorithms in maritime communications," in *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2013 IEEE*, June 2013, pp. 1–6.

[11] L. Torgerson, S. C. Burleigh, H. Weiss, A. J. Hooke, K. Fall, D. V. G. Cerf, K. Scott, and R. C. Durst, "Delay-Tolerant Networking Architecture," RFC 4838, Oct. 2015. [Online]. Available: https://rfc-editor.org/rfc/rfc4838.txt

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

[13] M. Auzias, Y. Mahéo, and F. Raimbault, "Coap over bp for a delay-tolerant internet of things," in *2015 3rd International Conference on Future Internet of Things and Cloud*, Aug 2015, pp. 118–123.

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

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

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

[17] A. Zolich, D. Palma, K. Kansanen, K. Fjørtoft, J. Sousa, K. H. Johansson, Y. Jiang, H. Dong, and T. A. Johansen, "Survey on communication and networks for autonomous marine systems," *Journal of Intelligent & Robotic Systems*, Apr 2018.

[18] IEEE, "IEEE Standard for Local and metropolitan area networks–Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)," *IEEE Std 802.15.4-2011*, pp. 1–314, Sept 2011.

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

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

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

[22] L. Wood, P. Holliday, D. Floreani, and W. M. Eddy, "Sharing the dream," in *2009 International Conference on Ultra Modern Telecommunications Workshops*, Oct 2009, pp. 1–2.

[23] D. T. Kelso, "Frequently asked questions: Two-line element set format," 1998, available at https://celestrak.com/columns/v04n03/.

[24] Aalborg University, "AAUSAT3," September 2014, online. [Online]. Available: http://www.space.aau.dk/aausat3/

[25] A. Graphics, "Stk," Online, 2016, accessed April 2016. [Online]. Available: http://www.agi.com/products/stk/

[26] R. Birkeland and D. Palma, "Freely-drifting small-satellite swarms for sensor networks in the arctic," in *Proceedings of the Third International Congress on Information and Communication Technology*, 2018.

[27] G. Tanganelli, C. Vallati, and E. Mingozzi, "Coapthon: Easy development of coap-based iot applications with python," in *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*, Dec 2015, pp. 63–68.

[28] B. Rhodes, "Pyephem home page," 2015, accessed October 2015. [Online]. Available: http://rhodesmill.org/pyephem/