Efficient Vertical Handover in Heterogeneous Low-Power Wide-Area Networks

Bart Moons, Abdulkadir Karaagac, Eli De Poorter, and Jeroen Hoebeke

***Abstract*—As the Internet of Things (IoT) continues to expand, the need to combine communication technologies to cope with the limitations of one another and to support more diverse requirements will proceed to increase. Consequently, we started to see IoT devices being equipped with multiple radio technologies to connect to different networks over time. However, the detection of the available radio technologies in an energy-efficient way for devices with limited battery capacity and processing power has not yet been investigated. As this is not a straightforward task, a novel approach in such heterogeneous networks is required. This article analyzes different low-power wide-area network technologies and how they can be integrated in such a heterogeneous system. Our contributions are threefold. First, an optimal protocol stack for a constrained device with access to multiple communication technologies is put forward to hide the underlying complexity for the application layer. Next, the architecture to hide the complexity of a heterogeneous network is presented. Finally, it is demonstrated how devices with limited processing power and battery capacity can have access to higher bandwidth networks combined with longer range networks and on top are able to save energy compared to their homogeneous counterparts, by measuring the impact of the novel vertical handover algorithm.**

***Index Terms*—Constrained application protocol (CoAP), heterogeneous networks, Internet of Things (IoT), low-power wide-area networks (LPWANs), network detection, static context header compression (SCHC).**

I. Introduction

As THE Internet of Things (IoT) continues to grow and, apart from the research community, starts to gain interest for new business use cases, the need for more diverse settings arises. To meet these demands, new low-power wide-area network (LPWAN) technologies, such as DASH7, Sigfox, and LoRa, have entered the market. Many of these communication technologies make use of the unlicensed industrial, scientific, and medical (ISM) 915-MHz band (Region 2) or the license-free European short-range device (SRD) 863–870-MHz band. Due to their unlicensed character, long range and low (energy) cost, they are the perfect candidate for massive low-cost sensor deployment.

Manuscript received October 1, 2019; revised December 10, 2019; accepted December 20, 2019. Date of publication December 24, 2019; date of current version March 12, 2020. This work was supported in part by the Flemish FWO SBO through Intelligent Dense and Long Range IoT Networks Project under Grant S004017N, and in part by the ICON Project MuSCLe-IoT. *(Corresponding author: Bart Moons.)*

The authors are with the Department of Applied Engineering, Ghent University, 9052 Ghent, Belgium, and also with imec, Leuven, Belgium (e-mail: bamoons.moons@ugent.be; eli.depoorter@ugent.be; jeroen.hoebeke@ugent.be).

Digital Object Identifier 10.1109/JIOT.2019.2961950

However, as these technologies are targeting different use cases, their characteristics differ drastically. LoRa and Sigfox, for example, have a throughput of a few hundreds of bits per second but offer in return a range up to 50 km [1]. DASH7, on the other hand, offers a shorter range, but provides throughputs of hundreds of kilobits per second [2]. Also, a lot of research has been conducted around DASH7 localization, which can be used to locate an object with a median location error down to 3.9 m using a single message [3]. Consequently, some technologies are a better choice for low latency and higher bandwidth requirements, while others may be better suited for long range, periodic sensor updates.

Nevertheless, by combining these technologies, several issues obstructing large-scale IoT adoption may be solved, such as over-the-air (OTA) updates and more accurate GPS-less localization for constrained devices. Although not much research has been conducted around this topic, the electronics company Murata recently brought dual-mode LoRa/Sigfox modules to the market, supporting other modulation types too [e.g., (G)FSK and OOK] [4]. A single chip is, therefore, able to switch between different networks with the use of a single antenna.

Such devices, however, require to move away from homogeneous to heterogeneous networks where a device can, depending on its current requirements, search for “Always the Best Connectivity” (ABC) [5]. As these technologies currently coexist as vertical silos next to each other, a higher complexity is involved in managing and communicating with such devices and networks. Therefore, multiple problems must be tackled in such configurations, the first and foremost problem being an efficient approach in detecting the presence of and switching to a more capable network. This has been put forward as a vertical handover, i.e., the handover between base stations of different wireless technologies, and the handover decision; the selection of the most appropriate wireless network [6].

In this article, the focus relies merely on network detection as this has the largest impact on the energy consumption, which has been determined as the main targeted efficiency.

The first contribution of this article is consequently a handover algorithm with configurable parameters to provide resilience and to serve a multitude of use cases. The impact of several configurations and the tradeoffs amongst these parameters are studied on the basis of simulations. The obtained results indicate that the correctly configured devices will consume substantially less energy compared to their homogeneous counterparts as they can take advantage of the complementary characteristics of the different wireless communication technologies they are equipped with. The tradeoff between latency, discovery time, and reliability becomes clear when the effect of the different parameters is studied. Furthermore, a single protocol stack is presented where the application layer is unaware of the underlying communication technology and a single packet payload structure can be used. The last contribution of this article is the further elaboration on the architecture of such heterogeneous networks, as presented in the previous work [7], to take away the complexity toward application developers.

The remainder of this article is structured as follows. First, a case study is presented to emphasize the need for heterogeneous LPWA networks. This is followed by the problem statement and research goals, which highlight the issues faced when developing a low-power network detection algorithm. Section V introduces reference technologies and pieces of background information used throughout our study in the remainder of this article, followed by a section on related work. Thereafter, the network detection algorithm, the architecture, and the proposed protocol stack are presented in Section VI. Finally, the energy efficiency of the presented algorithm is evaluated to show its flexibility, resilience, and how possible extension toward other technologies was taken into account.

II. Case Study: Construction and Logistics

In order to highlight some of the issues faced when developing applications in the heterogeneous LPWANs, a case study is presented in this section. The use case covers a construction and logistics company, where cranes and other materials are transported between the construction site and their warehouse. All equipments are used extensively and require regular maintenance to avoid high damage costs. Currently, such a company might depend on the discipline of their employees to measure the actual usage. This, however, is an error-prone task which is better solved by measuring the actual usage by equipping the material with accelerometer-enabled devices. Due to the assets' mobility, the trackers are battery powered and must drain as little energy as possible. In order to track the location of the construction tools, the trackers could be equipped with a global navigation satellite system (GNSS). However, such receivers generally consume a lot of power and are, therefore, not suited for this use case. In order to track the assets, the state-of-the-art LPWAN localization techniques can be used by sending regular updates, including accelerometer data, to the back-end over the best available network. Once the asset's usage has reached a certain threshold, maintenance should be notified over one of the available LPWAN technologies. At the construction site and at the warehouse, a DASH7 or private LoRa network is deployed, which allows for OTA updates and accurate low-power localization. Since Sigfox ought to provide global coverage, the parts in-between both sites are covered by this communication technology. An overview of the presented use case is given in Fig. 1.

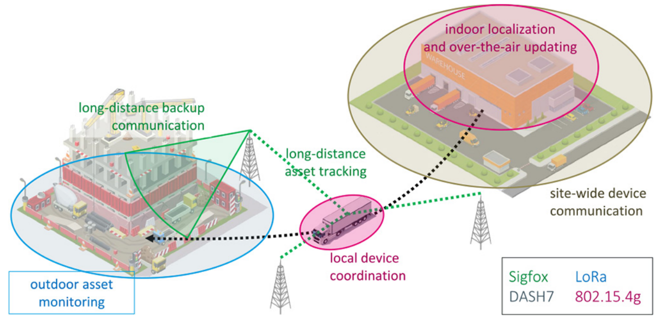


Fig. 1. Asset tracking use case for a logistics and construction company.

III. Problem Statement and Research Goals

While some case studies will have devices with a highly predictable trajectory, other cases might encounter objects moving around more randomly without any prior knowledge about the available networks. This requires a resilient algorithm, which incorporates easy, automated configuration for each device and use case. The algorithm should also be able to adapt itself depending on any form of input, i.e., the parameters of the algorithm should be modifiable by input from the device itself as well as the back-end.

A second consideration made was that once a device is connected to a network, it should check at regular intervals whether it is still connected to the network. However, some technologies will impose a lot of restrictions to the device in terms of downlink communication (e.g., Sigfox allows 4 downlink slots of 8 bytes every day). Another reason not to have many downlink slots might be the duty cycle of the gateway (i.e., 0.1%, 1%, or 10% [8]). In the dense networks, requesting an acknowledgment for each uplink is impractical. Signaling from the back-end to inform the device about future availability of other networks is therefore limited.

A third observation is the capability of the available technologies. Some are more powerful in terms of bandwidth and maximum transfer unit (MTU), whereas others are more focused on long range and limit the other characteristics. It is desirable for the device to switch to a better network once available, since the total time-on-air and energy consumption should be reduced to a minimum and the reception of large OTA updates made possible. Polling for a better network, however, introduces duty-cycle and energy costs, which imposes restrictions on the polling frequency. The implementation should also not limit the algorithm to the hereafter presented technologies, hence, the possibility to integrate new communication technologies should be made easy.

Finally, a single protocol stack is desirable as the application layer should be unaware of the current underlying technology. A CoAP/UDP/IPv6 approach has been put forward to comply with the Internet protocol standards, e.g., IP, UDP, and TCP, which form the global language spoken on the Internet for over 40 years. The major strength of these languages being their maturity and the interoperability between all Internet inhabitants. Another benefit is the portability, where different applications can be used independent of the underlying technology. However, some of the available technologies in LPWANs might not be able to transport certain Internet protocols and might have to follow a different approach. Sigfox, for example, allows each uplink transmission to use a maximum of 12 bytes, which does not cope with the 40 bytes header overhead of the IPv6 standard.

The goal of this article is to tackle the heterogeneity of use cases in a multimodal environment by focusing on a system that can integrate with a multitude of communication technologies for constrained IoT devices. Our research aims to answer the following questions.

1) How can a multimodal-constrained IoT device connect to the best available network for a variety of use cases and still offer enough flexibility?
2) How can the low-resource IoT devices use a single stack across a multitude of communication technologies, unaware of the current technology?
3) What is the impact of the presented approach on the energy budget of the device?

IV. Related Work

Not much research has been conducted around the topic of LPWAN detection and switching. Though, Wetterwalk *et al.* [9] proposed a very high-level system architecture of a heterogeneous network consisting of static context header compression (SCHC) (see Section V-C) enabled NB-IoT/LoRa devices. A machine learning algorithm is proposed at the back-end, which will determine the best communication technology for downlink traffic after sending simultaneously over both networks, resulting in a lot of energy overhead. The architecture is, however, restricted to LoRa/NB-IoT and neither did the authors perform any form of evaluation.

Chen *et al.* [10] proposed a central management system for the heterogeneous LPWAN, where different communication technologies can be incorporated in a single back-end application. However, heterogeneous devices are not being taken into account, hence, no vertical handover algorithm is present. Finally, Lemic *et al.* [11] proposed a mathematical model for location-based network discovery. The proposed method requires the device to know the location of base stations and an accurate estimate of its own location, which is not always feasible.

V. Reference Technologies

This section gives an overview of the different technologies and protocols that were used in the proposed setup and have been referred to in the following sections. First, the different LPWAN technologies are discussed, followed by a short discussion about constrained application protocol (CoAP) and SCHC, which are two protocols enabling single stack, Internet-compliant-constrained IoT devices.

A. LPWAN Technology Overview

LPWANs are formed out of cheap sensors running applications that require low bandwidth communications over long range, at a low cost and at low power. Currently, several technologies are emerging in this domain providing low cost and low power by using bands in the sub-GHz spectrum.

1) LoRa: LoRa is a radio access technology in the unlicensed sub-GHz band using chirp spread spectrum (CSS) modulation, patented by Semtech in 2014. CSS spreads out a narrow band signal over a wider channel bandwidth, making it more robust to noise and interference. Multiple spreading factors (SFs) are supported by LoRa, i.e., SF7–SF12, offering a tradeoff between a higher data rate and a longer range, respectively [12]. By using forward error correction (FEC) with code rates (CRs) ranging from 4/5 to 4/8, even more robustness can be provided [13]. The data rate and range are affected by a combination of SF, CR, and chosen bandwidth.

On top of the LoRa physical layer, the open LoRaWAN MAC layer standard has been defined by the LoRa Alliance. This layer provides a medium access control mechanism and defines three types of end-devices: Classes A–C, mainly providing different ways of bidirectional communication. For a LoRa Class A device, each uplink transmission is followed by two downlink receive windows of 1 and 2 s, respectively, during which the end-device will listen for a preamble, indicating downlink communication. Class B devices will listen for downlink traffic at predefined times after synchronization to the network server using network beacons. Devices continuously listening for downlink packets are of type Class C.

Each LoRa packet starts with a programmable preamble part, ranging from 6 to 65 532 symbols, followed by two sync words and two downchirp symbols, used to synchronize traffic between the sender and receiver. Therefore, header overhead of the physical layer can be very limited. When the length of the payload is known in advance and configured on both sides, the explicit header (EH) can also be removed, which otherwise contains the payload length and a cyclic redundancy check (CRC) encoded with a CR of 4/8 [14].

2) DASH7: The DASH7 alliance protocol (D7A) specifies a full vertical network stack, covering the complete OSI model, focusing on mid-range communication. It was initially developed for 433-MHz wireless communication, based on the ISO/IEC 18000-7 standard, which now also includes the 868-MHz SRD and 915-MHz ISM bands [2], [15]. The presentation layer contains D7AP files, consisting of configurations and user data, which can be executed as scripts. DASH7 applications are intended to be built using those files. The downside of this full vertical stack is that it introduces a lot of overhead when using an IPv6-based standardized approach. In fact, when using the DASH7 specification, a CoAP/UDP/IPv6 packet is encapsulated on top of the DASH7 application layer protocol.

In a DASH7 network, two types of communication models can be used. The first one is based on low power wake up, where a sleeping end device will discover a requesting signal by regularly waking up and detecting an advertising frame containing the time of the upcoming request, as shown in Fig. 2.

Using the second approach, bidirectional communication can be achieved by means of dormant sessions. A sleeping end point will wake up at predefined times, known by the gateway, that will send data during these active sessions.

Data rates, modulation types, passband, and stopband requirements are specified by channel classes. Data are encoded using PN9 encoding, which might be used in combination with an FEC scheme. As the FEC encoder is a 1/2 rate

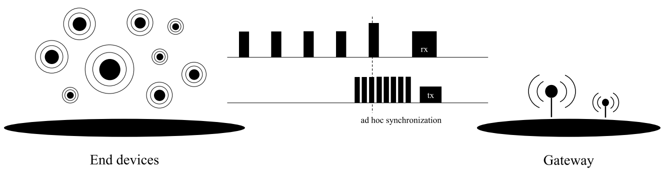


Fig. 2. DASH7 low power wake up and ad hoc synchronization.

convolutional code, the data rate will decrease accordingly. Each packet is preceded by a ramp-up period and a preamble to synchronize the clock of the receiver followed by a sync word to align the packet payload. As all of these parameters are configurable, more or less robustness, and consequently overhead, can be provided [15].

3) Sigfox: Sigfox is a proprietary technology, patented by the French company of the same name. Sigfox itself or in partnership with others offer an already deployed end-to-end 0G network, which already covers more than 60 countries. Their technology is based on binary phase-shift keying (BPSK) in the 868-MHz ultranarrow SRD band (UNB) [12]. Narrowband modulation techniques are able to obtain a higher link budget because the noise level in a single narrow band is minimal [16]. The benefit of such a robust radio signal decreases the data rate, whereas the time-on-air increases. Another drawback of the UNB modulation is the significant link asymmetry. Downlink communication is limited to four 8-byte messages every day.

4) Others: The previous paragraphs aim to provide information about technologies that were used throughout the remainder of this article. However, as the algorithm discussed in Section VI provides flexibility in adding new technologies, other relevant technologies are discussed as follows.

1) *Cellular Technologies:* Narrowband IoT or NB-IoT and long-term evolution machine type communications (LTE-M) are both cellular technologies proposed by the Third Generation Partnership Project (3GPP). Both aiming to provide low power, low cost, and long range with improved indoor and outdoor coverage. LTE-M provides low latency and higher bandwidth, whereas NB-IoT targets devices with ultralow device cost and power consumption, at the expense of up to 10-s latency. The most important distinction between cellular and other LPWA technologies is their operation in the licensed spectrum and hence, their ability to communicate without a duty cycle limitation [17]. Although NB-IoT reuses many of the mechanisms defined in LTE, two extensions have been defined in order to save power. The first one being the extended discontinuous reception or eDRX, which is used to check the paging channel periodically for incoming data and has been extended from 2.56 s in LTE to 175 min in NB-IoT. Even more power can be saved by entering the power-saving mode (PSM), which allows the constrained device to remain registered to the network, without monitoring the paging channel. The duration of the PSM cycle can last up to approximately 413 days. Downlink traffic is, therefore, limited by the periodicity of the PSM or DRX cycle [18].

TABLE I
DIFFERENCES BETWEEN WEIGHTLESS-W/-N/-P

	Weightless-N	Weightless-P	Weightless-W
Directionality	1-way	2-way	2-way
Feature set	Simple	Full	Extensive
Range	5km+	2km+	5km+
Battery life	10 years	3-8 years	3-5 years
Feature set	Very low	Low	Low-medium
Range	Very low	Medium	Medium

2) *Weightless:* Three different standards have been proposed by the Weightless Special Interest Group: Weightless-T, deployable in target value (TV) whitespace, Weightless-P, providing high performance and the uplink-only Weightless-N protocol, focusing on ultralow cost. The main differences are listed in Table I. In Weightless-P, every channel comprises 12.5 kHz of the spectrum and can be assigned an adaptive data rate, ranging from 200 b/s to 100 kb/s, organized by the time-synchronized gateways. Weightless-P is a downlink-oriented protocol, whereas Weightless-N only allows uplink traffic.

B. CoAP

The CoAP can be seen as the Hypertext Transfer Protocol for constrained devices, as it enables a representational state transfer (REST) communication approach on small embedded devices. The lightweight nature of CoAP makes it the perfect candidate for our protocol stack. Every object contains a list of resources, representing data available from sensors or actions available to actuators. Every resource is accessible through a URI and can be interacted with using the well-known REST methods GET, PUT, POST, and DELETE [19].

Considering the asset tracking use case, many sensor nodes will send frequent updates about their usage. Acknowledging every message will add a significant cumulative load on the network and is not necessary. Therefore, the following CoAP mechanisms were considered.

1) Observe: The CoAP Observe extension is a simple mechanism to retrieve a representation of a resource and keep this updated by the subject as long as the observer is interested. The extension uses a best-effort approach for sending updates to the observer.

In a CoAP Observe scenario, the data collection is always initiated by the observer, who also has to maintain all relationships for each subject. This requires a lot of bookkeeping and the back-end to know all its data sources beforehand [20].

2) No-Response Option: While CoAP implements a nonconfirmable (NON) mechanism to omit the acknowledgment of a particular message, the server will still reply with a response code, due to the request/response nature of the protocol. Therefore, the working group published an amendment to the protocol specification introducing the No-Response option in order to get rid of any kind of reverse traffic [21].

In many LPWAN use cases, where downlink traffic is scarce, the No-Response option makes classic updates

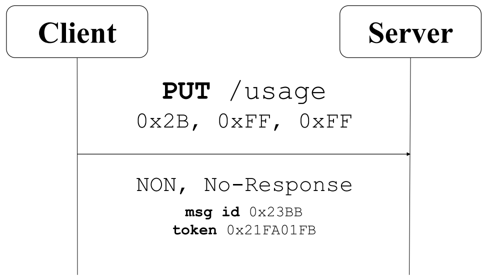


Fig. 3. Typical CoAP request from the application.

consume even less resources than Observe and was, therefore, considered a better fit. A typical CoAP request from the application can be seen in Fig. 3

C. SCHC

As discussed in Section III, some technologies do not support the use of traditional Internet protocols and may, therefore, not benefit from the advantages introduced by a standardized communication approach. As a consequence, a new IETF working group was formed and drafted the SCHC mechanism as a new standard [22]. This protocol makes use of a static context shared between two communication endpoints, keeping track of regularly used network headers, in order to achieve compression. The selected context is represented by an ID in order to inform the other side about the original headers of the packet [23]. Using this technique, it is possible to shrink the headers down to 95% of the original size [7]. On top, the specification also prescribes how large or uncompressed packets, which do not fit in a single Layer 2 protocol data unit (PDU), should be fragmented. Both the compression and fragmentation mechanisms are discussed in the following sections.

1) Compression: In order to compress a protocol header, each header field is matched against the corresponding entries in a rule (such as the one in Table II). Each original header field value is matched against the TV using the matching operator (MO) from that rule entry. After an exact match, the compression–decompression action (CDA) is applied to the header field and the result elided or added to the compressed header. The `NOTSENT` CDA, for example, will completely leave out the header field from the original header. Others, such as the most significant bit (`MSB(x)`) MO will limit the transferable value to the `x` MSBs so the other side is able to combine the received bits and the TV from the rule entry to retrieve the original value.

The application will make use of the SCHC rule as shown in Table II.

Since SCHC is responsible for reliability, there are no messages that require acknowledgments from the CoAP layer. Therefore, the same message ID may be used over different requests. Nevertheless, CoAP requires the use of separate tokens to intertwine a number of packet exchanges, explaining the use of the MSB MO on this header field. The MO will only match the MSBs to the original value as indicated in the field length (28 in this case). The least significant bits are added to the compressed header. Our example shows a variation on the last four bits of the token, allowing 16 simultaneous request/responses.

TABLE II
SCHC RULE USED TO COMPRESS THE REQUEST FROM FIG. 3

Field	FL	DI	Target Value	MO	CDA
Version	2	BI	0x01	&equal	NOTSENT
Type	2	BI	0x01 (NON)	&equal	NOTSENT
Token Length	4	BI	0x04	&equal	NOTSENT
Code	8	UP	0x03 (PUT)	&equal	NOTSENT
Message ID	16	BI	0x23BB	&equal	NOTSENT
Token	32	BI	0x21FA01F0	&MSB(28)	LSB(4)
URI-Path	40	UP	"usage"	&equal	NOTSENT
No-Response	8	UP	0x1A	&equal	NOTSENT

```
[001] network_layer_send(): sending 22
    bytes of application layer data
[002] CoAP header (18 B):
[003] 54 03 23 BB 21 FA 01 FB B5 75 73 61
    67 65 D1 EA 1A FF
[004] Application data (4 B):
[005] 00 00 00 07
[006] schc_compress(): compressing 18 bytes
    of CoAP header
[007] Rule id: 21 (0x15)
[008] 15 B0 00 00 00 70
```

Listing 1. Extract of the CoAP layer compression

After adding the rule id, a matching header with this rule will compress the 18-byte long CoAP header down to 1 byte (4 bits Token + 4 bits padding), followed by the application data, which is shown in Listing 1.

2) Fragmentation: As IPv6 demands a minimum underlying MTU of 1280 bytes and payload sizes exceeding the MTU require more advanced techniques, a fragmentation mechanism is offered by SCHC.

A predefined amount of fragments are grouped together in a so-called window. Once a complete window is received, one of the three reliability modes can be used to ensure a correct reassembly and to offer optional reliability.

1) *ACK-Always:* Each window is acknowledged, regardless of any missing fragments.
2) *ACK-On-Error:* Only windows are acknowledged when a fragment went missing belonging to that window.
3) *No-ACK:* No reliability is offered beyond that of the underlying communication technology.

To ensure correct desegmentation, a fragment compressed number (FCN) is added, which is also used to indicate the end of a window.

VI. LOW-POWER WAN DISCOVERY

This section will leverage on the previous work [7] to enable seamless handovers in LPWANs. After the network architecture is presented, the algorithm is explained, both on back-end side and constrained node side.

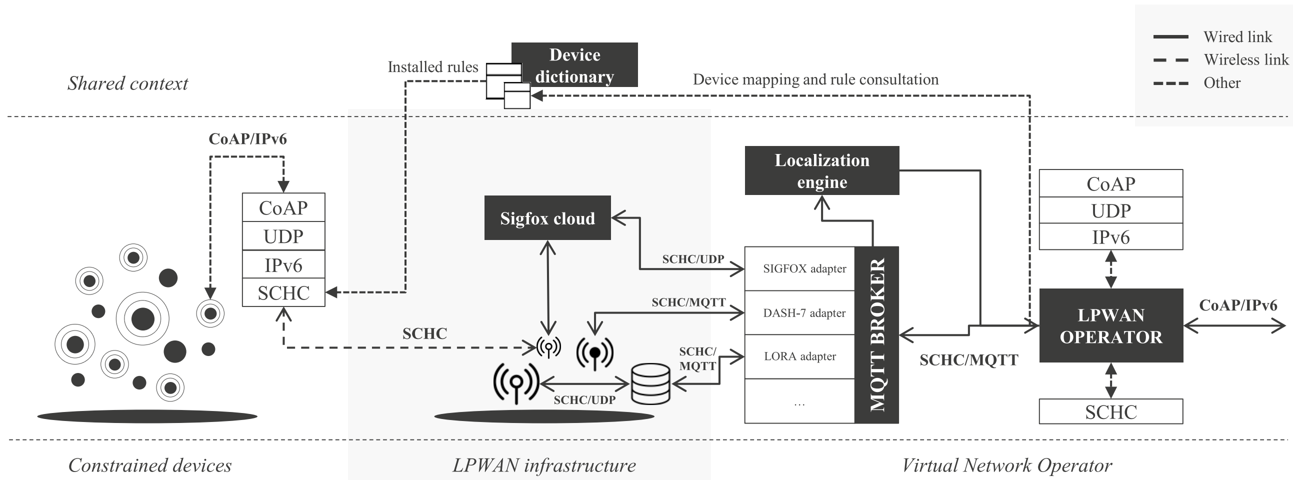


Fig. 4. Modular architecture of the VNO allows easy addition of technologies.

A. Overall System Architecture

In order to take away the complexity of a heterogeneous network, a novel system architecture is required. As proposed in [7], a modular virtual network operator (VNO) is used, where any type of network can be plugged in by using adapters. In Fig. 4, a high-level overview of the adapter-based architecture is shown.

Once a new network is deployed or being used, a suitable adapter will be installed by the VNO in order to interface with the network infrastructure, e.g., the SigFox cloud, a LoRAWAN network server, etc. Toward the right, all adapters have a unified data format, removing the complexity of the underlying network architecture.

The low resource multimodal IoT devices, shown on the left, use a uniform CoAP/IPv6/SCHC stack across multiple technologies. For each of these devices, the VNO employs an entry in the dictionary where its SCHC rules are exposed. In uplink, end-devices will send compressed SCHC packets, which eventually end up at the LPWAN operator component that will perform decompression. A localization engine is informed about incoming data in order to perform low power localization, based on the received wireless signal. In downlink, clients can generate CoAP/IPv6 packets, which will end up at the VNO that will compress the CoAP/IPv6 packets and forward them to the correct physical device. The compressed packet is forwarded to the corresponding adapter and sent over the active technology.

To realize a working network selection algorithm, which is able to adapt to the current technology and conditions of the network (e.g., dense networks where the back-end can provide information about less occupied channels) and provide enough flexibility, both the end-devices and VNO are extended with extra intelligence. These extensions and the design vertical handover algorithm are presented in the following two sections.

B. Network Discovery Method

The easiest method for a wireless device to discover reachable wireless networks is by keeping all interfaces on all the time. Another way of discovering a wireless network might happen by actively scanning a channel by sending Probe Requests and waiting for responses, such as used in the IEEE 802.11 scanning phase [24]. Also, as proposed in [6], the position information of the device and a location-service server can assist the device to efficiently discover and connect to the best available network.

These methods, however, are very energy consuming and are impractical for some LPWAN technologies as they also have to take duty-cycle restrictions into account. Therefore, an LPWAN tailored algorithm is required which makes a tradeoff between power efficiency and network discovery time. Since some use cases require more or less power efficiency and/or network discovery time than others, these requirements should be adjustable by taste. The core of the algorithm is therefore built up around four configurable parameters per technology which directly affect these requirements.

1) `polling_threshold`: This threshold is used to check if a better network is available.
2) `downlink_threshold`: This threshold is used to check if the current network is still available.
3) `max_downlink_retries`: This parameter is used to indicate the number of retries once a downlink is requested.
4) `priority`: This parameter is used to indicate which technology gets priority when 2 or more technologies overlap in time.

The unit of the first two thresholds can be either a time period or the number of messages transmitted by the end device (e.g., for end-devices with periodic uplink transmissions). The latter unit will be used in the remainder of this article. An example configuration is given in Table III.

This configuration means that when the constrained node is not connected to LoRa it will check every ten messages if such a network is available. If the device is currently connected to a LoRa network, it will expect an acknowledgment every four messages. This logic is realized by means of a state machine, as is explained in the following section.

TABLE III
NETWORK DRIVER THRESHOLDS

Technology	`polling`	`downlink`	`retries`	`priority`
Sigfox	-	35	0	0
LoRa	10	4	1	1
DASH7	6	2	2	2

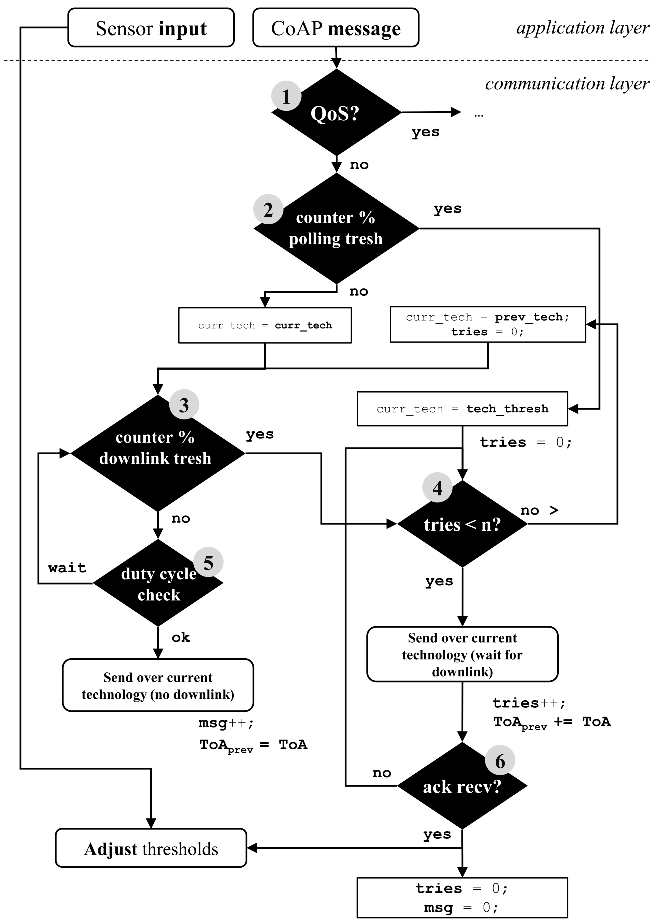


Fig. 5. State machine of the constrained device.

1) End-Device State Machine: Fig. 5 depicts how the logic for the constrained device is separated in two layers: 1) the application layer and 2) the communication layer.

The application layer takes care of CoAP requests and responses and processes input data from the sensors, while the communication layer keeps track of the current network interface and implements the logic to switch between the different interfaces. Once a CoAP request is pending at the application layer, it will pass the message to the communication layer. From there the following flow applies.

1) If quality-of-service (QoS) is required by the application, a different approach may be requested, otherwise, every message passed to the communication layer will increment a counter, which is used to check if it matches with one of the two parameter thresholds of a technology.
2) The message counter is checked against all network interfaces' `polling_threshold` parameter. Once the result of *counter* mod *polling_threshold* is 0, the network interface is added to the list to poll for, otherwise.
3) The counter is matched against the current network driver its `downlink_threshold` parameter. If the mod operation returns 1, a regular uplink will be sent over the current technology.
4) Otherwise, the list of network drivers to poll for is used to request an acknowledgment, conform the number of `max_downlink_tries` of the network driver, from the back-end. If no response is received, the device will return to the technology previously connected to.
5) In order to meet the regulatory limitations of the duty cycle limited technologies, the message will first pass the duty cycle check component, which keeps track of the time-on-air of every duty cycle limited transmission. If the current technology is bound to a duty cycle, then the next transmission will be scheduled as follows:

$$t_{\text{wait}} = (100 - DC) \cdot \text{ToA}_{\text{prev}} \tag{1}$$

with DC, the duty cycle in percent and ToA, the previous (cumulated) time-on-air. When requesting downlink information, the duty cycle is not taken into account to ensure fast handovers, but cumulated for the next uplink-only transmission. This is in line with the ERC/REC 70-03 regulations, as long as the total time-on-air is calculated based on a one hour period [25].
6) Finally, if an acknowledgment is received, two things are indicated as follows.
   a) The current technology is available.
   b) The contents of the acknowledgment may be used to update the parameters of the threshold.

```
struct network_driver {
   struct network_driver* next;
   char name[4];
   uint16_t mtu;
   uint8_t polling_threshold;
   uint8_t downlink_threshold;
   uint8_t max_downlink_retries;
   uint8_t priority;
   void (*init)(void);
   void (*stop)(void);
   uint8_t (*send)(uint8_t* buf, uint16_t
       len);
}
```

Listing 2. `network_driver` structure

2) Network Drivers: Information for each physical interface is kept in a `network_driver` instance (Listing 2) as part of a linked list, derived from the lightweight IP (lwIP) implementation [26].

A network driver will forward data to the correct network interface through the `send` pointer and can be initialized and stopped using the appropriate function pointers. Each network driver also implements the parameters required by the algorithm as discussed in the previous section. All interfaces are kept in a linked list, which makes it easy to loop over and add or remove any.

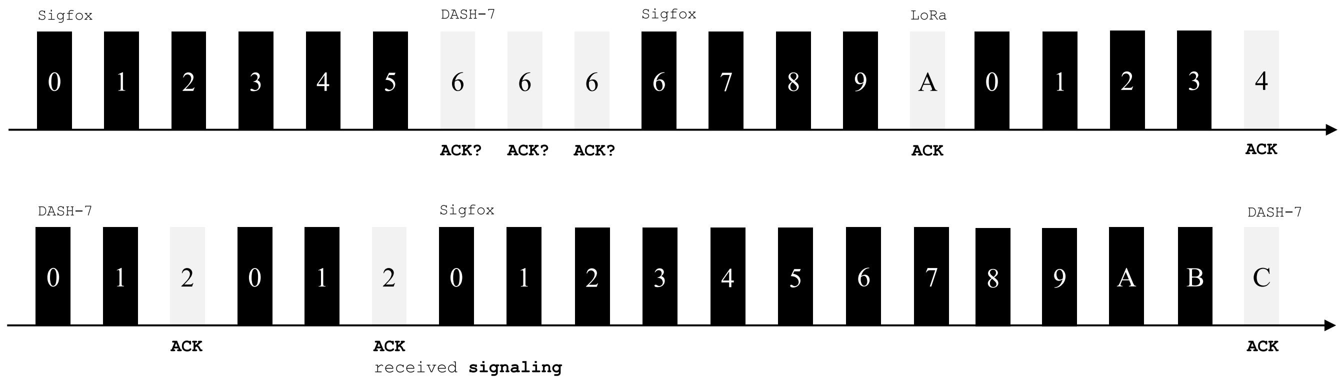


Fig. 6. Switching example and signaling from the back-end to the device.

3) Example: As Table III indicates, the algorithm will check every six unacknowledged messages if any DASH7 network is available. The upper part of Fig. 6 demonstrates the initial configuration, where the device is connected to the Sigfox network and fails to discover a DASH7 network after its `polling_threshold` (6) and its number of `downlink_retries` (2). Sigfox is used as a fall back, as we assume this is always available. Four messages later, the device reaches the `polling_threshold` for LoRa and receives an acknowledgment over this technology. Once a new network is discovered, the message counter is reset and the device will start to communicate over this network. Once the message counter reaches the technology's `downlink_threshold`, the device will ask for a signaling message from the back-end to check if the network is still available.

The second example shows a device connected to a DASH7 network with the `downlink_threshold` parameter set to 2. This means the DASH7 device will ask for a downlink for every 2 uplink messages in order to test the network availability. As the back-end has information about the future network availability, it signals the device to use Sigfox for the coming hours. By adjusting the thresholds of the other technologies, the device will not poll for the availability of other technologies, preserving energy, and flexibility.

C. Virtual Network Operator

The back-end implements a routing table, keeping track of the IPv6 address(es) of a single device with multiple extended unique identifiers (EUI) as well as the active technology. An outgoing IPv6 packet, i.e., an IPv6 packet going to the LPWA network, is matched against a device and forwarded to the device over the last active technology. However, the low power nature, i.e., the ability to receive data after an uplink packet, of the network requires the back-end to keep track of outgoing packets as shown in Fig. 7.

Upon a new request from the IPv6 network, the IPv6/UDP/CoAP headers are compressed using the SCHC adaptation layer. Next, the active technology is checked. Some configurations, e.g., LoRa Class C, allow instantaneous downlink communication, for others, the packets are sequentially added to the queue. Each request coming from the LPWA network will trigger the VNO to match the EUI of a particular technology to a device, update the active technology, check if there are any packets present for that device in the queue, and forward the first added packet (FIFO).

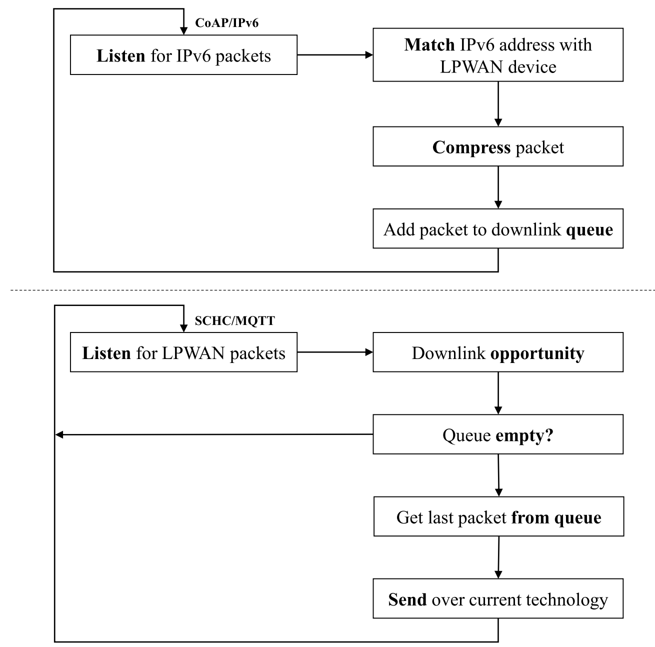


Fig. 7. Main loop in the back-end.

Once a constrained device sends a request to the back-end system, asking for confirmation about the availability of the network, updated `polling_threshold` and `downlink_threshold` parameters can be piggybacked on the network status acknowledgment.

VII. Performance Evaluation

A. Energy Overhead

As LPWAN devices are required to operate on a single battery charge for multiple years, energy consumption is one of the prime criteria in evaluating the feasibility of new concepts and implementations. While the presented handover schema may have many benefits (the most important one being the higher available bandwidth and bit rate), it may also introduce

TABLE IV
LPWAN OVERVIEW OF DATA-RATE, OUTPUT POWER, RECEIVER SENSITIVITY, AVERAGE TRANSMIT AND RECEIVE POWER REQUIREMENTS, PHYSICAL LAYER HEADER SIZE, MAC LAYER HEADER SIZE, AND MTU

Technology	Configuration	R (kbit/s)	TX_{op} (dBm)	RX_{sens} (dBm)	I_{rx} (A)	I_{tx} (A)	PHY (B)	MAC (B)	MTU (B)
Sigfox	Uplink	0.1	14	-	-	0.030	6	8	12
	Downlink	0.6	-	-126	0.01	-	18	2	8
LoRa	SF7 CR 4/5 BW 125 kHz	5.4688	13	-123	0.0108	0.029	-	13 - 28	222 - 207
	SF8 CR 4/5 BW 125 kHz	3.125	13	-126	0.0108	0.029	-	13 - 28	222 - 207
	SF9 CR 4/5 BW 125 kHz	1.7578	13	-129	0.0108	0.029	-	13 - 28	115 - 100
	SF10 CR 4/5 BW 125 kHz	0.9766	13	-132	0.0108	0.029	-	13 - 28	51 - 31
	SF11 CR 4/5 BW 125 kHz	0.5371	13	-133	0.0108	0.029	-	13 - 28	51 - 31
	SF12 CR 4/5 BW 125 kHz	0.293	13	-136	0.0108	0.029	-	13 - 28	51 - 31
DASH7	Lo-Rate PN9 w/ FEC	4.8	13	-	0.0108	0.029	8	5 - 13	251
	Normal PN9 w/ FEC	27.7775	13	-	0.0108	0.029	8	5 - 13	251
	Hi-Rate PN9 w/ FEC	83.3335	13	-	0.0108	0.029	10	5 - 13	251

some undesirable downsides, such as extra energy consumption. Therefore, a simulation was performed in MATLAB in order to evaluate the feasibility of the proposed algorithm.

1) Energy Model: In order to evaluate the proposed algorithm based on energy consumption, header overhead, MTU, current per transmission, and receive power and data rate were gathered from datasheets and are summarized in Table IV.

As the energy consumption is mainly determined by the time-on-air of a device, and the time-on-air is depending on the bit rate and the physical layer packet size, the following equation can be used:

$$J_{\text{tx}} = \frac{(\text{PL} + H) \cdot 8}{R_{\text{tech}}} \cdot I_{\text{tech}} \cdot V \tag{2}$$

where PL and H are, respectively, the payload length and header length in bytes, divided by the data rate in *bps*. This equation does not apply to LoRa as the proprietary CSS modulation is used there. Nevertheless, Semtech does provide ways to calculate the data rate and time on air for LoRa-enabled devices [27].

First, in order to know the time on air, the total number of payload symbols must be calculated using the following formula:

$$n = 8 + \left\lceil \frac{8\text{PL} - 4\text{SF} + 16\text{CRC} - 20\text{EH}}{4(\text{SF} - 2\text{DE})} \right\rceil \cdot (\text{CR} + 4) \tag{3}$$

with EH and CRC being the presence of an EH or CRC, respectively (1 or 0) and DE being the low data rate optimization.

Once the total amount of symbols are calculated, also the time (in milliseconds) required to transmit one symbol must be known

$$t_{\text{sym}} = \frac{2^{\text{SF}}}{BW}. \tag{4}$$

Furthermore, as every frame starts with a preamble with a configurable length (l), this must be calculated as well

$$t_{\text{pr}} = (l + 4.25) \cdot t_{\text{sym}}. \tag{5}$$

Finally, the previous equations can be combined when calculating the energy for an LoRa-enabled device, divided by 1000 to convert from milliseconds to seconds

$$J_{\text{tx}} = \frac{I_{\text{tx}}(n \cdot t_{\text{sym}} + t_{\text{pr}}) + 2I_{\text{rx}} \cdot t_{\text{pr}}}{1000} \cdot V. \tag{6}$$

Since an LoRa class A device will open 1 or 2 receive windows for possible downlink communication, which is at least equal to the time required to detect a preamble (t_{pr}), this is included in the model.

In order to show the energy efficiency for each technology, the energy overhead per 12 bytes uplink and 8 bytes downlink is shown in Fig. 8. Sigfox tends to have the largest energy consumption, since 12 bytes in uplink require 2.08 s of airtime. Due to a data rate twice as low for downlink communication, LoRa with SF12 will consume almost twice the energy of an 8 byte Sigfox packet. One receive window for LoRa is assumed for downlink communication. The graph clearly shows the low energy overhead of DASH7.

Based on this, it can be concluded that it would be interesting to benefit from the lower energy consumption and higher data rate offered by DASH7 while still obtaining the range Sigfox has to offer.

In the next section, the benefit and overhead of the proposed handover algorithm is evaluated using a simulation in MATLAB.

2) Simulation: In this section, the algorithm will be evaluated theoretically to determine the energy consumption overhead. The energy overhead calculations of a heterogeneous LoRa–DASH7–Sigfox device based on Table IV combined with the network driver principle of Section VI-B2 are implemented in MATLAB. The simulation also implements the state machine of Section VI-B and calculates based on the network availability, the energy overhead of every transmitted message, and possible downlink traffic.

As the availability of the networks cannot be modeled based on an exact model, a probabilistic model is maintained based on a standard normal probability distribution object (p). In our model, four cases are put forward: worst case, where the algorithm is running, but no other networks are available. *bad*

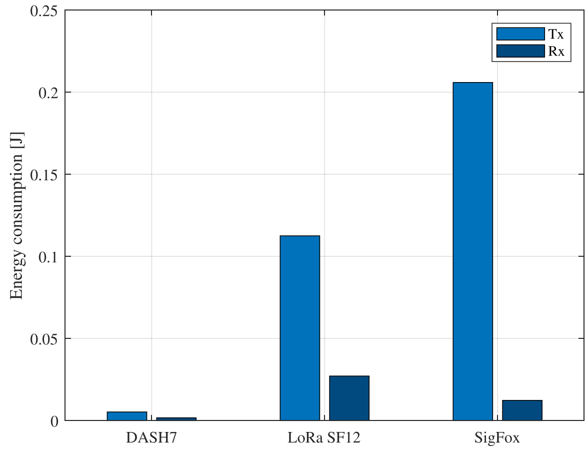


Fig. 8. Energy overhead for each technology for 12 bytes in uplink and 8 bytes in downlink conform the presented models.

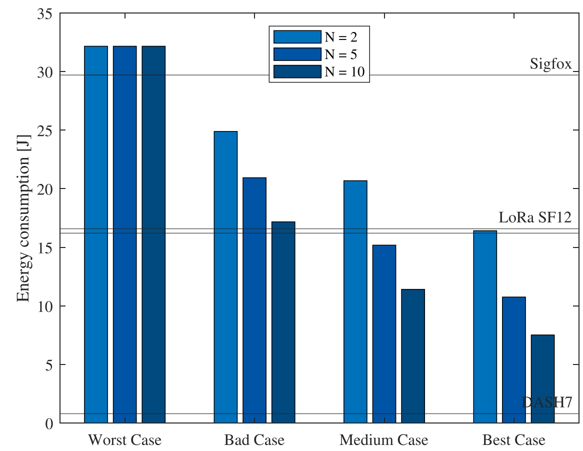


Fig. 9. Heterogeneous devices using the vertical handover algorithm versus homogeneous devices sending 12 bytes every 10 min over a timespan of 24 h.

TABLE V
NETWORK AVAILABILITY PROBABILITY (p)

Technology	Worst Case	Bad Case	Medium Case	Best Case
Sigfox	1	1	1	1
LoRa	0	0.25	0.5	1
DASH7	0	0.25	0.5	0.75

TABLE VI
MEDIAN ENERGY CONSUMPTION (IN JOULE) FOR A HETEROGENEOUS DEVICE FOR DIFFERENT CASES OVER A 24-H TIMESPAN

N	Worst Case	Bad Case	Medium Case	Best Case
2	32.16	24.89	20.68	16.41
5	32.16	20.94	15.18	10.75
10	32.16	17.17	11.41	7.51

case, *medium case*, and *best case* with increasing network availability of the other networks. These values are shown in Table V.

However, once a device is connected to a network, the probability of sending over the same network increases as the device is still in the presence of the network. Therefore, the variable N is defined, expressing the amount of consecutive messages over the same technology before disconnection.

For every simulation, the corresponding probabilities are applied when the device attempts to send a message. As an example, a device with an uplink transmission frequency of 10 min over a 24-h timespan simulation was run. The values in Table VI indicate how a higher N and p result in a lower energy consumption. Only for a worst-case scenario, the energy budget of the device is not influenced by N, as the only available network is Sigfox.

In order to evaluate the energy overhead of the handover algorithm, a comparison is made the between single technology devices and a multimodal device. The single technology devices (Sigfox, LoRa SF12, and DASH7) transmit an uplink packet every 10 min and receive downlink communication based on the values of Table III (i.e., Sigfox 35, LoRa 10, and DASH7 6). The heterogeneous devices wield the same uplink and downlink frequency as indicated in Table III. In Fig. 9, the energy consumption of the single technology devices over a 24-h timespan is indicated by the horizontal lines. The heterogeneous devices, on the other hand, have been modeled for 12 different cases from Table VI and are illustrated with bars for every N.

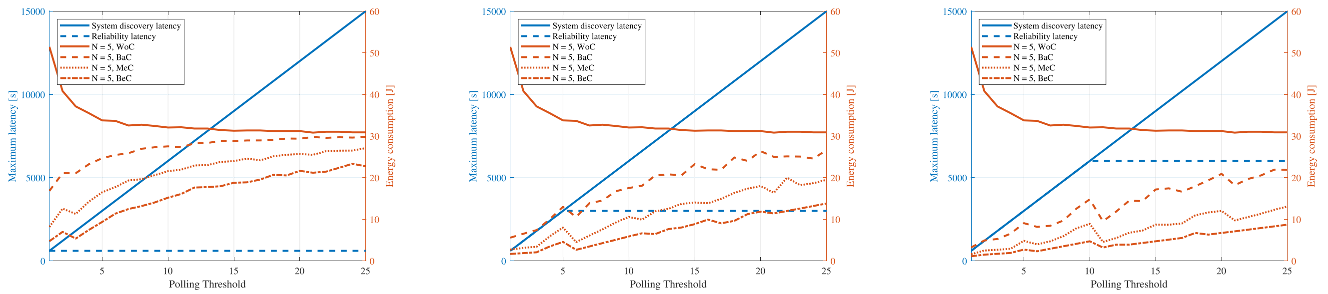
It is clear from the figure that in a worst-case scenario, the heterogeneous device, configured using the parameters from Table III, only consumes 8% more energy than a single technology Sigfox device. However, once the probability of connecting to a better network increases, the multimodal device quickly outperforms the homogeneous device. For a *bad case*, where the heterogeneous device has only 25% chance of connecting to a LoRa or DASH7 network, with probability $N = 10$, it will already outperform single technology LoRa (SF12) devices.

Once the device is in the proximity of a DASH7 network, a higher data rate is available, hence a shorter time on air. As a result, OTA updates and data offloading can be conducted over this medium-range technology. However, once the device starts moving away from the network, which has a range from 1 to 5 km, it may benefit from a lower data rate, longer range technology, such as Sigfox or LoRa.

Now that it has been shown that deploying a multimodal network is beneficial for LPWAN devices in terms of energy consumption and available bandwidth, the next section will study the effect of different parameter configurations in order to find an optimal point of operation.

B. Network Discovery Time and Reliability

Due to the battery powered nature of the targeted LPWAN devices, an optimal point should be chosen where the device consumes the least possible energy, nevertheless an acceptable latency is maintained. In this sense, two types of latency can be distinguished: 1) the *network discovery time*, which is the maximum time before a better network is detected and 2) the *reliability latency*, which is the time before the device will notice a disconnection from the current network. While the polling threshold will directly impact the network discovery



(a) (b) (c)

Fig. 10. By varying the threshold parameters, the energy consumption, network discovery time and reliability latency are affected. (a) Downlink threshold = 1. (b) Downlink threshold = 5. (c) Downlink threshold = 10.

time, both the polling and downlink thresholds may affect the reliability latency.

1) Network Discovery Time: Fig. 10 indicates how a lower polling threshold, i.e., the intertechnology handover check, does not necessarily mean that more energy will be consumed. For the worst-case scenario, the heterogeneous device is not able to connect to a lower energy, higher bandwidth network, which will result in a higher energy consumption. However, the higher the probability that the device succeeds in connecting to such networks, the advantage of these low energy networks comes through.

Therefore, once a device enters an area where a higher bandwidth network is available, it should connect to it as fast as possible to make use of the improved characteristics of the network. The maximum latency (in seconds) with which it may discover a higher bandwidth network is the maximum *network discovery latency* ($S_{\max}$). The *network discovery latency* depends on the `polling_threshold` (pt) of that technology, as this determines after how many consecutive messages a network can be discovered, and the transmission frequency (f) in Hz and can be modeled as

$$S_{\max} = \frac{1}{f} \cdot \text{pt}. \tag{7}$$

2) Reliability Latency: As a downlink from a discovered network also improves reliability (i.e., piggybacked information about previous transmissions), both thresholds [pt and `downlink_threshold` (dt)] should be taken into account while modeling the minimum *reliability latency* ($L_{\min}$)

$$L_{\min} = \frac{\text{pt} + \text{dt} - |\text{pt} - \text{dt}|}{2f}. \tag{8}$$

However, if the device is not able to discover another network, the maximum *reliability latency* ($L_{\max}$) is equal to

$$L_{\max} = \frac{1}{f} \cdot \text{dt}. \tag{9}$$

The maximum *reliability latency* and *network discovery latency* are modeled in Fig. 10 for various configurations of the threshold parameters. It can be seen that the *reliability latency* is equal to the threshold with the lowest value until the `polling_threshold` exceeds the `downlink_threshold`. From then on the latency will be equal to $L_{\max}$. On the contrary, it is clear how a higher `downlink_threshold` will restrict the device in fast network discovery, as pt does not have an impact on the maximum *reliability latency*.

The *network discovery latency* is affected directly by the `polling_threshold` only, as for every configuration, a constant sending frequency of 10 min has been used.

Furthermore, it can be seen that the energy consumption of a device in all cases decreases when increasing the dt parameter. The constrained node will fire more uplink transmissions without asking for downlink confirmation, which will reduce the time on air, nevertheless with an increasing reliability latency as a consequence.

Another observation made is that an increasing pt forces the device to switch between their different network interfaces less frequently, however, increases the energy consumption. This can be explained that by increasing the frequency with which the device switches between network interfaces, the probability of finding a higher bandwidth network increases.

Finally, a reverse curve can be noticed in the worst-case scenario due to the fact that the device will try to detect other networks less frequently. In contrast to the previous observation, no probability influences the underlying network technology, which directly impacts the energy consumption. Also, the curve does not change by increasing the dt parameter, since the device will remain on the Sigfox network throughout the whole simulation.

C. Configuration

From the previous section, it may be concluded that the different parameters have a large impact on the latency and energy consumption. Therefore, to have a better understanding when configuring the parameters of the algorithm, four extreme configurations were simulated 100 times with their distribution shown in Fig. 11.

This can be broken down as follows.

1) $pt : 1, dt : 1$—The most efficient configuration, i.e., the configuration with the lowest *reliability latency* and the fastest *network discovery time*. The device will poll for a better network and will ask confirmation about the current network after every uplink transmission. This, however, is impractical in large networks due to the gateways' duty cycle restrictions.

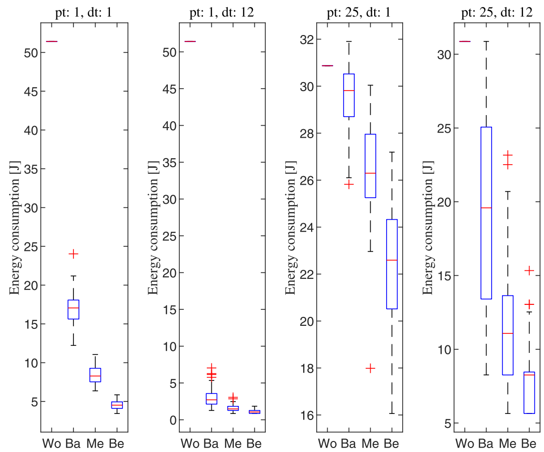


Fig. 11. Comparison between different configurations for the vertical handover algorithm with N set to 5 for 24 h.

2) $pt : 1, dt : 12$—The multimodal device will only ask for acknowledgments after 12 uplink transmissions, and may, in a worst-case scenario, only notice a disconnection from the network after 12 uplink transmissions. However, the probability of connecting to a better network is high, which increases the *reliability latency*.
3) $pt : 25, dt : 1$—The device will notice immediate disconnection from the network, however, once connected to the network with the highest energy consumption and lowest data rate, it will only try to discover a higher bandwidth network after 25 uplink transmissions, which affects the distribution across the different cases significantly.
4) $pt : 25, dt : 12$—The median energy consumption decreases, since fewer downlink transmissions are requested, however, the distribution increases due to the increased `downlink_threshold`, which will result in less stable network detection.

From this, it can be concluded that by using the most efficient (in terms of latency) configuration (i.e., pt : 1, dt : 1), energy might be saved when there is a high probability of connecting to a lower energy hungry network. However, due to duty cycle regulations, it is not possible for gateways to respond to each message of every device. Therefore, the threshold parameters should be set as high as possible. Nevertheless, this decreases the *reliability latency* and increases the *network discovery time*. Hence, certain device conditions require different configurations, which might be solved by signaling updated parameter thresholds to the device or update the threshold configuration based on sensor information.

D. Reliability Overhead

Some use cases require reliable communication and cannot afford missing a maximum of $f \cdot L_{\max}$ messages. In order to achieve reliable communication, worst-case, the total number of unacknowledged messages that have the possibility to be lost, must be retransmitted using Sigfox. Therefore, as depicted in Fig. 12, the maximum overhead when adding reliability is the total number of unacknowledged messages multiplied by the energy required for an uplink using Sigfox. When compared to Fig. 11, it is clear that when reliability is required, a higher dt has a higher probability in consuming less energy.

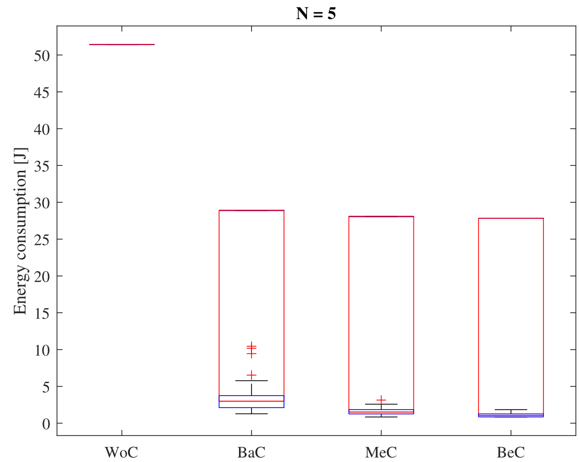


Fig. 12. Maximum energy overhead while safeguarding reliability.

TABLE VII
CODE SPACE REQUIRED FOR THE DIFFERENT COMPONENTS

	LPH	*p*CoAP	μIP	SCHC	**Total**
RAM	538 B	300 B	409 B	1589 B	2836 B
ROM	1968 B	1858 B	1622 B	16030 B	21478 B

E. Implementation Overhead

As constrained devices have only a limited amount of memory (<50-kB RAM and <256 kB of ROM), it is important to keep memory consumption to a minimum. Therefore, the different components of the proposed system are presented in Table VII with their corresponding overhead. The values were listed and processed using the ARM `readelf` tool, which performs a similar function to `objdump`.

It can be seen from the table that the low power handover (LPH) algorithm requires 538 bytes of RAM, mainly required for message buffers and to keep track of the different network drivers, as explained in Section VI-B2. The state machine itself requires almost 2 kB. In order to have a fully working protocol stack, an adapted version of Adam Dunkels' μIP library is used, where TCP support can be disabled. This adds 1622 bytes of ROM and 409 bytes of RAM. Furthermore, the *p*CoAP library has been integrated to achieve minimal CoAP support and to (de)compress SCHC CoAP packets, resulting in 1858 kB of ROM. Only a limited amount of memory is required for the CoAP stack, since this implementation does not contain more complex CoAP mechanisms.

It can be seen that the implementation overhead of the different components is kept to a minimum in order to target constrained devices.

VIII. CONCLUSION

In this article, a novel network detection algorithm for selecting the best available LPWA technology in a heterogeneous network was presented. To the best of our knowledge, no algorithms are currently available for the LPWAN network detection and a novel algorithm and architecture were, therefore, presented.

In order to have a single stack over multiple technologies, SCHC was presented and used as an adaptation layer below a CoAP/UDP/IPv6 stack in order to enable a standard-based, Internet compliant IoT. We showed that some technologies, however, provide a full vertical stack, resulting in a lot of overhead.

Next, in our evaluation, it comes forward that a multimodal device and the presented algorithm will help to save energy using a correct configuration compared to a homogeneous Sigfox device and some LoRa configurations. Apart from these energy savings, we enable OTA updates for a Sigfox device, which by itself is only capable of receiving 32 bytes every day.

Finally, it is shown that the total implementation overhead is kept to a minimum in order to cope with the constrained IoT device requirements.

A. Future Work

Since the ranking of the presented technologies was straightforward, the current implementation of the algorithm uses a priority-based list of ranked radio access technologies. However, when adding more diversity, such as cellular-based technologies, to the platform, a more sophisticated multiple attribute decision making (MADM) algorithm might be used, such as simple additive weight (SAW), technique to order preference by similarity to ideal solution (TOPSIS), or weighted product model (WPM) [5].

Furthermore, heterogeneous devices can make use of the same approach to distribute traffic over different channels. Since duty cycle regulations apply per channel, devices transmitting less frequently can be moved to a channel with a lower duty cycle. The duty cycle check component should then be adjusted in order to keep track of the imposed limitations per channel.

The signaling protocol was introduced only briefly but does offer great possibilities. An algorithm could run in the back-end which could predict the trajectory of a device and reply over the signaling slots with updated parameters of the technologies. Very predictable as well as more unpredictable trajectories could be recognized by the machine learning algorithms, which could take advantage of this in order to reduce the energy consumption even more. Also accelerometer data (or other sensor input) might be used to check whether the device is moving or not. Movement of the device might trigger a different parameter configuration in order to search more active for other networks, as a stationary device is presumably in the presence of high bandwidth networks while a moving device might only have access to very long-range networks.

Finally, the algorithm has been built in such a way that other technologies can easily be added. The current evaluation showed the combination of an uplink-oriented protocol (SigFox) for basic connectivity complemented with intermittent, lower range, higher data-rate connectivity (LoRa, DASH-7). We shortly introduced other protocols, such as Weightless-N for uplink-oriented connectivity and NB-IoT for a higher data rate, low latency connection. Other scenarios can be investigated in future work, where, for example, a combination of Bluetooth low energy (BLE) and LoRa or NB-IoT and LoRa might be interesting to move away from a deploy-and-forget scenario, where constrained devices can be updated, by taking advantage of the licensed and the unlicensed spectrum. Furthermore, by making use of different types of configurable thresholds, a tradeoff is offered between energy preservation, resilience, and flexibility.

ACKNOWLEDGMENT

MuSCLe-IoT is a project realized in collaboration with imec, with project support from VLAIO (Flanders Innovation and Entrepreneurship). Project partners are imec, Flash Private Mobile Networks, Engie M2M, Sensolus, and Aertssen.

REFERENCES

[1] K. Mekki, E. Bajic, F. Chaxel, and F. Meyer, "A comparative study of LPWAN technologies for large-scale IoT deployment," *ICT Exp.*, vol. 5, no. 1, pp. 1–7, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2405959517302953

[2] M. Weyn, G. Ergeerts, R. Berkvens, B. Wojciechowski, and Y. Tabakov, "DASH7 alliance protocol 1.0: Low-power, mid-range sensor and actuator communication," in *Proc. IEEE Conf. Stand. Commun. Netw. (CSCN)*, Oct. 2015, pp. 54–59.

[3] R. Berkvens, B. Bellekens, and M. Weyn, "Signal strength indoor localization using a single DASH7 message," in *Proc. Int. Conf. Indoor Position. Indoor Navig. (IPIN)*, Sapporo, Japan, Sep. 2017, pp. 1–7. [Online]. Available: http://ieeexplore.ieee.org/document/8115875/

[4] *Murata Partners With STMicro to Add SigFox Connectivity to Its LoRaWAN Module Enabling 'Best of Both Worlds' Capability'*, Murata Manufacturing Co., Ltd., Nagaokakyo, Japan, 2017. [Online]. Available: https://www.murata.com/en-global/products/info/connectivitymodule/lpwa/2017/0718

[5] F. Bendaoud, M. Abdennebi, and F. Didi, "Network selection in wireless heterogeneous networks: A survey," *J. Telecommun. Inf. Technol.*, vol. 4, pp. 64–74, Jan. 2019. [Online]. Available: https://www.itl.waw.pl/czasopisma/JTIT/2018/4/64.pdf

[6] W.-T. Chen, J.-C. Liu, and H.-K. Huang, "An adaptive scheme for vertical handoff in wireless overlay networks," in *Proc. 10th Int. Conf. Parallel Distrib. Syst. (ICPADS)*, Jul. 2004, pp. 541–548.

[7] J. Hoebeke *et al.*, "A cloud-based virtual network operator for managing multimodal LPWA networks and devices," in *Proc. 3rd Cloudification Internet Things (CIoT)*, Jul. 2018, pp. 1–8.

[8] D. Castells-Rufas, A. Galin-Pons, and J. Carrabina, "The regulation of unlicensed sub-GHz bands: Are stronger restrictions required for LPWAN-based IoT success?" *arXiv:1812.00031*, p. 17, Nov. 2018.

[9] P. Wetterwalk, P. Thubert, and E. Levy-Abegnoli, *Heterogenous Wireless: Dual LoRa-NB-IoT Redundant Connectivity*, Tech. Disclosure Commons, Jun. 2018. [Online]. Available: https://www.tdcommons.org/dpubs_series/1278

[10] M. Chen, Y. Miao, X. Jian, X. Wang, and I. Humar, "Cognitive-LPWAN: Towards intelligent wireless services in hybrid low power wide area networks," *IEEE Trans. Green Commun. Netw.*, vol. 3, no. 2, pp. 409–417, Jun. 2019. [Online]. Available: http://arxiv.org/abs/1810.00300

[11] F. Lemic, A. Behboodi, J. Famaey, and R. Mathar, "Location-based discovery and vertical handover in heterogeneous low-power wide-area networks," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10150–10165, Dec. 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8804217/

[12] U. Raza, P. Kulkarni, and M. Sooriyabandara. (Jun. 2016). *Low Power Wide Area Networks: An Overview*. [Online]. Available: https://arxiv.org/abs/1606.07360

[13] J. Haxhibeqiri, E. De Poorter, I. Moerman, and J. Hoebeke, "A survey of LoRaWAN for IoT: From technology to application," *Sensors*, vol. 18, no. 11, p. 3995, Nov. 2018. [Online]. Available: http://www.mdpi.com/1424–8220/18/11/3995

[14] L. Alliance. (2017). *LoRaWAN 1.1 Specification*. [Online]. Available: https://lora-alliance.org/sites/default/files/2018–04/lorawantm_specification_-v1.1.pdf
[15] D. Alliance. (2017). *DASH7 Alliance Specification V1.1*. [Online]. Available: https://dash7-alliance.org/download-specification/
[16] P. Ruckebusch, S. Giannoulis, I. Moerman, J. Hoebeke, and E. De Poorter, "Modelling the energy consumption for over-the-air software updates in LPWAN networks: SigFox, LoRa and IEEE 802.15.4g," *Internet Things*, vols. 3–4, pp. 104–119, Oct. 2018. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2542660518300362
[17] J. Finnegan and S. Brown, "A comparative survey of LPWA networking," *arXiv:1802.04222*, Feb. 2018. [Online]. Available: http://arxiv.org/abs/1802.04222
[18] L. Feltrin *et al.*, "Narrowband IoT: A survey on downlink and uplink perspectives," *IEEE Wireless Commun.*, vol. 26, no. 1, pp. 78–86, Feb. 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8641430/
[19] Z. Shelby, K. Hartke, and C. Bormann, "The constrained application protocol (CoAP)," IETF, Fremont, CA, USA, Rep. 7252, Jun. 2014. [Online]. Available: https://www.rfc-editor.org/info/rfc7252
[20] K. Hartke, "Observing resources in the constrained application protocol (CoAP)," IETF, Fremont, CA, USA, Rep. 7641, Sep. 2015. [Online]. Available: https://www.rfc-editor.org/info/rfc7641
[21] A. Bhattacharyya, S. Bandyopadhyay, A. Pal, and T. Bose, "Constrained application protocol (CoAP) option for no server response," IETF, Fremont, CA, USA, Rep. 7967, Aug. 2016. [Online]. Available: https://www.rfc-editor.org/info/rfc7967
[22] A. Minaburo, L. Toutain, C. Gomez, D. Barthel, and J. Zuniga, "Static context header compression (SCHC) and fragmentation for LPWAN, application to UDP/IPv6," IETF, Internet-Draft, draft-ietf-lpwan-ipv6-static-context-hc-21, Jul. 2019. [Online]. Available: https://tools.ietf.org/pdf/draft-ietf-lpwan-ipv6-static-context-hc-21.pdf
[23] B. Moons, A. Karaagac, J. Haxhibeqiri, E. De Poorter, and J. Hoebeke, "Using SCHC for an optimized protocol stack in multimodal LPWAN solutions," in *Proc. IEEE 5th World Forum Internet Things (WF-IoT)*, 2019, pp. 1–6. [Online]. Available: http://hdl.handle.net/1854/LU-8613162
[24] G. Castignani, A. E. A. Moret, and N. Montavont, "A study of the discovery process in 802.11 networks," *ACM SIGMOBILE Mobile Comput. Commun. Rev.*, vol. 15, no. 1, pp. 25–36, Jan. 2011. [Online]. Available: https://hal.archives-ouvertes.fr/hal-00609309
[25] M. Loy and R. Karingattil, *ISM-Band and Short Range Device Regulatory Compliance Overview*, Texas Instrum., Dallas, TX, USA, 2005.
[26] A. Dunkels. (Feb. 2001). *Design and Implementation of the lwIP TCP/IP Stack*. [Online]. Available: https://www.artila.com/download/RIO/RIO-2010PG/lwip.pdf
[27] *137 MHz to 1020 MHz Low Power Long Range Transceiver*, Semtech, Camarillo, CA, USA, Aug. 2016. [Online]. Available: https://www.mouser.com/ds/2/761/sx1276–1278113.pdf

Bart Moons received the master's degree in electronics and ICT from the University of Antwerp, Antwerp, Belgium, in 2017, with specialization in ambient technology, where he had the opportunity to work on reconfigurability and reprogrammability aspects of low end wireless sensor network devices in the Internet of Things as part of his master thesis. He is currently pursuing the Ph.D. degree with the Internet and Data Laboratory Research Group, Department of Information Technology, Ghent University, Ghent, Belgium.

His current research interests include the Internet of Things (IoT), heterogeneous wireless networks, the Web of Things and standard-based networking in the IoT.

Abdulkadir Karaagac received the master's degree in communication systems from EPFL, Lausanne, Switzerland, in 2013. He is currently pursuing the Ph.D. degree with the Internet and Data Laboratory Research Group, Department of Information Technology, Ghent University, Ghent, Belgium.

His current research interests include the Internet of Things and wireless networks, with a focus on mobile and wireless connectivity, robust wireless communication, network diagnosis, and interoperability.

Eli De Poorter received the master's and Ph.D. degrees in computer science from Ghent University, Ghent, Belgium, in 2006 and 2011, respectively.

He is a Professor with Ghent University, Ghent, Belgium, where he is a Coordinator of several national and international projects. Since 2017, he has been also affiliated with IMEC Research Institute, Leuven, Belgium. He has authored or coauthored more than 100 papers published in international journals or in the proceedings of international conferences. His main research interests include wireless network protocols, network architectures, wireless sensor and *ad hoc* networks, IoT, indoor localization and self-learning networks, with a strong focus on experimental wireless testbed research.

Jeroen Hoebeke received the master's and Ph.D. degrees in computer science from Ghent University, Ghent, Belgium, in 2002 and 2007, respectively.

He is a Professor with the Internet Technology and Data Science Lab, Ghent University, Ghent, Belgium, and imec, Leuven, Belgium. He is conducting research on wireless communication solutions for the Internet of Things (IoT). He investigates how open standards can be used to roll out connected devices and easily integrate them in IoT applications. He has been active in several IoT domains, such as smart building, logistics, healthcare, industry 4.0, etc. He has authored or coauthored more than 100 publications in international journals or conference proceedings.