This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2020.2985925, IEEE Internet of Things Journal

Mobility Management with Session Continuity during Handover in LPWAN

Wael Ayoub*†, Fabienne Nouvel*, Abed Ellatif Samhat†, Mohamad Mroue† and Jean-christophe Prévotet*
*Institut National des Sciences Appliquées de Rennes — IETR-INSA, Rennes, France.
† Faculty of Engineering - CRSI, Lebanese University, Hadath Campus, Hadath, Lebanon.

Email*: firstname.lastname@insa-rennes.fr
Email†: samhat@ul.edu.lb, mohamad.mroue@ul.edu.lb

***Abstract*—In this paper, we consider the mobility of an end device (ED) in low-power wide area networks (LPWAN) and we focus on the continuity of the ED session with the application server when this ED is moving from the coverage of one operator to that of another one. One of the methods to achieve the continuity of the session after roaming is the use of an IPv6-based scheme. As LPWAN is characterized by a fixed message rate and very small bandwidth as well as asymmetric communication, an efficient header compression scheme is required. Recently, the IETF LPWAN working group has proposed static context header compression (SCHC) to compress IPv6 into LPWAN through a rule-based mechanism. In this work, we first extend SCHC scheme to support session continuity and the new scheme is called MSCHC (Mobile SCHC). MSCHC consists of several contexts, instead of the static context for SCHC and we also improve the use of the memory by dividing the rules into layers. Then, we investigate the mobility of the ED to show how continuity of the session can be achieved while transmitting and receiving data when the ED is roaming between different operators. The proposed solution is based on the use of light mobile IPv6 messages compressed with MSCHC. Finally, the proposed mechanism is implemented in the popular LoRaWAN technology, evaluated and compared with the existing solutions provided by the LoRaWAN v1.1 standard.**

***Index Terms*—IoT communication, LPWAN, LoRaWAN, MIPv6, Long-Range, Mobility, Roaming, MSCHC.**

I. INTRODUCTION

There will be more than 20 billion of smart things connected to the Internet by the year 2020 [1]. The vast majority of these devices will be connected by Wide Area Networks (WANs) technologies. In addition to conventional cellular networks, new technologies are gaining a significant portion of the market in the recent years such as LoRaWAN [2], NB-IoT [3], etc. LPWANs support the deployment of a massive number of smart devices under one gateway. This deployment is increasing exponentially to include more IoT applications from several domains such as intelligent infrastructure, transportation, industrial, retail, etc. Today, Internet protocol is required for Internet connection [4], and the limitation of IPv4 address space necessitates the transition to IPv6. It is generally recognized that the core network will be IPv6-based for the next generation of the Internet [5]. Recently, extensive studies have attempted to integrate IPv6 into heterogeneous networks including low energy technologies with resource constraints, such as LPWAN as in IoT6 approach [6]. IPv6 offers many benefits to IoT, and its completion is only a matter of time [7]. However, it can not be applied directly to LPWAN technologies even with the use of fragmentation as shown in [8]. Note that the LPWANs challenges include low data rate and small bandwidth in addition to low power consumption and radio constraints.

To address the adoption of IPv6 for LPWAN and to meet the challenges of these technologies, the Internet Engineering Task Force (IETF) created the LPWAN working group (WG) in 2016. This WG has proposed a new mechanism, the Static Context Header Compression (SCHC), that is used to compress the IPv6 header to run over LPWAN technologies. Moreover, the WG has extended the mechanism to compress UDP/IPv6 [9], and Constrained Application Protocol (CoAP) header [10]. SCHC mechanism is based on a static context shared between the compression and decompression units at both end-device (ED) and network sides. For that, the mechanism avoids complex synchronization. It supposes that context is static and does not change with time, which is the case of static device. SCHC transforms IPv6, and UDP headers into a few bits, called "RuleID", by removing known and duplicate information. All the rules present in the ED context must also be in the context of the Network Server (NS). That allows to reduce the network costs, save bandwidth, increase the transmission rate, and decrease the time required to transmit a packet, using low energy consumption.

However, mobility is a requirement in many IoT applications such as smart cities [11], health-care [12], smart vehicles [13], aging society [14], hospital [15], etc. Although SCHC is an efficient mechanism used to adopt IPv6 for LPWAN, but IPv6 protocol main header does not support ED mobility and switching between different gateways (GW)s. The issue of mobility management in LPWAN will be investigated in this work. In this paper, we consider the roaming of EDs supporting IPv6 between two different NS operators while achieving the continuity of the session with the Application Server (AS). As SCHC only proposed a static context for EDs, we propose an extension of SCHC mechanism, named as Mobile SCHC (MSCHC). MSCHC will consist of several contexts, instead of only one for SCHC. Then, we propose a mobility management solution based on a light mobile IPv6 messages compressed with MSCHC. This solution achieves the continuity of the session when the ED is roaming between different operators while transmitting and receiving data.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2020.2985925, IEEE Internet of Things Journal

The rest of this document is organized as follows: First, we present the related work in section II. Then, in section III we present our contribution, the implementation of MSCHC in the LPWAN architecture and explain our proposal to support mobility while roaming. In section IV, we present another contribution in terms of the use of mobile IPv6 protocol and how to divide the fields into different layers that MSCHC supports. In section V, we provide a summary of the LoRaWAN standard, the class support including our proposed updates, the different roaming mechanisms and the steps that an ED follows to achieve connectivity based on our proposed mechanism. In section VI, we illustrate our proposed mobility management mechanism using lightweight MIPv6 in MSCHC. Then, in section VII, we provide an implementation of the two solutions, passive and handover, proposed by the LoRaWAN v1.1 standard and we compare them with our contribution. Finally, Section VIII provides a conclusion and the future work.

II. Related work

Since 2003, IP features motivate researchers to adopt it even with constrained devices such as the microcontroller [16]. The actual prospects of adopting IPv6 in IoT [17], can be listed as follows:

IPv6 Adoption for IoT: It provides IoT devices with direct and easy access to control. IPv6 is the addressing scheme for any data transfer on the Web. The limited size (32-bit address space) of its predecessor, IPv4, made the transition to IPv6 inevitable. According to Google, IPv6 adoption rate follows an exponential curve, which doubles about every 6 months [–google$_2$019].

Address self-configuration: IPv6 provides an automatic address configuration mechanism (stateless mechanism). The ED can set their addresses autonomously. This can greatly reduce the effort and cost of the configuration.

Scalability: IPv6 offers a highly scalable address scheme. It provides more than 2 billion addresses per square millimeter of the Earth's surface [17]. It is enough to meet the needs of any present and future communication device especially IoT.

IPv6 solves the Network address translation (NAT) barrier: NAT users borrow and share IP addresses with others. As a result, they do not have their own public IP address, which makes them homeless Internet users. They can access the Internet, but are not directly accessible from the Internet as the case of current IoT technologies. Furthermore, it breaks the original end-to-end connection and greatly weakens any authentication process.

Enabling the extension of the Internet to the web of Things: Because of its large address space, IPv6 allows the extension of Internet to any device and service. Experiments have demonstrated the successful use of IPv6 addresses for large scale deployment of sensors in smart buildings, smart cities and even agriculture and livestock [17]. In addition, the CoAP protocol allows the constrained devices to behave as easily accessible web services that fully comply with the REST architecture.

Fully Internet compliant: IPv6 is fully compatible with the Internet. It is possible to use a global network to develop an own network of smart objects or interconnect own smart devices with the rest of the world.

Strong Security enablers: IPv6 provides end-to-end connectivity, with a more distributed routing mechanism. In addition, IPv6 is backed by a large community of users and researchers who support a continuous improvement of its security features.

Mobility and interoperability: IPv6 specifies multiple protocols that allow EDs to remain accessible when moving. Each mobile ED is identified by its home address, regardless of its current Internet connection point. Although it is also associated with a care-of-address, that provides information about the current location. IPv6 packets directed to the home address of a mobile ED are routed transparently to their care-of-address. The IPv6 extension protocols allow the EDs to cache the link of a home mobile ED's address with its care-of-address and then send any packet destined for the mobile ED directly to that care-of-address. All IPv6 EDs, whether mobile or stationary, can communicate with a mobile ED.

Tiny stacks available: IPv6 adaption on IoT has been researched for few years. The research community has developed compressed versions of IPv6 such as 6LoWPAN, SCHC, etc. These compression techniques are simple and efficient mechanisms to shorten the size of the IPv6 address for constrained devices. Thus, network operators servers can translate those compressed addresses into normal IPv6 addresses. But these solutions still lack roaming support for mobile EDs.

Thus IPv6 offers several features and arguments that demonstrate its success in the future of IoT [8]. In the following, we analyze the previous and current solutions that have been developed to run IPv6 on constrained IoT devices and investigate their applicability with LPWAN technologies.

In [18], 6LoWPAN was launched at the end of 2004 by the IETF WG to activate IPv6 on IEEE802.15.4 Networks. In [19–21], authors use 6LoWPAN to compress IPv6 header to fit in the LPWAN frame. 6LoWPAN was extended into 6Lo [22] in 2013 with the same objectives. 6LoWPAN and 6Lo are an adaptation layer that allows constraining devices to send/receive data using IPv6. However, the two solutions are not applicable in the case of LPWAN technologies [23] because these technologies do not support L2 layer fragmentation. The maximum transmitted size is less than that considered by the two solutions. Furthermore, the communication in the LPWAN is asymmetric, and the uplink and downlink communications are not synchronized. Therefore, related works that adopt 6LoWPAN with IPv6 on constrained IoT technologies such as LoRaWAN, SigFox, etc. are not an optimal choices. Moreover, the region laws limit the messaging rate of such technologies (e.g., 50 messages per day only). In addition to 6LoWPAN and 6Lo, 6TiSCH was developed by the IETF WG "Time Slotted Channel Hopping" in 2013 to be a standard for low power industrial wireless monitoring applications [24]. This standard was a solution for mesh networks that avoid the collision by using time intervals. However, LPWANs are based on a star topology and non-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2020.2985925, IEEE Internet of Things Journal

synchronized communication.

The RESTful Constrained Environments (CORE) IETF WG has released the Constrained Application Protocol (CoAP) [25]. CoAP is a RESTful framework specified for constrained nodes that run on constrained IP networks. It is based on asynchronous request/response communication between the applications running on the devices. CoAP is a lightweight application layer protocol running over UDP which is suitable for low memory and low power devices. It supports Machine-to-Machine (M2M) communication and provides the application with seamless connectivity to the technology used on the underlying radio. However, the CoAP protocol does not support the main point of this article which is mobility during communication as shown in [26]. To solve this issue, in [26–28], the authors propose a mechanism to manage mobility in CoAP. However, the proposed mechanism requires a lot of messaging to synchronize and manage mobility, which exceeds the message rate limitation set on LPWAN technologies.

Static Context Header Compression (SCHC) [9] is a header compression scheme that supports L2 layer fragmentation. SCHC is designed for low bandwidth networks and constrained IoT standards that do not support L2 layer fragmentation. SCHC avoids the complexity of synchronization mechanisms and supports the level of fragmentation for LPWAN technologies. SCHC is very suitable for devices that have limitations in terms of power consumption, processing, and radio transmission. SCHC is an adaptation layer that is executed between L2 (Data Link) and L3 (Network) layers of the communication protocol stack to compress IPv6 / UDP headers into ruleIDs. However, [29] shows that SCHC is not efficient in memory usage for constrained devices. It considers static context for compression which does not allow mobility of devices. In [30], LSCHC is proposed to improve the use of memory. But it does not take into account the dynamic changes of headers during mobility. In [31], an enhancement mechanism has been suggested for processing dynamic data, but the mechanism is complex and consumes power and time. Moreover, a data management mechanism is needed to support the mechanism in [31] to ensure that the shared contexts between ED and NS are the same and sorted in the same way at the same time. Although mechanisms for IPv6 with IoT are proposed, improvements are necessary to consider mobility.

III. Proposed MSCHC solution within the LPWAN Architecture

The IETF LPWAN WG was recently established to investigate the adoption of IPv6 over constrained LPWAN networks. These networks are characterized by a low bandwidth, a low power consumption, and IoT standards that do not support L2 layer fragmentation. This WG has proposed the SCHC mechanism as a header compression scheme that covers the IPv6/UDP. Table I illustrates the specifications of the SCHC mechanism.

SCHC [30] has been proposed based on two initial conditions:

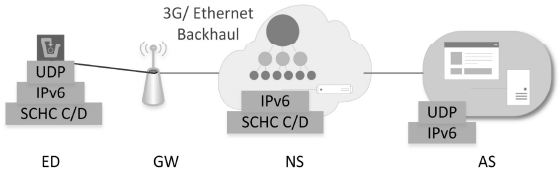

Fig. 1. SCHC Standard Communication Architecture

TABLE I
SCHC mechanism specifications

Specification	Static Context Header Compression (SCHC)
Standard	Starting in 2016 IEEE forms LPWAN WG
Work in Progress	Yes
Data Rate	Depend on the L2 layer of the technology used
Identity Header size	few bits - 2 bytes
Support Fragmentation	Yes
Place on Protocol stack	Adaptation layer below IPv6 and above Link Layer
Support Mobility	No
Synchronous	No
Communication	Asymmetrical

- Implementation based on star topology, as shown in Figure 1, where most LPWAN technologies share this typical architecture [8].
- Static context where the tasks remain unchanged.

The communication between the ED and the AS is asymmetric and bidirectional. It passes through the GW and NS as shown in Figure 1. The connection between ED and GW is wireless, it depends on the used IoT technology. Also, it is subjected to region laws, including message rate limitation and low bandwidth. However, the communication between GW and NS or NS and AS is not restricted and uses IP connections with any of the available technologies, i.e., LTE, Ethernet, Fiber, etc. Using IPv6 in the communication between ED and GW becomes possible when using an efficient compression mechanism that compresses the headers before transmitting on the constrained links of LPWAN technologies.

SCHC compression mechanism is formed of two parts: the first is performed at the client side, i.e., on the ED, the second is executed at the network side, i.e., on the NS. The network part is also responsible for managing all client part contexts and maintains a context for each ED. Thus, SCHC mechanism addresses such technologies and relies on a common static context shared between ED and NS. Moreover, the static context saved represents the headers that does not change over time or the changes are known. This concept allows SCHC to avoid the complex resynchronizations . Also, it allows the mechanism to use fewer resources than other stateful header compression schemes. Technically, SCHC layer compresses and decompresses the packet headers as shown in Figure 2. The contexts are described as ruleIDs that are

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2020.2985925, IEEE Internet of Things Journal

specified for each ED. Also, all rules that are present in the context of ED are also present in the context of NS. Therefore, UDP, and IPv6 compression headers can be combined and represented in one ruleID. However, authors in [30] show that this combination has many disadvantages in term of memory usage. As a solution, the authors proposed to classify alternative rules per layer. This solution reduces memory usage by removing redundancy of ruleIDs and allows the reuse of ruleID. However, this solution does not consider mobility or address the dynamic change of headers during mobility.

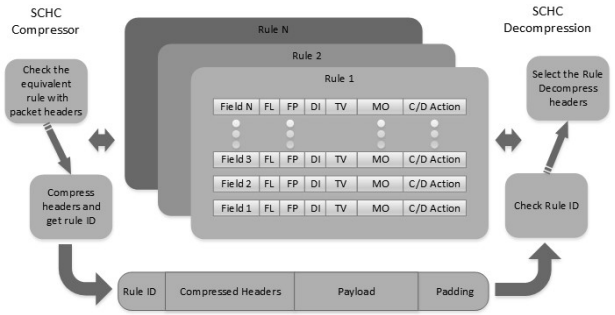

Fig. 2. SCHC Standard Implementation Architecture

To avoid memory waste and redundancy of contexts generated by SCHC mechanism, and to support session continuity after roaming, we propose the Mobile SCHC (MSCHC) technique. Our proposal consists of several contexts, rather than the unique context for the SCHC. Each context contains rules for a single layer. In MSCHC, the ruleID has been divided into segments; each segment represents a layer. The size of each segment can be defined based on LPWAN technology and the number of rules in each context. In this paper, we propose four layers for MSCHC instead of one as proposed in SCHC. As shown in Figure 3, the application layer context is used to compress the fields of the used application layer protocol. The same concept applies to transport and network layers. But, IPv6 at network layer does not support mobility. To solve this issue, we will use one of the extension protocols of the IPv6. Thus, extension headers will be added to the main IPv6 headers. Moreover, to avoid the drawbacks of SCHC that uses single ruleID for all headers, we decided to separate the extension headers into a new layer rather than adding fields to the network layer. The network layer compresses the main IPv6 headers, and the extension layer compresses the extension headers of IPv6. By splitting the rules between layers, each rule will be related to a single layer. The concatenation of the four ruleIDs of ALCs, TLCs, NLCs, and ELCs forms the RuleID, as shown in Figure 3. This RuleID will be sent / received during data exchange between ED and NS. Thus, the proposed solution MSCHC will saves memory, adds flexibility in selecting an appropriate rule on the compression side, and reduces the complexity of treatment at compression and decompression. However, MSCHC adds few bits in the constrained radio. This proposal allows mobility for constrained IoT technologies which was not possible using SCHC.

On the ED side, an application is running to send/receive data. When data are ready to send, ED uses MSCHC layer compression to select the most appropriate ruleIDs that fit the packet headers as shown in Figure 2. The ruleIDs that identify the headers are included into the "RuleID" part of the frame, and then the frame is transmitted. The data received from ED are forwarded to NS by the GW. Concerning MSCHC mechanism, the GW acts as a repeater. On the NS side, the MSCHC layer decompression module checks the received RuleID and ED identity, then the context corresponding to the ED is loaded. Then, the MSCHC decompression mechanism searches among the ruleIDs contexts in memory and reconstructs the headers of the original packet. Then, NS forwards the decompressed packet to the correspondent AS. If acknowledgment (ACK) is required, packets received by NS from AS are compressed using the context of the corresponding ED using the same procedure used on ED. Then, compressed packet is forwarded to the ED using a GW selected by the NS. Finally, ED decompresses the packet and reconstructs the original header.

MSCHC mechanism can be a new application added to the NS. This mechanism is an alternative to SCHC in routing and packet security [32]. It can be a stand-alone server that runs on the core network to compress/decompress received packets then returns them to NS. As will be shown below, the MSCHC mechanism does not add any complexity or processing in the ED. The mechanism is simply to work as a codebook to represent the headings in small size codes measured in bits. This codebook is stored in the ED and on a network operator server responsible for compressing / decompressing the headers.

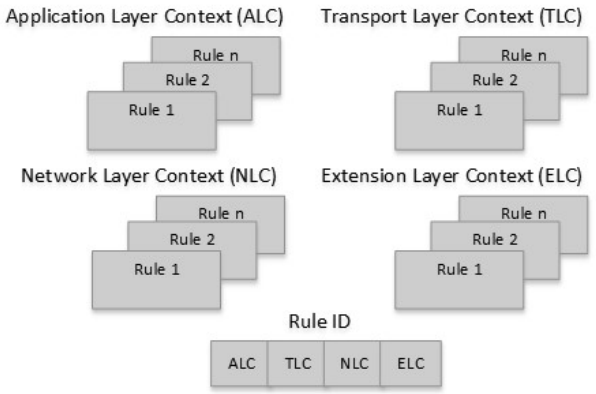

Fig. 3. Proposed MSCHC RuleID

The context shared between ED and NS consists of rules as shown in Figure 2. Each rule contains several fields where each one represents a parameter from a header. A field can be an array that contains specific information. A rule represents a compressed header, and each rule is defined using a ruleID. The ruleID is formed of:

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2020.2985925, IEEE Internet of Things Journal

- Field ID (FID): A unique value that defines the header field.
- Field Length (FL): Indicates the length of the header segment.
- Field Position (FP): States which instance must be selected if the same field has several instances in the header.
- Direction Indicator (DI): Indicates the packet direction, whether it is Uplink (Up) from ED to AS, Downlink (Dw) from AS to ED, or Bidirectional (Bi).
- Target Value (TV): Value to be compared with the headers present in the memory when a packet is received. These values can be of several types, (i.e., integer, string, arrays, lists, JSON and CBOR).
- Matching Operator (MO): Used to compare the target value, (i.e., values in memory with the received packet header).
- Compression Decompression Action (C/D): describes the method to be used during compression and decompression.

MSCHC mechanism follows different steps when compressing and decompressing as shown in Figure 2. The compression process starts by checking the headers of the packet and comparing them with the context saved in memory. Each context is referred to a layer and saved under a ruleID. The most suited context that matches the header values of the packet is selected. Then, the header values are represented by a ruleID. In the following, we will explain the procedure of the MSCHC mechanism in comparing context and header fields to select a ruleID.

The MSCHC mechanism procedure starts by loading all the contexts corresponding to the header that must be compressed. First, MSCHC compressing function checks the direction of the packet to verify the compatibility between the packet header and the ruleID context. In case of direction mismatch, MSCHC compressing function moves to another ruleID context. Second, each value in the packet header is compared with its TV in the context saved using the corresponding MO. If all parts of the packet header meet the same operators, the packet header values will be processed based on the relevant C/D procedures. Otherwise, MSCHC compressor function checks the next ruleID context. When more than one ruleID matches the packet header fields, MSCHC compressor function selects the ruleID that results with the least header compression. Finally, MSCHC mechanism concatenates ruleIDs in the "RuleID" field followed by the compressed headers then the payload. Padding bits are added if the datagram is not multiple of 8 bits. The size of RuleID is not fixed, it depends on the implementation, number of streams, and LPWAN technology used when deploying MSCHC. At best, the MSCHC mechanism can select a ruleID that does not require any compression header. In this case, the packet header is equal to the ruleID size only. At NS side, the ruleIDs are specific for each ED and to identify the correct ruleIDs, NS combines the ruleID with the L2 level identifier, Dev-EUI in LoRaWAN, which is usually represented by the MAC address of the ED.

On the other side, the frame is extracted, then the MSCHC decompression function identifies the ruleIDs used in the compression. Finally, MSCHC applies the C/D actions to reconstruct the original header. When no rule satisfies the verification in the compression function, the packet will not be compressed. In such case, the alternative solution is to use the fragmentation mechanism that SCHC supports.

IV. IPv6 Extension Protocol Selection to Support Mobility with MSCHC Layer

Different IPv6-based mobility mechanisms have been proposed as described in [33]. Our requirements behind selecting one of these protocols are: session continuity at the transport layer, simple tool to adopt, and latency acceptable up to 10 seconds. From all these mechanisms, we adopt a light implementation of the MIPv6 protocol [34] as the most appropriate solution that can fit and support mobility in LPWANs with minimum updates. Even though in IoT, network-based Handover (HO) mechanisms are preferred over host-based ones. But in LPWAN architecture, ED and NS can be seen as one host to AS. Furthermore, the vast overhead generated by MIPv6 is compressed using MSCHC. Thus, the main disadvantage of MIPv6 is compensated. On the other hand, the latency generated by MIPv6 HO messages is not important since we are not dealing with real-time applications. Thus, we grant ourselves a latency range of up to 10 seconds that is equivalent to the random access process in NB-IoT technology [35].

In general, IPv6 is divided into two distinct headers: Main/Regular IPv6 Header and IPv6 Extension Header to support mobility [36] as shown in Figure 4.

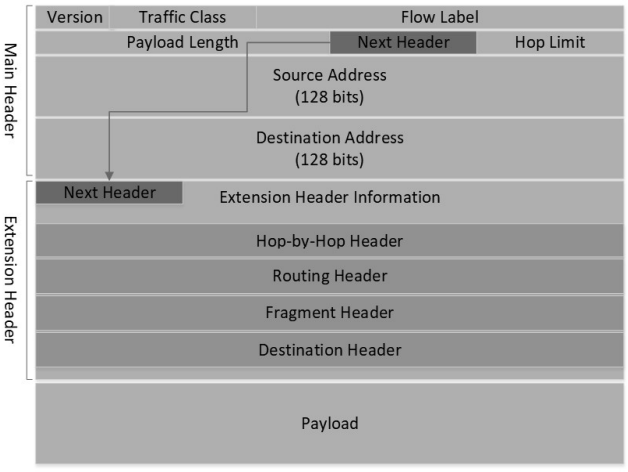

Fig. 4. MIPv6 main Header and Extension Fields

The main fields that support mobility in the IPv6 packet are the two extension header:

- Destination Header: This header is used by the ED to directly send a packet to AS using the new IPv6 address. When ED moves from a network to another, the new network assigns a new IPv6 address known as care-of-address (CoA) to the ED. In the IPv6 main header,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2020.2985925, IEEE Internet of Things Journal

TABLE II
EXAMPLE OF MAPPING IPV6 INTO MSCHC RULE

Rule 0						
Field	FP	FL	DI	TV	MO	C/D Action
Version	1	4	UP/BI	6	equal	not-sent
DiffServ	1	8	UP/BI	0	equal	not-sent
FL	1	20	UP/BI	0	equal	not-sent
Length	1	16	UP/BI		ignore	comp-length
NH	1	8	UP/BI	17	equal	not-sent
HL	1	8	UP/BI	255	ignore	not-sent
ED addr.	1	128	UP/BI	:::3:1::1	equal	not-sent
AS addr.	1	128	UP/BI	:::2:1::1	equal	not-sent

ED sets the source address to the CoA, while in the Destination Header, ED sets the original address, which is known as home address (HoA).

- Routing Header: This header is used by the AS to send a packet to the ED using the CoA directly. AS sets the destination address field in the main IPv6 header to the CoA of ED, and the HoA of ED will be set in the Routing Header.

In Table II, we illustrate how we compress the different IPv6 header fields and show the mapping using MSCHC. MO field indicates the operator that will be used by the MSCHC C/D to compare between the header and TV. These operators can be used for different types of data such as integers, strings, and structures.

- Equal: the value of the field must correspond to that of the TV.
- Ignore: the TV ignored, and the received value is always true.
- MSB (length): true if the most significant bits of the length field are the same as those in the TV.
- Match-mapping: true when the field value matches to one of the values of the TV.

Once a rule is selected, all MO are given a positive result followed by compression and decompression operations. In the following, we illustrate the actions that can be found in the action field of Table II, with a description for each. The possible C/D actions are:

- Not-sent: value not sent, and decompression operation extracts the value from TV.
- Value-sent: at compression operation, data are sent in the compression header after the "RuleID" field. The decompression operator decompresses the headers, and checks the MO.
- Mapping-sent: used to send an index value that refers to one of the values in the TV described in the form of a list. The decompression function uses the index and recovers the value from the TV.
- LSB: When a part of the entire content of a packet field is already known, the Least Significant Bits (LSB) serve in sending the modified part and avoid sending all the content.

In section VI, we will describe the procedure of our lightweight MIPv6 compression with the LoRaWAN architecture to avoid repeating information. Before illustrating how the continuity of the session between ED and AS is achieved, we will briefly explain the roaming mechanisms supported by the LoRaWAN v1.1 standard between ED and NS when the network provider changes. Then, we will present our proposed mechanism with the required configurations on LoRaWANv1.1 standard. In section VII, we will provide a comparison between the LoRaWAN solutions and our proposed solution in a real implementation.

V. CONFIGURATIONS REQUIRED ON LORAWAN TO SUPPORT SESSION CONTINUITY AFTER HANDOVER

LoRaWAN is an open standard architecture developed by LoRa Alliance [37] to provide connectivity for an ED with its concerned AS as shown in Figure 5. LoRaWAN supports macro diversity, allowing ED to communicate with one or more GWs. That enables the mobility of ED within the same network operator between different GWs. The LoRaWAN link layer protocol manages the mobility. LoRaWAN layer runs above the LoRa physical layer. LoRaWAN defines three different classes, as shown in Figure 6, optimized for different usages depending on the IoT application requirements.

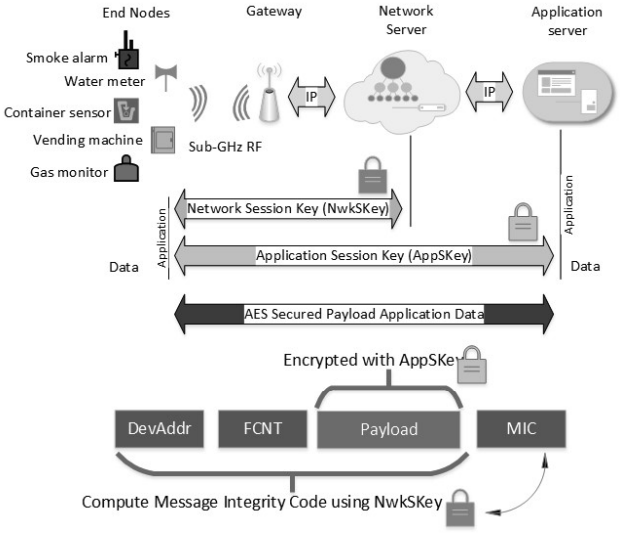

Fig. 5. LoRaWAN Architecture

A. LoRaWAN Classes

Figure 6 illustrates the operation mechanism for each class.

- Class A (Bi-directional EDs): is developed to stay most of the time in the sleeping mode. For that, this class is the most energy efficient one. This ED wakes up only if data is available to send. Following each transmission, two receiving windows RX1 and RX2 are opened, then ED returns to sleeping mode.
- Class B (Bi-directional EDs with scheduled receive slots): same as class A but these devices listen to incoming messages using RX2 on regular intervals synchronized with a beacon.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2020.2985925, IEEE Internet of Things Journal

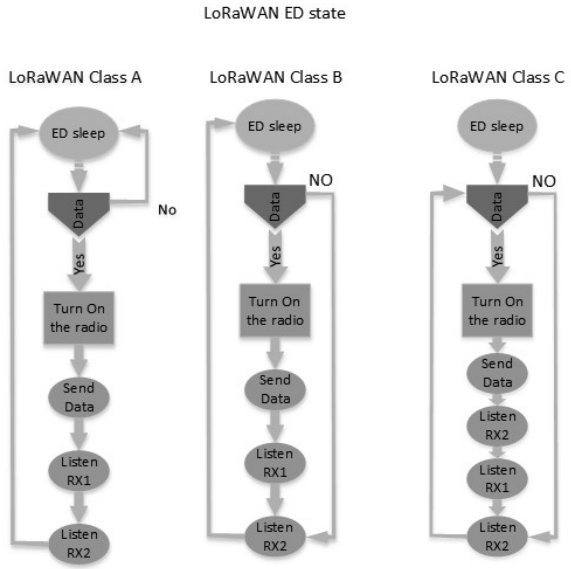

Fig. 6. LoRaWAN Classes

- Class C (Bi-directional EDs with maximal receive slots): ED continuously listens for incoming messages on RX2 unless transmitting. When power is not constrained this class can be used.

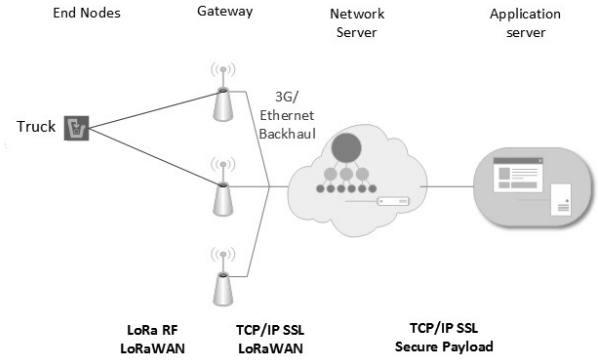

Fig. 7. Mobility under same network operator.

To meet with our proposal, class A will be reconfigured to have RX2 open before TX, while classes B and C remain unchanged.

B. Session Continuity

When dealing with roaming, ED movement may occur between different GWs under the Home NS of the network provider as shown in Figure 7. Otherwise, a movement may happen out of the Home NS to a foreign NS (Visited NS), using the same technology LoRaWAN, as shown in Figure 8. In our previous work [35], we showed the mobility management of ED between different GWs connected to a single NS. In this work, we consider the mobility between different NSs and how this mobility can be obtained without loss of session

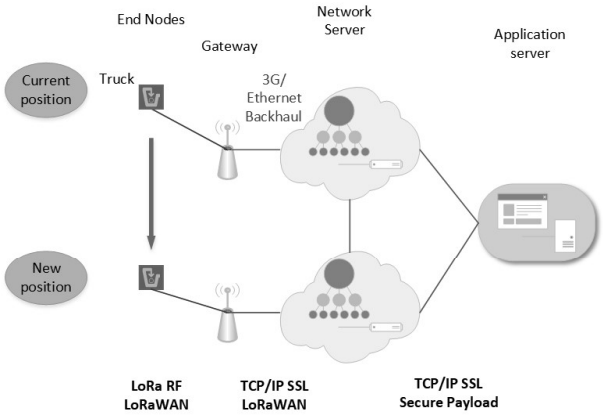

Fig. 8. Mobility between different network operators.

with AS. Mobility between two operators or NSs, as shown

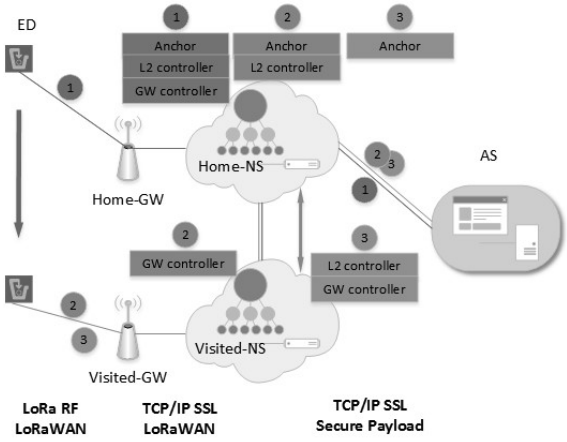

Fig. 9. Types of Roaming supported in LoRaWAN.

in Figure 8 has been addressed in LoRaWAN v1.1 [37] using passive handover. Until now, ED is attached to only one NS. ED cannot establish a connection with a visited NS unless the home NS allows this based on the standard laws. This issue is solved by collaboration between the NSs operators as we will explain below.

ED address is formed of two parts: the first part is the network identity (AddrPrefix), and the second part is the ED identity (NwkAddr). In LoRaWAN, if the device is activated over the air, roaming between different NSs can be achieved [35]. This means that activation over air supports mobility. But personal activation does not support mobility since the address and connection parameters cannot be modified over the air. In LoRaWAN, ED sends data in a broadcast manner without associating it with a GW since LoRaWAN supports macro-diversity. Eventually, the ED is only linked to the NS as shown in Figure 5 and, depending on the key of the network session, the NS decides whether or not to send the packet. In the case of roaming, NS will accept the uplinks received from

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2020.2985925, IEEE Internet of Things Journal

ED if they hold the AddrPrefixs of current and collaborative NSs.

Consider the movement of an ED out of the coverage of Home NS to a new area covered by GWs of collaborative NS. As shown in Figure 9 (see the blocks marked in red), if roaming is not supported, Home NS has three levels of responsibility:

- Anchor: Responsible for the communication with AS and to join the server.
- L2 controller: The LoRaWAN link layer holds all the functionality specified in the LoRaWAN specifications such as ADR management, device location, etc.
- GW controller: Responsible for PHY layer functions such as radio access network, the power of transmission, etc.

In the case of roaming, one of the two mechanisms can be achieved depending on the collaboration between the NSs operators. In passive roaming as shown in Figure 9 (see the blocks marked in green), the visited GW will forward the received uplink data to the visited NS which in turn forwards the packet to the Home NS of the ED. This roaming is transparent to ED. Visited NS is not responsible for any of the LoRaWAN L2 layer functionality such as control and management for this ED. In passive roaming, visited NS behaves as a GW for Home NS. It forwards the received packets from the ED without decapsulation. In handover roaming, as shown in Figure 9 (see the blocks marked in blue), ED needs to register with the visited NS and to obtain a new identity. After this roaming, the ED will be associated with with the visited NS, and the visited NS will be responsible of all LoRaWAN functionalities and control. Even though, the visited NS still sends the data to the Home NS that contains the Anchor functionality. Then, Home NS in turn sends the data to the corresponding AS.

Consider the handover roaming mechanism supported in LoRaWAN v1.1. In this mechanism, ED is registered with the visited NS. Therefore, visited NS has the functionality of LoRaWAN layer as shown in Figure 9 (blue blocks). The disadvantages of this mechanism are when considering the case where the visited NS and AS are in topologically related networks, while the Home NS is in another network. For that, in our proposed mechanism, the Anchor layer is moved to the visited NS for routing optimization. In our contribution, visited NS directly sends the data to AS without passing by Home NS to avoid loops. In addition to no roaming scenario, we discriminate between the three roaming scenarios: Passive and Handover Roaming without routing optimization proposed by LoRaWAN and our proposal Handover Roaming with routing optimization.

C. Connectivity Mechanism

During movement, ED follows different mechanisms to transmit the message to AS successfully. In each type of roaming, ED uses a different mechanism to achieve connectivity before delivering the information as shown in Figures 10 and 11.

1) No Roaming: In a normal situation as shown in Figure 10 under no roaming, ED uplinks message number one (Msg 1), and any Home GW, that receives the frame, forwards it to the Home NS. Upon receiving, the Home NS decapsulates the frame and sends the data to AS. Moreover, Home NS replies with an acknowledgment (ACK) to the ED. This ACK is forwarded by the nearest GW to the ED.

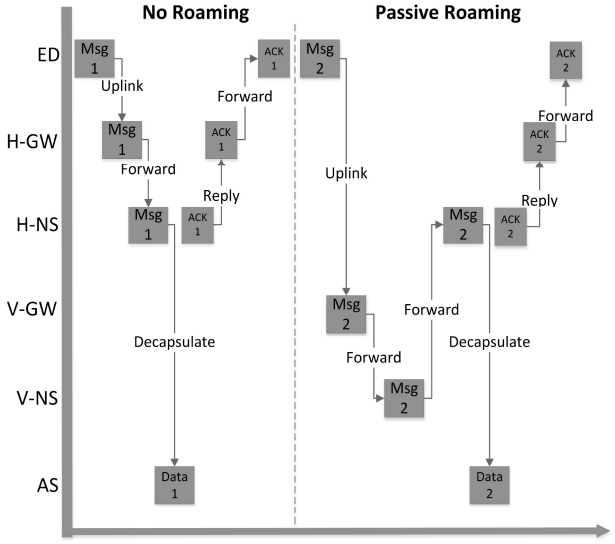

Fig. 10. Communication Mechanism in Passive Roaming

2) Passive Roaming: During the movement outside the Home NS coverage, the ED issues message number two (Msg2). As shown in Figure 10, under passive roaming, visited GW receives the message and forwards it to the visited NS. During identity verification, NS is notified that it corresponds to collaborative NS. Then, in turn, visited NS forwards the message to Home NS, which has the responsibility of the ED. At the Home NS, the received message is decapsulated and forwarded to the corresponding AS. If ACK is requested by ED, Home NS responds with an ACK to visited NS. At the visited NS, the closest GW to ED is selected and used to downlink the requested ACK.

3) Handover Roaming without Routing Optimization: During the movement, as shown in Figure 11, under Handover roaming without routing optimization, ED uplinks message number three and wait for ACK. Due to that the roaming supported by the partners of the home NS is handover roaming, the visited GW drops the message unless ED is registered to the visited NS. On ED, if ACK is not received, it deduces that it is out of the coverage of Home NS. Now, ED sends a join request to the visited NS. The join request is forwarded by the visited GW to the NS. Then, visited NS contacts the Home NS to release responsibility of the ED. After that, visited NS reactivates the ED with a new L2 identity. Then, ED re-sends message number three. The GW forwards the message to the visited NS. On the NS, the message is decapsulated and the content is sent to Home NS. Then, Home NS in turn delivers the data to AS.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2020.2985925, IEEE Internet of Things Journal

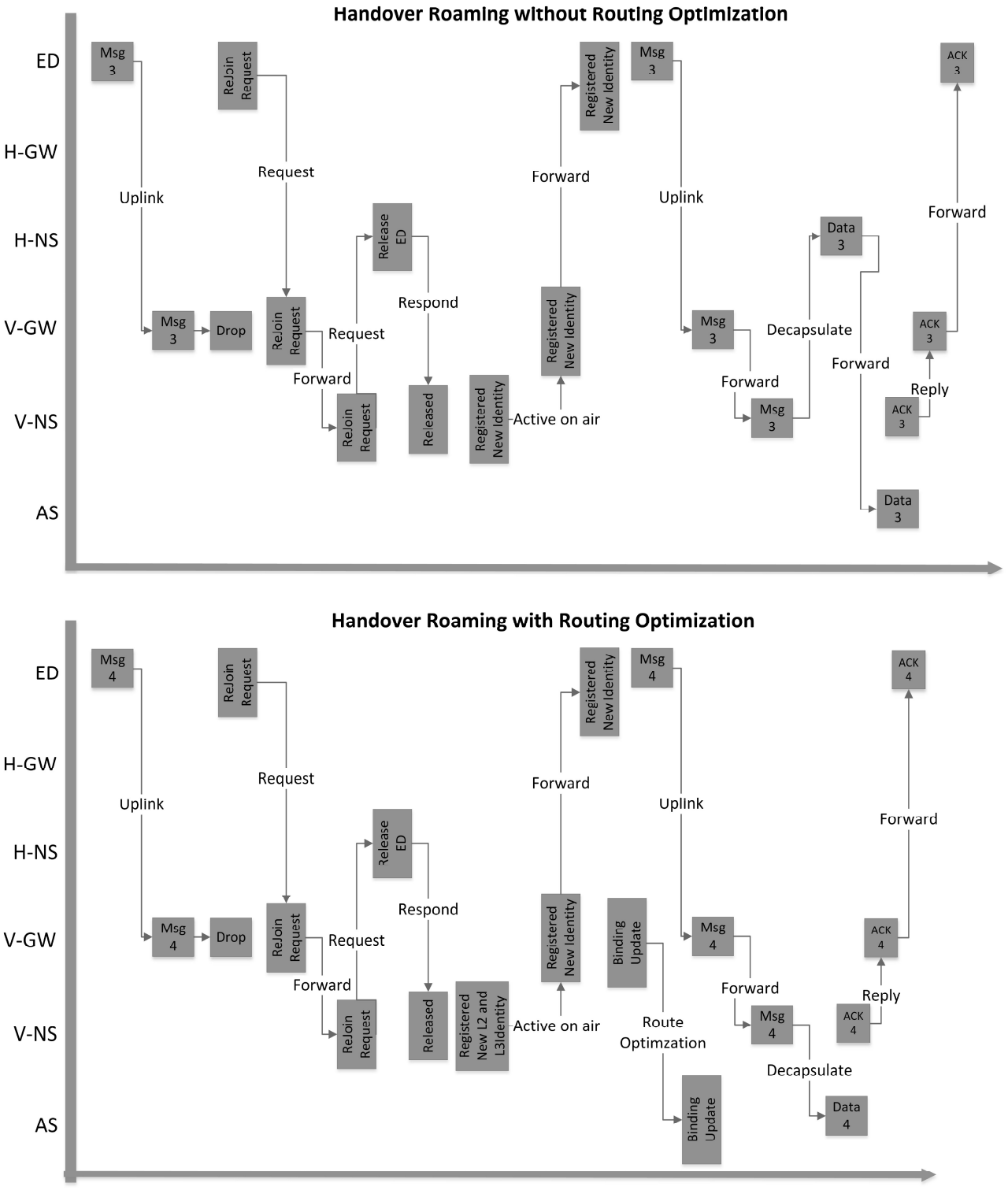

Fig. 11. Communication Mechanism of ED in Handover Roaming

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2020.2985925, IEEE Internet of Things Journal

4) Handover Roaming with Routing Optimization: This type of roaming is similar to the Handover roaming without routing optimization, with only one difference; As shown in Figure 11, under Handover roaming with routing optimization, after assigning the ED to the visited NS, the visited NS makes a binding update with the AS. After this procedure, the visited NS forwards the message directly to AS without passing by home NS. Moreover, the visited NS relays with ACK for ED.

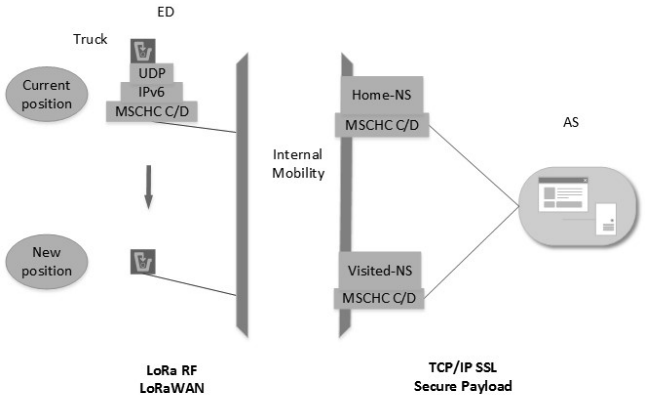

Fig. 12. Deploying of MSCHC mechanism on LoRaWAN Architecture

After briefly explaining the LoRaWAN roaming mechanisms and our proposed mechanism, we will start with the continuity of the session between ED and AS after roaming. In [35], the mobility between ED and GW, and ED and NS have been demonstrated and highlighted. In [35] and in this section we showed that, mobility management between different GWs belonging to the same NS or roaming between different NSs, is executed at the data link layer. Although this layer supports mobility, the session at the transport layer between the ED and AS is lost whenever the identity or location of the ED changes. Thus, L2 layer Handover support in LoRaWAN causes message redundancy at NS due to receiving the same message from the same ED but with different identities. To solve this issue, and to support session continuity with AS, the MIPv6 protocol has been proposed in this paper to provide continuity of session and routing optimization during the handover roaming. While MIPv6 is running at L3 layer and LoRaWAN supports mobility at L2 layer, we will define the two mobility mechanisms, as "internal" for mobility supported by LoRaWAN at the data link layer and "external" for mobility supported by MIPv6 on the network layer. The new architecture is illustrated in Figure 12.

Due to the lack of enforcement of MIPv6 implementation on constrained technologies, several header compression solutions were investigated in [8]. As a result, 6LoWPAN and SCHC were the most applicable ones. Then, in [38], we compare and simulate the two mechanisms using the NS3 simulator. The results show that SCHC is more applicable and provides a better compression rate than 6LoWPAN. Therefore, the SCHC mechanism is selected. The SCHC mechanisms have been updated to MSCHC and installed as a unified layer added on the server between ED and AS.

VI. Mapping MIPv6 LoRaWAN: the Proposed Solution

As listed in section V, there are three roaming scenarios. In scenario 1, the visited NS is transparent to the ED and it behaves as a GW corresponding to the Home NS. In this case, the L2 and L3 identities (LoRaWAN and IPv6) do not change. In scenario 2, the roaming mechanism is transparent for the L3 layer (network layer) where the IPv6 protocol is executed. In this roaming, the ED L2 identity changes but the IPv6 address remains the same. In scenarios 1 and 2, our proposed mechanism can be executed without any modifications on the current LoRaWAN v1.1 specification. In scenario 3, the identity of the ED at L2 and L3 layers changes.

The MIPv6 mechanism operates in a procedure formed of three steps: Agent discovery, Registration, and Tunneling. In the following we consider the handover roaming with routing optimization case. The three stages of MIPv6 mechanism will be divided over three subsections, in each, we discuss the mapping of the MIPv6 procedure using MSCHC over LoRaWAN.

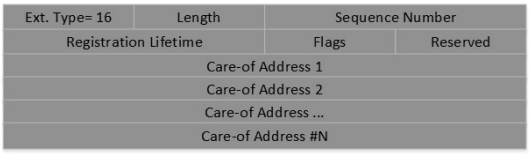

Fig. 13. Beacon Frame

A. Agent discovery

We propose to add the "Agent Announcement" method to GW. Periodically, the GW transmits beacon frames on the RX2 channel. Figure 13 illustrates the shape of the beacon frame. These tags contain information about the network service identity and the available IPv6 CoA (CoA) address. The RX2 channel has been selected to transmit these beacon frames because it uses a fixed rate and channel known to all EDs. The GW periodically sends beacon messages compressed by MSCHC on the RX2 channel, as shown in Table III. Also, a configuration is suggested for the Class A LoRaWAN function mechanism, where RX2 had to be opened before transmitting. Opening the RX2 offers the class A ED with many advantages, such as energy saving and collision avoidance. After mobility and location changes, class A EDs can detect changes when receiving the beacon frames of the GW corresponding to the visited NS. Then, ED reserves the transmission to avoid collisions and interference on the visited network with EDs using the same channel. Besides, the EDs can save the energy that would be consumed by transmitting a frame that would be abandoned by the visited NS GW. No update is needed for class B and class C, where class B ED periodically monitors the RX2 and class C ED listens on the RX2 all the time except when transmitting.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2020.2985925, IEEE Internet of Things Journal

TABLE III
BEACON FRAME COMPRESSED USING MSCHC AT GW SIDE

Beacon Frame						
Field	FP	FL	DL	TV	MO	C/D Action
ExtType	1	8	bi	16	equal	not-sent
RegLT	1	16	bi	-	equal	not-sent
Seq	1	16	bi	0x0000	ignore	value-sent
Length	1	8	bi	None	ignore	comp-length
Reserved	1	8	bi	None	ignore	not-sent
Prefix	1	64	bi	2001:63:80:8	ignore	value-sent
CoA 1	1	64	bi	::2	ignore	value-sent
CoA 2	1	64	bi	::3	ignore	value-sent
CoA N	1	64	bi	::32	ignore	value-sent

TABLE IV
BEFORE ROAMING: IPV6 COMPRESSION USING MSCHC AT ED SIDE

Rule 1						
NLC layer						
Field	FP	FL	DL	TV	MO	C/D Action
Version	1	4	bi	6	equal	not-sent
DiffServ	1	8	bi	0x00	equal	not-sent
FL	1	20	bi	0x000000	equal	not-sent
Length	1	16	bi	None	ignore	comp-length
NH	1	8	bi	17	equal	not-sent
HL	1	8	bi	30	equal	not-sent
ED prefix	1	64	bi	2001:63:80:7	equal	not-sent
ED addr.	1	64	bi	::2	equal	not-sent
AS prefix	1	64	bi	2001:63:80:9	equal	not-sent
AS addr.	1	64	bi	::2	equal	not-sent
Source Address= 2001:63:80:7::2 (Home Address of ED) Destination Address = 2001:63:80:9::2 (Address of Application server)						

When the location changes as shown in Figure 6, class C ED will notice these changes directly and without delay, as it will instantly receive the tag frames identifying the visited NS sites. Class B ED detects changes once it listens to RX2, depending on the time it uses to monitor periodically. Class A ED detects the change of location once the data is available for sending. This depends on the implementation of the device, whether it is a periodic transmission or not. In the latter case, the transmission is done in two cases: when it detects a change in the data or when it receives a request.

TABLE V
ROAMING PACKET: MIPV6 COMPRESSED WITH MSCHC AT ED SIDE

Rule 2						
NLC layer						
Field	FP	FL	DL	TV	MO	C/D Action
Version	1	4	bi	6	equal	not-sent
DiffServ	1	8	bi	0x00	equal	not-sent
FL	1	20	bi	0x000000	ignore	value-sent
Length	1	16	bi	None	ignore	comp-length
NH	1	8	bi	17	ignore	value-sent
HL	1	1	bi	30	ignore	value-sent
ED P CoA	1	64	bi	2001:63:80:8	ignore	value-sent
ED CoA	1	64	bi	::2	ignore	value-sent
AS prefix	1	64	bi	2001:63:80:9	ignore	value-sent
AS addr.	1	64	bi	::2	ignore	value-sent
ELC layer						
Field	FP	FL	DL	TV	MO	C/D Action
ED P HoA	1	64	bi	2001:63:80:7	ignore	value-sent
ED HoA	1	64	bi	::2	ignore	value-sent

TABLE VI
AFTER ROAMING: MIPV6 COMPRESSION USING MSCHC AT ED SIDE

Rule 3						
NLC layer						
Field	FP	FL	DL	TV	MO	C/D Action
Version	1	4	bi	6	equal	not-sent
DiffServ	1	8	bi	0x00	equal	not-sent
FL	1	20	bi	0x000000	equal	not-sent
Length	1	16	bi	None	ignore	comp-length
NH	1	8	bi	17	equal	not-sent
HL	1	1	bi	30	equal	not-sent
ED P CoA	1	64	bi	2001:63:80:8	equal	not-sent
ED CoA	1	64	bi	::2	equal	not-sent
AS prefix	1	64	bi	2001:63:80:9	equal	not-sent
AS addr.	1	64	bi	::2	equal	not-sent
ELC layer						
Field	FP	FL	DL	TV	MO	C/D Action
ED P HoA	1	64	bi	2001:63:80:7	equal	not-sent
ED HoA	1	64	bi	::2	equal	not-sent
Source Address= 2001:63:80:8::2 (Care of Address of ED) Destination Address = 2001:63:80:9::2 (Address of Application server) Destination Header = 2001:63:80:7::2 (Home Address of ED)						

B. Registration

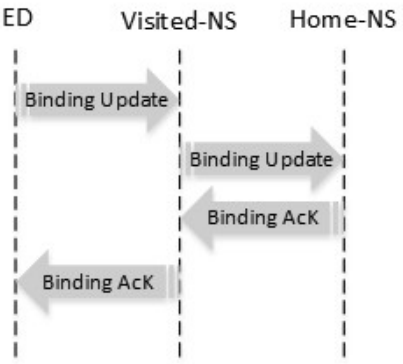

Fig. 14. Registration mechanism

Before location changes, ED uses the Home IPv6 header (HoA) to send the information to AS compressed by MSCHC [38] as shown in Table IV. In the new location, ED detects this change when it receives beacon frames from the visited NS during listening on RX2 before transmitting. Then, ED extracts one of the CoA addresses of the received beacon frame sent by the visited GW. This CoA is used for the link update procedure with visited NS. After that, the ED performs link updates with the visited NS, as shown in Figure 14. To initialize the registration process between the ED and the visited NS, the ED sends a frame using CoA as the source address and retains the HoA as the destination header in the extensions of the IPv6 header, as shown in Table V. In the visited NS, the identity of the ED will be detected from the first HoA prefix found in the field of the destination header. Besides, the visited NS communicates with Home NS of the ED in order to release its responsibility. This procedure is known as a binding update between the visited NS and the Home NS. Finally, the visited NS assigns this ED with an IPv6 address on layer L3.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2020.2985925, IEEE Internet of Things Journal

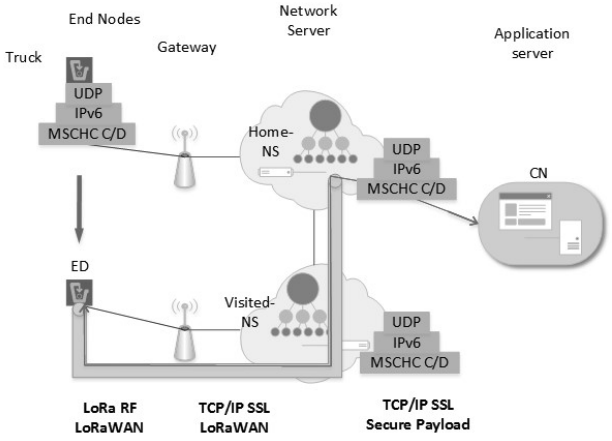

Fig. 15. Tunneling with HA

TABLE VII
MIPv6 COMPRESSION USING MSCHC AT AS SIDE

Rule 4						
NLC layer						
Field	FP	FL	DL	TV	MO	C/D Action
Version	1	4	bi	6	equal	not-sent
DiffServ	1	8	bi	0x00	equal	not-sent
FL	1	20	bi	0x000000	equal	not-sent
Length	1	16	bi	None	equal	not-sent
NH	1	8	bi	17	equal	not-sent
HL	1	1	bi	30	equal	not-sent
AS prefix	1	64	bi	2001:63:80:9	equal	not-sent
AS addr.	1	64	bi	::2	equal	not-sent
ED P addr.	1	64	bi	2001:63:80:8	equal	not-sent
ED addr.	1	64	bi	::2	equal	not-sent
ELC layer						
Field	FP	FL	DL	TV	MO	C/D Action
R.H P	1	64	bi	2001:63:80:7	equal	not-sent
R.H addr.	1	64	bi	::2	equal	not-sent
Source Address= 2001:63:80:9::2 (Address of Application server) Destination Address = 2001:63:80:8::2 (Care of Address of ED) Routing Header = 2001:63:80:7::2 (Home Address of ED)						

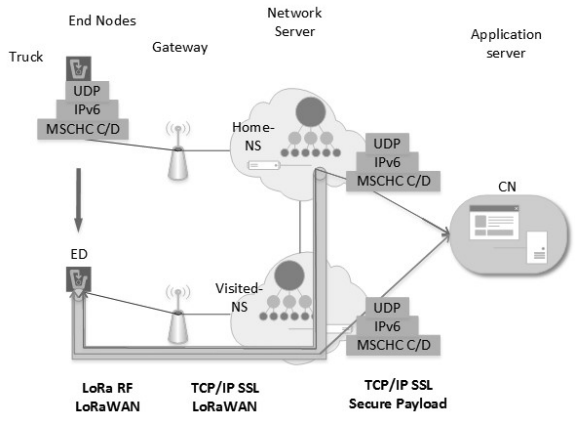

Fig. 16. Tunneling with CN using routing optimization

C. Tunneling

1) Without Routing Optimization: In the downlink, as shown in Figure 15, Home NS transmits the packets to the visited NS. Then the visited NS transmits the packets to ED. Using this mechanism in uplink, ED sets the CoA as the source address for IP packet, and in the extension headers, ED sets the HoA. The destination address will be the address of the AS as shown in Table VI. Upon receiving the IP packet, the visited NS decapsulates the packet by removing the outer header of the IP packet and sends the original packet to the Home NS. Then, the Home NS sends the packet to AS. However, using MIPv6 without routing optimization imposes the use of Home NS as packet forwarder for uplink and downlink between ED and AS. This process is sometimes too bandwidth consuming. In particular, when AS and visited NS are located on topologically-related networks.

2) With Routing Optimization: To solve the redundancy generated in previous case, MIPv6 provides an optimization mechanism called route optimization that an AS could support. As shown in Figure 16, we propose that an ED send packets directly to an AS. These packets will pass through the visited NS using the CoA as the source address (green line). The HoA of the ED is passed through the Destination header field of the IPv6 extension headers as shown in Figure 4. In addition, an AS can send packets directly to the ED (green line) thanks to CoA of the ED that is carried by the routing header as illustrated in Table VII.

VII. IMPLEMENTATION

To validate the proposition, a testbed has been realized. We consider the scenario shown in Figure 17, where a truck moves while sending information. The two roaming solutions proposed by LoRaWAN v1.1 are implemented and compared with our proposed mechanism. We compared the three scenarios in terms of latency, payload size and bandwidth per packet. The implementation details are given below.

A. Architecture

The network consists of one ED, two LoRa GW, two LoRaWAN NS and one AS. The network topology between ED and GW is star. In LoRaWAN v1.1, passive and handover roaming scenarios are supported, the source IPv6 address of ED does not change, and the mobility of ED is transparent to layer L3. As shown in Figure 17, ED moves while transmitting data. Any GW that receives the data will verify the ED AddrPrefix; If it corresponds to the current or cooperative networks, the packet is forwarded. Otherwise, the packet is discarded. If the AddrPrefix check is successful, the GW will send the data to NS. In our implementation, both NSs (local and visited) are locally connected to a router, and the router provides access to the Internet. The mechanisms of light weight MIPv6 and MSCHC [38] have been installed in the ED and in the two NSs. Then, any NS that receives the message will forward it to AS. Next, we will show the used equipments, the implementation procedure and the measurements done during this experiment.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2020.2985925, IEEE Internet of Things Journal

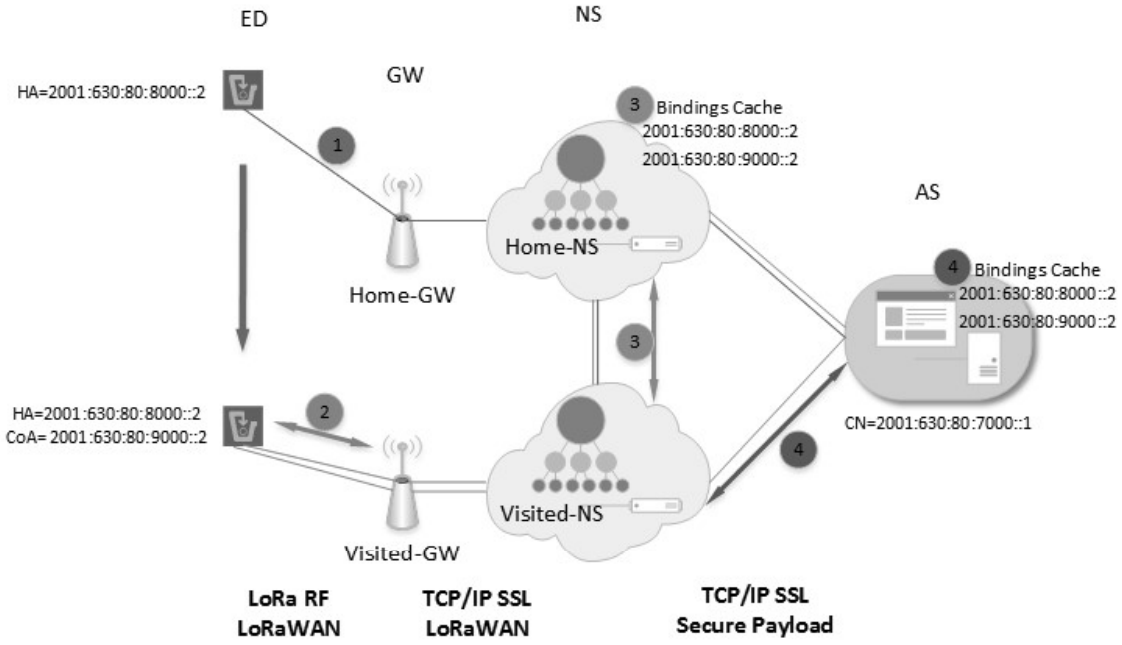

Fig. 17. Handover Procedure in LoRaWAN using MIPv6 compressed by MSCHC

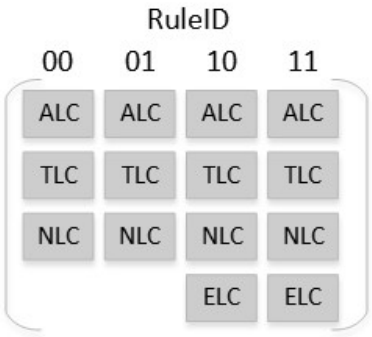

Fig. 18. Rules saved on ED

B. Equipments and software

To test our proposal, we built two LoRaWAN networks. As shown in figure 17, a LoRaWAN network consists of an ED, GW, NS and AS. We consider having two different operators implementing LoRaWAN networks with one NS and one GW for each. We consider one ED that sends data while moving and one AS that receives the sent data.

The ED consists of:

- Arduino UNO R3: This is an open source board based on the Microchip ATmega328P microcontroller and developed by Arduino. The board provides a set of digital/analog input/output pins and is compatible with the LoRa modules found in the market. The technical specifications of this hardware platform is shown in Table VIII.
- Dragino LoRa shield v1.4: LoRa shield is based on the Semtech SX1276/SX1278 chip. This is a long range transceiver on a Arduino shield form factor and based on Open source library. The Shield allows the user to send data and reach extremely long ranges at low data-rates. It provides ultra-long range spread spectrum communication and high interference immunity whilst minimising current consumption. Moreover, it can achieve a sensitivity of over -148dBm with an integrated power amplifier of up to +20 dBm for the link budget.

After selecting the hardware platform, we installed the Arduino IDE software on PC in order to configure and program the ATmega323P microcontroller implementing the ED. The manufacturer of the shield provides programmers with an open source library [39] with examples to program the ATmega323P microcontroller.

In addition to ED, the ”LoRaServerIO” project [40] provides ”LoRaGW” which is an open source software for LoRaWAN GW. We used the hardware module iC880A to implement the GW. This module can listen and receive frames with different spreading factors (SF) of several EDs on up to 8 channels in parallel. It supports the ”Dynamic Data Rate Adaptation” mechanism, which allows GW to hear distant EDs that use a higher SF while listening to closest EDs that use lower SFs. The iC880A module was combined with a Raspberry Pi (RPi) using the SPI communication protocol. We used an open software code given in the same project [41] to configure and program the module with the RPi. On the RPi, the ”Raspbain” operating system is installed. The technical specifications of the RPi hardware module are given in Table VIII.

For the NS, we benefit from the ”LoraServerIO” project

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2020.2985925, IEEE Internet of Things Journal

TABLE VIII
HARDWARE SPECIFICATIONS

	Ardunio UNO	Raspberry Pi 2
Model	UNO R3	Model B
Processor	ATMega328	Broadcom BCM2836, 32-bit quad-core ARM Cortex-A7
Clock Speed	16 MHz	900 MHz
RAM	2 KB	1 GB
Flash	32 KB	SD card 16 GB
EEPROM	1 KB	-
Input Voltage	5 V via USB	5 V
Min. Current	42 mA	300 mA
Operating System	-	Raspbian (Linux)
Programming Language	AVR C	C/C++ / Python / Java/ Scratch / Ruby
General Description	Microcontroller	Mini Computer
Usage needs	Control	Heavy Processing

that provides "LoRa Server". This is an open source software for LoRaWAN NS that can be compiled and used on the RPi. LoRa Server has the functionality of the NS component including the elimination of duplications in uplink frames, managing the schedule of downlink frames, updating the positions of the EDs in a search table and handling the received packets by the GW. We used two RPi and two GWs modules to build two standalone independent operators.

To implement the AS, an open source software "LoRa App Server" from [40] project is used to manage the functionality of LoRaWAN AS. It is responsible of the "inventory" part of ED in the LoRaWAN infrastructure. It handles the union request and the encryption of the payloads of the application. Also, it offers access to the web. It organizes applications and devices for better and easy management. This server is a software that does not require a standalone hardware. Thus, we installed it on a PC running "ubuntu" operating system.

C. Communication

As illustrated in Figure 17, in step 1 (red line), the truck (ED) is moving within the coverage of Home NS. When the information is ready to be sent, the device packages the data in the form of an IPv6 packet. Then, this packet passes through the MSCHC [38] that compresses the headers as shown in Table IV according to the rules saved in the device as shown in Figure 18. Since Home NS already knows the IP addresses of the ED and that of AS, ED selects rule ID "01" as shown in Table IV and sends the information. The RuleID represents the concatenation of the ruleIDs. Each ruleID represents a compression of the header fields of one layer of the ED protocol stack. The RuleID will be eight bits, the compression headers will be empty because no value sent, the payload will contain the information, and a padding bit will be added if necessary. Then, comparing with the conventional LoRaWAN package that can hold a maximum of 256 bytes, 8 bits represent the headers of the application, transport, network, and extension layers; 2 bits for each layer. Padding bits will be added if they are generally less than 8 bits. In general, the aggregated data is 1 byte more than a common LoRaWAN frame.

Now, consider that the truck moves out of the area covered by the Home NS GW. In step 2 (yellow line), ED receives the beacon frames as shown in Table III of the visited GW. Then, ED selects one of the CoAs found in this framework and uses it as a new identity for the link update procedure. Then, the ED sends a packet, as shown in Table V, containing the CoA as the source address, the AS address as the destination address and HoA set in the extension headers. The values of all these addresses sent since these data are not known by the visited NS. Therefore, the ED will create a new RuleID header numbered "10" corresponding to the communication with the visited NS. The sent packet will have 1 byte as RuleID, and the compression header will contain the three addresses with the prefix, equivalent to 128 bits multiplied by 3 (48 bytes). The remaining 200 bytes of the LoRaWAN frame can be used to retain information.

This packet is sent once to the visited NS after the roaming of the ED, then the visited NS stores all these values in the context corresponding to this ED. The frames, following the binding update procedure, will contain only 1-byte as RuleID without additional compression headers for the LoRaWAN frame. A new RuleID will be added every time the ED moves to another network operator as shown in Figure 18. The RuleID represents the concatenations of the sub ruleIDs of AL, TL, NL, and EL as shown in Equ. 1. The RuleID $\mathbf{R}$ is the concatenation of the four ruleIDs. Each ruleID is formed of $\bar{\bar{S}}_i$ bits. X_i is the total number of rules at a specific layer.

$$\mathbf{R} = [S_A, S_T, S_N, S_E] \tag{1}$$

$$\bar{\bar{S}}_i = \log_2 (X_i)$$

$$i = \{A, T, N, E\}$$

After the visited NS receives a frame from the ED, the visited NS uses the first prefix of the ED address to find the Home NS to which the ED belongs. In step 3 (green line), the visited NS communicates with Home NS to update the new ED location and sends the frame to Home NS. Then, Home NS forwarded the frames to ED passing by visited NS. The visited NS can use the routing optimization method, step 4 (blue line), to communicate directly with AS.

D. Results

During the experiments of the four scenarios, the ED was configured to transmit a short 12 characters message. The Wireshark program was used to measure the time and length of the packet. This program was installed in the two GWs and the results of the measurements are shown in Table IX.

In NoRM, each character encoded by the LoRaWAN ED is a byte. Therefore, the transmitted frame holds 12 bytes in the payload. In addition to LoRaWAN ED, the MSCHC ED requires at least one byte to represent the used RuleID. Therefore, the payload of the transmitted frame holds 13 bytes. Moreover, to LoRaWAN ED, the MSCHC ED context is stored

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2020.2985925, IEEE Internet of Things Journal

TABLE IX
TESTING RESULTS

Roaming Scenarios		No Roaming (NoRM)	Passive Roaming (PRM)	Handover Roaming Without Routing Optimization (nHORM)	Handover Roaming With Routing Optimization (HORM)
Mobility Support		No	Yes		
Payload size of transmitted frame	LoRaWAN	12 (bytes)			
	SCHC	13 (bytes)		48 bytes then 13 bytes	
Latency (ms)	LoRaWAN	72 ms	90 ms	5-6 sec (L2 HO) + 20 ms (Link) + [72 ms or 243 ms]	L2 HO = 5.6 sec
	MSCHC	243 ms	260 ms		L2 (5-6 sec) +L3(1.542ms) HO=7.41 sec
Bandwidth bytes per packet	LoRaWAN	Non	230 bytes		Non
	MSCHC	Non	234 bytes		Non

in the memory of the ED and the home NS before location change. Therefore, MSCHC ED will not send the header context of the communication protocol stack in the payload. Thus, the payload length of the MSCHC ED message received by the home NS was 13 bytes whereas it was 12 bytes for LoRaWAN ED. The time for the arrival of the two packages to the GW was measured using Wireshark. The results were as follows: in LoRaWAN ED, the latency from transmission to reception was 72 ms, where $T_{L(LoRaWAN)_{NoRO}}$ is equal, as shown in equation 2, to $T_{ToA}(M)$ which represents the time required for transmitting a 12-byte message using SF7 to air. M represents the payload size of the transmitted LoRaWAN frame in bytes. Time on Air (ToA) is described in [42] to calculate the values theoretically. While for MSCHC ED, the latency was 243 ms. This time can be represented by the equation 3, $T_{L(MSCHC)_{NoRO}}$, where the lag is equal to the sum of the time required to transmit 13 bytes in the air using SF7 denoted by $T_{ToA}(M)$ and time needed to compress the headers indicated by the code processing time T_{cp}. Figure 19 shows the results of the measurements for the reception of 5 packets without movement for the two EDs in LoRaWAN-NoRM and MSCHC-NoRM. As shown in the Figure 19, there is a delay of approximately 150 ms between the two EDs. This delay represents the processing time that cause MSCHC ED this latency over LoRaWAN ED.

$$T_{L(LoRaWAN)_{NoRO}} = T_{ToA}(M) \tag{2}$$

$$T_{L(MSCHC)_{NoRO}} = T_{ToA}(M) + T_{cp} \tag{3}$$

$$M = PayloadSize(Bytes)$$

In PRM: the message size is the same, but the latency increases as shown in Figure 19 for the LoRaWAN-PRM and MSCHC-PRM devices. This added latency will be expressed by the equation 4 for LoRaWAN ED and equation 5 for MSCHC ED. As shown in the two equations, a new parameter T_{Link} is added in both equations. T_{Link} represents the latency generated by the communication link between the visited NS and the Home NS. In our case, the value of T_{Link} is between 10 and 20 ms. Also, in PRM, the amount of data on the link between the two NSs must be considered. In the case

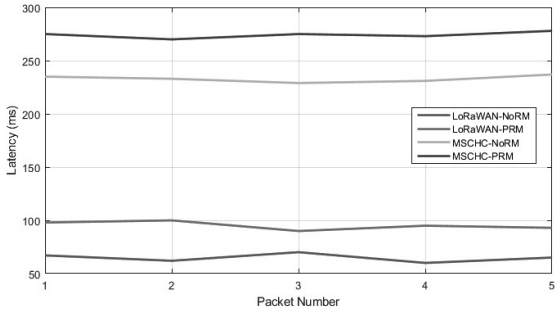

Fig. 19. Latency of packets received without Handover

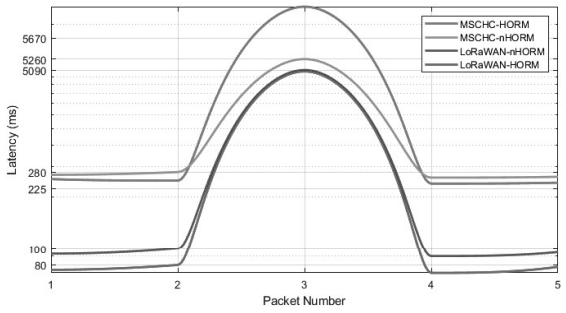

Fig. 20. Latency of packets received during Handover

of LoRaWAN ED, each frame transmitted by ED formed 230 bytes in the link between the two NS, while MSCHC ED is 234 bytes. The results, shown in Figure 19, illustrate that during the ED movement, these location changes of ED cause roaming mobility. This roaming was of passive type between the two NSs.

In this type of mobility, new parameters are to be considered: the latency generated by the link between the two NSs, and the packet size of the encapsulated LoRaWAN frame on the link. The advantage of this mobility is that it is transparent for both EDs. But, its disadvantage is that NS must consider the link latency when receiving uplink frame request acknowledgment. Due to that ED opens RX1 only for a specific time, and on the case of a load on the bandwidth

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2020.2985925, IEEE Internet of Things Journal

between the two NSs, the probability of losing ACK increases. Moreover, the load on the bandwidth and the latency for the uplink and downlink increases. That requires adaptive calculation algorithms for transmission/reception between ED and NS; Otherwise, the acknowledgments will not be received, and ED will repeat the transmission. The repetition of the transmission causes interference, collision, congestion in the bandwidth between two servers. Also, it increases the loss of messages and the power consumption of ED, especially if SF 12 used. Therefore, the current LoRaWAN communication mechanism that uses fixed time intervals for communication is not more preferable in passive roaming, especially for class A ED. This mechanism cause the loss of confirmation messages that may increase the repetition of ED messages.

$$T_{L(LoRaWAN)_{PRM}} = T_{L(LoRaWAN)_{NoRO}} + T_{Link} \quad (4)$$

$$T_{L(MSCHC)_{PRM}} = T_{L(MSCHC)_{NoRO}} + T_{Link} \quad (5)$$

In nHORM: the address of layer L2 (LoRaWAN) is changed but address L3 (IPv6) remains the same. Due to that handover occurs in layer L2, the message size for LoRaWAN ED remains the same and only the identity of the device changes. While for MSCHC ED, in the previous scenarios, the ruleID were only transmitted in addition to the data in the LoRaWAN frame since the context of the payload headers is saved on home NS. But now MSCHC ED must transmit the context of the payload headers to the visited NS in the first packet after roaming as shown in table V. This packet will be denoted by roaming packet (RP). After that, the visited NS will save this context, which is represented by a size of 48 bytes in our implementation. In the following frames, and if there are no changes to the payload headers, the MSCHC ED will only transmit the RuleID in addition to the data in the payload of the LoRaWAN frame. This difference for MSCHC ED on nHORM from PRM occurs only in the transmitting of the RP after roaming.

As shown in the Figure 20 for LoRaWAN-HORM and MSCHC-HORM, roaming occurs between packets 2 and 4. Latency $T_{L2_{HO}}$ for an ED to register and obtain a new address L2 is between 5 and 6 seconds according to the measurements. For more details on the registration process, the author of [43] describes in section II the on-air activation process that ED follows to participate with an NS. The average latency in LoRaWAN ED, for five time test repetition, was 5.3 s during roaming. This measure can be represented as $T_{L(LoRaWAN)_{nHORM}}$ which will be equal to the sum of the latency in $T_{L(LoRaWAN)_{PRM}}$ and the time required for HO $T_{L2_{HO}}$ between two NS as shown in the equation 6 during the roaming packet. Otherwise, $T_{L(LoRaWAN)_{nHORM}}$ will be equal to $T_{L(LoRaWAN)_{PRM}}$ in the following packets after roaming. While for MSCHC ED, the latency was 5.47 s for $RP = 1$ and 260 ms for the following packages. During roaming packet in MSCHC ED, the packet size increases. Thus, 48 bytes are added to the payload of LoRaWAN frame in addition to the previous 13 bytes. Thus, increasing the payload size leads to increasing the ToA. Therefore, a new parameter $T_{ToA}(N)$ is added to the overall latency $T_{L(MSCHC)_{nHORM}}$. $T_{ToA}(N)$ represents the time required to transmit additional bytes on the payload. N represents the length of the compressed context in bytes. Then, the overall latency $T_{L(MSCHC)_{nHORM}}$ for MSCHC ED can be expressed in the equation 7. As in PRM, the two EDs generate an additional load on the bandwidth between two NSs for every transmitted frame. LoRaWAN ED added 230 bytes whereas MSCHC ED added 234 bytes. Using the nHORM mechanism, the visited NS will confirm the reception of the received uplinks. Therefore, the problem of ACK reception in downlink found on PRM scenario does not exist here.

$$\begin{cases} T_{L(LoRaWAN)_{nHORM}} = T_{L(LoRaWAN)_{PRM}} \\ +T_{L2_{HO}} & RP = 1 \\ \\ T_{L(LoRaWAN)_{nHORM}} = T_{L(LoRaWAN)_{PRM}} & RP > 1 \end{cases} \quad (6)$$

$$\begin{cases} T_{L(MSCHC)_{nHORM}} = T_{L(MSCHC)_{PRM}} \\ +T_{ToA}(N) + T_{L2_{HO}} & RP = 1 \\ \\ T_{L(MSCHC)_{nHORM}} = T_{L(MSCHC)_{PRM}} & RP > 1 \end{cases} \quad (7)$$

$$N = CompressedContext(Bytes)$$

In HORM: the visited NS avoids routing by optimizing the route and directly sending the data to AS without going through the home NS. In LoRaWAN ED, the results show that there is no difference with the nHORM mechanism, since routing optimization is a mechanism done between the visited NS and the AS. That optimization can be performed without the ED notice. But the latency generated by T_{Link} in HORM was removed as shown in the Figure 20 when comparing LoRaWAN-HORM with LoRaWAN-nHORM after roaming. These results shown in Table IX for LoRaWAN ED can be expressed by the equation 8. Where the overall latency $T_{L(LoRaWAN)_{HORM}}$ during the RP is equal to time $T_{L(LoRaWAN)_{NoRO}}$ to transmit 12 bytes by LoRaWAN ED when no roaming exists in addition to the time $T_{L2_{HO}}$ for L2 handover. After roaming, LoRaWAN ED returns as if no roaming exists. While for MSCHC ED, the identity of layer L3 will change, and a new IPv6 address will be assigned to the device. As shown in the Figure 20, the time delay generated by layer L3 HO $T_{L3_{HO}}$ is approximately equal to 1,542 s in the comparison of the difference between the peak of MSCHC-HORM with that of MSCHC-nHORM. But the latency of the rest of the packets is less than that of MSCHC-nHORM. This $T_{L3_{HO}}$ does not represent MIPv6 HO; It represents a lightweight implementation of MIPv6 with improvements to achieve our goal. Then, the overall latency for MSCHC ED can be described by the equation 9. Also, in the HORM scenario, the bandwidth between the two NSs is zero, since there is no forward data between the two servers. After roaming,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JIOT.2020.2985925, IEEE Internet of Things Journal

MSCHC ED overall latency $T_{L(MSCHC)_{HORM}}$ will be equal to $T_{L(MSCHC)_{NoRO}}$ as if there is no roaming.

$$\begin{cases} T_{L(LoRaWAN)_{HORM}} = T_{L(LoRaWAN)_{NoRO}} & \\ +T_{L2_{HO}} & RP = 1 \\ \\ T_{L(LoRaWAN)_{HORM}} = T_{L(LoRaWAN)_{NoRO}} & otherwise \end{cases} \tag{8}$$

$$\begin{cases} T_{L(MSCHC)_{HORM}} = T_{L(MSCHC)_{NoRO}} & \\ +T_{ToA}(N) + T_{L2_{HO}} + T_{L3_{HO}} & RP = 1 \\ \\ T_{L(MSCHC)_{HORM}} = T_{L(MSCHC)_{NoRO}} & otherwise \end{cases} \tag{9}$$

$$N = CompressedContext(Bytes)$$

As can be seen in the results, in the case of PRM, there is no HO latency, the communication is fluid and transparent for ED. But a lag of link between the servers created in addition to the load on the bandwidth between the servers. Also, to the PRM problems, nHORM added the HO latency to the first roaming packet received during the mobility that causes roaming. While in HORM, the HO latency increases in the roaming packet during movement as shown in Figure 20, but resolves the problem of link loading, latency created from the nHORM mechanism and the time of the downlink. Also, it optimizes routing and reduces control messaging between the two servers.

Depending on the application specifications, it is possible to use the PRM or HORM mechanism when mobility causes roaming. If the time is critical or if a large amount of ED is moving and confirmation is not required for reception, PRM is preferred. If Quality of service (QoS), less lost messages, little bandwidth between servers, better administration of ED and permissible latency, are required then HORM is preferred. Even with the mobility delay generated in HORM during the RP, this mechanism works better when mobility occurs. The visited NS will be the Home NS for the ED, and the new NS will contact the AS directly as there was no mobility. In our proposal, we consider the use of OTAA ED, where the ED will be assigned a new identity and a network session key when it moves to a foreign network operator as the mechanism supported in the handover roaming scenarios in LoRaWANv1.1. In this case, the foreign NS will decrypt the received packages and forward them directly to AS after performing the routing optimization mechanism.

Compared to the roaming mechanism in LoRaWAN, our contribution supports full roaming. As specified in the LoRaWANv1.1 standard, even if ED is assigned with the new network operator in handover roaming, the foreign operator must forward the uplink data to the home operator. Then, the home operator forwards the uplink data to the AS. Therefore, in the two roaming scenarios proposed by the LoRaWANv1.1 standard, the uplink data of the ED must be returned to the home operator first and then it is forwarded to the AS. We have to mention that the AS is not a server corresponding to the operator of the home network. In cellular network roaming, the called mobile phone (server) is still within the operator of the home network and the calling mobile phone (client) is transferred to another operator. In LoRaWAN, AS corresponds to a user, company, etc., on the internet. Therefore, it is not necessary to route all packets through the home operator. For that, it is obvious that full roaming is not supported in LoRaWANv1.1 standard. While our contribution supports full roaming in addition to the IPv6 features. Our proposed solution will be considered in future work covering especially the areas of heterogeneous connectivity and vertical handover between technologies.

VIII. CONCLUSION

Adding the IP stack on an IoT device was a successful step. IPv6 offers extremely large addressing capabilities. Also, it facilitates the merging of the physical and digital world, allowing IoT to grow faster. We used IPv6 in our work as an adaptation layer to support mobility for IoT devices while addressing the heterogeneity of technologies. In this paper, we considered the continuity of the session in the transport layer of the ED in relation to the AS after roaming mobility. We focused on LPWAN technologies that share the same network topology, which is the star topology. We proposed to use the MIPv6 protocol and we investigated the use of the SCHC mechanism to compress the headers generated from the IP packet. Since SCHC is static and not efficient in the use of memory for restricted devices, we proposed MSCHC mechanism to divide the rules by layers. Then, we proposed a mechanism to adapt MIPv6 using MSCHC to support mobility with session continuity in LPWAN. We selected LoRaWAN technology to test our solution. The proposed solution was implemented in a LoRaWAN network and different scenarios were examined. Our proposed mechanism contributes to: session continuity, bandwidth reduction among operators, avoiding the loss of ACK and decreasing the delay between ED and AS. Future work will focus on minimizing HO latency and improving the MSCHC mechanism in classification, compression and data management between ED and NS.

REFERENCES

[1] L. Goasduff, "Gartner says 5.8 billion enterprise and automotive iot endpoints will be in use in 2020," 2019.

[2] F. Adelantado, X. Vilajosana, P. Tuset-Peiro, B. Martinez, J. Melia-Segui, and T. Watteyne, "Understanding the limits of lorawan," *IEEE Communications Magazine*, vol. 55, no. 9, pp. 34–40, 2017.

[3] Y. P. E. Wang, X. Lin, A. Adhikary, A. Grovlen, Y. Sui, Y. Blankenship, J. Bergman, and H. S. Razaghi, "A primer on 3gpp narrowband internet of things," *IEEE Communications Magazine*, vol. 55, pp. 117–123, March 2017.

[4] S. Ziegler, A. Skarmeta, P. Kirstein, and L. Ladid, "Evaluation and recommendations on ipv6 for the internet of things," in *2015 IEEE 2nd World Forum on Internet of Things (WF-IoT)*, pp. 548–552, Dec 2015.

[5] P. Wu, Y. Cui, J. Wu, J. Liu, and C. Metz, "Transition from ipv4 to ipv6: A state-of-the-art survey," *IEEE Communications Surveys Tutorials*, vol. 15, pp. 1407–1424, Third 2013.

[6] W. Kastner, M. Kofler, M. Jung, G. Gridling, and J. Weidinger, "Building automation systems integration into the internet of things the iot6 approach, its realization and validation," in *Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA)*, pp. 1–9, Sept 2014.

[7] S. Ziegler, C. Crettaz, and I. Thomas, "Ipv6 as a global addressing scheme and integrator for the internet of things and the cloud," in *2014 28th International Conference on Advanced Information Networking and Applications Workshops*, pp. 797–802, May 2014.

[8] W. Ayoub, M. Mroue, F. Nouvel, A. E. Samhat, and J. c. Prévotet, "Towards ip over lpwans technologies: Lorawan, dash7, nb-iot," in *2018 Sixth International Conference on Digital Information, Networking, and Wireless Communications (DINWC)*, pp. 43–47, April 2018.

[9] A. Minaburo, L. Toutain, C. Gomez, D. Barthel, and J.-C. Zúñiga, "Static Context Header Compression (SCHC) and fragmentation for LPWAN, application to UDP/IPv6," Internet-Draft draft-ietf-lpwan-ipv6-static-context-hc-21, Internet Engineering Task Force, July 2019. Work in Progress.

[10] A. Minaburo, L. Toutain, and R. Andreasen, "LPWAN Static Context Header Compression (SCHC) for CoAP," Internet-Draft draft-ietf-lpwan-coap-static-context-hc-09, Internet Engineering Task Force, July 2019. Work in Progress.

[11] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of things for smart cities," *IEEE Internet of Things Journal*, vol. 1, pp. 22–32, Feb 2014.

[12] S. M. R. Islam, D. Kwak, M. H. Kabir, M. Hossain, and K. S. Kwak, "The internet of things for health care: A comprehensive survey," *IEEE Access*, vol. 3, pp. 678–708, 2015.

[13] P. Hank, S. Müller, O. Vermesan, and J. V. D. Keybus, "Automotive ethernet: In-vehicle networking and smart mobility," in *2013 Design, Automation Test in Europe Conference Exhibition (DATE)*, pp. 1735–1739, March 2013.

[14] Y. Shibata and G. Sato, "Iot based mobility information infrastructure in challenged network environment toward aging society," in *2017 31st International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, pp. 645–648, March 2017.

[15] M. S. Shahamabadi, B. B. M. Ali, P. Varahram, and A. J. Jara, "A network mobility solution based on 6lowpan hospital wireless sensor network (nemo-hwsn)," in *2013 Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, pp. 433–438, July 2013.

[16] A. Dunkels, "Full tcp/ip for 8-bit architectures," in *Proceedings of the 1st International Conference on Mobile Systems, Applications and Services*, MobiSys '03, (New York, NY, USA), pp. 85–98, ACM, 2003.

[17] S. Ziegler, C. Crettaz, L. Ladid, S. Krco, B. Pokric, A. F. Skarmeta, A. Jara, W. Kastner, and M. Jung, "Iot6 – moving to an ipv6-based future iot," in *The Future Internet* (A. Galis and A. Gavras, eds.), (Berlin, Heidelberg), pp. 161–172, Springer Berlin Heidelberg, 2013.

[18] Z. Sheng, S. Yang, Y. Yu, A. V. Vasilakos, J. A. Mccann, and K. K. Leung, "A survey on the ietf protocol suite for the internet of things: standards, challenges, and opportunities," *IEEE Wireless Communications*, vol. 20, pp. 91–98, December 2013.

[19] S. Thielemans, M. Bezunartea, and K. Steenhaut, "Establishing transparent ipv6 communication on lora based low power wide area networks (lpwans)," in *2017 Wireless Telecommunications Symposium (WTS)*, pp. 1–6, April 2017.

[20] P. Weber, D. Jäckle, D. Rahusen, and A. Sikora, "Ipv6 over lorawan," in *2016 3rd International Symposium on Wireless Systems within the Conferences on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS-SWS)*, pp. 75–79, Sept 2016.

[21] A. J. Jara, D. Fernandez, P. Lopez, M. A. Zamora, and A. F. Skarmeta, "Lightweight mobile ipv6: A mobility protocol for enabling transparent ipv6 mobility in the internet of things," in *2013 IEEE Global Communications Conference (GLOBECOM)*, pp. 2791–2797, Dec 2013.

[22] I. Farris, S. Pizzi, M. Merenda, A. Molinaro, R. Carotenuto, and A. Iera,

[24] D. Dujovne, T. Watteyne, X. Vilajosana, and P. Thubert, "6tisch: deterministic ip-enabled industrial internet (of things)," *IEEE Communications Magazine*, vol. 52, pp. 36–41, December 2014.

"6lo-rfid: A framework for full integration of smart uhf rfid tags into the internet of things," *IEEE Network*, vol. 31, no. 5, pp. 66–73, 2017.

[23] C. Gomez, J. Paradells, C. Bormann, and J. Crowcroft, "From 6lowpan to 6lo: Expanding the universe of ipv6-supported technologies for the internet of things," *IEEE Communications Magazine*, vol. 55, pp. 148–155, DECEMBER 2017.

[25] C. Bormann, A. P. Castellani, and Z. Shelby, "Coap: An application protocol for billions of tiny internet nodes," *IEEE Internet Computing*, vol. 16, pp. 62–67, March 2012.

[26] S. M. Chun and J. T. Park, "Mobile coap for iot mobility management," in *2015 12th Annual IEEE Consumer Communications and Networking Conference (CCNC)*, pp. 283–289, Jan 2015.

[27] S.-M. Chun and J.-T. Park, "A mechanism for reliable mobility management for internet of things using coap," in *Sensors*, 2017.

[28] D. Garcia-Carrillo, R. Marin-Lopez, A. Kandasamy, and A. Pelov, "A coap-based network access authentication service for low-power wide area networks: Lo-coap-eap," *Sensors*, vol. 17, no. 11, 2017.

[29] R. Sanchez-Iborra, J. Sánchez-Gómez, J. Santa, P. J. Fernández, and A. F. Skarmeta, "Ipv6 communications over lora for future iov services," in *2018 IEEE 4th World Forum on Internet of Things (WF-IoT)*, pp. 92–97, Feb 2018.

[30] K. Q. Abdelfadeel, V. Cionca, and D. Pesch, "LSCHC: layered static context header compression for lpwans," *CoRR*, vol. abs/1708.05209, 2017.

[31] K. Abdelfadeel, V. Cionca, and D. Pesch, "Dynamic context for static context header compression in lpwans," 04 2018.

[32] O. Gimenez, I. Petrov, and J. Catalano, "Static Context Header Compression (SCHC) over LoRaWAN," Internet-Draft draft-ietf-lpwan-schc-over-lorawan-03, Internet Engineering Task Force, Oct. 2019. Work in Progress.

[33] S. M. Ghaleb, S. Subramaniam, Z. A. Zukarnain, and A. Muhammed, "Mobility management for iot: a survey," *EURASIP Journal on Wireless Communications and Networking*, vol. 2016, p. 165, Jul 2016.

[34] D. Minoli, *Layer 3 Connectivity: Mobile IPv6 Technologies for the IoT*, pp. 392–. Wiley Telecom, 2013.

[35] W. Ayoub, A. E. Samhat, F. Nouvel, M. Mroue, and J. Prévotet, "Internet of mobile things: Overview of lorawan, dash7, and nb-iot in lpwans standards and supported mobility," *IEEE Communications Surveys Tutorials*, pp. 1–1, 2018.

[36] D. Minoli, *Layer 3 Connectivity: IPv6 Technologies for the IoT*. Wiley, 2013.

[37] LoRa Alliance, https://www.lora-alliance.org/lorawan-for-developers, *LoRaWAN v1.1 Specification*, october 11, 2017 ed.

[38] W. Ayoub, F. Nouvel, S. Hmede, A. E. Samhat, M. Mroue, and J. Prévotet, "Implementation of schc in ns-3 and comparison with 6lowpan," in *2019 26th International Conference on Telecommunications (ICT)*, pp. 432–436, April 2019.

[39] M. Kooijman, "Arduino-lmic library." https://github.com/matthijskooijman/arduino-lmic, Dec 5, 2017.

[40] cablelabs, sidnfonds, and acklio, "Loraserver project." https://www.loraserver.io/, Accessed on 13 Nov 2018.

[41] L. network, "Lora gateway project." https://github.com/Lora-net/lora_gateway, Apr 5, 2017.

[42] F. Cuomo, M. Campo, E. Bassetti, L. Cartella, F. Sole, and G. Bianchi, "Adaptive mitigation of the air-time pressure in lora multi-gateway architectures," in *European Wireless 2018; 24th European Wireless Conference*, pp. 1–6, May 2018.

[43] J. Toussaint, N. E. Rachkidy, and A. Guitton, "Performance analysis of the on-the-air activation in lorawan," in *2016 IEEE 7th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, pp. 1–7, Oct 2016.