IPv6 over LPWANs: connecting Low Power Wide Area Networks to the Internet (of Things)

Carles Gomez[1], Ana Minaburo[2], Laurent Toutain[3], Dominique Barthel[4], Juan Carlos Zuniga[5]
[1]Universitat Politècnica de Catalunya, [2]Acklio, [3]IMT-Atlantique, [4]Orange Labs, [5]Sigfox
E-mail: carlesgo@entel.upc.edu, ana@ackl.io, Laurent.Toutain@imt-atlantique.fr, dominique.barthel@orange.com, juancarlos.zuniga@sigfox.com

Abstract

LPWANs have recently emerged as a promising solution for enabling industrial IoT applications. To fully exploit their potential, LPWANs need to be connected to the Internet. However, the severe capacity constraints of LPWAN technologies challenge IPv6 support, and even 6LoWPAN-based adaptations are not sufficient. In this paper, we present SCHC, an ultralightweight IPv6 adaptation layer designed for LPWANs, which is being standardized by the IETF.

1. Introduction

The Internet of Things (IoT) is a networking paradigm whereby a vast number of connected, typically resource-constrained devices (e.g. battery-enabled sensors and actuators), sense or act upon the physical world to enable intelligent environments. This vision constitutes a revolution that is expected to transform our society by substantially enhancing productivity, sustainability, and human life quality.

The IoT is currently developing in several dimensions. As the number of connected IoT devices grows steadily, the number of communications technologies for IoT devices increases, too. Well-established IoT technologies, such as IEEE 802.15.4 and Bluetooth Low Energy (BLE), are characterized by a rather short communication range, generally in the order of tens or a few hundreds of meters. However, with such a reduced range, a considerable amount of infrastructure (e.g. relay nodes and/or gateways) is needed to ensure connectivity of IoT devices over a large area (e.g. a city). This approach requires a potentially complex networking solution and leads to high network deployment, maintenance and management cost.

In order to overcome the aforementioned issues, the category of wireless communication technologies called Low Power Wide Area Networks (LPWANs) has emerged. LPWAN technologies define star topology networks whereby a single base station covers up to hundreds of thousands of IoT devices with a multiyear IoT device battery lifetime, while supporting a multikilometer link range [1]. These characteristics are achieved at the expense of extremely low data rates and small payloads, which are sufficient to many common industrial IoT applications. In fact, LPWANs have quickly attracted the interest of industry, academia and standards development organizations,

with 4 billion LPWAN devices predicted by 2025[1]. Flagship LPWAN technologies include LoRaWAN, Sigfox and Narrowband IoT (NB-IoT). Furthermore, the IEEE 802.15.4w task group has been recently chartered to optimize IEEE 802.15.4 for LPWAN scenarios.

To fully exploit the potential of LPWANs, Internet connectivity support is required. Therefore, LPWAN devices need to be able to run IP. In particular, IP version 6 (IPv6) is assumed, since it offers a massive address space and self-configuration tools. However, IPv6 was designed for resource-rich networking environments (e.g. Ethernet), whereas typical IoT network scenarios offer significantly constrained energy, computation, and communication capabilities. For over one decade, the IETF IPv6 over Low-power Wireless Personal Area Networks (6LoWPAN) Working Group (WG) and the IETF IPv6 over Networks of Resource-constrained Nodes (6Lo) WG have developed adaptation layers to enable and optimize IPv6 over a wide range of IoT link-layer technologies, hereafter called 6LoWPAN/6Lo technologies. These include IEEE 802.15.4, BLE, ITU-T G.9959 (Z-Wave), Digital Enhanced Cordless Telecommunications – Ultra Low Energy (DECT-ULE) and Near Field Communication (NFC), among others [2]. Nevertheless, 6LoWPAN/6Lo adaptation style would incur unaffordable overhead over LPWANs, given the extremely restricted communication resources of LPWAN technologies. For example, the sustained capacity of 6LoWPAN/6Lo technologies is of at least a few kbit/s, while some LPWAN technologies are limited to as low as the mbit/s (i.e. millibit/s!) order.

In 2016 the IETF LPWAN WG was chartered to provide support of IPv6 and upper layer Internet protocols over LPWANs [3]. This WG is now reaching completion of the Static Context Header Compression and Fragmentation (SCHC) specification, of which we are authors. In this article, we motivate, present and evaluate SCHC.

The remainder of this article is organized as follows. Section 2 introduces the main target LPWAN technologies considered by the IETF LPWAN WG. Section 3 presents the IPv6-based protocol stack for LPWANs. Sections 4 and 5 describe and evaluate SCHC, respectively. Open issues are overviewed in Section 6. Finally, Section 7 provides conclusions.

2. Target LPWAN technologies

This section briefly reviews the LPWAN technologies considered by the IETF LPWAN WG in the design of SCHC, namely: LoRaWAN, Sigfox, NB-IoT and IEEE 802.15.4w. These technologies are discussed and compared with 6LoWPAN/6Lo wireless technologies.

[1] https://www.abiresearch.com/press/4-billion-iot-devices-will-rely-lpwan-technologies (accessed on March 14th 2019)

2.1. LoRaWAN

LoRaWAN is a popular LPWAN technology that was first specified in 2015 by an industry consortium called the LoRa Alliance. LoRaWAN defines a protocol architecture that comprises a Physical (PHY) layer, a Medium Access Control (MAC) layer and customer applications on top of the MAC layer [4].

At the PHY layer, LoRaWAN operates in unlicensed bands and uses the LoRa modulation, which is based on Chirp Spread Spectrum (CSS). A range of Spreading Factor (SF) options are supported, leading to different corresponding Data Rates (DRs) and robustness levels. In order to save energy, a LoRaWAN IoT device typically only turns its receiver on shortly after it transmits an uplink message, which is done asynchronously.

2.2. Sigfox

Sigfox is a technology created by the eponymous company, which was founded in 2009. Currently, Sigfox has been deployed in more than 60 countries. In Sigfox, IoT devices asynchronously transmit messages by using Ultra Narrow Band (UNB) in unlicensed spectrum. Each message sent by an IoT device is transmitted three times, using a different frequency for each of the three transmission attempts. If a device is willing to receive messages, it indicates so in the uplink message, after which the device opens a receiving window. Otherwise, when the device is inactive, it keeps its radio interface off [5].

2.3. NB-IoT

NB-IoT is specified in 3GPP Release 13, published in 2016. NB-IoT uses a subset of the Long Term Evolution (LTE) standard, with the aim to meet IoT requirements, such as low device cost and relaxed bandwidth requirements [6]. In contrast with LoRaWAN and Sigfox, NB-IoT operates in licensed frequency bands. In NB-IoT, the IoT device remains by default in low energy consumption states, except for the periodic transmission of location reports and monitoring of a paging channel for incoming downlink data. Uplink data transmission may be carried out after a successful, IoT device-initiated, contention-based random access procedure. Downlink data may also be received immediately after uplink data transmission [7].

2.4. IEEE 802.15.4w

IEEE 802.15.4w is an IEEE 802.15.4 amendment currently being developed, intended to address LPWAN use cases, by enhancing the existing IEEE 802.15.4k specification. The latter was designed for Low Energy Critical Infrastructure Monitoring (LECIM). The proposed enhancements, still being discussed at the time of writing, comprise improved Forward Error Correction (FEC) codes, sub-packet spreading in time and frequency, and a scalable multiple access frame structure. The intended goals include improving interference resilience, energy efficiency, and scalability.

2.5. Discussion

We now compare the communication capacity features of LoRaWAN, Sigfox and NB-IoT with those of 6LoWPAN/6Lo technologies, focusing on the aspects that are relevant for IPv6 support (Table 1). Overall, LoRaWAN and Sigfox are significantly more constrained, whereas NB-IoT has similar characteristics to 6LoWPAN/6Lo technologies, as discussed next.

In order to benefit link range, LoRaWAN and Sigfox use unlicensed sub-GHz bands instead of higher ISM bands (e.g. the 2.4 GHz band). However, in some world regions, the former are subject to spectrum access regulations which both LoRaWAN and Sigfox enforce by keeping the device radio duty cycle (RDC) below a given limit (e.g. 1% in the uplink, in some channels in Europe). As a result, message rates in these two technologies may be extremely low, even down to a few messages per day. In contrast, 6LoWPAN/6Lo technologies either use bands that are not subject to such regulatory constraints, or use alternative spectrum sharing techniques, therefore they do not suffer the same issues. Note that, since NB-IoT uses licensed frequency bands, it is also free of message rate limitations stemming from spectrum access regulations.

Also favoring a long link range, both LoRaWAN and Sigfox use PHY layer data rates (10^2 to 10^4 bit/s) lower than those of 6LoWPAN/6Lo technologies (10^4 to 10^6 bit/s). In consequence, their frame size needs to be small to limit device energy consumption due to communication. The maximum frame payload size in Sigfox and in some LoRaWAN scenarios is extremely short (of ~10 bytes), well below that of 6LoWPAN/6Lo technologies. This feature also reduces the probability of collision and thus favors network scalability. However, it also severely decreases sustained transmission capacity. Furthermore, neither Sigfox nor LoRaWAN natively support fragmentation and reassembly (hereinafter denoted *fragmentation*, for brevity), thus they do not allow sending larger upper layer data units.

The extreme constraints exhibited by LoRaWAN and Sigfox motivated the development of SCHC, a new adaptation layer specifically designed to support IPv6 over LPWANs, as detailed in the next section. While NB-IoT is not as limited as LoRaWAN and Sigfox, it will also benefit from the high efficiency of SCHC.

	LPWAN technologies			6LoWPAN/6Lo wireless technologies				
	LoRaWAN	**Sigfox**	**NB-IoT**	**IEEE 802.15.4**	**BLE**	**ITU-T G.9959**	**DECT-ULE**	**NFC**
Frequency band(s) (MHz)	868 (EU), 915 (US), 783 (China)	868 (EU), 915 (US), 923 (Japan)	Various: 416 (min), 2200 (max)	868 (EU), 915 (US), 2400 (worldwide)	2400	868 (EU), 915 (US)	1900	13.56
Type of band	Unlicensed	Unlicensed	Licensed	Unlicensed	Unlicensed	Unlicensed	Dedicated	Unlicensed
Modulation	CSS	DBPSK (uplink), GFSK (downlink)	π/2-BPSK or π/4-QPSK (upl.), QPSK (downlink)	BPSK (sub-GHz), O-QPSK (2.4 GHz)	GFSK	FSK/FSK/GFSK (R1/R2/R3)	GFSK	OOK, BPSK
Receiver sensitivity (dBm)	-137 (typical)	-142 (typical)	-141 (typical)	-92 min. (sub-GHz), -85 min. (2.4 GHz)	-70 (Bluetooth 4.0)	-95/-92/-89 (R1/R2/R3)	-86	N.A.
PHY layer data rate (kbit/s)	0.25 ÷ 5.47 (EU), 50 (optional)	0.1/0.6	250 (uplink), 226.7 (downlink)	20/40/250	125/500/ 1000/2000	9.6/40/100 (R1/R2/R3)	1152	106/212/424
Message rate constraints	Duty cycle < 1% (EU, China)	140/4 messages per day (uplink/downlink)	No	No	No	No	No	No
Capacity per device (order of magnitude, in bit/s)	10^0 (DR0, EU), 10^2 (DR5, EU)	10^{-1} (uplink) 10^{-3} (down.)	10^4	10^3 (sub-GHz), 10^5 (2.4 GHz)	10^5 (at 1 Mbit/s)	10^3 (R1), 10^4 (R2/R3)	10^5	10^4 (at 424 kbit/s)
MAC mechanism	Aloha-based (optional ACKs + retries)	Aloha-based (3 transmissions)	Slotted Aloha (random access) + scheduling	CSMA/CA, TDMA	TDMA	CSMA/CA	TDMA	TDMA link initialization
Maximum frame payload size (bytes)	11 (DR0, USA) ÷ 242 (worldwide)	12 (uplink), 8 (downlink)	1600	105	23	158	38	125
Fragmentation and reassembly	No	No	Yes	No	Yes	Yes	Yes	Yes
Network topology	Star	Star	Star	Star, mesh	Star, mesh	Mesh	Star	Point-to-point
Standards Developm. Organization	LoRa Alliance ™	Sigfox (company)	3GPP	IEEE	Bluetooth SIG	ITU-T	ETSI	NFC Forum

Table 1. Main details of LoRaWAN, Sigfox and NB-IoT. IEEE 802.15.4w is excluded from this table, since, at the time of writing, its features are yet to be determined.

3. IPv6-based protocol stack for LPWANs

Over more than a decade, the IETF has developed a lightweight, IPv6-based protocol stack suitable for IoT devices (Fig. 1.a). Such protocol stack includes three components that have been designed for IoT scenarios: a 6LoWPAN/6Lo adaptation layer, the IPv6 Routing protocol for Low-power and lossy networks (RPL) [8], and the Constrained Application Protocol (CoAP) [9]. However, to provide the best fit for LPWAN technologies, the IPv6-based protocol stack assumed by the IETF LPWAN WG presents some particularities (Fig. 1.b). We now review the IoT-specific components of the lightweight IPv6-based protocol stack and justify the protocol stack modifications made for LPWANs.

The 6LoWPAN adaptation layer was developed to enable and optimize IPv6 over IEEE 802.15.4 networks [10]. 6LoWPAN provides IPv6 and UDP header compression (which saves energy and bandwidth resources), fragmentation (which allows carrying 1280-byte packets as required for IPv6 over the 127-byte maximum payload size of IEEE 802.15.4 frames), and an optimized version of the IPv6 neighbor discovery protocol (which offers parameter and device discovery for constrained devices). Subsequently, 6lo adaptation layers have reused 6LoWPAN components to support IPv6 over other IoT technologies [2]. However, 6LoWPAN/6Lo-style of IPv6 adaptation is not suitable for the extreme constraints of LPWANs. For this reason, the IETF LPWAN WG has developed the SCHC adaptation layer, specifically designed for LPWAN technologies, as explained in the next section.

At the network layer, a routing protocol is required for technologies that support the mesh topology, such as IEEE 802.15.4 or Z-Wave. RPL is the routing protocol designed by the IETF for IoT networks. RPL is optimized for data collection applications, while minimizing IoT device memory and energy consumption. However, since LPWAN technologies are based on the star topology, a routing protocol is not needed for LPWANs, which simplifies the corresponding protocol stack.

Finally, CoAP is a lightweight request/response application-layer protocol, based on the same architectural principles as HTTP, albeit with significantly lower complexity and overhead (e.g. its base header, without options, has a size of 4 bytes). While CoAP was originally designed to be transported over UDP (with optional end-to-end reliability and congestion control supported by CoAP itself), issues with middleboxes, such as UDP-unfriendly corporate firewalls, have led to the recent design and publication of a CoAP specification over TCP [11]. However, the larger TCP header size and the connection establishment overhead are inadequate for LPWANs, thus only UDP is assumed at the transport layer for LPWANs.

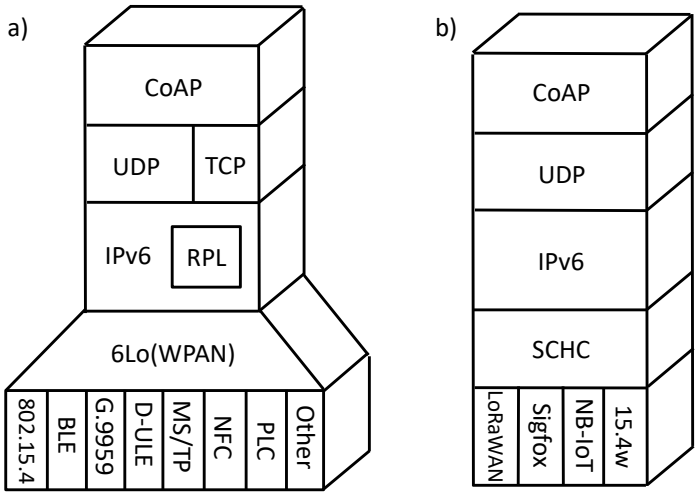


Figure 1. a) 6LoWPAN/6Lo IPv6-based protocol stack, b) LPWAN IPv6-based protocol stack

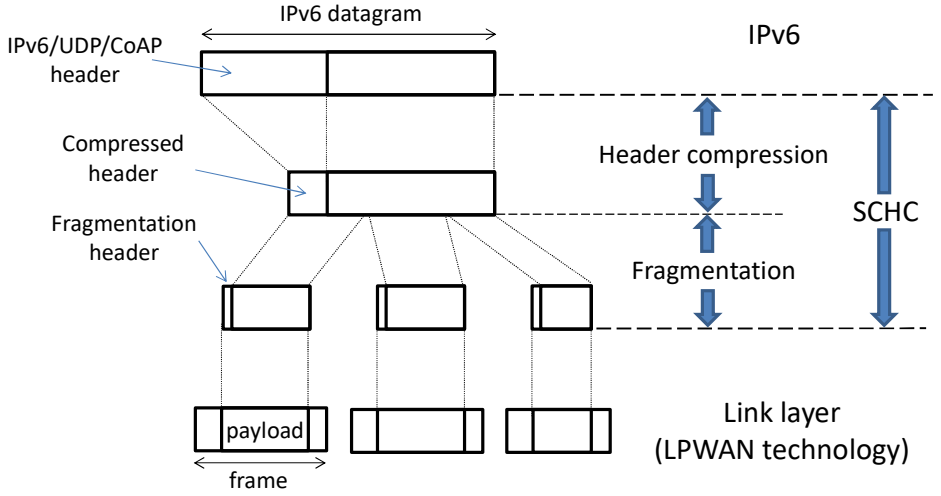


Figure 2. SCHC functionality overview: header compression and fragmentation

4. SCHC adaptation layer

This section describes the SCHC adaptation layer. SCHC is located between IPv6 and an underlying LPWAN technology. SCHC comprises two sublayers: header compression, and fragmentation (Fig. 2). The next two subsections present the main design principles and features of these two sublayers, respectively.

4.1. Header compression

Without proper adaptation, IP-based protocols would introduce a large overhead over LPWANs, since typical packet header sizes are significant when compared with the extremely low LPWAN frame payload sizes. Several header compression mechanisms have been developed in the past for efficient packet transmission over different technologies. Work in this area has been carried out since the 90s, when Van Jacobson

proposed a mechanism based on exploiting intraflow packet header redundancy to compress TCP/IP headers over slow serial links [12]. Subsequently, specialized header compression mechanisms have been designed for the characteristics of different constrained environments. The last such IP-based packet header compression efforts are Robust Header Compression (ROHC) [13] and 6LoWPAN header compression. We now review these two mechanisms, we highlight why they are not suitable for LPWANs, and we then present SCHC header compression.

4.1.1. Use of ROHC over LPWANs

ROHC was designed to compress network- and transport-layer headers of multimedia flows over low bitrate and high packet loss rate links, such as 3G cellular links. ROHC exploits packet header redundancy within a packet flow. To this end, packets are initially sent uncompressed, and subsequently only packet header differences are sent (after being efficiently encoded). In ROHC, an IPv6/UDP header may typically be compressed down to a minimum size of 3 bytes. Packet header information is maintained in a context on both compressor and decompressor sides. ROHC defines signaling that allows a decompressor to report to the other endpoint when context is damaged, e.g. due to channel losses. Such event causes context desynchronization, which is solved by transmitting an uncompressed header. However, this behavior is unsuitable for the capacity constraints of LPWANs. Furthermore, ROHC has not been defined to compress the CoAP header.

4.1.2. Use of 6LoWPAN header compression over LPWANs

6LoWPAN header compression was designed for efficient IPv6 (and UDP) packet transmission over IEEE 802.15.4 networks. ROHC-style header compression was considered too complex for the resource-constrained devices that characterize such networks. In order to reduce context desynchronization problems, 6LoWPAN header compression is partly based on stateless techniques, by leveraging the receiver ability to reconstruct some IPv6 header fields based on layer two header fields, plus a statistical expectation that other IPv6 header fields will carry values that are typical in 6LoWPAN networks. A bitmap at the start of the compressed header format indicates what fields have been compressed and how they can be decompressed. Stateless UDP header compression is also supported. Because stateless approaches cannot compress global IPv6 addresses, a stateful, yet quasi-static mechanism based on network-wide shared context is also used in 6LoWPAN. 6LoWPAN provides no method to compress any application-layer protocol header (when 6LoWPAN was designed, CoAP had not yet been created).

With 6LoWPAN header compression, a typical 48-byte IPv6/UDP header can be compressed down to a 7-byte format. This result is suitable for the maximum payload size in IEEE 802.15.4 frames, which is in the order of ~100 bytes. However, for an underlying technology with a frame payload size of ~10 bytes, as occurs in many LPWAN scenarios, a 6LoWPAN-compressed IPv6/UDP header would incur too high an overhead.

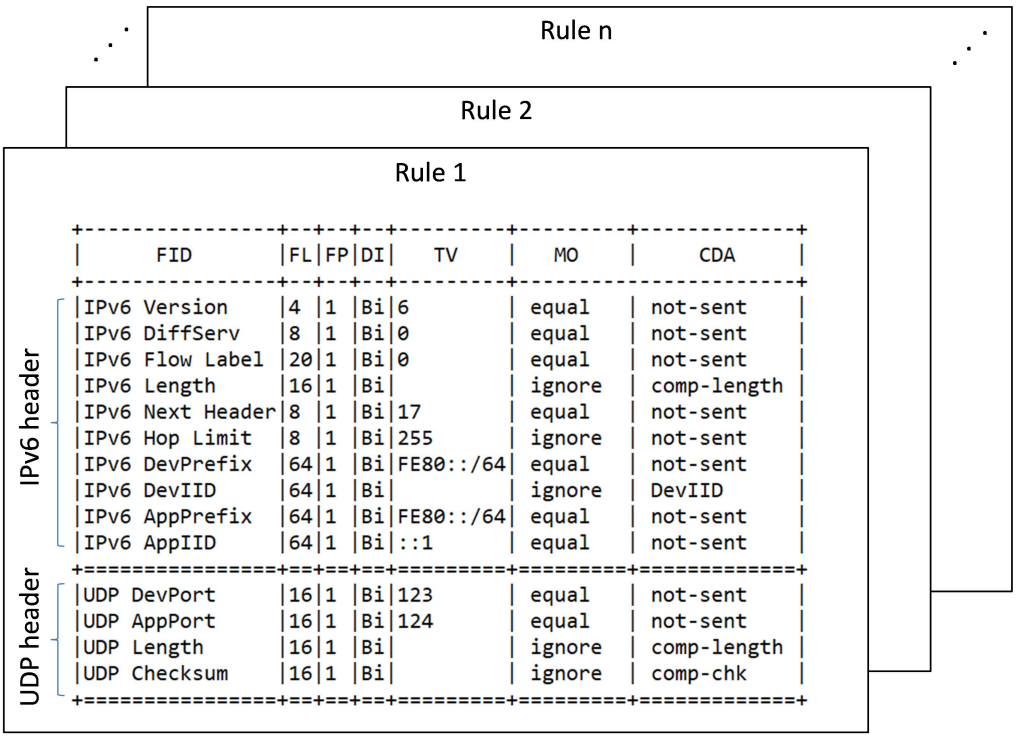
4.1.3. SCHC header compression

SCHC header compression has been purposefully designed for LPWANs, and is applicable to protocols such as IPv6, UDP and CoAP. SCHC relies on static context shared between the compressor and the decompressor, which leverages a priori knowledge of the traffic to be compressed. In fact, new applications are not expected to be frequently installed on an LPWAN device over its lifetime. The static context approach avoids the complexity of context resynchronization mechanisms and the need for receiver feedback, while allowing ultralightweight header compression.

In SCHC, a context is defined as a set of Rules, each one provided with a Rule identifier (Rule ID). A Rule comprises a set of descriptions of how each packet header field is to be compressed (Fig. 3). A Rule may be used for the compression of one or more protocol headers, e.g., an IPv6 header, the set of IPv6/UDP/CoAP headers, etc.

When a packet needs to be sent, the SCHC compressor first selects the Rule in the context that best matches the header format and header field values of the packet being handled. Then, the sender replaces the original packet header by the Rule ID corresponding to this Rule. When a Rule ID cannot unambiguously represent a complete packet header, a compression header residue is generated. The concatenation of the Rule ID and the compression residue (if any) constitute the compressed header. The Rule ID size is expected to be small, while still allowing the encoding of a large number of Rules (e.g. a 1-byte Rule ID supports a Rule space of up to 256 different Rules). When receiving a compressed packet, the decompressor reconstructs the original packet header based on the received compressed header and on the stored context.

a)



FID	FL	FP	DI	TV	MO	CDA
IPv6 Version	4	1	Bi	6	equal	not-sent
IPv6 DiffServ	8	1	Bi	0	equal	not-sent
IPv6 Flow Label	20	1	Bi	0	equal	not-sent
IPv6 Length	16	1	Bi		ignore	comp-length
IPv6 Next Header	8	1	Bi	17	equal	not-sent
IPv6 Hop Limit	8	1	Bi	255	ignore	not-sent
IPv6 DevPrefix	64	1	Bi	FE80::/64	equal	not-sent
IPv6 DevIID	64	1	Bi		ignore	DevIID
IPv6 AppPrefix	64	1	Bi	FE80::/64	equal	not-sent
IPv6 AppIID	64	1	Bi	::1	equal	not-sent
UDP DevPort	16	1	Bi	123	equal	not-sent
UDP AppPort	16	1	Bi	124	equal	not-sent
UDP Length	16	1	Bi		ignore	comp-length
UDP Checksum	16	1	Bi		ignore	comp-chk

b)

Description component	Definition and relevant details
Field IDentifier (FID)	An identifier that uniquely designates a protocol and a header field.
Field Length (FL)	The size of the header field, expressed in bits.
Field Position (FP)	The occurrence of the header field the description applies to. The FP is relevant when a header field can be present more than once in a header.
Direction Indicator (DI)	The packet direction the description applies to. It may take 3 values: *Up* and *Dw* (for packets sent and received by the LPWAN device, respectively), and *Bi* (for packets that travel in both directions).
Target Value (TV)	The value compared with the header field value.
Matching Operator MO)	The operator used to compare the TV and the header field value. Examples of MOs defined are the following: - *equal*: the result is "true" if the header field value is equal to the TV. - *ignore*: the result is "true" regardless of the header field value.
Compression/Decompression Action (CDA)	A description of the compression and decompression operation to be applied to a header field once the TV and header field value have been compared by using an MO. Examples of CDAs defined include: - *not-sent*: the header field is elided (compr.) and recovered from the stored Rule (decompr.). - *comp-length*: the header field is elided (compr.) and deduced at the receiver (decompr.). - *DevIID*: the LPWAN device Interface Identifier is elided (compr.) and reconstructed based on the device link layer address (decompr.). - *comp-chk*: the UDP checksum is elided (when compressing) and computed again (decompr.).

Figure 3.a) Example of a SCHC Rule (hereafter called Rule 1), designed for compressing IPv6 and UDP header fields. Each row in the Rule is a description of how the corresponding packet header field is to be compressed or decompressed. b) The components of a field description, their definition and relevant details. An IPv6/UDP packet header whose values match the TVs in Rule 1 can be fully compressed, yielding no compression residue

4.2. Fragmentation

IPv6 requires any underlying layer to support the transmission of packets of at least 1280 bytes. This measure was introduced in the IPv6 specification with the aim of achieving high performance (e.g. throughput) for data transmission over a presumed resource-rich Internet. However, LPWAN networking is fundamentally different, as it has been designed for infrequent message exchanges of short-sized payloads. In fact, some LPWAN technologies and scenarios offer an extremely short maximum frame payload size, even down to ~10 bytes. Even after applying the highly efficient SCHC header compression, many IPv6 packets will not fit into a single LPWAN frame. Besides, neither LoRaWAN nor Sigfox mode supports fragmentation and reassembly functionality. To overcome this issue, fragmentation is used at the SCHC adaptation layer, in the form of a sublayer located below the header compression one (Fig. 2).

In order to provide a solution for fragmentation over LPWANs, 6LoWPAN fragmentation was first considered as a possible basis. However, 6LoWPAN fragmentation had been designed for IEEE 802.15.4 networks, which present significant differences with LPWANs. First, IEEE 802.15.4 networks are often deployed as mesh networks, which requires 6LoWPAN fragmentation to handle out-of-sequence fragment delivery. Since LPWANs follow the star topology, fragmentation over LPWANs can avoid the related complexity. Secondly, the maximum frame payload size in IEEE 802.15.4 is up to one order of magnitude greater than the LPWAN ones. Thus, minimizing fragmentation header overhead is a considerably stronger requirement for the latter. In fact, the 6LoWPAN fragmentation header yields an overhead of 4-5 bytes per fragment, which is too high for ~10-byte LPWAN maximum frame payload sizes, as it would exacerbate frame encapsulation overhead. Leveraging the star topology of LPWANs, and using short-sized fragment identifiers, SCHC fragmentation supports a variety of options and mechanisms with even a single-byte fragmentation header size. Finally, a singular characteristic of LPWANs is the severe, even extreme, message rate limitations in some technologies. Under such circumstances, each LPWAN frame transmission becomes very expensive. However, any fragment loss (e.g. due to wireless link corruption) would lead to unsuccessful delivery of the whole higher layer packet being carried. In LPWANs, amortizing the scarce transmission resources consumed by retransmitting only the lost fragments may be desirable. However, 6LoWPAN fragmentation does not offer fragment retransmission, as of today. In order to provide flexibility to satisfy the heterogeneous needs of different LPWAN technologies or scenarios, SCHC fragmentation offers three fragment delivery reliability modes: No-ACK, ACK-Always, and ACK-on-Error.

No-ACK is a best-effort mode whereby fragment retries are not supported, and the fragment receiver does not inform the fragment sender regarding the transmission outcome. Both ACK-Always and ACK-on-Error provide selective fragment retransmission mechanisms (i.e. data integrity), based on Acknowledgments (ACKs) issued by the fragment receiver. The fragment receiver sends an ACK only after a *window* of fragments (i.e. a subset of the fragments carrying an IPv6 packet) has been

transmitted. An ACK reports whether each fragment of a window has been received or not. For efficiency, this information is encoded by means of a bitmap, where the *n*-th bitmap bit indicates whether the corresponding *n*-th fragment has been received or not. In ACK-Always, the fragment receiver unconditionally sends an ACK after a window of fragments. In contrast, in ACK-on-Error, the ACK is only sent when at least one fragment in the window has been lost, except in the last window, where an ACK is always sent to indicate whether the fragmented packet transmission has been successful. In order to avoid low performance due to ACK losses, in ACK-on-Error, upon reception of the last fragment of a packet, the receiver may send ACKs reporting missing fragments from the whole packet. While the frequent feedback in ACK-Always allows early detection of severe link problems, ACK-on-Error reduces message overhead.

Even though these fragmentation mechanisms have been designed to transport long IPv6 packets, the mechanisms can equally be applied to non-IP data messages.

5. Performance evaluation

We next evaluate SCHC, focusing on both header compression and fragmentation mechanisms.

5.1. Header compression

Fig. 4 illustrates the header compression performance of ROHC, 6LoWPAN, and SCHC, when applied to an IPv6/UDP/CoAP header. For the sake of comparison, an uncompressed header is also included in the figure. We assume the header uses IPv6 global addresses.

The main drawback of ROHC is that packets intended for context initialization or resynchronization are sent uncompressed. In LPWANs, this would represent low performance, further degraded by the need to apply fragmentation to such packets when the underlying LPWAN technology maximum frame payload size is ~10 bytes. In addition, such packets need to carry an additional ROHC header to describe their content, yielding a negative compression gain for them. In addition, CoAP compression is not supported by ROHC.

6LoWPAN-style header compression can reduce the size of the IPv6/UDP/CoAP header by a factor close to 5. However, the resulting header size is still too large for the frame payload sizes in many LPWAN scenarios.

In contrast with ROHC and 6LoWPAN, SCHC can yield a 3-byte IPv6/UDP/CoAP compressed header, which is a much better fit for LPWANs. This result can be obtained for a Rule optimized for a specific packet header (e.g. Rule 1 in Fig. 3), assuming a 1-byte Rule ID. For comparison purposes, Fig. 4 also includes the case of SCHC header compression where the Rule used produces a 2-byte compression residue.

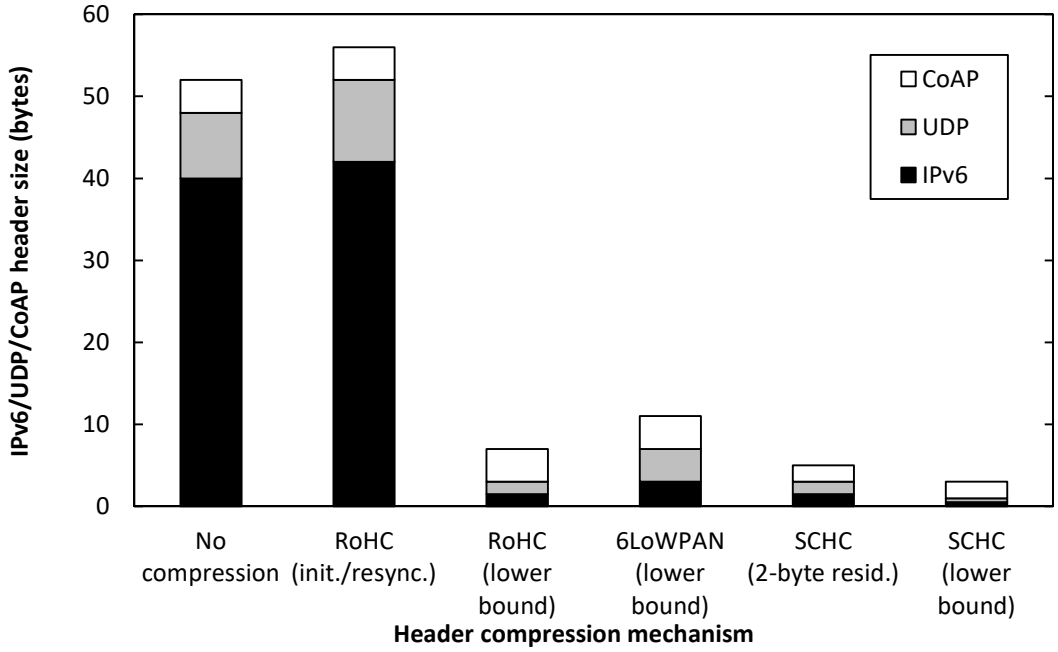


Figure 4. Comparison of header compression mechanisms applied to an IPv6/UDP/CoAP header

5.2. Fragmentation

We next evaluate the performance of the three SCHC fragmentation modes (No-ACK, ACK-Always, and ACK-on-Error), in terms of the average number of fragment transmission attempts and the number of ACKs, for the range of packet sizes required by IPv6, and for different Frame Loss Rate (FLR) values. We assume a 10-byte maximum frame payload size, uncorrelated frame losses, and equal uplink and downlink FLR values. For ACK-Always and ACK-on-Error, we also study the impact of the window size. In order to investigate the upper bound of all performance parameters considered, an infinite number of retries is used. Results are shown in Fig. 5.

Since No-ACK neither supports fragment retries nor receiver feedback, it yields the lowest amount of transmitted frames, at the expense of low reliability. For large-sized or critical-data packets, ACK-based modes are recommended. While ACK-Always exhibits the highest overhead, both in number of fragment transmission attempts and in number of ACKs, it yields the highest PDR. ACK-on-Error behaves minimalistically, by sending ACKs only when fragments are lost (except for the mandatory ACK sent at the end of the packet transmission).

For ACK-based modes, increasing the window size (W) decreases the number of ACKs. However, it may also increase the fragment identifier size, and in turn, the fragment header size (F). Fig. 5.a) depicts how F=2 tends to increase the number of fragments by ~12%.

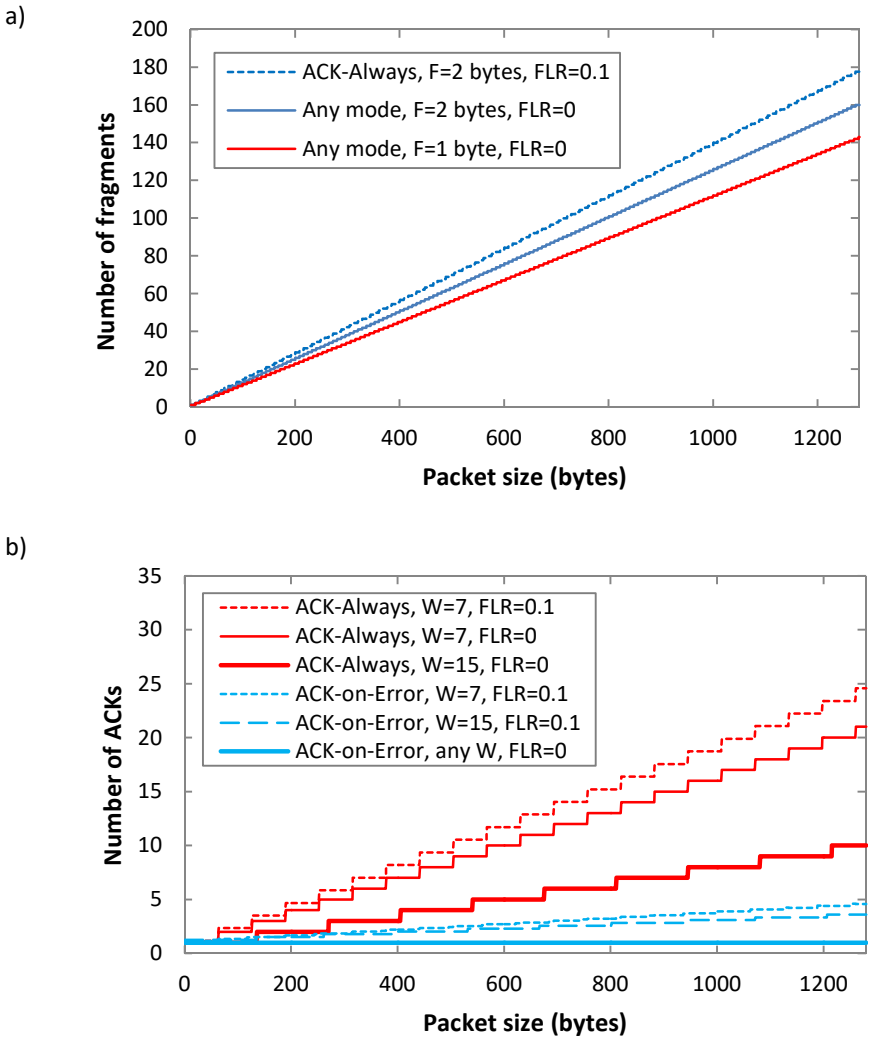


Figure 5. Performance evaluation of fragmentation modes and settings: a) average number of fragment transmissions, b) average number of ACKs

6. Open issues

At the time of writing, the design and standardization of SCHC is reaching completion. However, areas of additional functionality development and potential performance improvement have already been identified. This section reviews the main SCHC-related open research issues and standardization items.

6.1. Optimizing SCHC for each LPWAN technology

SCHC has been designed with the aim to satisfy common requirements of LPWAN technologies. Intentionally, SCHC offers generic functionality without specifying which mechanisms (e.g. fragmentation modes) or parameter settings (e.g. Rule ID size, fragmentation window size, etc.) need to be used over each specific LPWAN technology. This approach allows optimizing SCHC for each LPWAN technology, but requires specifications defining how SCHC is used over a given LPWAN technology. Currently, initial draft versions of such specifications have already been produced for

LoRaWAN, Sigfox, NB-IoT and IEEE 802.15.4w. Nevertheless, design and research work are still needed to complete, validate and evaluate the performance of SCHC over each specific LPWAN technology.

6.2. Context provisioning

A currently open question on SCHC header compression is how the context can be provided to the compressor and decompressor endpoints. Different alternatives include using: i) preinstalled context, ii) out-of-band means, and iii) an in-band provisioning protocol. Determining a suitable solution requires considering the crucial trade-off between configuration flexibility and bandwidth demand, as well as the capacity of the LPWAN technology in use.

6.3. Header compression for other protocols

SCHC header compression is based on a generic mechanism that needs to be applied in a specific way to each target protocol. At the time of writing, SCHC header compression has only been defined for IPv6, UDP and CoAP. However, further protocols may be used in the future in LPWAN scenarios, and may therefore benefit from SCHC header compression.

6.4. Packet-mode fragmentation

In the reliable fragment delivery modes offered by SCHC, an ACK is sent by the fragment receiver (always or conditionally) after the transmission of a window of fragments. An ideal reliable fragment delivery mechanism would be packet-oriented, i.e., a single ACK would report on the delivery success of all the fragments that carry an IP or non-IP data packet. However, fitting the fragment delivery report for a large packet in a single ACK may be challenging, given the extreme frame payload size constraints in some LPWAN technologies and scenarios. Different encoding techniques may be used at the receiver to report any lost fragments. Alternatives to the bitmap used in SCHC include using a list of lost fragment identifiers, and delta encoding applied to the identifiers of lost fragments. The efficiency of each technique depends on the frame error pattern. Determining the most suitable technique for each scenario needs to be investigated.

6.5. Security

LoRaWAN, Sigfox and NB-IoT offer encryption and authentication services. However, end-to-end security may also be needed in some IPv6-based LPWANs. There exist different approaches for securing CoAP, including use of Datagram Transport Layer Security (DTLS) and Object Security for Constrained RESTful Environments (OSCORE). Only the latter protects CoAP messages across intermediary nodes such as proxies, by transforming the messages into self-contained data structures with a header, a potentially encrypted payload, and an authentication field. Currently, support for compressing the OSCORE header by using SCHC is being developed.

Privacy is an open issue, as detection of (even encrypted) messages triggered by sensors detecting certain events may be exploited. Mitigation techniques (e.g. sending fake messages) are challenged by the capacity constraints of LPWAN technologies. The latter also pose a problem for key management, as documents such as certificates are usually bulky, and solutions are also needed in this space.

7. Conclusions

SCHC enables ultralightweight IPv6 support for LPWANs by providing specifically designed header compression and fragmentation functionality. Developed under a generic and flexible approach, SCHC can be configured for optimized operation over various underlying technologies (e.g. LoRaWAN, Sigfox, NB-IoT or IEEE 802.15.4w). SCHC is expected to become a fundamental contributor to the expansion of the Internet (of Things).

Acknowledgments

Carles Gomez has been partially supported by ERDF and the Spanish Government through project TEC2016-79988-P, AEI/FEDER, UE.

References

[1] U. Raza, P. Kulkarni, M. Sooriyabandara, “Low Power Wide Area Networks: An Overview”, IEEE Communications Surveys & Tutorials, Vol. 19, Issue 2, January 2017, pp. 855-873.

[2] C. Gomez, J. Paradells, C. Bormann, J. Crowcroft, "From 6LoWPAN to 6Lo: Expanding the Universe of IPv6-Supported Technologies for the Internet of Things", IEEE Communications Magazine, Vol. 55, Issue 12, pp. 148-155, December 2017.

[3] P. Thubert, A. Pelov, S. Krishnan, "Low-Power Wide-Area Networks at the IETF", IEEE Communications Standards, Vol. 1, Issue 1, March 2017, pp. 76-79.

[4] J. Haxhibeqiri, E. De Poorter, I. Moerman, J. Hoebeke, “A Survey of LoRaWAN for IoT: From Technology to Application”, Sensors, Vol. 18, 3995, November 2018.

[5] C. Gomez, J.C. Veras, R. Vidal, L. Casals, J. Paradells, “A Sigfox Energy Consumption Model”, Sensors, Vol. 19, 681, February 2019.

[6] Y.-P. E. Wang, X. Lin, A. Adhikary, A. Grövlen, Y. Sui, Y. Blankenship, J. Bergman, H.S. Razaghi, “A Primer on 3GPP Narrowband Internet of Things”, IEEE Communications Magazine, Vol. 55, Issue 3, March 2017, pp. 117-123.

[7] L. Feltrin, G. Tsoukaneri, M. Condoluci, C. Buratti, T. Mahmoodi, M. Dohler, R. Verdone, “Narrowband IoT: A Survey on Downlink and Uplink Perspectives”, IEEE Wireless Communications, Vol. 26, Issue 1, February 2019, pp. 78 – 86.

[8] H.-S. Kim, J. Ko. D.E. Culler, J. Paek, “Challenging the IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL): A Survey”, IEEE Communications Surveys & Tutorials, Vol. 19, Issue 4, September 2017, pp. 2502 – 2525.

[9] C. Bormann, A.P. Castellani, Z. Shelby, “CoAP: An Application Protocol for Billions of Tiny Internet Nodes”, IEEE Internet Computing, Vol. 16, Issue 2, March 2012, pp. 62-67.

[10] Z. Shelby, C. Bormann, “6LoWPAN: The Wireless Embedded Internet”, Vol. 43. John Wiley & Sons, August 2011.

[11] C. Bormann, S. Lemay, H. Tschofenig, K. Hartke, B. Silverajan, B. Raymor, “CoAP (Constrained Application Protocol) over TCP, TLS, and WebSockets”, RFC 8323, February 2018. (Available at https://tools.ietf.org/html/rfc8323, accessed on April 17th 2019.)

[12] V. Jacobson, “Compressing TCP/IP Headers for Low-Speed Serial Links”, RFC 1144, February 1990. (Available at https://tools.ietf.org/html/rfc1144, accessed on April 17th 2019.)

[13] K. Sandlund, G. Pelletier, L-E. Jonsson, “The RObust Header Compression (ROHC) Framework”, RFC 5795, March 2010. (Available at https://tools.ietf.org/html/rfc5795, accessed on April 17th 2019.)