



**HAL**  
open science

## Implementation of SCHC in NS-3 Simulator and Comparison with 6LoWPAN

Wael Ayoub, Fabienne Nouvel, Sarah Hmede, Abed Ellatif Samhat, Mohamad Mroue, Jean-Christophe Prévotet

► **To cite this version:**

Wael Ayoub, Fabienne Nouvel, Sarah Hmede, Abed Ellatif Samhat, Mohamad Mroue, et al.. Implementation of SCHC in NS-3 Simulator and Comparison with 6LoWPAN. 26th International Conference on Telecommunications (ICT), Apr 2019, HANOI, Vietnam. hal-02051757

**HAL Id: hal-02051757**

**<https://hal.science/hal-02051757>**

Submitted on 28 Feb 2019

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Implementation of SCHC in NS-3 Simulator and Comparison with 6LoWPAN

Wael Ayoub<sup>\*†</sup>, Fabienne Nouvel<sup>\*</sup>, Sarah Hmede<sup>†</sup>, Abed Ellatif Samhat<sup>†</sup>, Mohamad Mroue<sup>†</sup>  
and Jean-christophe Prévotet<sup>\*</sup>

<sup>\*</sup>Institut National des Sciences Appliquées de Rennes — IETR-INSA, Rennes, France.

<sup>†</sup> Faculty of Engineering - CRSI, Lebanese University, Hadath Campus, Hadath, Lebanon

Email<sup>\*</sup>: firstname.lastname@insa-rennes.fr

Email<sup>†</sup>: samhat@ul.edu.lb, mohamad.mroue@ul.edu.lb, sarah.hmede@outlook.com

**Abstract**—The rapid growth of IoT applications usage enables the Internet connectivity of a massive number of devices using different technologies. Most of these technologies, such as Low Power Wide Area Networks (LPWANs), are non-IP due to the difficulties of using IP on constrained devices. These nodes are characterized by more constraints with respect to other IoT technologies. According to [1], IPv6 offers many benefits for IoT, which motivated the IETF to form a Working Group (WG) to study and propose new solutions to run IPv6 on the new technologies of IoT [2], [3]. The key to solving this issue is the header compression mechanisms. In this paper, we analyze the two IETF standardized solutions, SCHC and 6LoWPAN, to compress IPv6 over constrained nodes within LPWAN. Based on [3], we implement the SCHC mechanism [4] in the network simulator NS3 [5]. We also show that SCHC protocol solution as an adaptation layer between the network layer and the link layer is better in term of header compression by providing a smaller header size compared to 6LoWPAN.

**Index Terms**—IoT communication, LPWAN, IPv6, SCHC, 6LoWPAN, NS3, Long-Range, and Header Compression.

## I. INTRODUCTION

In recent years, the Internet of Things (IoT) has attracted the attention of industry and researchers. Arm declared in [6], [7] that 100 billion chips had been shipped over 26 years between 1991 and 2017 to support connected devices, whereas the same quantity has been shipped since 2017. To respond to the exponential growth in the number of connected objects in the coming years, creating a reliable and efficient network infrastructure is essential. Apart from exceptional cases, connected objects transmit only a few data for long distances. In most applications, there is a real need for autonomy and a lifespan of several years. Existing networks, such as WIFI, Bluetooth, cellular networks, etc., do not respond to this need. That led to the emergence of new IoT communication networks developed to adapt to these requirements known as LPWANs.

The emerging of LPWANs has received considerable attention from the research community and industry. LPWANs studies and developments have obtained momentum in many domains such as smart city, traffic control, health care, water/energy/waste management, agriculture, etc. This overgrowth led to the presence of heterogeneous architectures, standards, middlewares, and a diversity of applications. Currently and for future, LPWAN technologies are required to

provide scalability, stability, and on wide areas, for End-Devices (ED). Furthermore, this heterogeneity is an obstruction for industry and developments, that usually seek a homogeneous solution to support the diversity of technologies to gain marketing and other purposes. Such technologies include LoRaWAN, Sigfox, DASH7, etc.

Previously, heterogeneity of technologies was solved in several ways. One of the most common solutions that researchers adopted was to create a middleware that can be adapted and be common between the different technologies. However, with the presence of new types of technologies, i.e., LPWAN, these middlewares are no more applicable and require a significant update. The objective is to create a common solution for heterogeneity that can be easily adapted to any new technology. In [1], authors show that IPv6 offers several features that make it a successful solution for IoT developments. IPv6 provides scalability, security, stateless address auto-configuration, and a set of complementary solutions for the IoT needs: e.g., 6LoWPAN, 6TiSCH, etc. Furthermore, IPv6 may be a solution for heterogeneity, and it supports mobility.

However, IPv6 solutions must consider the LPWANs constraints and challenges. These challenges can be summarized as follows:

- Large-scale constrained networks including a massive number of ED.
- Payload size support in the physical frame as low as 12 bytes, i.e., SigFox.
- Limited rate of frames on air per data.
- Low data rate and small bandwidth up to 250 kbps.
- Frame loss increases as the number of ED increases.
- Do not support fragmentation in case of large frames.
- Power consumption and radio constraints

Among these complementary solutions that IPv6 provides, we investigate in this paper two IPv6 mechanisms that consider the LPWANs constraints: IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN) [8] and Static Context Header Compression (SCHC) [1]. These two solutions, standardized by IETF, are used to extend the protocol stack of LPWAN technology with an adaptation layer. This layer converges from non-IP to IPv6-based technology which enables

homogeneous access for applications and services. Based on this study, we propose to use the SCHC protocol solution as an adaptation layer between the network layer and the link layer for the LPWAN technologies to reduce IPv6 header size.

This paper is structured as follows: In section II we illustrate related work. In Section III, we present the SCHC mechanism, explain the terms and fields followed by a brief, concentrated description of the header compression in 6LoWPAN. In Section IV, we illustrate the implementation procedure of the SCHC mechanism on the NS3 and explain the compression/decompression structure of the algorithm. In Section V, we draw a scenario to compare between SCHC and 6LoWPAN on NS3. Finally, Section VI provides a summary of the two mechanisms and the requirements considered in SCHC with the future work.

## II. RELATED WORK

A lot of studies have been carried out to compare the two solutions, but to our best knowledge, they do not propose a real implementation to compare them. In [9], authors propose LSCHC mechanism based on SCHC standard for compressing of IPv6. Also, authors address the memory constraint generated by SCHC mechanism and propose the LSCHC as a solution. The proposed mechanism solves the memory constraint of the device but adds constraints on the radio communication which is against the main goals of SCHC.

In [10], and [11], authors benefit from the use of 6LoWPAN mechanism in compressing IPv6 for LoRaWAN technology [12]. Moreover, in [13] authors propose Co-iOAM mechanism for compression based on 6LoWPAN. In [14], authors compare the 6LoWPAN with LPWANs layer by layer. A similar comparison on the use of 6LoWPAN will be given in section V. In [15], an adaptation layer is proposed based on the SCHC mechanism for compression and decompression. This solution deals with vehicular communication and moving devices. But SCHC mechanism is developed for the static device, so update and improvements on this mechanism are required to support mobility.

## III. HEADER COMPRESSION MECHANISMS

### A. SCHC

The IETF has established an LPWAN WG to explore a solution that adopts IPv6 over constrained networks in LPWANs. The proposed solution is based on two conditions:

- Solution for star topology networks that are characterized by small bandwidth and low power, and support for fragmentation.
- Static predetermined flow of data.

This WG has proposed the Static Context Header Compression (SCHC) [3] mechanism as a header compression scheme that compresses IPv6 header into less than 1 byte. However, and like its predecessor 6LoWPAN, SCHC is currently being updated to cover the IPv6/UDP/CoAP (Constrained Application Protocol) headers and to be adapted to other LPWANs technologies. SCHC compression mechanism considers static devices with predetermined context saved on the device and

server memory. This context does not change during packet transmission. SCHC avoids the complexity of synchronization mechanisms. These mechanisms are not required when dealing with LPWANs. Moreover, SCHC is an adaptation layer that is placed between the network and data link layer as 6LoWPAN. SCHC compresses the IPv6/UDP/CoAP headers into ruleID and sends the data to lower layers for transmission.

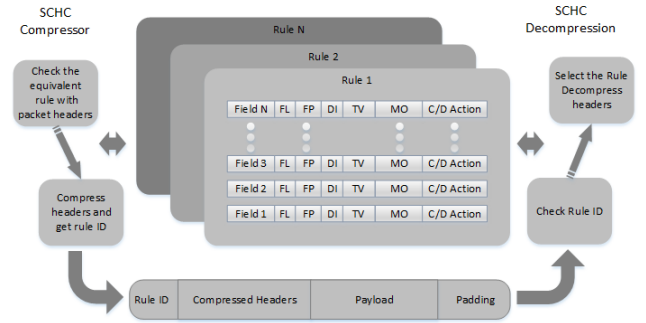


Fig. 1. SCHC Architecture.

The context saved on device and server consists of rules as shown in Fig. 1. Each rule consists of several fields. Each field corresponds to a parameter from the header. A rule holds the information of the header. The ruleID is formed by:

- Field ID (FID): A unique value used to represent the header field.
- Field Length (FL): Used to specify the header segment length.
- Field Position (FP): In case the field is an array, field position is used to select a definite item in the array.
- Direction Indicator (DI): Used to specify the direction of the packet, whether uplink (Up); data are from device to server, downlink (Dw); data are from the server to device, or Bidirectional (Bi).
- Target Value (TV): This field contains the value that ED received to compare with the saved value.
- Matching Operator (MO): This operator is applied in comparing received TV with saved one.
- Compression Decompression Action (CDA): It describes the method used beside each field to compress and decompress the TV.

### B. 6LoWPAN

IPv6 over Low-Power Wireless Personal Area Networks 6LoWPAN [16] offers a lot of advantages for low power wireless sensor networks and other forms of low power wireless networks like LPWANs. In late 2004, 6LoWPAN WG formed by the Internet Engineering Task Force, IETF [17] to enable IPv6 over IEEE802.15.4 networks. The mission of the 6LoWPAN WG was to build an adaptation layer under the IPv6 protocol on the network layer and above the IEEE 802.15.4. This open standard was aimed to run for low power wireless system that operates at the 2.4 GHz frequencies, but now it is adapted and updated to run with many other

wireless technologies such as Bluetooth, Power Line Communication, LoRaWAN, etc. In the 6LoWPAN standard, the WG defined the compression/decompression mechanisms and encapsulation/fragmentation that enable running IPv6 packet data on constrained wireless networks. Currently, 6LoWPAN is an adaptation layer that allows IPv6 packets to be carried efficiently within small link layer frames. Moreover, this adaptation layer defines one reduced header format for IPv6 and one reduced format for UDP packet headers. Today, 6LoWPAN is one of the most important technologies for the integration of IPv6 in smart objects based on Wireless Sensor Networks with low power, limited bandwidth, and reduced memory capabilities. Furthermore, the 6LoWPAN bridge translates the 6LoWPAN header of the packets received via the WPAN interface to the IPv6/UDP headers. These packets are transmitted through the MAC/PHY interface, and vice versa, transparently.

IPv6 header = 40 bytes

Ver	Traffic class	Flow label	Payload length	Next header	Hop limit	Source address	Destination address
-----	---------------	------------	----------------	-------------	-----------	----------------	---------------------

1- Within network = 2 bytes

Dispatch	Compression header
----------	--------------------

2- With known prefix of another network = 12 bytes

Dispatch	Compression header	CID	Hop limit	Destination address
----------	--------------------	-----	-----------	---------------------

3- With unknown prefix of another network = 20 bytes

Dispatch	Compression header	CID	Hop limit	Source address	Destination address
----------	--------------------	-----	-----------	----------------	---------------------

Fig. 2. 6LoWPAN Communication Scenarios.

Fig. 2 shows three communication scenarios for 6LoWPAN:

- In scenario 1, communication is within the 6LoWPAN network between two devices. In this case, the devices use the link-local addresses. 6LoWPAN can compress the IPv6 header down to 2 bytes.
- In scenario 2, communication is with a device outside the 6LoWPAN network, but the prefix of the external network is known. In this case, 6LoWPAN can compress the IPv6 header down to 12 bytes.
- In scenario 3, communication is similar to 2, but the prefix of the external network is unknown. In this case, 6LoWPAN compresses the IPv6 header down to 20 bytes.

#### IV. IMPLEMENTATION OF SCHC MODULE

In order to compare the two mechanisms, SCHC and 6LoWPAN, we used Network Simulator 3 (NS3) [5]. NS3 is free popular software for discrete event simulation. The 6LoWPAN mechanism was already implemented on NS3 but SCHC still not available. Based on [3], we created the SCHC module [4] that implements the functions of compression and decompression. Then, this module was integrated and tested in NS3. In the following, we provide a brief description of the compression and decompression mechanisms on SCHC. The

implementation, structure, and installation of the module on NS3 can be found in [4].

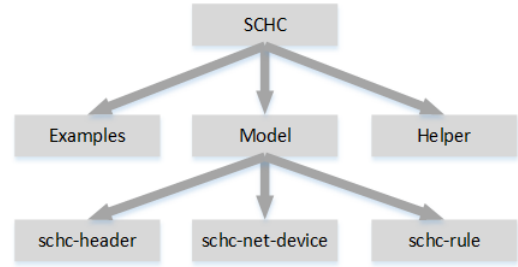


Fig. 3. SCHC Module Structure.

The implementation of the SCHC algorithm on NS3 was done using the C++ language. This choice was motivated by the fact that a C++ compiler is usually present on most available hardware platforms. SCHC has been structured and implemented in several files found under the SCHC module folder as shown in Fig. 3. A new module named SCHC was created to model the behavior of the SCHC protocol. This module is essentially a collection of classes that work together to describe the compression and decompression functions used in SCHC. The set of classes and files needed to simulate a network formed by devices that use SCHC as an adaptation layer are described in the following.

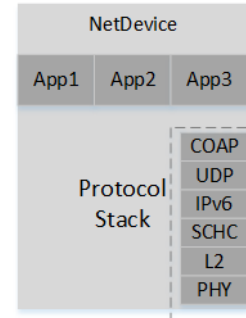


Fig. 4. Protocol Stack Installed in NetDevice.

##### A. helper

The helper file contains the SCHC-helper class that installs an SCHC adaptation layer between IPV6 and L2 layer (Data link layer). Furthermore, the SCHC-helper class was used to install a protocol stack that contains the SCHC protocol on the NetDevice as shown in Fig. 4.

##### B. Model

1) *SCHC-rule*: It specifies the rules contained in the SCHC context that defines the structure of the algorithm of the SCHC protocol. A rule will be of type structure formed by a ruleID and an array of type SCHC\_Rule\_field (rule\_field) containing the description of the different fields that make up an IPV6/UDP packet as shown in Fig. 5. The ruleID is a binary value and consists of a series of zeros and ones.

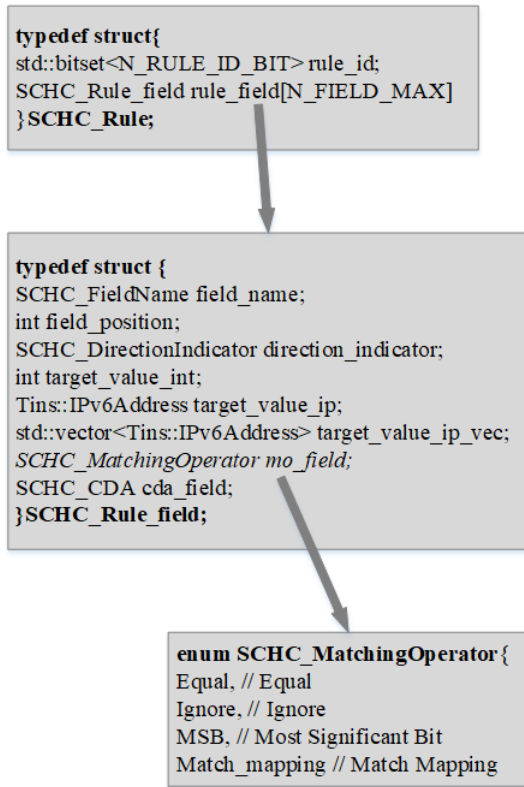


Fig. 5. SCHC Rule Structure.

2) *Compression mechanism*: The compression mechanism compresses the headers of the packet received from the network layer using the CompressSCHC function. This function follows the flowchart structure represented in Fig. 6. Before starting the SCHC compression algorithm, separation for the IP and the UDP headers from the payload is done. The goal of the compression is to find a ruleID, represented by a few bits, to identify the contexts of the fields in the packet header UDP/IP. To select a ruleID, the packet header fields will be matched with the saved context in memory. The best ruleID that matches the context of the packet header will be selected. Then this ruleID is used to compress the packet header. As shown in Fig. 6, the SCHC compression starts by loading all the rules saved in memory in a "for loop." The algorithm iterates the rules one by one and checks for each iteration if all the fields of the rule, are in agreement with the IP and UDP header fields. In case of a mismatch, the algorithm goes to the next rule. Then, the algorithm loads the fields of each rule in a "for loop." In a first step, the algorithm chooses the field description based on its direction, based on the direction indicator (DI) parameter. Any field description with no proper DI will be ignored, and the algorithm will check the next field. The rule will be ignored if all the packet fields do not have an appropriate field description with the required DI. Then, the following rule will be checked. With the successful match of DI, the algorithm recognizes the fields regarding their Field Position (FP). The rule will be discarded

if the FP does not match. After successful matches of DI and FP with the packet header context, the algorithm starts by comparing each field of the packet header (example: as shown in Fig. 2 of the IPv6) with the Target Value (TV) saved on the rule. This comparison between received TV and saved one is based on the rules specified by the received Matching Operator (MO) as shown in Table I. After matching all the fields and the packet header with the selected rule, the compression starts based on the actions of the rule. After that, a new "for loop" is created to compress the packet header values. The compression mechanism is performed by applying, for each field, the compression action corresponding to the rule. Once the compression operations are done, the algorithm returns the ruleID. In case of more than one rule matches, the rule that provides better header compression will be selected. If no rule matches, the headers will be sent uncompressed in the field specified by SCHC for transmitted headers. Depending on the L2 layer of the technology and allowed frame size, SCHC decides whether to fragment the packet or not.

3) *Decompression mechanism*: On the network side, decompression mechanism is different from the that on Device. The network side holds rules that correspond to several devices. A compressed packet received on the network side needs the device address and ruleID. The SCHC decompressor loads the ruleIDs corresponding to this device. These ruleIDs are available in the database of the network under the device identity. Otherwise, on the device side, only the ruleID is needed, since each device holds its own rules. The decompression algorithm is implemented in the sch-net-device.cc file following the structure is shown in Fig. 6. In decompression, the input is the context containing the "compression header" and the "ruleID." The method returns a Boolean value, which will be true when the package has been properly uncompressed, or false if an error has occurred. First, the algorithm starts by separating the SCHC headers from the payload and makes the first check to determine if the rule is present in the context. Then, it starts the process of rebuilding the original package. At this stage, the algorithm begins to analyze all the fields described in the rule, check the direction of the communication flow, apply CDA method to decompress the value, and rebuild all the fields to obtain the IP/UDP packet headers.

## V. SIMULATION RESULTS

To evaluate the performance of the SCHC protocol, we performed three simulation scenarios as shown in Fig. 7. In the three scenarios, we sent a ping from the device to server passing by the gateway. In scenario A, packets transmitted without compression. In scenario B, packet headers are compressed using 6LoWPAN compression. In scenario C, the headers are compressed using the SCHC compression. In SCHC, we consider the ruleID with the compression headers space together as a header. The SCHC context header was composed of a ruleID formed of 3 bits, and n bytes for the header compress. Table II shows the compression gain in case of normal transmission and when compressing the headers

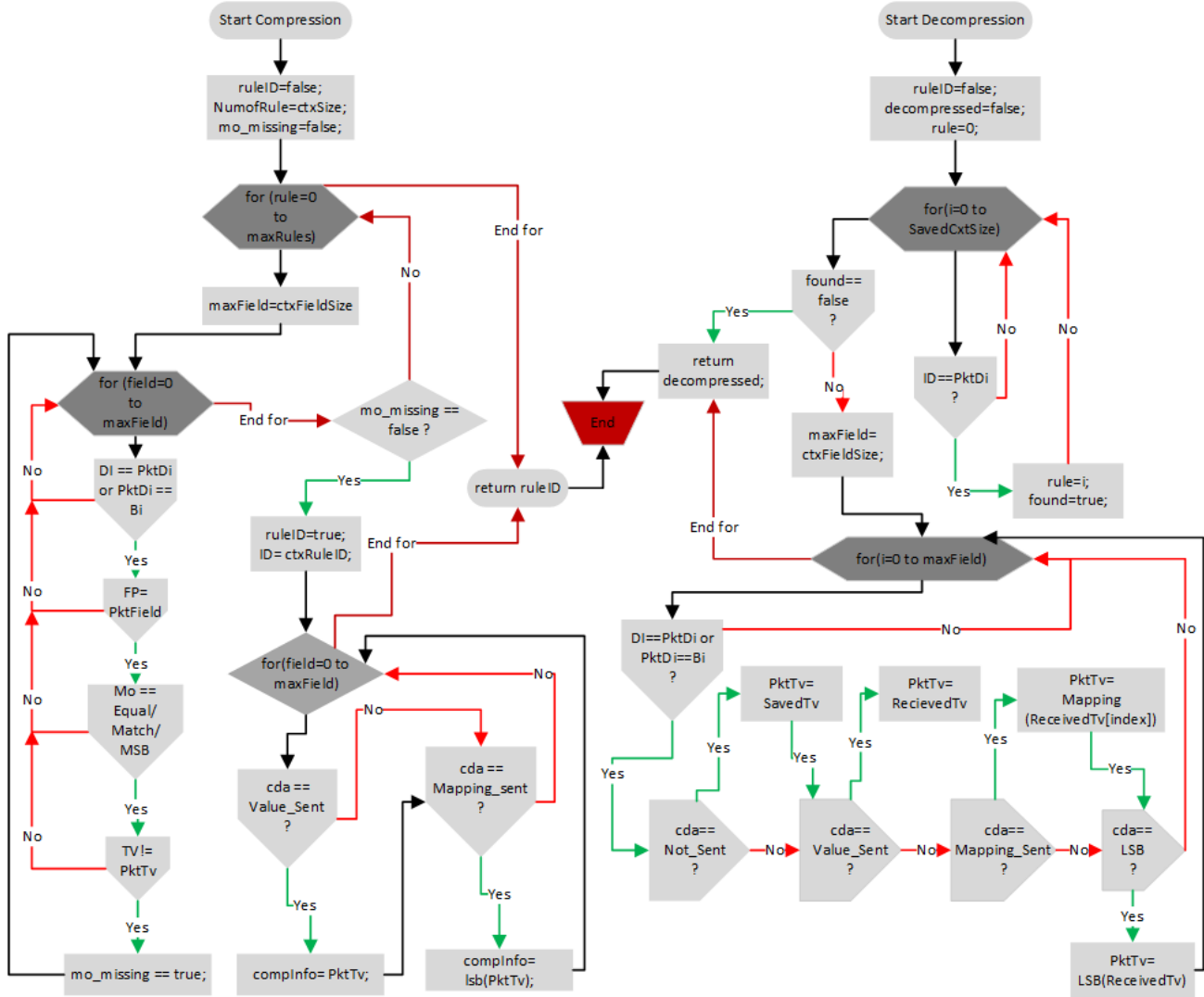


Fig. 6. SCHC Algorithm Structure for Compression and Decompression.

using 6LoWPAN and SCHC. This table shows that SCHC can reach a compression factor of 0.9 compared to 0.43 for 6LoWPAN. The compression ratio is calculated as follows:

$$C = \frac{L_{original} - L_{compressed}}{L_{original}} \% \quad (1)$$

where  $C$  is the compression ratio,  $L_{original}$  represents the packet header length, and  $L_{compressed}$  is the header compression length.

In term of header compression, SCHC device starts by sending a map for the UDP/IP<sub>v6</sub> headers used by the server. Consider the scenario where the ruleID was "001" (3bits), and the compression header fields contain the UDP/IP<sub>v6</sub> header values as shown in Table I. The first transmitting packet header was 5 bytes, and all the following packets headers were only 3 bits as ruleID and 5 bits for padding. As shown in Table I, most of the fields are known to the server. So device rule configuration does not need to be sent. The only sent fields are the UDP

source and destination ports which are equal to 4 bytes. The 5 bytes header is composed of 3 bits for ruleID, 4 bytes data in the "compression headers" and 5 bits of padding. In the case of mobility, SCHC mechanism does not support mobility and only considers static devices. In the case of network topology, SCHC only deals with star networks which make it successful for LPWANs technologies but not for mesh IoT technologies such as Zigbee, etc. SCHC has been a solution to IoT networks with radio constraints, but as headers increase, the amount of data saved in memory increases and the searching time for the applicable rule will also increase. This requires better data management mechanism and searching algorithm to manage the stored data between device and server. In summary, SCHC is a trade-off between radio constraints, memory usage, and processing time.

Consider a static device communicating with a server. All data transmitted from the device to server holds the same header. Even with the use of header compression, we will show

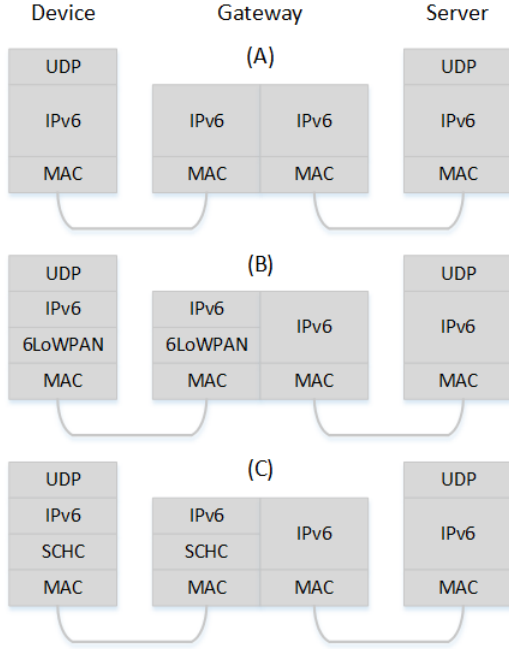


Fig. 7. Testing Scenarios on NS3.

TABLE I  
RULE SENT

Rule 1						
Field	FP	FL	DL	TV	MO	C/D
IPv6 Header Fields						
Version	1	4	bi	6	equal	not-sent
Traffic Class	1	8	bi	0x00	equal	not-sent
Flow Label	1	20	bi	0x000000	ignore	not-sent
Payload Length	1	16	bi	None	ignore	Compute-Length
Next Header	1	8	bi	17	equal	not-sent
Hop Limit	1	1	bi	30	ignore	not-sent
Prefix Source id	1	64	bi	2001:63:80:8	equal	not-sent
Source id	1	64	bi	::2	equal	not-sent
Prefix id Destination	1	64	bi	2001:63:80:9	equal	not-sent
Destination	1	64	bi	::2	equal	not-sent
UDP Header Fields						
Source Port	1	2	bi	5682	ignore	value-sent
Destination Port	1	2	bi	5555	ignore	value-sent
Length	1	2	bi	None	ignore	Compute-length
Checksum	1	2	bi	None	ignore	Compute-checksum

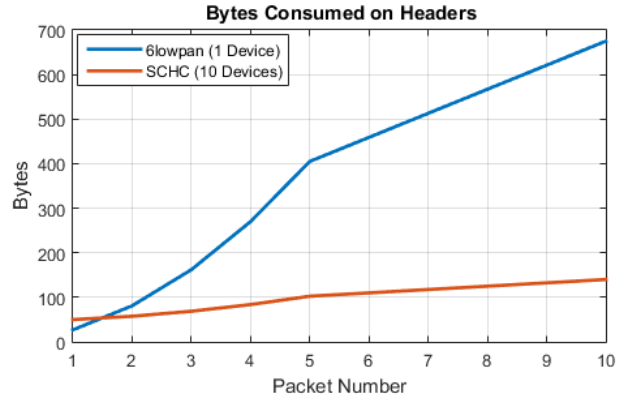


Fig. 8. Bytes of the headers generated from 10 packets.

TABLE II  
THREE SCENARIOS COMPARISON

Scenario	A:IPv6	B:6LoWPAN	C:SCHC
Data rate	high	medium	low
Memory Usage	low	low	High
Mobility support	Yes	Yes	No
Interoperability	Yes	Yes	Specific technologies
Scalability	Yes	Yes	No
Network topology	Mesh	Mesh/star	Star
Header compression	no compression	>6 bytes <37 bytes	>1 bit <6 bytes
48 bytes header compression	48 bytes	27 bytes	5 bytes then 3 bits
Compression gain	0%	43%	90%

the number of bytes generated to send only ten packets by each standard. During the transmission of the packets, the necessary header fields hold unchangeable values such as IP and the port of communication between the source and destination. To show the amount of these bytes, we set at the gateway a byte counter to count the headers size generated by each standard after the transmission of 10 packets. As shown in Fig. 8, we compare one 6LoWPAN device with ten SCHC devices. The results show that more than 600 bytes are consumed as headers to transmit ten packets. While ten devices using SCHC standard consume less than 150 bytes. Therefore, in the case of radio constraints, SCHC standard is preferable.

## VI. CONCLUSION

This work analyzed two header compression solutions used to compress IPv6 protocol to run on constrained devices. The mechanism of the SCHC protocol has been implemented on NS3. The mechanism of 6LoWPAN was already implemented and proposed on NS3. We simulated three scenarios, and the results showed that SCHC is more efficient for the LPWAN context. However, SCHC is a mechanism that is characterized by a static context. Thus, SCHC could be considered as a solution for only predetermined fixed flows which is the case of most LPWANs technologies. Furthermore, SCHC uses a single static context to save the different rules, and rules can cover several layers of the network stack. In case we

have two IPv6/UDP flows to the same IPv6 host, then one IPv6 header and two different UDP ports are considered. That corresponds to the case of running two concurrent applications on different UDP ports. SCHC mechanism would include a rule for each flow, resulting in a similar context. In this case, SCHC mechanism stores two similar versions of the fields for the IPv6 header. That storage increases memory usage in constrained devices with duplicate information. The limitations mentioned above will be the subject of future work. Our implemented module only deals with the compression of IPv6/UDP packets. However, this work could be extended to support the compression of application layer protocol CoAP.

#### REFERENCES

- [1] W. Ayoub, M. Mroue, F. Nouvel, A. E. Samhat, and J. c. Prvotet, "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.
- [2] 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.
- [3] C. G. D. B. J. Z. A. Minaburo, L. Toutain, "Lpwan static context header compression (schc) and fragmentation for ipv6 and udp draft-ietf-lpwan-ipv6-static-context-hc-17," October 22, 2018.
- [4] W. Ayoub and S. Hmede, "Schc mechanism module on ns3." <https://gitlab.insa-rennes.fr/Wael.Ayoub/schc-module-ns3>, October 2018.
- [14] H. Al-Kashoash and A. Kemp, "Comparison of 6lowpan and lpwan for the internet of things," *Australian Journal of Electrical and Electronics Engineering*, vol. 13, pp. 268–274, December 2017. © 2017 Engineers Australia. This is an Accepted Manuscript of an article published by Taylor & Francis in Australian Journal of Electrical and Electronics Engineering on 1 December 2017, available online: <http://www.tandfonline.com/10.1080/1448837X.2017.1409920>. Uploaded in accordance with the publisher's self-archiving policy.
- [5] [https://www.nsnam.org/releases/September 4, 2018](https://www.nsnam.org/releases/September%204%2C%202018).
- [6] C. Yanamadala and K. Marneweck, "Why silicon security may be crucial for next design." <https://pages.arm.com/Webinar-recording-silicon-security-crucial-for-iot-device-typ.html?aliId=42779824>, October 2018.
- [7] E. A. Kobus Marneweck, Senior Product Manager, "An introduction to the arm cortex-m35p processor, white paper." <https://pages.arm.com/rs/312-SAX-488/images/Whitepaper-ARM-Cortex-M35P-Whitepaper.pdf>, 2018.
- [8] H. A. A. Al-Kashoash and A. H. Kemp, "Comparison of 6lowpan and lpwan for the internet of things," *Australian Journal of Electrical and Electronics Engineering*, vol. 13, no. 4, pp. 268–274, 2016.
- [9] K. Q. Abdelfadeel, V. Cionca, and D. Pesch, "Lschc: Layered static context header compression for lpwans," in *Proceedings of the 12th Workshop on Challenged Networks, CHANTS '17*, (New York, NY, USA), pp. 13–18, ACM, 2017.
- [10] P. Weber, D. Jckle, 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.
- [11] 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.
- [12] W. Ayoub, A. E. Samhat, F. Nouvel, M. Mroue, and J. Prvotet, "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.
- [13] R. Ballamajalu, S. V. R. Anand, and M. Hegde, "Co-ioam: In-situ telemetry metadata transport for resource constrained networks within ietf standards framework," in *2018 10th International Conference on Communication Systems Networks (COMSNETS)*, pp. 573–576, Jan 2018.
- [15] R. Sanchez-Iborra, J. Snchez-Gmez, J. Santa, P. J. Fernndez, 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.
- [16] D. Minoli, *IPv6 Over LowPower WPAN (6Lowpan)*. Wiley, 2013.
- [17] P. Thubert, "Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks." RFC 6282, Sept. 2011.