



# AN INTERNET PROTOCOL ADAPTATION LAYER FOR UNDERWATER ACOUSTIC NETWORKS

**Benoît Parrein**<sup>1\*</sup>

**Nils Morozs**<sup>2</sup>

**Laurent Toutain**<sup>3</sup>

<sup>1</sup> Nantes Université, Polytech Nantes, CNRS, LS2N, UMR 6004, F-44000 Nantes, France

<sup>2</sup> School of Physics, Engineering and Technology, University of York, York YO10 5DD, United Kingdom

<sup>3</sup> IMT Atlantique, CNRS, IRISA, UMR 6074, F-35000 Rennes, France

## ABSTRACT

The Internet of Underwater Things (IoUT) is on the way. The key application areas of IoUT include ocean engineering, maritime security and environmental monitoring. To achieve interoperability between different underwater communicating systems and/or to ensure connectivity from the air to underwater (and reversely), an Internet Protocol (IP) adaptation layer is necessary. In this paper, we propose Underwater-SCHC (U-SCHC) based on the Static Context Header Compression protocol (SCHC), a standard at the Internet Engineering Task Force (IETF) for terrestrial Internet of Things. The purpose of the paper is to describe the protocol encapsulation in the context of very low bit rates and small payloads as are usually observed in underwater acoustic systems. As a result, we reach the header compression ratio of 99.74% with just a single bit allocated to the UDP/IPv6 stack.

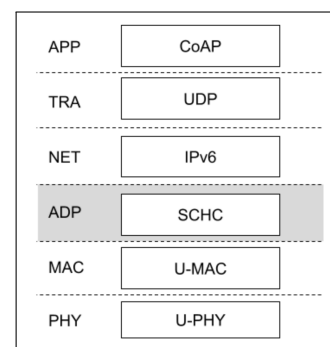
**Keywords:** *header compression, IP adaptation layer, SCHC protocol, underwater acoustic networks.*

## 1. INTRODUCTION

Prosaically, if we talk about the Internet of Underwater Things (IoUT), we have to put the Internet Protocol (IP) somewhere. Pragmatically, if you compare the UDP/IPv6 traditional header size of 48 Bytes with typical frame sizes in underwater acoustic networks, e.g. 64 bits in the JANUS standard [1], it is apparent that a standard IP layer

*\*Corresponding author: benoit.parrein@univ-nantes.fr.*

**Copyright:** ©2023 Benoît Parrein et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.



**Figure 1.** Full 6-layer stack covered in this article, including the adaptation layer supported by the SCHC protocol.

and the underwater acoustic link layer are not compatible each other. Nevertheless, including IP and Transport layers in this extreme context of communication can enable: *i)* interoperability in a field where proprietary protocols are the norm *ii)* end-to-end IP communication with more reliable transport *iii)* useful extra services as fragmentation and end-to-end encryption to cite a few. This short paper describes U-SCHC as a way of encapsulating not just UDP/IP layers but also an application protocol (here CoAP) by considering the Static Context Header Compression (SCHC) proposed in terrestrial Low Power Wide Area Network (LP-WAN) [2]. In other words, this short paper presents a way to insert a troll's foot in a baby's shoe. This paper is not about the implementation of such extreme encapsulation (by example on packet driven simulator or programmable acoustic modem) that will be covered in another study with classical benchmark.

## 2. RELATED WORK

IP adaptation layers in Underwater Wireless Acoustic (UWA) networks are not new. To the best of our knowledge, [3] made the first formulation in the mid 90's. In his Master of Science thesis, Reimers underlined already the importance of interoperability and standardisation alignment. On page 67 of his report, before focusing on Forward Error Correcting (FEC) code inside the payload, the author writes about "TCP/UDP/IP similar packet header" (IPv4 here of course) without giving information on the real implementation. More recently, Schneider [4] reaches the size of 16 bits for UDP/IP header by using entropy coding. However the compression scheme is proposed only for IPv4 and it is not suitable for dynamic changes (e.g. variable number of nodes) in the network. The work of Sun *et al.* [5] is the most notable reference in terms of the IP adaptation layer for UWA networks. The authors proposed a compression scheme for UDP/IP version 4 and version 6 based on the 6LoWPAN (IPv6 over Low-power Wireless Personal Area Networks) adaptation layer from the Internet of Engineering Task Force (IETF). Their UDP/IPv6 header is compressed in 16 bits. As SCHC overpassed 6LoWPAN for terrestrial networks, we aim to overpass the 16-bit limit with U-SCHC in this study.

## 3. SCHC AND COAP BACKGROUND

In the following, we provide some background for Static Context Header Compression (SCHC) and Constrained Application Protocol (CoAP).

### 3.1 SCHC in a nutshell

The Static Context Header Compression (SCHC) protocol (pronounced "CHIC") is part of a long tradition of header compression in the IETF. SCHC proposes to compress the UDP/IPv6 stack (48 Bytes of header) into constrained terrestrial Low Power Wide Area Networks (LPWAN) such as LoRaWAN or Sigfox (with a small code of 1 Byte length usually). For that type of terrestrial IoT networks, bandwidth and frame size are very limited indeed (around 100 bit/s and 10 Bytes respectively in the extreme case). With such nominal values, the 6LoWPAN adaptation protocol is not suitable (usually 250 kbit/s of data rates and more than 100 Bytes in frame size are needed for IEEE802.15.4). Generally speaking, SCHC works with static rules that define the processing for each IPv6 and UDP field (and for the upper layer if necessary). These rules feature Compression-Decompression Actions

(CDA) that are usually based on not sending the field, because it can easily be reconstructed from the rule itself or take values from lower layers (such as the MAC address or device IID for example). The variable fields, such as the IPv6 addresses or UDP ports are usually covered by a residue. The set of rule id and residue constitute the final compressed value of the header. It approaches theoretically the entropy limits by suppressing all redundant information. The purpose of this work is to precisely define the rules for UWA networks. The SCHC protocol is a standard at the IETF [6]. More information about SCHC can be found in [2].

### 3.2 Some words on CoAP

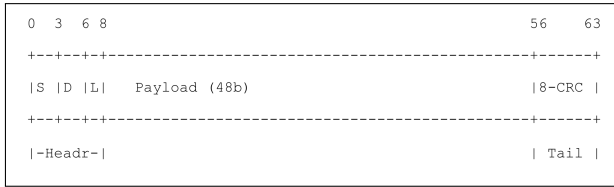
We expect extremely low code-length for UDP/IPv6 encapsulation that allows to integrate Application layer such as Constrained Application Protocol (CoAP) in our UWA case study. CoAP is also a proposed standard at the IETF [7]. CoAP is an application-layer protocol dedicated for resource-constrained Internet devices. Roughly speaking, CoAP translates HTTP (mainly defined by GET and POST methods) for constrained networks such as wireless sensor networks. The smallest normal CoAP header is 4 Bytes long. In this study, we keep just the Message ID, Methods (GET, POST and some other useful methods) and Unique Resource Identifier (URI) fields from the original CoAP header to build a compatible header less than 1 Byte in length on the top of UDP/IPv6 stack.

## 4. COAP/SCHC FOR UNDERWATER WIRELESS ACOUSTIC NETWORKS

This section is the core of the study considering the CoAP/SCHC encapsulation for UWA networks called U-SCHC in this work.

### 4.1 Assumption on the MAC layer

To build our full stack in Fig. 1, it is necessary to make some assumptions on the typical requirements of the UWA MAC layer. In our study we target a frame size of 64 bits at the MAC layer. This length matches with many commercial modems and with the JANUS standard [1]. This length is also less than 10 Bytes, which is the most known constrained LPWAN, so it is quite challenging to integrate UDP/IPv6 on that type of transfer unit. Fig. 2 shows the generic MAC frame considered in this work. This frame has an 8-bit header including Source (S) and Destination (D) fields of 3 bits each followed by a length



**Figure 2.** Our generic underwater acoustic 64-bit MAC frame with an 8-bits header, 48-bit payload and 8-bit CRC at the tail.

of payload of 2 bits that can encode 4 distinct sizes of payload. At the bit 8, the payload starts with possible variable sizes e.g 8, 16, 32 and 48 bits at a maximum. The tail of the frame is composed of and 8-bit CRC that checks the integrity at the MAC layer (just above the acoustic physical layer). Compliant with the OSI model, we are totally agnostic about the type of the MAC protocol above the NET layer e.g contention or contention free for example. We just assume the possibility to retrieve Device IID (from S and D fields) and the length of the payload from the MAC layer in a cross-layer manner.

#### 4.2 Definition of the rules

The definition of the rules in SCHC is the most critical (and difficult) part. To get them, you must fit to your case study by considering all the possible cases and requested services. All remaining bits must carry non redundant information to reduce the final entropy. To write our own rules we consider 2 types of IPv6 address prefix (Link-Local and Global-Unicast) and interaction with the MAC layer above to retrieve the source/destination addresses to get the full IP addresses. Above IP layer, we consider fixed source and destination port and just 1 application for our study. In Fig. 3 shows the first rule (Rule 0) that encompasses all of the above. This rule is for Link-Local addresses only (supporting communications between a gateway and underwater nodes in a star topology for example). The length, source and destination addresses are simply ignored in that rule because the information is already in the MAC layer described above. The UDP fields are also ignored (constant here) or reconstructed by the rule. In the same way, we construct the Rule 1 for the Global-Unicast prefix (e.g 2001:db8::/64). This requires just 1 bit to define the rule id (to distinguish just Link-Local or Global-Unicast prefix usage). The distinction between those 2 types of address allows to separate 2 types

```

RULE 0
+-----+-----+-----+-----+
| FID | FL|FP|DI | TV | MO | CDA |
+-----+-----+-----+-----+
|IPv6 Version |4|1|Bi|6 | equal |not-sent |
|IPv6 DSCP |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 SrcPrefix |64|1|Bi|FE80::/64 | equal |not-sent |
|IPv6 SrcIID |64|1|Bi| | ignore |SrcIID (S) |
|IPv6 DstPrefix |64|1|Bi|FE80::/64 | equal |not-sent |
|IPv6 DstIID |64|1|Bi|:::1 | ignore |DstIID (D) |
+-----+-----+-----+-----+
|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 |
+-----+-----+-----+-----+

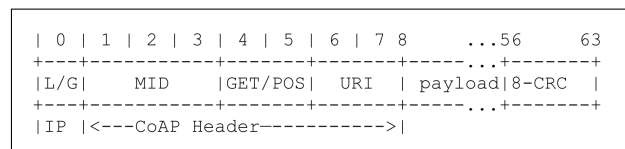
```

**Figure 3.** Rule 0 for the Link-Local prefix (inspired from [2]).

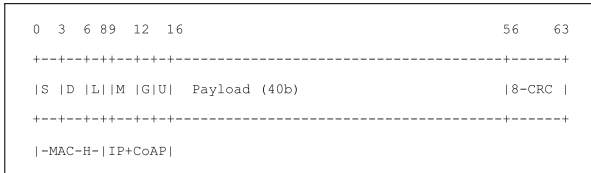
of network operations, for example: underwater neighbour discovery at a link coverage and interactions with the shore for unicast addressing and routing.

#### 4.3 The full CoAP/SCHC encapsulation

With 1 bit required for the UDP/IPv6 protocol, we can reserve 7 bits for CoAP to be Byte-aligned (in comparison with 4 Bytes of standard CoAP header [7]). As stated above in the Section 3.2, we keep only the Message ID (MID), the Methods and the URI fields in our crafted CoAP header. For the MID field, we just consider 3 bits (16 bits in the original RFC) to support message deduplication, acknowledgement and other lightweight reliable actions. The Methods field is reduced here to 2 bits to support at least GET and POST methods and some success and error code in return. Finally, 2 bits are reserved for the URI to uniquely identify the object we want to handle in the application layer e.g Temperature, Pressure,... Fig. 4 sums up the 8-bit long CoAP/SCHC header. Fig. 5 shows the full packet of Section 4.1 including the final payload length of 40 bits (some bits can be



**Figure 4.** CoAP/SCHC compressed header (8 bits long in total).



**Figure 5.** Full MAC, IP, Transport and Application packet: 1 Byte for MAC header, 1 Byte for CoAP/SCHC header, 5 Bytes of payload and 1 Byte of Cyclic-Redundancy-Check (CRC).

taken from there for a more complex MAC layer) and the 8-CRC as a tail. Tab. 1 summarises all the compression header gain we have with U-SCHC and compares it with no compression and with the proposition of Sun *et al.* [5] (no CoAP encapsulation was available on this work). Actually with 1 bit over 384 we almost reach the 100% compression ratio of the UDP/IPv6 standard encapsulation.

## 5. CONCLUSIONS

In this paper, we propose U-SCHC, the Underwater version of the Static Context Header Compression (Underwater-”CHIC”) to adapt UDP/IPv6 stack for underwater wireless acoustic networks. By adding application layer capabilities such as CoAP protocol, we just need 8 bits to code the full header (1 single bit for the transport and network layers). The next work is to implement this idea on a packet driven simulator (such as DESERT [8]) to measure the quality of service. An open implementation of SCHC in Python is available here <sup>1</sup>.

**Table 1.** Performance comparison in header compression ratio between the work of Sun *et al.* [5] and our Underwater-SCHC (Underwater-”CHIC”).

Protocol	No Comp.	Work of [5]	U-SCHC
CoAP	32	n.a.	7
UDP	64	8	0
IPv6	320	8	1
Total (in bits)	384	16	1
<b>Ratio (%)</b>	<b>0.00</b>	<b>95.83</b>	<b>99.74</b>

<sup>1</sup> <https://github.com/openschc/openschc>

## 6. REFERENCES

- [1] J. Potter, J. Alves, D. Green, G. Zappa, I. Nissen, and K. McCoy, “The JANUS underwater communications standard,” in *2014 underwater communications and networking (UComms)*, pp. 1–4, IEEE, 2014.
- [2] C. Gomez, A. Minaburo, L. Toutain, D. Barthel, and J. C. Zuniga, “IPv6 over LPWANs: Connecting low power wide area networks to the Internet (of Things),” *IEEE Wireless Communications*, vol. 27, no. 1, pp. 206–213, 2020.
- [3] S. P. Reimers, “Towards Internet Protocol over sea-water (IP/SW): Forward error correction (FEC) using hamming codes for reliable acoustic telemetry.,” tech. rep., NAVAL POSTGRADUATE SCHOOL MONTEREY CA, 1995.
- [4] T. Schneider, “Transmitting Internet Protocol packets efficiently on underwater networks using entropy-encoder header translation,” in *2016 IEEE/OES Autonomous Underwater Vehicles (AUV)*, pp. 241–245, IEEE, 2016.
- [5] Y. Sun and T. Melodia, “The Internet underwater: An IP-compatible protocol stack for commercial undersea modems,” in *Proceedings of the 8th International Conference on Underwater Networks & Systems, WUWNet ’13*, (New York, NY, USA), Association for Computing Machinery, 2013.
- [6] A. Minaburo, L. Toutain, C. Gomez, D. Barthel, and J. Zuniga, “SCHC: Generic framework for static context header compression and fragmentation.” RFC 8724, April 2020. DOI 10.17487/RFC8724, <https://www.rfc-editor.org/info/rfc8724>.
- [7] Z. Shelby, K. Hartke, and C. Bormann, “The Constrained Application Protocol (CoAP).” RFC 7252, June 2014.
- [8] F. Campagnaro, R. Francescon, F. Guerra, F. Favaro, P. Casari, R. Diamant, and M. Zorzi, “The DESERT underwater framework v2: Improved capabilities and extension tools,” in *2016 IEEE Third Underwater Communications and Networking Conference (UComms)*, pp. 1–5, 2016.