SCHC Working Group
Internet-Draft
Intended status: Informational
Expires: 10 August 2025

A. Pelov
IMT Atlantique
P. Thubert

A. Minaburo
Consultant

6 February 2025

Static Context Header Compression (SCHC) Architecture
draft-ietf-schc-architecture-04

Abstract

This document defines the SCHC architecture.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 10 August 2025.

Table of Contents

1. Introduction

The IETF LPWAN WG defined the necessary operations to enable IPv6 over selected Low-Power Wide Area Networking (LPWAN) radio technologies. [rfc8376] presents an overview of those technologies.

The Static Context Header Compression (SCHC) [rfc8724] technology is the core product of the IETF LPWAN working group and was the basis to form the SCHC Working Group. [rfc8724] defines a generic framework for header compression and fragmentation, based on a static context that is pre-installed on the SCHC endpoints.

This document details the constitutive elements of a SCHC-based solution, and how the solution can be deployed. It provides a general architecture for a SCHC deployment, positioning the required specifications, describing the possible deployment types, and indicating models whereby the rules can be distributed and installed to enable reliable and scalable operations.

2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Terminology

* C/D. Compression and Decompression.

* Context. All the information related to the Rules for SCHC Header, Non-Compression, C/D and F/R and CORECONF_Management.

* FID. Field Identifiers, describing the name of the field in a protocol header.

* F/R. Fragmentation and Reassembly.

* Rule. A description of the header fields to performs compression/ decompression, fragmentation/reassembly, SCHC end points and CORECONF_Management.

* SCHC Gateway (end point). The SCHC end point located upstream, e.g., in a Network Core Software.

* SCHC Device (end point). The SCHC end point located downstream, e.g., in a constrained physical device.

* SCHC end point. An entity (e.g., Device, Application and Network Gateway) involved in the SCHC process. Each SCHC end point will have its Set of Rules (SoR), based on the profile, the protocols, the device, the behaviour and a Set of Variables (SoV).

* SCHC Instance. The session between SCHC end points in two or more peer nodes operating SCHC to communicate using a common SoR and a matching SoV. There are 2 SCHC Instances or more involved per SCHC stratum, one for the SCHC Stratum Header and one or more for the SCHC payload, i.e., the SCHC-compressed data.

* SCHC Instance Manager. Provides the management of SCHC end points, the SoR of each end point and the dialog between hosts to keep the SCHC synchronization, and the establishment of SCHC Instances with peer nodes.

* SoR (Set of rules). Group of Rules used in a SCHC end point. The set of rules contains Rules for different nature as compression, no compression, fragmentation, SCHC end points and CORECONF management.

* SoV (Set of Variables). External information that needs to be known to identify the correct protocol, the SCHC Instance id, and the flow when there is one.

* SCHC Stratum. A SCHC Stratum is the SCHC analoguous to a classical layer in the IP architecture, but its operation may cover multiple IP layers or only a subset of a layer.

4. Building Blocks

This section specifies the principal blocks defined for building and using the SCHC architecture in any network topology and protocol.

4.1. SCHC Stratum (plural: strata)

A SCHC Stratum is the SCHC analoguous to a classical layer in the IP architecture, but its operation may cover multiple IP layers or only a subset of a layer, e.g., IP only, IP+UDP, CoAP, or OSCORE [rfc8824]. The term stratum is thus used to avoid confusion with traditional layers. Also, SCHC Strata are not stacked, though they can be nested.

The SCHC Stratum data in a datagram is composed of a SCHC Stratum Header (which may be compressed to the point that it is fully implicit and thus elided), a SCHC payload (that is used to uncompress a section of the SCHC datagram), and user payload that is unaffected by the SCHC Stratum. The SCHC Stratum operation requires at least 2 end points, one for the SCHC Stratum Header and one or more for the SCHC payload.

A SCHC compressed packet may contain multiple stratum data, to be handled by sequential (nested) SCHC Strata, where the inner (nested) Stratum operates within the payload of the outter (nesting) Stratum.

A SCHC Stratum is instantiated in participating nodes as a pair of SCHC end points, and matching SCHC end points in communicating nodes are associated to form a SCHC end point. A SCHC end point may be Point to point (P2P), or Point to Multipoint. A P2MP SCHC end point is unidirectional, meaning that all the SCHC datagrams are generated by the same node. A P2P SCHC end point may be unidirectional or bidirectional, symmetrical (between peers) or asymetrical (between a device and an application).

A SCHC end point operates datagram fragmentation and/or data compression and decompression, and maintains the state and timers associated with the Stratum operation over the consecutive packets or fragments.

The SCHC end points that handle the compression for nested Strata might differ for the same packet, meaning that the payload of a given Stratum might be compressed/uncompressed by a different entity, possibly in a different node. It results that the degree of compression (the number of Strata) for a given packet may vary as the packet progresses through the layers inside a node and then through the network.

4.2. Discriminator

The key to determine how to decompress a SCHC Stratum Header in a stratum is called a Discriminator.

The Discriminator is typically extrinsic to the stratum data.

It may be found in the packet context, e.g., the ID of the interface, VLAN, SSID, or PPP SCHC Instance on which the packet is received.

It may also be received in the packet, natively or uncompressed from a nesting stratum, e.g.:

* A source and destination MAC or IP addresses of the packet carrying SCHC packets

* A source and destination port number of the transport layer carrying SCHC packets

* A next header field

* An MPLS label

* A TLS Association

* Any other kind of connection id.

The Discriminator enables to determine the SCHC end point that is used to decompress the SCHC Stratum Header, called a SCHC Stratum Header end point.

Once uncompressed, the SCHC Stratum Header enables to determine the SCHC end point, called a SCHC Payload end point, that is used to restore the packet data that is compressed in the stratum.

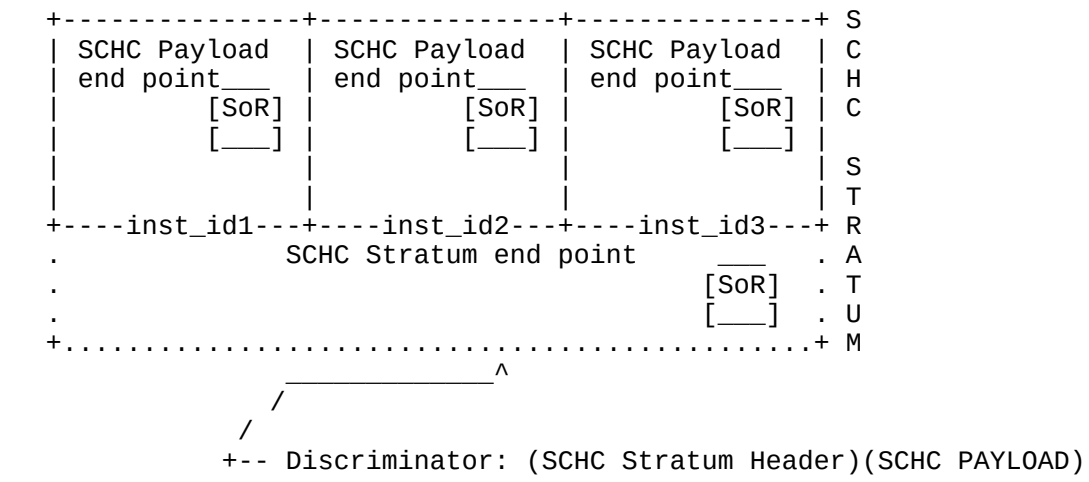
4.3. SCHC Stratum Header end point

The SCHC Stratum Header end point manages the SCHC Stratum Headers and provides the information and the selection of a SCHC Payload end point.

The rules for that end point might be such that all the fields in the SCHC Stratum Header are well-known, in which case the header is fully elided in the stratum data and recreated from the rules.

The rules might also leverage intrinsic data that is found in-line in the stratum data, in which case the first bits of the stratum data are effectively residue to the compression of the SCHC Stratum Header. Finally, the rules may leverage extrinsic data as the Discriminator does.

Figure 1 illustrates the case where a given stratum may compress multiple protocols SCHC Instances, each corresponding to a different SCHC Payload end point.



Each SCHC Payload End point uses its own Set of Rules,
but share the same SCHC Stratum Header.

Figure 1: SCHC end points for a stratum

4.3.1. SCHC Stratum Header

The SCHC Stratum Header carries information that is required for the SCHC strata operation. For example, it selects the correct end point and checks the validity of the datagram. There IS NOT always a RuleID if there is only one Rule for the SCHC Stratum Header, whose length is 0. The SCHC Stratum Header format is not fixed, and the SoR MUST have one or more Rules describing the formats. SCHC Stratum Header contains different fields. For end point, when the SCHC Stratum Header may identify the next protocol in the stack, the format of the SCHC Stratum Header takes the format as Figure 2 shows.

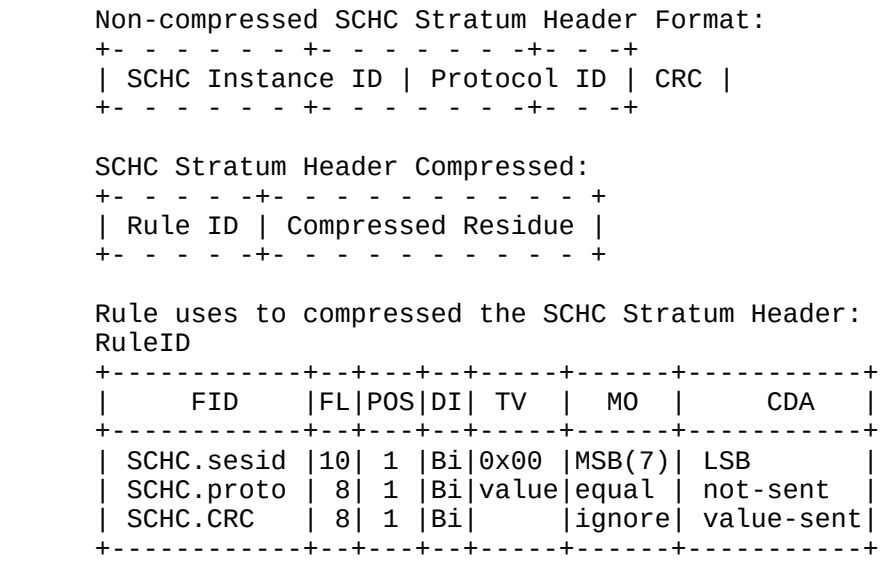


Figure 2: Example of SCHC Stratum Header Format and the corresponding Rule

In this example the Rule defines:

* A SCHC InstanceID is 10 bits length and it is used to identify the SoR used for this end point of SCHC.

* A Protocol ID in 1-byte length giving the value send in the layer below the SCHC packet to identify the uncompressed protocol stack.

* And A CRC. The CRC field is 8 bits length and covers the SCHC Stratum Header and the SCHC packet from error. When it is elided by the compression, the layer-4 checksum MUST be replaced by another validation sequence.

4.4. SCHC Payload end point

SCHC Payload end point is characterized by a particular SoR common with the corresponding distant end point. The [rfc8724] defines a protocol operation between a pair of peers. In a SCHC strata, several SCHC end points may contain different SoR.

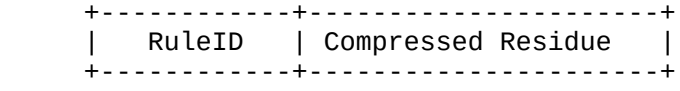
When the SCHC Device is a highly constrained unit, there is typically only one end point for that Device, and all the traffic from and to the device is exchanged with the same Network Gateway. All the traffic can thus be implicitly associated with the single end point that the device supports, and the Device does not need to manipulate the concept. For that reason, SCHC avoids to signal explicitly the end point identification in its data packets.

The Network Gateway, on the other hand, maintains multiple end points, one per SCHC Device. The end point is derived from the lower layer, typically the source of an incoming SCHC packet as a discriminator in the Figure 1. The end point is used in particular to select the set of rules that apply to the SCHC Device, and the current state of their exchange, e.g., timers and previous fragments.

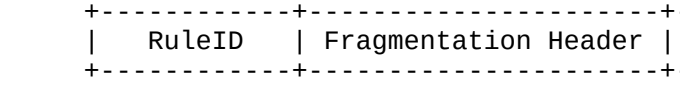
4.4.1. SCHC Payload

When compressed, the SCHC Payload is composed of a RuleID followed by the content described in the Rule. The content may be a C/D packet, a F/R packet, a CORECONF_Management or a Non Compressed packet. As defined in the [rfc8724], the SCHC packet for C/D is composed of the Compressed Header followed by the payload from the original packet. Figure 3 shows the compressed header format that is composed of the RuledID and a Compressed Residue, which is the output of compressing a packet header with a Rule.

C/D Compressed Packet:



F/R Compressed Packet:



CORECONF_Management Compressed Packet:

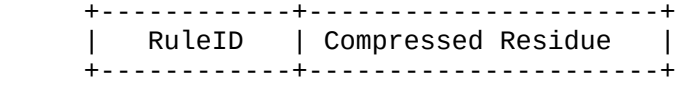


Figure 3: SCHC Packet

4.5. SCHC Profiles

A SCHC profile is the specification to adapt the use of SCHC with the necessities of the technology to which it is applied. In the case of star topologies and because LPWAN technologies [rfc8376] have strict yet distinct constraints, e.g., in terms of maximum frame size, throughput, and directionality, also a SCHC end point and the fragmentation model with the parameters' values for its use.

Appendix D. "SCHC Parameters" of [rfc8724] lists the information that an LPWAN technology-specific document must provide to profile SCHC fragmentation for that technology.

As an example, [rfc9011] provides the SCHC fragmentation profile for LoRaWAN networks.

4.6. SCHC Operation

The SCHC operation requires a shared sense of which SCHC Device is Uplink (Dev to App) and which is Downlink (App to Dev), see [rfc8376]. In a star deployment, the hub is always considered Uplink and the spokes are Downlink. The expectation is that the hub and spoke derive knowledge of their role from the network configuration and SCHC does not need to signal which is hub thus Uplink vs. which is spoke thus Downlink. In other words, the link direction is determined from extrinsic properties, and is not advertised in the protocol.

Nevertheless, SCHC is very generic and its applicability is not limited to star-oriented deployments and/or to use cases where applications are very static and the state provisioned in advance. In particular, a peer-to-peer (P2P) SCHC end point (see Section 4.4) may be set up between peers of equivalent capabilities, and the link direction cannot be inferred, either from the network topology nor from the device capability.

In that case, by convention, the device that initiates the connection that sustains the SCHC end point is considered as being Downlink, i.e. it plays the role of the Dev in [rfc8724].

This convention can be reversed, e.g., by configuration, but for proper SCHC operation, it is required that the method used ensures that both ends are aware of their role, and then again this determination is based on extrinsic properties.

4.6.1. SCHC Rules

SCHC Rules are a description of the header protocols fields, into a list of Field Descriptors. The [rfc8724] gives the format of the Rule description for C/D, F/R and non-compression. In the same manner the SCHC Stratum Header and SCHc CORECONF_Management will use the [rfc8724] field descriptors to compress the format information.

Each type of Rule is identified with a RuleID. There are different types of Rules: C/D, F/R, SCHC Stratum Header, CORECONF_Management and No Compression. Notice that each Rule type used an independent range of RuleID to identify its rules.

A Rule does not describe how the compressor parses a packet header. Rules only describe the behavior for each header field.

SCHC Action. ToDo

4.6.2. SoR identification

ToDo

4.7. SCHC Management

RFC9363 writes that only the management can be done by the two entities of the end point, and other SoR cannot be manipulated.

Management rules are explicitly define in the SoR, see Figure 4. They are compression Rules for CORECONF messages to get or modify the SoR of the end point. The management can be limited with the [I-D.ietf-schc-access-control] access definition.

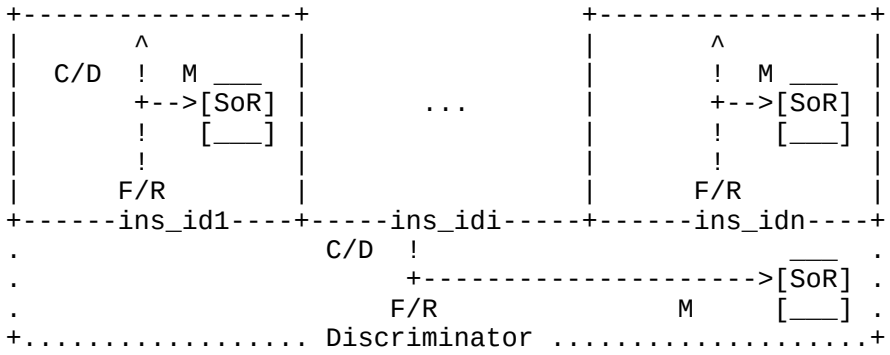


Figure 4: Inband Management

4.7.1. SCHC Instance Manager

The SCHC Instance Manager provides the management of SCHC end points, the SoR of each end point and the dialog between hosts to keep the SCHC synchronization, and the establishment of SCHC Instances with peer nodes. Changes that involve the SoR must be transactional, in a way that ensures that the compression and decompression of a packet is done with the same SoR on every end points.

The management of the SCHC end points includes the capability for the end-points to modify the common SoR, by:

* modifyng rules values (such as TV, MO or CDA) in existing rules,

* adding or

* removing rules.

The rule management uses the CORECONF interface based on CoAP. The management traffic is carried as SCHC compressed packets tagged to some specific rule IDs.

4.7.2. SCHC Data Model

A SCHC end point, summarized in the Figure 5, implies C/D and/or F/R and CORECONF_Management and SCHC end points Rules present in both end and that both ends are provisioned with the same SoR.

Figure 5: Summarized SCHC elements

A common rule representation that expresses the SCHC rules in an interoperable fashion is needed to be able to provision end-points from different vendors to that effect, [rfc9363] defines a rule representation using the YANG [rfc7950] formalism.

[rfc9363] defines a YANG data model to represent the rules. This enables the use of several protocols for rule management, such as NETCONF[RFC6241], RESTCONF[RFC8040], and CORECONF[I-D.ietf-core-comi]. NETCONF uses SSH, RESTCONF uses HTTPS, and CORECONF uses CoAP(s) as their respective transport layer protocols. The data is represented in XML under NETCONF, in JSON[RFC8259] under RESTCONF and in CBOR[RFC8949] under CORECONF.

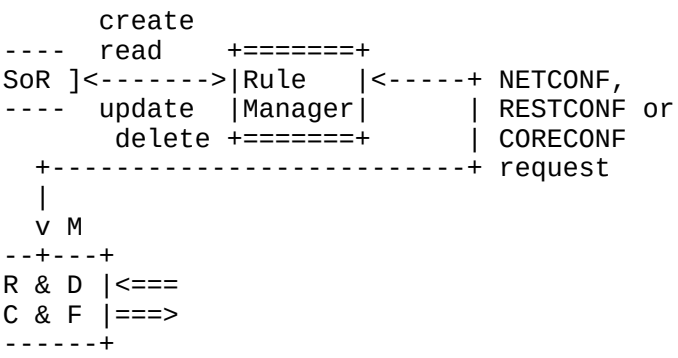


Figure 6: Summerized SCHC elements

The Rule Manager (RM) is in charge of handling data derived from the YANG Data Model and apply changes to the context and SoR of each SCHC end point Figure 6.

The RM is an Application using the Internet to exchange information, therefore:

* for the network-level SCHC, the communication does not require routing. Each of the end-points having an RM and both RMs can be viewed on the same link, therefore wellknown Link Local addresses can be used to identify the Device and the core RM. L2 security MAY be deemed as sufficient, if it provides the necessary level of protection.

* for application-level SCHC, routing is involved and global IP addresses SHOULD be used. End-to-end encryption is RECOMMENDED.

Management messages can also be carried in the negotiation protocol, for instace, the [I-D.ietf-schc-over-ppp] proposes a solution. The RM traffic may be itself compressed by SCHC: if CORECONF protocol is used, [rfc8824] can be applied.

5. SCHC Architecture

As described in [rfc8824], SCHC combining several SCHC end points. The [rfc8724] states that a SCHC end point needs the rules to process C/D and F/R before the SCHC Instance starts and that the SoR of the end point control layer cannot be modified. However, the rules may be updated in certain end points to improve the performance of C/D, F/R, or CORECONF_Management. The [I-D.ietf-schc-access-control] defines the possible modifications and who can modify, update, create and delete Rules or part of them in the end points' SoR.

As represented in Figure 7, the compression of the IP and UDP headers may be operated by a network SCHC end point whereas the end-to-end compression of the application payload happens between the Device and the application. The compression of the application payload may be split in two end points to deal with the encrypted portion of the application PDU. Fragmentation applies before LPWAN transmission layer.

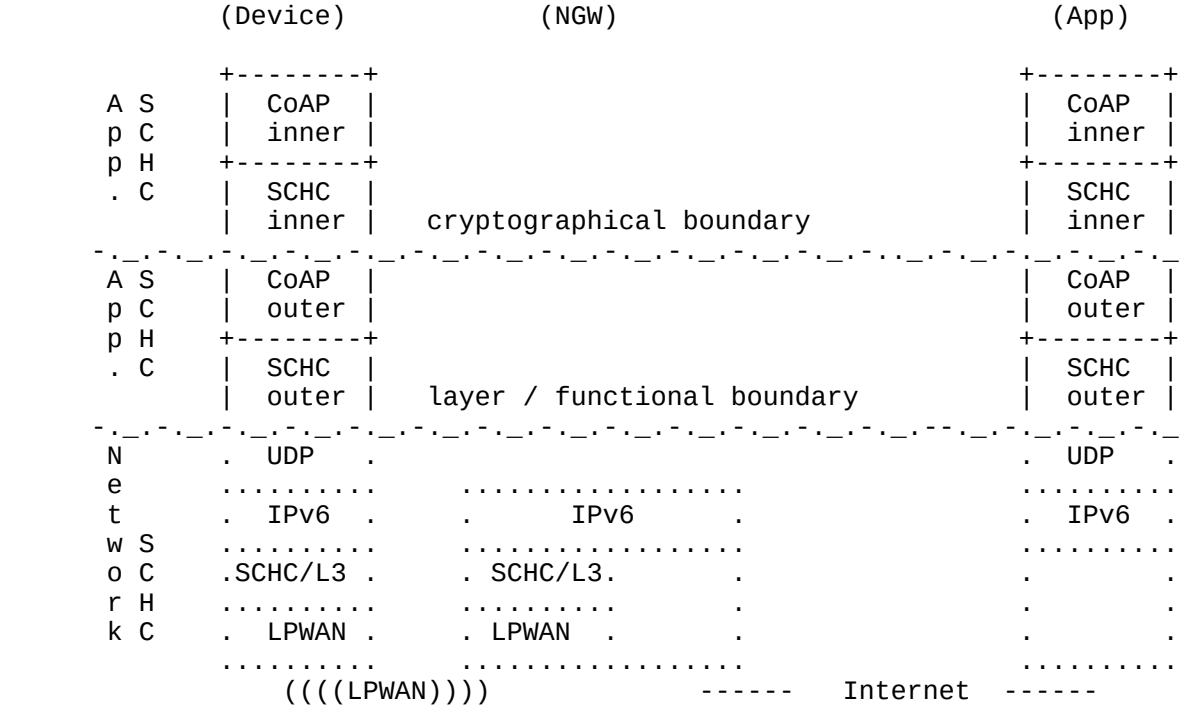
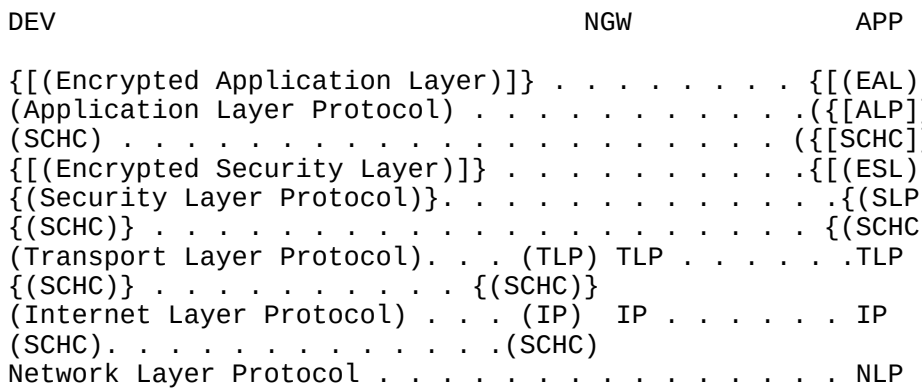


Figure 7: Different SCHC end points in a global system

This document defines a generic architecture for SCHC that can be used at any of these levels. The goal of the architectural document is to orchestrate the different protocols and data model defined by the LPWAN and SCHC working groups to design an operational and interoperable framework for allowing IP application over constrained networks.

The Figure 8 shows the protocol stack and the corresponding SCHC stratas enabling the compression of the different protocol headers. The SCHC Stratum Header eases the introduction of intermediary host in the end-to-end communication transparently. All the SCHC Stratum Headers are compressed and in some cases are elided, for example for LPWAN networks. The layers using encryption does not have a SCHC Stratum Header in the middle because they are the same entity. Figure 9 shows an example of an IP/UDP/CoAP in an LPWAN network.

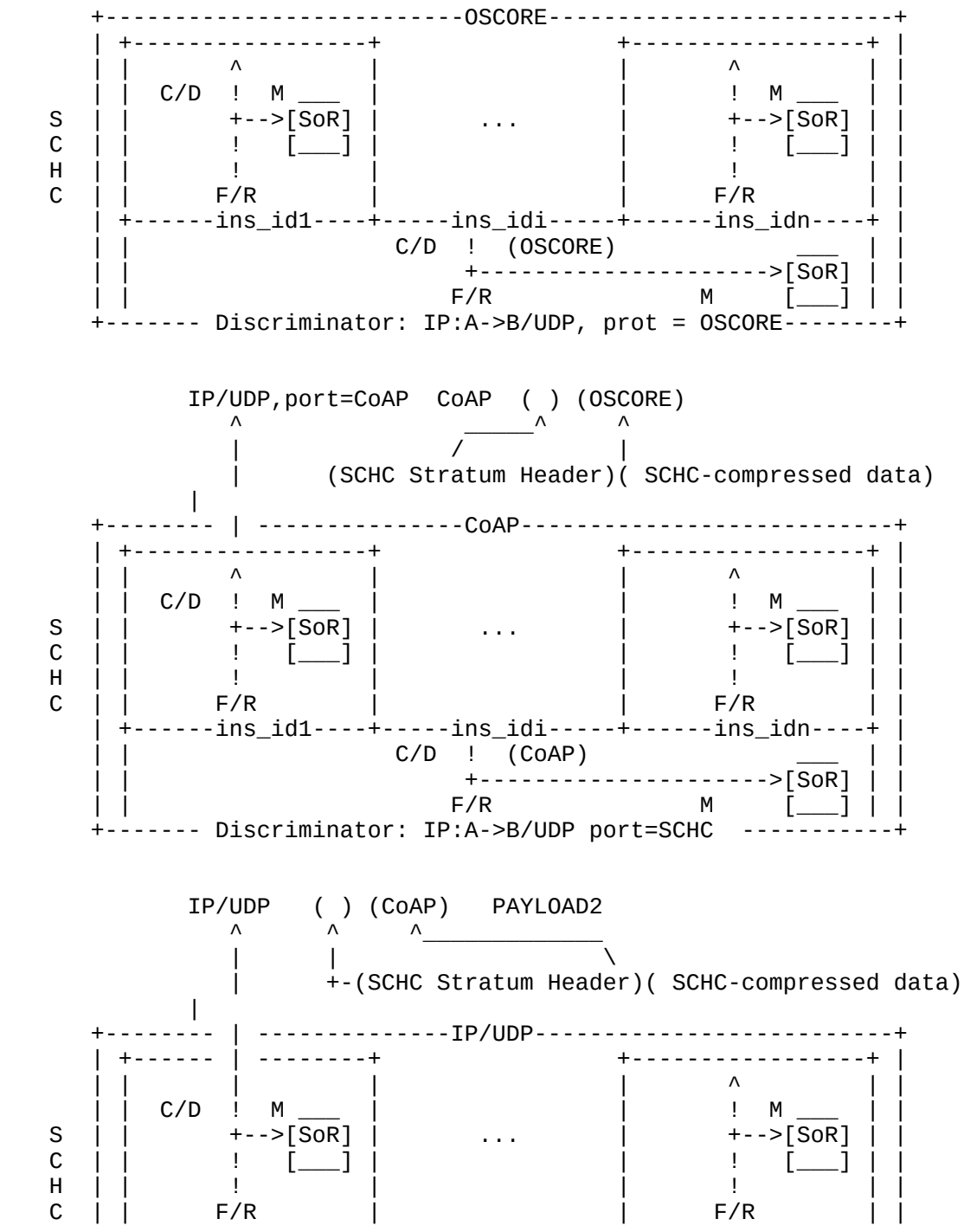


Where: {} Optional; [] Encrypted; () Compressed.

Figure 8: SCHC Architecture

In Figure 8, each line represents a layer or a stratum, parentheses surround a compressed header, and if it is optional, it has curly brackets. All the SCHC strata are compressed. Square brackets represent the encrypted data; if the encryption is optional, curly brackets precede the square brackets.

Figure 9 represents the stack of SCHC end points that operate over 3 strata, one for OSCORE, one for CoAP, and one for IP and UDP.


OSCORE
C/D
M
SoR
...
F/R
ins_id1
ins_idi
ins_idn
C/D
(OSCORE)
F/R
M
SoR
Discriminator: IP:A->B/UDP, prot = OSCORE
SCHC
IP/UDP,port=CoAP
CoAP
( ) (OSCORE)
(SCHC Stratum Header)(SCHC-compressed data)
CoAP
C/D
M
SoR
...
F/R
ins_id1
ins_idi
ins_idn
C/D
(CoAP)
F/R
M
SoR
Discriminator: IP:A->B/UDP port=SCHC
IP/UDP
( ) (CoAP)
PAYLOAD2
(SCHC Stratum Header)(SCHC-compressed data)
IP/UDP
C/D
M
SoR
...
F/R

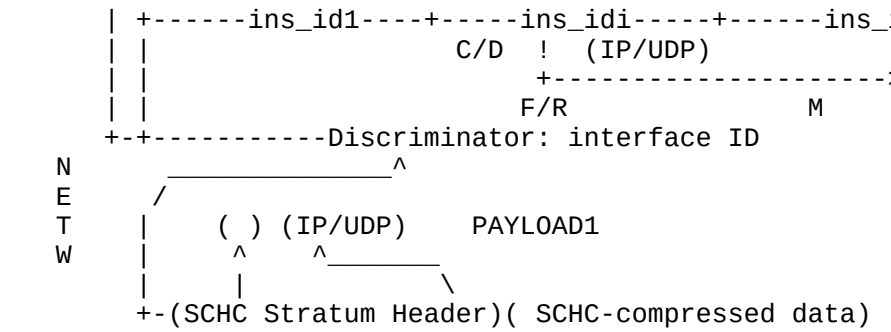


Figure 9: SCHC Strata Example

6. The Static Context Header Compression

SCHC [rfc8724] specifies an extreme compression capability based on a description that must match on the compressor and decompressor side. This description comprises a set of Compression/Decompression (C/D) rules.

The SCHC Parser analyzes incoming packets and creates a list of fields that it matches against the compression rules. The rule that matches is used to compress the packet, and the rule identifier (RuleID) is transmitted together with the compression residue to the decompressor. Based on the RuleID and the residue, the decompressor can rebuild the original packet and forward it in its uncompressed form over the Internet. When no Rule matches the header, the No Compression Rule is used. When several Rules match the header the implementation must choose one. How it is done or based on which parameters is out of the scope of this document. SCHC compresses datagrams and there is no notion of flows.

[rfc8724] also provides a Fragmentation/Reassembly (F/R) capability to cope with the maximum and/or variable frame size of a Link, which is extremely constrained in the case of an LPWAN network.

If a SCHC-compressed packet is too large to be sent in a single Link-Layer PDU, the SCHC fragmentation can be applied on the compressed packet. The process of SCHC fragmentation is similar to that of compression; the fragmentation rules that are programmed for this Device are checked to find the most appropriate one, regarding the SCHC packet size, the link error rate, and the reliability level required by the application.

The ruleID allows to determine if it is a compression or fragmentation rule or any other type of Rule.

6.1. SCHC over Network Technologies

SCHC can be used in multiple environments and multiple protocols. It was designed by default to work on native MAC frames with LPWAN technologies such as LoRaWAN[rfc9011], IEEE std 802.15.4 [I-D.ietf-6lo-schc-15dot4], and SigFox[rfc9442].

To operate SCHC over Ethernet, IPv6, and UDP, the definition of, respectively, an Ethertype, an IP Protocol Number, and a UDP Port Number are necessary, more in [I-D.ietf-intarea-schc-protocol-numbers]. In either case, there’s a need for a SCHC Stratum Header that is sufficient to identify the SCHC peers (endpoints) and their role (device vs. app), as well as the SCHC Instance between those peers that the packet pertains to.

In either of the above cases, the expectation is that the SCHC Stratum Header is transferred in a compressed form. This implies that the rules to uncompress the header are well known and separate from the rules that are used to uncompress the SCHC payload. The expectation is that for each stratum, the format of the SCHC Stratum Header and the compression rules are well known, with enough information to identify the SCHC Instance at that stratum, but there is no expectation that they are the same across strata.

6.1.1. SCHC over PPP

The LPWAN architecture (Figure 14) generalizes the model to any kind of peers. In the case of more capable devices, a SCHC Device may maintain more than one end point with the same peer, or a set of different peers. Since SCHC does not signal the end point in its packets, the information must be derived from a lower layer point to point information. For end point, the SCHC end point control can be associated one-to-one with a tunnel, a TLS SCHC Instance, or a TCP or a PPP connection.

For end point, [I-D.ietf-schc-over-ppp] describes a type of deployment where the C/D and/or F/R operations are performed between peers of equal capabilities over a PPP [rfc2516] connection. SCHC over PPP illustrates that with SCHC, the protocols that are compressed can be discovered dynamically and the rules can be fetched on-demand using CORECONF messages Rules, ensuring that the peers use the exact same set of rules.

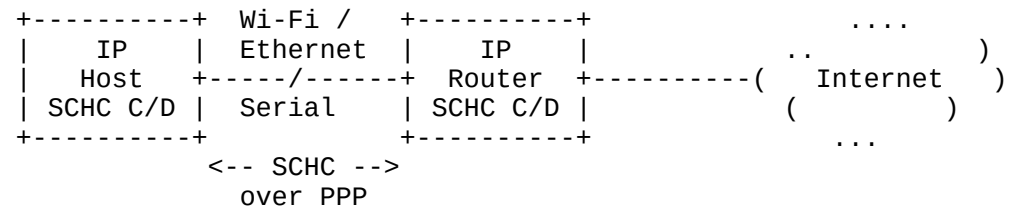


Figure 10: PPP-based SCHC Deployment

In that case, the SCHC end point is derived from the PPP connection. This means that there can be only one end point per PPP connection, and that all the flow and only the flow of that end point is exchanged within the PPP connection. As discussed in Section 7, the Uplink direction is from the node that initiated the PPP connection to the node that accepted it.

6.1.2. SCHC over Ethernet

Before the SCHC compression takes place, the SCHC Stratum Header showed in the Figure 11, is virtually inserted before the real protocol header and data that are compressed in the SCHC Instance, e.g. a IPv6 in this figure.

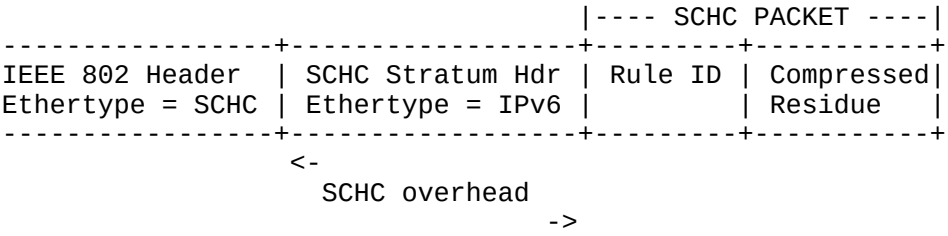


Figure 11: SCHC over Ethernet

6.1.3. SCHC over IPv6

In the case of IPv6, the expectation is that the Upper Layer Protocol (ULP) checksum can be elided in the SCHC compression of the ULP, because the SCHC Stratum Header may have its own checksum that protects both the SCHC Stratum Header and the whole ULP, header and payload.

The SCHC Stratum Header between IPv6 and the ULP is not needed because of the Next Header field on the IPv6 header format.

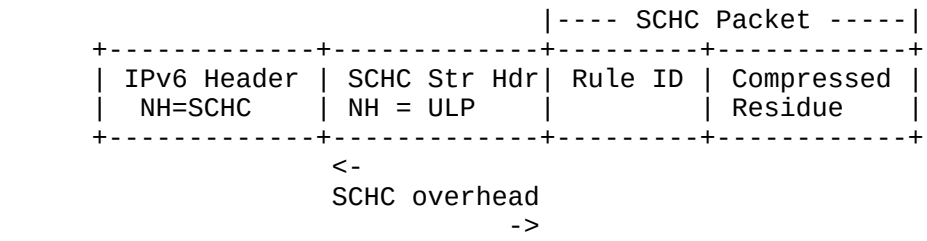
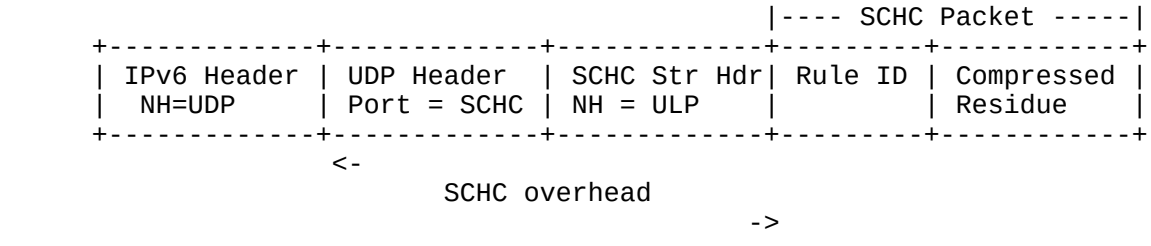


Figure 12: SCHC over IPv6

In the air, both the SCHC Stratum Header and the ULP are compressed. The SCHC Instance endpoints are typically identified by the source and destination IP addresses. If the roles are well-known, then the endpoint information can be elided and deduced from the IP header. If there is only one SCHC Instance, it can be elided as well, otherwise a rule and residue are needed to extract the SCHC Instance ID.

6.1.4. SCHC over UDP

When SCHC operates over the Internet, middleboxes may block packets with a next header that is SCHC. To avoid that issue, it would be desirable to prepend a UDP header before the SCHC Stratum Header as shown in figure Figure 13.



~

Figure 13: SCHC over UDP

In that case, the destination port can indicate SCHC as in an header chain, and the source port can indicate the SCHC Instance in which case it can be elided in the compressed form of the SCHC Stratum Header. The UDP checksum protects both the SCHC Stratum Header and the whole ULP, so the SCHC and the ULP checksums can both be elided. In other words, in the SCHC over UDP case, the SCHC Stratum Header can be fully elided, but the packet must carry the overhead of a full UDP header.

7. SCHC Endpoints for LPWAN Networks

Section 3 of [rfc8724] depicts a typical network architecture for an LPWAN network, simplified from that shown in [rfc8376] and reproduced in Figure 14.

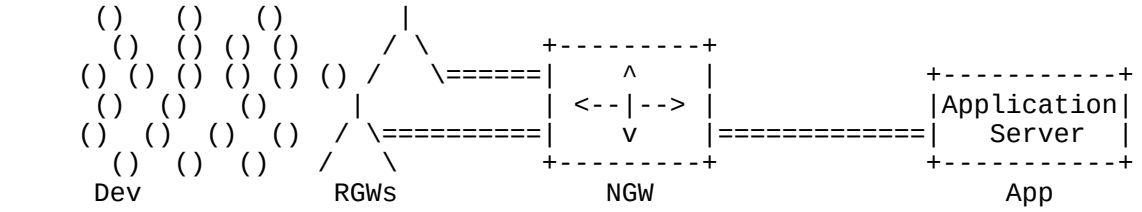


Figure 14: Typical LPWAN Network Architecture

Typically, an LPWAN network topology is star-oriented, which means that all packets between the same source-destination pair follow the same path from/to a central point. In that model, highly constrained Devices (Dev) exchange information with LPWAN Application Servers (App) through a central Network Gateway (NGW), which can be powered and is typically a lot less constrained than the Devices. Because Devices embed built-in applications, the traffic flows to be compressed are known in advance and the location of the C/D and F/R functions (e.g., at the Dev and NGW), and the associated rules, can be pre provisioned in the system before use.

7.1. SCHC Device Lifecycle

In the context of LPWANs, the expectation is that SCHC rules are associated with a physical device that is deployed in a network. This section describes the actions taken to enable an automatic commissioning of the device in the network.

7.1.1. Device Development

The expectation for the development cycle is that message formats are documented as a data model that is used to generate rules. Several models are possible:

1. In the application model, an interface definition language and binary communication protocol such as Apache Thrift is used, and the parser code includes the SCHC operation. This model imposes that both ends are compiled with the generated structures and linked with generated code that represents the rule operation.

2. In the device model, the rules are generated separately. Only the device-side code is linked with generated code. The Rules are published separately to be used by a generic SCHC engine that operates in a middle box such as a SCHC gateway.

3. In the protocol model, both endpoint generate a packet format that is imposed by a protocol. In that case, the protocol itself is the source to generate the Rules. Both ends of the SCHC compression are operated in middle boxes, and special attention must be taken to ensure that they operate on the compatible SoR, basically the same major version of the same SoR.

Depending on the deployment, the tools that generate the Rules should provide knobs to optimize the SoR, e.g., more rules vs. larger residue.

7.1.2. Rules Publication

In the device model and in the protocol model, at least one of the endpoints must obtain the SoR dynamically. The expectation is that the SoR are published to a reachable repository and versionned (minor, major). Each SoR should have its own Uniform Resource Names (URN) [RFC8141] and a version.

The SoR should be authenticated to ensure that it is genuine, or obtained from a trusted app store. A corrupted SoR may be used for multiple forms of attacks, more in Section 8.

7.1.3. SCHC Device Deployment

The device and the network should mutually authenticate themselves. The autonomic approach [RFC8993] provides a model to achieve this at scale with zero touch, in networks where enough bandwidth and compute are available. In highly constrained networks, one touch is usually necessary to program keys in the devices.

The initial handshake between the SCHC endpoints should comprise a capability exchange whereby URN and the version of the SoR are obtained or compared. SCHC may not be used if both ends can not agree on an URN and a major version.
Manufacturer Usage Descriptions (MUD) [RFC8520] may be used for that purpose in the device model.

Upon the handshake, both ends can agree on a SoR, their role when the rules are asymmetrical, and fetch the SoR if necessary. Optionally, a node that fetched a SoR may inform the other end that it is reacy from transmission.

7.1.4. SCHC Device Maintenance

URN update without device update (bug fix) FUOTA => new URN => reprovisioning

7.1.5. SCHC Device Decommissionning

Signal from device/vendor/network admin

8. Security Considerations

SCHC is sensitive to the rules that could be abused to form arbitrary long messages or as a form of attack against the C/D and/or F/R functions, say to generate a buffer overflow and either modify the Device or crash it. It is thus critical to ensure that the rules are distributed in a fashion that is protected against tempering, e.g., encrypted and signed.

9. IANA Consideration

This document has no request to IANA

10. Acknowledgements

The authors would like to thank (in alphabetic order): Laurent Toutain

11. References

11.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/rfc/rfc2119>.

[RFC8141] Saint-Andre, P. and J. Klensin, "Uniform Resource Names (URNs)", RFC 8141, DOI 10.17487/RFC8141, April 2017, <https://www.rfc-editor.org/rfc/rfc8141>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

[rfc8724] Minaburo, A., Toutain, L., Gomez, C., Barthel, D., and JC. Zuniga, "SCHC: Generic Framework for Static Context Header Compression and Fragmentation", RFC 8724, DOI 10.17487/RFC8724, April 2020, <https://www.rfc-editor.org/rfc/rfc8724>.

[rfc8824] Minaburo, A., Toutain, L., and R. Andreasen, "Static Context Header Compression (SCHC) for the Constrained Application Protocol (CoAP)", RFC 8824, DOI 10.17487/RFC8824, June 2021, <https://www.rfc-editor.org/rfc/rfc8824>.

11.2. Informative References

[I-D.ietf-6lo-schc-15dot4] Gomez, C. and A. Minaburo, "Transmission of SCHC-compressed packets over IEEE 802.15.4 networks", Work in Progress, Internet-Draft, draft-ietf-6lo-schc-15dot4-07, 2 October 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-6lo-schc-15dot4-07>.

[I-D.ietf-core-comi] Veillette, M., Van der Stok, P., Pelov, A., Bierman, A., and C. Bormann, "CoAP Management Interface (CORECONF)", Work in Progress, Internet-Draft, draft-ietf-core-comi-19, 3 November 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-core-comi-19>.

[I-D.ietf-intarea-schc-protocol-numbers] Moskowitz, R., Card, S. W., Wiethuechter, A., and P. Thubert, "Protocol Numbers for SCHC", Work in Progress, Internet-Draft, draft-ietf-intarea-schc-protocol-numbers-02, 8 April 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-intarea-schc-protocol-numbers-02>.

[I-D.ietf-schc-access-control] Minaburo, A., Toutain, L., and I. Martinez, "SCHC Access Control", Work in Progress, Internet-Draft, draft-ietf-schc-access-control-00, 13 December 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-schc-access-control-00>.

[I-D.ietf-schc-over-ppp] Thubert, P., "SCHC over PPP", Work in Progress, Internet-Draft, draft-ietf-schc-over-ppp-00, 25 July 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-schc-over-ppp-00>.

[rfc2516] Mamakos, L., Lidl, K., Evarts, J., Carrel, D., Simone, D., and R. Wheeler, "A Method for Transmitting PPP Over Ethernet (PPPoE)", RFC 2516, DOI 10.17487/RFC2516, February 1999, <https://www.rfc-editor.org/rfc/rfc2516>.

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <https://www.rfc-editor.org/rfc/rfc6241>.

[rfc7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <https://www.rfc-editor.org/rfc/rfc7950>.

[RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <https://www.rfc-editor.org/rfc/rfc8040>.

[RFC8259] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", STD 90, RFC 8259, DOI 10.17487/RFC8259, December 2017, <https://www.rfc-editor.org/rfc/rfc8259>.

[rfc8376] Farrell, S., Ed., "Low-Power Wide Area Network (LPWAN) Overview", RFC 8376, DOI 10.17487/RFC8376, May 2018, <https://www.rfc-editor.org/rfc/rfc8376>.

[RFC8520] Lear, E., Droms, R., and D. Romascanu, "Manufacturer Usage Description Specification", RFC 8520, DOI 10.17487/RFC8520, March 2019, <https://www.rfc-editor.org/rfc/rfc8520>.

[RFC8949] Bormann, C. and P. Hoffman, "Concise Binary Object Representation (CBOR)", STD 94, RFC 8949, DOI 10.17487/RFC8949, December 2020, <https://www.rfc-editor.org/rfc/rfc8949>.

[RFC8993] Behringer, M., Ed., Carpenter, B., Eckert, T., Ciavaglia, L., and J. Nobre, "A Reference Model for Autonomic Networking", RFC 8993, DOI 10.17487/RFC8993, May 2021, <https://www.rfc-editor.org/rfc/rfc8993>.

[rfc9011] Gimenez, O., Ed. and I. Petrov, Ed., "Static Context Header Compression and Fragmentation (SCHC) over LoRaWAN", RFC 9011, DOI 10.17487/RFC9011, April 2021, <https://www.rfc-editor.org/rfc/rfc9011>.

[rfc9363] Minaburo, A. and L. Toutain, "A YANG Data Model for Static Context Header Compression (SCHC)", RFC 9363, DOI 10.17487/RFC9363, March 2023, <https://www.rfc-editor.org/rfc/rfc9363>.

[rfc9442] Z炭单iga, J., Gomez, C., Aguilar, S., Toutain, L., C辿spedes, S., Wistuba, D., and J. Boite, "Static Context Header Compression (SCHC) over Sigfox Low-Power Wide Area Network (LPWAN)", RFC 9442, DOI 10.17487/RFC9442, July 2023, <https://www.rfc-editor.org/rfc/rfc9442>.

Authors' Addresses

Alexander Pelov
IMT Atlantique
rue de la Chataigneraie
35576 Cesson-Sevigne Cedex
France
Email: alexander.pelov@imt-atlantique.fr

Pascal Thubert
06330 Roquefort les Pins
France
Email: pascal.thubert@gmail.com

Ana Minaburo
Consultant
35510 Cesson-Sevigne Cedex
France
Email: anaminaburo@gmail.com