Workgroup: lpwan Working Group
Internet-Draft: draft-barthel-schc-oam-schc-00
Published: 19 January 2024
Intended Status: Informational
Expires: 22 July 2024
Authors: D. Barthel L. Toutain
IMT Atlantique

OAM for LPWAN using Static Context Header Compression (SCHC)

Abstract

This document describes how SCHC can be used to efficiently perform basic Operation, Administration and Maintenance (OAM) on Low Power Wide Area Networks (LPWANs) by compressing ICMPv6/IPv6 headers, or by shielding the LPWAN network and the Device from undesirable ICMPv6 traffic.

This document specifies additional behavior for SCHC [RFC8724] and extends the YANG Data Model defined in [RFC9363].

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 22 July 2024.

Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction

With IP protocols now generalizing to constrained networks, users expect to be able to Operate, Administer and Maintain (OAM) [RFC6291] such networks with the familiar tools and protocols they already use on less constrained networks.

However, this requires a little care, because OAM traffic adds load to the network, and LPWANs could easily be overwhelmed by it. LPWANs' salient characteristics are described in [RFC8376].

This document specifies ways to compress such OAM traffic over LPWANs, or to eschew it altogether.

OAM uses specific messages sent into the data plane to probe a network. Most often, these messages do not carry meaningful data. Instead, network metrics are inferred from analysing the OAM traffic.

For example, such traffic is used

- *to detect if a host is up or down,

*to measure the Round-Trip Time (RTT) and its variation over time,

*to determine avalaible bandwidth, or

*to learn the path used by packets to reach a destination.

The primitive functionalities of OAM are achieved with the ICMPv6 protocol [RFC4443]. ICMPv6 messages are transported over IPv6 [RFC8200].

ICMPv6 defines a generic message format, used to inform the source of IPv6 packets of errors during packet delivery..

[RFC4443] instantiates 4 such error messages: Destination Unreachable, Packet Too Big, Time Exceeded and Parameter Problem.

[RFC4443] also defines the Echo Request and Echo Reply messages, which provide support for the ping application.

Other ICMPv6 messages are defined in other RFCs, such as an extended format of the same messages [RFC4884] and other messages used by the Neighbor Discovery Protocol [RFC4861].

This document focuses on using Static Context Header Compression (SCHC) to compress [RFC4443] messages that need to be transmitted over the LPWAN network, and on having the LPWAN gateway proxying the Device to save it the unwanted traffic. More specifically, this document describes recommended compression of ICMPv6/IPv6 messages (including header fields and structured payload) and extends SCHC by specifying new surrogate behavior, addressing four scenarios:

*OAM reachability messages coming from the internet: the core SCHC acts as a proxy and may decide to respond by itself, thereby acting as a surrogate to the Device.

*OAM messages initiated by LPWAN Devices: they can be anticipated and sent in their SCHC-compressed form like regular Device traffic.

*OAM error messages returned from the internet after an LPWAN Device transmission. The core SCHC forwards a compressed version of the error message to the Device.

*traffic coming from the internet that would generate an error at the Device: if it can detect the situation, the core SCHC responds with an ICMPv6 error message, acting as a surrogate to the Device.

2. Terminology

This draft re-uses the Terminology defined in [RFC8724].

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. Use cases

In the LPWAN architecture, we can distinguish the following cases:

* The Device is the originator of an Echo Request message, and therefore the destination of the Echo Reply message. These messages are compressed/decompressed by the device and by the core SCHC using SCHC rules that match the ICMPv6 fields.

* The Device is the destination of an Echo Request message, and therefore the purported source of an Echo Reply message. The core SCHC can either forward the SCHC-compressed Echo Request message to the Device, or proxy the Device by answering with an Echo Reply message on its behalf, in order to spare the constrained link. The proxy answer can be related to the Device observed activity.

* The Device is the (purported) source of an ICMP error message, mainly in response to an incorrect incoming IPv6 message. In this case, as much as possible, the core SCHC should act as a proxy and originate the ICMP Destination Unreachable message, so that the Device and the LPWAN network are protected from this unwanted traffic.

* the Device is the destination of the ICMPv6 message, mainly in response to a packet sent by the Device to the network that generates an error. In this case, we want the ICMPv6 message to reach the Device, and this document describes in Section 4.5.1 what SCHC compression should be applied.

These cases are further described in Section 4.

4. Detailed behavior

4.1. ICMPv6 compression

This section defines ICMPv6 fields that can be compressed by SCHC. {{RFC4443}} defines several formats regarding the type of the ICMPv6 message.

From them, several fields can be extracted. Note that names listed here are just informative and lisibility, the Field ID identifiers are specified int augmentation of the YANG Data Model (cf. Section 7):

These fields are present in all the messages:

* ICMPv6 Type indicates the fields present in the message.

* ICMPv6 Code is related to the ICMPv6 type and has not impact on the message format.

* ICMPv6 Checksum covers the ICMPv6 message and part of the IPv6 header to protect against errors.

* ICMPv6 Payload contains either part of the message at the orgine of the error or some data in the case of ping.

The other fields depends of the message type:

* ICMPv6 MTU is used by Packet Too Big message (type = 2) to carry the MTU expected by a node rejecting the packet forwarding

* ICMPv6 Pointer is used by Parameter Problem message to indicate the position of a detected error in the original message

* ICMPv6 Identifier and ICMPv6 Sequence Number are used by ping echo (type 128) and reply (type 129) messages.

ICMPv6 is the support for several protocols to configure nodes. These protocols defines new types and may add optionnal information after the ICMPv6 header

4.2. Device does a ping

A Device may send an Echo Request message to check the availability of the network and of the host running the Application.

If a ping request is generated by a Device, then SCHC compression applies.

The format of an ICMPv6 Echo Request message is described in Figure 1, with Type=128 and Code=0.

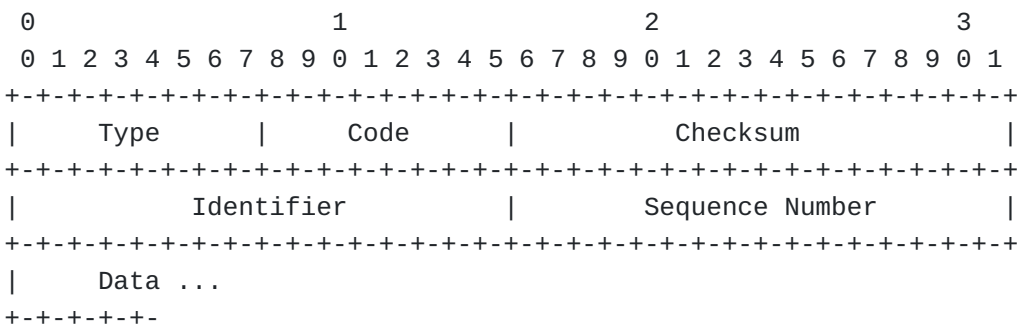

Figure 1: ICMPv6 Echo Request message format

If we assume that one rule will be devoted to compressing Echo Request messages, then Type and Code are known in the rule to be 128 and 0 and can therefore be elided with the not-sent CDA.

Checksum can be reconstructed with the compute-checksum CDA and therefore is not transmitted.

[RFC4443] states that Identifier and Sequence Number are meant to “aid in matching Echo Replies to this Echo Request” and that they “may be zero”. Data is “zero or more bytes of arbitrary data”.

For constrained devices or networks, we recommend that Identifier be zero, Sequence Number be a counter on 3 bits, and Data be zero bytes (absent). Therefore, Identifier is elided with the not-sent CDA, Sequence Number is transmitted on 3 bits with the LSB CDA and no Data is transmitted.

The transmission cost of the Echo Request message is therefore the size of the Rule Id + 3 bits. The rule ID length can be chosen to avoid adding padding.

When the destination receives the Echo Request message, it will respond back with a Echo Reply message. This message bears the same format as the Echo Request message but with Type = 129 (see Figure 1).

[RFC4443] states that the Identifier, Sequence Number and Data fields of the Echo Reply message shall contain the same values as the invoking Echo Request message. Therefore, a rule shall be used similar to that used for compressing the Echo Request message.

4.2.1. Rule example

The following rule gives an example of a SCHC compression. The type can be elided if the direction is taken into account. Identifier is ignored and generated as 0 at decompression. This implies that only one single ping can be launched at any given time on a device.

Finally, only the least significant 8 bits of the sequence number are sent on the LPWAN, allowing a serie of 255 consecutive pings.

Field	FL	FP	DI	Value	Matching Operator	CDA		Sent bits
IPv6 Headers description								
ICMPv6 Type	8	1	Up	128	equal	not-sent		
ICMPv6 Type	8	1	Dw	129	equal	not-sent		
ICMPv6 Code	8	1	Bi	0	equal	not-sent		
ICMPv6 Identifier	16	1	Bi	0	ignore	not-sent		
ICMPv6 Sequence	16	1	Bi	0	MSB(13)	LSB		3

Table 1: Example of compression rule for a ping from the device

NOTE: Add an example where the Payload is also compressed.

4.3. Device is ping'ed

If the Device is ping’ed (i.e., is the destination of an Echo Request message), the device receives the compress message and generate an Echo. In that case, the fields sequence number and identifier cannot be compressed if the source is not aware of the compression scheme.

But the default behavior is to avoid propagating the Echo Request message over the LPWAN.

This is done by proxying the ping request on the core SCHC. This requires to introduce a new processing when the rule is selected. The selection of a compression rule triggers the compression and sends the SCHC packet to the other end. Specifying an Action, change this behavior. In our case, being processed by the compressor, the packet description is processed by a ping proxy. Since the rule is used for the selection, so CDAs are not necessary and set to "not-sent".

The ping-proxy takes a parameter in second, gives the interval during which the device is considered active. During this interval, the proxy-ping echoes ping requests, after this duration, the ping request will be discarded.

The resulting behavior is shown on Figure 2 and described below:

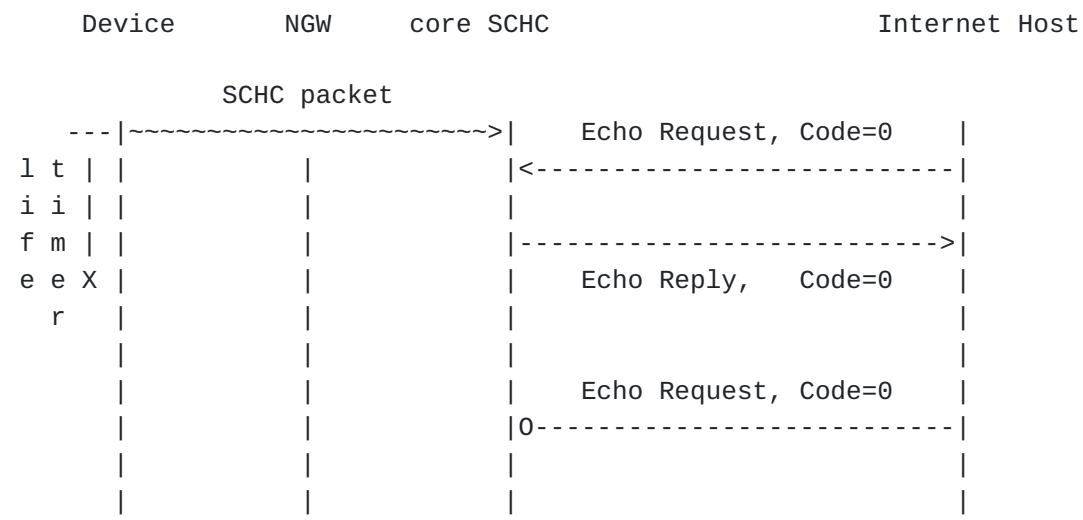

Figure 2: Examples of ICMPv6 Echo Request/Reply

NOTE: Do we add a proxy-ping-or-sent with instead a not answering send the compress packet to the device ?

4.3.1. Rule example

The following rule shows an example of a compression rule for pinging a device.

Field	**FL**	**FP**	**DI**	**Value**	**Matching Operator**	**CDA**		**Sent bits**
Action: proxy-ping(300)								
IPv6 Headers description								
ICMPv6 Type	8	1	Dw	128	equal	not-sent		
ICMPv6 Code	8	1	Bi	0	equal	not-sent		
ICMPv6 Identifier	16	1	Bi	0	ignore	not-sent		
ICMPv6 Sequence	16	1	Bi	0	MSB(13)	LSB		3

Table 2: Example of compression rule for a ping to a device

In this example, type and code are elided, the identifer has to be sent, and the sequence number is limited to one byte.

4.4. Device is the source of an ICMPv6 error message

As stated in [RFC4443], a node should generate an ICMPv6 message in response to an IPv6 packet that is malformed or which cannot be processed due to some incorrect field value.

The general intent of this document is to spare both the Device and the LPWAN network this un-necessary traffic. The incorrect packets should be caught at the core SCHC and the ICMPv6 notification should be sent back from there.

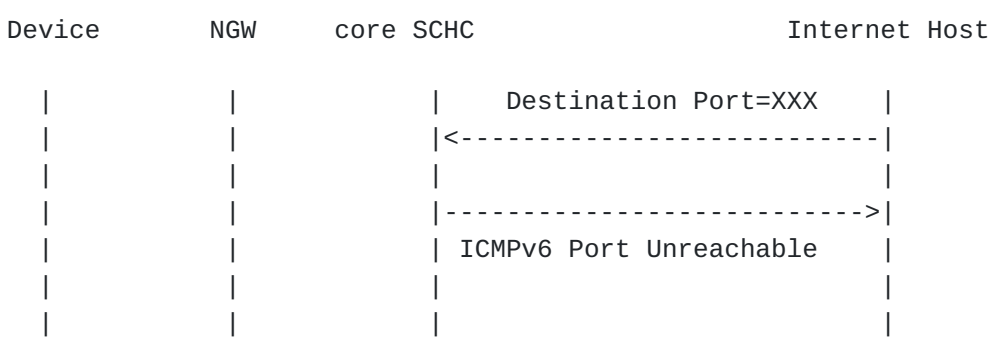

Figure 3: Example of ICMPv6 error message sent back to the Internet

Figure 3 shows an example of an IPv6 packet trying to reach a Device.

Let's assume that no rule matches the incoming packet (i.e. there is no co-compression rule)

Instead of sending the packet over the LPWAN and having this packet rejected by the Device, the core SCHC issues an ICMPv6 error message “Destination Unreachable” (Type 1) with Code 1 (“Port Unreachable”) on behalf of the Device.

In that case the SCHC C/D MAY act as a router (i.e. it MUST have a routable IPv6 address to generate an ICMPv6 message). When compressing a packet containing an IPv6 header, no compression rules are found and:

* if a rule contains some extension headers, a parameter problem may be generated (type 4),

* no rule contains the IPv6 device address found in the incoming packet, a no route to destination ICMPv6 message (type 0, code 3) may be generated,

* a device IPv6 address is found, but no port matches, a port unreachable ICMPv6 message (type 0, code 4) may be generated,

* if the incoming packet is too large for any of the fragmentation rules, an ICMPv6 Message Too big MAY be generated with the largest size allowaed by the fragmentation rules.

4.5. Device is the destination of an ICMPv6 error message

In this situation, we assume that a Device has been configured to send information to a server on the Internet. If this server becomes no longer accessible, an ICMPv6 message will be generated back towards the Device by either an intermediate router or the destination. This information can be useful to the Device, for example for reducing the reporting rate in case of periodic reporting of data. Therefore, we compress the ICMPv6 message using SCHC and forward it to the Device over the LPWAN. We also introduce new MO and CDA that can be used to test the presence and/or compress the returning payload.

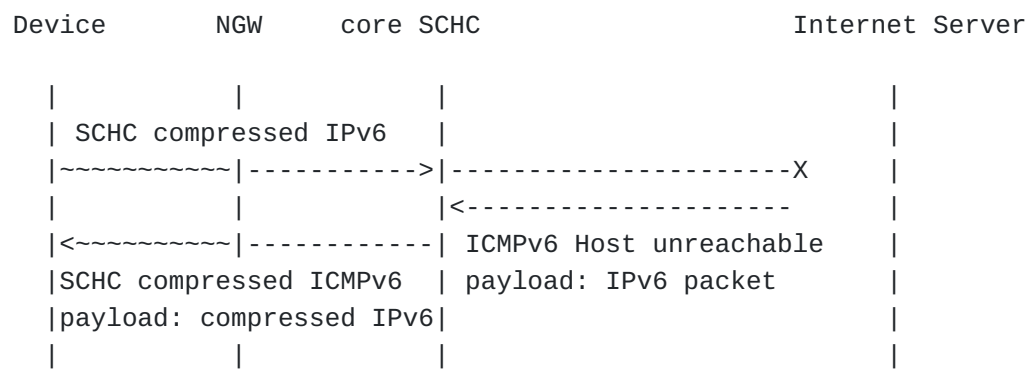

Figure 4: Example of ICMPv6 error message sent back to the Device

Figure 4 illustrates this behavior. The ICMPv6 error message is compressed as described in Section 4.5.1 and forwarded over the LPWAN to the Device.

The SCHC returning message contains the SCHC residue of the ICMPv6 message and MAY contain the compressed original message contained in the ICMP message. The compression can be done by the core SCHC by reversing the direction as if this message was issued by the device.

4.5.1. ICMPv6 error message compression.

The ICMPv6 error messages defined in [RFC4443] contain the fields shown in Figure 5.

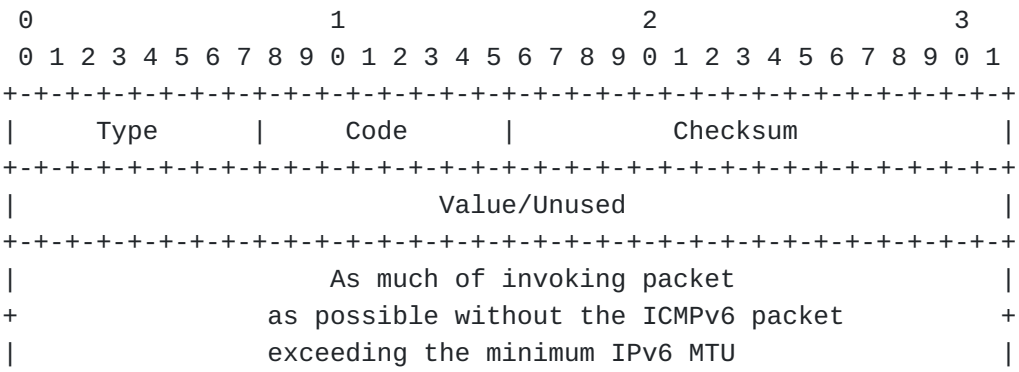

Figure 5: ICMPv6 Error Message format

[RFC4443] states that Type can take the values 1 to 4, and Code can be set to values between 0 and 6. Value is unused for the Destination Unreachable and Time Exceeded messages. It contains the MTU for the Packet Too Big message and a pointer to the byte causing the error for the Parameter Error message. Therefore, Value is never expected to be greater than 1280 in LPWAN networks.

The payload is viewed as a field. An unsued field MUST not appear in the compressoin rules.

The source address of the message SHOULD be "ignore", since it can be initiated by any router on the path.

The following generic rule can therefore be used to compress all ICMPv6 error messages as defined today. More specific rules can also be defined to achieve better compression of some error messages.

The Type field can be associated to a matching list [1, 2, 3, 4] and is therefore compressed down to 2 bits. Code can be reduced to 3 bits using the LSB CDA. Value can be sent on 11 bits using the LSB CDA, but if the Device is known to send smaller packets, then the size of this field can be further reduced.

The first rule example Table 3 just sends the ICMP type and code as residue to the device.

Field	**FL**	**FP**	**DI**	**Value**	**Matching Operator**	**CDA**		**Sent bits**
IPv6 Headers description								
ICMPv6 Type	8	1	Dw	128	equal	not-sent		
ICMPv6 Code	8	1	Dw	[0,1,2,3,4,5,6]	match-mapping	mapping-sent		3
	var	1	Dw	0	ignore	not-sent		

Field	FL	FP	DI	Value	Matching Operator	CDA		Sent bits
ICMPv6 Payload								

Table 3: Example of compression rule for a ICMP error to a device

The second rule example Table 4 also only sends the ICMP type and code as residue to the device, but it introduces the new MO "rev-rule-match". This MO will check if a rule matches the payload.

Field	FL	FP	DI	Value	Matching Operator	CDA		Sent bits
IPv6 Headers description								
ICMPv6 Type	8	1	Dw	128	equal	not-sent		
ICMPv6 Code	8	1	Dw	[0,1,2,3,4,5,6]	match-mapping	mapping-sent		
ICMPv6 Payload	var	1	Dw	0	rev-rule-match	not-sent		

Table 4: Example of compression rule for a ICMP error to a device

By [RFC4443], the rest of the ICMPv6 message must contain as much as possible of the IPv6 offending (invoking) packet that triggered this ICMPv6 error message. This information is used to try and identify the SCHC rule that was used to decompress the offending IPv6 packet. If the rule can be found then the Rule Id is added at the end of the compressed ICMPv6 message. Otherwise the compressed packet ends with the compressed Value field.

The third rule example Table 5 also sends the ICMP type, code and the compresssed payload as residue. It can be noted that this field is identified as "variable" in the rule which will introduce a size before the IPv6 compressed header.

Field	FL	FP	DI	Value	Matching Operator	CDA		Sent bits
IPv6 Headers description								
ICMPv6 Type	8	1	Dw	128	equal	not-sent		
ICMPv6 Code	8	1	Dw	[0,1,2,3,4,5,6]	match-mapping	mapping-sent		
ICMPv6 Payload	var	1	Dw	0	rev-rule-match	rev-compress-sent		(compressed IPv6 header*9) + 4 or +12

Table 5: Example of compression rule for a ICMP error to a device

LT: do we add packet too big, for instance if a fragmentation rule cannot handle a size larger than 1280?

5. Rule Action

The Action is a new attribute in the rule. When a rule matching the packet is selected, the action is applied first and indicates if the regular compression based on CDA should be applied.

6. YANG identities and tree

Figure 6 shows the augmentation of the Data Model defined in [RFC9363]

This YANG module extends Field ID identities to includes fields contained in ICMPv6 header. Note that the ICMPv6 payload is parsed to the specific field "fid-icmpv6-payload"

It also defines two new Most identities:

- *mo-rev-rule-match: The value contained in the Field Value matches a rule. The direction used for matching isthe opposite of the incoming message: UP becomes DOWN and DOWN becomes UP. This MO can be used to test if the Payload contained in the ICMPv6 message matches a rule. This means that the original packet, at the origine of the ICMPv6 message, may have been generated from the SCHC decompression.

- *mo-rule-match: The value contained in the Target Value matches a rule. The direction is the one of the incoming message. This MO is not used for ICMPv6 messages, but since it can be used in other situations, it has been included in the Data Model.

The Field Value may be compressed by a rule. The result SHOULD be included in the SCHC message as a variable length residue. It contains the Rule ID used by the compression, the residue, the payload and some padding bits since the variable length init is in bytes.

- *cda-rev-compress-sent: The direction used for compression is the opposite of the incoming message: UP becomes DOWN and DOWN becomes UP.

- *cda-compress-sent: The direction used for compression is the same as for the incoming message.

```
module: ietf-schc-oam

  augment /schc:schc/schc:rule/schc:nature/schc:compression:
    +--rw proxy-behavior?         schc-oam:proxy-type
    +--rw proxy-behavior-value* [index]
       +--rw index    uint16
       +--rw value?   binary
```

Figure 6: YANG tree

7. YANG Module

```
module ietf-schc-oam {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-schc-oam";
  prefix schc-oam;

  import ietf-schc {
      prefix schc;
  }

  organization
    "IETF IPv6 over Low Power Wide-Area Networks (lpwan) working group";
  contact
    "WG Web:   <https://datatracker.ietf.org/wg/lpwan/about/>
     WG List:  <mailto:p-wan@ietf.org>
     Editor:   Laurent Toutain
       <mailto:laurent.toutain@imt-atlantique.fr>
     Editor:   Ana Minaburo
       <mailto:ana@ackl.io>";
  description
     "
     Copyright (c) 2021 IETF Trust and the persons identified as
     authors of the code.  All rights reserved.

     Redistribution and use in source and binary forms, with or
     without modification, is permitted pursuant to, and subject to
     the license terms contained in, the Simplified BSD License set
     forth in Section 4.c of the IETF Trust's Legal Provisions
     Relating to IETF Documents
     (https://trustee.ietf.org/license-info).

     This version of this YANG module is part of RFC XXXX
     (https://www.rfc-editor.org/info/rfcXXXX); see the RFC itself
     for full legal notices.

     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 (RFC 2119) (RFC 8174) when, and only when,
     they appear in all capitals, as shown here.

     ****************************************************************

     This module extends the ietf-schc module to include the compound-ac
     behavior for Ack On Error as defined in RFC YYYY.
     It introduces a new leaf for Ack on Error defining the format of th
     SCHC Ack and add the possibility to send several bitmaps in a singl
     answer.";

  revision 2024-01-19 {
    description
```

```
      "Initial version for RFC YYYY ";
    reference
      "RFC YYYY: OAM";
  }

  identity fid-icmpv6-base-type {
    base schc:fid-base-type;
    description
      "Field IP base type for ICMPv6 headers described in RFC 4443";
    reference
      "RFC 4443   Internet Control Message Protocol (ICMPv6)
                  for the Internet Protocol Version 6 (IPv6) Specificati
  }

// ICMPv6 Fields

  identity fid-icmpv6-type {
    base schc:fid-icmpv6-base-type;
    description
      "ICMPv6 code field present in all ICMPv6 messages.";
  }

  identity fid-icmpv6-code {
    base schc:fid-icmpv6-base-type;
    description
      "ICMPv6 code field present in all ICMPv6 messages.";
  }

  identity fid-icmpv6-checksum {
    base schc:fid-icmpv6-base-type;
    description
      "ICMPv6 checksum field present in all ICMPv6 messages.";
  }

    identity fid-icmpv6-mtu {
    base schc:fid-icmpv6-base-type;
    description
      "ICMPv6 MTU, present in Packet Too Big message.";
  }

  identity fid-icmpv6-pointer {
    base schc:fid-icmpv6-base-type;
    description
      "ICMPv6 Pointer, present in Parameter Problem message.";
  }

  identity fid-icmpv6-identifier {
    base schc:fid-icmpv6-base-type;
    description
      "ICMPv6 identifier field, present in Echo Request/Reply message.";
```

```
  }

  identity fid-icmpv6-sequence {
    base schc:fid-icmpv6-base-type;
    description
      "ICMPv6 sequence number field, present in Echo Request/Reply messa
  }

  identity fid-icmpv6-payload {
    base schc:fid-icmpv6-base-type;
    description
      "ICMPv6 payload following ICMPv6 header.
      If payload is empty, this field exists with a length of 0.";
  }

// MO and CDA

  identity mo-rule-match {
    base schc:mo-base-type;
    description
        "Macthing operator return true, if the TV matches a rule
        keeping UP and DOWN direction." ;
  }

  identity mo-rev-rule-match {
    base schc:mo-base-type;
    description
        "Macthing operator return true, if the TV matches a rule
        reversing UP and DOWN direction." ;
  }


  identity cda-compress-sent {
    base schc:mo-base-type;
    description
        "Send a compressed version of TV keeping UP and
        DOWN direction." ;
  }

  identity  cda-rev-compress-sent {
    base schc:mo-base-type;
    description
        "Send a compressed version of TV reversing UP and
        DOWN direction." ;
  }

// Proxy actions

  identity  proxy-schc-message{
    description
```

```
      "Define how the message is proxied after compression.";
  }

  identity proxy-none {
    base proxy-schc-message;
    description
      "The message is not proxied and sent to L2,
      default behavior of RFC 8724.";
  }

  identity proxy-pingv6 {
    base proxy-schc-message;
    description
      "The message is processed by an ping6 proxy.";
  }

  typedef proxy-type {
    type identityref {
      base proxy-schc-message;
    }
    description
      "The type used in rules to define an action.";
  }

// SCHC rule

  augment "/schc:schc/schc:rule/schc:nature/schc:compression" {
    leaf proxy-behavior {
        type schc-oam:proxy-type;
        default "schc-oam:proxy-none";
        description
              "Entity proxying the SCHC message.";
    }
    list proxy-behavior-value {
        key "index";
        uses schc:tv-struct;
        description
              "Parameters associated to the proxy action.";
    }
    description
      "Leaves added to SCHC rules for proxy.";
  }


}
```

Figure 7: YANG module

8. Security considerations

flood the return path with ICMP error messages.

9. IANA Considerations

TODO

10. Contributors

The following people have been co-authors of precursor versions of this draft. Their contribution is deeply appreciated and acknowledged.

*Arunprabhu Kandasamy (Acklio)

*Diego Dujovne (Universidad Diego Portales)

*Juan Carlos Zuniga (Cisco)

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/info/rfc2119>.

[RFC4443] Conta, A., Deering, S., and M. Gupta, Ed., "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", STD 89, RFC 4443, DOI 10.17487/RFC4443, March 2006, <https://www.rfc-editor.org/info/rfc4443>.

[RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, DOI 10.17487/RFC4861, September 2007, <https://www.rfc-editor.org/info/rfc4861>.

[RFC4884] Bonica, R., Gan, D., Tappan, D., and C. Pignataro, "Extended ICMP to Support Multi-Part Messages", RFC 4884, DOI 10.17487/RFC4884, April 2007, <https://www.rfc-editor.org/info/rfc4884>.

[RFC6291] Andersson, L., van Helvoort, H., Bonica, R., Romascanu, D., and S. Mansfield, "Guidelines for the Use of the "OAM" Acronym in the IETF", BCP 161, RFC 6291, DOI 10.17487/RFC6291, June 2011, <https://www.rfc-editor.org/info/rfc6291>.

[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/info/rfc8174>.

[RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <https://www.rfc-editor.org/info/rfc8200>.

[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/info/rfc8724>.

[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/info/rfc9363>.

11.2. Informative References

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

Authors' Addresses

Dominique Barthel
France

Email: dominique.barthel@orange.com

Laurent Toutain
IMT Atlantique
2 rue de la Chataigneraie
CS 17607
35576 Cesson-Sevigne Cedex
France

Email: laurent.toutain@imt-atlantique.fr