schc Working Group
Internet-Draft
Intended status: Informational
Expires: 15 December 2025

D. Barthel

L. Toutain
IMT Atlantique

13 June 2025

Static Context Header Compression (SCHC) for the Internet Control Message Protocol (ICMPv6)
draft-ietf-schc-icmpv6-compression-02

Abstract

This document describes how the ICMPv6 protocol can be integrated into the SCHC architecture. It extends the YANG Data Model with new field IDs specific to ICMPv6 headers.

To enhance the compression of ICMPv6 error messages, the document also introduces two new Matching Operators and two new Compression Decompression Actions to manipulate the ICMPv6 payload.

Finally, for constrained networks such as LPWAN, it introduces a proxy behavior, where a SCHC Core end-point may anticipate the device reaction to incorrect messages.

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 15 December 2025.

Copyright Notice

Table of Contents

1. Introduction

When applying SCHC compression to IPv6 networks, users expect to perform control operations such as ping to ensure that devices are still active. In the same way, ICMPv6 error messages can be helpful for the Device which may adapt its behavior when the Application becomes unreachable. The Application may also benefit from the ICMPv6 error message produced by the SCHC entity when the compression is not possible.

The compression described in this document is not limited to traffic over LPWANs, but can be applied to any kind of network. The ICMPv6 messages covered by this document are those defined in ICMPv6 protocol [RFC4443]. The compression described in this document does not cover other ICMPv6 messages, such as an extended format of the same messages [RFC4884] and other messages used by the Neighbor Discovery Protocol [RFC4861].

ICMPv6 defines a generic message format, which is used to inform the source of an IPv6 packets about errors during the packet delivery. It also specifies messages used by the ping command to test connectivity with a remote node.

[RFC4443] instantiates four such error messages:

* Destination Unreachable (type = 1),

* Packet Too Big (type = 2),

* Time Exceeded (type = 3) and

* Parameter Problem (type = 4).

[RFC4443] also defines two informational messages, the Echo Request (type=128) and Echo Reply messages (type = 129), which provide support for the ping application.

This document describes recommended compression of ICMPv6/IPv6 messages (including header fields and structured payload) and extends SCHC by specifying new Field Identifiers for ICMPv6 and two MO and two CDA to compress the ICMPv6 payload. This covers different scenarios:

* ICMPv6 messages initiated by SCHC End-Points. They can be sent in their SCHC-compressed form, in ICMPv6 messages traffic. This includes error messages, as well as informational echo request/ reply traffic

* ICMPv6 error messages returned from the Internet after End-Point transmission. The core SCHC forwards a compressed version of the error message to the End-Point, including if necessary a compressed payload.

* Traffic coming from the Internet that would generate an error on the End-Point: if it can detect the situation, the SCHC Core directly responds with an ICMPv6 error message, acting as a surrogate to the End-Point.

2. Terminology

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

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.

* SCHC Core: SCHC End-Point located at the boundary of a regular IP network and a network that applies SCHC compression and fragmentation

* SCHC Device: The other end of the SCHC instance formed with the SCHC core.

* Application: entity sending packets to the SCHC Device or receiving packets from the SCHC Device. The Application may be co-located with the SCHC Core, but is usually located on the regular Internet.

* Regular Internet: Network location carrying uncompressed IPv6 packets.

3. Use cases

In the following sections, we will describe at Section 4 ICMPv6 message compression for all ICMPv6 messages specified in [RFC4443]. We will then extend this basic compression with a specific focus on 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 the SCHC Core using SCHC rules that match the ICMPv6 fields (see Section 5).

* The Device should have sent an ICMP error message, mainly in response to an incorrect incoming IPv6 message. In this case, as much as possible, the SCHC Core should act on behalf of the Device and originate the Unreachable ICMP Destination message, so that the Device and the network are protected from this unwanted traffic (see Section 6).

* 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 7.3 what SCHC compression should be applied. Since ICMPv6 error messages contain in the payload the original message which has triggered the error, SCHC can compress it using the rules in the reverse direction (see Section 7).

4. ICMPv6 compression

This section defines ICMPv6 fields that can be compressed by SCHC. [RFC4443] defines several formats with respect to the type of the ICMPv6 message.

From them, several fields can be extracted (the field ID identifiers are specified in the augmentation of the YANG Data Model Section 9):

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 does not have an impact on the message format.

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

* ICMPv6 Payload is part of the ICMPv6 protocol and is not directly originated from upper layers protocols, so this field may be compressed by SCHC at the ICMPv6 level. In the ICMPv6 error message, the payload can be compressed by SCHC compression rules, as it contains the IPv6 message header responsible for the error. For Echo Request and Echo Reply it contains a specific pattern to set the message length.

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.

Since the fields present in an ICMP message differ from one type to another, it is not possible to use a single rule to compress all ICMP messages. The next section gives some examples of ICMP message compression rules.

4.1. Rule Examples

Table 1 gives an example of the Destination Unreachable message sent to a Device. The Type is 1 and can be elided, Code can be reduced to 3 bits with a Matching List. The Unused field does not appear in the rule, and payload is sent integrally. Since the payload size cannot be easily guessed, this field is marked as variable, which adds 4 bits if its length is less than 255 bytes and 12 bits otherwise.

To reduce the size of the SCHC message, the Payload can be elided with the not-sent CDA instead of the value-sent CDA, or the new CDA introduced Section 7.2 may be used to compress it using SCHC rules.

Field	FL	FP	DI	Value	Matching Operator	CDA	Sent bits
IPv6 Headers description							
ICMPv6 Type	8	1	Dw	1	equal	not-sent	
ICMPv6 Code	8	1	Dw	[0,1,2,3,4,5,6]	match-mapping	mapping-sent	3
ICMPv6 Checksum	1	1	Dw		ignore	compute-*	
ICMPv6 Payload	var	1	Dw	0	ignore	value-sent	(data length*8) + 4 or +12

Table 1: Example of Destination Unreachable compression rule.

Table 2 shows an example of the Packet Too Big message compression Rule. In this Rule, the MTU field is present. If the maximum MTU is 1500 Bytes, the value is coded on 11 bits, therefore, the 21 left-most bit can be elided, with the MSB/LSB MO/CDA.

Field	FL	FP	DI	Value	Matching Operator	CDA	Sent bits
IPv6 Headers description							
ICMPv6 Type	8	1	Dw	2	equal	not-sent	
ICMPv6 Code	8	1	Dw	0	equal	not-sent	3
ICMPv6 Checksum	1	1	Dw		ignore	compute-*	
ICMPv6 MTU	32	1	Dw		MSB(21)	LSB	
ICMPv6 Payload	var	1	Dw	0	ignore	value-sent	(data length*8) + 4 or +12

Table 2: Example of Packet Too Big compression rule.

5. Device does a ping

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

If a ping Echo 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.

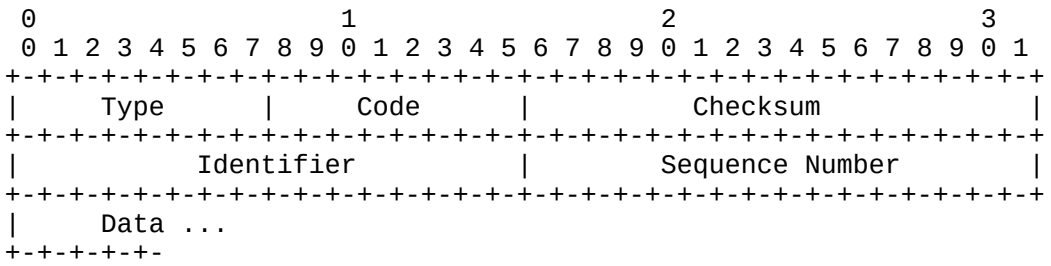

Figure 1: ICMPv6 Echo Request message format

If we assume that one rule will be devoted to compressing Echo Request messages, then the 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-* CDA and therefore is not transmitted.

[RFC4443] states that the identifier and sequence number are meant to "help in matching Echo Responses to this Echo Request" and that they "may be zero". Data are "zero or more bytes of arbitrary data".

For constrained devices or networks, we recommend that the Identifier be zero, the Sequence Number be a counter on 3 bits, and the 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 data part is defined in the rule through the ICMPv6 Payload field. The payload can be sent as a residue with a value-sent. It is also possible to elide the data by setting them in the Target Value and use a not-sent CDA.

When the destination receives the Echo Request message, it will respond with an 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.

5.1. 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
ICMPv6 Checksum	1	1	Dw		ignore	compute-*	
ICMPv6 Payload	var	1	Bi	0	ignore	value-sent	(data*8) + 4 or +12

Table 3: Example of compression rule for a ping from the device

The transmission cost of the Echo Request message is therefore the size of the Rule Id + 3 bits and the data size increased of the Payload residue size. The rule ID Length can be chosen to avoid adding padding.

6. 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 SCHC Core and the ICMPv6 notification should be sent back from there.

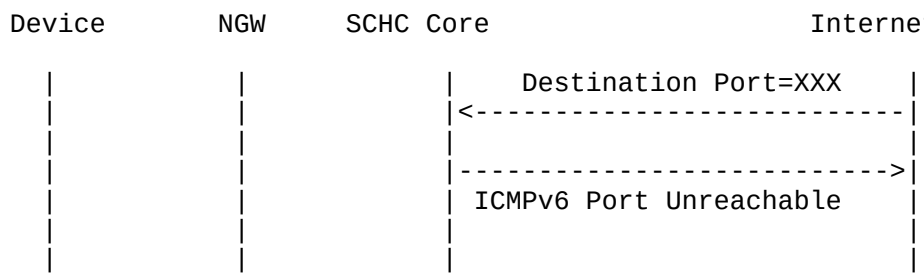

Figure 2: Example of ICMPv6 error message sent back to the Internet

Figure 2 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 SCHC Core 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 allowed by the fragmentation rules.

7. Device is the destination of an ICMPv6 error message

In this situation, 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, the ICMPv6 error message should reach the Device. The data inside this error message includes the packet at the origin of the error. It should be compressed by the SCHC Core, but in the reverse direction. New MOs and CDAs are introduced to perform this operation. The MO check is a rule that matches the Target Value in the forward or reverse direction and the CDA performs this compression.

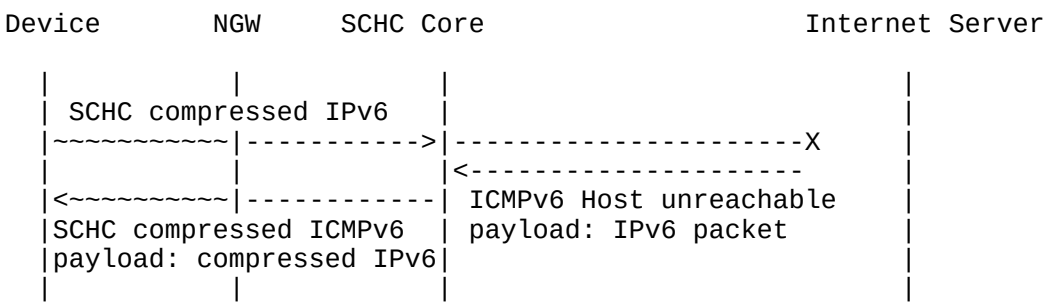

Figure 3: Example of ICMPv6 error message sent back to the Device

Figure 3 illustrates this behavior. The ICMPv6 error message is compressed as described in Section 7.3 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 SCHC Core by reversing the direction as if this message was issued by the device.

7.1. Matching Operator rule match and reverse rule match.

If the Target Value contains a header, this matching operator returns True if a Rule exists in the current Set of Rule to compress it. The selection can either be done:

* in the same direction of the End-Point, this can be used to compress a protocol encapsulated in the header.

* in the reverse direction of the end point, as in an ICMPv6 error message.

7.2. Compression Decompression Actions to compress Target Values.

These CDAs compress-sent and rev-compress-sent compress the Target Value using rules defined in the current Set of Rules. This CDA MUST be used in conjunction with the Matching Operators defined in Section 7.1 according to the direction. The compression is using the same direction as the End-Point, the reverse compression uses the opposite direction.

7.3. Example of ICMPv6 error message compression.

The ICMPv6 error messages defined in [RFC4443] contain the fields shown in Figure 4.

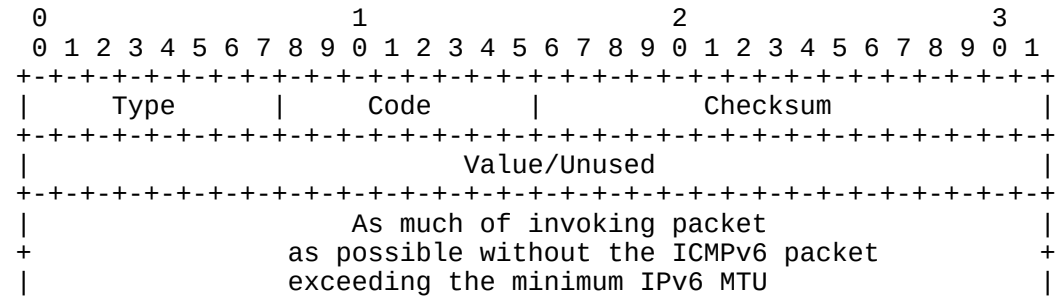

Figure 4: ICMPv6 Error Message format

[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.

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 with a matching list [1, 2, 3, 4] and is therefore compressed 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 4 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	3	equal	not-sent	
ICMPv6 Code	8	[0, 1]	Dw	0	equal	not-sent	1
ICMPv6 Checksum	1	1	Dw		ignore	compute-*	
ICMPv6 Payload	var	1	Dw	0	ignore	not-sent	0

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

The second rule example Table 5 also only sends the ICMP type and code as residue to the device, but 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	3	equal	not-sent	
ICMPv6 Code	8	1	Dw	[0,1]	match-mapping	mapping-sent	1
ICMPv6 Checksum	1	1	Dw		ignore	compute-*	
ICMPv6 Payload	var	1	Dw	0	rev-rule-match	not-sent	

Table 5: 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 6 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 of 4 or 12 bits.

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

Table 6: Example of compression rule for a ICMP error to a device

8. YANG identities and tree

This YANG module extends Field ID identities defined in [RFC9363]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 Mactching Operator identities:

* mo-rev-rule-match: The value contained in the Field Value matches a rule. The direction used for matching is the 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 in it 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.

9. YANG Module

```
<CODE BEGINS> file "ietf-schc-icmpv6@2024-11-20.yang"
module ietf-schc-icmpv6 {
 yang-version 1.1;
 namespace "urn:ietf:params:xml:ns:yang:ietf-schc-icmpv6";
 prefix schc-icmpv6;

 import ietf-schc {
   prefix schc;
 }

 organization
   "IETF Static Context Header Compression (schc) working group";
 contact
   "WG Web:   <https://datatracker.ietf.org/wg/schc/about/>
    WG List:  <mailto:p-wan@ietf.org>
    Editor:   Laurent Toutain
      <mailto:laurent.toutain@imt-atlantique.fr>
    Editor:   Ana Minaburo
      <mailto:ana@minaburo.com>";
 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
     describe ICMPv6 Field IDs, MO and CDA described in RFC YYYY.
     It does not introduce new leaf in the Data Model.";

  revision 2024-11-20 {
    description
      "Initial version for RFC YYYY ";
    reference
      "RFC YYYY: ICMPv6";
  }

  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)
                  Specification";
  }

  // ICMPv6 Fields

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

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

  identity fid-icmpv6-checksum {
    base fid-icmpv6-base-type;
    description
```

```
      "ICMPv6 checksum field present in all ICMPv6 messages.";
  }

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

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

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

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

  identity fid-icmpv6-payload {
    base 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.";
  }
}
<CODE ENDS>
```

Figure 5: YANG module

10. Security considerations

flood the return path with ICMP error messages.

11. IANA Considerations

TODO

12. 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)

13. References

13.1. 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>.

[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>.

[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>.

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