Network Working Group A. Minaburo
Internet-Draft Acklio
Intended status: Informational L. Toutain
Expires: December 8, 2016 Institut MINES TELECOM ; TELECOM Bretagne
June 6, 2016

6LPWA Static Context Header Compression (SCHC) for IPV6 and UDP

draft-toutain-6lpwa-ipv6-static-context-hc-00

Abstract

This document describes a header compression scheme for IPv6, IPv6/UDP based on static contexts. This technique is especially tailored for LPWA networks and could be extended to other protocol stacks.

During the IETF history several compression mechanisms have been proposed. First mechanisms, such as RoHC, are using a context to store header field values and send smaller incremental differences on the link. Values in the context evolve dynamically with information contained in the compressed header. The challenge is to maintain sender's and receiver's contexts synchronized even with packet losses. Based on the fact that IPv6 contains only static fields, 6LoWPAN developed an efficient context-free compression mechanisms, allowing better flexibility and performance.

The Static Context Header Compression (SCHC) combines the advantages of RoHC formal notation, which offers a great level of flexibility in the processing of fields, and 6LoWPAN behavior to elide fields that are known from the other side. Static context means that values in the context field do not change during the transmission, avoiding complex resynchronization mechanisms, incompatible with LPWA characteristics. In most of the cases, IPv6/UDP/CoAP headers are reduced to a small context identifier.

Status of This Memo

1. Introduction

Headers compression is mandatory to bring the internet protocols to the node within a LPWA network. 6LoWPAN and its evolutions do not fulfil the drastic constraints imposed by the radio technology [I-D.minaburo-lp-wan-gap-analysis].

Nevertheless, LPWA networks offer good properties for an efficient header compression:

- o Topology is star oriented. For the needs of this draft, the architecture can be summarized to End-Systems (ES) exchanging information with a single LPWA Compressor (LC). In most of the cases, End Systems and LC form a star topology.

- o Traffic flows are mostly deterministic, since End-Systems embed built-in applications. Contrary to computers or smartphones, new applications cannot be easily installed.

First mechanisms such as RoHC use a context to store header field values and send smaller incremental differences on the link. The first version of RoHC targeted IP/UDP/RTP stack. RoHCv2 extends the principle to any protocol and introduces a formal notation [RFC4997]

describing the header and associating compression functions to each field. To be efficient the sender and the receiver must check that the context remains synchronized (i.e. contains the same values). Context synchronization imposes to periodically send a full header or at least dynamic fields. If fully compressed, the header can be

compatible with LPWA constraints. However, the first exchanges or a context resynchronisation impose to send uncompressed headers, which may be bigger than the original one. This will force the use of inefficient fragmentation mechanisms. For some LPWA technologies, duty cycle limits can also delay the resynchronization. Figure 1 illustrates this behavior.

```
                       sync
             ^         +-+         sync     sync             ^
             | IPv6    | |         +-+       +-+             | IPv6
             v         | |         | |       | |             v
      +------------+   | +-+-+     | |       | |    +------------+
      |       +--+ |   | | | |     | |       | |    | +--+       |
      |       | c| |   | | | +-+-+-+ +-+-+-+-+ |    | | c|       |
      |       | t| |   | | | | | | | | | | | | |    | | t|       |
      |       | x| |   +-+-+-+-+-+-+-+-+-+-+-+-+    | | x|       |
      |       | t| | <----------------------------> | | t|       |
      |       +--+ |                                | +--+       |
      +------------+                                +------------+
```

Figure 1: RoHC Compressed Header size evolution.

On the other hand, 6LoWPAN [RFC4944] is context-free based on the fact that IPv6, its extensions or UDP headers do not contain incremental fields. The compression mechanism described in [RFC6282] is based on sending a 2-byte bitmap, which describe how the header should be decompressed, either using some standard values or information sent after this bitmap. [RFC6282] also allows for UDP compression.

In the best case, when Hop limit is a standard value, flow label, DiffServ fields are set to 0 and Link Local addresses are used over a single hop network, the 6LoWPAN compressed header is reduced to 4 bytes. This compression ratio is possible because the IID are

derived from the MAC addresses and the link local prefix is known from both sides. In that case, the IPv6 compression is 4 bytes and UDP compression is 2 bytes, which fills half of the payload of a SIGFOX frame, or more than 10% of a LoRaWAN payload (with spreading factor 12).

The Static Context Header Compression (SCHC) combines the advantages of RoHC formal notation, which offers a great level of flexibility in the processing of fields, and 6LoWPAN behavior to elide fields that are known from the other side. Static context means that values in the context field do not change during the transmission, avoiding complex resynchronization mechanisms, incompatible with LPWA

characteristics. In most of the cases, IPv6/UDP/CoAP headers are reduced to a small context identifier.

2. Static Context Header Compression

Static Context Header Compression (SCHC) avoids context synchronization, which is the most bandwidth-consuming operation in RoHC. Based on the fact that the nature of data flows is highly predictable in LPWA networks, a static context may be stored on the End-System (ES). The other end, the LPWA Compressor (LC) can learn the context through a provisionning protocol during the identification phase.

The context contains an ordered lists of rules. Each rule is identified by a value, also called context identifier. If the layer 2 allows it, the context id can be carried in the layer 2 header. Otherwise the context id is the first byte of the L2 payload. Being at the boundary between Layer 2 and Layer 3, the context id is also called a SHIM. Different ES will use the same SHIM to identify their own context. An LC needs the ES MAC address to identify the appropriate context in its memory.

Context rules will be used for several purposes:

- o Flow compression: context rules contain a high-level description of the headers' fields and associate a function to each of them.

- o Flow decompression: the function associated to each field indicates also the decompression behavior.

o Uncompressed flow selection: The information stored in the context rule is also used to match incoming packets to check if the compression rule can be applied. There is a strong relation between filtering and decompression. For instance, a flow may be defined as a set of values that correspond to a set of fields. This flow is identified by a SHIM. A destination sharing the same context is able to reconstruct the original header upon reception of a given SHIM.

o Compressed flow selection: when receiving a compressed packet, information in the context (typically the SHIM) will be used to select the decompression rule in combination with the ES MAC address.

o Packet filtering: LPWA can be easily subject to DoS attacks. If a packet is not explicitly assigned to a specific context, then incoming packets will be discarded.

3. Filtering functions

The compression/decompression mechanisms proposed in this Figure 2 is a combinaison of 6LoWPAN principles, which are efficient in sending only information what cannot be reconstructed at the other end, and RoHCv2 which assigns compression and decompression functions to each field. The use of a context avoids sending well-known information.

Function	Selec comp	Selec dec.	Compression	Decompression
ignore	no	no	elided	add value stored in ctxt
send-value	no	no	send value	build field from value
send-value-lbs	yes	no	send lsb	concatenation ctxt val+lsb
send-value-filter	no	yes	send value	elided
not-sent	yes	no	elided	add value stored in ctxt
just-check	yes	yes	nothing	nothing
compute-IPv6-length	no	no	elided	compute IPv6 length
compute-UDP-length	no	no	elided	compute UDP length
compute-UDP-checksum	no	no	elided	compute UDP checksum
ESiid-MAC	no	no	elided	build IID from L2 ES addr

```
|LAiid-MAC           |no    |no    |elided       |build IID from L2 LA addr |
\--------------------+-----+-----+-------------+--------------------------/
```

Figure 2: Simplified Protocol Stack for LP-WAN

Figure 2 lists all the functions defined to compress and decompress a field. The first column gives the function's name.

The second column describes the rule selection property of the function. Selection determines if the compression rule can be applied to a packet. A comparison is made between the value stored in the context and the field's value. Generally it is an equality between the field value and a associated context value, but functions may define more complex matching rules. To succeed and apply the compression/decompression rule, the comparisons of all header fields marked as "yes" in this column must be true.

The third column indicates which function can be used to select the appropriate rules for decompression. Typically it will be the SHIM and the MAC address.

Fourth column outlines the compression process.

Last column outlines the decompression process.

As with 6LoWPAN, the compression process may produce some data, where fields that were not compressed (or were partially compressed) will be sent in the order order of the original packet. Information added by the compression phase must be aligned on byte boundaries, but each individual compression function may generate any size.

```
/--------------------+---------------------+-------------------------------------
| Field              |Function             | Behavior
+--------------------+---------------------+-------------------------------------
|IPv6 version        |ignore               |No IPv4: not sent, not used for sel
|                    |send-value-filter*   |With IPv4: sent value, used for sel
+--------------------+---------------------+-------------------------------------
|IPv6 DiffServ       |not-sent*            |The value is not sent, but each end
|IPv6 Flow Label     |                     |agree on a value, which can be some
```

```
|                    |                     |different from 0.
|                    |send-value           |If DiffServ field varies it is sent
+--------------------+---------------------+-----------------------------------
|IPv6 Length         |compute-IPv6-length  |Dedicated function to reconstruct v
+--------------------+---------------------+-----------------------------------
[IPv6 Next Header    |not-sent*            |Value is known in the ctxt.
|                    |send value           |Same behavior as 6LoWPAN
+--------------------+---------------------+-----------------------------------
|IPv6 Hop Limit      |ignore               |The receiver will put a value store
|                    |                     |the context. It may be different fr
|                    |                     |one originally sent, but in s star
|                    |                     |topology, there is not risk of loop
|                    |not-sent*            |Receiver and sender agree on the va
|                    |                     |If the value is not correct the pac
|                    |                     |is discarded
|                    |send-value           |Explicitly sent
+--------------------+---------------------+-----------------------------------
|IPv6 ESPrefix       |not-sent*            |The 64 bit prefix is stored on the
|IPv6 LCPrefix       |send-value           |Explicitly send 64 bits on the link
+--------------------+---------------------+-----------------------------------
|IPv6 ESiid          |not-sent             |IID is not sent, but stored in the
|IPv6 LCiid          |ESiid-MAC | LCiid-MAC|IID is built from the ES MAC addres
|                    |send-value*          |IID is explicitly sent on the link.
|                    |                     |size depends of the L2 technology
+--------------------+---------------------+-----------------------------------
|UDP ESport          |not-sent             |In the context
|UDP LCport          |send-value*          |Send the 2 bytes of the port number
|                    |send-value-lsb*      |Send least significant bits of the
|                    |                     |number.
+--------------------+---------------------+-----------------------------------
|UDP length          |compute-UDP-length   |Dedicated function to reconstruct v
|UDP Checksum        |compute-UDP-checksum |Dedicated function to reconstruct v
+--------------------+---------------------+-----------------------------------
  * field used for rule selection.
```

Figure 3: SCHC functions' example assignment for IPv6 and UDP

Figure 3 gives an example of function assignment to IPv6/UDP fields, a star after the function name indicates when a field participates in the context id selection.

3.1. Compression functions

3.1.1. Ignore

Ignore function defines a field that does not participate to the rule selection process. The field value will not be sent on the wire and can be reconstructed on the other side.

The ignore function can be assigned to the IPv6 version field (if IPv4 is not used in the system). IPv6 Hop Limit may also be a candidate in some cases. Hop Limit value will not affect the flow selection process. The receiver may assign a static value. If there is a risk of loop creation (i.e. non-star topology), the send-value function must be used instead.

3.1.2. Send-value

This function is used to transmit the full field value that is not stored in the context. In the decompression phase, the receiver uses the transmitted value for reconstructing the field. This field cannot participate to the selection process since it can vary other the time.

The send-value function may be used to send interface IID in a meshed topology.

3.1.3. Send-value-lsb

This function allows to send only the less significant bits of a value. The context contains the size of the less significant bits and a reference value.

Send-value-lbs is involved in the rule selection. The most significant bit of field's value must matches the most significant bits of the context reference value.

The sender send on the radio link only the less significant bits. The receiver reconstruct the initial value by concatenating the most significant bits of reference value contained in the context and the less significant bits received.

This function can be used to define a port range and allow several IP flows to share the same context.

3.1.4. Send-value-filter

In the compression phase, a field assigned with this value is sent on the radio link. It does not influence the rule selection.

Value sent on the link influences the rule selection for decompression.

Typically, this function is used to transmit the SHIM field and proceed to rule identification when the header is decompressed. The SHIM is elided from the uncompressed header.

3.1.5. Not-sent

In the compression phase, a field assigned with this value is not sent on the radio link. This influence the rule selection.

In the decompression phase, the uncompressed value is the one stored in the context. Since the value is not send on the radio link, it cannot influence the flow selection.

IPv6 protocol identifier, UDP ports number fields can be assigned to this function. This avoid to send then on the link.

3.1.6. just-check

The field value is checked for the rule selection, but nothing is sent on the radio link.

This can be used to include L2 parameters such as addresses in the rule selection.

3.1.7. compute-IPv6-lenght, compute-UDP-length, compute-UDP-checksum

Fields assigned with this functions are not sent on the radio link, they do not participate to the rule selection process. They are computed during the decompression phase.

This functions are specific to a field in the header.

3.1.8. ESiid-MAC, LCiid-MAC

These functions are used to process respectively the End System and the LPWA AP Interface Identifier. The IID value is computed from addresses present in the MAC header. The computation depends of the technology and the MAC address size. The values of the field do not

participate to the flow selection since they are sent on the radio link at layer 2.

These functions can be used in case of the star topology.

4. Examples

This section gives some scenarios of the compression mechanism. Note that for reasons of simplicity in this example CoAP is not compressed, it will be described later with the same principles. The goal is to illustrate the SCHC behaviour.

4.1. IPv6/UDP compression in a star topology

The most common case will be a LPWA end-system embeds some applications running over CoAP. Typically one will be for the device management using the COMI/CoOL protocol (using UDP ports 123 and 124). The second one will be a CoAP server for measurements done by the end-system (using ports 5683). A third UDP traffic is for legacy applications using different ports numbers. Figure 4 presents the protocol stack for this end-system. IPv6 and UDP are represented with dotted lines since these protocols are compressed on the radio link. The context ID is represented by a SHIM (respectively 0, 1 and 2).

```
|
|/c  |/a   |
+----------+---------+---------+
|   CoAP   |  CoAP   | legacy  |
+----||----+---||----+---||----+
.   UDP    .  UDP    |   UDP   |
................................
.   IPv6   .  IPv6   .  IPv6   .
+--SHIM0------SHIM1-----SHIM2--------------------------+
|      6LPWA L2 technologies                           |
+------------------------------------------------------+
      End System or LPWA GW
```

Figure 4: Simplified Protocol Stack for LP-WAN

Note that in some LPWA technologies, only End Systems have a MAC

address. Therefore it is necessary to define an IID for the Link Local address for the LPWA Compressor.

Field	Function	Ctxt Value	Sent compressed
LPWA SHIM	send-value-filter	0	0
IPv6 version	ignore		

IPv6 DiffServ	not-sent	0	
IPv6 Flow Label	not-sent	0	
IPv6 Length	compute-IPv6-length	-----------------	
IPv6 Next Header	not-sent	17	
IPv6 Hop Limit	ignore	1	
IPv6 ESprefix	not-sent	FE80::/64	
IPv6 ESiid	ESiid-MAC		
IPv6 LCprefix	not-sent	FE80::/64	
IPv6 LCiid	LCiid-value	::1	
UDP ESport	not-sent	123	
UDP LCport	not-sent	124	
UDP Length	compute-UDP-length	-----------------	
UDP checksum	compute-UDP-checksum	-----------------	

Field	Function	Ctxt Value	Sent compressed
LPWA SHIM	send-value-filter	1	1
IPv6 version	ignore		
IPv6 DiffServ	not-sent	0	
IPv6 Flow Label	not-sent	0	
IPv6 Length	compute-IPv6-length	-----------------	
IPv6 Next Header	not-sent	17	
IPv6 Hop Limit	ignore	1	
IPv6 ESprefix	not-sent	FE80::/64	
IPv6 ESiid	ESiid-MAC		
IPv6 LCprefix	not-sent	FE80::/64	
IPv6 LCiid	LCiid-value	::1	

UDP ESport	not-sent	5683	
UDP LCport	not-sent	5683	
UDP Length	compute-UDP-length	----------------	
UDP checksum	compute-UDP-checksum	----------------	

Field	Function	Ctxt Value	Sent compressed
LPWA SHIM	send-value-filter	2	2
IPv6 version	ignore		
IPv6 DiffServ	not-sent	0	
IPv6 Flow Label	not-sent	0	
IPv6 Length	compute-IPv6-length	----------------	
IPv6 Next Header	not-sent	17	

IPv6 Hop Limit	ignore	1	
IPv6 ESprefix	not-sent	FE80::/64	
IPv6 ESiid	ESiid-MAC		
IPv6 LCprefix	not-sent	FE80::/64	
IPv6 LCiid	LCiid-value	::1	
UDP ESport	send-value		port number
UDP LCport	send-value		port number
UDP Length	compute-UDP-length	----------------	
UDP checksum	compute-UDP-checksum	----------------	

Figure 5: Simplified Protocol Stack for LP-WAN

Figure 6 shows an alternative way to compress more efficiently port numbers. The send-value-lsb allows to send in one byte the two ports number differences. Since the compressed information must be aligned on byte boundary, it has been chosen in the example a size of 4 bits for each lsb.

Field	Function	Ctxt Value	Sent compressed
LPWA SHIM	send-value-filter	2	2

IPv6 version	ignore		
IPv6 DiffServ	not-sent	0	
IPv6 Flow Label	not-sent	0	
IPv6 Length	compute-IPv6-length	----------------	
IPv6 Next Header	not-sent	17	
IPv6 Hop Limit	ignore	1	
IPv6 ESprefix	not-sent	FE80::/64	
IPv6 ESiid	ESiid-MAC		
IPv6 LCprefix	not-sent	FE80::/64	
IPv6 LCiid	LCiid-value	::1	
UDP ESport	send-value-lsb	4+ES ref val	lsb
UDP LCport	send-value-lsb	4+LC ref val	lsb
UDP Length	compute-UDP-length	----------------	
UDP checksum	compute-UDP-checksum	----------------	

Figure 6: Alternative compressions of port numbers

4.2. Global addresses

The scenario depicted Figure 7, management remains with Link Local addresses, but the CoAP message are sent to an external server 2001:db8:1::1 and the legacy to another server 2001:db8:2::1/64. The EC must be configured with the prefix used by the LC to forward traffic. This prefix could be changed using a management procedure if needed.

Field	Function	Ctxt Value	Sent compressed
LPWA SHIM	send-value-filter	0	0
IPv6 version	ignore		
IPv6 DiffServ	not-sent	0	
IPv6 Flow Label	not-sent	0	
IPv6 Length	compute-IPv6-length	----------------	

IPv6 Next Header	not-sent	17	
IPv6 Hop Limit	ignore	1	
IPv6 ESprefix	not-sent	FE80::/64	
IPv6 ESiid	ESiid-MAC		
IPv6 LCprefix	not-sent	FE80::/64	
IPv6 LCiid	LCiid-value	::1	
UDP ESport	not-sent	123	
UDP LCport	not-sent	124	
UDP Length	compute-UDP-length	-----------------	
UDP checksum	compute-UDP-checksum	-----------------	

Field	Function	Ctxt Value	Sent compressed
LPWA SHIM	send-value-filter	1	1
IPv6 version	ignore		
IPv6 DiffServ	not-sent	0	
IPv6 Flow Label	not-sent	0	
IPv6 Length	compute-IPv6-length	-----------------	
IPv6 Next Header	not-sent	17	
IPv6 Hop Limit	ignore	1	
IPv6 ESprefix	not-sent	2001:db8:3::/64	
IPv6 ESiid	ESiid-MAC		
IPv6 LCprefix	not-sent	2001:bd8:1::/64	
IPv6 LCiid	LCiid-value	::1	
UDP ESport	not-sent	5683	

UDP LCport	not-sent	5683	
UDP Length	compute-UDP-length	-----------------	
UDP checksum	compute-UDP-checksum	-----------------	

Field	Function	Ctxt Value	Sent compressed
LPWA SHIM	send-value-filter	2	2
IPv6 version	ignore		

```
|IPv6 DiffServ   | not-sent               | 0              |
|IPv6 Flow Label | not-sent               | 0              |
|IPv6 Length     | compute-IPv6-length    |----------------|
|IPv6 Next Header| not-sent               | 17             |
|IPv6 Hop Limit  | ignore                 | 1              |
|IPv6 ESprefix   | not-sent               |2001:db8:3::/64 |
|IPv6 ESiid      | ESiid-MAC              |                |
|IPv6 LCprefix   | not-sent               |2001:db8:2::/64 |
|IPv6 LCiid      | LCiid-value            | ::1            |
+================+========================+================+=================
|UDP ESport      | send-value             |                | port number
|UDP LCport      | send-value             |                | port number
|UDP Length      | compute-UDP-length     |----------------|
|UDP checksum    | compute-UDP-checksum   |----------------|
+================+========================+================+=================
```

Figure 7: Compression with global addresses

5. CoAP compression

TBD

6. Normative References

[I-D.minaburo-lp-wan-gap-analysis]
Minaburo, A., Pelov, A., and L. Toutain, "LP-WAN GAP Analysis", draft-minaburo-lp-wan-gap-analysis-01 (work in progress), February 2016.

[RFC4944] Montenegro, G., Kushalnagar, N., Hui, J., and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks", RFC 4944, DOI 10.17487/RFC4944, September 2007, <http://www.rfc-editor.org/info/rfc4944>.

[RFC4997] Finking, R. and G. Pelletier, "Formal Notation for RObust Header Compression (ROHC-FN)", RFC 4997, DOI 10.17487/RFC4997, July 2007, <http://www.rfc-editor.org/info/rfc4997>.

[RFC6282] Hui, J., Ed. and P. Thubert, "Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks", RFC 6282, DOI 10.17487/RFC6282, September 2011, <http://www.rfc-editor.org/info/rfc6282>.

Authors' Addresses

Ana Minaburo
Acklio
2bis rue de la Chataigneraie
35510 Cesson-Sevigne Cedex
France

Email: ana@ackl.io

Laurent Toutain
Institut MINES TELECOM ; TELECOM Bretagne
2 rue de la Chataigneraie
CS 17607
35576 Cesson-Sevigne Cedex
France

Email: Laurent.Toutain@telecom-bretagne.eu