Improvements for the ROHC Profile Set Update
<draft-ietf-rohc-rfc3095bis-improvements-02.txt>

Status of this memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with Section 6 of BCP 79.

By submitting this Internet-Draft, each author accepts the provisions of Section 3 of BCP 78.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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".

The list of current Internet-Drafts can be accessed at http://www.ietf.org/1id-abstracts.html

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html

This document is a submission of the IETF ROHC WG. Comments should be directed to the ROHC WG mailing list, rohc@ietf.org.

Abstract

RFC 3095 defines the RObust Header Compression (ROHC) framework and profiles. Now with 5 years of ROHC experience, it has been decided to revise RFC 3095 through the development of an updated profile set. This document collects all improvements that have been agreed to be addressed by this updated and improved ROHC revision.
1. Introduction

RFC 3095 [1] defines the RObust Header Compression (ROHC) framework and profiles for IP, UDP, RTP and ESP. When specified, ROHC was created to meet many new challenging requirements, and lots of new compression methods were used. With 5 years of ROHC experience, many lessons have been learned when it comes to what parts of RFC 3095 were actually useful, as well as what parts should had been omitted or simplified. This document collects all improvements that have been agreed to be addressed when RFC 3095 is now to be revised through the development of an updated profile set.

Note that all section and chapter references in this document refer to [1], where not stated otherwise.
2. General improvements

2.1. Editorial restructuring

Experience has shown that the structure of RFC3095 could have been much better, and normative specifications could have been less fragmented. One goal should thus be to do proper restructuring to make the documentation easier to take in. One fundamental editorial restructuring is to explicitly define and specify the ROHC framework in a separate document. The profiles will then be specified in their own document(s), and will most probably incorporate the updated profiles from both RFC3095 and the "add-on" specifications ROHC IP-only and ROHC UDP-Lite into one profile set documentation.

2.2. List compression should not be used for IP extension headers

RFC 3095 defines list compression as a generic mechanism ([1], section 5.8) that is used for both RTP CSRC lists ([1], section 5.8.6) and IP extension headers ([1], section 5.8.4-5.8.5). In the former case, list compression is indeed very suitable, as the scheme maps very well to the expected behavior of CSRC items. However, using list compression for IP extension headers is really hard to justify, and makes ROHC unnecessarily complex. Instead, extension headers should be treated like all other headers, with static and dynamic chains. This is the approach taken for ROHC-TCP, and should be applied also to an updated RFC 3095.

2.3. List compression should only use the generic scheme

List compression ([1], section 5.8) defines four different encoding schemes to be used for compressed lists. There is one generic encoding scheme, and three additional optimization schemes based on reference-based list compression. Implementers have found that using reference-based schemes implies unreasonable decompressor memory demands and implementation complexity, while the potential gain is realistically none. The use of the type field in compressed lists should thus be deprecated and only type 0 encoding should be allowed.

2.4. Multiple operating modes should be avoided, as in ROHC-TCP

RFC 3095 uses several operating modes with complicated transition procedures to safeguard against incorrect packet interpretation, as packet formats differ between modes. The multi-mode approach of RFC 3095 has made the specification unnecessarily complex, and experience has shown that this was not a preferable approach. By streamlining the protocol to one single mode, the number of different packet formats will be reduced, the compression and decompression control logic needed will be significantly simplified, mode transition procedures can be eliminated, non-updating packets can be avoided, etc. When developing the ROHC TCP profile, the ROHC WG has already concluded that the RFC 3095 mode concept is not to be used again.
Consequently, there are no explicit modes in ROHC-TCP, but only one consistent logic is used exclusively, both for unidirectional and bidirectional operation. An updated version of the RFC 3095 profiles should follow that approach.

2.5. UO-1-ID should not be allowed to carry extension 3

UO-1-ID is the only UO-1 format that can carry an extension (see section 5.7.3 and 5.7.5 of [1]). The updating properties of UO-1 is also so that when carrying an extension 3, all fields except SN, TS, and IP-ID are non-updating, which is a fundamental exception to normal UO operation. The usefulness of partially updating UO packets is really questionable. This "feature" is also only available for contexts with a non-random IP-ID, and is thus mainly a useless inconsistency. In an updated RTP profile, which is the only profile using this packet type, UO-1-ID should thus not be allowed to carry extension 3.

2.6. No sequential compression for outer IP-ID

The ROHC 3095 profiles define a mechanism for compression of the IP-ID, not just for the innermost IP header, but also for a potential outer (second innermost) IP header (see section 5.7 of [1]). It is however really unrealistic to expect the outer header IP-ID to be compressible from the sequence number of the RTP header or a compressor-generated sequence number, while a ROHC-RTP decompressor must still be implemented to handle such a case if it indeed happens. This is extremely far-fetched, and the compressor should instead simply have to identify an outer IP-ID as either random or constant (for constant IP-ID handling, see section 3.3 of [2]).

2.7. ESP NULL-encryption compression should not compress trailer

The ESP NULL-encryption compression mechanism defined for ROHC compresses not just the header, but also the trailer of the packet. Apart from being a conceptual exception in the sense that it does not only compress the header but also tampers with the payload, the scheme makes assumptions that reduces its applicability, which is already very limited, to a single corner case. Considering the relative complexity of implementing it along with the small gain and limited applicability, this mechanism should be significantly simplified. The ESP NULL-encryption compression mechanism is defined in section 5.8.4.3 of [1].

2.8. CRC calculation should not use STATIC/DYNAMIC separation

The CRC_STATIC and CRC_DYNAMIC separation was intended to reduce the computational complexity of CRC calculation. However, the separation has proven to make implementation painful, and it actually also adds complexity both in terms of data structures and computation. The STATIC/DYNAMIC CRC separation should therefore be deprecated.
3. Minor improvements

3.1. Meaning of CC=0 for CSRC list presence

Regarding the CC field and CSRC list, the following interpretation has been proposed as an improvement:

"It should be noted that when the value of this CC field is equal to zero, there is no Generic CSRC list present in the dynamic chain, i.e. the field comment should have said "variable length, present if CC > 0"."

3.2. Size of list compression table for RTP CSRC

List compression formats are defined with 3 or 7 bit list item index identifiers (see section 5.8.6.1 of [1]). As there is no additional explicit restriction on the maximal number of list items, up to 128 items must be supported, which implies a significant memory demand for an extreme corner-case. One RTP packet can have up to 15 CSRC items, and that is probably a well over-provisioned number. Since items can always be re-assigned, it is therefore suggested to have an upper limit on the number of CSRC list item index identifiers, with a max value of either 16 or 32.

3.3. The p-value for 5-bit SN

For the RFC 3095 RTP profile, p-values for SN fields are defined in the beginning of section 5.7 of [1], as follows:

\[ p = \begin{cases} 1 & \text{if bits(SN) } \leq 4 \\ 2^{(\text{bits(SN)}-5)} - 1 & \text{if bits(SN) } > 4 \end{cases} \]

This would mean p=1 for bits(SN)=4, p=1 for bits(SN)=6, p>1 for bits (SN)>6, but for bits(SN)=5, p would then be 0. This is illogical, and obviously a mistake. One reason it was not noticed might be that the RTP profile does not have any packet format with 5 bits of SN. However, the ESP profile refer to the RTP profile for p values (section 5.12 of [1]), and in the ESP profile there are packet formats with 5 bits of SN. The p-values should thus be redefined as follows:

\[ p = \begin{cases} 1 & \text{if bits(SN) } \leq 5 \\ 2^{(\text{bits(SN)}-5)} - 1 & \text{if bits(SN) } > 5 \end{cases} \]

It should be noted that the UDP profile has a fixed p-value of -1, motivated by the use of a compressor-generated SN (section 5.11 of [1]), and is thus not affected by the incorrectly specified p-value, although the USP profile has packet formats with 5 bits of SN. Note however the recommendation in section 3.4 of this document.
3.4. The UDP profile should have same p-value as other profiles

Since the UDP profile makes use of a compressor-generated SN instead of an SN taken from an uncompressed header field, it has in section 5.11 of [1] been given a fixed p-value of -1. This made sense, as one design assumption was in-order delivery from compressor to decompressor. However, as the interest in using ROHC also over channels that can not guarantee in-order delivery has gained momentum, this design choice becomes less appropriate, as described in [3]. In an updated version of the UDP profile, it should thus be given the same p-values as for RTP and ESP, i.e. as outlined in section 3.3 of this document, potentially with an increased reordering-tolerance, see further section 5 of this document.

3.5. Local repair should be completely optional

In section 5.3.2.2.3-5.3.2.2.5 of [1], possibilities to do local decompressor repair attempts upon decompression failures are looked at, and example procedures are described. Section 5.3.2.2.3 says:

A. "First, attempt to determine whether SN LSB wraparound (case 3) is likely, and if so, attempt a correction. For this, the algorithm of section 5.3.2.2.4 MAY be used."

and it continues:

B. "Second, if the previous step did not attempt a correction, a repair should be attempted under the assumption that the reference SN has been incorrectly updated. For this, the algorithm of section 5.3.2.2.5 MAY be used."

These are good examples of potential implementation improvements, and the procedures are described as optional, i.e. with the use of "MAY". However, both these paragraphs then take one step further and actually mandates local repair procedures by saying:

C. "If another algorithm is used, it MUST have at least as high a rate of correct repairs as the one in 5.3.2.2.4 (or 5.3.2.2.5, respectively).

This should be a local decompressor implementation option, and it is therefore suggested to exclude the mandating text C.

4. Improvements already applied to the IP-only and UDP-Lite profiles

4.1. Handling Multiple Levels of IP Headers

The profiles in RFC 3095 can only handle compression of packet streams with at most two IP headers. The IP-only profile defines a generic way of handling multiple IP headers (see section 3.2 of [2]).
4.2. The CONTEXT_MEMORY Feedback Option

To provide means for a decompressor implementation to have an upper limit on its available context memory size, the IP-only profile defines an additional feedback option to use (see section 3.7 of [2]). The CONTEXT_MEMORY option informs the compressor that the decompressor does not have sufficient memory resources to handle the context of the packet stream, as the stream is currently compressed.

4.3. Compression of constant IP-ID (IPv4 only)

Most IPv4 stacks assign an IP-ID according to the value of a counter, increasing by one for each outgoing packet. ROHC-RTP therefore compresses the IP-ID field using offset IP-ID encoding based on the RTP SN. For stacks generating IP-ID values using a pseudo-random number generator, the field is not compressed and is sent as-is in its entirety as additional octets after the compressed header.

Cases have also been found where an IPv4 stack uses a constant value for the IP Identifier. When the IP-ID field is constant, it cannot be compressed using offset IP-ID encoding and the field must be sent in its entirety, although it is completely static and could have been completely omitted in compressed headers. The IP-only profile addresses this problem and defines an additional mechanism to cope efficiently with constant IP-ID (see section 3.3 of [2]).

5. Adding tolerance to reordering

RFC 3095 was written based on the assumption of in-order packet delivery between compressor and decompressor. Since the publication of RFC 3095, it has been clear that using ROHC would be desirable also in environments where in-order delivery can not be guaranteed. The challenges associated with such usage have been analyzed in an informational ROHC WG document "ROHC over channels that can reorder packets" [3], and the finding of that document should be used as a basis when developing an enhanced ROHC that can tolerate a certain amount of reordering, possibly a configurable reordering tolerance.

6. Implementation stuff that should go out of the specification

There is a significant amount of implementation ideas given in chapter 6 of, both potential implementation enhancement, implementation API parameters, as well as data structures. It is sometimes useful to have such material being provided in an appendix, as it can help implementers. However, in this case it has been clear that in the way these things were included in RFC 3095, more concerns than benefits were created. There are several reasons for this, one is that these parts were not included as an appendix, but actually
part of the specification itself. Also, the size and overall structure of the whole RFC can easily make an implementer confused about what is actually part of the standard.

In general, it is thus suggested that an update of RFC 3095 should have larger implementation example material, if any, in an appendix to make it clearer that it is not part of the actual standard. Further, reducing the amount of material would be desirable, to make the whole documentation more concise.

What to do with the various subsections of chapter 6 is discussed below, along with other informational parts or concepts that can be questioned.

6.1. Reverse decompression

Reverse decompression is described in section 6.1. This is a very questionable implementation enhancement, and should preferably be removed, or at least be put in an appendix.

6.2. Implementation parameters and signals

In section 6.3, various potential API parameters are defined, although only informatively, as they are not at all necessary from a ROHC protocol point of view. Unfortunately, the way this section is written might make implementer’s believe this is actually part of the standard, as it even makes use of RFC 2119 keywords.

This section should thus be revised, simplified, and it should be made clear that it is an API parameter proposal, nothing more, nothing less. The result could potentially be put in an appendix of the profiles specification.

6.3. Decompressor resource limitations

Section 6.4 of discusses how to handle resource limitations at the decompressor. This is typical implementation guidelines that fits very well in an "implementation issues" section, and should thus be kept. Note that the addition of the CONTEXT_MEMORY feedback option (see section 4.2 of this document) affects this discussion, which would have to be updated accordingly.

6.4. Implementation structures

Section 6.5 provides some explanatory material on data structures that a ROHC implementation will have to maintain in one form or another. The section explicitly states the explanatory nature of it, and points out that it is not intended to constrain implementations. However, it is far from clear whether this material is actually useful. It should therefore be revised, potentially removed, or at least put in an appendix.
6.5. The state concept

The concept of states (FO/SO), as used in a normative manner throughout RFC 3095, should be removed from the spec or at least rewritten so that it becomes clear that this is a descriptive concept and not at all normative. Mentioning of implementation parameters, such as k_2/n_2 and k_3/n_3 should be dropped, or it should at least be made clear that these are just example parameters in example algorithms. Probably the entire state concept can be dropped, since it really describes implementation choices. It can be used informatively, but that should then be made clear.

7. Security considerations

This document provides guidelines for how to specify new protocols, and those protocols will of course have to consider security aspects.

8. IANA considerations

This document does not require any IANA actions.

9. Informative References


10. Authors Addresses

Lars-Erik Jonsson
Ericsson AB
Box 920
SE-971 28 Lulea, Sweden
Phone: +46 8 404 29 61
EMail: lars-erik.jonsson@ericsson.com
Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Copyright Statement

Copyright (C) The Internet Society (2006). This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

Disclaimer of Validity

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

This Internet-Draft expires September 24, 2006.