RObust Header Compression (ROHC):
ROHC over Channels that can Reorder Packets
<draft-pelletier-rohc-over-reordering-00.txt>

Status of this memo

Abstract

   RObust Header Compression (ROHC), RFC 3095, defines a framework for
   header compression, along with a number of compression protocols
   (profiles). One operating assumption for the profiles defined in RFC
   3095 is that the channel between compressor and decompressor is
   required to maintain packet ordering. This document discusses aspects
   of using ROHC over channels that can reorder packets. It provides
   guidelines on how to implement existing profiles over such channels,
   as well as suggestions for the design of new profiles.

Table of Contents

1. Introduction

RObust Header Compression (ROHC), RFC 3095 [2], defines a framework for header compression, along with a number of compression protocols (profiles). One operating assumption for the profiles defined in RFC 3095 is that the channel between compressor and decompressor is required to maintain packet ordering for each compressed flow. The motivation behind this assumption was that the primary candidate channels considered did guarantee in-order delivery of header-compressed packets; making this assumption made it possible to improve the compression efficiency and the tolerance to packet loss, objectives that were on top of the requirements list at the time.

Since the publication of RFC 3095 in 2001, the question about ROHC operation over channels that do not guarantee in-order delivery has surfaced several times; arguments that ROHC cannot perform adequately over such channels have even been heard. Specifically, this has been raised as a weakness when compared to other header compression alternatives, as RFC 3095 explicitly states its inability to operate if in-order delivery is not guaranteed. For those familiar with the details of ROHC and of other header compression schemes, it is clear that this is a misconception; but it can also be easily understood that the wording used in RFC 3095 can lead to such interpretation.

This document discusses the various aspects of implementing ROHC over channels that can reorder header-compressed packets. It explains different ways of implementing the profiles found in RFC 3095, as well as other profiles based on those profiles, over reordering channels. This can be achieved either by ensuring that compressor implementations uses compressed headers that are sufficiently robust to the expected possible reordering, and/or by modifying decompressor implementations to tolerate reordered packets. Ideas regarding how existing profiles could be updated and how new profiles can be defined to cope efficiently with reordering are also discussed.

2. Terminology

This document uses terminology consistent with RFC 3759 [3], and is in itself only informative. Although it does discuss technical aspects of implementing the ROHC specifications in particular environments, it does not specify any new technology.

However, the document discusses possible ways of modifying existing ROHC implementations and/or specifications to address its objectives. In those parts of the document, the key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD, "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" are to be interpreted as

described in RFC 2119 [1].

ROHC

   The term "ROHC" herein refers to the following profiles:

      - 0x0001, 0x0002 and 0x0003 defined in RFC 3095 [2];
      - 0x0004 for compression of IP-only headers [5];
      - 0x0007 and 0x0008 for compression of UDP-Lite headers [6].

   The term "ROHC" excludes the following profiles, that are either
   not affected by reordering or that have the assumption of in-order
   delivery as a fundamental requirement for their proper operation:

      - 0x0000 (uncompressed) [2];
      - 0x0005 (LLA) [7] and 0x0105 (R-mode extension to LLA) [8];

Reordering

   A type of transmission taking place between compressor and
   decompressor where in-order delivery of header-compressed
   packets is not guaranteed.

Reordering Channel

   A connection over which reordering, as defined above, can occur.

Sequentially early packet

   A packet that reaches the decompressor before one or several
   packets of the same CID that were delayed on the link. At the time
   of the arrival of a sequentially early packet, the packet(s)
   delayed on the link cannot be differentiated from lost packet(s).

Sequentially late packet

   A packet is late within its sequence if it reaches the
   decompressor after one or several other packets belonging to the
   same CID have been received, although the sequentially late packet
   was sent from the compressor before the other packet(s).

Updating packet

   A packet that updates the context of the decompressor, i.e. all
   packets carrying a CRC calculated over the uncompressed header.

Non-updating packet

   A packet that carries a CRC calculated over the uncompressed
   header updates the context of the decompressor when it is
   successfully decompressed. A packet without such a CRC is thus

referred to as a non-updating packet.

Change packet

> A packet that updates one or more fields of the context other than
> the fields pertaining to the functions established with respect to
> the sequence number (SN). Specifically, it is a packet that
> updates fields other than the SN, IP-ID or RTP timestamp (TS).

3. Applicability of this Document to ROHC Profiles

This document addresses general reordering issues for ROHC profiles.
The foremost objectives are to ensure that ROHC implementations will
not forward packets with incorrectly decompressed headers to upper
layers, as well as to limit the possible increase in the rate of
decompression failures or in events leading to context damage, when
compression is applied over reordering channels.

3.1. Profiles within Scope

The solutions outlined in following sections are generally applicable
to profiles 0x0001 (RTP), 0x0002 (UDP) and 0x0003 (ESP) defined in
RFC 3095 [2]. Profile 0x0000 (uncompressed) is not affected by
reordering, as the headers are sent uncompressed. The solutions also
apply to profiles for IP-only (0x0004) [5] and for UDP-Lite (0x0007
and 0x0008) [6]. These profiles are based on the profiles of RFC 3095
[2] and inherently make the same in-order delivery assumption.

3.2. Profiles with Special Considerations

Special considerations are needed to make some of the implementation
solutions of sections 6.1 and 6.2 applicable to profiles 0x0002 (UDP)
[2], 0x0004 (IP-only) [5], and 0x0008 (UDP-Lite) [6]. For these
profiles, the SN is generated at the compressor, as it is not present
in headers being compressed. For the least significant bit (LSB)
encoding method, the interpretation interval offset (p) is always
p = -1 (see section 5.1.1) when interpreting the SN. The SN is thus
required to increase for each packet received at the decompressor,
which means that reordered packets cannot be decompressed.

3.3. Profiles Incompatible with Reordering

The ROHC LLA profiles defined in RFC 3242 [7] and RFC 3408 [8] have
been explicitly designed with in-order delivery as a fundamental
requirement to their proper operation. Profiles 0x0005 and 0x0105 can
therefore not be implemented over channels where reordering can
occur; this document therefore does not apply to these profiles.

[4](#). **Background**

ROHC was designed with the assumption that packets are delivered in-order from compressor to decompressor. This was considered as a reasonable working assumption for links where it was expected that ROHC would be used. However, many have expressed that it would be desirable to use ROHC also over connections where in-order delivery is not guaranteed [[9](#)].

[4.1](#). **Reordering Channels**

The reordering channels that are potential candidates to use ROHC are single-hop channels and multi-hop virtual channels.

A single-hop channel is a point-to-point link that constitutes a single IP hop. Note that one IP hop could be one or multiple physical links. For example, a single-hop reordering channel could be a wireless link that applies error detection and performs retransmissions to guarantee error-free delivery of all data. Another example could be a wireless connection that performs bicasting of data during a handoff procedure.

A multi-hop virtual channel is a virtual point-to-point link that traverses multiple IP hops. A multi-hop virtual channel would typically be an IP tunnel, where compression is applied over the tunnel by the endpoints of the tunnel (not to be confused with single link compression of tunneled packets).

[4.2](#). **Robustness Principles of ROHC**

Robustness is based on the optimistic approach in the unidirectional and optimistic modes of operation (U/O-mode), and on the secure reference principle in the bi-directional reliable mode (R-mode).

Both approaches have different characteristics in the presence of reordering between compressor and decompressor. However, in any mode, decompression of sequentially early packets will generally be handled quite well since they will be perceived and treated by the decompressor as if there had been one or more packet losses.

[4.2.1](#). **Optimistic Approach (U/O-mode)**

A ROHC compressor uses the optimistic approach to reduce header overhead when performing context updates in U/O-mode. The compressor normally repeats the same update until it is fairly confident that the decompressor has successfully received the information. The

number of consecutive packets needed to obtain this confidence is
open to implementations, and this number is normally related to the

packet loss characteristics of the link where header compression is
used (see also [2], section 5.3.1.1.1).

All packet types used in U/O-mode are context updating.

4.2.2. **Secure Reference Principle (R-mode)**

A ROHC compressor uses the secure reference principle in R-mode, to
ensure that context synchronization between ROHC peers cannot be lost
due to packet losses. The compressor obtains its confidence that the
decompressor has successfully updated the context from a packet
carrying a 7- or 8-bit CRC based on acknowledgements received from
the decompressor (see also [2], section 5.5.1.2).

The secure reference principle makes it possible for a compressor to
use packets that do not update the context (i.e. R-0 and R-1* [2]).

5. **Problem Description**

5.1. **ROHC and Reordering Channels**

This section reviews different aspects of ROHC susceptible of being
impacted by reordering of compressed packets between ROHC peers.

5.1.1. **LSB Interpretation Interval and Reordering**

The LSB encoding method defined in RFC 3095 ([2], section 5.7)
specifies the interpretation interval offset, called p, as follow:

For profiles 0x0001, 0x0003 and 0x0007:

  $p = 1$, when bits(SN) <= 4;
  $p = 2^{(bits(SN)-5)} - 1$ otherwise.

  The resulting table describing the interpretation interval is:

bits (SN) k	Offset p (reordering)	(2^k-1) - p (losses)
4	1	7
5	0	16
6	1	31
7	3	61
8	7	121

```
|      9      |      15       |      241      |
+-----------+--------------+--------------+
```

As shown in the table above, the ability for ROHC to handle
sequentially late packets depends on the number of bits sent in
each packet. For example, a sequentially late packet of type 0
(with either 4 or 6 bits of SN) sets the limit to one packet out
of sequence for successful decompression to be possible.

For profiles 0x0002, 0x0004 and 0x0008:

p = - 1, independently of bits(SN).

A value of p = -1 means that the interpretation interval offset
can only take positive values, and that no sequentially late
packet can be decompressed if reordering occurs over the link.

The trade-off between reordering and robustness

The ability of ROHC to handle sequentially late packets is limited
by the interval interpretation offset of the sliding window used
for LSB encoding. This offset has a very small value for packets
with a small number of sequence number (SN) bits, but grows with
the number of SN bits transmitted.

For channels where both packet losses and reordering can occur,
modifications to the interpretation interval faces a trade-off
between the amount of reordering and the number of consecutive
packets losses that can be handled by the decompressor. If the
negative offset (i.e. p) is increased to handle a larger amount of
reordering, the value of the positive offset of the interpretation
interval must be decreased. This may impact the compression
efficiency when the channel has a high loss rate.

This is shown in the figure:

```
   <--- interpretation interval (size is 2^k) ---->
   |------------------+---------------------------|
 Lower              v_ref                       Upper
 Bound                                          Bound
   <--- reordering --> <--------- losses --------->
    max delta(SN) = p   max delta(SN) = (2^k-1) - p
```

    where v_ref is the reference value as per [2].

In practice, the maximum variation in SN value (max delta(SN))
due to reordering that can be handled will normally correspond to
the maximum number of packets that can be reordered. The same

applies to the maximum number of consecutive packet losses covered
by the robustness interval.

5.1.2. **Reordering of Packets in R-mode**

5.1.2.1. **Updating Packets**

   The compressor always adds references in the sliding window for all
   updating packets sent. The compressor removes values smaller than
   values for which it has received an acknowledgement, to shrink the
   window and thereby increase the compression efficiency.

   The decompressor always updates the context when receiving an
   updating packet, and uses the new reference for decompression.
   Acknowledgements are sent to allow the compressor to shrink its
   sliding window.

   Reordering between updating packets

      The decompressor can update its context from the reception of a
      sequentially late updating packet. The decompressor reference is
      then updated with a value that is no longer in the sliding window
      of the compressor. This "missing reference" can be caused by
      reordering when operating in R-mode.

      The result is that the compressor and the decompressor lose
      synchronization with each other. When the decompressor
      acknowledges the sequentially late packet, the compressor might
      already have discarded the reference to this sequence number, and
      continue to compress packets based on more recent references (in
      packet arrival time). Decompression will then be attempted using
      the wrong reference.

5.1.2.2. **Non-Updating Packets**

   Reordering between non-updating packets only

      A non-updating packet that reaches the decompressor out-of-
      sequence with respect to other non-updating packets only can
      always be decompressed properly.

   Reordering between non-updating packets and updating packets

      When a non-updating packet is reordered and becomes sequentially
      late with respect to an updating packet, the decompressor may have
      already updated the context with a new reference when the late
      packet is received. It is thus possible for a non-updating packet
      to be decompressed based on the wrong reference because of
      reordering when operating in R-mode.

      Since decompression of non-updating packets cannot be verified,

this can lead to a packet erroneously decompressed being forwarded
to upper layers.

5.1.3. Reordering of Packets in U/O-mode

Sequentially late packets

The ability to decompress sequentially late packets is limited by
the offset p of the interpretation interval (see section 5.1.1).
Decompression of a sequentially late packet with SN = x is
possible if the value of the SN of the packet that last updated
the context was less than or equal to x + p.

Problems occur if context(SN) has increased by more than p with
respect to field(SN) carried within the packet to decompress.

This means that for a well-behaved stream with a constant unit
increase in the RTP SN, a packet can arrive up to p packets out of
sequence and still be correctly decompressed. Otherwise, it cannot
be properly decompressed. It also means that if the compressor
sends two consecutive packets with SN(packet1)=100 and
SN(packet2)=108 when p=7, packet1 cannot be decompressed if it
arrives even one packet late due to reordering.

Decompression can always be verified since all U/O-mode packet types
are context updating. Consequently, reordering of packets is not
deemed problematic when operating in U/O-mode. For channels known to
reorder packets, the U/O-mode should therefore be the preferred mode
of operation. The additional risk of losing context synchronization
or for erroneous packet to be delivered to upper layers is limited.

5.1.4. Reordering on the Feedback Channel

For R-mode, upon reception of an acknowledgement, the compressor
searches the sliding window to locate an updating packet with the
corresponding SN; if it is not found, the acknowledgement is invalid
and is discarded ([2], section 5.5.1.2). In other words, feedback
received out-of-order either is still useful or is discarded.

In U/O-mode, if the compressor updates its context based on feedback,
the same logic as for R-mode applies in practice.

Reordering on the feedback channel has thus no impact in either mode.

5.1.5. List Compression

```
<# Editor's Note:    This is for further study.                #>
<#                                                             #>
<#                                                             #>
<#                                                             #>
```

5.1.6. Reordering and Mode Transitions

   Transition from U/O-mode to R-mode

      This transition can be affected by reordering if a packet type 0
      (UO-0) is reordered and delayed by at least one round-trip time
      (RTT). If the decompressor initiates a mode change request to R-
      mode in the meantime, the reordered UO-0 packet may be handled as
      an R-0 packet; it can be erroneously decompressed and forwarded to
      upper layers. This is because the decompressor can switch to
      R-Mode as soon as it sends the acknowledgement Ack(SN, R) to the
      compressor (see also [2], section 5.6).

   Transition from R-mode to U/O-mode

      A similar situation as above can occur during this transition.
      However, because the outcome of the decompression is always
      verified using a CRC verification in U/O-mode, the reordered
      packet will most likely fail decompression and will be discarded.

   The above situation, while it is not deemed to occur frequently, is
   still possible; thus mode transitions from U/O-mode to R-mode should
   be avoided when reordering can occur.

5.2. Consequences of Reordering

   The context updating properties of the packets exchanged between ROHC
   peers are the most important factors to consider when deriving the
   impacts of reordering. For this reason, the robustness properties of
   the U/O-mode and of the R-mode are affected differently.

   The effects of reordering on ROHC can be summarized as follow:

   - Functionality incompatible with reordering;
   - Increased probability of context damage (loss of synchronization);
   - Increased number of decompression failures - Detected (U/O/R-mode);
   - Increased number of decompression failures - Undetected (R-mode).

5.2.1. Functionality Incompatible with Reordering

   There are some optional ROHC functions that cannot work in the
   presence of reordering between ROHC peers.

   The ROHC segmentation scheme (see [2], section 5.2.5) relies entirely
   on the in-order delivery of each segment, as there is no sequencing
   information in the segments. Therefore segmentation should not be
   used if there can be reordering between the ROHC peers.

Timer-based compression of RTP TS (see [2], section 4.5.4) is built
on an assumption of timely (minimal jitter) delivery. Therefore it
should be used with care over links where reordering can occur, with
respect to the amount of jitter that can be introduced by reordering.

The use of these optional features is open to implementations and is
local to the compressor only; it does not impact the decompressor.

5.2.2. Context Damage (Loss of Synchronization)

Reordering of packets between ROHC peers can impact the robustness
properties of the optimistic approach (U/O-mode) as well as the
reliability of the secure reference principle (R-mode).

The successful decompression of a sequentially late change packet
(U/O-mode) and/or updating packet (R-mode) can update the context of
the decompressor in a manner unexpected by the compressor. This can
lead to a loss of context synchronization between the ROHC peers.

5.2.3. Detected Decompression Failures (U/O/R-mode)

Reordering of packets between ROHC peers can lead to an increase in
the number of decompression failures for context updating packets
(see sections 5.1.2.1 and 5.1.3). Fortunately, as the outcome of the
decompression of updating packets can be verified, the decompressor
can reliably detect decompression failures caused by reordering and
discard the packet. Note that local repairs, subject to the
limitations stated in [2] section 5.3.2.3, can still be performed.

5.2.4. Undetected Decompression Failures (R-mode only)

Reordering of packets between ROHC peers can lead to an increase in
the number of decompression errors for non-updating packets. For R-
mode, decompression of R-0 and R-1* packets cannot be verified. If
reordering occurs and decompression is performed using the wrong
secure reference (see section 5.1.2.1 and 5.1.2.2), the decompressor
cannot reliably detect such errors. As a result, erroneous packets
may be forwarded to upper layers.

6. **Making ROHC Tolerant against Reordering**

   This chapter describes different approaches that can improve the
   performance of ROHC when used over reordering channels and minimize
   the effects of reordering. Examples are provided to guide
   implementers and designers of new profiles. The solutions target
   either the properties of ROHC implementations or the specifications
   of profiles. This is covered by sections 6.1 and 6.2 respectively.

6.1. **Properties of ROHC Implementations**

   Existing ROHC profiles can be implemented with the capability to
   properly handle packet reordering. The methods described in this
   section conform with, and thus do not require any modifications to,
   the ROHC specifications within scope of this document (see section
   3). Specifically, the methods presented in this section can be
   implemented without any impairment to interoperability with other
   ROHC implementations that do not use these methods.

   The methods suggested here may however lower compression efficiency,
   and these modifications should not be used when reordering is known
   not to occur. Some of these methods aim to increase the decompression
   success rate at the decompressor, while others aim to avoid context
   damages causing loss of context synchronization between compressor
   and decompressor.

   The methods proposed are each addressing specific issues listed in
   section 5, and can be combined to achieve better robustness against
   reordering.

6.1.1. **Compressing Headers with Robustness against Reordering**

   The methods described in this section are methods local only to the
   compressor implementation. They can be used without modifications or
   impact to the decompressor.

6.1.1.1. **Reordering and the Optimistic Approach**

   The optimistic approach is affected by the reordering characteristics
   of the channel when operating over a reordering channel. Compressor
   implementations should therefore adjust their optimistic approach
   strategy to match both packet loss and reordering characteristics.

   For example, the number of repetitions for each context update can be
   increased. The compressor should ensure that each update is repeated
   until it is reasonably confident that a least one change packet in

the sequence of repetitions has reached the decompressor before the
first packet sent after this sequence.

6.1.1.2. **Reordering and the Secure Reference Principle**

   Fundamental to the secure reference principle is that only values
   acknowledged by the decompressor can be used as reference for
   compression. In addition, some of the packet types used in R-mode do
   not include a CRC over the original uncompressed header, and the
   decompressor has no means to verify the outcome of the decompression.

   Decompression of non-updating packet types thus entirely relies on
   the cumulative effect of previous updates to the secure reference,
   and the compressed data is based on the current value of the
   reference. This reference must be synchronized between ROHC peers.
   For R-0 and R-1* packets, the reception of the encoded bits applied
   to the secure reference is sufficient for correct decompression, but
   only when in-order delivery between ROHC peers is guaranteed.

   Avoiding the "missing reference" problem (section 5.1.2.1)

      A compressor implementation can delay the advance in the sliding
      window to a reference acknowledged by the decompressor, until it
      has confidence that no acknowledgement for any of the values that
      could be discarded can be received. This confidence can be based
      on the maximum delay that reordering can introduce over the
      channel. It can also be based on the knowledge that the
      decompressor implements the context updating logic of section
      6.1.2.1 (e.g. by means of standardization).

6.1.1.3. **Robust Selection of Compressed Header**

   The interpretation interval for the LSB encoded sequence number can
   be adjusted to allow for larger negative offsets (see section 5.1.1).
   This would provide the capability to decompress sequentially late
   packets with a greater amount of reordering.

   To achieve this, the compressor should be implemented conservatively
   in terms of the choice of packet types to send, by transmitting
   packets with more sequence number bits. As shown in the table of
   section 5.1.1, using eight bits of SN allows a packet to be
   decompressed when the reordering leads to up to seven units in
   sequence number variation (i.e. delta(SN)). Increasing the number of
   SN bits (i.e. using a larger SN_k [2]) transmitted will make ROHC
   even more tolerant to reordering.

For example, a conservative compressor implementation could use the
packet types as shown in the table below:

```
+----------------------+------------------------+
| Optimal Packet Type  | Alternative Packet Type |
| (without reordering) |  (reordering possible)  |
+----------------------+------------------------+
| UO-0                 | UOR-2*-ext0            |
| R-0                  | R-1*-ext0             |
| R-0-CRC              | UOR-2*-ext0            |
| R-1*                 | R-1*-ext0             |
| UO-1                 | UOR-2-ext0            |
| UO-1-TS              | UOR-2-TS-ext0         |
| UO-1-ID              | UO-1-ID-ext3 (with S=1) |
|                      | UOR-2-ID-ext0         |
| UOR-2*               | UOR-2*-ext0            |
+----------------------+------------------------+
```

Such a compressor implementation would thus always be sending at
least 3 octets (R-mode) or 4 octets (U/O-mode). This is a trade-off
when compared to the 1 octet that can be sent by a more aggressive
implementation operating on a channel with no reordering.

Note that since the interpretation interval for profiles 0x0002,
0x0004 and 0x0008 is always p = -1 independently of bits(SN), the
methods suggested in this section will not work for these profiles
unless this value is modified (section 6.2.1).

6.1.2. **Implementing a Reordering Tolerant Decompressor**

The methods described in this section are methods local only to the
decompressor implementation. They can be used without modifications
or impact to the compressor.

6.1.2.1. **Bi-directional Reliable Mode (R-mode)**

The "missing reference" problem described in section 5.1.2.1 can be
avoided. If the decompressor can detect when two updating packets
(packets including CRCs) are reordered with respect to each other, it
should not update the context with the values of the sequentially
late update packet.

6.1.2.2. **Decompressor Feedback Considerations**

Reducing the feedback rate when the flow behaves linearly

The decompressor should reduce its feedback rate when a large
number of UOR-2 packets with extensions a received, when the flow
behaves linearly (i.e. when only fields pertaining to the
functions established with respect to the sequence number are
changing.

In particular, if the compressor implementation makes a more
conservative selection of packet types (section 6.1.1.3) in order
to handle reordering, the decompressor should try to avoid sending
more feedback than it would have if the more optimal packet types
had been used.

Note that if the decompressor does not make this adjustment,
packet losses or context damages will not increase. It might
however reduce link efficiency.

Acknowledgements and sequentially late packets

Reordered feedback (or feedback for packets received out-of-order)
will not cause problems (see section 5.1.4). However, the
decompressor should not send feedback for sequentially late
packets, as the current state of the context will better reflect
the compressor context than the content of the reordered packet.

6.1.2.3. **Considerations for Local Repair Mechanisms**

When decompression fails, and if reordering can be the cause of this
failure, a local repair may be attempted for the sequentially late
packet by going backward in the interpretation interval (as opposed
to moving forward as for packet losses).

6.2. **Specifying ROHC Profiles with Robustness against Reordering**

6.2.1. **Profiles with Interpretation Interval Offset p = -1**

New revisions of profiles 0x0002 (UDP) [2], 0x0004 (IP-only) [5], and
0x0008 (UDP-Lite) [6] should redefine how the value of the offset p
is determined, and use the same algorithm as in profile 0x0001 [2]
instead of p = -1 independently of bits(SN) (section 5.1.1).

While such a change would make these updated profiles slightly less
robust to packet losses, they would still be no less robust than

profile 0x0001.

6.2.2. Modifying the Interpretation Interval Offset

   The interpretation interval offset p could be modified for existing
   profile in order to handle reordering while improving the compression
   efficiency when compared to the solution of section 6.1.1.3.

6.2.2.1. Example profile for handling reordering

   The value of the interpretation interval offset p can be adjusted to
   achieve a robustness against reordering similar to the effect of
   selecting packet types as suggested in section 6.1.1.3.

   For example, assuming that having a value p=7 is enough while still
   considering robustness against packet losses a priority, a ratio
   where the positive offset is about twice as large as the negative
   offset can be used. This leaves a value of p = 2^k/ 3.

   The resulting values are shown in the following table:

```
            +-----------+--------------+-----------------+
            | bits (SN) |   Offset p   | Positive range  |
            |     k     | (reordering) |     (losses)    |
            +-----------+--------------+-----------------+
            |     4     |       5      |        10       |
            |     5     |      10      |        21       |
            |     6     |      21      |        42       |
            |     7     |      42      |        85       |
            |     8     |      85      |       170       |
            |     9     |     170      |       341       |
            +-----------+--------------+-----------------+
```

   Using this value for p, a fair amount of reordering can be handled
   without having to send UOR-2 packets most of the time. The trade-off
   is that this is at the expense of robustness against packet losses.

6.2.2.2. Defining the values of p for new profiles

   As described in RFC3095, the interpretation interval when sending k
   bits of SN is defined as:

      f(v_ref, k) = [v_ref - p, v_ref + (2^k - 1) - p]

   The negative bound (v_ref - p) limits the ability to handle
   reordering, while the positive bound (v_ref + (2^k - 1) - p) limits
   the ability to handle packet losses.

   Adjusting p will increase one of these ranges, while the other range

will decrease. When designing ROHC profiles, considerations on how
these correlate with each other should be taken.

For example, if it is desirable for a profile to be as robust against
reordering (negative range) and against packet losses (positive
range), these range can be made equal by setting p near (2^k / 2).

6.2.3. TCP Profile Considerations

The current draft for the ROHC TCP profile [4] contains packet
formats that allow sending as little as 1 bit of MSN (master sequence
number). Since the MSN is used in the same fashion as the sequence
number in profile 0x0002, it will not be possible to decompress
reordered packets if used over a reordering channel.

The work on the ROHC-TCP profile should consider using more bits of
MSN to enable simple implementation modifications when operating over
a reordering channel.

7. Security Consideration

This document does not include additional security risks to [2]. In
addition, it may lower risks related to context damage in R-mode with
injected packets when sequentially late packets do not update the
context (section 6.1.2.1).

8. IANA Considerations

This document does not require any IANA action.

9. Acknowledgments

The authors would appreciate feedback on this document, as input from
others would certainly help us improve it significantly.

10. Authors' Addresses

Ghyslain Pelletier   Tel: +46 920 20 24 32
Ericsson AB   EMail: ghyslain.pelletier@ericsson.com
Box 920
S-971 28 Lulea
Sweden

Lars-Erik Jonsson   Tel: +46 920 20 21 07
Ericsson AB   EMail: lars-erik.jonsson@ericsson.com
Box 920
S-971 28 Lulea

Sweden

Kristofer Sandlund   Tel: +46 920 609 17
Effnet AB   EMail: kristofer.sandlund@effnet.com
Stationsgatan 69
S-972 34 Lulea
Sweden

11. Informative References

[1]   S. Bradner, "Key words for use in RFCs to Indicate Requirement
      Levels", RFC 2119, March 1997.

[2]   C. Bormann, et. al, "RObust Header Compression (ROHC): Framework
      and four profiles: RTP, UDP, ESP, and uncompressed", RFC 3095,
      July 2001.

[3]   Jonsson, L-E., " RObust Header Compression (ROHC): Terminology
      and Channel Mapping Examples", RFC 3759, April 2004.

[4]   G. Pelletier, et. al, "RObust Header Compression (ROHC): TCP/IP
      Profile (ROHC-TCP)", Internet-Draft (work in progress),
      <draft-ietf-rohc-tcp-06.txt>, April 2004.

[5]   Jonsson, L-E. and G. Pelletier, "RObust Header Compression
      (ROHC): A compression profile for IP", Internet draft (work in
      progress), <draft-ietf-rohc-ip-only-05.txt>, October 2003.

[6]   G. Pelletier, "RObust Header Compression (ROHC): Profiles for
      UDP-Lite", Internet draft (work in progress), <draft-ietf-rohc-
      udp-lite-04.txt>, June 2004.

[7]   Jonsson, L-E. and G. Pelletier, "RObust Header Compression
      (ROHC): A Link-Layer Assisted Profile for IP/UDP/RTP", RFC 3242,
      April 2002.

[8]   Liu, Z. and K. Le, "Zero-byte Support for Bidirectional Reliable
      More (R-mode) in Extended Link-Layer Assisted Profile for RObust
      Header Compression (ROHC) Profile", RFC 3408, December 2002.

[9]   Ash, J., Goode B. and J. Hand, "Requirements for Header
      Compression over MPLS", Internet draft (work in progress),
      <draft-ietf-avt-hc-mpls-reqs-02.txt>, June 2004.

This Internet-Draft expires December 14, 2004.