Network Working Group
Request for Comments: 4164
Category: Standards Track
G. Pelletier
Ericsson
August 2005

RObust Header Compression (ROHC): Context Replication for ROHC Profiles

Status of This Memo

Copyright Notice

Abstract

This document defines context replication, a complement to the context initialization procedure found in Robust Header Compression (ROHC), as specified in RFC 3095. Profiles defining support for context replication may use the mechanism described herein to establish a new context based on another already existing context. Context replication is introduced to reduce the overhead of the context establishment procedure. It may be especially useful for the compression of multiple short-lived flows that may be occurring simultaneously or near-simultaneously, such as short-lived TCP flows.

Table of Contents

1. Introduction

There is often some redundancy between header fields of different flows that pass through the same compressor-decompressor pair. This means that some of the information needed to initialize the context for decompressing the headers of a new flow may already be present at the decompressor. It may be desirable to reuse this information and remove some of the overhead normally required for the initialization of a new header compression context at both the compressor and decompressor.

Reducing the overhead of the context establishment procedure is particularly useful when multiple short-lived connections (or flows) occur simultaneously, or near-simultaneously, between the same compressor-decompressor pair. Because each new packet stream requires most of the header information to be sent during the initialization phase before smaller compressed headers can be used, a multitude of short-lived connections may significantly reduce the overall gain from header compression.

Context replication allows some header fields, such as the IP source and/or destination addresses (16 octets each for IPv6), to be omitted within the special Initiation and Refresh (IR) packet type specifically defined for replication. It also allows other fields, such as source and/or destination ports, to be either omitted or sent in a compressed form from the very first packet of the header compressed flow.

Context replication is herein defined as a general ROHC mechanism. The benefits of context replication are not limited to any particular protocol and its support may be defined for any ROHC profile.

In particular, context replication is applicable to TCP compression because many TCP transfers are short-lived; a behavior analysis of TCP/IP header fields among multiple short-lived connections may be found in [5]. In addition, [4] introduces considerations and requirements for the ROHC-TCP profile [3] to efficiently compress such short-lived TCP transfers.

For profiles supporting this mechanism, the compressor performs context replication by reusing or creating a copy of an existing context, i.e., a base context, to create the replicated context. The replicated context is then updated to match the header fields of the new flow. The compressor then sends to the decompressor a packet that contains a reference to the selected base context, along with some data for the fields that need to be updated when creating the replicated context. Finally, the decompressor creates the replicated context based on the reference to the base context along with the uncompressed and compressed data from the received packet.

This document specifies the context replication procedure for ROHC profiles. It defines the general compressor and decompressor logic used during context replication, as well as the general format of the special IR packet required for this procedure. Profiles defining support for context replication must further specify the specific format(s) of this packet.

The fundamentals of the ROHC framework may be found in [2]. It is assumed throughout this document that these are understood.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [1].

This document reuses some of the terminology found in [2]. In addition, this document defines the following terms:

Base context

A base context is a context that has been validated by both the compressor and the decompressor. The compressor can use a base context as the reference when building a new context using replication.

Base CID (BCID)

The Base Context Identifier is the CID used to identify the base context, from which information needed for context replication can be extracted.

Context replication

Context replication is the mechanism that initializes a new context based on another already existing context (a base context).

3. Context Replication for ROHC Profiles

For profiles defining its support, context replication may be used as an alternative to the context initialization procedure found in [2]. Note that for such profiles, only the decompressor is mandated to support context replication; the use of the IR-CR packet is optional for the compressor.

This section describes the compressor and decompressor logic as well as the general format of the IR packet used with context replication.

3.1. Robustness Considerations

Context replication deviates from the initialization procedure defined in [2] in that it is able to achieve a certain level of compression from the first packet used to initialize the context for a new flow. Therefore, it is of particular importance that the context replication procedure be robust. This requires that a base context suitable for replication be used, that the integrity of the initialization packet be guaranteed, and finally that the outcome of the replication process be verified.

The primary mechanisms used to achieve robustness of the context replication procedure are the selection of the base context (based on prior feedback from the decompressor) and the use of checksums. Specifically, the compressor must obtain enough confidence that the base context selected for replication is valid and available at the decompressor before initiating the replication procedure. Thus, the most reliable way to select the base context is to choose a context for which at least the static part to be replicated has previously been acknowledged by the decompressor.

In addition, the presence of a CRC covering the information that initializes the context ensures the integrity of the IR header used for replication. Finally, an additional CRC calculated over the original uncompressed header allows the decompressor to validate the reconstructed header and the outcome of the replication process.

3.2. Replication of Control Fields

Control fields are fields that are either transmitted from a ROHC compressor to a ROHC decompressor or inferred based on the behavior of other fields, but are not part of the uncompressed header itself.

They can be used to control compression and decompression behavior, in particular, the set of packet formats to be used. Control fields are profile-specific. Examples of such fields include the NBO and RND flags [2], which indicate whether the IP-ID field is in Network

Byte Order and the type of behavior of the field, respectively. Another example is the parameter indicating the mode of operation [2].

The IR-CR differs from the IR packet [2] in that its purpose is to entirely specify what part of the base context is replicated, and to convey the complementary information needed to create a new context. Because of this, a profile supporting the use of the IR-CR packet SHOULD define for each control field if the value of the field is replicated from the base context to the new context, or if its value is reinitialized.

In addition, a compressor MUST NOT initiate context replication while a control field that is not reinitialized by replication is being updated, e.g., during the handshake for a mode transition [2].

3.3. Compressor States and Logic

Compression with ROHC normally starts in the IR state, where IR packets must be sent to initialize a new context at the decompressor. IR packets include all static and non-static fields of the original header in uncompressed form plus some additional information. The compressor stays in the IR state until it obtains confidence that the decompressor has received the information.

Context replication provides an optional mechanism to complement the ROHC initialization procedure. It defines a packet type, the IR packet for Context Replication (IR-CR), which can be used to initialize a new context. Consequently, the Context Replication (CR) state is introduced to the compressor state machine to encompass the additional logic required for the use of the IR-CR packet.

For profiles defining support for context replication, the compressor may thus transit directly from the IR state to the CR state if an already existing context can be selected as a base context for replication. This effectively replaces any IR/IR-DYN packets sent during the context establishment procedure with an IR-CR packet.

3.3.1. Context Replication (CR) State

The purpose of the CR state is to initialize a new context by reusing an already existing context. In this state, the compressor sends a combination of uncompressed and compressed information, along with a reference to a base context plus some additional information. Therefore, header information pertaining to fields that are being replicated is not sent.

The compressor stays in the CR state until it is confident that the decompressor has received the replication information correctly.

3.3.2. State Machine with Context Replication

The compressor always starts in the lower compression state (IR), and transits to the context replication state (CR) under the constraint that the compressor can select a base context that is suitable for the flow being compressed (see also Section 3.3.3.1).

The transition from the CR state to a higher compression state (e.g., the CO state for [3]) is based on the optimistic approach principle or feedback received from the decompressor.

The figure below shows the additional state for the compressor. The details of the state transitions and compression logic are given in sub-sections following the figure.

```
              BCID selection       Optimistic approach / ACK
           +----->----->------+    +----->----->----->-----+
           |                  |    |                       |
           |                  v    |                       v
      +---------+          +---------+             +-------------+
      |   IR    |          |   CR    |             |   Higher    |
      |  state  |          |  state  |             | order state |
      +---------+          +---------+             +-------------+
           ^                    |
           | NACK / STATIC-NACK |
           +---<-----<-----<----+
```

Note that context replication is a complement to the normal initialization procedure for ROHC profiles that support it. Therefore, the compressor transition to the CR state is an optional addition to the state machine, and does not affect already existing transitions between the IR state and higher order state(s).

3.3.3. State Transition Logic

Decisions about transition to and from the CR state are taken by the compressor on the basis of:

- availability of a base context
- positive feedback from the decompressor (Acknowledgements -- ACKs)
- negative feedback from the decompressor (Negative ACKs -- NACKs)
- confidence level regarding error-free decompression of a packet

Context replication is designed to operate over links where a feedback channel is available. This is necessary to ensure that the information used to create a new context is synchronized between the compressor and the decompressor. In addition, context replication may also make use of feedback from decompressor to compressor for transition back to the IR state and for OPTIONAL improved forward transition towards a state with a higher compression ratio.

The format that must be used by all profiles for the feedback field within the general ROHC format is specified in Section 5.2.2 of [2]; the feedback information is structured using two possible formats: FEEDBACK-1 and FEEDBACK-2. In particular, FEEDBACK-2 can carry one of three possible types of feedback information: ACK, NACK, or STATIC-NACK.

3.3.3.1. Selection of Base Context, Upward Transition

The compressor may initiate a transition from the IR state to the CR state when a suitable base context can be identified. To perform this transition, the compressor selects a context that has previously been acknowledged by the decompressor as the base context. The selected context MUST have been acknowledged by the decompressor using the CRC option (see also [2], Section 5.7.6.3) in the feedback message. The static part of the base context to be replicated MUST have been acknowledged by the decompressor and the base context MUST be valid at replication time.

This also implies that a compressor is not allowed to use the context replication mechanism if a feedback channel is not present. However, note that the presence of the feedback channel cannot provide the guarantee that a base context selected for replication has not been corrupted after it has been acknowledged, or that it is still part of the state managed by the decompressor when the IR-CR will be received.

More specifically, RFC 3095 [2] defines the context identifier (CID) as a reference to the state information (i.e., the context) used for compression and decompression. Multiple packet streams, each having its own context, may thus share a channel; and the CID space along with its representation within packet formats may be negotiated as part of the channel state. However, because RFC 3095 [2] does not explicitly define context state management between compressor and decompressor, in particular for connection-oriented flows (e.g., TCP), no more than a high degree of confidence can be achieved when selecting a base context.

In the case where feedback is not used by the decompressor, the compressor may have to periodically transit back to the IR state. In such a case, the same logic applies for the transition back to the higher order state via the CR state: a base context, previously acknowledged and suitable for replication, must be re-selected.

The criteria for whether an existing context is a suitable base context for replication for a new flow are left to implementations.

Whenever the sequencing information from the last acknowledgement received is available, the compressor MAY use it to determine what fields can be replicated to avoid replicating any fields that have changed significantly from the state corresponding to the acknowledged packet.

3.3.3.2. Optimistic Approach, Upward Transition

Transition to a higher order state can be carried out according to the optimistic approach principle. This means that the compressor may perform an upward state transition when it is fairly confident that the decompressor has received enough information to correctly decompress packets sent according to the higher compression state.

In general, there are many approaches where the compressor can obtain such information. The compressor may obtain its confidence by sending several IR-CR packets with the same information.

3.3.3.3. Optional Acknowledgements (ACKs), Upward Transition

An ACK may be sent by the decompressor to indicate that a context has been successfully initialized during context replication.

Upon reception of an ACK, the compressor may assume that the context replication procedure was successful and transit from its initial state (e.g., IR state) to a higher compression state.

3.3.3.4. Negative ACKs (NACKs), Downward Transition

A STATIC-NACK sent by the decompressor may indicate that the decompressor could not initialize a valid context during context replication, and that the corresponding context has been invalidated.

Upon reception of a STATIC-NACK, the compressor MUST transit back to its initial no context state. The compressor SHOULD also refrain from sending IR-CR packets using the same base context, at least until an acknowledgement subsequent to the reception of the

STATIC-NACK makes this context suitable for replication (Section 3.3.3.1). The compressor SHOULD re-initialize the decompressor context using an IR packet.

A NACK sent by the decompressor may indicate that a valid context has been successfully initialized but that the decompression of one or more subsequent packets has failed.

Upon reception of a NACK, the compressor MAY assume that the static part of the decompressor context is valid, but that the dynamic part is invalid; the compressor may take actions accordingly.

3.4. Decompressor Logic

3.4.1. Replication and Context Initialization

Upon reception of an IR-CR packet, the decompressor first determines its content ([2], Section 5.2.6). The profile indicated in the IR-CR packet determines how it is to be processed. If the CRC (8-bit CRC) fails to verify the packet, the packet MUST be discarded.

If the profile as indicated in the IR-CR packet defines the use of the Base CID, and if its corresponding field is present within the packet format, this field is used to identify the base context; otherwise, the CID is used.

3.4.2. Reconstruction and Verification

The decompressor creates a new context using the information present in the IR-CR packet together with the identified base context, and decompresses the original header.

The CRC calculated over the original uncompressed header and carried within the profile-specific part of the IR-CR headers (7-bit CRC) MUST be used to verify decompression.

When the decompression is verified and successful, the decompressor initializes or updates the context with the information received in the current header. The decompressor SHOULD send an ACK when it successfully validates the context as a result of the decompression of one or more IR-CR packets.

Otherwise, if the reconstructed header fails the CRC check, changes (either initialization or update) to the context MUST NOT be performed. When the decompressor fails to validate the header, actions as specified in Section 3.4.3 are taken.

3.4.3. Actions upon Failure

For profiles supporting context replication, the feedback logic of a decompressor is similar to the logic used for context initialization, as described in [2].

Specifically, when the decompressor fails to validate the context following the decompression of one or more initial IR-CR packets, it MUST invalidate the context and remain in its initial state. In addition, the decompressor SHOULD send a STATIC-NACK. In particular, a decompressor implementation performing strict memory management, such as deleting context state information when a connection-oriented flow (e.g., TCP) is known to have terminated, SHOULD send STATIC-NACK in this case. Otherwise, there is a risk that the compressor will maintain a specific CID as a potential candidate for a later replication attempt, while actually there is insufficient state left in the decompressor for this CID to act as a Base CID.

If the context has been successfully validated from the decompression of one or more initial IR-CR packets, the decompressor SHOULD send a NACK when it fails to verify the context following the decompression of one or more subsequent IR-CR packets.

3.4.4. Feedback Logic

The decompressor SHOULD use the CRC option (see [2], Section 5.7.6.3) when sending feedback corresponding to an IR or an IR-CR packet.

3.5. Packet Formats

The format of the IR-CR packet has been designed under the following constraints:

a) it must be possible to either overwrite a CID during context replication, or to use a different CID than the Base CID for the replicated context;
b) it must be possible to selectively include or exclude from the packet format some fields that may be replicable;
c) it must be possible for some fields that may be replicable to be represented within the packet format using either a compressed or an uncompressed form;
d) it must be possible for the decompressor to verify the success of the replication procedure;
e) it is anticipated that profiles, other than ROHC-TCP [3], will also define support for context replication. Therefore it is desirable that the packet format be profile independent.

3.5.1. CRCs in the IR-CR Packet

The IR packet, as defined in [2], is used to communicate static and/or dynamic parts of a context, and typically initialize the context. For example, the static and dynamic chains of IR packets may contain an uncompressed representation of the original header.

The IR packet format includes an 8-bit CRC, calculated over the initial part of the IR packet. This CRC is meant to protect any information that initializes the context. In particular, its coverage always includes any CID information as well as the profile used to interpret the remainder of the IR packet.

The purpose of the 8-bit CRC is to ensure the integrity of the IR header itself. Profiles may extend the coverage of this CRC to include the entire IR header, thus allowing the verification of the integrity of the entire uncompressed header. However, because the format of the IR packet is common to all ROHC profiles and verified as part of the initial processing of a ROHC decompressor (see [2], Section 5.2.6.), profiles may not redefine this CRC beyond the extent of its coverage.

RFC 3095 [2] also defines a 3-bit CRC and a 7-bit CRC for compressed headers, used to verify proper decompression and validate the context. This type of CRC is calculated over the original uncompressed header, as it is not sufficient to protect only the compressed data being exchanged between compressor and decompressor for the purpose of ensuring a robust reconstruction of the original header.

Thus, there is a clear distinction in purpose between the 8-bit CRC found in the IR packet and the 3-bit or 7-bit CRC found in compressed headers. With context replication, where the IR-CR packet may contain both compressed as well as uncompressed information and omit entirely replicable fields, this distinction in no longer present.

Profiles supporting context replication MUST define a CRC over the original uncompressed header as part of the profile-specific information in the IR-CR packet. This is necessary to allow a decompressor to verify that the replication process has succeeded.

3.5.1.1. 7-bit CRC

The 7-bit CRC in the IR-CR packet is calculated over all octets of the entire original header, before replication, in the same manner as described in Section 5.9.2 of [2].

The initial content of the CRC register is to be preset to all 1's. The CRC polynomial used for the 7-bit CRC in the IR-CR is:

$$C(x) = 1 + x + x^2 + x^3 + x^6 + x^7$$

3.5.1.2. 8-bit CRC

The coverage of the 8-bit CRC in the IR-CR packet is not profile dependent, as opposed to the ROHC IR(-DYN) packet (see [2], Sections 5.2.3 and 5.2.4). It MUST cover the entire packet, excluding the payload. In particular, this includes the CID or any add-CID octet as well as the Base CID field, if present. For profiles that define the usage of the Base CID within the packet format of the IR-CR as optional, this CRC MUST also cover the information used to indicate the presence of this field within the packet.

The initial content of the CRC register is to be preset to all 1's. The CRC polynomial used for the 8-bit CRC in the IR-CR is:

$$C(x) = 1 + x + x^2 + x^8$$

3.5.2. General Format of the IR-CR Packet

The context replication mechanism requires a dedicated IR packet format that uniquely identifies the IR-CR packet. This packet communicates the static and the dynamic parts of the replicated context. It may also communicate a reference to a base context.

With consideration to the extensibility of the IR packet type defined in [2], support for replication can be added using the profile-specific part of the IR packet. Note that there is one bit, (x), left in the IR header for "Profile specific information". The definition of this bit is profile specific. Thus, profiles supporting context replication MAY use this bit as a flag indicating whether the packet is an IR packet or an IR-CR packet. Note also that profiles may define an alternative method to identify the IR-CR packet within the profile-specific information, instead of using this bit.

The IR-CR header associates a CID with a profile, and initializes the context using the context replication mechanism. It is not recommended to use this packet to repair a damaged context.

The IR-CR has the following general format:

```
  0   1   2   3   4   5   6   7
 --- --- --- --- --- --- --- ---
:         Add-CID octet         : if for small CIDs and (CID != 0)
+---+---+---+---+---+---+---+---+
| 1   1   1   1   1   1   0   x | IR type octet
+---+---+---+---+---+---+---+---+
:                               :
/      0-2 octets of CID        / 1-2 octets if for large CIDs
:                               :
+---+---+---+---+---+---+---+---+
|            Profile            | 1 octet
+---+---+---+---+---+---+---+---+
|              CRC              | 1 octet
+---+---+---+---+---+---+---+---+
|                               |
/ Profile-specific information  / variable length
|                               |
 - - - - - - - - - - - - - - - -
|                               |
/           Payload             / variable length
|                               |
 - - - - - - - - - - - - - - - -
```

x: Profile-specific information. Interpreted according to the profile indicated in the Profile field.

Profile: The profile to be associated with the CID. In the IR-CR packet, the profile identifier is abbreviated to the 8 least significant bits (LSBs). It selects the highest-number profile in the channel state parameter PROFILES that matches the 8 LSBs given (see also [2]).

CRC: 8-bit CRC computed using the polynomial of Section 3.5.1.2.

Profile-specific information: The contents of this part of the IR-CR packet are defined by the individual profiles. This information is interpreted according to the profile indicated in the Profile field. It MUST include a 7-bit CRC over the original uncompressed header using the polynomial of Section 3.5.1.1. It also includes the static and dynamic subheader information used for replication; thus, which header fields are replicated and their respective encoding methods are outside the scope of this document.

Payload: The payload of the corresponding original packet, if any.

3.5.3. Properties of the Base Context Identifier (BCID)

The Base CID within the packet format of the IR-CR may be assigned a different value than the context identifier associated with the new flow (i.e., BCID != CID); otherwise, the base context is overwritten with the new context by the replication process.

When the channel uses small CIDs, a four-bit field within the packet format of the IR-CR minimally represents the BCID with a value from 0 to 15. In particular, the four bits of Add-CID used with small CIDs [2] are not needed for the BCID, as this information is already provided by the CID of the IR-CR packet itself. When large CIDs are used, the BCID is represented in the IR-CR with one or two octets, and it is coded in the same way as a large CID [2].

4. Security Considerations

This document adds an alternative mechanism for ROHC profiles to increase the compression efficiency when initializing a new context, by reusing information already existing at the decompressor. This is achieved by introducing new state transition logic, new feedback logic, and a new packet type -- all based on logic and packet formats already defined in RFC 3095 [2].

In this respect, this document is not believed to bring any additional weakness to potential attacks to those already listed in [2]. However, it does increase the potential impacts of these attacks by creating dependencies between multiple contexts. Specifically, corruption of one context can fail compressor attempts to initialize another context at the decompressor, or to propagate to another context, if the compressor uses a corrupted context as a base for replication.

5. Acknowledgements

The author would like to thank Richard Price, Kristofer Sandlund, Fredrik Lindstroem, Zhigang Liu, and HongBin Liao for valuable input, as well as Mark West and Lars-Erik Jonsson who also served as committed working group document reviewers.

6. References

6.1. Normative References

[1] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[2] Bormann, C., Burmeister, C., Degermark, M., Fukushima, H., Hannu, H., Jonsson, L-E., Hakenberg, R., Koren, T., Le, K., Liu, Z., Martensson, A., Miyazaki, A., Svanbro, K., Wiebke, T., Yoshimura, T., and H. Zheng, "RObust Header Compression (ROHC): Framework and four profiles: RTP, UDP, ESP, and uncompressed", RFC 3095, July 2001.

6.2. Informative References

[3] Pelletier, G., Jonsson, L-E., Sandlund, K., and M. West, "RObust Header Compression (ROHC): A Profile for TCP/IP (ROHC-TCP)", Work in Progress, July 2005.

[4] Jonsson, L-E., "RObust Header Compression (ROHC): Requirements on TCP/IP Header Compression", RFC 4163, August 2005.

[5] West, M. and S. McCann, "TCP/IP Field Behavior", Work in Progress, October 2004.

[6] Finking, R. and G. Pelletier, "Formal Notation for Robust Header Compression (ROHC-FN)", Work in Progress, June 2005.

Appendix A: General Format of the IR-CR Packet (Informative)

A.1. General Structure (Informative)

This section provides an example of the format of the profile-specific information within the general format of the IR-CR.

```
  0   1   2   3   4   5   6   7
+---+---+---+---+---+---+---+---+
|                               |
/ replication base information  / variable length
|                               |
+---+---+---+---+---+---+---+---+
|                               |
/    replication information    / variable length
|                               |
 - - - - - - - - - - - - - - - -
```

Replication base information: The contents of this part of the IR-CR packet are defined by the individual profiles. This information is interpreted according to the profile indicated in the Profile field. It MUST include a 7-bit CRC over the original uncompressed header using the polynomial of Section 3.4.1.1. See Appendix A.2.

Replication information: The contents of this part of the IR-CR packet are also defined by the individual profiles. This part contains the static and dynamic subheader information used for replication. How this information is structured is profile specific; profiles may define the contents of this field using a chain structure (static and dynamic replication chains) or by defining header formats for replication (e.g., ROHC-TCP [3]).

A.2. Profile-Specific Replication Information (Informative)

This section provides a more detailed example of the possible format of the replication information field described in Appendix A.1:

```
+---+---+---+---+---+---+---+---+
| B |          CRC7             |  1 octet
+---+---+---+---+---+---+---+---+
|                               |  present if B = 1,
/           Base CID            /  1 octet if for small CIDs, or
|                               |  1-2 octets if for large CIDs
+---+---+---+---+---+---+---+---+
```

B: B = 1 indicates that the Base CID field is present.

CRC7: The CRC over the original, uncompressed, header. This 7-bit CRC is computed according to Section 3.4.1.1.

Base CID: The CID identifying the base context used for replication.

Appendix B: Inter-Profile Context Replication (Informative)

Context replication as defined in this document does not explicitly support the concept of context replication between profiles. However, it might be of interest when developing new compression profiles.

Inter-profile context replication would require that the decompressor have access to data structures from the base context, which belongs to a profile different than the profile using replication. This information would have to be made available in a format consistent with the data structures and encoding method(s) in use for all header fields that are being replicated.

B.1. Defining Support for Inter-Profile Context Replication

A ROHC profile describes how to compress a specific protocol stack, and includes one or more sets of packet formats. The packet formats will typically compress the protocol headers relative to a context of field values from previous headers in a flow. This context may also contain some control data. Thus, the packet formats specify a mapping between the uncompressed and compressed version of a protocol field.

This mapping is achieved through the use of one or more encoding methods, which are simply functions applied to compress or decompress a field. An encoding method is in turn defined using a name, a set of function parameters, and a formal expression (i.e., using the ROHC-FN [6]) or a textual description (i.e., a la RFC 3095 [2]) of its behaviour.

To compress one or more fields of a specific protocol stack, different profiles may define their packet formats using different encoding methods, or using a variant of a similar technique. A typical example of the latter is list compression, such as used for IP extension headers. This implies that context entries for a field belonging to a specific protocol stack may differ in their content, representation, and structure from one profile to another.

As a consequence of the above, a profile that supports context replication can only use a base context from another profile explicitly supporting the concept of a base context. That is, existing profiles not supporting this concept must be updated first to ensure that they can export the necessary context data entries that use a meaningful representation during replication.

Specifically, inter-profile context replication would require that decompressor implementations (including existing ones) of other profiles be updated when adding support for a profile that uses context replication. Therefore, inter-profile context replication cannot be seen as an implementation-specific issue.

The compressor must know if the decompressor supports inter-profile context replication before initiating the procedure. The compressor must also know which contexts (belonging to which profile) may be used as a base context. Therefore, a compressor cannot initiate context replication using a base context belonging to a different profile, unless that profile explicitly provides the proper mapping for its context entries or that profile is defined formally using ROHC-FN [6] in a manner that makes both profiles compatible. The set of profiles negotiated for the channel (see also RFC 3095 [2]) can then be used to determine if a context for a specific profile can be used as a base context.

B.2. Compatibility between Different Profiles (Informative)

Compatibility between profiles, when replicating a field for a particular protocol stack, can be expressed as follow: a field that is compressed by different profiles is compatible for inter-profile replication if it is defined in the set of packet formats using the same mapping function between its uncompressed and compressed version.

For example, the IP Destination Address field which, based on the packet formats and compression strategies defined in RFC 3095 [2], is implicitly compressed using an encoding method equivalent to the static() method defined in ROHC-FN [6].

In particular, for profiles that define their packet formats using a formal notation such as ROHC-FN [6], two different encoding methods may not have the same name. Thus, a field from a protocol stack is said to be compatible for replication between two different profiles if it has an equivalent definition within respective packet formats.

Author's Address

Ghyslain Pelletier
Box 920
Ericsson AB
SE-971 28 Lulea, Sweden

Phone: +46 8 404 29 43
Fax: +46 920 996 21
EMail: ghyslain.pelletier@ericsson.com

Full Copyright Statement

Intellectual Property

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.