Robust Header Compression (RoHC) WG
Internet Draft
Document: <draft-miyazaki-rohc-kwhc-01.txt>
May 01, 2000
Expires: November 01, 2000

Akihiro Miyazaki
Hideaki Fukushima
Thomas Wiebke
Rolf Hakenberg
Carsten Burmeister

Matsushita

Robust Header Compression using Keyword-Packets

Status of this Memo

1. Abstract

Header Compression has proven to be essential for using RTP over bandwidth limited links. While all of these links have only a limited bandwidth, some are additional extremely unreliable, e.g. wireless links. It is shown, that existing Header Compression schemes, that work well over reliable links, perform very bad, if many errors occur.

This led to the development of a new robust Header Compression mechanism, that is introduced in this paper. The core idea is the definition of keyword packets, which reduces the amount of discarded packets at the decompressor.

The Header Compression scheme, the used packet formats and the decompression algorithm, are described in detail.

Table of Contents

Document History

00 March 01, 2000: First release.

01 May 01, 2000: Two different profiles (sets of packet types) are defined, optimized for different environments. Least Significant Bit (LSB) encoding is used. Some changes in the algorithm.

2. Conventions used in this document

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 [2].

3. Introduction

The increasing use of real-time services over the Internet, such as voice or video applications, led to a strong demand for a protocol to be used to transport data real-time or nearly real-time. The Real-time Transport Protocol (RTP) provides the means for such services. After the publication as an RFC, more and more applications were developed to use RTP.

Another important development in the last years has been the user’s demand for mobility, i.e. accessing many services, including video and data applications, anytime and anywhere.

While RTP is commonly used in fixed networks, which is the environment it was developed for, using it over mobile networks needs some further considerations. A problem that occurs is the limited bandwidth in the mobile channel. There is a need to use this bandwidth very efficiently to satisfy enough users. Enough users per cell are very important to be economically feasible.

Each of the protocols RTP, UDP and IP adds its own header, which sums up to 40 byte in most of the cases, i.e. 12 bytes RTP header, 8 bytes UDP header and 20 bytes IP version 4 header (If IP version 6 is used, the amount of overhead is increased by another 20 bytes, because of the extended address format).

To solve the problem of the large amount of overhead header compression mechanisms have been developed. CRTP is commonly used over reliable links. It compresses the header to a minimum of 2 bytes. While this works fine over reliable channels, it induces many additional errors, when used over unreliable channels. This is because the first or second order differences to the previous packet, which are efficiently coded, are transmitted. If a packet gets lost, due to bit errors over the wireless channel, the compressed header of the next packet cannot be decompressed. The decompressor must send a request for a packet with a decompressed header to resynchronize, but in the meanwhile all received compressed packets have to be discarded. This effect is shown in detail in [8].

In this paper we would like to present a header compression mechanism that is more robust against packet loss and hence performs better over unreliable channels.

4. The concept of keyword header compression

This section describes the concept of the keyword header compression proposal and how this concept leads to a better performance, when used over unreliable links.

The main idea behind this proposal is the definition of a keyword field. The packets with the same keyword field value, reference the same context state. Hence, if a non-keyword packet gets lost, the receiver is able to decompress the following packet.

The context state to be used is defined by sending a keyword packet, i.e. a packet that has a new keyword value and which contents update the context to the new context state.

The following packets are encoded by transmitting the Least Significant Bits (LSB) of regular changing fields (e.g. RTP Sequence Number) and the original values of fields that did change, but are usually constant (e.g. RTP Marker bit, RTP Payload Type).

While compressing and transmitting the packets, the LSB encoded values may increase and need more coded bytes in the compressed header. If the header size exceeds a certain threshold or a value had changed and seems to be stable now, a new keyword packet should be defined.

If the keyword packets are transmitted correctly, the receiver is able to decompress any incoming compressed header. However, bit errors may occur while transmitting keyword packets as well. In this case the receiver must send a message to the compressor, to ask for a packet with a header that reestablishes its context. This is always a new keyword packet.

Because every packet with a new keyword value, which is not a keyword packet must be discarded, the loss probability of the context should be minimized. It might be useful to send several keyword packets with the same keyword to establish a new context state for both sides. The LSB encoded values are transmitted as usual in those packets, while one has to take care that the original values that changed are transmitted in every of those keyword packets. Using this mechanism should reduce the context state loss probability, because only one of those keyword packets has to be received correctly.

Different sets of packet types, used for different environments might lead to a better performance. This paper defines two different packet type sets. While the mechanisms how to use these packets is quite similar, the packet formats are optimized for different environments. The difference between the environments is the use of the IPv4 Identification field. If the value is assigned sequentially, increasing by one for each packet, the header compression mechanism should take advantage of it. Anyway, because we cannot assume this behavior, another set of packet formats is defined, which is optimized for non sequential IPv4 Identification values.

The two sets of packet formats are called profiles in the remainder of this document. How it is decided and negotiated, which profile to use is not scope of this document. It should only be shown, that the differentiation between these environments might make sense.

5. Profile 1, optimized for sequential IP Identification

This section shows five different packets that are used to transmit the data and signal errors from the receiver to the sender. The packet formats are optimised for the use in an environment, where the IP Identification is assigned sequentially for the compressed packet stream. The format of the packets is described and it is explained in detail how the decompressor is able to regenerate the complete header from the given information. The exact compression behaviour is implementation specific, but it has to be ensured, that any decompressor is able to regenerate the complete header in the described way.

5.1 Full Header packet

The Full Header packet is sent at the beginning of the session to establish a valid context. It is only sent another time if requested by the receiver. The receiver must request a Full Header packet only if the initial packet was lost, or a severe error occurred, that cannot be solved by a Compressed Packet.

The format of this packet’s header is quite similar to the original IP/UDP/RTP header. However, as described in other header compression papers, such as CRTP [7], the length fields of the IP and UDP packets are redundant. They are usually signalled by the link layer. This enables us to use these fields to signal the header compression specific session context identifier (CID)as follows:

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|C-L|                           |   First length field
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     (CID)     |     (CID)     |   Second length field
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

C-L (CID Length):
  00 – no CID
  01 - 8 bit
  10 - 16 bit

The selection of 0, 8 or 16 bit CIDs enables the compressor to setup enough sessions while keeping the overhead to a minimum.

Packet type identification might not be done by the link layer. In this case another byte is added before the original full header:

```
+-+-+-+-+-+-+-+-+
|1|1|1|0|-|-|-|-|    Packet Type Identification
+-+-+-+-+-+-+-+-+
:  RTP/UDP/IP   :
:    packet     :
```

5.2 Basic-Compressed packet

The header of the Basic-Compressed packet is divided into a basic header that is transmitted for every packet of this type and header extensions that are only used if necessary.

5.2.1 Basic header

The basic header’s format is as follows:

```
  0   1   2   3   4   5   6   7
+...+...+...+...+...+...+...+...+
:      MSB of Session CID       :  if 16 bit CID is used
+...+...+...+...+...+...+...+...+
:      LSB of Session CID       :  if 16 or 8 bit CID is used
+---+---+---+---+---+---+---+---+
| 1 | 1 | 0 |KW |NKW| M | E | S |
+---+---+---+---+---+---+---+---+
:         LSB of RTP SN         :
+...+...+...+...+...+...+...+...+
:         MSB of RTP SN         :  if S==1
+...+...+...+...+...+...+...+...+
:                               :
:           Extension(s)        :  if E==1
:                               :
+...+...+...+...+...+...+...+...+
:                               :
+           UDP Checksum        +  if non-zero in context
:                               :
+---+---+---+---+---+---+---+---+
|           RTP Data            |
:                               :
```

CID:
The first two bytes can be used for the session CIDs.

KW (Keyword):
The Keyword field must be present in every packet. To detect loss of keyword packets, it must be changed at each renewal.

NKW (New Keyword Indication):
If this bit is set, the compressor defines this packet as a keyword packet. The context state after decompressing this packet is stored and referenced in the following packets. Several successive keyword packets should be sent, each containing the relevant information, to ensure the reception at the decompressor.

M (RTP Marker):
The M-bit represents the original RTP Marker.

E (Extension):
This bit indicates that at least one extension is used. The different possible extensions, that are used to transmit information about irregular changes in the header fields, are described in detail in the following sections.

S (RTP Sequence Number Indication):
This bit indicates if the LSB of the RTP Sequence Number or the original value follows directly.

S=0: 8 bit LSB RTP Sequence Number
S=1: 16 bit RTP Sequence Number

UDP Checksum:
If the UDP Checksum is enabled, this field contains the 16-bit Checksum, else it is not present.

5.2.2 Update Extension

This extension is sent every time some header fields changed in an irregular way and cannot be calculated from the RTP Sequence Number. This might be the case e.g. for the RTP Timestamp after a silent period, or for the IPv4 Time To Live value. If the NKW-bit is set, the fields transmitted in this extension define the new context state to be referenced by the following packets. Several successive keyword packets should be sent, each containing the relevant fields, to ensure the reception at the decompressor.

The format of the Update Extension is defined below:

```
  0   1   2   3   4   5   6   7
+---+---+---+---+---+---+---+---+
| 0 |SN |  ID   |TOS|TTL|PT | E |
+---+---+---+---+---+---+---+---+
.                               .
.   (LSB) IPv4 Identification   .  if ID > 0
.                               .
+...+...+...+...+...+...+...+...+
.                               .
.      (LSB) RTP Timestamp      .  if TS == 1
.                               .
+...+...+...+...+...+...+...+...+
:      IPv4 Type of Service     :  if TOS == 1
+...+...+...+...+...+...+...+...+
:      IPv4 Time To Live        :  if TTL == 1
+...+...+...+...+...+...+...+...+
:      RTP Payload Type     : - :  if PT == 1
+...+...+...+...+...+...+...+...+
```

The first bit (0) indicates that the Update extension is used.

SN (RTP Sequence Number Indication):
This bit indicates if the original RTP Sequence Number is transmitted in the RTP Sequence Number field.

ID (IPv4 Identification Indication):
This bit indicates if the original IPv4 Indiction value is transmitted in the IPv4 Identification field or the LSB of the Identification or nothing.
   ID=0: No IPv4 Identification
   ID=1: 8 bit LSB IPv4 Identification
   ID=2: 16 bit IPv$ Identification
   ID=3: not used

TS (RTP Timestamp Indication):
This bit indicates if the RTP Timestamp is transmitted. If it is set to one, one of the following fields are used in the (LSB) RTO Timestamp field:

```
+---+---+---+---+---+---+---+---+
| 0 |                           |
+---+  15 LSB of RTP Timestamp  +
|                               |
+---+---+---+---+---+---+---+---+
```

```
+---+---+---+---+---+---+---+---+
| 1 | 0 |                       |
+---+---+                       +
|                               |
+      22 LSB of RTP Timestamp  +
|                               |
+---+---+---+---+---+---+---+---+

+---+---+---+---+---+---+---+---+
| 1 | 1 | 0 |                   |
+---+---+---+                   +
|                               |
+      29 LSB of RTP Timestamp  +
|                               |
+                               +
|                               |
+---+---+---+---+---+---+---+---+

+---+---+---+---+---+---+---+---+
| 1 | 1 | 1 | 0 | -   -   -   - |
+---+---+---+---+---+---+---+---+
|                               |
+                               +
|                               |
+        RTP Timestamp          +
|                               |
+                               +
|                               |
+---+---+---+---+---+---+---+---+
```

TOS (IPv4 Type Of Service Indication):
This bit indicates the transmission of the IPv4 Type Of Service value in the IPv4 Type Of Service field.

TTL (IPv4 Time To Live Indication):
This bit indicates the transmission of the IPv4 Time To Live value in the IPv4 Time To Live field.

PT (RTP Payload Type Indication):
This bit indicates the transmission of the RTP Payload Type value in the RTP Payload Type field.

E (Extension):
This bit indicates that another extension follows this one.

5.2.3 Default Delta Extension

The compressor will follow the changes in the RTP Timestamp values and the IPv4 Identification values, relative to the changes in the RTP Sequence Number values. To do this a delta value according to the following equation might be used:

dTS = (TS(n)–TS(n-1)) / (SN(n)–SN(n-1)), with

```
dTS      : delta Timestamp
TS(n)    : Timestamp of current packet
TS(n-1)  : Timestamp of previous packet
SN(n)    : Sequence Number of current packet
SN(n-1)  : Sequence Number of previous packet
```

If the compressor detects that for several packets the delta timestamp or delta identification value is the same, this delta value can be used to calculate the timestamp or identification from the sequence number. To do so, the decompressor has to be informed about this default delta value. The compressor uses this extension to signal default delta timestamp (ddTS) or default delta identification (ddID) values to the decompressor. This extension should be sent in keyword packets only. If it is used, an update extension, containing the timestamp or respectively the identification field must be sent as well.

The format of the Default Delta Extension is given below:

```
  0   1   2   3   4   5   6   7
+---+---+---+---+---+---+---+---+
| 1 | 0 | - | - | ddTL  | ddIDL |
+---+---+---+---+---+---+---+---+
:              ddTS             :   if ddTL > 0
+...+...+...+...+...+...+...+...+
:              ddTS             :   if ddTL > 1
+...+...+...+...+...+...+...+...+
:              ddTS             :   if ddTL > 2
+...+...+...+...+...+...+...+...+
:              ddID             :   if ddIDL > 0
+...+...+...+...+...+...+...+...+
:              ddID             :   if ddIDL > 1
+...+...+...+...+...+...+...+...+
```

The first four bits identify this extension.

ddTL (default delta Timstamp Length):
This field indicates the length of the default delta Timstamp field:

- ddTL=0: no default delta Timestamp field present
- ddTL=1: 1 byte
- ddTL=2: 2 byte
- ddTL=3: 3 byte

ddIDL (default delta Identification Length):
This field indicates the length of the default delta Identification field:

- ddIDL=0: no default delta Identification field present
- ddIDL=1: 1 byte
- ddIDL=2: 2 byte
- ddIDL=3: not used

5.3 One-Byte-Header or Two-Byte-Header packet

If the compressor communicated the default delta values to the decompressor, the decompressor should be able to recalculate the identification and timestamp value from the sequence number. Hence it is not necessary to transmit these values in all packets.

The One-Byte-Header or Two-Bytes-Header packets cannot be used if other fields than the sequence number, timestamp and identification changed. The timestamp and identification also have to change according to the following equations:

$$TS(n) = TS(n-1) + (SN(n) - SN(n-1)) * ddTS$$
$$ID(n) = ID(n-1) + (SN(n) - SN(n-1)) * ddID$$

```
ddTS    : default delta Timestamp
ddID    : default delta Identification
TS(n)   : Timestamp of current packet
TS(n-1) : Timestamp of previous packet
SN(n)   : Sequence Number of current packet
SN(n-1) : Sequence Number of previous packet
ID(n)   : Identification of current packet
ID(n-1) : Identification of previous packet
```

In this state the compressor might use the One-Byte-Header or Two-Byte-Header packet. These packets contain only the LSB of the RTP Sequence Number and the keyword, which is enough information for the decompressor to regenerate the original header.

The packet format for the One-Byte-Header packet is given below:

```
  0   1   2   3   4   5   6   7
+...+...+...+...+...+...+...+...+
:      MSB of Session CID       :  if 16 bit CID is used
+...+...+...+...+...+...+...+...+
:      LSB of Session CID       :  if 16 or 8 bit CID is used
+---+---+---+---+---+---+---+---+
| 0 |KW |LSB RTP Sequence Number|
+---+---+---+---+---+---+---+---+
```

The packet format for the Two-Byte-Header packet is given below:

```
  0   1   2   3   4   5   6   7
+...+...+...+...+...+...+...+...+
:      MSB of Session CID       :  if 16 bit CID is used
+...+...+...+...+...+...+...+...+
:      LSB of Session CID       :  if 16 or 8 bit CID is used
+---+---+---+---+---+---+---+---+
| 1 | 0 |KW |                   |
+---+---+---+
|      LSB RTP Sequence Number  |
+---+---+---+---+---+---+---+---+
```

The decision which of these packets is to be used should be done according to the number of packets already sent after the last keyword packet (or the first keyword packet of a set of keyword packets sent successively). The following equation shows the condition for the use of the One-Byte-Header packet:

SN_max – SN_min < 64 – R.

```
     SN_max: max Sequence Number since the last keyword packet
     SN_min: min Sequence Number since the last keyword packet
         R : An integer defining the max possible reordering degree
```

If the following condition could not be fulfilled, it is not possible to use these packet types and a new keyword packet should be sent, containing at least and update extension:

SN_max – SN_min < 8192 – R.

5.4 Context State packet

This header compression proposal is aimed to perform good, even if used over an unreliable channel. Hence bit errors can occur quite frequently and packets will get lost. If the lost packet was a non-keyword packet, this does not effect the decompressor at all, but reception of a packet with a new keyword, without receiving an corresponding keyword packet invalidates the decompressor’s context. From this moment any compressed packet, even if it was received correctly, cannot be decompressed, until the context is updated correctly again.

To minimize the probability of this situation, several successive keyword packets should be sent (with the same keyword). But even all these packets might get lost. Hence a mechanism is needed to inform the compressor about a lost context, to request an update extension in a new keyword packet.

To request a context update, the decompressor must send immediately after detecting an invalid context, a Context State packet. This packet contains the last correctly received keyword and RTP Sequence Number. The compressor knows at reception of such a Context State packet, what information it has to send in the update extension, to update the decompressor’s context correctly.

Another error, that could occur, is the loss of the first packet, i.e. the Full Header packet. Since most of the header information, e.g. addresses and ports, are transmitted only in this packet, it is essential for the decompressor to establish a valid context to receive this packet. If the decompressor receives a Compressed packet, with a new session CID, that was not initialized, by a Full Header packet, this Full Header packet must have been lost. In this case the decompressor must request a new Full Header packet, by the means of the Context State packet.

The format of the Context State packet is as follows:

```
+...+...+...+...+...+...+...+...+
:      MSB of Session CID       :  if 16 bit CID is used
+...+...+...+...+...+...+...+...+
:      LSB of Session CID       :  if 8 or 12 bit CID is used
+---+---+---+---+---+---+---+---+
|FHL|KW |                       |
+---+---+---+---+---+---+---+---+
:                               :
+      RTP Sequence Number      +  if FHL == 0
:                               :
+...+...+...+...+...+...+...+...+
```

CID:
The first two bytes can be used for the session CID.

FHL (Full Header Lost):
If this bit is set to one, the first Full Header packet was lost, no context was established and a new Full Header packet is requested. If it is set to zero a context update is required and the RTP

Sequence Number of the last correctly decompressed packet is transmitted as well.

6. Profile 2, optimized for non-sequential IP Identification

This section shows five different packets that are used to transmit the data from the sender to the receiver and signal errors from the receiver to the sender. The packet formats are optimised for the use in an environment, where the IP Identification is not assigned strictly sequentially for the compressed packet stream. The identification value is expected to increase by a small random number (e.g. smaller than 64). The format of the packets is described and it is explained in detail how the decompressor is able to regenerate the complete header from the given information. The exact compression behaviour is implementation specific, but it has to be ensured, that any decompressor is able to regenerate the complete header in the described way.

6.1 Full Header packet

The Full Header packet is sent once (or several times) at the beginning of the session to establish a valid context. It is used exactly as in profile 1 and has the same format (see section 5.1 for details).

6.2 Basic-Compressed packet

The header of the Basic-Compressed packet is divided into a basic header that is transmitted for every packet of this type and header extensions that are only used if necessary.

6.2.1 Basic header

The basic header’s format is as follows:

```
  0   1   2   3   4   5   6   7
+...+...+...+...+...+...+...+...+
:      MSB of Session CID       :  if 16 bit CID is used
+...+...+...+...+...+...+...+...+
:      LSB of Session CID       :  if 16 or 8 bit CID is used
+---+---+---+---+---+---+---+---+
| 1 | 1 | 0 |KW |NKW| M | E |S/I|
+---+---+---+---+---+---+---+---+
: Type  :                       :
+...+...+  Sequence Number &    +  if S/I==1
:          Identification       :
+...+...+...+...+...+...+...+...+
:                               :
:           Extension(s)        :  if E==1
:                               :
+...+...+...+...+...+...+...+...+
:                               :
+           UDP Checksum        +  if non-zero in context
:                               :
+---+---+---+---+---+---+---+---+
|           RTP Data            |
:                               :
```

CID:
The first two bytes can be used for the session CIDs.

KW (Keyword):
The Keyword field must be present in every packet. To detect loss of keyword packets, it must be changed at each renewal.

NKW (New Keyword Indication):
If this bit is set, the compressor defines this packet as a keyword packet. The context state after decompressing this packet is stored and referenced in the following packets. Several successive keyword packets should be sent, each containing the relevant information, to ensure the reception at the decompressor.

M (RTP Marker):
The M-bit represents the original RTP Marker.

E (Extension):
This bit indicates that at least one extension is used. The different possible extensions, that are used to transmit information about irregular changes in the header fields, are described in detail in the following sections.

S/I (RTP Sequence Number & Identification Indication):
This bit signals that the LSB of the RTP Sequence Number and IPv4 Identification follow directly.

Type:
These two bits indicate how the following bytes are used for the Sequence Number and Identification:

- Type = 0:
  - 7 bit RTP Sequence Number
  - 7 bit IPv4 Identification
- Type = 1:
  - 6 bit RTP Sequence Number
  - 16 bit IPv4 Identification
- Type = 2:
  - 8 bit RTP Sequence Number
  - 14 bit IPv4 Identification
- Type = 3:
  - 10 bit RTP Sequence Number
  - 12 bit IPv4 Identification

UDP Checksum:
If the UDP Checksum is enabled, this field contains the 16-bit Checksum, else it is not present.

6.2.2 Update Extension

This extension is used similar as in profile one and also has the same format. For details see section 5.2.2.

6.2.3 Default Delta Extension

This extension is used similar as in profile one and also has the same format. For details see section 5.2.3.

6.3 Two-Byte-Header or Three-Byte-Header packet

If the compressor communicated the default delta values to the decompressor, the decompressor should be able to recalculate the timestamp value from the sequence number. Hence it is not necessary to transmit this value in all packets.

These packets cannot be used if other fields than the sequence number, timestamp and identification changed. The timestamp also has to change according to the following equations:

$$TS(n) = TS(n-1) + (SN(n) - SN(n-1)) * ddTS$$

```
ddTS    : delta Timestamp
TS(n)   : Timestamp of current packet
TS(n-1) : Timestamp of previous packet
SN(n)   : Sequence Number of current packet
SN(n-1) : Sequence Number of previous packet
```

In this state the compressor might use the Two-Byte-Header or Three-Byte-Header packet. These packets contain only the LSB of the RTP Sequence Number, LSB of IPv4 Identification and the keyword, which is enough information for the decompressor to regenerate the original header.

The packet format for the Two-Byte-Header packet is given below:

```
  0   1   2   3   4   5   6   7
+...+...+...+...+...+...+...+...+
:      MSB of Session CID       :  if 16 bit CID is used
+...+...+...+...+...+...+...+...+
:      LSB of Session CID       :  if 16 or 8 bit CID is used
+---+---+---+---+---+---+---+---+
| 0 |KW |LSB RTP Sequence Number|
+---+---+---+---+---+---+---+---+
|    LSB IPv4 Identification    |
+---+---+---+---+---+---+---+---+
```

The packet format for the Three-Byte-Header packet is given below:

```
  0   1   2   3   4   5   6   7
+...+...+...+...+...+...+...+...+
:      MSB of Session CID       :  if 16 bit CID is used
+...+...+...+...+...+...+...+...+
:      LSB of Session CID       :  if 16 or 8 bit CID is used
+---+---+---+---+---+---+---+---+
| 1 | 0 |KW | T |               |
+---+---+---+---+               +
|      LSB RTP Sequence Number  |
+              &                +
|      LSB IPv4 Identification  |
+---+---+---+---+---+---+---+---+
```

The T-bit indicates how the next bits are used to transport the RTP Sequence Number and the IPv4 Identification:

T=0:
- 10 bit RTP Sequence Number
- 10 bit IPv4 Identification

T=1:
- 8 bit RTP Sequence Number
- 12 bit IPv4 Identification

The decision which of these packets is to be used should be done according to the number of packets already sent after the last keyword packet (or the first keyword packet of a set of keyword packets sent successively). The following equation shows the conditions for the use of the Two-Byte-Header packet:

```
(SN_max – SN_min < 64 – R) && (ID_max – ID_min < 256 –R2)

     SN_max: max Sequence Number since the last keyword packet
     SN_min: min Sequence Number since the last keyword packet
         R : An integer defining the max possible reordering degree
     ID_max: max Identification value since the last keyword packet
     ID_min: min Identification value since the last keyword packet
         R2: An integer defining the max possible reordering degree
             for the IP identification value.
```

If the following condition could not be fulfilled, it is not possible to use these packet types and a new keyword packet should be sent, containing at least and update extension:

```
(SN_max – SN_min < 1024 – R) && (ID_max – ID_min < 1024 – R2) ||
(SN_max – SN_min < 256 – R) && (ID_max – ID_min < 4048 – R2)
```

6.4 Context State packet

The use and format of the context state packet is similar to profile 1, see section 5.3 for details.

7. Conclusions

It was stated that CRTP induces too many additional errors, when used over unreliable links. A new solution, keyword header compression, was described, which reduces the number of additional errors.

The solution is split into two different profiles, optimized for different scenarios. The average header size should be reduced compared to CRTP as well as the number of additional errors. If used in a cautious way, e.g. sending several succeeding keyword packets, the amount of additional errors should be neglectable.

8. Security Considerations

Security issues are not considered in this memo.

9. Intellectual property considerations

Matsushita has filed patent applications that might possibly have technical relation to this contribution.
If part(s) of the contribution by Matsushita employee is (are) included in a standard and Matsushita has patents and/or patent application(s) that are essential to implementation of such included part(s) in said standard, Matsushita is prepared to grant - on the basis of reciprocity (grantback) - a license on such included part(s) on reasonable, non-discriminatory terms and conditions (in according with paragraph 10.3.3 of the RFC 2026).

10. Acknowledgements

This proposal combines several ideas of other papers and introduces some new features and mechanisms. The main inspiration for some of these mechanisms came from discussions on the 47th IETF Meeting in Adelaide, the discussions on the ROHC mailing list and the Internet Drafts [7], [8], [11], [12], [13] and [14].

11. References

1 Bradner, S., "The Internet Standards Process -- Revision 3", BCP9, RFC 2026, October 1996.

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

3 Schulzrinne, H., Casner, S., Frederick, R. and V. Jacobson, "RTP: A Transport Protocol for real-time applications", RFC 1889, January 1996.

4 Postel, J.,"User Datagram Protocol", STD 6, RFC 768, August 1980.

5 Postel, J.,"Internet Protocol", STD 5, RFC 791, September 1981.

6 Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.

7 Casner, S. and V. Jacobson, "Compressing IP/UDP/RTP Headers for Low-Speed Serial Links", RFC 2508, February 1999.

8 Mikael Degermark, Hans Hannu, Lars-Erik Jonsson, Krister Svanbro, "CRTP over cellular radio links", Internet Draft (work in progress), June 1999.

9 Homepage of 3GPP: http://www.3gpp.org

10 "Procedure for Evaluation of Transmission Technologies for FPLMTS", ITU-R TG8-1, 8-1/TEMP/233-E, September 1995.

11 Martensson, Einarsson, Jonsson,"ROCCO Conversational Video Profiles", Internet Draft (work in progress), March 2000.

12 Le, Clanton, Liu, Zheng,"Adaptive Header Compression (ACE) for Real-Time Multimedia", Internet Draft (work in progress), March 2000.

13 Jonsson, Degermark, Hannu, Svanbro," RObust Checksum-based header COmpression (ROCCO)", Internet Draft (work in progress), March 2000.

14 Svanbro,"Lower Layer Guidelines for Robust Header Compression", Internet Draft (work in progress), March 2000.

12. Author’s Addresses

Akihiro Miyazaki
Matsushita Electric Industrial Co., Ltd
1006, Kadoma, Kadoma City, Osaka, Japan
Tel. +81-6-6900-9192
Fax. +81-6-6900-9193
Mail akihiro@isl.mei.co.jp

Hideaki Fukushima
Matsushita Electric Industrial Co., Ltd
1006, Kadoma, Kadoma City, Osaka, Japan
Tel. +81-6-6900-9192
Fax. +81-6-6900-9193
Mail fukusima@isl.mei.co.jp

Thomas Wiebke
Panasonic European Laboratories GmbH
Monzastr. 4c, 63225 Langen, Germany
Tel. +49-(0)6103-766-161
Fax. +49-(0)6103-766-166
Mail wiebke@panasonic.de

Rolf Hakenberg
Panasonic European Laboratories GmbH
Monzastr. 4c, 63225 Langen, Germany
Tel. +49-(0)6103-766-162
Fax. +49-(0)6103-766-166
Mail hakenberg@panasonic.de

Carsten Burmeister
Panasonic European Laboratories GmbH
Monzastr. 4c, 63225 Langen, Germany
Tel. +49-(0)6103-766-263
Fax. +49-(0)6103-766-166
Mail burmeister@panasonic.de

Full Copyright Statement