

Robust Header Compression (RoHC) WG
Internet Draft
Document: <[draft-ietf-rohc-rtp-kw-00.txt](#)>
May 23th, 2000
Expires: November 23th, 2000

Akihiro Miyazaki
Hideaki Fukushima
Thomas Wiebke
Rolf Hakenberg
Carsten Burmeister

Matsushita

Robust Header Compression using Keyword-Packets

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of [Section 10 of RFC2026](#) [1].

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts. Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at

<http://www.ietf.org/ietf/1id-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at

<http://www.ietf.org/shadow.html>.

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 update packets, which are referenced by non-update packets. This reduces the interdependency between the packets and with this 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

1. Abstract
2. Conventions used in this document
3. Introduction
4. The concept of keyword header compression
 - 4.1 Keyword field, update and non-update packet
 - 4.2 The use of LSB encoding
 - 4.3 Refreshing the context by sending update packets
 - 4.3.1 Minimizing the loss probability of update packets
 - 4.3.2 Restrict the use of new keywords
 - 4.4 Loss of update packets
 - 4.5 Adaptation to the environment
5. Profile 1, optimized for sequential IPv4 Identification
 - 5.1 Full Header packet
 - 5.2 Basic-Compressed packet
 - 5.2.1 Basic header
 - 5.2.2 Changing Field extension
 - 5.2.3 Default-Delta extension
 - 5.3 One-Byte-Header or Two-Byte-Header packet
 - 5.4 Context State packet
6. Profile 2, optimized for non-sequential IPv4 Identification
 - 6.1 Full Header packet
 - 6.2 Basic-Compressed packet
 - 6.2.1 Basic header
 - 6.2.2 Changing Field extension
 - 6.2.3 Default-Delta extension
 - 6.3 Two-Byte-Header or Three-Byte-Header packet
 - 6.4 Context State packet
7. Conclusions
8. Security considerations
9. Intellectual property consideration
10. Acknowledgements
11. References
12. Author's addresses

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.

00 May 23, 2000: Submitted as a working group document. Changes include some clarifications of the concept of update packets and their use ([Section 4](#)).

Miyazaki

Expires November 23th, 2000

2

Header Compression using Keyword Packets

May 23th, 2000

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 present a header compression mechanism that is more robust against packet loss and hence performs better over unreliable channels.

Miyazaki

Expires November 23th, 2000

3

Header Compression using Keyword Packets

May 23th, 2000

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.

4.1 Keyword field, update and non-update packet

The problem of CRTP is its interdependency between all packets. If one packet gets lost, all following packets, until the next update, are lost. This is due to the fact that every packet, that is sent by the compressor, updates the context. Hence the context state at the decompressor will become invalid, if one of these packets is lost.

A possibility to solve this problem is to update the context only from time to time. However the context that is used has to be communicated and it has to be ensured that the decompressor received the context correctly.

The main idea is the definition of a keyword field. The packets with the same keyword field value, reference the same context state. The context state to be used is defined by sending a update 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 called non-update packets, because they do not update the context.

Hence, if a non-update packet gets lost, the receiver is able to decompress the following packets.

4.2 The use of LSB encoding

The packets that follow an update packet, are encoded by transmitting the Least Significant Bits (LSB) of regular changing fields (e.g. RTP Sequence Number). In some cases it might not be necessary to transmit the regular changing fields at all, e.g. if the timestamp can be calculated from the sequence number it is not transmitted. The packets also contain the original values of fields that did change since the last update, but are usually assumed to be constant (e.g. RTP Marker bit, RTP Payload Type).

The number of LSB to be used, is defined by the value itself. The compressor has to ensure that the decompressor is able to decode the value correctly. The decoding is done by simply replacing the transmitted LSB with the LSB of the context value that is referenced.

Miyazaki

Expires November 23th, 2000

4

Header Compression using Keyword Packets

May 23th, 2000

4.3 Refreshing the context by sending update packets

From time to time it will be necessary to update the context. There are mainly two reasons to do so.

First, while compressing and transmitting the compressed non-update packets, the LSB encoded values may increase and need more coded bits in the compressed header. If the header size exceeds a certain threshold, a new keyword should be sent in an update packet. This enables the compressor to use less LSB in the following non-update packets. E.g. after a while the number of LSB to encode the RTP sequence number will grow. If this value exceeds 6 bits, it might be useful to send an update packet, because the information would not fit into an one-byte header packet any more. After the successful update the compressor is able to send one-byte header packets again.

Second, if a value had changed and seems to be stable now, a new update packet should be sent. Otherwise the changed value would have to be transmitted in all following packets. E.g. after a silent

period, the timestamp changes by another value than the default difference timestamp. From this on it is not possible for the decompressor to calculate the timestamp from the RTP sequence number. Hence the compressor either sends the LSB of the timestamp in every packet, or updates the context, to enable the decompressor to calculate the timestamp from the RTP sequence number again.

4.3.1 Minimizing the loss probability of update packets

Because every packet with a new keyword value, which is not an update packet must be discarded, the loss probability of the context should be minimized. It might be useful to send several update packets with the same keyword value 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 of the fields that changed irregularly are transmitted in every of those update packets. The use of this mechanism reduces the context state loss probability, because only one of those update packets has to be received correctly.

4.3.2 Restrict the use of new keywords

The number of different keywords is limited by the number of bits used for the keyword field. In this paper we use only one bit, which leads to two different keywords. To ensure that consecutive packet loss of a few packets does not lead to wrong decompression, the use of new keyword values must be limited.

Miyazaki

Expires November 23th, 2000

5

Header Compression using Keyword Packets

May 23th, 2000

It is only allowed to send a new keyword in an update packet, if N non-update packets were sent since the last keyword change. This restriction is possible, because one never is forced to send an update packet. It is always possible to send all information in a non-update packet. Maybe there will be some overhead during this period, because all changes have to be transmitted in every packet, but if the keywords are used properly, this should only very seldom happen.

To use the keyword properly means that it is only changed if the compressor is rather sure that the values will remain constant for

the next packets. An example of a non-properly used keyword change is the definition of a new default delta timestamp value (in an update packet), just because it changed for one packet. This might be due to a silent period and might change back to the original value in the next packet again.

If the compressor follows this restriction, $2 \cdot N$ consecutive packets have to be lost, before the decompressor would not detect the loss of the update packet. To avoid even this situation a time-out might be applied, after which the decompressor will only accept new update packets or Full Header packets.

4.4 Loss of update packets

Only if the update packets are transmitted correctly, the receiver is able to decompress any incoming compressed header. Therefore if the update packets are transmitted multiple times, the probability that none of this packets is received, is very low. However, packet loss may occur while transmitting update packets. In case none of the update packets was received and the decompressor received a packet with a new keyword that is not an update packet, it must send a message to the compressor, to ask for a packet with a header that reestablishes its context. This is always an update packet or a Full Header packet.

4.5 Adaptation to the environment

The compressor has a lot of freedom in the compression algorithm. Even though the use of new keywords is restricted, the compressor decides when the keywords should be changed. Two strategies are possible, which are a tradeoff between compression efficiency and robustness against packet loss. One possibility is to send a new keyword as often as needed and possible. E.g. the keyword is changed, if the header size exceeds 1 byte. Another possibility is to sent new keywords less frequent. While on the one hand the compression efficiency might be better in the first case, the second possibility is more robust and less susceptible for packet loss.

Using this freedom the compressor may adapt the compression to the environment (i.e. the experienced BER or RTT). Another parameter of

the environment that should be taken into consideration is the assignment of the IPv4 Identification value. While it is possible to have a totally random IP Identification, it might also be possible that it is increased for every packet by a fixed value (sequential IP ID). Different sets of packet types, used for different environments might lead to a better performance. This paper defines two different packet type sets. The packet formats are optimized for different environments. If the IP ID is assigned sequentially, increasing by a fixed value 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 IPv4 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 IPv4 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 or a severe error occurred. 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.

To ensure the correct reception of the fields that are only transmitted in this packet type, it might be useful to use this packet type for several succeeding packets. The next packet type to use is always an update packet, which contains a new keyword. The decompressor will discard any other received packet and sent a

context state feedback, until it receives an update packet to establish a valid context (the keyword is part of the context).

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 û no CID
01 - 8 bit
10 - 16 bit

```

The selection of 0, 8 or 16 bit CIDs enables the compressor to set-up 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. Update and non-update packets can be sent in Basic-Compressed packet format. The type is identified by the new keyword flag, which is set for update packets.

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 update packets, it must be changed at each renewal.

NKW (New Keyword Indication):

If this bit is set, the compressor defines this packet as an update packet. The context state after decompressing this packet is stored and referenced in the following packets. Several successive update 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.

Miyazaki

Expires November 23th, 2000

9

Header Compression using Keyword Packets

May 23th, 2000

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 Changing Field 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 (i.e. the packet is an update packet), the fields transmitted in this extension define the new context state to be referenced by the following packets. Several successive update packets should be sent, each containing the relevant fields, to ensure the reception at the decompressor.

The format of the Changing Field Extension is defined below:

0 1 2 3 4 5 6 7

```

+---+---+---+---+---+---+---+---+
| 0 | ID | TS | 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
+...+...+...+...+...+...+...+...+

```

Miyazaki

Expires November 23th, 2000

10

Header Compression using Keyword Packets

May 23th, 2000

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

ID (IPv4 Identification Indication):

This bit indicates if the original IPv4 Indication 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 IPv4 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) RTP Timestamp field:

```

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

+---+---+---+---+---+---+---+---+
| 1 | 0 |                                     |
+---+---+

```


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 update packets only. If it is used, a change extension, containing the timestamp or respectively the identification field must be sent as well.

The format of the Default Delta Extension is given below:

Miyazaki Expires November 23th, 2000 12
 Header Compression using Keyword Packets May 23th, 2000

```

  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 Timestamp Length):

This field indicates the length of the default delta Timestamp 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

Packets of these two types are always non-update packets. They reference the last update packet and carry the same keyword value.

If the compressor communicated the default delta values to the decompressor and all changes are regular, 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:

$$TS(n) = TS(n-1) + (SN(n) \hat{=} SN(n-1)) * ddTS$$

$$ID(n) = ID(n-1) + (SN(n) \hat{=} SN(n-1)) * ddID$$

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 the context RTP sequence number. The not transmitted MSB of the RTP sequence number must not change.

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

CID:

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

Miyazaki

Expires November 23th, 2000

15

Header Compression using Keyword Packets

May 23th, 2000

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 IPv4 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 IPv4 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 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. As described for the previous profile, this packet type can be used for update packets (new-keyword flag set to one) or non-update packets.

6.2.1 Basic header

The basic header's format is as follows:

Miyazaki Expires November 23th, 2000 16
Header Compression using Keyword Packets May 23th, 2000

```

    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 update packets, it must be changed at each update.

NKW (New Keyword Indication):

If this bit is set, the compressor defines this packet as an update packet. The context state after decompressing this packet is stored and referenced in the following packets. Several successive update 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.

Miyazaki

Expires November 23th, 2000

17

Header Compression using Keyword Packets

May 23th, 2000

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 Changing Field 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

These two packet types can only be used for non-update packets. They reference the last update packet and therefore carry the same keyword value.

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:

$$TS(n) = TS(n-1) + (SN(n) \hat{-} SN(n-1)) * ddTS$$

ddTS : delta Timestamp
 TS(n) : Timestamp of current packet
 TS(n-1) : Timestamp of previous packet

Miyazaki	Expires November 23th, 2000	18
Header Compression using Keyword Packets		May 23th, 2000

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

update packet (or the first update packet of a set of update packets sent successively). The not transmitted MSB of these values must not have changed.

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 is 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 near zero.

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

Miyazaki

Expires November 23th, 2000

22

Header Compression using Keyword Packets

May 23th, 2000

Full Copyright Statement

"Copyright (C) The Internet Society (2000). All Rights Reserved.
This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

Miyazaki

Expires November 23th, 2000

23