

Network Working Group  
INTERNET-DRAFT  
Expires: September 2000

Anton Martensson, Ericsson  
Torbjorn Einarsson, Ericsson  
Lars-Erik Jonsson, Ericsson  
Sweden  
March 10, 2000

ROCCO Conversational Video Profiles  
<[draft-martensson-rocco-video-00.txt](#)>

#### Status of this memo

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

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 cite them other than as "work in progress".

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/lid-abstracts.txt>

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

This document is an individual submission to the IETF. Comments should be directed to the authors.

#### Abstract

The Robust Checksum-based header Compression [[ROCCO](#)] scheme is a header compression scheme designed to work over error prone channels. The scheme is adaptable to the characteristics of the link over which it is used and also to the properties of the packet streams it compresses.

This document describes a number of ROCCO profiles designed for conversational video over error prone channels. The profiles compress the size of the IP/UDP/RTP header down to a minimum of 2 octets.

---

INTERNET-DRAFT      ROCCO Conversational Video Profiles      March 10, 2000

## Table of contents

<a href="#">1.</a>	<a href="#">Introduction.....</a>	<a href="#">3</a>
<a href="#">2.</a>	<a href="#">Terminology.....</a>	<a href="#">4</a>
<a href="#">3.</a>	<a href="#">Header compression profiles for video packet streams.....</a>	<a href="#">4</a>
<a href="#">3.1.</a>	<a href="#">Usage scenarios, environment and requirements.....</a>	<a href="#">4</a>
<a href="#">3.2.</a>	<a href="#">Analysis of change patterns of header fields.....</a>	<a href="#">5</a>
<a href="#">3.2.1.</a>	<a href="#">RTP Marker.....</a>	<a href="#">6</a>
<a href="#">3.2.2.</a>	<a href="#">RTP Timestamp.....</a>	<a href="#">7</a>
<a href="#">3.3.</a>	<a href="#">Profile definitions.....</a>	<a href="#">7</a>
<a href="#">3.3.1.</a>	<a href="#">List of defined profiles.....</a>	<a href="#">7</a>
<a href="#">3.3.2.</a>	<a href="#">Additional, common profile characteristics....</a>	<a href="#">9</a>
<a href="#">3.4.</a>	<a href="#">Encoding of changing fields.....</a>	<a href="#">10</a>
<a href="#">3.4.1.</a>	<a href="#">Sequence number changes to handle.....</a>	<a href="#">10</a>
<a href="#">3.4.2.</a>	<a href="#">Sequence number compression.....</a>	<a href="#">11</a>
<a href="#">3.4.3.</a>	<a href="#">Sequence number decompression.....</a>	<a href="#">11</a>
<a href="#">3.4.4.</a>	<a href="#">RTP Sequence number wrap-around.....</a>	<a href="#">12</a>
<a href="#">3.4.5.</a>	<a href="#">Timestamp changes to handle.....</a>	<a href="#">13</a>
<a href="#">3.4.6.</a>	<a href="#">Timestamp compression.....</a>	<a href="#">14</a>
<a href="#">3.4.7.</a>	<a href="#">Timestamp decompression.....</a>	<a href="#">15</a>
<a href="#">3.4.8.</a>	<a href="#">RTP Timestamp wrap-around.....</a>	<a href="#">15</a>
<a href="#">3.4.9.</a>	<a href="#">Changes of Delta Timestamp.....</a>	<a href="#">16</a>
<a href="#">3.5.</a>	<a href="#">Header formats.....</a>	<a href="#">16</a>
<a href="#">3.5.1.</a>	<a href="#">Packet identification.....</a>	<a href="#">16</a>
<a href="#">3.5.2.</a>	<a href="#">Static information packet, initialization....</a>	<a href="#">16</a>
<a href="#">3.5.3.</a>	<a href="#">Dynamic information packet.....</a>	<a href="#">18</a>
<a href="#">3.5.4.</a>	<a href="#">Compressed packets.....</a>	<a href="#">21</a>
<a href="#">3.5.5.</a>	<a href="#">Extensions to compressed headers.....</a>	<a href="#">22</a>
<a href="#">3.5.6.</a>	<a href="#">Feedback packets.....</a>	<a href="#">25</a>
<a href="#">3.6.</a>	<a href="#">Startup of timestamp encoding.....</a>	<a href="#">26</a>
<a href="#">3.7.</a>	<a href="#">Context update procedures.....</a>	<a href="#">26</a>
<a href="#">3.8.</a>	<a href="#">ROCCO over simplex links.....</a>	<a href="#">26</a>
<a href="#">4.</a>	<a href="#">Compression performance.....</a>	<a href="#">26</a>
<a href="#">5.</a>	<a href="#">Implementation status.....</a>	<a href="#">27</a>
<a href="#">6.</a>	<a href="#">Security considerations.....</a>	<a href="#">27</a>
<a href="#">7.</a>	<a href="#">Acknowledgements.....</a>	<a href="#">27</a>
<a href="#">8.</a>	<a href="#">Intellectual property considerations.....</a>	<a href="#">27</a>
<a href="#">9.</a>	<a href="#">Authors' addresses.....</a>	<a href="#">28</a>
<a href="#">10.</a>	<a href="#">References.....</a>	<a href="#">29</a>

---

INTERNET-DRAFT      ROCCO Conversational Video Profiles      March 10, 2000

## 1. Introduction

The Internet and cellular telephony are two communication technologies that have become commonly used by the general public. With Internet, the general public now has the possibility to see video clips in form of news, sports and entertainment. Currently, this is true only if a fixed connection is used. If a cellular link is used for the connection, the lack of bandwidth is a big problem. Within a few years, several new cellular standards with larger bandwidth will be ready for traffic. This will improve the situation for video over cellular links but the bandwidth will still be restricted. The performance over the cellular link could be improved a lot with an effective and error robust header compression scheme.

In the RObust Checksum-based header COmpression [[ROCCO](#)] draft, a robust header compression scheme for error prone channels is presented. The scheme is designed to handle packet losses both before and between the compression points. In the draft, profiles for IP-telephony are presented. The profiles compress the 40 octet IP/UDP/RTP headers down to a minimum size of 1 octet. Simulations show that the performance for these profiles is at least as good as with [CRTP] for error free channels while they outperform CRTP for error prone channels.

This document is an addition to the [[ROCCO](#)] document and defines profiles for IP-video. The profiles are designed for conversational video with a fixed reference clock. The profiles can handle variable picture frame rate, where variable frame rate means that frames from the source are skipped to get the lower frame rate. The profile is

also designed to handle bi-directionally predicted frames, B-frames, in the compressed packet, thus supporting non-conversational video as well.

## 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](#).

### Header Compression profile

A header compression profile is a specification of how to compress the headers of a certain kind of packet stream over a certain kind of link.

### Header compression CRC

A CRC computed by the compressor and included in each compressed header. Its main purpose is to provide a way for the decompressor to reliably verify the correctness of reconstructed headers. What values the CRC is computed over depends on the packet type it is

included in and typically, it covers the original header.

#### PCF

Picture clock frequency. The picture framerate delivered from the picture source. Example: In a PAL system using only one field, the PCF 25 Hz.

#### Timestamp delta

The timestamp delta is the change in the timestamp value between two consecutive packets.

### [3.](#) Header compression profiles for video packet streams

This section describes the video profiles of ROCCO. A number of profiles are defined providing support for both IPv6 [Ipv6] and IPv4 [Ipv4] in combination with various functionality.

#### [3.1.](#) Usage scenarios, environment and requirements

These profiles are intended for IP-video over cellular links with high error rates. The profiles are designed to successfully handle loss of at least three consecutive packets over the link, without introducing any additional loss. Packet type identification is included in all profiles which means that such functionality SHOULD NOT be provided by the link layer used. Regarding packet stream separation, various profiles are defined supporting different numbers of concurrent streams.

As a cellular link with similar characteristics is expected at the other end of the connection, the profiles are also designed to handle

some consecutive lost packets before the compression point without increasing the size of the compressed header. The profiles are also in general designed to handle reordering of single packets before the compression point, with only a small increase of the compressed header size.

#### [3.2.](#) Analysis of change patterns of header fields

To design suitable mechanisms for efficient compression of all header fields, their change patterns need to be analyzed. For this reason, an extended classification specialized on IP-video packet streams is made, which applies to all profiles defined in this document. This classification is based on the general classification in ROCCO [ROCCO], and considers the fields which were classified as CHANGING in that classification. These fields are further separated into five different subclasses:

STATIC	These are fields that were classified as CHANGING on a general basis, but are classified as STATIC for IP-telephony packet streams.
SEMISTATIC	These fields are STATIC most of the time. However, occasionally the value changes but returns to its original value after a known number of packets.
(R)ARELY-(C)HANGING	These are fields that changes their values occasionally and then keep their new values.
ALTERNATING	These fields have a few different values which they are alternating between.
IRREGULAR	These are the fields for which no useful change pattern can be identified.

To further expand the classification possibilities without making it more complex, the classification can be done either on the values of the field and/or on the values of the deltas for the field.

When the classification is done, something should finally be stated regarding possible additional knowledge about the field values and/or field deltas, according to the classification. For fields classified as STATIC or SEMISTATIC, the case may be that the value of the field is not only STATIC but also well KNOWN a priori (two states for SEMISTATIC fields). For fields with non-irregular change behaviors, it may be known that changes use to be within a LIMITED scope compared to the maximal change for the field. For others, the values are completely UNKNOWN.

Table 3.1 classifies all the CHANGING fields based on their expected change patterns for IP-telephony streams.

Field	Value/Delta	Class	Knowledge
Sequential	Delta	STATIC	KNOWN
IPv4 Id: Seq. jump	Delta	RC	LIMITED
Random	Value	IRREGULAR	UNKNOWN
IP TOS / Tr. Class	Value	RC	UNKNOWN
IP TTL / Hop Limit	Value	ALTERNATING	LIMITED
Disabled	Value	STATIC	KNOWN
UDP Checksum: Enabled	Value	IRREGULAR	UNKNOWN
No mix	Value	STATIC	KNOWN
RTP CSRC Count: Mixed	Value	RC	LIMITED
RTP Marker	Value	SEMISTATIC	KNOWN
RTP Payload Type	Value	RC	UNKNOWN
RTP Sequence Number	Delta	STATIC	KNOWN
RTP Timestamp	Delta	RC	LIMITED
No mix	-	-	-
RTP CSRC List: Mixed	Value	RC	UNKNOWN

Table 3.1 : Classification of CHANGING header fields

The only fields that differ from the speech profile of ROCCO defined in [ROCCO] are the RTP marker and the RTP Timestamp. The following subsection discusses these fields in detail. For a detailed discussion about the other fields, please see [ROCCO]. Note that table 3.1 and the discussions below do not consider changes caused by loss or reordering before the compression point.

### 3.2.1. RTP Marker

The marker bit should be set for the last packet of every picture,

which means that it has an alternating characteristic with well known values for both states.

### [3.2.2.](#) RTP Timestamp

The change in RTP timestamp between two packets with successive sequence numbers will either be zero or increase by a multiple of a fixed value corresponding to the picture clock frequency. The timestamp can also decrease with a multiple of the fix value if B-pictures are used. The delta interval expressed as a multiple of the picture clock frequency is in most cases very limited.

### [3.3.](#) Profile definitions

This document defines a number of different header compression profiles. The definitions are built up of requirements and capabilities of each profile in combination with information about which mechanisms that are used to implement the desired behaviors.

#### [3.3.1.](#) List of defined profiles

All defined profiles are listed in Table 3.2 together with their characteristics and pointers to implementation details that may differ from profile to profile.

The first six columns state requirements and capabilities of the profiles. The meaning of the columns are:

Nr

This is the identification number for each profile. These numbers are used when negotiating profiles in the header compression setup phase. These numbers are preliminary and will perhaps change in coming versions of this draft.

IPv

This is the IP version for which the profile is designed. Possible



values for this column are 6 and 4.

Str

This column gives the number of concurrent packet streams that are supported by the header compression profile.

Chk

This column indicates whether the profile supports packet streams with the UDP checksum (E)nabled or D(isabled).

Id

For profiles supporting IPv4, this column indicates which behavior of the IPv4 Identification field the profile is optimized for. Possible values in this column are:

- |                     |  |
|---------------------|--|
| (S)EQUENTIAL        | These profiles can not handle streams with IP Identification values assigned using any other policy than sequential.   |
| (S)EQUENTIAL (J)UMP | These profiles are recommended if the Identification assignment is sequential jump [ <a href="#">ROCCO</a> ]. To signal the Identification, the 8 least significant bits are sent which means that the header size will increase with one octet. |
| (R)ANDOM            | These profiles are recommended if it is known that random assignment or sequential jump is used. The Identification field will be included "as-is" which means that the header size will increase with two octets.                               |

MHS

Minimal Header Size for the profile.

The next three columns indicate how each profile is implemented. This includes header formats for STATIC (STA, chapter 3.5.2), DYNAMIC (DYN, chapter 3.5.3) and COMPRESSED (COM, chapter 3.5.4) packets.

Nr	IPv	Str	Chk	Id	MHS	STA	DYN	COM
1001	6	1	D	-	2	1	1	1
1002	6	1	E	-	4	1	5	2
1003	6	256	D	-	3	2	2	3
1004	6	256	E	-	5	2	6	4
1005	4	1	D	S	2	3	3	1
1006	4	1	E	S	4	3	7	2
1007	4	256	D	S	3	4	4	3
1008	4	256	E	S	5	4	8	4

Nr	IPv	Str	Chk	Id	MHS	STA	CUP	COM
1009	4	1	D	SJ	3	3	3	5
1010	4	1	E	SJ	5	3	7	6
1011	4	256	D	SJ	4	4	4	7
1012	4	256	E	SJ	6	4	8	8
1013	4	1	D	R	4	3	3	9
1014	4	1	E	R	6	3	7	10

+-----+	+-----+	+-----+	+-----+	+-----+	+-----+	+-----+	+-----+	+-----+	+-----+	+-----+	+-----+
1015	4	256	D	R	5	4	4	11			
+-----+	+-----+	+-----+	+-----+	+-----+	+-----+	+-----+	+-----+	+-----+	+-----+	+-----+	+-----+
1016	4	256	E	R	7	4	8	12			
+-----+	+-----+	+-----+	+-----+	+-----+	+-----+	+-----+	+-----+	+-----+	+-----+	+-----+	+-----+

Table 3.2 : List of defined profiles

### 3.3.2. Additional, common profile characteristics

In addition to what was stated in the left part of Table 3.1, all the profiles defined in this document applies to the following:

#### Link layer requirements

Except for the general link layer requirements (framing, length & profile negotiation) stated in [[ROCCO](#)], these profiles require also a reliable link layer CRC covering at least the header part of the packet. The CRC SHOULD ensure that packets with errors in the header part are never delivered.

#### Packet stream characteristics

These profiles are designed for packet streams carrying IP-video data.

#### Pre link characteristics

Several consecutive packet losses before the header compression point are for most of these profiles handled without introducing additional header overhead. Packet reordering on pre links is expected to be very uncommon.

#### Link Characteristics

The link over which header compression is performed is expected to

have a loss characteristic that very seldom leads to loss of more than three consecutive packets.

#### Variable frame rate

The profiles are designed to handle variable picture frame rate, where variable frame rate means that picture frames from the source sequence are skipped to reach the new frame rate.

#### Fixed video reference clock

The profiles are designed to compress the IP/UDP/RTP headers more if a fixed video reference clock is used. This means that the RTP timestamp has to change with a multiple of a fixed value which corresponds to the used picture clock frequency or a multiple thereof if only every N'th picture is coded.

### [3.4.](#) Encoding of changing fields

The analysis in [section 3.2](#) excludes changes due to packet loss or reordering before the compression point. Such changes will have an impact on the regularity of the RTP sequence number, the RTP timestamp and, for Ipv4, the IP ID value. As described in [\[ROCCO\]](#) the IP ID value (if sequentially assigned) will follow the RTP timestamp. The task is then to communicate RTP sequence number and RTP timestamp information in some way.

In the sequence number encoding, the sequence number modulo 7 is sent. In the timestamp encoding the RTP Timestamp is first divided by a delta Timestamp to decrease the change of the timestamp value between two packets and then a number of the least significant bits (LSB) is sent to the decoder. The Header Compression CRC computed over the original IP/UDP/RTP header is then used by the decompressor to verify the correctness of the decoding, i.e. that no wrap-around or loss of synchronization has occurred.

#### [3.4.1.](#) Sequence number changes to handle

The source increases the RTP sequence number with one for each packet. However, due to losses and reordering of packets on pre links the changes seen by the compressor may vary. The sequence number changes on the decompressor side may differ in that increase faster due to packet losses on the link between compressor and decompressor.

---

INTERNET-DRAFT      ROCCO Conversational Video Profiles      March 10, 2000

### [3.4.2.](#) Sequence number compression

In the COMPRESSED base packet, three bits are used for the sequence number (SeqNr) encoding. However, one bitpattern combination (111) is reserved for packet type identification (see [section 3.5.1](#)). To use these seven sequence number code points in an effective way, the RTP sequence number is divided by 7:

$$\text{SeqNr} = \text{SEQR} * 7 + \text{SEQ7}$$

where SEQ7 is the sequence number modulo 7 and SEQR is the integer part of the sequence number after division by 7.

The SEQ7 value is sent in the first three bits in the COMPRESSED base header. It does not matter if the packet losses are before or after the compression point or a combination of the two.

To be able to handle single reordering of packets before the compression point the seven code points SHOULD be interpreted in such a way that they cover the range [-1, 5] of sequence number change. This brings the possibility to lose up to 4 consecutive packets without losing the sequence number synchronization. It is possible to increase this range to cover larger sequence number jumps and to be even more robust to packet losses. A number of the least significant bits of SEQR can be sent in an extension to the COMPRESSED base packet. These bits SHOULD be interpreted in such a way that they cover the range [-3, (7\*2<sup>SEQR\_NB</sup>)-4] of sequence number change, where SEQR\_NB is the number of SEQR bits sent to the decompressor. The extended range may be used by the encoder to increase the probability of sustaining multiple packet losses on the link, when many packets are lost already before the compression point.

### [3.4.3.](#) Sequence number decompression

The encoding and decoding of the received sequence number is done in the following way, where S(n) is the RTP sequence number to be encoded/decoded and S(n'-1) denotes the latest successfully decoded sequence number.

Input sequence number: S(n)

Encoded sequence number: SEQ7(n) = S(n) modulo 7

Decoded sequence number: S(n') = S(n'-1) - S(n'-1) modulo 7 +  
S(n') modulo 7

To handle modulo wrap-around, an additional verification is inserted. If the decoded sequence number  $S(n')$  is  $S(n'-1)-2$  or smaller,  $S(n')$  is increased with 7. A mechanism to handle full-field wrap-around is also necessary and is presented in 3.4.4.

If the extended sequence number range is used, it is necessary to take the SEQR bits into account when decoding the sequence number.

Decoded sequence number:  $S(n') = S(n'-1) - \text{SEQR}_{\text{LSB}}(n'-1) * 7$   
 $- S(n'-1) \bmod 7$   
 $+ \text{SEQR}_{\text{LSB}}(n') * 7$   
 $+ S(n') \bmod 7$

Where SEQR LSB are the least significant bits of SEQR. If no SEQR bits were sent for the preceding sequence number, the decoder has to calculate these to be able to correctly decode the sequence number.

In this case, the modulo wrap-around test is changed to cover the new extended sequence number range in such a way that the sequence number delta is in the interval  $[-3, (7*2^{\text{SEQR}_{\text{LSB\_NB}}}-4)]$ , where  $\text{SEQR\_NB}$  is the number of SEQR<sub>LSB</sub> bits received by the decompressor.

If the CRC fails in the decompressor one possible reason could be that a wrap-around of the SEQ7 field has occurred. In this case it is still possible to decode the sequence number if guesses are used. Please see [\[ROCCO\]](#) for a detailed description.

#### [3.4.4](#) RTP Sequence Number wrap-around

The SEQ7 and SEQR<sub>LSB</sub> will not change linearly during the RTP sequence number wrap-around. The changes of SEQ7 and SEQR<sub>LSB</sub> are as follows:

SeqNr		SEQ7		SEQR <sub>LSB</sub> (bin)
=====				
<a href="#">65534</a>		0		... <a href="#">0010</a>
<a href="#">65535</a>		1		... <a href="#">0010</a>
<a href="#">0</a>		0		... <a href="#">0000</a>

[1](#) | 1 | ...[0000](#)

Table 3.3 : Change of SEQ7 and SEQRLSB during a wrap-around

During the RTP sequence number wrap-around the SEQ7 value will change from 1 to 0. If packet losses occur in connection to the wrap-around the probability that the sequence number is decoded wrong will increase. To avoid that, it is RECOMMENDED that the sequence number is either encoded with a DYNAMIC packet or with an extended range in an extension to the COMPRESSED packet. If extended range is used the following table shows how to interpret SEQ7 and SEQRLSB in the decoder during the wrap-around:

SeqNr	SEQ7	SEQRLSB(bin)
=====		
<a href="#">65532</a>	5	... <a href="#">01</a>
<a href="#">65533</a>	6	... <a href="#">01</a>
<a href="#">65534</a>	0	... <a href="#">10</a>
<a href="#">65535</a>	1	... <a href="#">10</a>
<a href="#">0</a>	0	... <a href="#">00</a>
.	.	.
.	.	.
<a href="#">11</a>	4	... <a href="#">01</a>

Table 3.4 : How to interpret SEQ7 and SEQRLSB during a wrap-around.

#### [3.4.5](#). Timestamp changes to handle

All relevant video standards up to now uses a reference clock at 90000 Hz [[RFC2032](#)]. For compression efficiency, it is assumed that the RTP timestamp changes with a multiple of a fix value, which corresponds to the current picture clock frequency (PCF) or a multiple thereof.

The difference in timestamp compared to the packet with the preceding sequence number may be

- \* 0, if the packet belongs to the same picture

- \* > 0 and a multiple of the picture clock frequency for ordinary Intra(I) and Inter(P) pictures as well as B-pictures following B-pictures

- \* < 0 and a multiple of the picture clock frequency for the first B-picture after an I or P picture

The picture clock frequency used in H.261 [[H.261](#)] and H.263 ver.1 [[H.263v1](#)] is 29.97 Hz which corresponds to a minimum timestamp increase between two coded pictures equal to 3003. For H.263 ver.2 [[H.263v2](#)] and MPEG-4 [[MPEG-4](#)] one can use a custom picture clock frequency. Common frequencies are 25 Hz and 30 Hz which corresponds to an increase in timestamp of a multiple of 3600 and 3000, respectively.

When packet losses occur either before or after the compression point, the increase of the timestamp will still be a multiple of the picture clock frequency. This property is used in the compression since it is sufficient to know how many PCF multiples the timestamp has increased.

The change of Timestamp should be a multiple of the picture clock frequency and correlated with the video bitstream temporal reference according to [[RFC2429](#)]. This is the case we have assumed and it applies to many commercial codecs but not all. For the codecs that do

not follow this scheme, the video profile still has the possibility to send the timestamp via extensions to the COMPRESSED header.

#### [3.4.6.](#) Timestamp compression

The change in RTP timestamp between two packets with successive sequence numbers will in most cases be zero or increase with a multiple of a fixed value. This value corresponds to the picture clock frequency used in the video encoder. If B-pictures are used it is also possible that the timestamp decreases between two pictures.



To be able to handle these cases and also to tolerate some packet losses, some LSB of a downscaled timestamp are transmitted. The downscaling is done in the following way:

$$TS = TSQ * PCTSI + TSR$$

TS = RTP timestamp

PCTSI = Picture Clock Interval in TimeStamp ticks. Example: 3600 for 25 Hz picture clock frequency ( $90000/25 = 3600$ ).

TSQ = Timestamp quotient. The integer part after the division.

TSR = Timestamp residual.  $TSR = TS \text{ modulo } PCTSI$ .

If the change of timestamp between two packets is a multiple of PCTSI, the TSR will remain the same and only TSQ will change. To encode the TSQ, the five least significant bits are sent in the COMPRESSED base packet. If B-pictures are used or if there has been reordering of packets it is possible that the timestamp will decrease between two packets. To be able to handle these situations, the TSQ bits SHOULD be interpreted in such a way that they cover the range of delta TSQ corresponding to  $[-6, 25]$ . It is possible to increase this range to cover larger timestamp jumps. The extended range is done by sending more TSQ bits in an extension to the COMPRESSED packet. These extra bits SHOULD increase the range both upwards and downwards so they cover the range  $[-10, 2^{(N+5)}-11]$ , where N is the number of TSQ bits sent in the COMPRESSED extension. The extended range is useful when the RTP Timestamp in the compression point has increased a lot from the last packet.

Note: The least significant bits of the TSQ are sent in the COMPRESSED header while the extended bits are sent in the extension.

Example: Assume that we have 5 TSQ bits in the COMPRESSED base header and an additional 2 TSQ bits in an extension. If we only have the five TSQ bits, it covers the range  $[-6, 25]$ , but with the additional 2 bits in the extension the range will be  $[-10, 117]$ .

The PCTSI is sent to the decompressor in the TS delta field either in the DYNAMIC packet or in an extension to the COMPRESSED packet.

If the timestamp is not changing with a multiple of the current PCTSI, the update of the timestamp has to be done in a different way.

The timestamp can be sent in either an DYNAMIC packet or an extension to a COMPRESSED packet. If the COMPRESSED packet is used a number of the least significant bits of the timestamp (TS LSB) are sent in the extension. Since the TSQ bits are not useful when the timestamp is not changing with a multiple of the current PCTSI, the decompressor MUST discard these. It is RECOMMENDED that at least 21 TS LSB's are sent in the extension. To be able to handle situations with decreasing timestamp the TS LSB's SHOULD be interpreted in such a way that they cover the range of delta timestamp corresponding to  $[-2^{16}, 2^{TS\_LSB\_NB}-2^{16}-1]$  from the last sent timestamp, where TS\_LSB\_NB is the size of the TS\_LSB field in bits.

#### [3.4.7.](#) Timestamp decompression

The decompression of the RTP timestamp is done in a similar way as the RTP sequence number except that modulo seven is not used and that the TSQ values have to be upscaled with the timestamp delta. Assume that  $T(n)$  is the current RTP timestamp and TSQ\_LSB is a number of least significant bits of TSQ.

$$\text{Decoded timestamp: } T(n') = T(n'-1) - \text{TSQ\_LSB}(n'-1) * \text{timestamp delta} + \text{TSQ\_LSB}(n') * \text{timestamp delta}$$

If the transmission of TSQ\_LSB( $n'-1$ ) and TSQ\_LSB( $n'$ ) is not done with the same number of TSQ bits, the decoder has to recalculate TSQ\_LSB( $n'-1$ ) to be of the same length as TSQ\_LSB( $n'$ ).

To handle TSQ\_LSB wrap-around, an additional verification is inserted. If TSQ\_LSB( $n'$ ) is TSQ\_LSB( $n'-1$ )-7 or smaller,  $T(n')$  is increased with  $2^{NB}$ , where NB is the number of bits used in the TSQ\_LSB( $n'$ ) transmission.

If the CRC fails in the decompressor one possible reason could be that a wrap-around of the TSQ field has occurred. In this case it is still possible to decode the sequence number if guesses are used. Please see [\[ROCCO\]](#) for a detailed description.

#### [3.4.8](#) RTP Timestamp wrap-around

During a wrap-around of the RTP timestamp field it is possible that the TSQ range will be smaller. The reason to this is that the RTP timestamp is divided by the PCTSI and the five least significant bits of TSQ will probably not change from [11111] to [00000]. To increase the probability to decompress the RTP timestamp correctly it is RECOMMENDED to send the timestamp in either a DYNAMIC packet or in an extension to a COMPRESSED packet. If a COMPRESSED packet is used it is RECOMMENDED that at least 21 TS LSB's are used.

Note: The TSR will probably change after the wrap-around and the compressor and the decompressor have to be aware of that.

---

INTERNET-DRAFT      ROCCO Conversational Video Profiles      March 10, 2000

#### [3.4.9.](#) Change of Delta Timestamp

When the Timestamp Delta is changed during a header compression session, both the new Timestamp Delta and the current timestamp MUST be sent to the decompressor. This can be done either by a DYNAMIC packet or a COMPRESSED packet with an extension.

### [3.5.](#) Header formats

The profiles defined in this document makes use of four different packet types; STATIC, DYNAMIC, COMPRESSED and FEEDBACK. This section defines these packet types together with a description of how to use and identify them.

#### [3.5.1.](#) Packet identification

To identify the packet types, four bit patterns for the initial five bits of the first octet are reserved. These are:

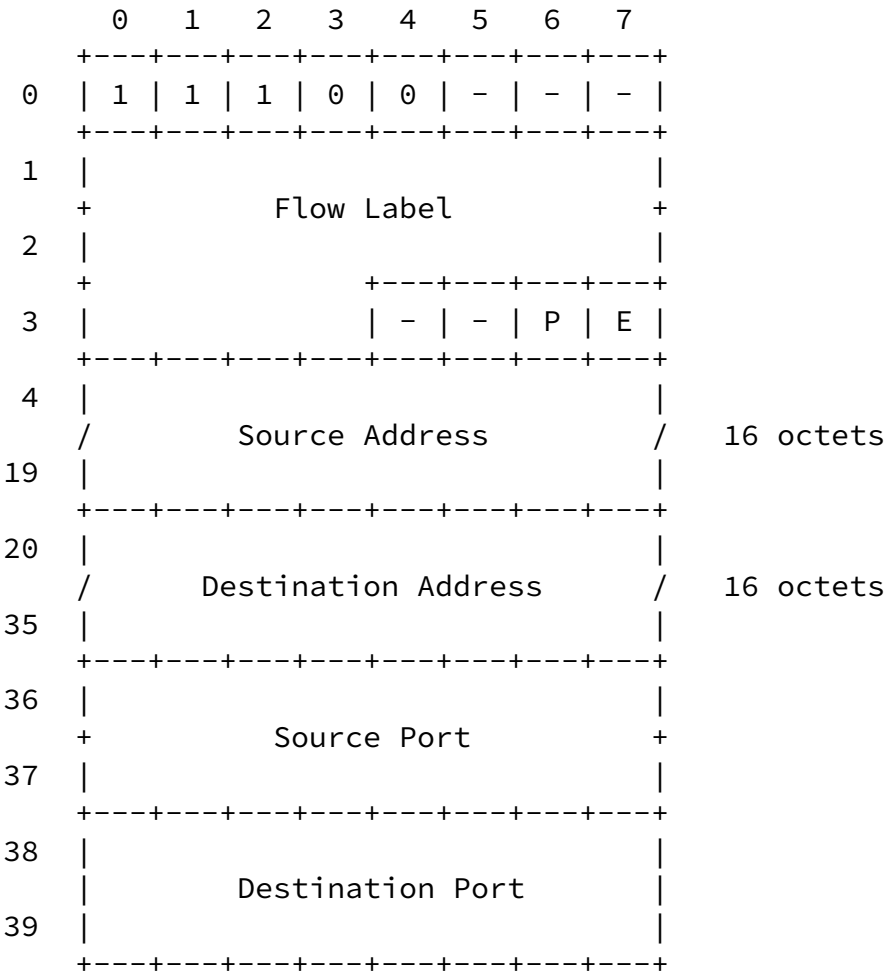
STATIC	11100
DYNAMIC	1111*
FEEDBACK	11101

All other bit-patterns indicate a COMPRESSED packet format. The meaning of those patterns are defined together with the specification of the packet formats.

#### [3.5.2.](#) Static information packet, initialization

The STATIC packet type is a packet containing no payload but only the header fields that are expected to be constant within the lifetime of the packet stream (classified as STATIC in chapter 3.2). The identification of the STATIC packet is done by the first five bits in the first octet. A packet of this kind MUST be sent once as the first packet from compressor to decompressor and also when requested by decompressor (see 3.5.6 and 3.7). The packet formats are shown below for IPv6 and IPv4, respectively. Note that some fields are only present in some of the STATIC packet types.0

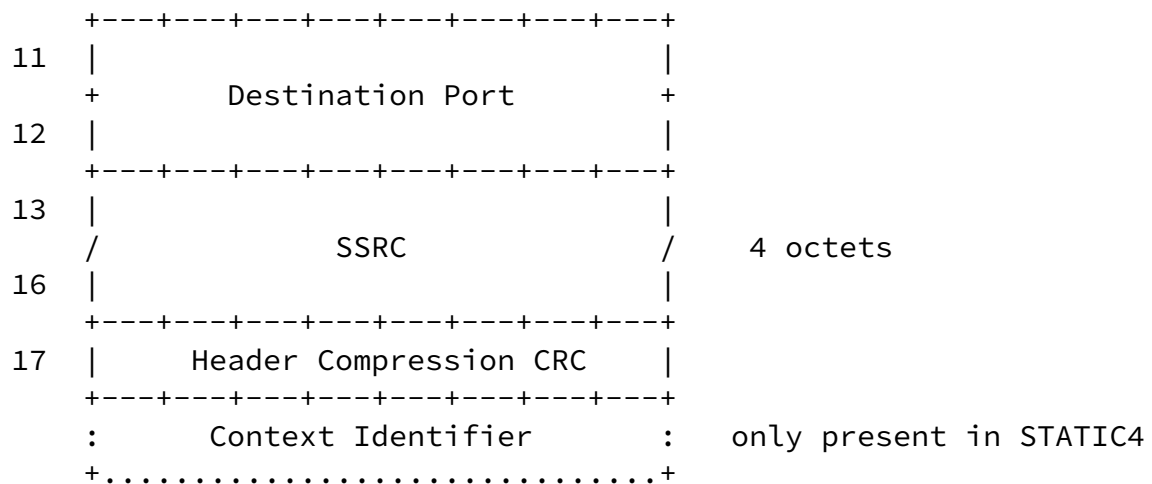
IPv6 (45-46 octets): STATIC1, STATIC2



40				
	/	SSRC	/	4 octets
43				
	+---+---+---+---+---+---+---+---+			
44		Header Compression CRC		
	+---+---+---+---+---+---+---+---+			
	:	Context Identifier	:	only present in STATIC2
	+.....+			

IPv4 (18-19 octets): STATIC3, STATIC4

		0	1	2	3	4	5	6	7								
		+---+---+---+---+---+---+---+---+															
0		1		1		1		0		0		F		P		E	
		+---+---+---+---+---+---+---+---+															
1																	
	/	Source Address								/	4 octets						
4																	
	+---+---+---+---+---+---+---+---+																
5																	
	/	Destination Address								/	4 octets						
8																	
	+---+---+---+---+---+---+---+---+																
9																	
	+	Source Port								+							
10																	



All fields except for the initial five bits, the padding bits(-), the Context Identifier and the Header Compression CRC are the ordinary IP, UDP and RTP fields (F=IPv4 May Fragment, P=RTP Padding, E=RTP Extension). The Header Compression CRC is calculated over the entire STATIC packet except the Header Compression field itself. The Header Compression CRC SHOULD be used in the decompressor to verify that the packet is correct. For more information please see [\[ROCCO\]](#).

Only one STATIC packet is sent at each occasion. If the decompressor receives DYNAMIC packets or compressed headers without having received a STATIC packet, the decompressor MUST request a STATIC packet.

### [3.5.3.](#) Dynamic information packet

The DYNAMIC packet type has a header containing all changing header fields in their original, uncompressed form, and carries a payload just as ordinary COMPRESSED packets. A DYNAMIC packets SHOULD be sent after the initial STATIC packet to set up the decompressor context for the first time. In addition, this packet is used whenever the

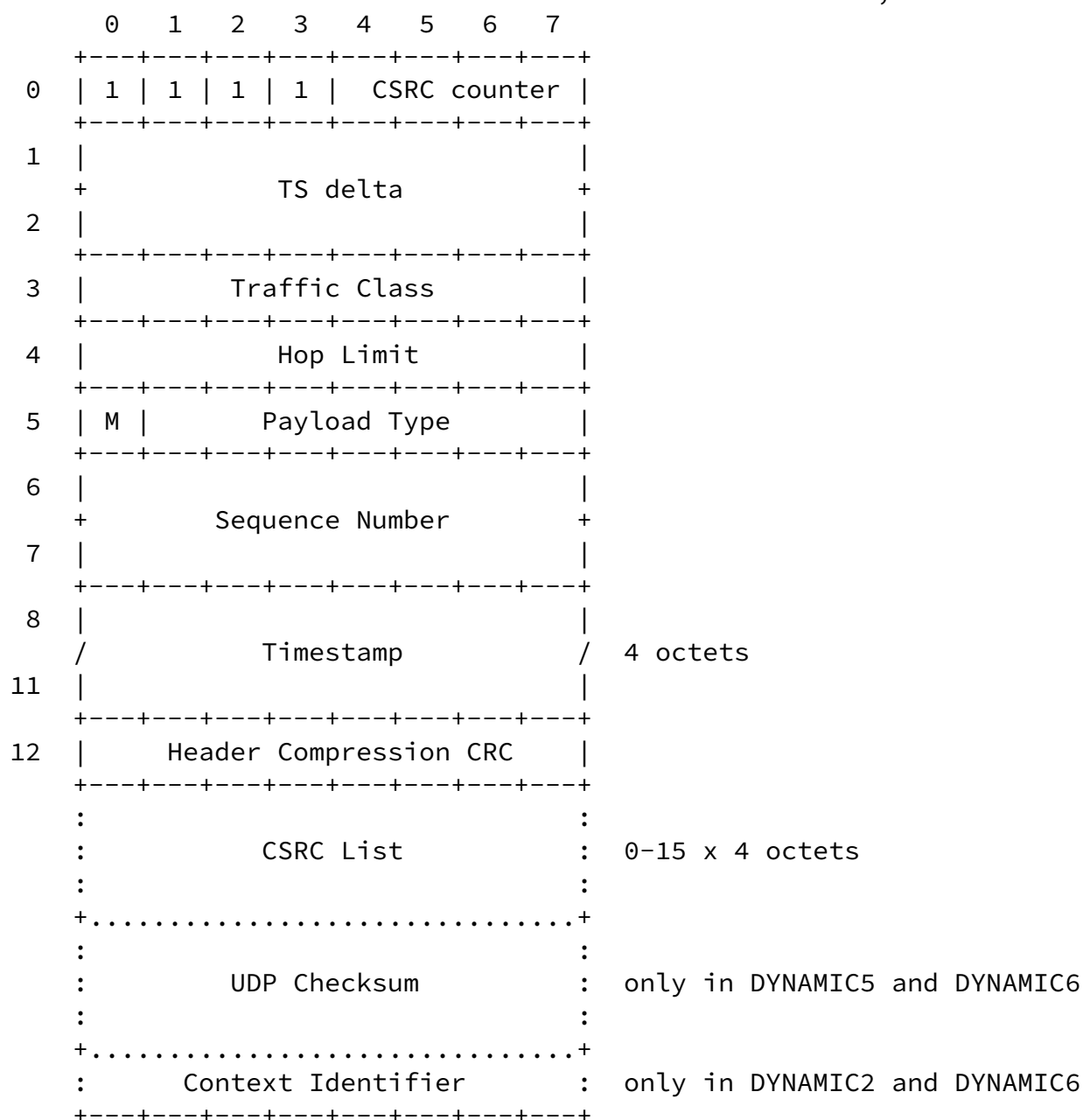
header fields change in a way that cannot be encoded in COMPRESSED packets.

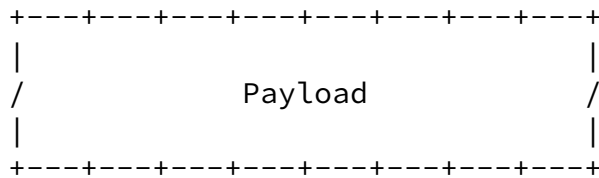
All fields except for the initial four bits, the Timestamp Delta , the Context Identifier and the Header Compression CRC are ordinary IP, UDP and RTP fields. The Timestamp Delta is the current delta

between RTP timestamps in consecutive RTP packets. If the Timestamp Delta is not known it SHOULD be initialized to 0. For information about how to calculate the header compression CRC please see [[ROCCO](#)].

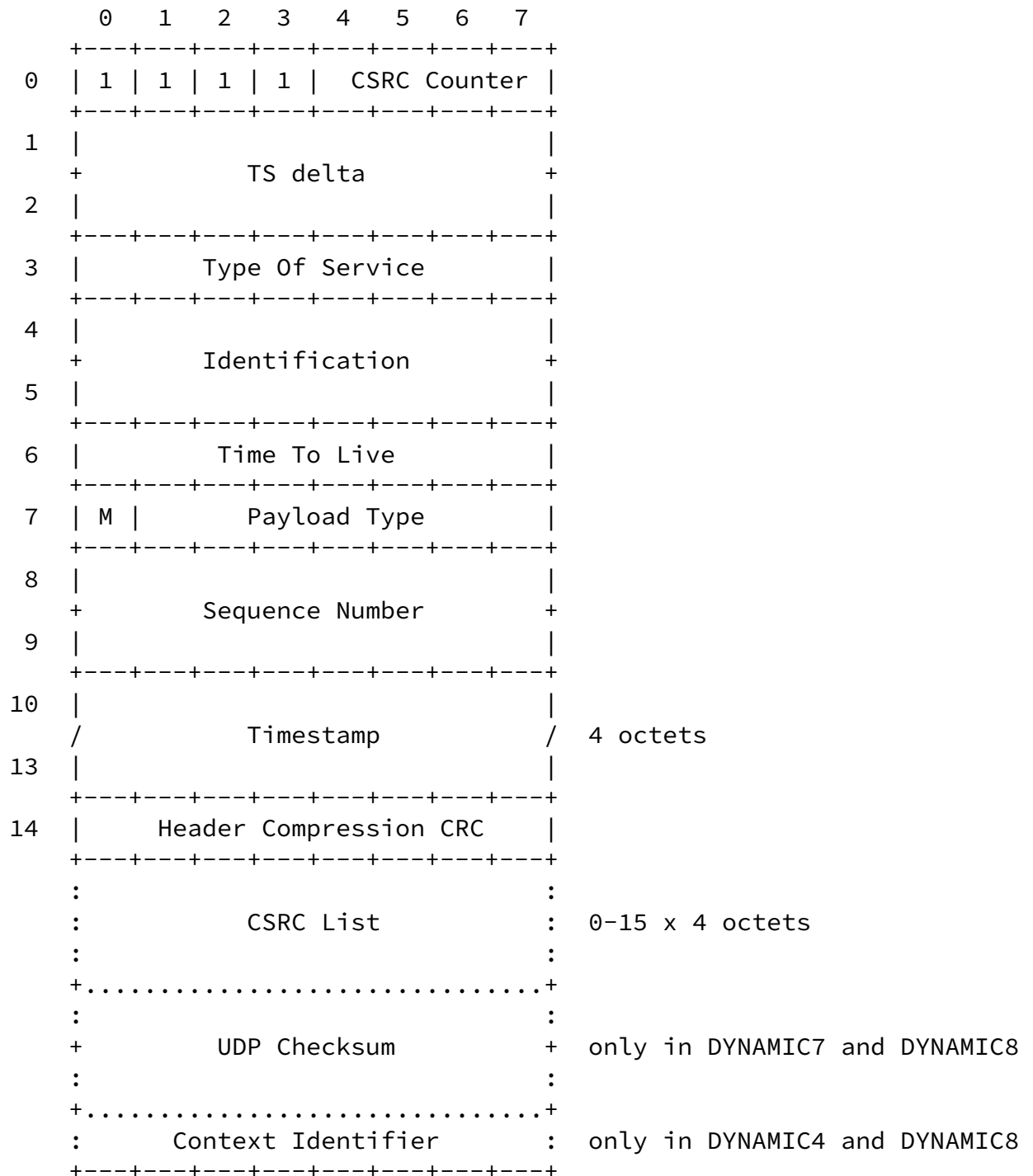
The packet formats are shown below for IPv6 and IPv4, respectively. Note that some fields are only present in some of the DYNAMIC packet types.

IPv6 (13-16 octets + CSRC List of 0-60 octets): DYNAMIC1, DYNAMIC2, DYNAMIC5, DYNAMIC6

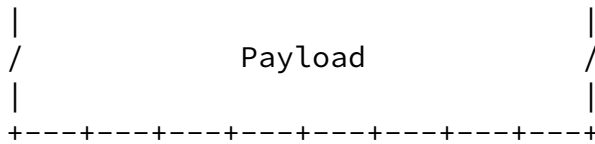




IPv4 (15-18 octets + CSRC List of 0-60 octets): DYNAMIC3, DYNAMIC4,  
DYNAMIC7, DYNAMIC8







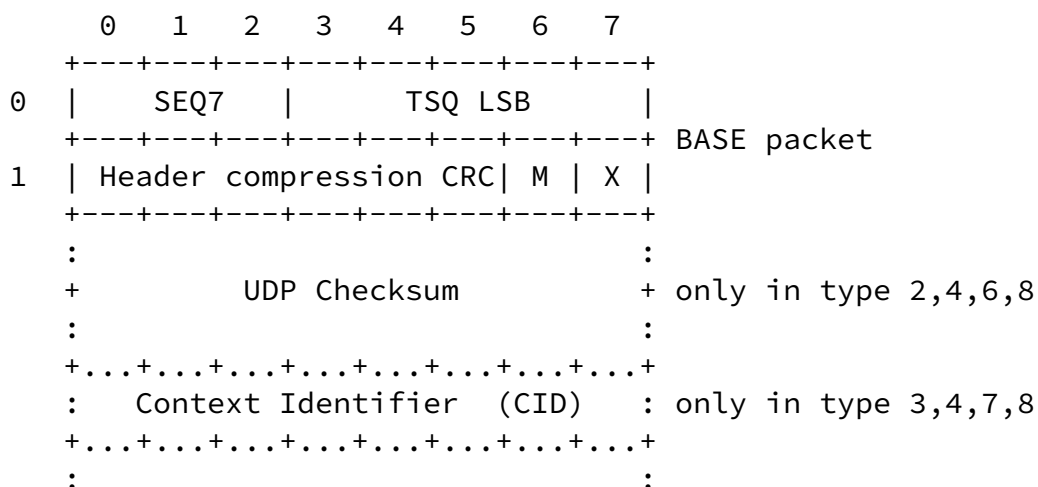
Each time a DYNAMIC packet is sent, several subsequent packets SHOULD also be DYNAMIC packets. This ensures that the update will succeed even when three consecutive packets are lost.

#### [3.5.4.](#) Compressed packets

The COMPRESSED packet type is the most commonly used one and it is designed to handle ordinary changes as efficiently as possible. When changes are regular, all information is carried in the base header. When changes are irregular it is possible to send more information in extensions to the COMPRESSED packet. These extensions are sent after the whole base header, i.e. after the Identification field if it is present.

The base header consists of five parts, a three bits SEQ field followed by a five bits TSQ field, a 6 bits Header Compression CRC, the RTP marker field and finally one bit indicating if header extensions are present. The COMPRESSED base-header formats are shown below. Note that some fields are only present in some of the COMPRESSED packet types.

Defines packet types: COMPRESSED1..COMPRESSED12



+	Identification LSB	+	only in type 5,6,7,8
:		:	
+...+...+...+...+...+...+...+...+			
	Identification		only in type 9,10,11,12
+...+...+...+...+...+...+...+...+			

The Header Compression CRC is computed over the original IP/UDP/RTP packet header. The CRC polynomial to be used is:

$$C(x) = 1 + x + x^3 + x^4 + x^6$$

For more information about how to calculate the Header Compression CRC please see [[ROCCO](#)].

### [3.5.5.](#) Extensions to compressed headers

Less regular changes of the header fields require an extension in addition to the base header. When there is an extension present in the COMPRESSED packet, it is indicated by the extension bit (X) being set. Extensions are of variable size depending on the information needed to be transmitted. However, the first three extension bits are used as an extension Type field for all extension formats. The extension can carry a TS delta field, a TS LSB field, a TSQ field, a SEQR field, a TSC field and a bit mask for additional fields. In the TSQ field, additional least significant bits of TSQ are sent to increase the timestamp range. The bits in the base header are always the least significant bits. The SEQR field is the LSB of the Sequence Number Residual defined in 3.4.2. Various bit mask patterns are possible and can consist of C,H,S,D,T and I. The interpretation of these bits are given at the end of this chapter.

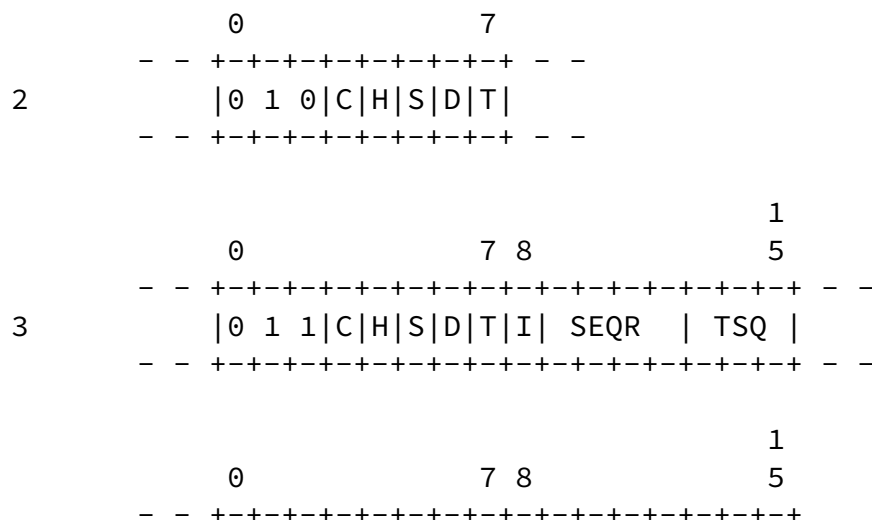
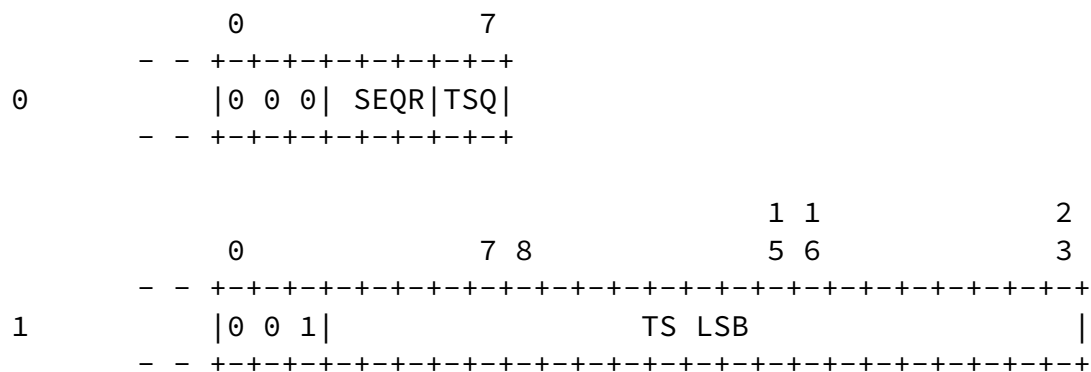
If both TSQ LSB and TS LSB is present in the same header the TSQ LSB is ignored and only the TS LSB bits is considered.

Three of the most common delta timestamps have been assigned to a special 2 bits TimeStamp Code (TSC). This code can be used in the COMPRESSED packet extension instead of a large Timestamp Delta field. The predefined values are:

TSC	Timestamp delta
=====	
00	3000 (30 Hz)
01	3003 (29.97 Hz)
10	3600 (25 Hz)
11	other, a 2 octet Timestamp delta field is added after the COMPRESSED packet extension.

The guiding principle for choosing extension type is to find the smallest header type that can communicate the information needed.

The defined extension types are shown below:



```

4      | 1 0 0 | C | H | S | D | I |      TS LSB      |
- - - +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+ - -
      |              TS LSB cont.              |
      +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+ - -

```

```

      0              7
- - - +--+--+--+--+--+--+--+--+ - -
5      | 1 0 1 | TSC | TSQ |
- - - +--+--+--+--+--+--+--+--+ - -

```

```

      0              7 8              1
      5
- - - +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+ - -
6      | 1 1 0 | TSC |      TS LSB      |
- - - +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+ - -
      |              TS LSB cont.              |
      +--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+ - -

```

TSC = Predefined TimeStamp delta Codes

00 = 3000

01 = 3003

10 = 3600

11 = other, followed by a 2 octet TS delta field  
after the TS LSB field

A bit mask indicating additional fields could include bits with the following meaning:

C - Traffic (C)lass / Type Of Service

H - (H)op Limit / Time To Live

S - Contributing (S)ources - CSRC

D - Timestamp (D)elta

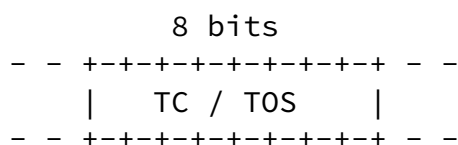
T - (T)imestamp LSB

I - (I)dentification

If any of these fields are included, they will appear in the order as listed above and the format of the fields will be as described below.

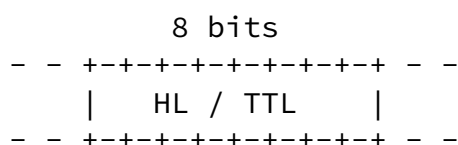
## C - Traffic Class / Type Of Service

The field contains the value of the original IP header field.



## H - Hop Limit / Time To Live

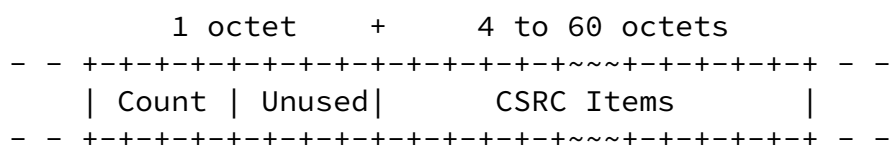
The field contains the value of the original IP header field.



## S - Contributing Sources

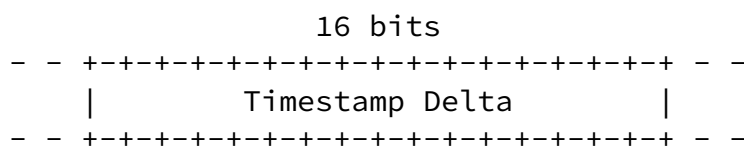
The CSRC field is built up of:

- a counter of the number of CSRC items present (4 bits)
- an unused field (4 bits)
- the CSRC items, 1 to 15 (4-60 octets)



## D - Timestamp Delta

The Timestamp Delta field is used to signal the timestamp delta used in the calculation of the downscaled timestamp defined in 3.4.5.

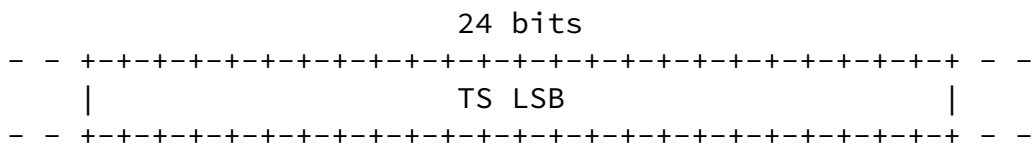


---

INTERNET-DRAFT      ROCCO Conversational Video Profiles      March 10, 2000

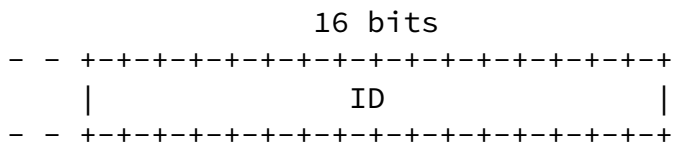
#### T - Timestamp LSB

The field contains the 24 least significant bits of the RTP timestamp.

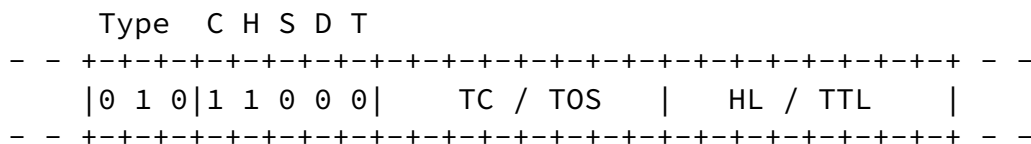


#### I - Identification

The field contains the value of the original IP header field.



An example where the TC/TOS and HL/TTL fields are present is a type 2 extension is shown below.



When information of any kind is sent in an extension, the corresponding information SHOULD also be sent in three subsequent packets (either as Extensions or as DYNAMIC packets) to satisfy the three-packet-loss requirement.

#### [7.5.6.](#) Feedback packets

Feedback packets are used by the decompressor to provide various types of feedback to the compressor. That could include active feedback to assure an error free function or passive feedback in case of invalidated context to request a context update from the compressor. The general feedback packet format is shown below:

	0	1	2	3	4	5	6	7	
	+---+---+---+---+---+---+---+---+								
FEEDBACK (GENERAL)	1	1	1	0	1	Type			
	+---+---+---+---+---+---+---+---+								
	: Context Identifier (CID)								: *)
	+...+...+...+...+...+...+...+...+								

The Type field tells what kind of feedback the packet corresponds to. For a description of the different available types please see [\[ROCCO\]](#).

### [3.6](#) Startup of timestamp encoding

In many cases the timestamp delta is not known when a new header compression session starts. It is then RECOMMENDED to use the following startup procedure:

- 1 - Set PCF=0;
- 2 - As long as TS has not changed, send TSQ=0
- 3 - As soon as TS change, send PCF and TSQ.  
This can be done in COMPRESSED extension number 5.

When the PCF is sent, several subsequent packets SHOULD also contain PCF information. This ensures that the update will succeed even packets are lost.

### [3.7.](#) Context update procedure

How to send and respond to various kinds of FEEDBACK packets are not defined in this document, but left to the implementers to decide. However, it is recommended to reduce both the number of requests and the number of corresponding updating packets. More guidelines for this issue will be included here when the implementation experience grows.

### [3.8](#) ROCCO over simplex links

For a discussion about how [\[ROCCO\]](#) can be used over simplex links please see [\[ROCCO\]](#).

## [4.](#) Compression performance

This section contains a short discussion about the compression performance of the [\[ROCCO\]](#) conversational video profile number 1005 (see table 3.2). Profile 1005 is designed for Ipv4 and assumes that IP Identification is assigned sequential, no UDP checksum is used and that only one packet stream is present. The size of the COMPRESSED packet is in this profile 2 octets.

If no packet reordering or packet losses is present only COMPRESSED packets will be sent after the startup procedure. This gives an average header size of 2 octets. All profiles defined in this document are designed to handle packet losses and single reordering

of packets. It is also possible for the compressor to extend the sequence number field and timestamp field, this to increase the possibility to decode these fields correctly and not lose the synchronization in the decompressor if many packet losses occur. Since the base COMPRESSED packet is designed to handle loss of up to 4 consecutive packets over the link, for most channels it is not necessary to use the extended fields very often. This gives that the increase of the average header size is very small even if many packet losses occur.

## [5.](#) Implementation status

The implementation of these ROCCO video profiles has started and will be ready within a few months. It will run in a testbed environment for realtime IP traffic over wireless channels.

## [6.](#) Security considerations

For security considerations please see [\[ROCCO\]](#).



## 7. Acknowledgements

We would like to thank Krister Svanbro, Hans Hannu (Ericsson) and Mikael Degermark (Luleå University) for close collaboration and fruitful discussions during the work with these profiles. We also want to thank Goran Roth and Joel Askelof for valuable advice during the work with this document.

## 8. Intellectual property considerations

This proposal in conformity with IETF [RFC 2002](#)

This is to affirm that Telefonaktiebolaget LM Ericsson and its subsidiaries, in accordance with corporate policy, will offer patent licensing for submissions rightfully made by its employees which are adopted or recommended as a standard by your organization as follows:

If part(s) of a submission by Ericsson employees is (are) included in a standard and Ericsson has patents and/or patent application(s) that are essential to implementation of such included part(s) in said standard, Ericsson is prepared to grant - on the basis of reciprocity (grantback) - a license on such included part(s) on reasonable, non-discriminatory terms and conditions.

For the avoidance of doubt this general patent licensing undertaking applies to this proposal.

Martensson, Einarsson, Jonsson

[Page 27]

---

INTERNET-DRAFT      ROCCO Conversational Video Profiles      March 10, 2000

## 9. Authors' addresses

Anton Martensson	Tel: +46 8 404 38 81
Ericsson Radio Systems AB	Fax: +46 8 757 55 50
Torshamnsgatan 23	EMail: anton.martensson@era.ericsson.se
SE-164 80 Stockholm, Sweden	

Torbjorn Einarsson	Tel: +46 8 757 23 46
--------------------	----------------------

Ericsson Radio Systems AB  
Torshamnsgatan 23  
SE-164 80 Stockholm, Sweden

Fax: +46 8 757 55 50  
Mobile: +46 70 673 34 24  
EMail: torbjorn.einarsson@ericsson.com

Lars-Erik Jonsson  
Ericsson Erisoft AB  
Box 920  
SE-971 28 Lulea, Sweden

Tel: +46 920 20 21 07  
Fax: +46 920 20 20 99  
Mobile: +46 70 554 82 71  
EMail: lars-erik.jonsson@ericsson.com

## 10. References

- [UDP] Jon Postel, "User Datagram Protocol", [RFC 768](#), August 1980.
- [IPv4] Jon Postel, "Internet Protocol", [RFC 791](#), September 1981.
- [IPv6] Steven Deering, Robert Hinden, "Internet Protocol, Version 6 (IPv6) Specification", [RFC 2460](#), December 1998.
- [RTP] Henning Schulzrinne, Stephen Casner, Ron Frederick, Van Jacobson, "RTP: A Transport Protocol for Real-Time Applications", [RFC 1889](#), January 1996.
- [ROCCO] Lars-Erik Jonsson, Mikael Degermark, Hans Hannu, Krister Svanbro, "RObust Checksum-based header COmpression (ROCCO)", Internet-Draft (work in progress), March 2000. <[draft-jonsson-robust-hc-04.txt](#)>
- [H.261] "Video codec for audiovisual services at p x 64 kbit/s," ITU-T Recommendation H.261, 1993.
- [H.263v1] "Video coding for Low Bit Rate Communication," ITU-T Recommendation H.263, March 1996.
- [H.263v2] "Video coding for Low Bit Rate Communication," ITU-T Recommendation H.263, January 1998.
- [MPEG-4] ISO/IEC 14496-2:1999, "Information technology - Coding of audio-visual objects - Part2: Visual", December 1999.
- [RFC2032] Turletti, T. and C. Huitema, "RTP Payload Format for H.261 Video Streams", [RFC 2032](#), October 1996.
- [RFC2429] C. Bormann, L. Cline, G. Deisher, T. Gardos, C. Maciocco, D. Newell, J. Ott, G. Sullivan, S. Wenger, C. Zhu, "RTP Payload Format for the 1998 Version of ITU-T Rec. H.263 Video (H.263+)", [RFC 2429](#), October 1998.

INTERNET-DRAFT      ROCCO Conversational Video Profiles      March 10, 2000

This Internet-Draft expires September 10, 2000.

