Video Quality Evaluation for Wireless Transmission with Robust Header Compression

F. Fitzek S. Hendrata P. Seeling M. Reisslein[*]

acticom GmbH – mobile networks
R & D Group
Germany
[fitzek|hendrata]@acticom.de

Arizona State University
Department of Electrical Engineering
USA
patrick.seeling|reisslein@asu.edu

1 July 2003

Technical Report acticom-03-003

IP Header compression mechanisms have always been an important topic for the research community to save bandwidth in the Internet. Due to the high license fees of 3G bands and the upcoming integration of IP based multimedia services, it is of particular importance to reduce the IP header overhead even in the wireless format. Reducing the IP overhead gives the network providers the possibility for a faster return of investment on their 3G networks and simultaneously enables real–time services by improving the latency of the IP packets over bandwidth limited links. Many compression methods exist already but they are either not designed for multimedia services, or not robust in the presence of error–prone links and therefore not suitable for wireless communication. For wireless environments robust header compression was introduced. Robust header compression was standardized by the Internet Engineering Task Force in RFC 3095 and will be an integral part of the 3GPP–UMTS specification. This compression scheme was designed to operate in error–prone environments by providing error detection and correction mechanisms in combination with robustness for IP based data streams. A connection oriented approach removing packet inter– and intra–dependencies reduces the IP header significantly. This paper gives a solid performance evaluation for robust header compression showing both the bandwidth savings for the IP protocol stack and the quality of services at the application layer by the means video services.

[*] The work of M. Reisslein is supported in part by the National Science Foundation through Grant No. Career ANI-0133252 and Grant No. ANI-0136774. Any opinions, findings, and conclusions or recommendations expressed in this material are these of the authors and do not necessarily reflect the views of the National Science Foundation.

Contents

List of Figures

List of Tables

1 Introduction

Existing wireless networks of the second generation (also know as 2G) are mostly circuit switched and have been developed and optimized for speech transmission. Wireless networks of the 3rd generation (3G) will offer a wide range of **I**nternet **P**rotocol (*IP*) based multimedia applications (such as video and gaming services) [8], which require more bandwidth than speech services and are highly delay sensitive. Multimedia applications often use **R**eal **T**ime **P**rotocol (*RTP*), **U**ser **D**atagram **P**rotocol (*UDP*) and IP. Each of the protocol layer adds a significant header overhead. Therefore, the higher bandwidth requirement is caused by the application itself and the IP overhead. While the compression of multimedia payload is mostly sufficient or even excellent (such as voice and video codecs), the most promising compression gain can be yield focusing on the IP header. Due to the high license fees of 3G bands and the migration of IP based services into the wireless format, it is necessary to reduce the IP header overhead which can be even greater than the payload for some services. IP header compression mechanisms have always been an important part of saving bandwidth over bandwidth limited links. In particular on delay sensitive voice application the header overhead is not negligible. E.g., using the LPC codec with 5.6 kbps, the header overhead of Ipv4 can obtain 74% and of Ipv6 even up to 81% of the IP packet. This gives a first impression how large the potential bandwidth saving gain is for voice services. This example is stressed in many other papers. The calculation of potential voice service savings is easy, because of the static audio frame size. For video services the frame size depends even on the content of the video scene and therefore the savings of a header compression scheme is hard to predict.

Many header compression schemes exist already but most of them are not suited for the wireless format and therefore will fail in this kind of environment. Besides the bandwidth savings even the latency of the IP packets can be improved by header compression schemes. This effect is even larger for links with small and limited bandwidth. A third effect of header compression schemes is that because of the smaller transmission time per packet, the error probability decreases. But this effect depends on the characteristic of the wireless link. For multimedia services in wireless environments **RO**bust **H**eader **C**ompression (*ROHC*) [3] was introduced. ROHC was standardized by the Internet Engineering Task Force in RFC 3095 [3] and will be an integral part of the 3GPP–UMTS specification. This compression scheme was designed to operate in error–prone environments by providing error detection and correction mechanisms in combination with robustness for IP based data streams. A connection oriented approach removing packet inter– and intra–dependencies yields a significantly reduction of the IP header. The remainder of the paper is structured in the following manner: First we give a motivation for header compression schemes. After having calculated the upper bound gain of header compression schemes for some multimedia services, we give an overview of existing header compression schemes. In Section 2 we present the important parts of ROHC understand this document. A solid performance evaluation is presented in Section 4 showing both the bandwidth savings for the IP protocol stack and the quality of services at the application layer by the means of video services.

1.1 Motivation for Header Compression

The motivation for IP based header compression is based on the fact that headers have significant redundancy. In Figure 1 the combined header for a real–time multimedia stream with IPv4 consisting of 40 bytes is given. This includes the 20 bytes IPv4 header, the 8 bytes UDP [12] header and the 12 bytes RTP [9] header. The header for IPv6 requires in total 60 bytes.

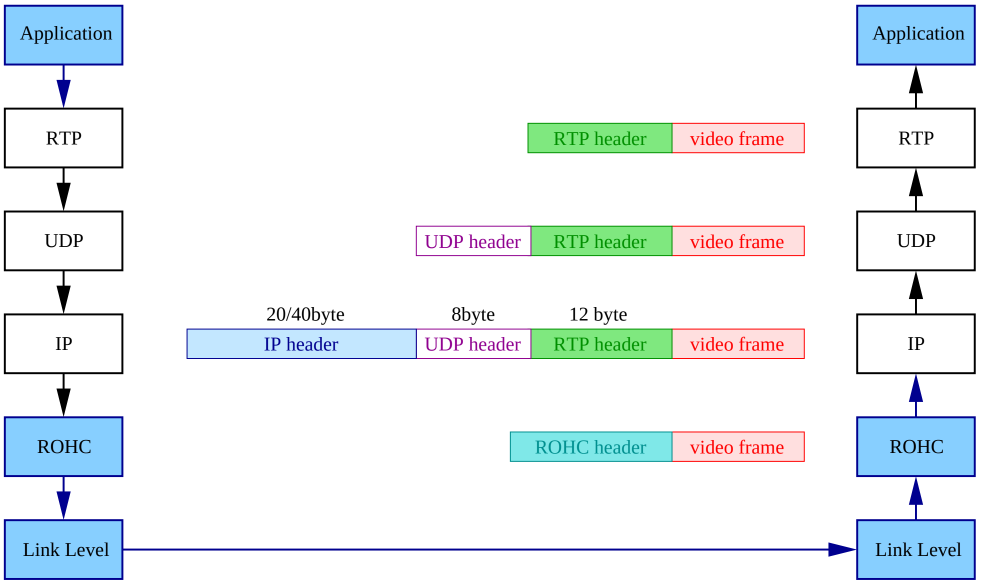


Figure 1: Header structure and protocol stack with relevant ROHC protocol layers IP, UDP, and RTP.

Redundancy exists among the different headers (IP, UDP, and RTP) but in particular between consecutive packets belonging to the same IP flow. In Figure 2 the header fields for RTP/UDP/IP packets are given in dependency on its dynamics. The header fields can be distinguished between *non changing* and *changing*. The *non changing* group consists of static, static-known, and inferred header fields and are more or less easy to compress. A large portion of the header field are static or static-known and therefore can be compressed easily or will not be sent at all after the first successful transmission between sender and receiver. Other header fields are referred to as inferred. These fields can be inferred from other header fields and are also easy to compress. The *changing* group consists out of not-classified-a-priori, rarely-changing, static or semi-static changing, and alternating changing header fields. These kinds of header fields are more difficult to compress and it depends on the header compression scheme how the compression is done.

1.2 Potential Savings of Header Compression

To get a notion about the possible savings by header compression schemes we compute the optimum bounds for video traffic. For this first and simple calculation, we assume that no IP overhead (including the RTP and UDP headers) is used at all. Equation 1 shows how we calculate the potential saving S_i for each IP packet i.

$$S_i = 1 - \frac{Packet(i)}{Header + Packet(i)} = \frac{Header}{Header + Packet(i)} \tag{1}$$

For this first simple example, the savings depend only on the mean packet length. The packet length depends on the service type used. While voice and audio services generates packets with static lengths, video services are characterized by variable packet length. While the packet length characteristic is given by the service type, the mean packet length depends on the codec used. The mean saving $\overline{S}$ for the whole stream, given in Equation 2, reflects the amount of bandwidth

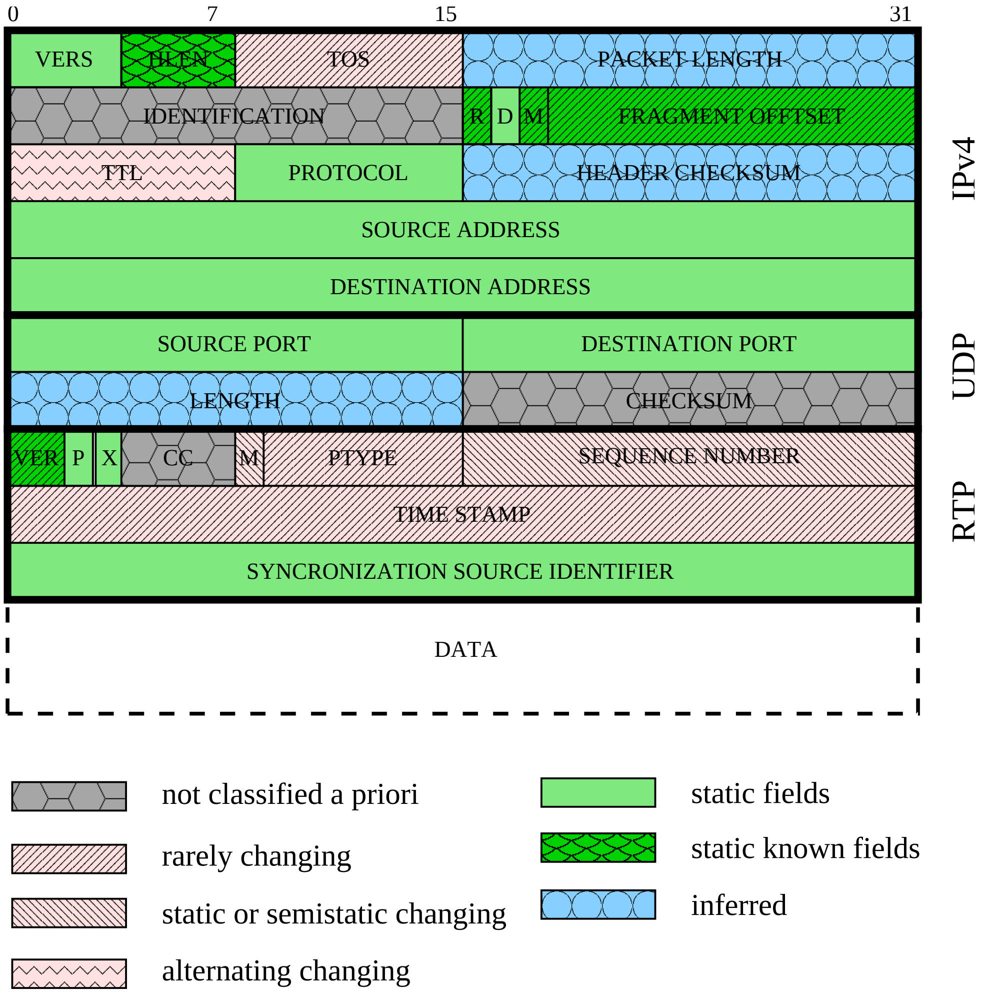


Figure 2: Header fields for RTP/UDP/IP packets (Version 4) with the appropriate dynamics.

that a wireless network provider can save by a header compression scheme for an IP session consisting of N frames.

$$\overline{S} = \frac{1}{N}\sum_{i=1}^{N} S_i \tag{2}$$

In comparison to voice streams with fixed frame sizes, the frame sizes of a video stream vary over time [18]. The size of the video frames depend on the content of the video sequence and the used encoder settings. We encoded the generally accepted video reference streams such as *formen*, *container*, *akiyo*, and *silent* (given in Table 1) in the QCIF format (176x144 pixel) with the H.26L encoder [19, 11] using the IBBPBBPBBPBB group of picture (GoP) structure. The video quality is set to 30 for all streams, which seems to be a high quality level as there is not visible degradation in contrast to the original sequence. Furthermore we took the *container* sequence and encoded it with different quality levels. A theoretical upper bound for the mean potential savings $\overline{S}$ for H.26L [20] encoded video streams are given in Table 2 for the video format QCIF.

video sequence name	quality	frames
container	10	300
container	20	300
container	30	300
container	40	300
container	51	300
bridge (close)	30	2001
carphone	30	382
claire	30	494
foreman	30	400
grandma	30	870
highway	30	2001
mother	30	961
news	30	300
salesman	30	449
silent	30	300

Table 1: Transmitted video streams.

Table 2: Theoretical upper bound savings in terms of bandwidth for H.26L video traffic (QCIF).

	quality	mean bit rate	IPv4	IPv6
video sequence name		[kbps]	$\overline{S}$ [%]	$\overline{S}$ [%]
container	10	855.0	1.1	1.6
container	20	213.0	4.3	6.3
container	30	65.8	12.7	17.9
container	40	24.1	28.5	37.5
container	51	9.1	51.0	61.0
bridge close	30	69.9	10.3	14.6
carphone	30	135.4	6.6	9.6
claire	30	44.3	17.8	24.5
foremen	30	121.9	7.28	10.5
grandma	30	56.4	14.5	20.3
highway	30	57.2	12.3	17.3
mother	30	66.4	12.6	17.8
news	30	100.9	8.7	12.5
salesman	30	81.5	10.5	15.0
silent	30	101.7	8.6	12.4

First of all we want to note that all video sequences differ in the potential savings. In contrast to voice services, the savings for video services depends also on the content. As mentioned before the potential savings for IPv6 are greater than for IPv4 because of the IP header length (IPv4 with 40 byte and IPv6 with 60 byte). Obviously, the smaller the IP packets are the higher is the impact of the IP overhead and therefore the saving $\overline{S}$. The highest savings are found for low video quality and IPv6 with 61% of potential savings. For the video sequences of quality level 30, the savings vary from 6.6% to 24.5%. Even if the gain is smaller than for voice services, these values motivate us to investigate the realistic savings possible with ROHC.

1.3 Related Work On Header Compression Schemes

The first proposed IP header compression scheme **C**ompressed **T**ransport **C**ontrol **P**rotocol (*CTCP* or *VJHC*) for the Internet was introduced by Van Jacobson in 1990 [10] as RFC1144 and focuses on the TCP protocol. VJHC processes TCP and IP headers together to achieve better compression ratios. A second benefit from the combined procession is the reduced complexity of the algorithms employed. VJHC is based on delta coding. The differences between two header packets are referred to as the delta. Instead of transmitting the whole header, VJHC transmits only the delta. Due to this approach high compression is achieved. Simultaneously vulnerability comes along with this approach. In case only one delta coded header got corrupted, all following packets will be error–prone. Error detection is based on the framing process and the TCP–checksum. No error checks are performed within the decompressor. This approach yields the advantage of no necessary signaling between compressor and decompressor. The disadvantage is the sensitivity error–prone links as found in [7, 13, 4, 6, 5].

In [13], robustness at the cost of less compression was introduced by Perkins. The delta–coding for the neighboring packets has been replaced by a reference frame. Several consecutive packets are aggregated to a frame. The first packet of a frame is sent uncompressed and the following packets use the delta coding referring to this first packet. Obviously the differences to packets at the end of a frame are larger than for those at the beginning. Therefore the compression gain is not as high as for VJHC. The advantage of this approach is the usage of shorter delta coding ranges. In case packets get corrupted, this will not lead necessarily to the loss of synchronism. This is a clear advantage over VJHC. An optimization for the header compression of Perkins was introduced by Calveras in [6, 5]. The presented approach *CAHC* minimizes the overhead by adapting the frame length as a function of the channel state.

Header Compression schemes introduced by Perkins and Calveras obtain base updates by sending both compressed and uncompressed headers. Whenever one of such base updates is lost due to transmission errors, the synchronism between compressor and decompressor is lost and performance degradates in contrast to VJHC. To improve robustness against packet losses in such cases, Rossi etal. proposed modification to the base update procedure in [15, 16].

2 Robust Header Compression

RObust **H**eader **C**ompression (*ROHC*) [3] in its original specification as in RFC 3095 is a header compression scheme with profiles for three protocol suites: RTP/UDP/IP, UDP/IP and ESP/IP. In case any other protocol suite is used ROHC will not perform compression at all (uncompressed profile) but there are other profiles in development to support more protocol suites (IP only, TCP/IP). As shown in Figure 3, ROHC is located in the standard protocol stack between the IP–based network layer and the link layer. The need for saving bandwidth is limited to the

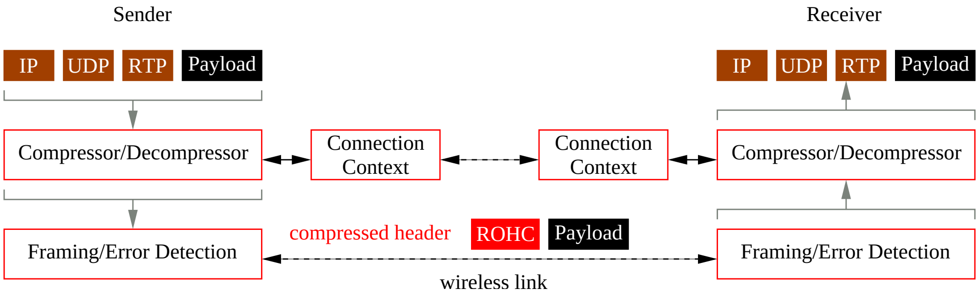


Figure 3: Transmission of compressed packets between ROHC compressor and decompressor.

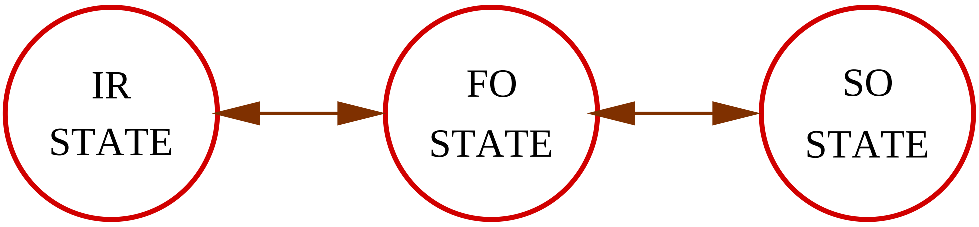


Figure 4: Compressor States.

wireless link from the wireless terminal to the radio base station. So the compression must work only between these two terminals and for the rest of the Internet this operation remains invisible. In the simplest configuration in one of the wireless terminals there must be a compressor and on the other part a decompressor. ROHC controls the interaction between these two instances in order to achieve two goals: 1.) The network providers desire the significant bandwidth gain obtainable by reducing the IP header to a shorter ROHC header. 2.) On the other hand it is necessary to ensure a QoS acceptable from the customers. The compressor, in fact, can sacrifice the bandwidth profit in order to keep the decompressor synchronized even if errors occur on the link. So ROHC does not work always on the top of its compression capacity. On the contrary different levels of compression, called states, constitutes a new robust solution against the perils of the wireless links.

2.1 Context and States

ROHC uses a flow-oriented approach to compress packets. Each flow is mapped to a context at the compressor and the decompressor and is identified by a context identifier (CID). A context is a set of state variables and contains (among other variables) the static and dynamic header fields that define a flow.

The ROHC compressor and decompressor can each be regarded as a state machine with three states. Compressor and decompressor start at the lowest state which is defined as ”no context established”, i.e., compressor and decompressor have no agreement on compressing or decompressing a certain flow. Thus, the compressor needs to send a ROHC packet containing all the flow and packet information (static and dynamic) to establish the context. This packet is the largest ROHC packet that the compressor can send. In the second state, the static part of a context is regarded as established between compressor and decompressor while the dynamic part

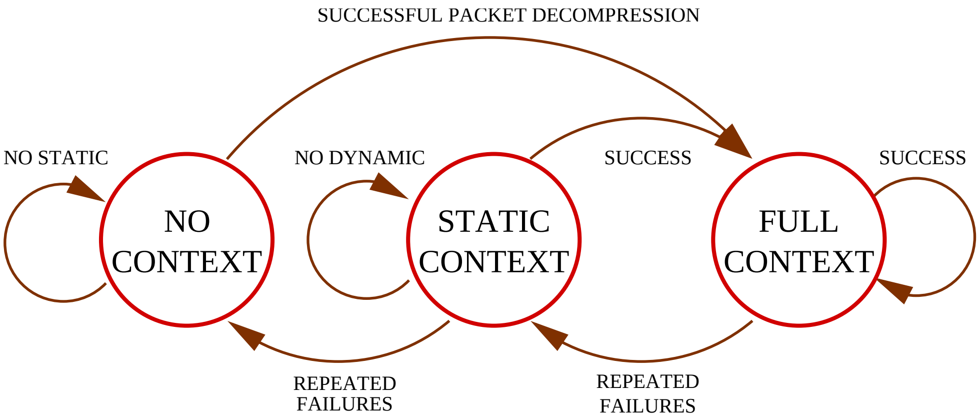


Figure 5: Decompressor States for all modes.

is not. In this state, the compressor sends slightly larger ROHC header than when it is in the third state where the static as well as the dynamic part of a context are established. Fallbacks to lower states occur when the compressor detects a change in the dynamic or static part of a flow, or when the decompressor detects an error in the dynamic or static part of a context.

The compressor endeavors to operate as long as possible in the highest state under the constraint of being confident that the decompressor has enough and updated informations to decompress the headers correctly. Otherwise it must transit to a lower state to prevent context damage and to avoid context error propagation.

2.2 Compression of header fields

Compression of the static part of headers for a flow is trivial since they only need to be transmitted at context establishment and then stay constant. More sophisticated algorithms are needed for compression of the dynamic part.

ROHC basically uses two algorithms to compress and decompress dynamic header fields: "Self-describing variable length values' and Windowed Least Significant Bits (W-LSB) encoding. The first one reduces the number of bits needed to transmit a field upon the actual value of that field (low values need fewer bits). The latter algorithm, after parameterizing to the dynamic change characteristic of the header field to be compressed, yields the minimum bits needed to be transferred to reconstruct the new value from an old value that the decompressor received previously. Especially the W-LSB compression algorithm in combination with an elaborated scheme to protect sensible data in ROHC compressed headers contribute to the robustness of ROHC.

2.2.1 Compressor States

As given in Figure 4 the three compressor states are the Initialization and Refresh state (IR), the First Order state (FO) and the Second Order state (SO). In the IR state there is no compression gain so it is necessary to transit to a higher state as soon as possible. When confident of its success to establish a context, the compressor can change to the SO state immediately. In the SO state, only the transmission of a sequence number is necessary and the value of all other header fields are interred from it. These ROHC headers are the smallest ones with in general one byte

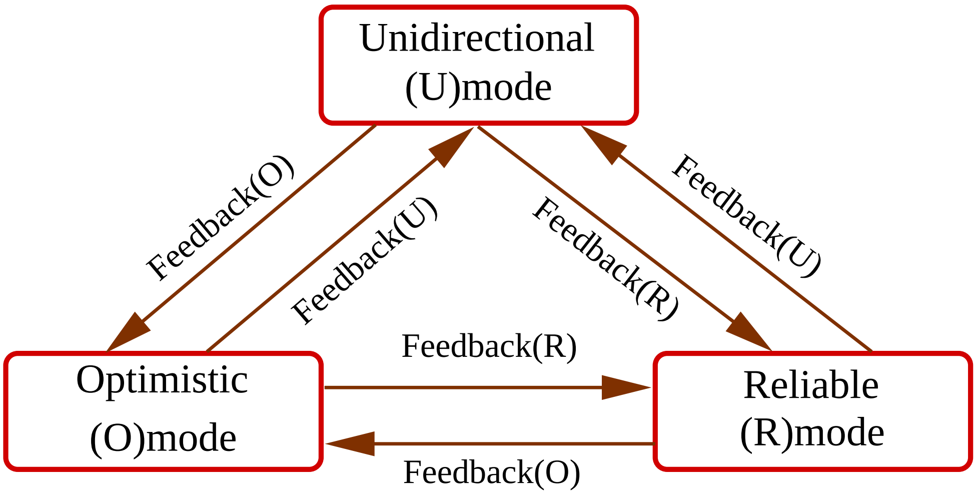


Figure 6: Modes transitions.

size. If an irregularity in a flow to be compressed occurs, then the compressor falls back to to the FO state. Depending on the irregularity different ROHC headers with sizes of two, three or more bytes are used in this state. If the flow returns to a regular behavior again, the compressor transits up to the SO state.

2.2.2 Decompressor States

The three decompressor state names depicted in Figure 5 refer clearly to the grade of context completeness. In the No Context (NC) state, the decompressor lacks the static and dynamic part of a context, so it can decompress only IR packets, i.e. packets sent in the IR compressor state. In the Static Context (SC) state, the decompressor lacks only the dynamic part (fully or partially) and therefore needs packets that contain information on dynamic header fields in order to complete the context again. The decompressor usually works in the Full Context state which is reached after the entire context has been established. In case of repeated failures in decompression attempts the decompressor always transits to the SC state first. Then it often is sufficient to rightly decompress an FO packet to recover to the FC state. Otherwise, continuing receive errors lead to the transition of the decompressor to the NC state.

2.3 Modes and States Transitions

To offer the ability to run over different kind of links, ROHC operates in one of three modes: Unidirectional, Bidirectional Optimistic and Bidirectional Reliable. Similarly to the states, ROHC must start at the lowest mode (unidirectional) but then it can transit upwards if the link is bidirectional. Contrary to the states, in fact, modes are not related with the compression level but they determine which actions ROHC must perform in every state and in state transitions according to the link characteristics in terms of feedback and context updates. Mode transitions are established by the decompressor. It can insert a mode transition request in a feedback packet indicating the desired mode as Figure 6 illustrates.

2.3.1 Unidirectional Mode (U–Mode)

This mode is conceived for links not having a return channel. There is no way for the compressor to be certain whether the decompressor is having the correct context and thus decompressing correctly. So in order to give the decompressor a chance to update its context in case of context errors, the compressor periodically falls back to the FO and IR states. Furthermore, before transiting to a higher state, the compressor sends important updates to the context multiple times to be more robust against single packet errors. This is the least robust and least efficient mode among the three ROHC modes.

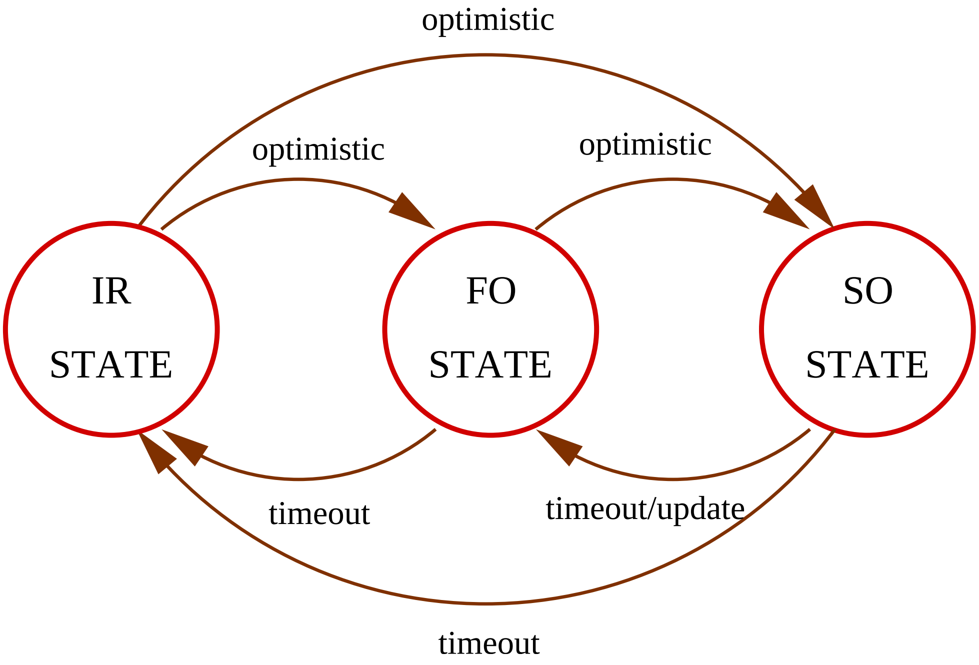


Figure 7: Compressor States transitions for Unidirectional mode.

2.3.2 Bidirectional Optimistic Mode (O–mode)

In this mode, the compressor uses feedback received on the return channel from the compressor to react on successful or unsuccessful transmission of packets. This feedback messages are recovery requests (NACKs) and sometimes context update acknowledgments (ACKs) for important parts of the context as shown in Figure 8. The compressor is thus able to optimize compression efficiency by avoiding sending redundant information as in the unidirectional mode.

2.3.3 Bidirectional Reliable Mode (R–mode)

With this mode, the behavior of the compressor and decompressor are even closer coupled than with the optimistic mode. For example, context updating packets have to be acknowledged by the decompressor before the compressor can refer to the updated context. Every context update or upward transition need the reception of a decompressor ACK and the rare NACKs provide a quick context recovery. Therefore, the compressor always knows in which state the

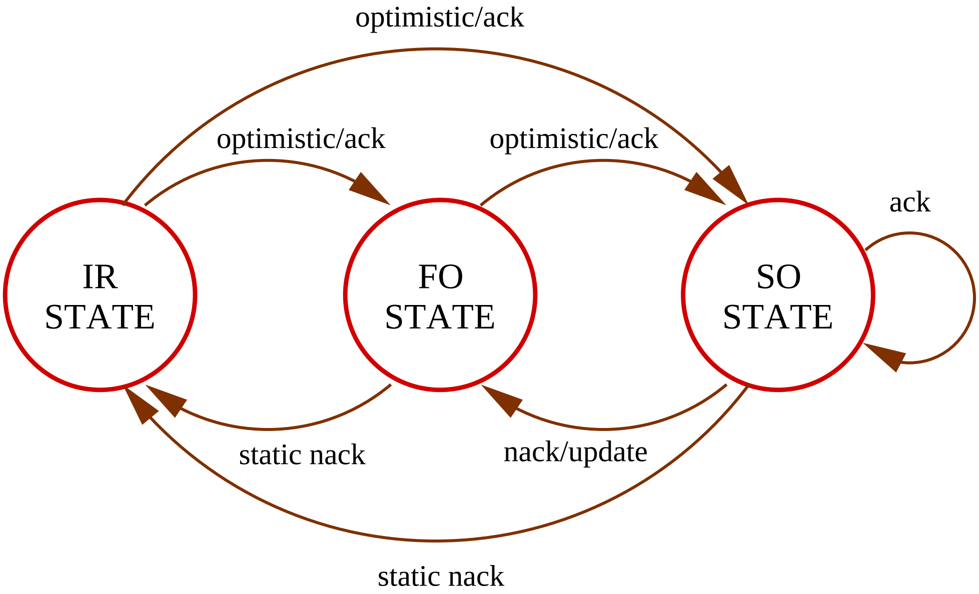


Figure 8: Compressor States transitions for Bidirectional Optimistic mode.

decompressor is and when to make a state transition. The goal is to reduce the remaining packet error probability to increase the reliability of the compressed transmission.

3 Methodology of the Performance Evaluation

To conduct the performance evaluation of ROHC the test–bed depicted in Figure 10 is used. Two SIEMENS Scenic PRO M7 PCs running a SUSE Linux 7.3 are connected via a network with each other. The network can be either a normal Ethernet connection with simulated errors or real wireless components such as GPRS, GSM, and DECT in combination with the PPP extension [2] for ROHC. On both machines the acticom ROHC implementation [1] is used as a kernel model.

Within the performance evaluation of ROHC we distinguish between three qualities, namely the Network Quality the Objective Quality, and the Subjective Quality. While the network quality reflects the provider's view, the objective and the subjective quality reflects the customer's view. In Figure 11 the relationship between the quality levels is given.

- The Network Quality focuses only on the efficiency of the header compression scheme. Thus, one might say that the network quality reflects the network provider's view (How much money i can save?). Within this paper we have three performance features for the network quality: 1.) Header Compression Efficiency: We define the header compression efficiency HCE per packet as the ratio of saving in the header field and the uncompressed header:

$$HCE = 1 - \frac{H_C}{H_U} = \frac{H_U - H_C}{H_U}, \tag{3}$$

  2.) Total Compression Efficiency:

$$TCE = 1 - \frac{T_C}{T_U} = \frac{T_U - T_C}{T_U} \tag{4}$$

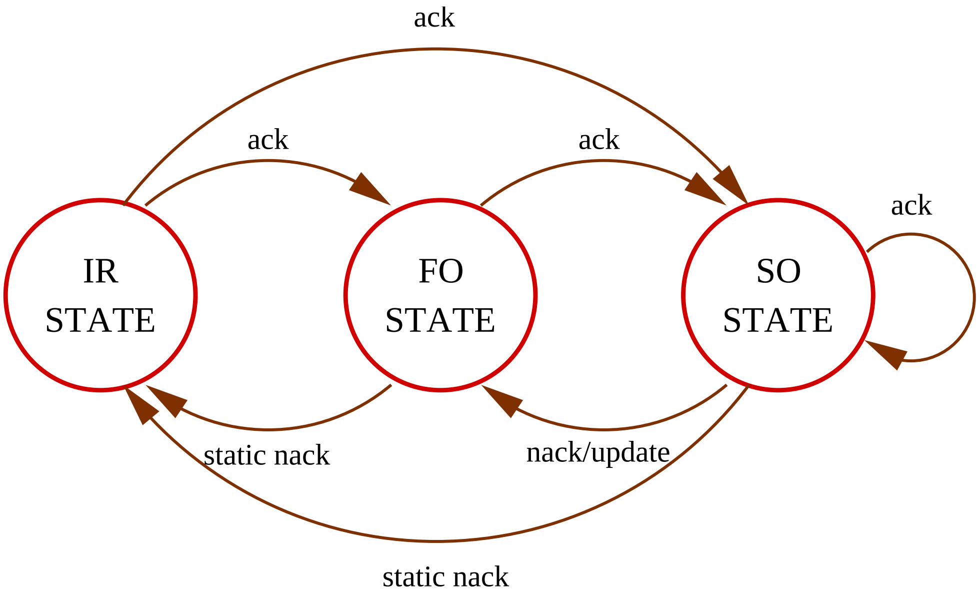


Figure 9: Compressor States transitions for Bidirectional Reliable mode.

3.) Compressed Bandwidth.

All three performance features are measured with the NetMeter tool (see Figure 12).

- The Objective Quality reflects the customer's view (see Figure 11). We want to distinguish between the Objective Video Quality and the Objective Voice Quality.

Figure 13 illustrates the transmission chain of the video streams. The *original* video sequences in the YUV format are encoded with H.26L at the sender–side[1]. In case H.26L is used, the RTP headers can be either generated by the encoder itself or by the RTP application using RFC 2190 [21][2]. The encoded bit stream is sent with an RTP tool (proprietary development of acticom)

[1] At the moment the encoding process is done off–line, but in the future it will be real–time encoding if efficient encoder software is available.

[2] No RFC is available for H2̇6L video transmission over RTP yet, so we used the RFC for H2̇63.

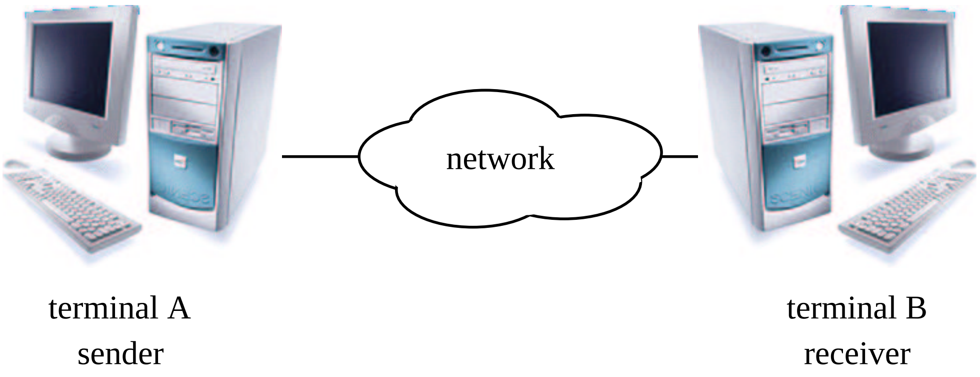


Figure 10: Test–bed for ROHC performance evaluation with one sender receiver pair.

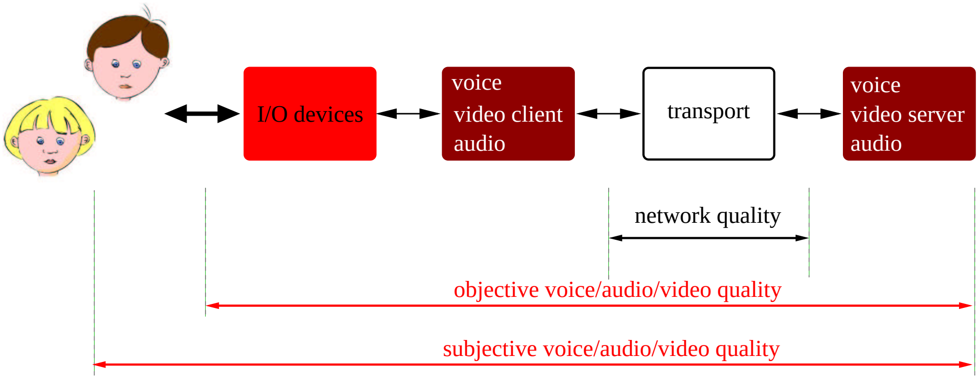


Figure 11: Different quality levels for ROHC performance evaluation.

according to RFC 1890 [17] over the ROHC test–bed to the receiver. At the receiver side the payload is extracted, the payload decoded, and re–composed to one *received* video sequence.

For comparison we encode and decode the video sequence without the transmission over the ROHC test–bed. The received video sequence is called the *coded* video sequence (see Figure 13).

In order to estimate the video quality we need a metric which e.g. compares the reconstructed frame at the receiver side to the original frame. We use the peak signal to noise ratio (PSNR). The PSNR represents the objective video quality each video frame by a single number. To achieve subjective video quality measurements the only solution is to test the video sequences by a large group of test persons. Therefore, we renounce the subjective quality for video services and focus on the objective quality level only.

A video frame is composed by $N \cdot M$ pixels (where M is the length and N the height of the frame). Each pixel is presented by one luminance value and a set of pixels by two chrominance values. Because the human eye is more sensible to the change in luminance we focus only on this parameter. The mean squared error (MSE) and the PSNR in decibels are computed by the following two equations:

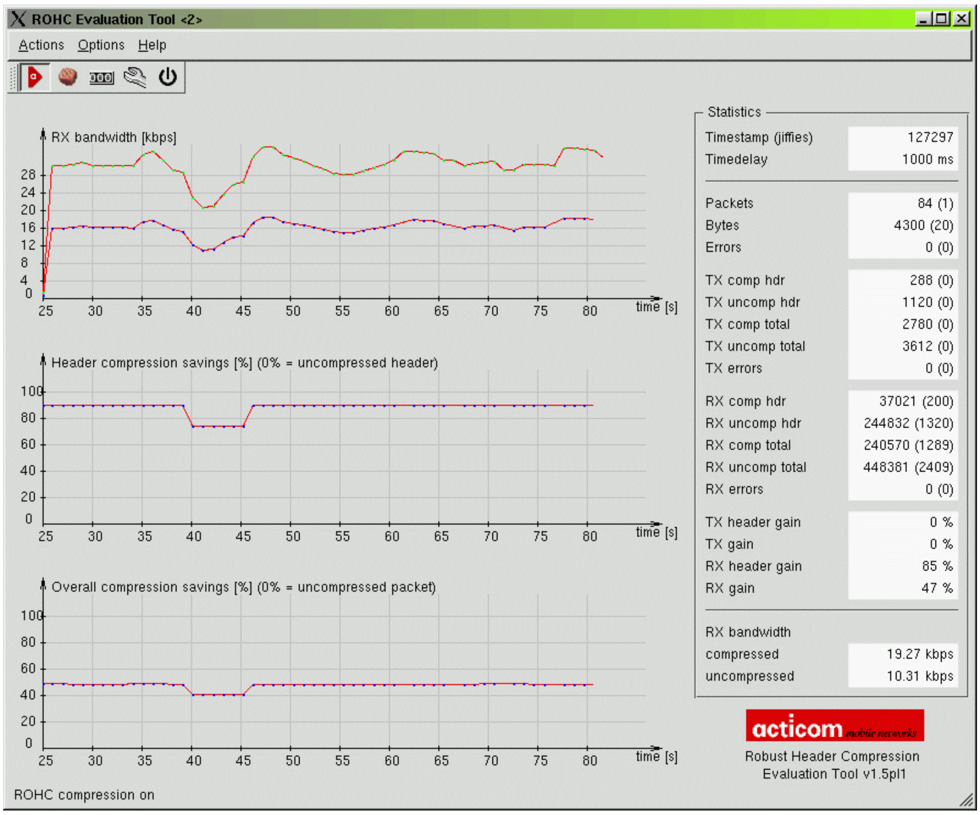
$$
\begin{aligned}
MSE &= \frac{\sum\limits_{\forall i,j} [f(i,j) - F(i,j)]^2}{N \cdot M} && (5)\\
PSNR &= 20 \cdot log_{10} \left(\frac{255}{\sqrt{MSE}} \right), && (6)
\end{aligned}
$$

where $f(i,j)$ represents the original source frame and $F(i,j)$ the reconstructed error–prone frame containing N by M pixels. As mentioned before, wireless links have smaller bandwidth than wired links, therefore we concentrate on the transmission of videos in the QCIF format (M=176,N=144).

To estimate the quality through the PSNR value, the error–prone stream is compared with the original stream. In the case of missing frames we have to insert dummy frames which are a copies of the last successful frame to ensure that the YUV streams will not run out of synchronism. In case of frame losses, we freeze the last successful frame to assure synchronism (see Figure 13).

As given in Figure 13 we define three PSNR values:

- $PSNR_1$ for the picture signal–to–reconstructed frame ratio between the *original* (YUV_{orig})



Statistics

Timestamp (jiffies)	127297
Timedelay	1000 ms
Packets	84 (1)
Bytes	4300 (20)
Errors	0 (0)
TX comp hdr	288 (0)
TX uncomp hdr	1120 (0)
TX comp total	2780 (0)
TX uncomp total	3612 (0)
TX errors	0 (0)
RX comp hdr	37021 (200)
RX uncomp hdr	244832 (1320)
RX comp total	240570 (1289)
RX uncomp total	448381 (2409)
RX errors	0 (0)
TX header gain	0 %
TX gain	0 %
RX header gain	85 %
RX gain	47 %
RX bandwidth	
compressed	19.27 kbps
uncompressed	10.31 kbps

ROHC compression on

Figure 12: NetMeter tool displaying the performance features of the network quality.

and the encoded/decoded (*coded*) YUV (YUV_{coded}) without any network transmission path. The $PSNR_1$ value describes the degradation in quality due to the video encoding process.

- $PSNR_2$ is the picture signal–to–reconstructed frame ratio between the *coded* YUV_{orig} and the *received* and decoded YUV (YUV_{recv}) stream including the network transmission path. The $PSNR_2$ value describes the degradation in quality due to the ROHC process.

- $PSNR_3$ is the picture signal–to–reconstructed frame ratio between the *original* YUV_{orig} and the *received* and decoded YUV (YUV_{recv}) stream. The $PSNR_3$ value describes the degradation in quality due to the video encoding and the ROHC process.

Using the videometer tool (see Figure 14) the impact of the video coding and the ROHC coding can be seen. The videometer tools displays the *original*, the *coded*, and the *received* YUV stream. Furthermore the videometer–tool displays the differences between the YUV streams and plots the PSNR values over the time.

4 Results of Performance Evaluation

The results provided here are referring to the video streams as given in Table 1. Our test were done using the *optimistic* mode. For our measurements concerning the ROHC performance on error-prone data links, we utilized the independent errors characteristics. Even if wireless links

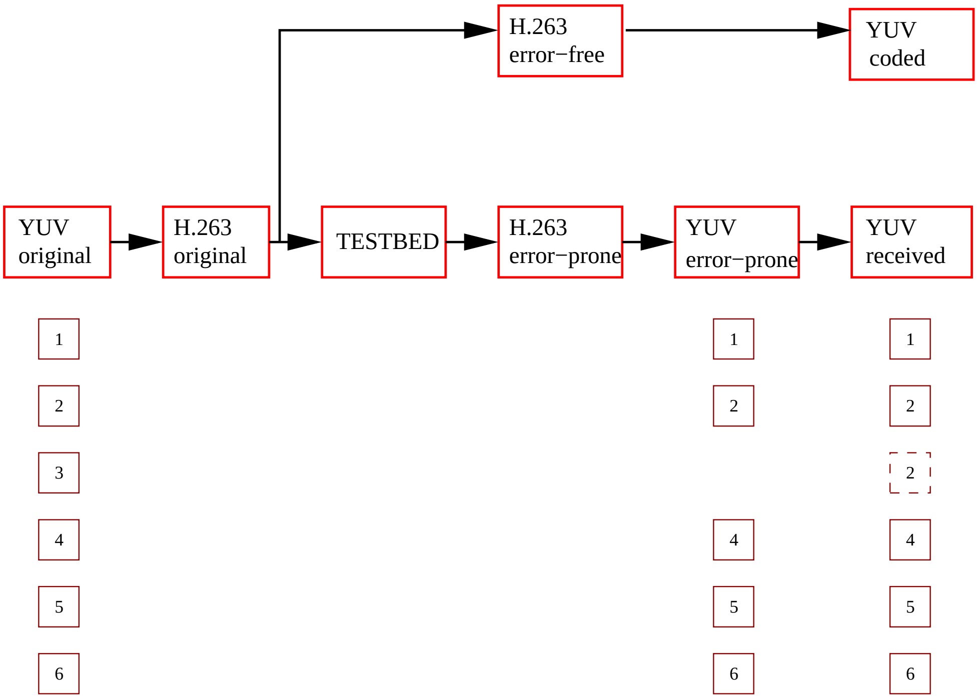


Figure 13: Methodology for PSNR calculation for video transmission over wireless links.

are characterized by correlated bit errors behavior, we have to take under consideration the data link characteristics. Researchers in [] reported that bit errors are uncorrelated for a specific data channel type. Only IPv4 for used for our measurements.

In Table 3 the network and the objective video quality are presented for the introduced set of video sequences. The compression of the header is for all sequences around 84%, which lead to a mean header size of 6 byte. Note, the uncompressed header size is 40 byte for IPv4. The header size could be reduced further by e.g. leaving out the UDP checksum. But this fine tuning is out of the scope of this paper. While the compression gain of the header is nearly stable, the compression of the whole video sequence depends on the video sequence. Comparing with the theoretical upper bound values the realistic compression gain is only slightly smaller. The lowest compression gain for the investigated video sequences of quality level 30 can be reported as 5.4% and the highest value equals 14.7%. In case of smaller video quality the compression gain can be 40% for the container sequence.

In the following we want to investigate if the compression over error-prone link leads to any degradation of the video quality. But as can be observed from the Figure 15, the utilization of ROHC compression does not result in quality losses. Please note that the high level of discretization derives from the decoder that is not always able to decode a sequence with errors totally. This effect gets stronger as the bit error probability (BEP) increases. As this is true for both, ROHC–based and uncompressed transmissions, the ROHC–compression in comparison is not to be seen as inflicting accountable objective quality losses.

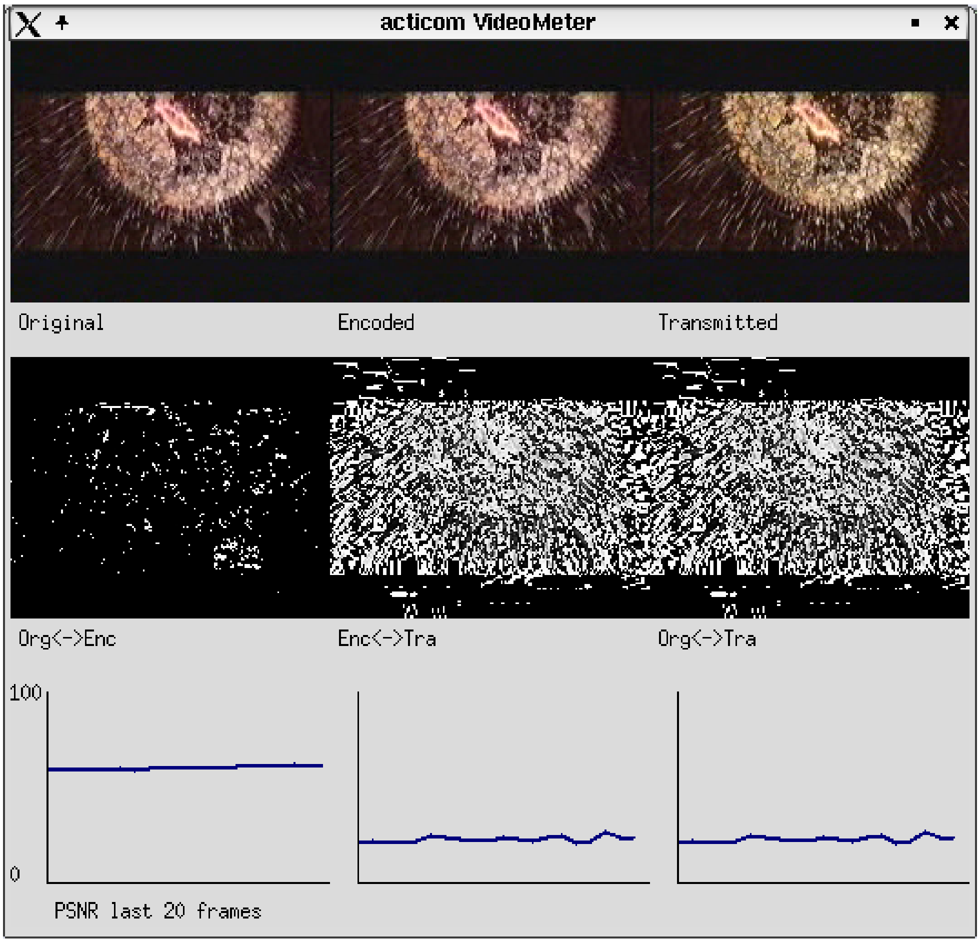


Figure 14: Screen–shot of the videometer tool with original, coded and transmitted video sequences.

5 Conclusions

The overall saving for a network provider depends on the traffic mixtures generated by the customers and is therefore hard to predict. In this paper we presented some realistic results for bandwidth saving in dependency of video services based on the H.26L codec and the link conditions. For the investigated video streams ROHC would lead to a bandwidth improvement of 9% using a really high video quality. By reducing the video quality we found possible savings around 40%. While we could report an improvement of the voice quality using ROHC in [14], the video quality does not change by applying ROHC.

Acknowledgments

The authors would like to thank the whole acticom team for their support regarding the ROHC implementation and the testbed. Furthermore we thank Andrea Carpene, Adam Wolisz and Holger Karl.

Table 3: Network and objective video quality (QCIF, ROHC in optimistic mode on error-prone link).

video		Network			Objective
	quality	header	total	$\overline{Header}$	$\overline{PSNR}$
	level	[%]	[%]	[byte]	
container	10	84.2490	0.9317	6.37	29.76
container	20	83.7182	3.5756	6.52	37.81
container	30	84.2695	10.5022	6.29	34.57
container	40	84.2695	23.0230	6.27	29.15
container	51	83.6596	40.0264	6.51	22.43
bridge close	30	84.5228	8.5537	6.19	18.25
carphone	30	83.8695	5.4516	6.44	33.08
claire	30	84.5560	14.7071	6.19	37.89
foreman	30	83.8829	6.0484	6.44	27.57
grandma	30	84.5160	12.0303	6.19	34.95
highway	30	84.2655	10.1713	6.29	34.09
mother	30	84.5827	10.5016	6.16	38.83
news	30	83.6094	7.1458	6.55	35.15
salesman	30	84.5092	8.7668	6.21	33.74
silent	30	84.2695	7.1592	6.28	32.39

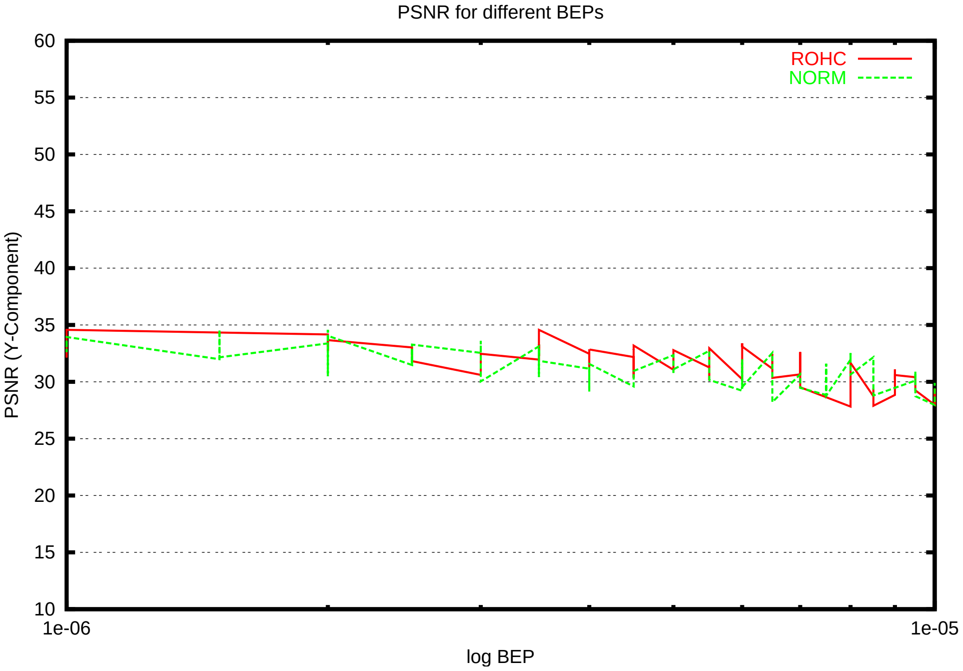


Figure 15: PSNR for different bit error probabilities (BEP).

References

[1] acticom GmbH. Robust header compression module. http://www.acticom.de/products/rohc/. 13

[2] C. Bormann. Robust Header Compression (ROHC) over PPP. Technical report, Request for Comments3241, April 2002. 13

[3] C. Bormann, C. Burmeister, M. Degermark, H. Fukushima, H. Hannu, L-E. Jonsson, R. Hakenberg, T. Koren, K. Le, Z. Liu, A. Martensson, A. Miyazaki, K. Svanbro, T. Wiebke, T. Yoshimura, and H. Zheng. RObust Header Compression (ROHC): Framework and four profiles: RTP, UDP, ESP, and uncompressed. Technical report, Request for Comments 3095, July 2001. 4, 8

[4] A. Calveras, M. Arnau, and J. Paradells. A controlled Overhead for TCP/IP Header Compression Algorithm over Wireless Links. In *The 11th International Conference on Wireless Communications (Wireless'99)*, 1999. Calgary, Alberta, Canada. 8

[5] A. Calveras, M. Arnau, and J. Paradells. An Improvement of TCP/IP Header Compression Algorithm for Wireless Links. In *Third World Multiconference on Systemics, Cybernetics and Informatics (SCI'99) and the Fifth International Conference on Information Systems Analysis and Synthesis (ISAS'99)*, volume 4, pages 39–46. IEEE, July/August 1999. Orlando, USA. 8

[6] A. Calveras and J. Paradells. TCP/IP Over Wireless Links: Performance Evaluation. In *48th Annual Vehicular Technology Conference VTC '98*, volume 3, pages 1755–1759. IEEE, May 1998. Ottawa (Ontario), Canada. 8

[7] Nordgren B. Degermark, M. and S. Pink. IP Header Compression. Technical report, Request for Comments 2507, February 1999. 8

[8] Frank H.P. Fitzek, A. Köpsel, Adam Wolisz, M. Reisslein, and M. A. Krishnam. Providing Application–Level QoS in 3G/4G Wireless Systems: A Comprehensive Framework Based on Multi–Rate CDMA. In *IEEE International Conference on Third Generation Wireless Communications*, pages 344–349, June 2001. 4

[9] Internet Engineering Task Force. RTP: A Transport Protocol for Real-Time Applications. RFC 1889, January 1996. 4

[10] V. Jacobson. Compressing TCP/IP Headers for Low-Speed Serial Links. Technical report, Request for Comments 1144, February 1990. 8

[11] JVT. JM / TML Software Encoder/decoder. http://bs.hhi.de/ suehring/tml/, 2002. H.26L Software Coordination by Carsten Suehring. 6

[12] Information Sciences Institute University of Southern California. User Datagram Protocol. RFC 768, August 1980. 4

[13] S. J. Perkins and M. W. Mutka. Dependency Removal for Transport Protocol Header Compression over Noisy Channels. In *International Conference on Communications (ICC)*, volume 2, pages 1025–1029, Montreal, Canada, June 1997. 8

[14] S. Rein, F. Fitzek, and M. Reisslein. Voice Quality Evaluation for Wireless Transmission with ROHC. In *International Conference on Internet and Multimedia Systems and Applications*, page t.b.d., Honolulu, Hawaii, USA, August 2003. 18

[15] M. Rossi, A. Giovanardi, M. Zorzi, and G. Mazzini. Improved header compression for TCP/IP over wireless links. *Electronics Letters*, 36(23):1958 – 1960, November 2000. 8

[16] M. Rossi, A. Giovanardi, M. Zorzi, and G. Mazzini. TCP/IP Header Compression: Proposal and Performance Investigation on a WCDMA Air Interface. In *12th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, volume 1, pages A–78 – A–82. IEEE, September 2001. 8

[17] H. Schulzrinne. RTP Profile for Audio and Video Conferences with Minimal Control. Technical report, Request for Comments 1890, January 1996. 15

[18] P. Seeling, F. Fitzek, M. Reisslein, and A. Wolisz. H26L Video Traffic Measurements. Technical Report TKN-02-000, TKN, August 2002. www.acticom.info. 6

[19] Th. Wiegand. H.26L Test Model Long–Term Number 9 (TML-9) draft0. ITU-T Study Group 16, Dec. 2001. 6

[20] Thomas Wiegand. H26lstandard. Technical report, JVT, 2002. 6

[21] C. Zhu. RTP Payload Format for H.263 Video Streams. Technical report, Request for Comments 2190, September 1997. 14