Header Compression Schemes for Wireless Internet Access

Frank Fitzek, Stefan Hendrata, Patrick Seeling, and Martin Reisslein

July 30, 2003

Contents

1 Introduction **6**
1.1 Motivation for Header Compression 7
1.2 Potential Savings of Header Compression 8

2 Header Compression Schemes **11**
2.1 Compressed Transport Control Protocol (CTCP, VJHC) 11
2.2 Refinements of CTCP 12
2.3 IP Header Compression (IPHC) 13
2.4 Compressed Real Time Protocol (CRTP) 14

3 Header Compression Schemes for Wireless Channels **16**
3.1 Drawback of Existing Schemes on Wireless Links 16
3.2 Robust Header Compression (ROHC) 16
3.3 Context and States 18
3.4 Compression of Header Fields 19
3.4.1 Compressor States 20
3.4.2 Decompressor States 20
3.5 Modes and State Transitions 21
3.5.1 Unidirectional Mode (U–Mode) 21
3.5.2 Bidirectional Optimistic Mode (O–mode) 22
3.5.3 Bidirectional Reliable Mode (R–mode) 23
3.6 Timer based compression of RTP timestamps 23
3.7 List Compression 23
3.8 More Profiles: TCP, IP–only, UDP–lite 24
3.9 ROHC over Wireless Ethernet Media 25
3.10 Signaling Compression 25
3.11 ROHC Summary 25
3.12 Evaluation of ROHC Performance for Multimedia Services 26
3.13 Deployment of Header Compression in Cellular Networks 28

4 Conclusion **30**

5 Acknowledgements **31**

List of Figures

1 Header structure and protocol stack with relevant layers (ROHC, IP, UDP, and RTP). 38
2 Header fields for RTP/UDP/IP packets (Version 4) and their dynamics. . 39
3 Header fields for RTP/UDP/IP packets (Version 6) and their dynamics. . 40
4 General concept of header compression: A current base header is maintained at compressor and decompressor. Header redundancy with respect to the base header is removed to obtain the compressed header (CH). . . 41
5 Van Jacobson [1]: Delta coding with respect to immediately preceeding header. 42
6 Perkins and Mutka [2]: Delta coding with respect to first (uncompressed) header in frame. 43
7 Transmission of compressed packets between ROHC compressor and decompressor. 44
8 Compressor States. 45
9 Decompressor States for all modes. 46
10 Mode transitions. 47
11 Compressor State transitions for Unidirectional mode: After repeatedly sending a packet with lower (IR or FO state) compression the compressor optimistically transitions to the FO or SO state. Updates in the header field pattern and a periodic timeout return the compressor to the FO state. A longer period timeout takes the compressor to the IR state. . . . 48
12 Compressor State transitions for Bidirectional Optimistic mode: After receiving an (optional) ACK from the decompressor or repeated transmissions (and optimistic assumption of established context) the compressor transits upward. The compressor returns to the FO state to update the header field development pattern or upon request (NACK) from the decompressor. With a Static–NACK the decompressor requests a static context update. 49
13 Compressor State transitions for Bidirectional Reliable mode: The compressor transitions upward only after receiving ACKs from the decompressor. Updates of the context or decompressor requests (NACK, Static–NACK) cause downward transitions. 50
14 NetMeter tool for the evaluation of ROHC compression [3]. 51
15 Objective voice quality (segmental SNR) as a function of bit error probability on a wireless link for transmission with ROHC and without header compression: ROHC improves the voice quality for moderate to large bit error probabilities, while using roughly 46% less bandwidth than transmission without header compression. 52
16 VideoMeter [4] for video quality evaluation. 53
17 Location of ROHC in UMTS transmission chain. ROHC is provided by the packet data convergence protocol (PDCP) between the mobile end system and RNC. 54

List of Tables

1 Theoretical upper bound savings (in terms of bandwidth) for voice traffic. 55

2 Transmitted video streams. 56

3 Upper bound on total bandwidth savings $\overline{S}$ for H.26L encoded video (QCIF) for different video encoding quantization scales (smaller quantization scales give higher encoded video quality) and different video sequences. 57

4 Average header size, header compression (relative to 40 byte IPv4 header), total bandwidth savings, and average video frame PSNR quality for H.26L video over error–prone link (BEP=10^{-5}, QCIF, ROHC in optimistic mode). 58

About the Authors

Dr. Frank Fitzek

Frank H. P. Fitzek received his diploma (Dipl.-Ing.) degree in electrical engineering from the University of Technology - Rheinisch-Westflische Technische Hochschule (RWTH) - Aachen, Germany, in 1997 and his Ph.D. (Dr.-Ing.) in Electrical Engineering from the Technical University Berlin, Germany in 2002. He co-founded the start-up company acticom GmbH in Berlin in 1999. At acticom GmbH he is currently involved in the development of advanced wireless IP services for WLAN and 3G. His current research interests are in the areas of QoS support for voice and video over wireless networks, robust header compression, security in self organizing networks, and the integration of mobile ad-hoc networks in cellular systems. Simultaneously, he is teaching at the University of Ferrara. Dr. Fitzek is member of the IEEE and serves on the Editorial Board of the IEEE Communications Surveys and Tutorials.

Dipl.-Ing. Patrick Seeling

Patrick Seeling is a graduate student in the Department of Electrical Engineering at Arizona State University. He received the Dipl.-Ing. degree in Industrial Engineering and Management from the Technical University of Berlin (TUB), Germany, in 2002. His research interests are in the area of video communications in wired and wireless networks. He is a student member of IEEE.

Dr. Martin Reisslein

Martin Reisslein is an Assistant Professor in the Department of Electrical Engineering at Arizona State University, Tempe. He received the Dipl.-Ing. (FH) degree from the Fachhochschule Dieburg, Germany, in 1994, and the M.S.E. degree from the University of Pennsylvania, Philadelphia, in 1996. Both in electrical engineering. He received his Ph.D. in systems engineering from the University of Pennsylvania in 1998. During the academic year 1994-1995 he visited the University of Pennsylvania as a Fulbright scholar. From July 1998 through October 2000 he was a scientist with the German National Research Center for Information Technology (GMD FOKUS), Berlin. While in Berlin he was teaching courses on performance evaluation and computer networking at the Technical University Berlin. He has served on the Technical Program Committees of IEEE Infocom and IEEE Globecom. He is Editor-in-Chief of the IEEE Communications Surveys and Tutorials. He maintains an extensive library of video traces for network performance evaluation, including frame size traces of MPEG–4 and H.263 encoded video, at http://www.eas.asu.edu/trace. He is co-recipient of the Best Paper Award of the SPIE Photonics East 2000 — Terabit Optical Networking conference. His research interests are in the areas of Internet Quality of Service, video traffic characterization, wireless networking, and optical networking.

1 Introduction

The wireless market is widely believed to be the most important future market for data services. Despite the efforts that were made to increase usability and abilities of all different wireless devices, the wireless protocol domain is still challenging. As new services and protocols emerge for wired networks, the need to incorporate those services and protocols in wireless communication systems arises. Existing wireless networks of the second generation (also known as 2G) are mostly circuit–switched and have been developed and optimized for voice transmission. Wireless networks of the 3rd generation (3G) have to support a broad range of application scenarios. 3G networks and terminals such as smart–phones, PDAs, and Laptops are marketed with new services, most importantly multimedia. However, IP based multimedia applications require more bandwidth than traditional voice services. Multimedia applications comprise several different application scenarios, including audio, video, and gaming [5]. The bandwidth needs of these applications are higher than offered in 2G systems. In addition, multimedia applications require more stringent network quality of service (QoS). One of the key issues for the QoS is the suitability for real–time applications. Therefore, delay and jitter are key considerations within the QoS domain. Multimedia applications often use RTP, UDP, and IP as protocols. Each of the protocol layers adds header overhead. Thus, the bandwidth requirements derive from the application (i.e., the payload) and the protocol overhead. Compression of the multimedia payload is mostly excellently achieved on the application level (e.g., with current voice and video compression schemes). Attempting additional compression on the payload thus yields no benefits. Significant compression of the packet traffic can be achieved by reducing the amount of overhead information. For LPC coded voice, for instance, the IP overhead is 81%, as detailed shortly in Table 1. In general, for multimedia services, header compression achieves a dramatic saving in bandwidth. Given the high license fees of 3G bands and the migration of IP based services into the wireless format, it is necessary to reduce the header overhead of IP based traffic. IP is the underlying network layer protocol used for most multimedia application scenarios. Focusing on this protocol domain thus promises the highest gains. IP header compression mechanisms have always been an important part of saving bandwidth over bandwidth limited links. Many header compression schemes exist already, but most of them are not suitable for the wireless environment. Wireless links have typically a very high and variable bit error probability (BEP) due to shadow– and multi–path fading and mobility. With a reduction of the required bandwidth, the latency and Packet Error Probability (PEP) can be improved. This is because the probalility that a given packet is affected by link errors is reduced for smaller packets. For multimedia services in wireless environments ROHC was introduced. ROHC was standardized by the Internet Engineering Task Force in RFC 3095 [6] and will be an integral part of the 3GPP–UMTS specification [7]. 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 significant reduction of the IP header and other headers.

In this chapter, we present header compression schemes for the wireless domain. We start by outlining the motivations for and theoretical upper limits of header compression in general. Next, we describe different header compression schemes for the wired Internet. We examine the drawbacks and shortcomings of these compression schemes when applied in the wireless domain. Next, we introduce the Robust Header Compression (ROHC) scheme that has been developed for wireless channels. We study the performance of ROHC for audio and video traffic and discuss the ROHC deployment in 3G wireless systems.

1.1 Motivation for Header Compression

The motivation for IP header compression is based on the facts that (*i*) the multimedia payload is typically compressed at the application layer, (*ii*) the headers occupy a large portion of the packet for some services, and (*iii*) the headers have significant redundancy. In Figure 1 the combined header for a real–time multimedia stream with IPv4, resulting in 40 bytes protocol overhead is illustrated. The protocol headers include the 20 byte IPv4 header, the 8 byte UDP [8] header and the 12 byte RTP [9] header. With IPv6, there is a total of 60 bytes of overhead. In case of GSM coded audio transmission the payload is only 33 bytes (13.2 kbps $\times$ 20 ms) long — the header of IPv4 accounts for 55% of the packet. For IPv6 this ratio is even larger. There are some redundancies among the different headers (IP, UDP, and RTP) of a given packet, but typically there are even larger redundancies between contiguous packets of a given IP flow. Thus, there are two types of header redundancies:

Intra–packet The headers for the different protocols within a single packet carry identical or deducible information.

Inter–packet The headers between consecutive packets have only marginal (incremental) differences.

The structure of typical multimedia packets is illustrated in Figures 2 for IPv4 and 3 for IPv6. The specific header fields for the RTP/UDP/IP headers are marked with respect to their dynamics. A classification of the header fields into groups results in *non changing* and *changing*. The *non changing* group consists of static, static-known, and inferred header fields. Figures 2 and 3 illustrate that a large portion of the header fields are static or static–known (20.5 bytes in IPv4 and 44.5 bytes in IPv6). The compression of these fields can be achieved with low to moderate complexity. In fact, these fields could be completely omitted after the first successful transmission. The Segment Length (2 bytes) and Packet Length (2 bytes) fields (as well as the Header Checksum in IPv4 with 2 bytes) are referred to as inferred. These entries can be inferred from other header fields and are also relatively easily compressed. The *changing* group consists of not–classified–a–priori, rarely–changing, static or semi–static changing, and alternating changing header fields. These header fields are more difficult to compress and it depends on the applied header compression scheme how the compression is achieved.

1.2 Potential Savings of Header Compression

To get a basic idea of the possible savings of header compression, we compute an upper bound for voice communication and for audio streaming. For this upper bound calculation, we assume that with header compression the overhead due to IP, UDP, and RTP is zero. The upper bound on the savings, denoted by S_i, for packet i is then

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

where $Packet(i)$ denotes the size of the payload data and $Header$ denotes the size of the (uncompressed) IP, UDP, and RTP headers. Clearly, the potential savings depend only on the mean packet length. The packet length depends on the service type used. The mean saving

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

gives the portion of bandwidth that a wireless network provider can potentially save for a session with N packets. The potential savings for voice service are given in Table 1. We consider the Linear Predictive Coding (LPC) with 5.6 kbps, the GSM codec with 13.2 kbps, and a codec following the ITU-T standard G.711 with 60.0 kbps. Note that LPC gives acceptable quality for voice communications, the GSM and ITU-T G.711 codecs provide higher quality suitable for audio streaming. For the calculation it is assumed that a packet is generated every 20 ms. Therefore, S_i is the same for all IP packets (and thus equal to $\overline{S}$).

The potential savings for IPv6 are larger than for IPv4 because of the IP header length (IPv4 with 40 byte and IPv6 with 60 byte). The second indication from Table 1 is that smaller IP packets correspond with larger savings $\overline{S}$. This is due to the larger ratio of the header compared to the smaller packets. Overall, we observe there is a large potential for bandwidth savings. The savings are most significant for low–bitrate streams, which tend to be common in wireless networks.

In contrast to voice streams with fixed frame and packet sizes, the frame sizes of a video stream vary over time [10, 11]. The size of the video frames depends on the content of the video sequence and the applied encoding scheme and settings. We encoded the generally accepted video reference streams *container*, *bridge*, *carphone*, *claire*, *foreman*, *grandma*, *highway*, *mother*, *news*, *salesman*, and *silent* (as given in Table 2) in the QCIF format (176x144 pixel) with the H.26L encoder [12, 13] using the IBBPBBPBBPBB group of picture (GoP) structure. To evaluate the upper bound on the mean savings $\overline{S}$ we assume that each video frame is carried in one packet. The measured frame sizes give the upper bounds for the mean potential savings $\overline{S}$ for H.26L encoded video streams given in Table 3. Here we assumed that no layered coding is used, which would further increase the header compression gain. This is because with layered coding, each video layer is transmitted in its own RTP session and has its own headers.

2 Header Compression Schemes

Several different schemes for compressing the network and transport layer protocol headers have emerged in the 90's. These schemes all have in common that their respective focus is only on a certain combination of protocols. All are mainly based on the same compressor/decompressor concept. In Figure 4 the general concept of header compression is illustrated. On the sender side the compressor removes redundancy from the incoming packet with respect to a reference (base) header. This reference is also known (and maintained) at the receiver and allows the receiver to decompress the incoming compressed packet headers. In the following, we present the most common of these header compression schemes. We refer the reader to the referenced literature for more details on these schemes.

2.1 Compressed Transport Control Protocol (CTCP, VJHC)

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 [1] as RFC1144 and focuses on the TCP protocol. VJHC processes TCP and IP headers together and compresses the 40 byte TCP+IP header to a 4 byte compressed header. A second benefit from the combined processing is the reduced complexity of the employed algorithms. VJHC is based on delta coding as illustrated in Figure 5. The differences between two packet headers are referred to as the "delta". Instead of transmitting the entire header, VJHC transmits only the delta. This approach achieves high compression. On the downside, it introduces vulnerability. If only one delta coded header is corrupted, all the following packets are erroneous. To recover from these errors and re–establish the current base header, VJHC sends all TCP re–transmissions uncompressed. Thus, VJHC does not require any signaling between compressor and decompressor. The disadvantage is the sensitivity to error–prone links as investigated in [14, 15, 16, 17, 2].

2.2 Refinements of CTCP

In [2], robustness at the cost of less compression was introduced by Perkins and Mutka. As illustrated in Figure 6, the delta–coding for the adjacent 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 delta coding with respect to the first (uncompressed) packet in the frame. Clearly, the differences to packets at the end of a frame are larger than for those at the beginning. The compression gain is thus limited (and lower than for VJHC). The advantage of this approach is the usage of shorter delta coding ranges. Corrupted packets do not necessarily lead to the loss of synchronization. This is a clear advantage over VJHC. An optimization for the header compression of Perkins was introduced by Calveras *et al.* in [16, 15]. This optimization minimizes the overhead by adapting the frame length as a function of the channel state and updating the base with compressed headers. The header compression schemes introduced by Perkins and Mutka and Calveras *et al.* obtain base updates by sending both compressed and uncompressed headers. Whenever one of these base updates is lost due to transmission errors, the synchronization between compressor and decompressor is lost and performance degrades.

Rossi *et al.* [18, 19] proposed the following modification to the base update procedure. Whenever the compressor decides to update the current base, it sends the new base to the decompressor following the standard delta coding mechanism, i.e., the compressor sends the new base in a compressed header taking the current base as the reference base. In addition, a *request flag* is set in the compressed base. The decompressor receives the new base, decompresses it using the current base and temporarily stores it as the new proposed base. The new base is at this time only a proposed base, the decompressor will start to use it only upon specific indication by the compressor. When the decompressor receives a new base, a TCP ACK including the sequence number of the packet carrying the new base is transmitted to the compressor. When the compressor receives the ACK with the sequence number of the packet carrying the new base, it knows that the proposed base has been correctly received by the decompressor. Therefore, it can safely start to use the new base for compression. This change of the used base is communicated to the decompressor by a *bistable flag*. The compressor learns from the TCP ACKs that the decompressor has started to use the new base. So, a minimum of two consecutive TCP ACKs is needed to change the base used for compression. The header base update is performed as frequently as possible to minimize the delta field's resource occupancy. Since the header base update involves two adjacent ACKs, this is used as the rate of the header base change. Further details and comparisons with other proposed algorithms can be found in [18, 19]

2.3 IP Header Compression (IPHC)

IP Header **C**ompression (*IPHC*) [17] provides a number of extensions to VJHC. The most important extensions are support for UDP, IPv6, and additional TCP features. With the explicit support of UDP come additional features, such as multicast. Nevertheless, support for RTP is still not given which makes the scheme unsuitable for many multimedia applications. Similar to VJHC, IPHC relies on the change of header fields as well as on the derivation of header field contents. The encoding also employs the delta–scheme, transmitting only the changes in the header fields. The error correction schemes of VJHC are used for TCP packets. For non–TCP packets, no differential encoding between sender and receiver is used. Thus, the compression for UDP–based streams is worse than for TCP–based streams, but the context is not affected by packet losses. Generally, context update packets are sent periodically to maintain the state at both ends. For the application of differential coding over lossy links, two new algorithms are introduced:

TWICE [20] The decompressor computes a checksum to determine if its context information has been updated properly. If this fails, a lost header-compressed packet is assumed. The decompressor recomputes the checksum by skipping the packet and assuming that the lost and current packet had the same delta values. If the second checksum computation is successful, the decompressor adjusts the base header (context) by two deltas.

Header Request When the decompressor is unable to repair the context after a loss, it requests a complete header from the compressor. This is only possible on bi-directional links, since the decompressor must communicate with its compressor. The decompressor sends a context state message that includes all compressed packet streams in need of a context update.

The performance of IPHC for packet voice over the wired internet and the interactions between IPHC and the interleaving of source coder symbols are studied in [21]. We also note that a header compression scheme specifically for IPv6 based communication with mobile wireless clients has been developed [22]. This IPv6 header compression scheme employs (*i*) address reassignment (translation), which maps the 16 byte IPv6 address to a shorter address, (*ii*) LZW coding [23, 24] for the initial header, and (*iii*) incremental comparator encoding for subsequent headers.

2.4 Compressed Real Time Protocol (CRTP)

The **C**ompressed **R**eal **T**ime **P**rotocol (*CRTP*) scheme presented in RFC2508 [25] compresses the 40 bytes header of IP to 4 bytes if the UDP checksum is enabled, or to 2 bytes if it is not. This is possible by compressing the RTP/UDP/IP headers together, similar to the VJHC approach. With the characteristics of the RTP protocol, the changes for the RTP header fields become partially predictable. In addition, changes in some fields are constant over long periods of time. Thus, the expected change in these fields can be implied without even transmitting the differences. These implied fields are also referred to as *first order* changes. They are stored with the general context for each specific connection. The differences within fields that have to be compressed are referred to as *second order* differences. An example for these are video frame skips. Video frames are generally transmitted every 40 ms. In case a frame cannot be encoded (e.g., due to lack of processing power or because of a slower play-out ratio), the implied time no longer is accurate. Therefore, the new *first order* is set to the *second order* and the connection context is updated. CRTP cannot use a repair mechanism as VJHC does because UDP/RTP are unidirectional protocols without retransmissions. CRTP uses a signaling message from decompressor to compressor to impart that the context is out of synchronization. For lossy links and long round trip delays, CRTP does not perform well. After a single lost packet several sequential packets are lost within the round trip time. Thus, CRTP is not suitable for cellular links (where the header compression currently is envisioned to be implemented in the wireless terminal and the radio network controller, resulting in significant round trip times, see Sec. 3.13). A performance evaluation for CRTP is given in [26].

RObust Checksum–based COmpression (ROCCO) [27] is a refinement of CRTP. ROCCO includes a checksum over the original (uncompressed) header in the compressed header. The checksum facilitates local recovery of the synchronization. In addition, ROCCO incorporates compression profiles (tailored for specific applications, e.g., audio or video streaming) and has a code with hints on the change of header fields in the compressed header. These mechanisms improve the header compression performance, especially for highly error–prone links and long round trip times [28, 27]. Similarly, Enhanced Compressed RTP (ECRTP) [29] is a refinement of CRTP. ECRTP uses local retransmissions to more efficiently recover from wireless link errors.

3 Header Compression Schemes for Wireless Channels

3.1 Drawback of Existing Schemes on Wireless Links

The majority of the header compression schemes were designed for wired links. Wired links are characterized by low error rates. Wireless links, on the other hand, are characterized by higher and bursty transmission error rates. In addition, wireless services — such as those for 3G — often require realtime protocol support. Thus, schemes designed with different goals have several shortcomings outlined in the following:

VJHC and its refinements [1, 2, 18, 19] These schemes were designed for usage with TCP. They are therefore unsuitable for multimedia applications running on UDP. The used delta coding scheme makes these protocols vulnerable to link–errors. As noted above, the packets from the instant an error occurs to the end of the TCP timeout window are retransmitted uncompressed. This considerably reduces the achieved performance in wireless environments.

IPHC, CRTP [25, 17] Both, IPHC and CRTP, do not offer the efficiency and robustness needed for wireless links [30]. The local recovery mechanisms of ROCCO are very helpful in ensuring efficiency and robustness; these basic ideas are incorporated in the design of ROHC.

3.2 Robust Header Compression (ROHC)

ROHC is envisioned as an extensible framework for robust and efficient header compression over highly error–prone links with long round–trip times. This design is motivated by the large bit error rates (typically on the order of 10^{-4} – 10^{-2}) and long round trip times (typically 100–200 msec) of cellular networks. The design of ROHC is based on the experiences from the header compression schemes reviewed above. In particular, ROHC incorporates elements from ROCCO and Adaptive Header Compression (ACE) [31] which may be viewed as a preliminary form of ROHC.

ROHC 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. The IP protocol header can be version 4 or version 6. In case any other protocol suite is used, ROHC does not perform compression by using the uncompressed profile. We note that there are other profiles in development to support more protocol suites, including IP only, and TCP/IP, see for instance [32, 33, 34, 35, 36, 37]. (We also note that there have been research efforts towards a general, protocol-independent header compression framework [38]. This framework combines (*i*) a high level header description language for flexible specification of the header properties of a specific protocol, and (*ii*) a code generator tool for automatic generation of the corresponding header compression code.) As shown in Figure 7, 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 wireless link. So the compression should work only between two wireless nodes, whereas for the rest of the Internet this operation remains transparent. In the simplest configuration, the wireless sender contains the compressor and the decompressor is located in the wireless receiver (see Sec. 3.13 for the actual location of ROHC compressor and decompressor in 3G systems). ROHC controls the interaction between these two nodes in order to achieve two main goals:

1. The network providers desire a significant bandwidth saving obtainable by reducing the headers to a shorter ROHC header.

2. Despite the compression it is necessary to ensure a QoS acceptable for the customers of the network providers.

The compressor can sacrifice the bandwidth saving in order to keep the decompressor synchronized even if errors occur on the link. ROHC can thus sacrifice compression efficiency for error correction capability and does therefore not always work at the peak of its compression abilities. Different levels of compression, called states, are used within ROHC. This state–based approach is a new robust solution against the perils of the wireless link. In this section we give a basic overview of how ROHC achieves robust header compression. For details on the algorithms and schemes, please refer to the original specification in RFC 3095.

3.3 Context and States

In general, data packets that are transferred over a wireless link are not independent from each other but share certain common parameters such as equal source and destination addresses. Moreover, they usually can be grouped together logically, e.g., data packets that constitute an audio stream and data packets that make up the accompanying video stream. Thus, it makes sense to use a stream–oriented approach in ROHC to compress packet headers. Each stream or flow is identified by its parameters that are common to all packets belonging to this stream. The compressor and decompressor maintain a context for each stream which is identified by the same context identifier (CID) on both sides. A context — being a set of state data — contains, for example, the static and dynamic header fields that define a stream.

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'. In this state, compressor and decompressor have no agreement on compressing or decompressing a certain stream. Thus, the compressor needs to send a ROHC packet containing all the stream and packet information (static and dynamic) to establish the context. This packet is the largest ROHC header that the compressor sends. In the second state, the static part of the context is regarded as established between compressor and decompressor while the dynamic part is not. In this state, the compressor sends ROHC headers containing information on the dynamic part of a context. These headers are smaller than those sent in the first state but slightly larger than the headers used in the third state. In the third and final state, the static as well as the dynamic part of a context are established and the compressor needs to send only minimal information to advance the regular sequence of compressed header fields. Fall-backs to lower states occur when the compressor detects a change or irregularity in the static or dynamic part (i.e., pattern) of a stream, or when the decompressor detects an error in the static or dynamic part of a context.

The compressor strives to operate as long as possible in the third state under the constraint of being confident that the decompressor has enough and up-to-date information to decompress the headers correctly. Otherwise it must transit to a lower state to prevent context inconsistency and to avoid context error propagation.

3.4 Compression of Header Fields

The compression of the static part of headers for a stream is similar to the other header compression schemes. Header fields that do not change need only to be transmitted at context establishment and remain constant afterwards. More sophisticated algorithms are needed for compression of the dynamic part. With ROHC, the values of the dynamic header fields are derived as linear functions from the sequence number of each packet which in general increases by one for each new packet. However, these functions for the dynamic header fields are expected to change and the compressor must therefore be able to effectively communicate these changes to the decompressor.

For compressing and decompressing dynamic header fields — either directly or by the use of a function — ROHC employs two basic algorithms:

Self–describing variable length values This algorithm reduces the number of bits needed to transmit an absolute value. Small values are described with fewer bits than large values.

Windowed Least Significant Bits (W–LSB) encoding Dynamically changing values usually have their characteristic dynamic behavior, e.g., always incrementing by one for sequence numbers. Using this knowledge of the dynamic behavior, a window is constructed around a reference value which is a previous, correctly transmitted value. Depending on the distance of the new value from the reference value and the relative position of the window with regard to the reference value, the number of bits to transmit the compressed new value is determined. These bits thus describe the advancement of a value from a reference value and their number is small for header field values that do not randomly change but continously increase or decrease by usually small differences. The advantage of this algorithm is the consideration of the dynamic behavior of the value when defining the position of the window resulting in a minimization of the average number of bits needed to be transmitted in order to describe the sequence of values for a header field in a stream.

The W–LSB compression algorithm in combination with an elaborate protection scheme for sensible data in ROHC–compressed headers contribute to the robustness of ROHC.

3.4.1 Compressor States

The three compressor states illustrated in Figure 8 are the

- Initialization and Refresh state (IR),
- First Order state (FO), and
- Second Order state (SO).

In the IR state there is no context for compression available. Thus, the compressor and decompressor have to transit to a higher state as soon as possible for effective compression. 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 derived from it. These SO state ROHC headers are the smallest ones with in general one byte size. If an irregularity in a stream occurs, the compressor falls back 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 stream returns again to a regular behavior (pattern), the compressor transits up to the SO state.

3.4.2 Decompressor States

The three decompressor state names depicted in Figure 9 refer to the grade of context completeness. In the No Context (NC) state, the decompressor lacks the static and dynamic part of a context. Consequently, it can only decompress IR packets, i.e. packets sent with uncompressed header fields 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 correctly decompress an FO packet to recover to the FC state. Otherwise, receiving further errors leads to the transition to the NC state.

3.5 Modes and State Transitions

To offer the ability to run over different types of links, ROHC operates in one of three modes: Unidirectional, Bidirectional Optimistic, and Bidirectional Reliable mode. Similarly to the states, ROHC starts at the most basic mode (unidirectional) but can then transit to the other modes if the link is bidirectional. Contrary to the states, modes are not directly related to the level of compression. The modes differ from each other in the amount of coupling between compressor and decompressor by the use of feedback packets sent by the decompressor to the compressor. For example, the Unidirectional mode does not make use of feedback packets at all, while the Bidirectional Reliable mode tightly couples compressor and decompressor by requiring a feedback packet for each update of the context. Mode transitions can be initiated by the decompressor at any time for an established context. To do so, the decompressor inserts a mode transition request in a feedback packet, indicating the desired mode as Figure 10 illustrates.

3.5.1 Unidirectional Mode (U–Mode)

This mode is designed for links without a return channel. There is no way for the compressor to be certain whether the decompressor has received the correct context information and is thus decompressing correctly. It can only optimistically assume that the decompressor has received the context data correctly by repeatedly sending the same information. However, the decompressor must have a chance to update and correct its context in case of context errors. The compressor therefore periodically times out and falls back to the FO and IR states, as illustrated in Figure 11. Typically, the timeout period for fallbacks to the IR state is larger than the timeout period for fallbacks to the

FO state. The decompressor uses the periodically sent FO and IR packets to verify and possibly correct its context. The compressor also falls back to the FO state whenever the pattern of header field evolution changes. Whenever the compressor is in the IR or FO state it sends multiple packets with the same lower level of compression until it is confident that the decompressor has established the flow context. The compressor then optimistically transits upward to the higher compression FO or SO state. This adds robustness against single packet errors. The U–Mode is the least robust and least efficient mode among the three ROHC modes.

3.5.2 Bidirectional Optimistic Mode (O–mode)

As an extension to the Unidirectional Mode, the Bidirectional Optimistic mode uses feedback packets that are sent from the decompressor to the compressor in order to accelerate state transitions at the compressor and to avoid the periodic fallbacks to the FO and IR states. As shown in Figure 12 context update acknowledgements (ACKs) are used to notify the compressor that the decompressor has successfully received context information. (These ACKs from the decompressor are optional, thus the compressor may still need to use the optimistic upward transitions.) In case of a context error, a context recovery request (NACK) is sent to the compressor, causing a retransmission of context information to update and repair the context at the decompressor. With a Static–NACK the decompressor forces the compressor back to the IR state, thus re–establishing the static context. With these context updates on request, the compressor can achieve a higher compression efficiency compared to the unidirectional mode.

Due to the mostly weak protection (3–bit CRCs) of context updating data sent in the Bidirectional Optimistic mode, there is still a not to be neglected probability of context damage that can result in a sequence of incorrectly transmitted packets. For applications that prefer a more reliable transmission with a lower probability of incorrect packet transmission, the Bidirectional Reliable mode was conceived.

3.5.3 Bidirectional Reliable Mode (R–mode)

To achieve a lower probability of incorrect packets, a more powerful error correction (7-bit CRCs) is used for context updating information in the Bidirectional Reliable mode. In addition, the compressor transits to the FO or SO state only after receiving an ACK from the decompressor, as illustrated in Figure 13. The compressor transits downward to update the context or upon decompressor request (NACK, Static–NACK). With this mode, the behavior of the compressor and decompressor are thus even closer coupled than with the optimistic mode. The rare NACKs provide a quick context recovery in case of errors. Therefore, the compressor always knows in which state the decompressor is and when to make a state transition.

3.6 Timer based compression of RTP timestamps

A special, more efficient compression method for the RTP timestamp header field can be applied under certain conditions. If the application hands over RTP packets in real-time to the ROHC compressor, the time difference between the handover of two RTP packets is proportional to the difference of the RTP timestamp header field values in these two packets. Provided that the transmission channel has a low delay jitter, the decompressor can then use the time difference between the reception of two compressed packets to estimate the new RTP timestamp value based on the previous value. With timer based compression, the compressor needs to send even fewer bits for compression of the RTP timestamp than with the standard W–LSB encoding based method. These bits are used to refine the estimation of the decompressor. The number of bits needed for refinement depends on the amount of delay jitter on the transmission channel which has to be determined.

3.7 List Compression

Some data in headers of data packets that belong to a stream contain a variable number of items as for example the Contributing SouRCe. (CSRC) [9] list in the RTP header. They also rarely change within a stream. In order to compress these lists, an elaborate scheme was developed in ROHC called "list compression". With list compression, the compressor can refer to items of which it is confident that the decompressor has received them correctly in previous packets using a short item identifier. The list is maintained and updated between compressor and decompressor by list operations such as insertions, removals etc. that describe changes to the list.

In addition to CSRC lists, the list compression scheme is used to compress extension headers such as GRE headers [39, 40], authentication headers, minimal encapsulation headers, IPv6 extension headers etc.

3.8 More Profiles: TCP, IP–only, UDP–lite

As already mentioned, ROHC is defined in RFC 3095 as a framework and is thus extensible with new profiles. In addition to already specified profiles for compressing IP/UDP/RTP, IP/UDP and IP/ESP headers, profiles for compressing the IP/TCP headers, and IP header only are in development as of the time of writing. Contrary to previous TCP compresssion schemes, the ROHC TCP profile is capable of referring to already established TCP compression contexts when establishing a new context resulting in a reduced overhead for new compression contexts. This improves compression efficiency for sequences of so–called "short–lived TCP transfers", which often occur with HTTP transfers used by World Wide Web (WWW) browsers. The IP–only profile is intended for header combinations that contain an IP header, but are not covered by the ROHC profiles specified so far. The compression of the IP header can at least reduce the size of the headers to some degree. As a modification to the already existing profiles for compression of the IP/UDP and IP/UDP/RTP headers, two new profiles are proposed that specify compression for IP/UDP-lite and IP/UDP-lite/RTP headers. With the standard UDP protocol, the UDP payload is either entirely or not at all protected by a CRC checksum. UDP-lite [41] offers flexibility in the amount of payload data that is CRC protected. This is advantageous for several audio and video codecs that prefer the delivery of erroneous packets (and apply error concealment [42] at the decoder level). This approach is attractive for real-time streaming, as it allows the decoder to make the best out of the delivered data.

3.9 ROHC over Wireless Ethernet Media

The specification for ROHC does not describe the inter–operation with the underlying link–layer in detail. Only a few requirements that have to be met by the link–layer are mentioned (e.g., no packet reordering or duplication on the channel between compressor and decompressor). Additional drafts and RFCs specify how ROHC operates on top of certain link–layer protocols (e.g., PPP [43]). A large group of link–layer technologies are formed by the Ethernet–based network technologies. Among them, the wireless local area variants, such as IEEE 802.11 (Wireless LAN) or Bluetooth, exhibit similar bit error characteristics. Consequently, there are efforts under way to standardize the operation of ROHC on top of wireless Ethernet–like media. An exemplary application that can benefit from the employment of ROHC is Voice-over-IP telephony in a wireless LAN environment.

3.10 Signaling Compression

Another significant outcome of the ROHC working group (next to RFC 3095) was RFC 3320 [44] specifying Signaling Compression (SigComp). SigComp uses an entirely different approach to compression than ROHC. With SigComp, a Universal Decompressor Virtual Machine (UDVM) executes code that is sent by the compressor to decompress packets. This allows for greater flexibility in utilizing different compression strategies and algorithms. SigComp is envisioned for the compression of Session Initiation Protocol (SIP) (RFC 3261 [45]) packets, with typical sizes of a few hundred bytes. SIP in turn is expected to play an important role in the development towards an all-IP structure in 2.5G and 3G networks.

3.11 ROHC Summary

Robust Header Compression is an efficient compression protocol that is especially suitable for transmission of real–time multimedia data over wireless links with high error probabilities. Sophisticated compression and encoding methods and elaborated schemes that provide robustness against transmission errors make Robust Header Compression superior to the previously described header compression protocols. However, a highly increased complexity of the compression algorithms in ROHC results in an increased demand for computing power. However, the advances in microelectronics during the last few years make it possible to cost–effectively provide this computing power in small, mobile devices. Ongoing research efforts primarily focus on the tuning of the parameters of ROHC, see e.g., [46], the performance evaluation of ROHC, and its adaptation to specific wireless systems, e.g., cellular networks or wireless LANs.

3.12 Evaluation of ROHC Performance for Multimedia Services

In this section, we give an overview of performance evaluations of ROHC for wireless multimedia. We consider voice and video traffic and present the achieved compression and audio/video quality over wireless links.

For the voice traffic evaluations, voice files were coded with the GSM encoder and transmitted over an error–prone network, see [47] for details. At the receiver side the data was decoded and compared to the original data. To obtain the network traffic and compression metrics during the transmission, the NetMeter [3] tool (see Figure 14) was used. Different objective quality measures were implemented to measure the voice quality. These measures estimate the speech intelligibility and have high correlations to subjective listening tests. Whereas listening tests only require the distorted voice files, objective measures compare the undistorted with the distorted files. The distorted file is not synchronized to the original file. A synchronization algorithm [47] to allow for computation of the objective measures was developed. This algorithm allows for a frame wise computation of the objective quality measures.

Three different voice samples were used to investigate the ROHC performance. All measurements yielded a compression gain of approximately 85% for the header and about 46% for the entire packet. From the network provider's view, this allows for almost a doubling of the capacity in terms of the total number of supported customers. This is a very interesting and promising result, given the costs of the 3G frequency bands. As already mentioned in the beginning of this chapter, first compression schemes offered a good compression gain, but were vulnerable in presence of the wireless link. Therefore, the achieved compression gain has to be evaluated in conjunction with the achieved voice quality. Figure 15 gives the the voice quality (segmental Signal to Noise Ratio (sSNR) in dB) as a function of the bit error probability on the wireless link for the transmission with ROHC and without header compression. (The bit errors are uniformly distributed in this experiment which is a reasonable model for the dedicated channels in UMTS; evaluations for the common UMTS channels which are reasonably modeled by bursty bit errors are ongoing.) We observe that for both approaches the voice quality decreases as the bit error probability increases, as is to be expected. However, we also observe that with increasing bit error probability, ROHC achieves higher voice quality than the transmission without header compression. This indicates that ROHC efficiently compresses the headers and thus the packets, making the small packets less vulnerable to wireless link errors. At the same time, ROHC is robust largely avoiding incorrect header decompression due to the packets that do suffer from wireless link errors. Overall, the results indicate that ROHC has the potential to (almost) double the number of supported voice users while providing the individual users with improved voice quality.

For the ROHC evaluations for video traffic, the publicly available video sequences in Table 2 [10] were encoded with the H.26L encoder [12, 13]. These sequences were encoded with different quantization scales. The quantization scales range from 1 to 51, with 1 giving the best quality and least compression. The sequence *container* was encoded with different quantization scale settings to evaluate the effects of the transmission with and without ROHC on different video quality levels. The other sequences were encoded at an intermediate quality level (quantization scale of 30) to evaluate the effect of different video content. The result of these encodings are bitstreams with different characteristics (e.g., the different mean bit rates, frame size variabilities, etc.). These different mean bitrates and the packetization of each video frame into one IP packet (giving different mean ratios of IP headers to packet payload), result in different upper bounds on the compression gain, see Table 3. To evaluate the ROHC performance, these sequences were transmitted over the same error–prone link as the audio data. The received video sequences was decoded and their objective quality in terms of the peak signal to noise ration (PSNR) determined with the VideoMeter tool [4]. The results are presented in

Table 4. The achieved header compression is close to the theoretical limit, indicating the effectiveness of the ROHC scheme.

Interestingly, we observe that the sequence *container* has a relatively low PSNR quality for the highest quality encoding setting. This results from the large sizes of the individual video frames as only low compression is achieved for high quality encoding. The subsequent packetization results in large IP packets. The larger a packet, the more likely the packet suffers from bit errors. Thus, more high–quality packets are damaged or lost during the transmission. This results in a significant loss in quality due to the bit errors (in conjunction with the error propagation in the predictively encoded P and B frames). This effect of lower quality in the decoded file (despite higher encoded video quality) would be even slightly worse without header compression.

Generally, we observe that ROHC reduces the total required bandwidth around 10% for intermediate quality video. For low quality video streams the savings are significantly larger. We also note that the bandwidth saving will be significantly larger for layered (scalable) encoded video, which is attractive for wireless environments as it can flexibly adapt to the wireless link conditions as well as the wide range of display and processing capabilities of the different wireless devices. With layered encoded video, each encoding layer is considered as a separate stream (flow) and has its own headers [48]. This tends to reduce the average ratio of frame sizes (packet pay loads) to headers.

3.13 Deployment of Header Compression in Cellular Networks

The location of header compression in the networking protocol stack depends on the communication system. Generally, the packet header information has to be fully available for the IP routing. Header compression is therefore designed for a single link (between two adjacent IP interfaces). Furthermore, bandwidth is generally plentiful in the wired Internet compared to wireless links. Therefore, header compression is only needed for the wireless link. In the existing and upcoming cellular systems, header compression is placed in different network components. The goal of 3GPP is an ALL–IP network. Higher releases of the UMTS standard push the IP endpoint towards the radio network subsystem. With every release the topology and/or related protocols change. Following, we give the location of header compression for different generations or releases of cellular networks.

In 2.5G networks, header compression is performed by means of the sub–network dependent convergence protocol (SNDCP) [49]. The protocol is located at the mobile end–system and the supporting GPRS serving node (SGSN) as part of the GSM recommendations. SNDCP allows transparent data transfer between the mobile end–system and the SGSN. For 2.5G networks only RFC 1144 [1] and RFC 2507 [17] have been standardized. Because these compression schemes are not within our focus, we refer interested readers to [30] for more detailed implementation discussion for header compression within the GPRS subsystem.

For the next higher generation, 3GPP defines that header compression mechanisms are provided by means of the packet data convergence protocol (PDCP) specified in [50]. In Release 99 only RFC 2507 [17] is recommended for PDCP. Higher releases such as R4 and R5 introduce the usage of ROHC. For these releases, header compression is placed in the mobile end–system and the radio network controller (RNC) as specified in 3G TS 25.322. The placement in these two entities was chosen to achieve both transparency and spectral efficiency.

In Figure 17 the location of the ROHC header compression mechanism as an integral part of PDCP is illustrated as specified by the 3GPP in [50]. The figure gives an example of IP communication between two nodes. Without loss of generality, we assume that one node is connected directly to the Internet. The second node is assumed to be a wireless terminal connected to the UTRAN. Using the PDCP protocol, it communicates with the RNC via the Node B. As stated above, the ROHC is placed in the PDCP. Due to this design, the ROHC communication starts and ends at the RNC. This implicates that IP routing information is fully available at this point (looking towards the backbone) and bandwidth efficiency is achieved at the radio access network (looking towards the mobile end–system). The SGSN and the GGSN make use of the GPRS tunneling protocol (GTP) to transport IP packets between the RNC and the Internet.

A mobile end–system can be connected to several Node Bs while an IP session is ongoing. The radio network subsystem (RNS) takes care of the mobility and hands over the mobile end–systems from one Node B to another with possible changes of the RNC. An RNC with several Node B's is referred to as the serving radio network subsystem (SRNS). While the compressor/decompressor pair in the mobile end–system remains the same, new compressor/decompressor pairs have to be established in the RNC if an SRNS relocation takes place. This results in lost contexts for the RNCs. In dependency of the chosen ROHC mode (unicast, optimistic, or reliable) the performance degrades with every handover that requires SRNS relocation. Clearly, for the unicast mode the performance degradation is the highest, because the lost context will only be detected after a given time out. ROHC in contrast to other header compression schemes is designed to be robust, which in turn helps in such situation of SRNS relocation. On the other hand, UMTS cells will be smaller than installed GSM cells and thus handovers will occur more frequently. Therefore, the 3GPP specifies in [7] the possibility to forward the decompression context from the old RNC to the serving RNC (SRNC), where the new decompressor is located. In this case, the performance degradation does not occur at the expense of higher signalling overhead. But this seems reasonable since backbone capacity is not scarce, compared to the the wireless bandwidth.

4 Conclusion

In this chapter we have given an overview of the basic mechanisms for protocol header compression on wireless links. We have examined the large protocol header overhead when streaming audio and video — the key motivation for employing header compression. We have traced the evolution of header compression mechanisms from the early proposals for compressing TCP/IP headers to the mechanims designed for the RTP/UDP/IP protocol stack typically employed for streaming. We discussed the challenges of header compression on wireless links with large and varying bit error rates and large round trip times. We reviewed a number of refinements to header compression that have been developed to address these challenges. These developments have culminated in the RObust Header Compression (ROHC) framework. We gave an overview of the compression mechanisms used in ROHC. We then evaluated the compression performance of ROHC for packet voice communication and video streaming over wireless links. We found that in typical scenarios ROHC cuts the bandwidth required for voice service by almost a factor of two and at the same time improves the voice quality (expecially for larger residual bit errors that are not corrected by physical or link layer techniques). For video streaming with single layer (non-scalable) encoded video, we found that ROHC reduces the bandwidth requirement by about 10% for intermediate quality video. For lower quality video, the bandwidth reductions are significantly larger (up to about 40%). Similarly, our results indicate that ROHC will achieve significant reductions of the required bandwidth for streaming layered (scalable) encoded video (which requires a unique flow for each encoding layer). Finally, we discussed the deployment of ROHC in the evolving wireless systems which push the IP endpoint closer and closer to the wireless terminal. With the current standards, ROHC is operating between the wireless terminal and its Radio Network Controller.

5 Acknowledgements

We are grateful to Stephan Rein of acticom GmbH/TU Berlin (visiting Arizona State Univerity in spring/summer of 2003), Michele Rossi of the University of Ferrara, Italy and Andrea Carpene of the University of Milano, Italy, and Thomas Sikora of the TU Berlin, Germany, for many insightful discussions and comments on drafts of this chapter.

This work was supported in part by the National Science Foundation under grant No. Career ANI-0133252 and grant No. ANI-0136774.

References

[1] V. Jacobson, "Compressing TCP/IP Headers for Low-Speed Serial Links, Request for Comments 1144," February 1990.

[2] S. J. Perkins and M. W. Mutka, "Dependency Removal for Transport Protocol Header Compression over Noisy Channels," in *Proc. of IEEE International Conference on Communications (ICC)*, vol. 2, Montreal, Canada, June 1997, pp. 1025–1029.

[3] A. Köpsel, "NetMeter tool for evaluation of ROHC compressions," acticom GmbH, Berlin, Germany, 2002.

[4] P. Seeling, F.H.P. Fitzek, and M. Reisslein, "Videometer," *IEEE Network Magazine*, vol. 17, no. 1, p. 5, Jan. 2003.

[5] F.H.P. Fitzek, A. Kpsel, A. 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*, June 2001, pp. 344–349.

[6] 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," Request for Comments 3095, Tech. Rep., July 2001.

[7] 3rd Generation Partnership Project, "Radio Access Bearer Support Enhancements," 3GPP, Tech. Rep., 2002.

[8] J. Postel, "User Datagram Protocol," August 1980, rFC 768.

[9] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A transport protocol for real time applications, RFC 1889," January 1996.

[10] M. Reisslein, J. Lassetter, S. Ratnam, O. Lotfallah, F. Fitzek, and S. Panchanathan, "Traffic and quality characterizaton of scalable encoded video: A large–scale trace–based study," Arizona State University, Dept. of Electrical Eng., Tech. Rep., 2002, `http://www.eas.asu.edu/trace`.

[11] F.H.P. Fitzek, P. Seeling, and M. Reisslein, "H.26L Pre–Standard Evaluation," acticom GmbH, Tech. Rep., Nov. 2002, `http://www.acticom.de`, `http://www.eas.asu.edu/trace`.

[12] JVT, "JM / TML Software Encoder/decoder," http://bs.hhi.de/s̃uehring/tml/, 2002, h.26L Software Coordination by Carsten Suehring.

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

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

[15] ——, "An Improvement of TCP/IP Header Compression Algorithm for Wireless Links," in *Proc. of Third World Multiconference on Systemics, Cybernetics and Informatics (SCI'99) and the Fifth International Conference on Information Systems Analysis and Synthesis (ISAS'99)*, vol. 4, Orlando, FL, July/August 1999, pp. 39–46.

[16] A. Calveras and J. Paradells, "TCP/IP Over Wireless Links: Performance Evaluation," in *Proc. of 48th IEEE Vehicular Technology Conference VTC '98*, vol. 3, Ottawa, Canada, May 1998, pp. 1755–1759.

[17] M. Degermark, B. Nordgren, and S. Pink, "IP Header Compression, Request for Comments 2507," February 1999.

[18] M. Rossi, A. Giovanardi, M. Zorzi, and G. Mazzini, "Improved header compression

for TCP/IP over wireless links," *Electronics Letters*, vol. 36, no. 23, pp. 1958 – 1960, November 2000.

[19] ——, "TCP/IP Header Compression: Proposal and Performance Investigation on a WCDMA Air Interface," in *Proc. of the 12th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, vol. 1, Sept. 2001, pp. A–78–A–82.

[20] M. Degermak, M. Engan, B. Nordgren, and S. Pink, "Low–loss TCP/IP header compression for wireless networks," in *Proceedings of ACM MobiCom '96*, vol. 3, New York, New York, Oct. 1997, pp. 375–387.

[21] C. Perkins and J. Crowcroft, "Effects of interleaving on RTP header compression," in *Proceedings of IEEE Infocom 2000*, Tel Aviv, Israel, 2000, pp. 111–117.

[22] J. Lim and H. Stern, "IPv6 header compression algorithm supporting mobility in wireless networks," in *Proceedings of the Southeastcon 2000*, 2000, pp. 535–540.

[23] T.A. Welch, "A technique for high performance data compression," *IEEE Computer*, vol. 6, no. 17, pp. 8–19, June 1984.

[24] J. Ziv and A. Lempel, "A universal algorithm for sequential data compression," *IEEE Transactions on Information Theory*, vol. 23, pp. 337–343, May 1977.

[25] S. Casner and V. Jacobson, "Compressing IP/UDP/RTP Headers for Low-Speed Serial Links, Request for Comments 2508," Feb. 1999.

[26] M. Degermark, H. Hannu, L. Jonsson, and K. Svanbro, "Evaluation of CRTP performance over cellular radio links," *IEEE Personal Communications*, vol. 7, no. 4, pp. 20–25, 2000.

[27] K. Svanbro, H. Hannu, L.-E. Jonsson, and M. Degermark, "Wireless Real–time IP Services Enabled by Header Compression," in *Proceedings of the IEEE Vehicular Technology Conference (VTC)*, vol. 2, Tokyo, Japan, 2000, pp. 1150–1154.

[28] A. Cellatoglu, S. Fabri, S. Worral, A. Sadka, and A. Kondoz, "Robust header compression for real–time services in cellular networks," in *Proceedings of the IEE 3G 2001*, London, GB, Mar. 2001, pp. 124–128.

[29] W.-T. Chen, D.-W. Chuang, and H.-C.Hsiao, "Enhancing CRTP by retransmission for wireless networks ," in *Proceedings of the Tenth International Conference on Computer Communications and Networks*, 2001, pp. 426–431.

[30] M.A. West, L.W. Conroy, R.E. Hancock, R. Price, and A.H. Surtees, "IP header and signalling compression for 3G systems," in *Proc. of 3G Mobile Communication Technologies*, May 2002, pp. 102–106.

[31] L. Khiem, C. Clanton, L. Zhigang, and Z. Haihong, "Efficient and robust header compression for real–time services," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, vol. 2, Chicago, IL, 2000, pp. 924–928.

[32] Z. Kostic, Q. Xiaoxin, and L. Chang, "Impact of TCP/IP header compression on the performance of a cellular system," in *Proceedings of the Wireless Communications and Networking Conference*, vol. 1, Chicago, IL, 2000, pp. 281–286.

[33] H. Liao, Q. Zhang, W. Zhu, and Y.-Q. Zhang, "A robust TCP/IP header compression scheme for wireless networks," in *Proceedings of the IEEE International Conference on 3G Wireless and Beyond*, San Francisco, CA, June 2001.

[34] G. Boggia, P. Camarda, and V.G. Squeo, "ROHC+: A New Header Compression Scheme for TCP Streams in 3G Wireless Systems," in *Proceedings of the IEEE International Conference on Communications (ICC)*, vol. 5, 2002, pp. 3271–3278.

[35] C. Jiao, L. Schwiebert, and G. Richard, "Adaptive header compression for wireless networks," in *Proceedings of the 26th Annual IEEE Conference on Local Computer Networks*, Nov. 2001, pp. 377–378.

[36] G. Pelletier, Q. Zhang, L.-E. Jonsson, H. Liao, and M. West, "RObust Header

Compression(ROHC): TCP/IP Profile (ROHC-TCP)," Nov. 2002, rFC–draft, work in progress.

[37] R. Price, R. Hancock, S. McCann, M. A. West, A. Surtees, P. Ollis, Q. Zhang, H. Liao, W. Zhu, and Y.-Q. Zhang, "TCP/IP compression for ROHC, proposed standard, draft-ietf-rohc-tcp-epic-02.txt," Nov. 2001.

[38] J. Lilley, J. Yang, H. Balakrishnan, and S. Seshan, "A unified header compression framework for low-bandwidth links," in *Proceedings of ACM MobiCom*, 2000, pp. 131–142.

[39] D. Farinacci, T. Li, S. Hanks, D. Meyer, and P. Traina, "RFC 2784: GRE: Generic routing encapsulation," Mar. 2000.

[40] G. Dommety, "RFC 2890: Key and sequence number extensions to GRE," Sept. 2000.

[41] L.-A. Larzon, M. Degermark, S. Pink, and G. Fairhurst, "The UDP-Lite Protocol," IETF–INTERNET-DRAFT, Tech. Rep., December 2002, draft-ietf-tsvwg-udp-lite-01.txt.

[42] Y. Wang and Q. Zhu, "Error control and concealment for video communication: A review," *Proceedings of the IEEE*, vol. 86, no. 5, pp. 974–997, May 1998.

[43] W. Simpson, "The Point–to–Point Protocol (PPP)," Tech. Rep., jul 1994, rFC 1661.

[44] R. Price, C. Bormann, J. Christoffersson, H. Hannu, Z. Liu, and J. Rosenberg, "RFC 3261: SigComp: Signaling compression," Jan. 2003.

[45] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "RFC 3261: SIP: Session initiation protocol," June 2000.

[46] B. Wang, H. Schwefel, K. Chua, R. Kutka, and C. Schmidt, "On implementation and improvement of robust header compression in UMTS," in *Proceedings of the*

13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, 2002, pp. 1151–1155.

[47] S. Rein, “Performance Measurements of Voice Quality over Error–Prone Wireless Networks using Robust Header Compression,” Master’s thesis, Communication Systems Group —Technical University of Berlin, March 2003.

[48] S. Ekmekci and T. Sikora, “Unbalanced Quantized Multiple Description Video Transmission using Path Diversity,” in *IS&T/SPIE’s Electronic Imaging 2003*, 2003, santa Clara, CA.

[49] M. J. Shah, “IP header compression in the SGSN,” in *Proc. of IEEE SoutheastCon*, 2002, pp. 158–161.

[50] 3rd Generation Partnership Project, “Packet Data Convergence Protocol (PDCP) Specification,” 3GPP, Tech. Rep., 2002.

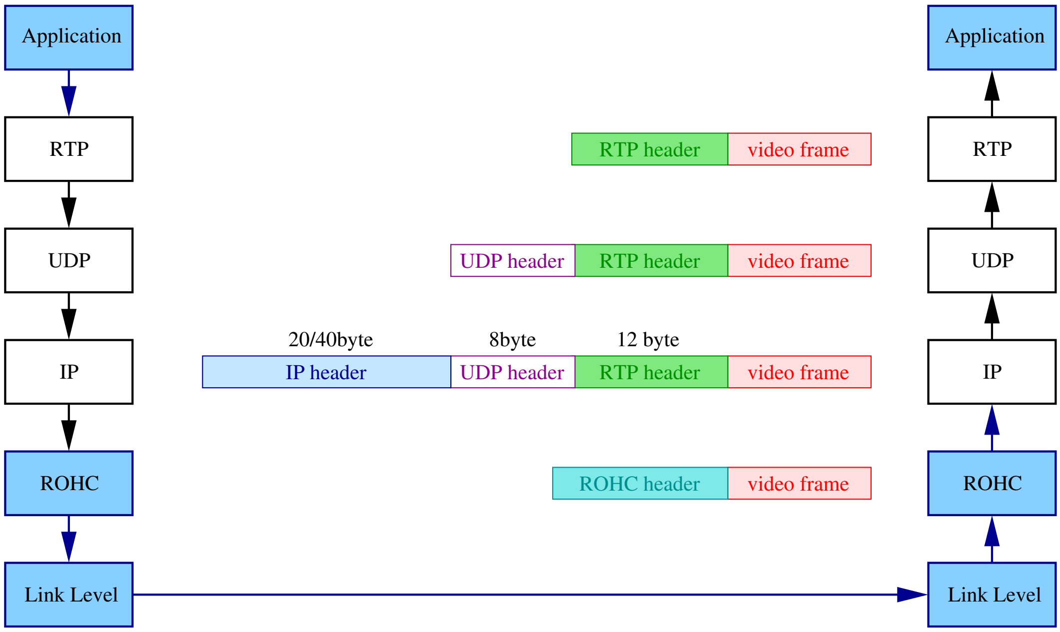

Figure 1: Header structure and protocol stack with relevant layers (ROHC, IP, UDP, and RTP).

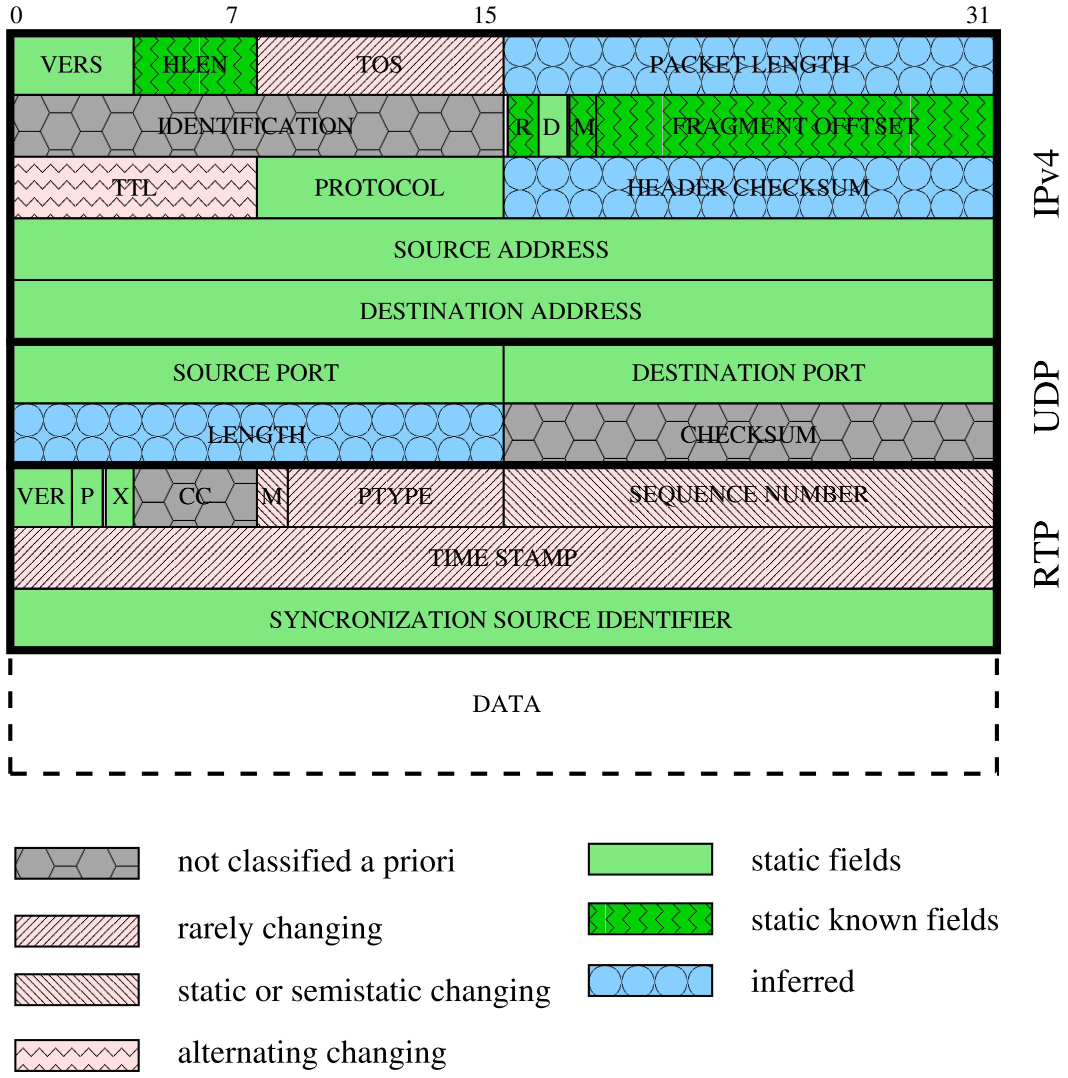

Figure 2: Header fields for RTP/UDP/IP packets (Version 4) and their dynamics.

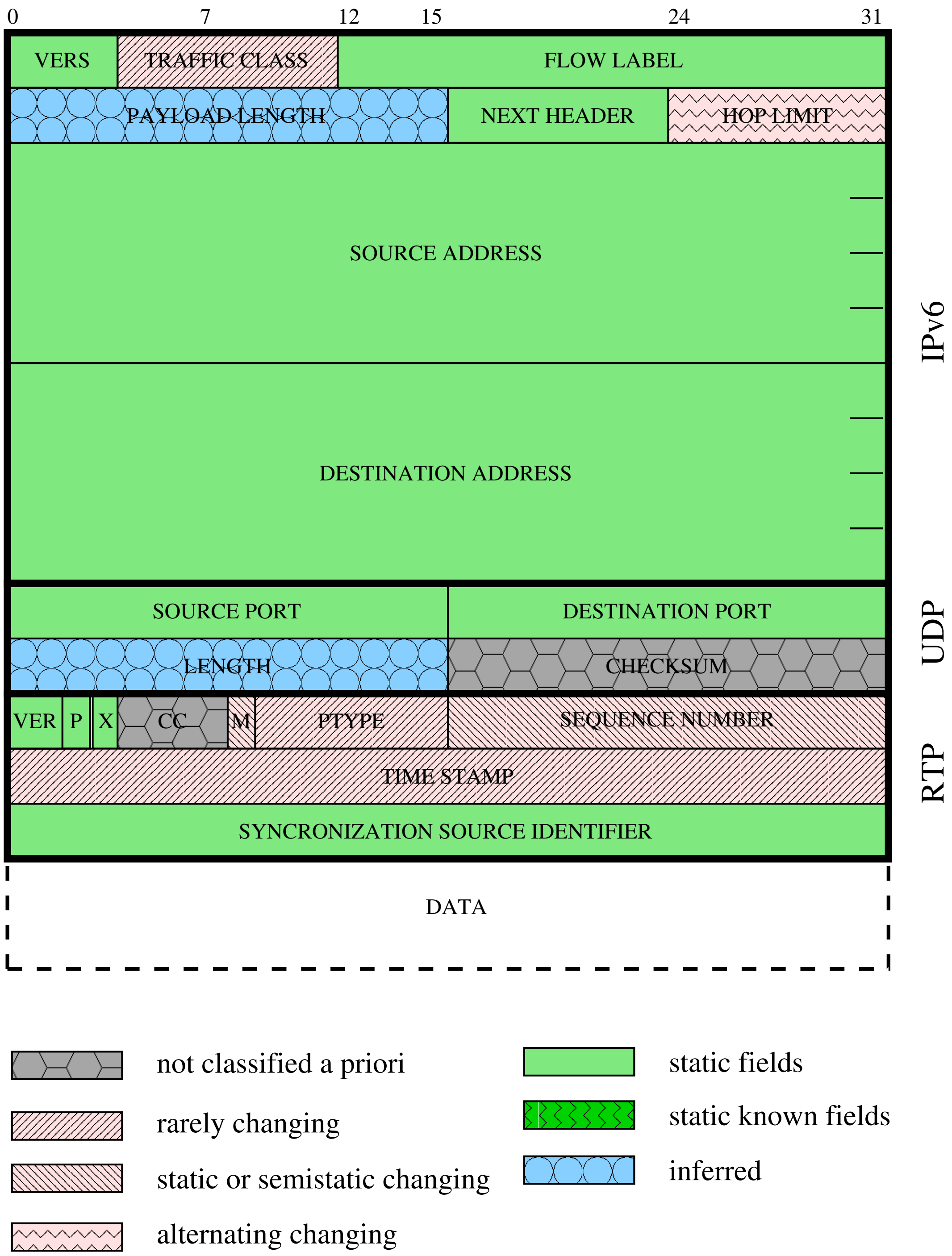

Figure 3: Header fields for RTP/UDP/IP packets (Version 6) and their dynamics.

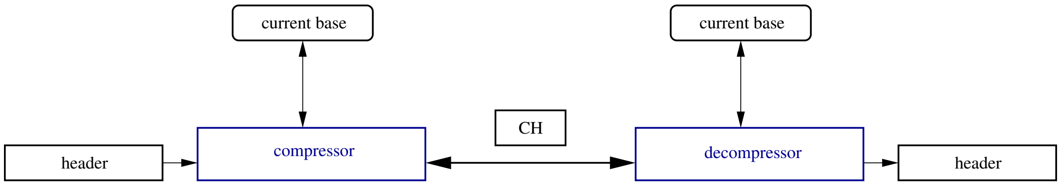

Figure 4: General concept of header compression: A current base header is maintained at compressor and decompressor. Header redundancy with respect to the base header is removed to obtain the compressed header (CH).

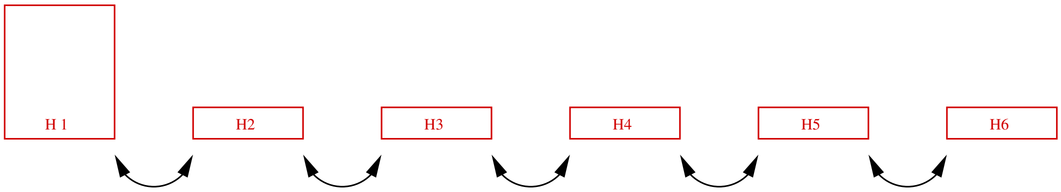

Figure 5: Van Jacobson [1]: Delta coding with respect to immediately preceeding header.

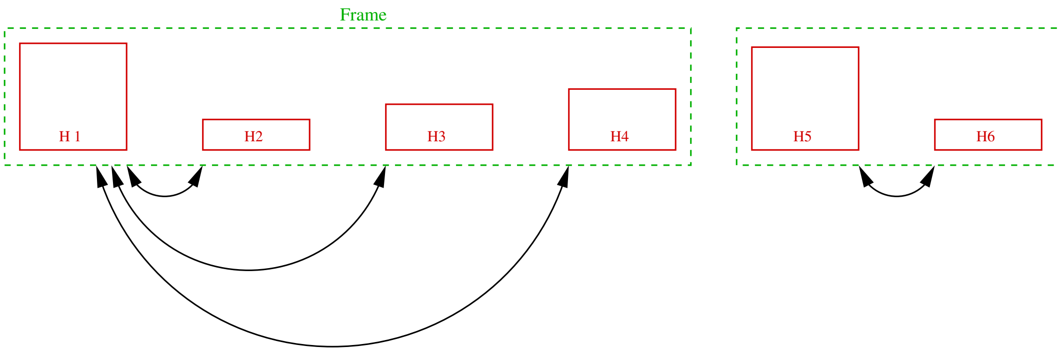

Figure 6: Perkins and Mutka [2]: Delta coding with respect to first (uncompressed) header in frame.

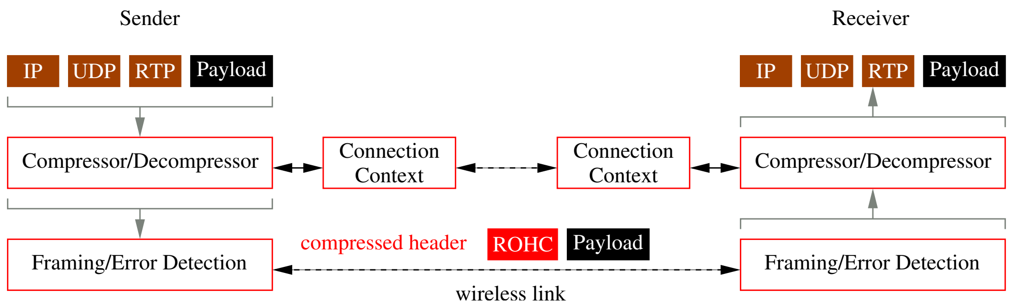

Figure 7: Transmission of compressed packets between ROHC compressor and decompressor.

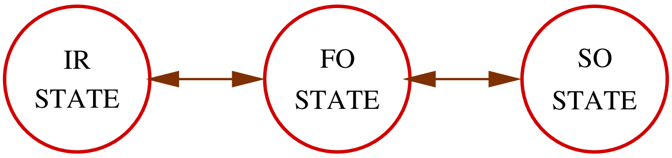

Figure 8: Compressor States.

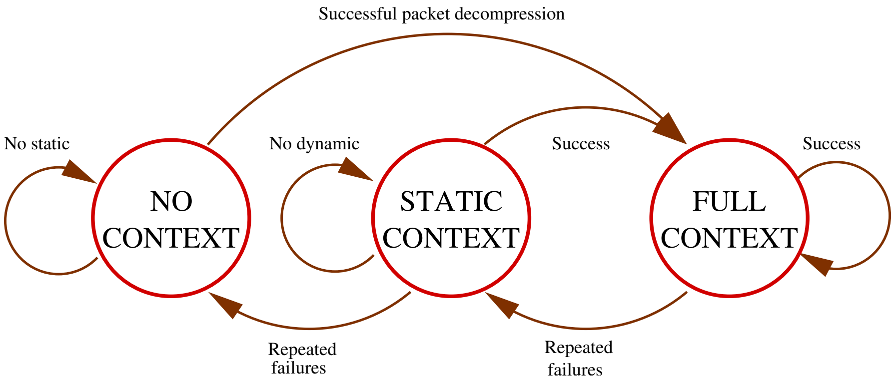

Figure 9: Decompressor States for all modes.

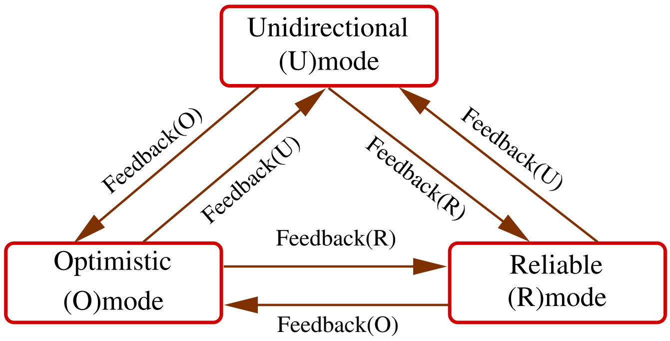

Figure 10: Mode transitions.

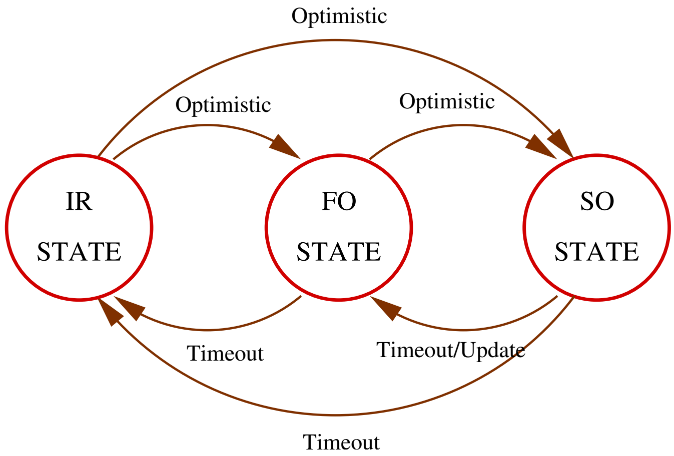

Figure 11: Compressor State transitions for Unidirectional mode: After repeatedly sending a packet with lower (IR or FO state) compression the compressor optimistically transitions to the FO or SO state. Updates in the header field pattern and a periodic timeout return the compressor to the FO state. A longer period timeout takes the compressor to the IR state.

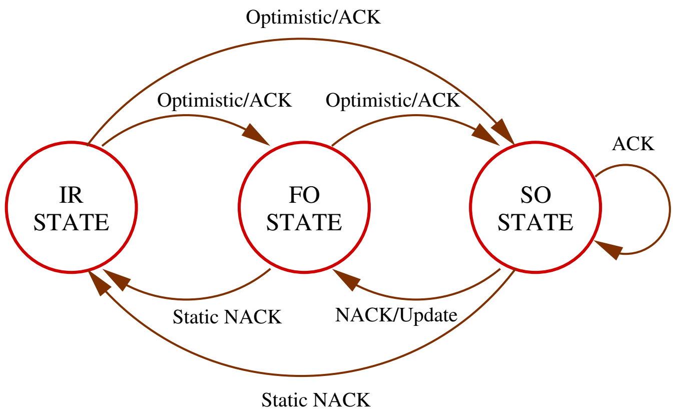

Figure 12: Compressor State transitions for Bidirectional Optimistic mode: After receiving an (optional) ACK from the decompressor or repeated transmissions (and optimistic assumption of established context) the compressor transits upward. The compressor returns to the FO state to update the header field development pattern or upon request (NACK) from the decompressor. With a Static–NACK the decompressor requests a static context update.

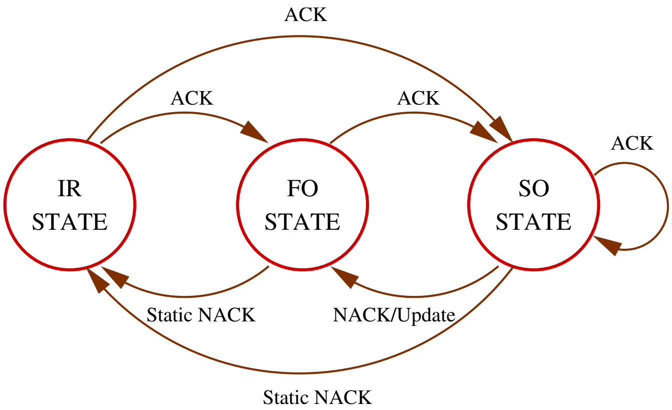

Figure 13: Compressor State transitions for Bidirectional Reliable mode: The compressor transitions upward only after receiving ACKs from the decompressor. Updates of the context or decompressor requests (NACK, Static–NACK) cause downward transitions.

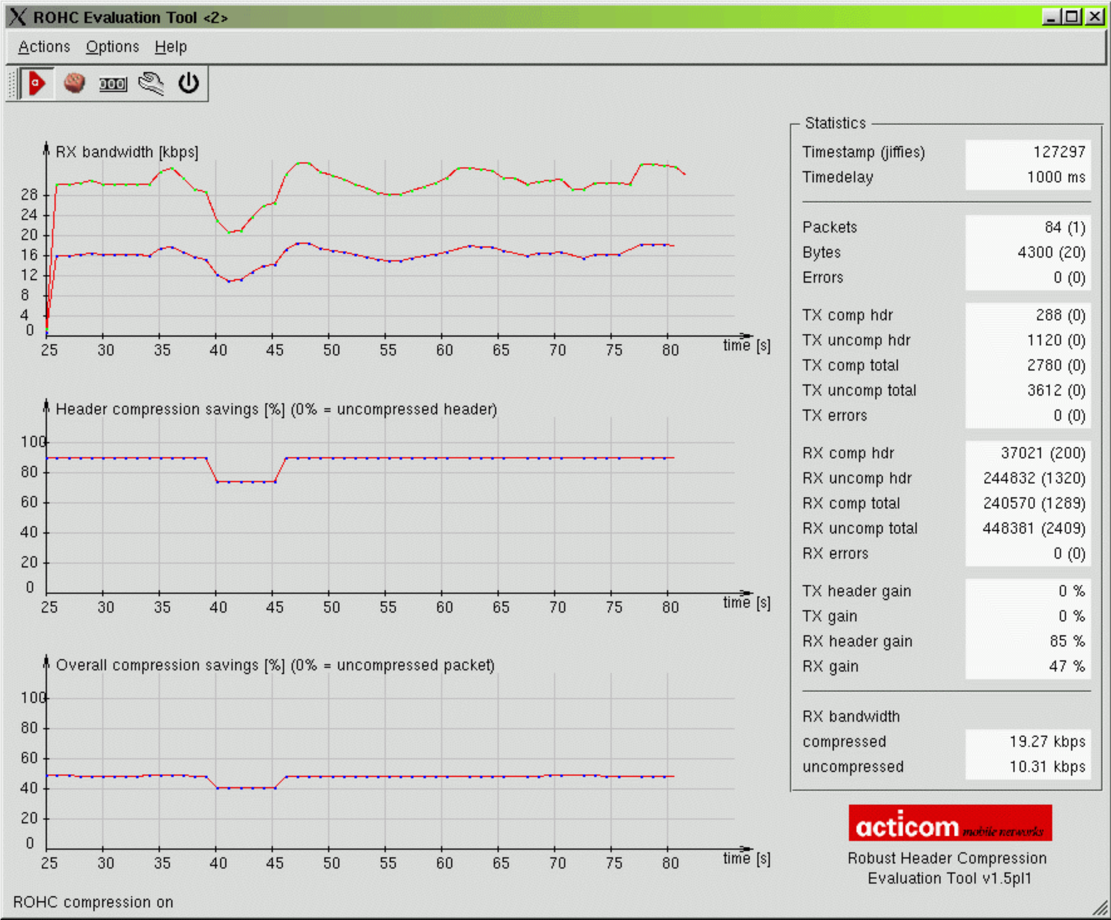

Figure 14: NetMeter tool for the evaluation of ROHC compression [3].

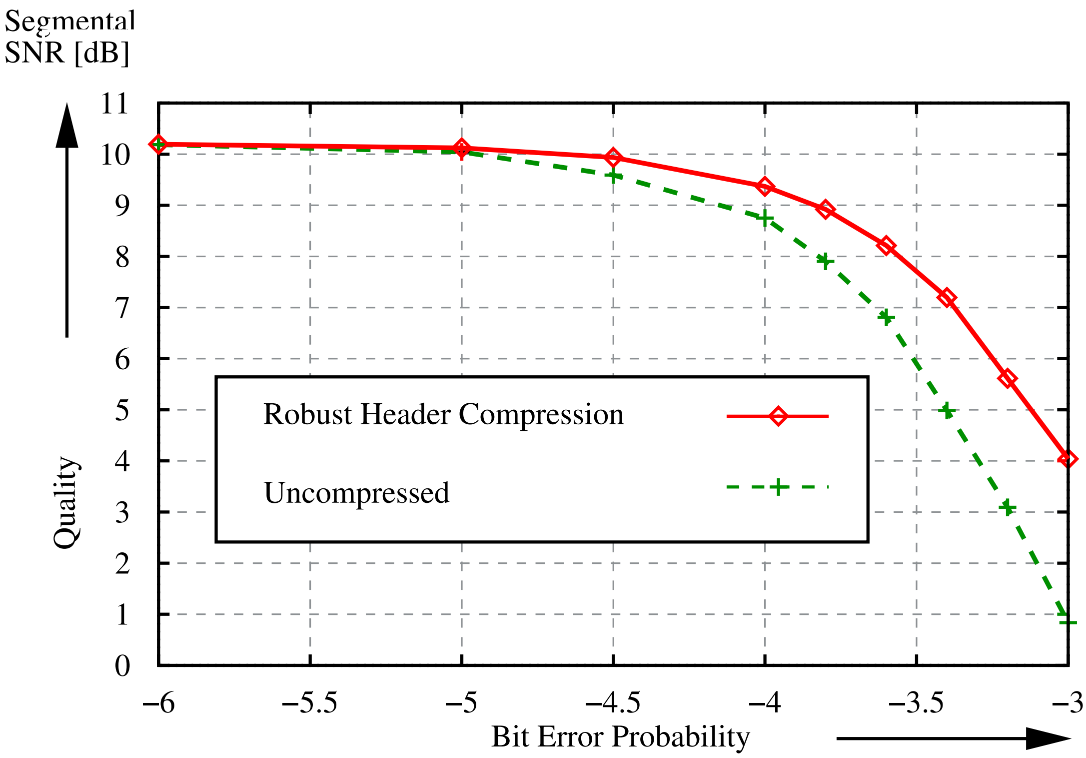

Figure 15: Objective voice quality (segmental SNR) as a function of bit error probability on a wireless link for transmission with ROHC and without header compression: ROHC improves the voice quality for moderate to large bit error probabilities, while using roughly 46% less bandwidth than transmission without header compression.

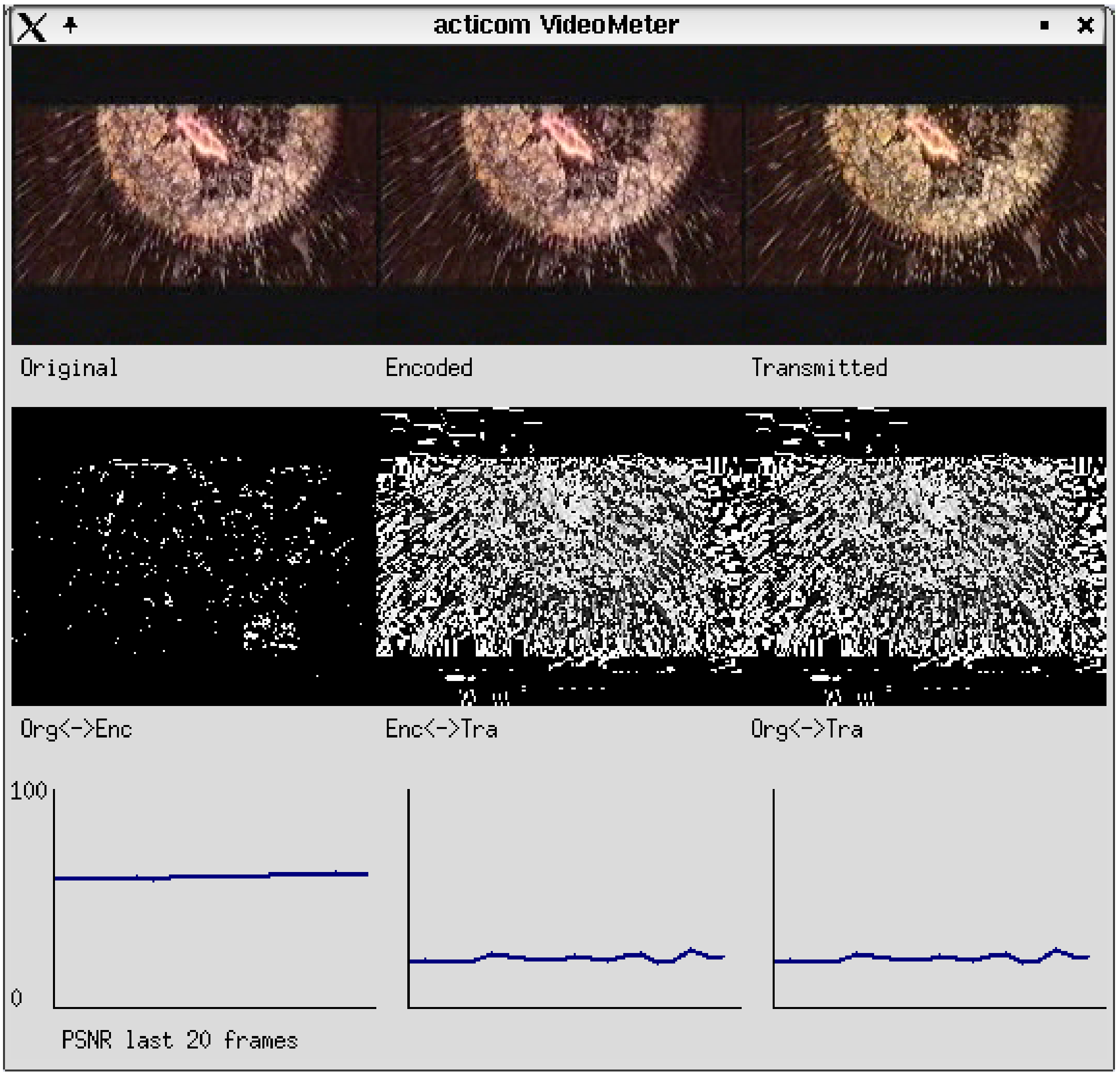

Figure 16: VideoMeter [4] for video quality evaluation.

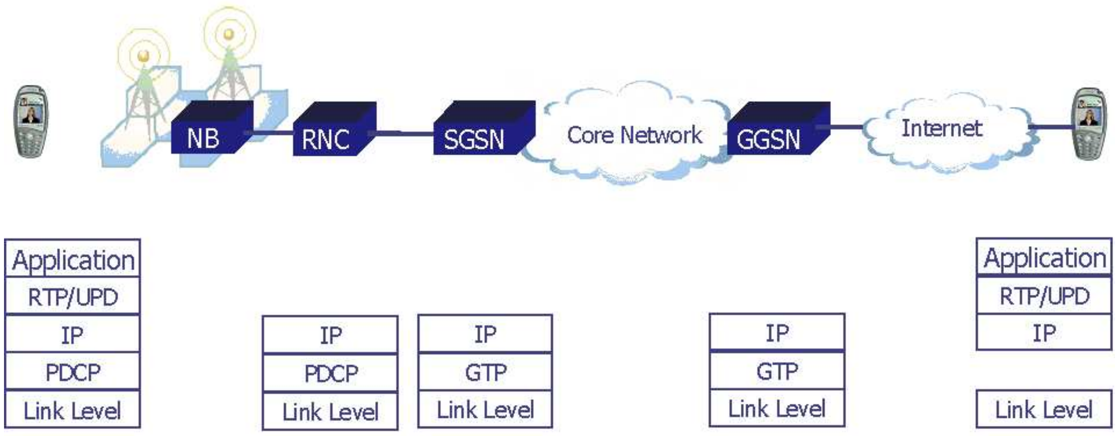

Figure 17: Location of ROHC in UMTS transmission chain. ROHC is provided by the packet data convergence protocol (PDCP) between the mobile end system and RNC.

Table 1: Theoretical upper bound savings (in terms of bandwidth) for voice traffic.

codec	mean bit rate [kbps]	IPv4 $\overline{S}$ [%]	IPv6 $\overline{S}$ [%]
LPC	5.6	74	81
GSM	13.2	55	65
ITU-T G.711	60.0	21	29

Table 2: Transmitted video streams.

Sequence	Quant. scale	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 and daughter	30	961
news	30	300
salesman	30	449
silent	30	300

Table 3: Upper bound on total bandwidth savings $\overline{S}$ for H.26L encoded video (QCIF) for different video encoding quantization scales (smaller quantization scales give higher encoded video quality) and different video sequences.

Video sequence	Quant. scale	Mean bit rate [kbps]	IPv4 $\overline{S}$ [%]	IPv6 $\overline{S}$ [%]
container	10	855	1.1	1.6
container	20	213	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 and daughter	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

Table 4: Average header size, header compression (relative to 40 byte IPv4 header), total bandwidth savings, and average video frame PSNR quality for H.26L video over error–prone link (BEP=10^{-5}, QCIF, ROHC in optimistic mode).

Video sequence	Quant. scale	Avg. header [byte]	Header compr. [%]	Network Total BW sav. [%]	Avg. vid. frame PSNR [dB]
container	10	6.4	84.2	0.9	29.8
container	20	6.5	83.7	3.6	37.8
container	30	6.3	84.3	10.5	34.6
container	40	6.3	84.3	23.0	29.2
container	51	6.5	83.7	40.0	22.4
bridge close	30	6.2	84.5	8.6	18.3
carphone	30	6.4	83.9	5.5	33.1
claire	30	6.2	84.6	14.7	37.9
foreman	30	6.4	83.9	6.0	27.6
grandma	30	6.2	84.5	12.0	35.0
highway	30	6.3	84.3	10.2	34.1
mother and daughter	30	6.2	84.6	10.5	38.9
news	30	6.6	83.6	7.1	35.2
salesman	30	6.2	84.5	8.8	33.8
silent	30	6.3	84.3	7.2	32.4