Periodica Polytechnica
Electrical Engineering
and Computer Science

60(3), pp. 151-162, 2016
DOI: 10.3311/PPee.8958

Performance Evaluation of Network Header Compression Schemes for UDP, RTP and TCP

Máté Tömösközi[1,2,5*], Patrick Seeling[4], Péter Ekler[5], Frank H. P. Fitzek[1,2,3]

RESEARCH ARTICLE

Received 31 December 2015; accepted 16 March 2016

Abstract

Modern cellular networks utilising the Long–Term Evolution (LTE) set of standards face an ever–increasing demand for mobile data from connected devices. Header compression is employed to minimise the overhead for IP–based cellular network traffic.

In this paper, we evaluate the three header compression implementations used by such networks with respect to their potential throughput increase and complexity for different mobile service scenarios. We compare RTP, UDP and TCP profile compressions regarding their compression gain and complexity. Specifically, we consider header compression as defined by (i) IP Header Compression (RFC 2507), (ii) Robust Header Compression version 1 (RFC 3095), and (iii) the recently updated Robust Header Compression version 2 (RFC 5225) with TCP/IP profile (RFC 6846).

This paper presents the performance evaluation of these header compression schemes for UDP, RTP and TCP, for both IPv4 and IPv6 streams in error–free and error–prone scenarios. A comparison between the Robust Header Compression methods and IP Header Compression is also provided. Our results show that all implementations have great potential for saving bandwidth in IP–based wireless networks, even under varying channel conditions. We also present for the first time an analysis of certain RTP header fields which, depending on the transmission characteristics, could have high impact on the overall compression gain.

Keywords

Robust header compression, Mobile multimedia, Cellular networks, Bandwidth savings

[1] acticom GmbH, Berlin, Germany

[2] Department of Electronic Systems, Aalborg University, Denmark

[3] Communication Networks Group, Technische Universität Dresden, Germany

[4] Department of Computer Science, Central Michigan University, Mount Pleasant, MI 48859

[5] Department of Automation and Applied Informatics, Faculty of Electrical Engineering and Informatics, Budapest University of Technology and Economics, Hungary

[*] Corresponding author, e-mail: mate.tomoskozi@aut.bme.hu

1 Introduction

Mobile data is increasingly transported over the Internet Protocol (IP) in both versions (IPv4 as well as IPv6), especially with the implementations of third generation and long–term evolution (LTE) networks of the fourth generation (4G). The increased data rates achievable in these newer networks allow for further media convergence on mobile devices, such as video conferencing or video streaming. Wireless systems in general, but especially cellular networks, in turn face increasing data consumption demands from mobile consumers – a trend that will likely continue in the foreseeable future, as indicated by, e.g., [20]. In addition to media consumption, there is a huge demand for data and voice transmissions (e.g., VoLTE, Voice over IP) for mobile devices. Similarly, the proliferation of social networks and similar applications are requiring many signalling messages for mobile scenarios. Because of the widespread adoption of LTE in cellular networks using IP–based services around the globe, a large disparity exists between the common packet payloads and the headers required for transmissions. Header compression methods can be applied to reduce the encapsulation overheads and, in turn, these compression methods can save significant amounts of bandwidth.

Due to a slower pace of capacity building through increased infrastructure, which is partially driven by the high costs of associated investments, the capacity increase due to the optimisation of mobile communications in cellular networks has attracted a great deal of research and practical implementations in the past. The motivation behind our present research is the comparison of both Robust Header Compression versions and the IP Header Compression commonly encountered in today's cellular networks. Additionally, our evaluation provides a comparative analysis based on implementations of these schemes currently used by network infrastructures around the world.

All IP packets carry the typical protocol encapsulation overheads of one or more protocols from the IP protocol stack, independent from actual payload sizes. For mobile multimedia settings, a common protocol combination is RTP/UDP/IP. These protocols specifically account for an encapsulation overhead of 40 bytes with IPv4 and 60 bytes with IPv6 for each individual

packet carried over the network. A large portion of IP and UDP packets (such as source and destination addresses or port numbers, etc.) could be omitted for most of the packet traffic as they will stay constant throughout the transmission. Other fields, such as the IP identification field or TCP sequence numbers, can be derived from a single master sequence number.

In the past, research and implementation efforts targeted a reduction of these protocol overheads, as the effect of reduced transmission sizes of individual network packets quickly multiply. The overall approach for compression of the protocol headers is based on a compressor/decompressor concept, whereby both are located between the data link layer and the network layer on a sender/receiver pair's protocol stack implementations. The reduction in protocol overheads, typically comprised of RTP/UDP/IP protocol headers in, e.g., a voice over IP (VoIP) scenario, exploits the redundancy commonly encountered (*i*) among the different headers of individual packets and (*ii*) between consecutive packets belonging to the same IP flow. Several header compression mechanisms for different protocol combinations were proposed in the past, starting with CTCP, introduced by Van Jacobson [12], which focuses on the TCP/IP protocol combination. IP Header Compression (IPHC) [4] provides several enhancements to CTCP, especially support for UDP and IPv6, while the Compressed Real Time Protocol (CRTP) [5] added support for RTP. Several enhancements and modifications for specific scenarios based on these main protocols exist, see, e.g., [10] for a brief overview of header compression history prior to Robust Header Compression (RoHC).

RoHC, as defined in RFC 3095 [2] by the Internet Engineering Task Force (IETF), has shown great results and was incorporated into 3GPP–UMTS specifications and standards for WiMAX networks, giving it significant industry adaptation. RoHC, just as most of the other header compression schemes, consists of a compressor and decompressor pair, which operate over different types of networks (e.g., unidirectional or bidirectional). Different protocol suites can be supported by RoHC through profiles (e.g., RTP/UDP/IP or TCP/IP).

While immediate savings due to the reduction of packet sizes carried over the network exist, additional indirect savings are achieved as well. The reduced amounts of data require less time of network interface activity, which can ultimately result in possible energy savings for the mobile device and network operator. In turn, batteries could last longer and mobile network providers could reduce their energy footprint, effectively "greening" their networks.

With the broad proliferation of the initial version of RoHC (RoHCv1) in the mobile realm (especially in cellular markets), a revision was proposed in RFC 5225 [17]. This updated version (RoHCv2) widens the possibilities and implementation details of the original RoHC, without obsoleting it. (We note that this applies in the same fashion to RoHCv1 enhancements defined in follow–up RFCs 3843, 4019, 4224, 4995 or 4997.) The methods and approaches to header compression within the original RoHC and RoHCv2 schemes are not directly subject to patent–based restrictions; however, several additional improvements using existing standardised implementations were subject to patenting efforts in the past. RoHCv2 targets increases in simplicity and robustness under similar network conditions while maintaining or increasing performance, see [17] for details concerning this new design.

Albeit these different design choices are available from network and device provider points of view, little comparative evaluation has been performed to assess the advantages for each scheme under realistic mobile service scenario conditions.

The rest of this paper is structured as follows. In the succeeding section, we briefly note previous research that has been performed in this area. We continue in Section 3 with a description of the overall experimental configuration and the performance metrics utilised to evaluate compression and complexity performance. In Section 4 we compare audio streams compressed by the RTP and the UDP profiles regarding both the achievable savings and the complexity used during compression and decompression. Subsequently, in Section 5 we look at the difference between IPv6 and IPv4 compression. Section 6 evaluates the compression for TCP streams. In Section 7 we determine the compression gain on error–prone channels. Finally, we analyse the impact of certain protocol fields on compression gain in Section 8 before we conclude in Section 9.

2 Related Work

There are several compression profiles defined for RoHC version 1 and version 2, which in some cases are completely optional and extend the core RTP/UDP/IP compression with further protocols or protocol combinations. The same is true for IPHC, albeit the RFC (see [12]) doesn't explicitly define profiles, but it separates UDP and TCP compression and provides an implementation hook for CRTP compression.

Nonetheless, prior evaluations focused only on RTP and mostly considered RoHCv1. Articles made with the same implementation focused, e.g., on the impact of RoHC on media performances have found that header compression cuts the required bandwidth in half for voice transmissions (GSM) and improves the overall voice quality (see [19] and [11]). Later they also found that video quality can be enhanced as well (see [21, 11]).

Furthermore, RoHCv1 and RoHCv2 performance was evaluated for RTP in [15] and was shown that both versions perform equally well for Voice–over–IP transmissions, albeit RoHCv2's gain is slightly better by 5–10 %. [15] also shows, that RoHCv2 uses more complexity on the compressor side, but is much faster during decompression than RoHCv1.

Research into the application of header compression schemes to multi–hop (ad–hoc) networks has revealed, that RoHC and IPHC perform reasonably well in such scenarios too

(see [8, 7, 1]). [14] proved, that an end–to–end compression scheme would reduce the delay in time–critical system, which is a downside of using point–to–point compression techniques like IPHC and RoHC.

For RoHCv1 it was highlighted in [13] and [9], that the compression savings heavily depend on the compressor's mode. The non–stateless nature of the compression also poses a potential security risk and can be mitigated by encryption and authentication techniques as highlighted in [6].

3 Methods and Metrics

In this paper, we evaluate the performance of RoHCv1, RoHCv2 and IPHC reference implementations provided by acticom GmbH[1] with respect to bandwidth savings and complexity through error–free and error–prone scenarios utilising both IP versions, as well as UDP, RTP and TCP protocols.

For the measurement testbed, we used two separate and independent setups. For all UDP and RTP streams we have captured and stored the outgoing packets on the link layer as to allow consecutive evaluations with the same underlying data stream. We then applied artificial packet losses (if any) before decompression. We were able to do this since UDP doesn't define any feedback of its own (i.e. it isn't connection based as TCP).

However, we had to take a different approach with TCP. We employed two Raspberry Pi machines equipped with WiFi units and connected these devices to each other via an ad–hoc connection. We also used the Unix/Linux platform's TUN/TAP interface to create a tunnel to which we directed a normal TCP data stream for compression from the Internet Layer. The benefit of doing this is, that we could simply use a regular application to drive the compressed transmission. This way, any losses that were artificially induced by us would be handled directly by the Linux kernel after the redirection of the decompressed stream into the operating system's protocol stack.

The Raspberry Pi platform was chosen due to broad availability, for its suitability as a platform for simulating nodes and because the platform had already shown the potential to support operations at high speeds. Specifically, this particular platform can be regarded as a general baseline for current mobile device capabilities.

Regarding the measurement metrics, an idealistic upper bound on the possible network bandwidth savings can be calculated by assuming that the compressed header size is zero, in which case for each generated packet i the savings S_i are given by

$$S_i = 1 - \frac{P_i}{UH + P_i} = \frac{UH}{UH + P_i} \tag{1}$$

where P_i denotes the payload data size of the i^{th} packet, and UH denotes the size of the uncompressed protocol headers, i.e., the protocol encapsulation overhead. The average savings for a sequence or session of N packets are in turn calculated as

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

describing the portion of the bandwidth that can be saved from a network provider point of view. To illustrate this using an example, we consider the AMR codec with the smallest payload of 12 bytes. In an ideal world, we could compress the protocol headers to zero byte. Assuming a full RTP/UDP/IPv4 or RTP/UDP/IPv6 encapsulation, the upper bound for savings, as given beforehand, equals 77 % and 83 %, respectively.

For a more realistic scenario, we need to note that the compressed header size of an individual packet CH_i is greater than zero, i.e., $CH_i > 0$. In turn, we derive actual performance measures similar to [21] as (*i*) the actual savings (or alternatively the gain) of the encapsulation overhead (headers) as

$$S_H(i) = \frac{UH - CH_i}{UH} \tag{3}$$

(*ii*) the actual savings for individual packets as

$$S_P(i) = \frac{UH - CH_i}{UH + P_i} \tag{4}$$

and (*iii*) the respective average savings according to Eq. (2). In addition to the bandwidth savings, we evaluate the compression performance by means of the complexity of the compression/decompression methods employed in both RoHC versions and IPHC. We measure the complexity through CPU time–stamping, i.e., the time required to compress (or decompress) the i^{th} packet in the stream, as t_i and compare it to the previous time stamp obtained at time t_{i-1}. This complexity or utilised time value could readily be mapped to mobile device power consumptions (see, for example, [3]), which is currently being investigated by the authors of this article.

4 RTP and UDP Audio Stream Compression

We initially present results for real–world measurement using the two RoHC versions and IPHC over a wireless channel presumed to be error–free. To enable measurements under common real–world scenarios, we utilised the Asterisk VoIP server[2], which connected a fixed desktop client and an Android smartphone, both using the ZoIPer VoIP client software[3]. This configuration employed the GSM 06.10 codec, which is the full–rate audio codec version that results in 33 bytes of payload.

4.1 Compression Savings

Figure 1 illustrates the achieved savings for the encapsulation overhead for RTP.

1 See http://www.acticom.de

2 See https://www.asterisk.org/ for details

3 See http://www.zoiper.com/ for details.

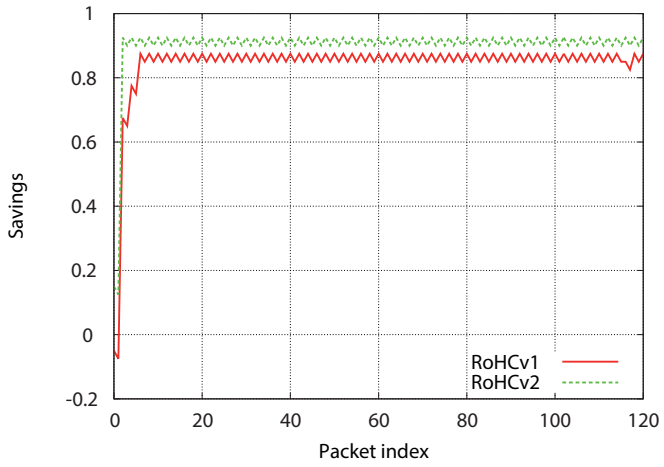

Fig. 1 Average header savings $S_H(i)$ attained with RoHCv1 and RoHCv2 RTP profile reference implementations for a VoIP call with GSM full rate codec using IPv4 over WLAN.

Initially, we note that the observed savings follow a "ramp-up" phase in the beginning of the measurement, which is followed by a continuous high level featuring slight oscillation. We furthermore observe that for the duration of the call, both versions achieve savings above 80 %, with RoHCv2 attaining higher levels. Clearly, this "ramp-up" characterises both RoHC versions transitioning from their default states to fully established context states. In turn, initially there is a slight overhead for context establishment, which then gives way to static field compression savings, followed by full savings realised through the full context.

Figure 2 illustrates the resulting packet–indexed savings $S_H(i)$ for UDP. We observe that the compressed header savings typically range from 50 % to about 60 % for both RoHC versions, while they are around 85 % for IPHC, which exhibits occasional drops in savings due to sending uncompressed refreshes. We note that in this evaluation, the best possible setup for each compression method was employed, i.e., while RoHC can rely on feedback to signal invalid contexts, IPHC lacks such functionality when compressing UDP. Therefore, IPHC must use periodic refreshes employing an exponential function to tackle invalid contexts. Such mechanism is also defined for RoHC (employing the *optimistic approach*), which would also exhibit similar characteristics to IPHC. We also observe, that RoHCv2 has a slightly better compression ratio than RoHCv1 (with about 5 % difference).

All–in–all, compressing RTP streams using only the RoHC UDP profiles provide about 20 % less *effective* savings than with RTP. However, IPHC is more efficient in this regard. It must be also noted, that for the calculation of the savings we always took into account the compressed profile's uncompressed header size. Meaning, that the value of UH in Eq. (1) is always less for UDP than for RTP (the RTP header is considered to be payload). Therefore a header saving of 90 % with RoHCv2 in RTP profile and a header saving of 90 % with IPHC UDP compression aren't equally efficient, since the combined uncompressed header sizes of a RTP/UDP/IP packet in UDP profile is considered less than with the RTP profile.

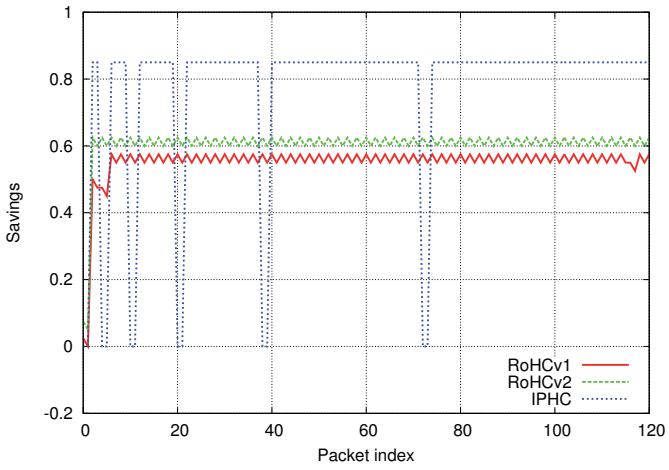

Fig. 2 Header savings $S_H(i)$ attained with RoHCv1, RoHCv2 UDP profile and IPHC reference implementations for a VoIP call with GSM full rate codec using IPv4 over WLAN.

4.2 Compression Complexity

Next, we evaluate the compression and decompression complexity by means of CPU–level timestamps, which can be mapped directly to CPU cycles utilised for (de)compression and energy consumption in turn. First, we present the complexity for RoHC RTP profiles in Fig. 3.

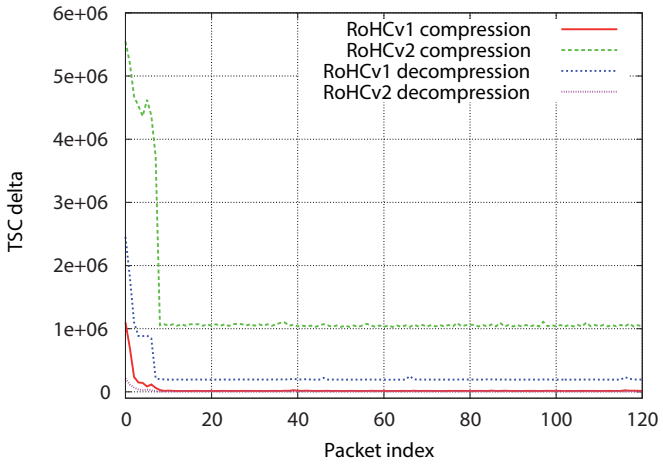

Fig. 3 Header (de)compression complexity measured by CPU timestamps (TSC) attained with RoHCv1 and RoHCv2 RTP profile reference implementations for VoIP call with GSM full rate codec using IPv4 over WLAN.

We observe that initially, a high level of complexity (as measured through timestamps) is followed by a sharp drop and steady lower level for both compression and decompression in all versions. The initial high levels of complexity directly correlates with the observations made for the savings in this scenario. For the initial packets, the compressor and decompressor have to establish different levels of context states, which

correspond to local resource reservations and compression for static fields, among other tasks. Once the context is established, a fairly low level of complexity follows for both versions. We can attribute the lower overall level to the availability of feedback in this bidirectional case.

Next, we illustrate the compression timestamps for the UDP compression in Fig. 4. Comparing the different compression levels, we observe that RoHCv1 and IPHC exhibit excellent performance, as indicated by almost negligible times required for compression. On the other hand, we note that the compression of RoHCv2 is more CPU intense.

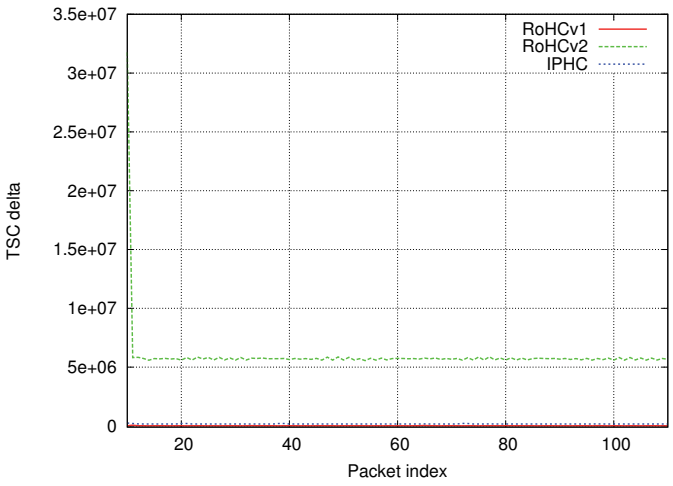

Fig. 4 Header compression complexity measured by CPU timestamps (TSC) attained with RoHCv1/RoHCv2 UDP profile and IPHC reference implementations for a VoIP call with GSM full rate codec using IPv4 over WLAN.

In Fig. 5, we illustrate the corresponding decompression complexity. We initially observe that the timestamp levels are lower by about 2 orders of magnitude for all decompressions for both versions of RoHC. Comparing the RoHC versions to IPHC, we identify a significant difference between these approaches, as RoHC requires only an approximate quarter of the CPU time required by IPHC. We furthermore note that the "spikes" for the timestamps of IPHC correspond to the periodic refreshes.

Overall, we additionally observe a fairly narrow range (except for minor outliers) of (de)compression times, indicating a generally stable performance. Comparing UDP and RTP compressions, it can be said, that the RoHCv2 compression is the faster one in both cases. However, in general UDP compression should be slightly better than RTP compression, since fewer header dynamics are needed to be taken into account during the execution.

5 IPv6 Stream Compression

We illustrate the RTP/UDP/IPv4 header compression savings in Fig. 6 for an audio only scenario based on a streaming session using the VLC multimedia player.

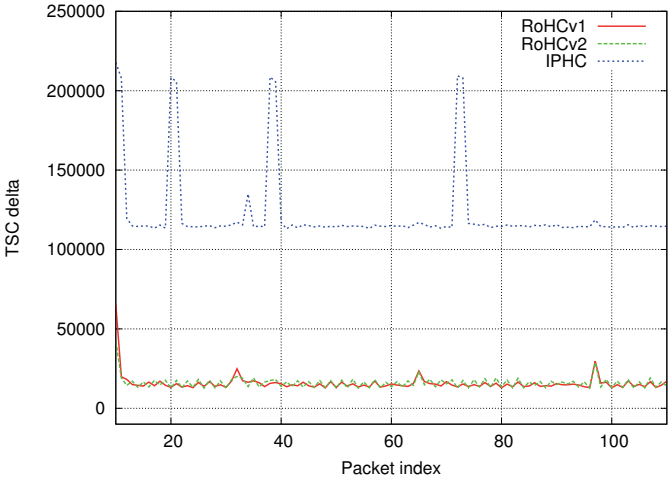

Fig. 5 Header decompression complexity measured by CPU timestamps (TSC) attained with RoHCv1, RoHCv2 UDP profile and IPHC reference implementations for a VoIP call with GSM full rate codec using IPv4 over WLAN.

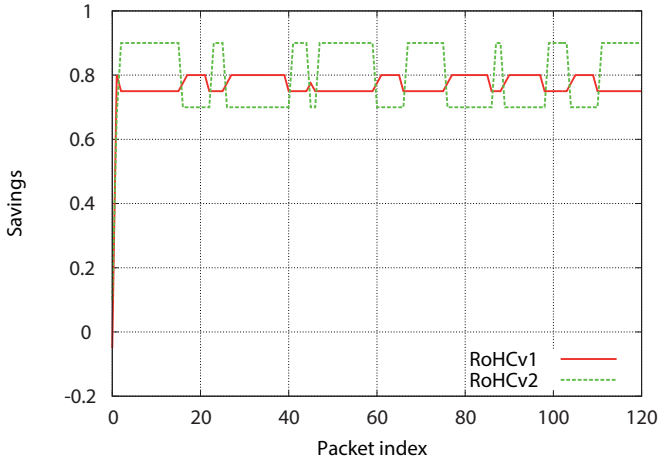

Fig. 6 Header compression savings $S_H(i)$ attained with RoHCv1 and RoHCv2 RTP profile reference implementations for an IPv4 audio only stream over a reliable unidirectional channel using the RTP profile.

We observe that the compressed header savings for RTP typically range from 70 % to 90 %, depending on the version and packet under consideration. More precisely, we note that RoHCv2 typically compresses with savings of approximately 70 % or 90 %, whereas RoHCv1 typically compresses packet headers at an approximate 77 % or 80 %. In comparison, we observe that RoHCv1 exhibits (*i*) a higher lowest level of attained savings, which is contrasted by (*ii*) a lower highest attainable savings level. Interestingly, RoHCv2 exhibits (*i*) a lower base level of savings in periods that RoHCv1 achieves high compression and (*ii*) higher compression where RoHCv1 achieves low compression. We also should note, that for the stream in question the RTP marker bit field is always set, which prevents RoHCv2 to achieve the best compression efficiency possible (a condition that is normally not present in audio transmissions).

We provide a comparison between the two RoHC versions as well as IP versions in Table 1.

Table 1 Average RoHCv1 and RoHCv2 RTP profile savings for an audio streaming scenario over a reliable channel with IPv4 and IPv6.

IP ver.	Comp. ver.	UH	Mean CH	$\overline{S_P}$	$\overline{S_H}$
v4	RoHCv1	40	5.40	16 %	86 %
	RoHCv2		3.60	17 %	91 %
v6	RoHCv1	60	4.80	24 %	92 %
	RoHCv2		3.60	24 %	94 %

It can be noted that the average compressed header (CH) attained with RoHCv1 is 1.8 bytes higher than with RoHCv2 in IPv4, and 1.2 bytes in IPv6. The average header size for RoHCv2 is independent of the IP version, whereas RoHCv1 performs better under IPv6, which can be attributed to the larger amount of savings due to static header fields (i.e., IP addresses). Comparing the complete packet savings, we note that both RoHC versions achieve a very close average performance of 16–17 % in IPv4 and about 24 % in IPv6.

We illustrate the UDP compression and total savings in Fig. 7 for the same audio–only scenario. We immediately observe that the compressed header savings for both RoHC versions are about 70 %, while IPHC shows 90 % header savings with the same periodical refreshes as before.

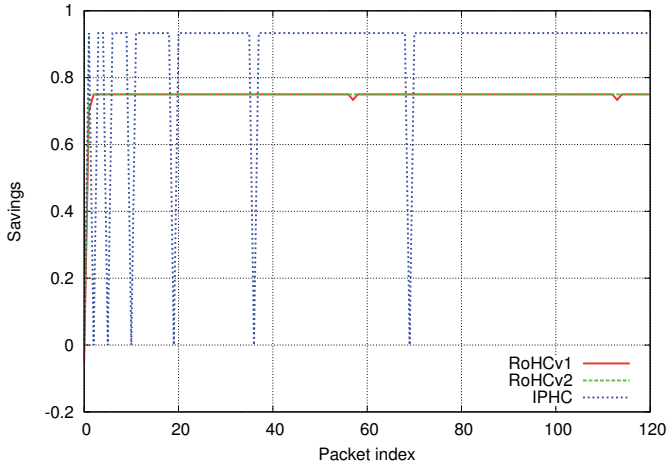

Fig. 7 Header compression savings $S_H(i)$ attained with RoHCv1, RoHCv2 UDP profile and IPHC reference implementations for an IPv4 audio only stream over a reliable unidirectional channel.

We provide a comparison between the two RoHC UDP profile versions and IPHC as well as the two IP versions in Table 2.

We note that the average compressed header sizes attained with RoHCv1 and RoHCv2 are almost exactly the same for both IP versions. However, the average header size of IPHC is about 9 bytes lower than what is observed for RoHC's compressed packets. Taking into account that IPHC relies on the link layer to signal packet types, it still outperforms RoHC by approximately 8 bytes. Comparing the packet savings in general, we note that IPHC with IPv6 is about 2 bytes better, because it can completely omit the transmission of the IP ID field.

We conclude that the average savings attainable for both RoHC RTP profile versions are within 5 % when considering the header only and less when considering the entire packet. While RoHCv2 features higher savings, but a lower base of compression, RoHCv1 features a more narrow intermediate range of savings. For UDP, both RoHC version perform at about the same level, however, IPHC outperforms them easily by at least 25 %.

Table 2 Average RoHCv1, RoHCv2 UDP profile and IPHC savings for an audio streaming scenario over a reliable channel with IPv4 and IPv6.

IP ver.	Comp. ver.	UH	Mean CH	$\overline{S_P}$	$\overline{S_H}$
v4	RoHCv1	28	15.03	13 %	46 %
	RoHCv2		15.02	13 %	46 %
	IPHC		6.29	33 %	77 %
v6	RoHCv1	48	15.05	14 %	69 %
	RoHCv2		15.03	14 %	69 %
	IPHC		4.48	51 %	91 %

6 TCP Stream Compression

This section highlights the compression performance achieved by employing the RoHC TCP profile and IPHC. We note that RoHC has only one defined TCP profile and the change from RoHCv1 to RoHCv2 does not introduce differences for this scenario.

In Fig. 8, we illustrate the compression ratios for a TCP acknowledgement stream of a digital radio station. We note that header compression generally exhibits very limited gains when the packet payloads are large, which is normally the case for TCP (segments are commonly full–framed from the link–layer's point of view). However, acknowledgement streams are ideally suitable candidates for compression, since they do not contain any significant amount of payload.

As we observe from the values presented in Fig. 8, employing both RoHC and IPHC can result in savings of around 80 % of the header data. We note, that IPHC outperforms RoHC at its peak efficiency by about 10 % on average. However, IPHC also loses about 5–10 % in its periodical fallbacks to larger packet types. The downward spikes illustrated in Fig. 8 are due to the constantly changing TCP flags and various TCP option fields. Upon closer inspection, we note that RoHC is in general more capable at handling such "erratic" changes of the TCP header fields, while IPHC always loses 5–20 % when compared to RoHC.

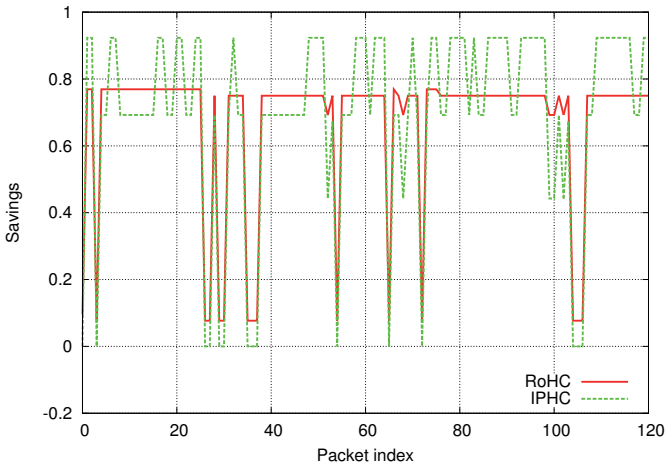

Fig. 8 Average header savings $S_H(i)$ attained with RoHC TCP profile and IPHC reference implementations for the acknowledgement stream of a digital radio transmission.

Figure 9 and 10 illustrate the TCP complexity during compression and decompression for this scenario, respectively. We initially observe that the results are in the same magnitude as the previously discussed UDP measurements. However, the overall compression is faster, while decompression is slower than observed for the UDP stream. As outlined before for the compression efficiency, the increased number of TCP header fields and their exhibited variations over packets are responsible for the increased compression time.

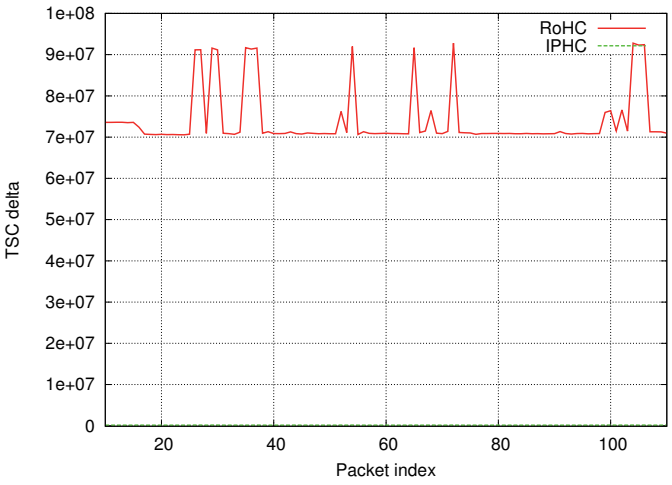

Fig. 9 Header compression complexity measured by CPU timestamps (TSC) attained with RoHC TCP profile and IPHC reference implementations for the acknowledgement stream of a digital radio transmission.

7 Lossy Channel Performance

To evaluate the performance of both RoHC versions under realistic conditions, we simulated a wireless channel using (*i*) an uncorrelated error model and (*ii*) the popular Gilbert–Elliot channel model as described in, e.g., [16]. In contrast to uncorrelated errors modelled by a coin flipping process, the model used here has correlated losses representing a wireless channel. The Gilbert–Elliot model is a two state Markov chain, where we simplify the model in such a way that (*i*) no errors occur in the good state and (*ii*) no packets are conveyed successfully in the bad state.

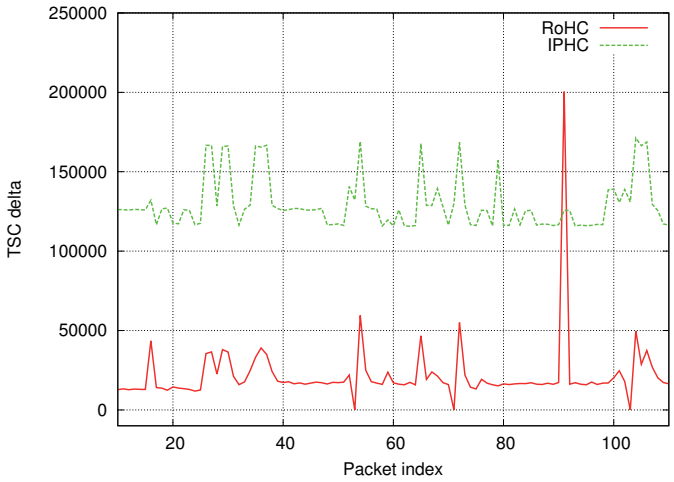

Fig. 10 Header decompression complexity measured by CPU timestamps (TSC) attained with RoHC TCP profile and IPHC reference implementations for the acknowledgement stream of a digital radio transmission.

We note here, that both error models only introduce losses on the wireless link. If compressed packets are lost, however, there are additional packet losses due to the decompression failure. Even though both RoHC versions are performing better than any other header compression schemes, e.g., those proposed by Van Jacobson [12] or Perkins and Mutka [18], there will be additional losses due to the lost decompressor state. In order to get a meaningful performance evaluation, burstiness is needed to get the decompressor out of sync. In earlier compression schemes, any error would make the decompressor loose the synchronism with the compressor (compare, e.g., results obtained with Van Jacobson or Perkins compression schemes for details), but RoHC was designed to be more robust with respect to single failures.

7.1 Compression Savings with RTP

To evaluate the performance of RoHC in a general channel configuration, we initially illustrate the average attainable savings and the 95 % confidence intervals for the two RoHC versions utilising a videoconferencing scenario over IPv4 in Fig. 11.

We immediately observe that the savings for both RoHC versions drop continuously with an increased packet loss probability. Overall, we note that the savings are about 4–5 % higher throughout when utilising RoHCv2, similar to the results obtained for the error–free channel evaluations. RoHCv1 realises more than 80 % savings for packet loss rates up to approximately 60 % . We additionally note that even for 80 % of packets lost, savings of more than 75 % are successfully realised with the RoHCv1 reference implementation (noting that RoHCv2 performs better by the above margin) over

IPv4 (with IPv6 likely yielding higher savings). Our findings are supported by the overall narrow 95 % confidence intervals illustrated in Fig. 11 as well.

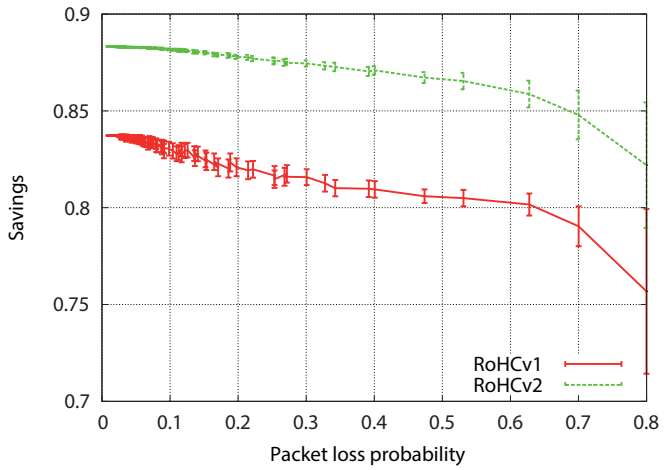

Fig. 11 Average savings $S_H(i)$ attained with RoHCv1 and RoHCv2 reference implementations for video conferencing over an IPv4 link with different uncorrelated packet loss probabilities.

We provide the aggregated performance metrics for both RoHC versions applied to both scenarios in Table 3.

Table 3 Average savings (Sav.) values for the audio only and a video conferencing scenario (with 20 % error rates in erroneous settings).

		Error-free Sav.	Uncorrel'd Sav.	Correl'd Sav.
Audio (comb.)	v1	86 %	87 %	87 %
	v2	91 %	91 %	92 %
Video	v1	71 %	71 %	71 %
	v2	76 %	75 %	75 %

Overall, we find that for uncorrelated and correlated channel losses alike, both reference implementation versions of RoHC exhibit significant (over 80 %) and stable (up to 60 % packet loss probability) savings utilising IPv4 in a video conferencing scenario, as summarized in Table 3. Due to the larger address fields in IPv6, the savings achievable with IPv6 would be even higher.

7.2 Compression Savings with UDP

Here, we consider the uncorrelated error–prone channel scenario and illustrate the obtained header compression savings for UDP in Fig. 12.

This approach is similar to the error–free measurements with RoHC and IPHC, illustrated in Fig. 2. However, the major difference between the RoHC compression and IPHC schemes cannot be seen on the figure. Since the IPHC UDP profile does not have any feedback capability, the IPHC decompressor will go out of sync with its compressor pair sometimes. This directly results in decompression failures, if the next correctly received compressed packet is in a different generation. Therefore, packet losses may affect correctly received IPHC packets later on. In our test stream we did not observe any decompression failures with either RoHC version.

We additionally note that the recommended IPHC compressor configurations optimise the periodic context refreshes very well. For loss–rates in the range of 0 % to 25 % approximately 0.001 % of the correctly received compressed packets are discarded because of invalid decompressor contexts. We furthermore note that longer streaks of decompression failures were observed as well, which result from out–of–sync decompressors.

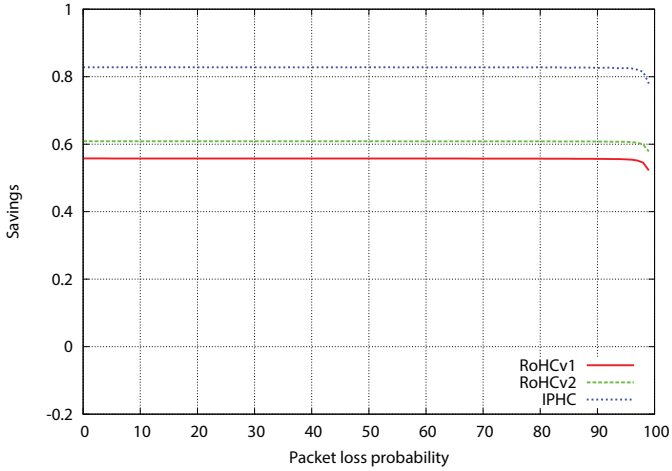

Fig. 12 Average header savings $S_H(i)$ attained with RoHC UDP profile and IPHC reference implementations for a UDP stream over an IPv4 link with different uncorrelated packet loss probabilities.

7.3 Compression Savings with TCP

We now shift to the evaluation of the compression savings for TCP. For this scenario, the corresponding feedback mechanisms are enabled for both IPHC and RoHC. The server application that is in charge of maintaining TCP creates a connection and sends fixed data to the client (downstream). The client, in this case, only acknowledges the received packets (upstream).

For these measurements, we compressed 100 packets 10 times and calculated the mean savings with 95 % confidence intervals for each measured loss–rate.

We illustrate the resulting savings for the different compression mechanisms in Fig. 13. We observe that the compression savings are inverse proportional to the loss–rate in case of the server stream. In the measurement time interval, we additionally observe that at 10 % loss–rates, the compression savings are approximately 10 % lower than in a scenario without losses. This observation is applicable for both, RoHC and IPHC. We additionally note that the IPHC packets are again about 20 % larger than the packets produced by RoHC.

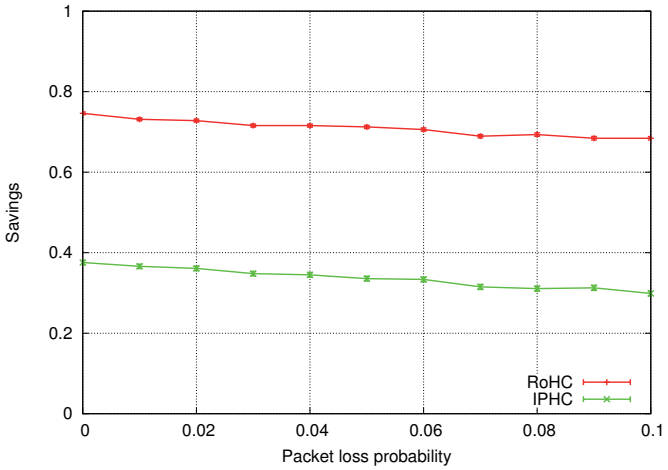

Fig. 13 Average header savings $S_H(i)$ attained with RoHC TCP profile and IPHC reference implementations for a radio stream over an IPv4 link containing the audio payload (downstream) with different uncorrelated packet loss probabilities.

In Fig. 14, we illustrate the acknowledgement stream as sent by the client (upstream) to the server. We observe that the decrease in savings are insignificant in the measured time intervals when compared to the downstream compression.

In both cases, we have very tight confidence intervals, which correspond to the observation that on average, the compressed packet sizes are very close to the mean size. Looking at the raw data, this usually refers to about 1–2 bytes difference between compressed packets.

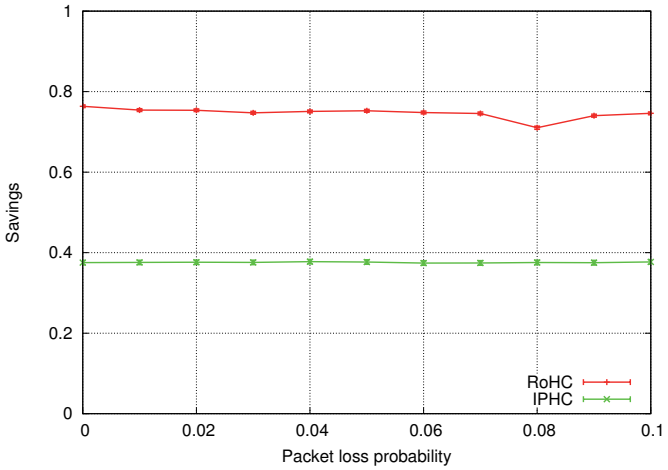

Fig. 14 Average header savings $S_H(i)$ attained with RoHC TCP profile and IPHC reference implementations for a radio stream over an IPv4 link containing the TCP acknowledgements (upstream) with different uncorrelated packet loss probabilities.

8 Analysis of Impact Fields

Since header compression is mainly used for the compression of digital audio transmissions, the designers of header compression focused predominantly on the RTP profile. Certain characteristics of IP transmissions are also ignored (fragmentation for example), making such streams uncompressable by RoHC (except with the uncompressed profile).

The fields of compressable headers are also classified into three different groups: static, dynamic and irregular. The last two are especially important, since they always change throughout the transmissions. This is assumed to be incremental and its rate is preffered to be fixed, which might not be the case for some transmissions.

Therefore, it is also interesting to look at the streams before compression with RTP, since not all packet flows are perfect for compression with RoHC and therefore they could decrease the compression gain.

The following header fields are all dynamic or irregular in nature and therefore impact the compression quality the most:

- *IP Id* (2 bytes): This field applies only to version 4 of the Internet Protocol. It is primarily used to identify fragments of a datagram. Since fragmentation is not allowed to be compressed by RoHC, this field should be optimally left 0, but that is usually not the case.
- *UDP Checksum* (2 bytes): This field is an error–checking number and is not optional when used with IPv6. It is always sent uncompressed in every packet (when it is non–0).
- *RTP Marker Bit* (1 bit): The usage of the marker bit is defined by the application layer. In a VoIP scenario, this is usually used to signal the start of a talkspurt. Consequently, it is relatively rarely set to 1. RoHC assumes, that it is 0, unless otherwise specified by the given packet.
- *RTP Sequence Number* (2 bytes): The sequence number is incremented by 1 for each RTP packet and is used for packet loss detection.
- *RTP Timestamp* (4 bytes): The timestamp is usually used for audio–video transmissions and defines the interval between two frames. For compression it is best when this interval remains constant between packets.

In this part, the impact fields are evaluated in a setup (except the UDP checksum) in which – by certain probabilities – a fluctuation is simulated in the difference between the same field's value over two consecutive packets. This effectively results in a degradation of compression quality, because the compressor is usually forced to switch from a sequential field behaviour to another transmission method.

In the measurement scenario, the fluctuation probability between packets is constantly increasing from approximately 0.0 to 1.0. We can also interpret the measurements presented here, as an extension of the correlated/uncorrelated error scenario presented in the previous section, but with the packet losses occurring before compression.

8.1 IP Id Delta

In this measurement, RoHC's tolerance to the fluctuation of the IPv4 identification field is tested. As seen in Fig. 15, the average compression ratio of the two different versions are very close to each other. RoHCv1's vary between 89 % and 90 %, while RoHCv2 is around 91 % throughout.

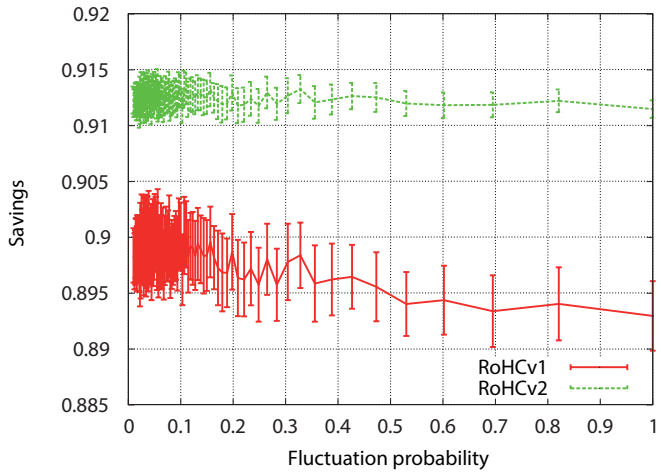

Fig. 15 Fluctuation of the IP Id field delta in relation to header savings.

It can be concluded, that the IPv4 identification field doesn't impact in any significant way the overall compression ratio, though v2 has a smaller confidence interval and a lot smoother curve.

8.2 RTP Marker Bit

In Fig. 16 we can see how the compression behaves when the marker bit fluctuates (from 1 to 0 and vice versa). Since there is no significant difference between the two versions' efficiency during the compression of the marker bit, we conclude, that the marker alone cannot degrade the compression significantly.

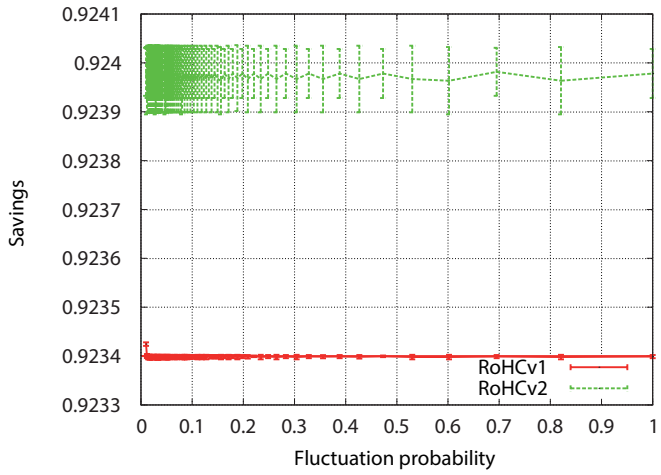

Fig. 16 Fluctuation of the RTP Marker bit field delta in relation to header savings.

However it must noted, that v2 can only transmit the marker change using the *pt_1_rnd* or, *pt_1_seq_ts*, etc. packets, which have a larger size by 1 byte than the optimal *pt_0_crc3* packet. Hence the confidence interval difference in Fig. 16.

8.3 RTP Sequence Number

We turn to the evaluation of the RTP sequence number. In Fig. 17 we can see, that there is a 3 % difference in gain between the two versions. Peculiarly, v2 has a lower compression ratio for small fluctuation probabilities, which is due to larger packets being used by version 2 when transmitting a sequence number change.

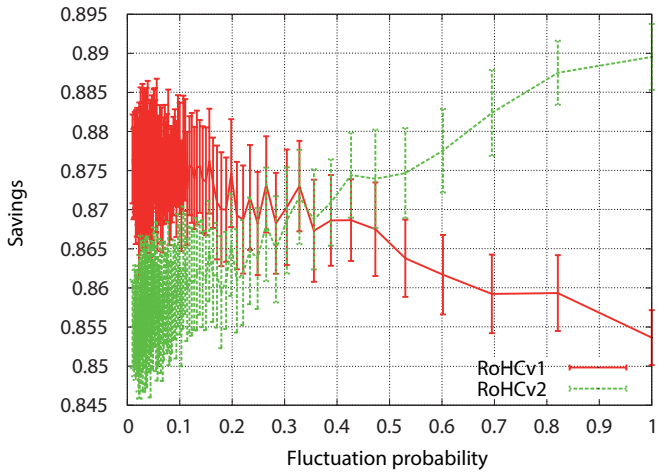

Fig. 17 Fluctuation of the RTP Sequence number field delta in relation to header savings.

The reason for this is, that in RoHCv2, the internal MSN[4] has the same value as the RTP sequence number[5]. However, the LSB (*Least Significant Bit*) compression relies on this MSN and the LSB fields need to be updated as well.

Moreover, it can also be seen in Fig. 17, that the situation reverses after 0.34 fluctuation probability and RoHCv2 reaches better results than RoHCv1. This can be interpreted as RoHCv2 being more tolerant toward frequent changes in sequence number delta than RoHCv1.

8.4 RTP Timestamp

Lastly, the RTP timestamp is evaluated in the non–timer–based compression mode.

Figure 18 shows, that both versions have mostly the same curve, while RoHCv2 produces the usual compression ratio. In this case, the fluctuation – similarly to the previous measurements – is alternating between a delta value of 1 and 2.

4 Master Sequence Number, it identifies the compressed packet, much like the RTP SN the uncompressed RTP packet.

5 This is done inorder to have 1 less field in the compressed packet.

However, in Fig. 19, the delta is increased by the fluctuation probability[6]. This curve confirms our observations, that (acticom's) RoHCv2 implementation is less tolerant to varying timestamp behaviours. Similar timestamp fluctuations are usually not good for compression to begin with, but in such a case, RoHCv1 can perform better by 5–10 %!

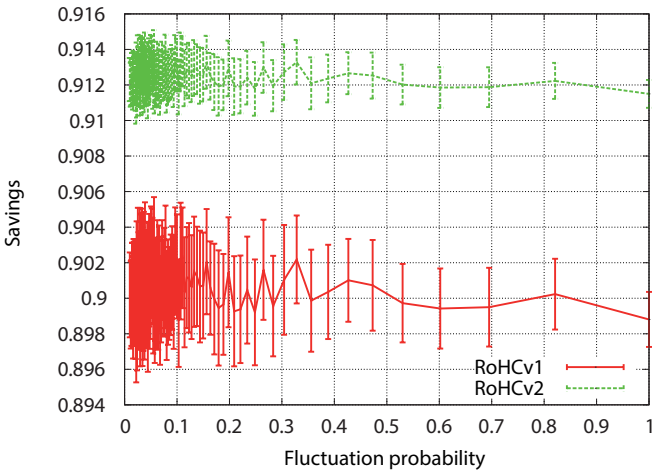

Fig. 18 Alternating RTP timestamp field delta in relation to the header savings.

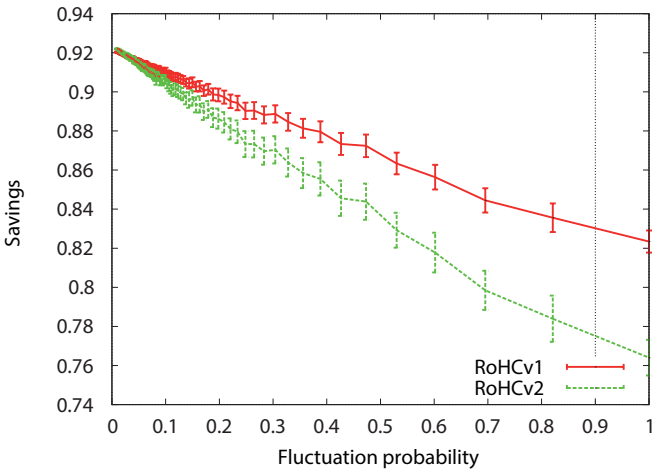

Fig. 19 Constantly increasing RTP timestamp field delta in relation to the header savings.

9 Conclusion

Currently, header compression schemes are being used by various IP–capable infrastructures, like WiMax and LTE, however, the proliferation of internet enabled devices (IoT) will benefit from reduced bandwidth usage through header compression. The adaptation of such procedures could potentially decrease the overall latency in the whole network, as well as reduce the probability of individual packet corruptions.

Throughout this paper, we reviewed several real–world performance measurements for Robust Header Compression and IP Header Compression. Additionally, we presented measurements targeting specific dynamic fields, which impact header compression for RoHC.

Overall, we observe that RoHCv1 and RoHCv2 RTP profiles achieve significant savings by compression of multimedia protocol headers, typically at least 70 % savings for IPv4. This allows network operators to increase the capacity of voice users by a factor of approximately 2–3 (assuming standard voice packet sizes) without any investment in the network infrastructure. The savings for IPv6 would be significantly higher, allowing for an additional increase by a factor of approximately 4–6. Although both versions achieve these significant savings, we find that in error–free environments with a single audio stream, ROHCv1 has a slight benefit over ROHCv2, while for multiple connections, ROHCv2 exhibits higher possible savings.

While IPHC is capable of achieving approximately 10 % to 20 % higher UDP/IP compression than either RoHC version, nonetheless it potentially aggravates errors locally due to it lacking feedback capabilities. But with the appropriate compressor configuration, this can be minimised sufficiently. Compressing streams with UDP/IP compression has the additional benefit of having a stable and persistent compressed header size which is at least the fourth of the original uncompressed size. Between the two RoHC versions, the second edition consistently shows better compression gains by 5–10 %.

RoHC and IPHC TCP compressions are quite similar to each other regarding their compression gain when we are not concerned with reordering. Albeit the IPHC benefits from a simpler compressed packet structure, which results in faster compression, it does not compress TCP fields when reordering is expected. This subsequently results in decreased header compression savings, which are roughly half of what Robust Header Compression can achieve. However, when we are not concerned with reordering, in the standard cases IPHC will perform equally or even slightly better than RoHC.

Our results also show, that while RoHCv2 uses significantly more complexity during compression than either IPHC (both for UDP and TCP) or RoHCv1 (in case of UDP), it is somewhat faster when decompressing.

The paper also presented several compressible fields, which impact the overall header compression gain. We conclude, that a fluctuation in these fields' delta generally impacts the achievable gains by 1 %, except for the RTP sequence number field, which is handled differently than any other compressed field and thereby behaves in a particular manner between the two RoHC versions (decreasing the performance for RoHCv1 drastically).

Based on these observations, research efforts are currently being made for attaining quantitative results regarding the energy consumption of header compression schemes on mobile devices. Moreover, we find, that the configuration of

6 In this case, the x-axis shows the probability of the RTP timestamp delta increasing by 1. On the left side, the delta increases rarely, but on right side it increases for almost every packet.

the compressor/decompressor instances plays a large role in the final compression quality. Our present efforts target the automation of the compression configuration, which will enable future implementations of header compressions to readily adapt to varying channel conditions on–line.

Acknowledgment

The authors thank acticom GmbH, especially Gerrit Schulte, for the support and software reference implementations of RoHCv1, RoHCv2 and IPHC, as well as their help in conducting the experiments.

References

[1] Arango, J., Pink, S., Ali, S., Hampel, D. "Header compression for ad-hoc networks." In: Military Communications Conference, 2005. MIL-COM 2005. IEEE, Atlantic City, NJ, Oct. 17-20, 2005, Vol. 5, pp. 3080-3086. DOI: 10.1109/MILCOM.2005.1606132

[2] Degermark, M., Fukushima, H., Hannu, H., Jonsson, L-E., Hakenberg, R., Koren, T., Le, K., Liu, Z., Martensson, A., Miyazaki, A., Svanbro, K., Wiebke, T., Yoshimura, T., Bormann, C., Burmeister, C., Zheng, H. "Robust header compression: ROHC: Framework and four profiles: Rtp, udp, esp, and uncompressed." Request for Comments 3095, 2001.

[3] Jung, W., Kang, C., Yoon, C., Kim, D., Cha, H. "Appscope: application energy metering framework for android smartphones using kernel activity monitoring." In: Proceedigs of 2012 USENIX Conference on Annual Technical Conference, pp. 36-36, Berkeley, CA, USA, 2012.

[4] Casner, S., Jacobson, V. "Compressing ip/udp/rtp headers for low-speed serial links." Request for Comments 2508, 1999.

[5] Casner, S., Jacobson, V. "Compressing ip/udp/rtp headers for low-speed serial links." Request for Comments 2508, 1999.

[6] Cheng, B.-N., Moore, S. "Securing robust header compression (rohc)." In: Military Communications Conference, 2013. MIL-COM 2013-2013. IEEE, San Diego, CA, Nov. 18-20, 2013, pp. 1383-1390. DOI: 10.1109/MILCOM.2013.235

[7] Cheng, B.-N., Wheeler, J., Hung, B. "Internet protocol header compression technology and its applicability on the tactical edge." *IEEE Communications Magazine*. 51(10), pp. 58-65. 2013. DOI: 10.1109/MCOM.2013.6619566

[8] Cheng, B.-N., Wheeler, J., Hung, B., Moore, S., Sukumar, P. "A comparison of ip header compression schemes in manets." In: IEEE 32nd International Performance Computing and Communications Conference (IPCCC), 2013, San Diego, CA, Dec. 6-8, 2013, pp. 1-9. DOI: 10.1109/PCCC.2013.6742791

[9] Cheng, B.-N., Zuena, J., Wheeler, J., Moore, S., Hung, B. "Manet ip header compression." In: Military Communications Conference, 2013. MIL-COM 2013-2013. IEEE, San Diego, CA, Nov. 18-20, 2013, pp. 494-503. DOI: 10.1109/MILCOM.2013.9

[10] Seeling, P., Fitzek, F. H., Hendrata, S., Reisslein, M. "Header compression schemes for wireless internet access." Electrical Engineering & Applied Signal Processing. CRC Press, ch. 10., 2004.

[11] Fitzek, F. H. P., Rein, S., Seeling, P., Reisslein, M. "Robust header compression (rohc) performance for multime-dia transmission over 3g/4g wireless networks." *Wireless Personal Communications*. 32(1), pp. 23-41. 2005. DOI: 10.1007/s11277-005-7733-2

[12] Jacobson, V. "Compressing tcp/ip headers for low-speed serial links." Request for Comments 1144, 1990.

[13] Jin, H., Hsu, R., Wang, J. "Performance comparison of header compression schemes for rtp/udp/ip packets." In: Wireless Communications and Networking Conference, 2004. WCNC. 2004 IEEE, March 21-25, 2004, Vol. 3, pp. 1691-1696. DOI: 10.1109/WCNC.2004.1311807

[14] Jivorasetkul, S., Shimamura, M., Iida, K. "Better network latency with end-to-end header compression in sdn architecture." In: IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM), 2013, Victoria, BC, Aug. 27-29, 2013, pp. 183-188. DOI: 10.1109/PACRIM.2013.6625471

[15] Fitzek, F. H., Tomoskozi, M., Seeling, P. "Performance evaluation and comparison of robust header compression (ROHC) rohcv1 and rohcv2 for multimedia delivery." In: Workshops Proceedings of the Global Communications Conference, GLOBECOM, Atlanta, GA, USA, 2013, pp. 544-549. DOI: 10.1109/GLOCOMW.2013.6825044

[16] Zorzi, M., Rao, R. R., Milstein, L. B. "On the accuracy of a first-order markovian model for data block transmission on fading channels." In: Fourth IEEE International Conference on Universal Personal Communications. 1995. Record., 1995, Tokyo, Nov 6-10, 1995, pp. 211-215. DOI: 10.1109/ICUPC.1995.496890

[17] Pelletier, G., Sandlund, K. "Robust header compression version 2 (rohcv2): Profiles for rtp, udp, ip, esp and udp-lite." Request for Comments 5225, 1997.

[18] Perkins, S. J., Mutka, M. W. "Dependency removal for transport protocol header compression over noisy channels." In: Communications, 1997. ICC, 97 Montreal, Towards the Knowledge Millennium. 1997 IEEE International Conference on, Montreal, Que., June 8-12, 1997, Vol. 2, pp. 1025-1029. DOI: 10.1109/ICC.1997.610035

[19] Rein, S., Reisslein, M., Fitzek, F. H. P. "Voice quality evaluation for wireless transmission with rohc." In: International Conference on Internet and Multimedia Systems and Applications, 2003.

[20] Ribicre, M., Charlton, P. Cisco visual networking index: Global mobile data traffic forecast update." Cisco, Inc., 2014-2019. [Online]. Available from: http://www.cisco.com/c/en/us/solutions/service-provider/visual-networking-index-vni/index.html [Accessed: February 2015]

[21] Seeling, P., Reisslein, M., Fitzek, F. H. P., Hendrata, S. "Video quality evaluation for wireless transmission with robust header compression." In: Information, Communications and Signal Processing, 2003 and Fourth Pacific Rim Conference on Multimedia. Proceedings of the 2003 Joint Conference of the Fourth International Conference on, Dec. 15-18, 2003, Vol. 3, pp. 1346-1350. DOI: 10.1109/ICICS.2003.1292684