Network Working Group
INTERNET-DRAFT
Expires: December 1999

Mikael Degermark, Lulea University
Hans Hannu, Ericsson
Lars-Erik Jonsson, Ericsson
Krister Svanbro, Ericsson
Sweden

June 18, 1999

CRTP over cellular radio links
<draft-degermark-crtp-cellular-00.ps>

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

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

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

Abstract

This document evaluates the performance of a header compression protocol for RTP, CRTP [RFC-2509], over links built on cellular radio access technology. The key characteristics affecting CRTP performance over such links are the high error rates and the relatively long roundtrip time over the link.

Bandwidth is typically expensive in cellular radio access networks, saving a single octet per voice packet can be equivalent to saving

many billion dollars in deployment since fewer base-stations are needed. This is beneficial for operators as well as end-users who can get cheaper wireless IP telephony service.

CRTP performance is evaluated for two kinds of link layers operating over a realistic radio channel with high bit-error rates. Two main conclusions are drawn. The first is that CRTP does not perform well for this type of link. The second is that in high-error environments it is very beneficial to have a checksum covering the compressed header only, not the payload, so that the decompressor sees all non-damaged headers. When a strong checksum covers the entire link layer frame, header compression performs badly since too many headers are discarded due to damaged payloads.

TABLE OF CONTENTS

1. Introduction.....3

2. Header compression.....3

3. Link layers.....5
   3.1. PPP in HDLC-like framing.....5
   3.2. Link layer with partial checksum.....6

4. Description of simulations.....6
   4.1. Simulated scenario.....6
   4.2. The cellular link and the back channel.....7

5. Frame error rates (FER).....8

6. Evaluation of CRTP for cellular radio links.....8
   6.1. An ideal header compression scheme.....9
   6.2. CRTP without Twice.....10
   6.3. CRTP with Twice.....12
   6.4. Loss patterns.....13
   6.5. Using only COMPRESSED_NON_TCP packets.....16

7. Conclusions.....17
   7.1. How to improve CRTP performance.....18

8. Authors addresses.....19

9. References.....19

1. Introduction

With IP telephony gaining momentum and cellular telephony having hundreds of millions of users, it seems inevitable that some future wireless telephony systems will be based on IP technology. What we today know as cellular phones may in addition to telephony and video have IP stacks, web browsers, email clients, networked games, etc. If based on IP, the telephony service will be much more flexible than today. This document concentrates on the problem of providing a good IP solution for speech, but it is clear that applications for video, games, etc, will also have to be supported.

It is vital for cellular phone systems to use the radio resources efficiently in order to support a sufficient number of users per cell. Only then can deployment costs be kept low enough. It will also be important to provide sufficiently high quality voice and video. In particular the voice service should be as good as what users expect from the cellular phone systems of today. A lower quality may only be accepted if costs are significantly lower than today.

The radio channels used in cellular systems have very high bit-error rates (BER) due to shadow fading, multipath fading, and continuous mobility. The radio signals of one user will interfere with the radio signals of other users, so with the desired number of users per cell, BERs will be high. Even after error correcting channel coding, the remaining BER can be as high as 1e-3 (one in 1000) or even 1e-2 (one in 100) in bad environments.

The only cost efficient way to achieve sufficient voice quality over such channels is to use clever speech coders and decoders that can tolerate some damage to the encoded sound data. It is not feasible to use a link layer that delivers all data reliably through an ARQ scheme with link-local retransmission. High delays would be the result. If the long maximum delays caused by an ARQ scheme were acceptable, it would be better to spread the signal over time in order to reduce the BER, rather than using an ARQ protocol. Neither is it feasible to have the link layer discard all damaged frames. The large fraction of discarded frames would result in insufficient speech quality.

Unless explicitly stated otherwise, the numbers and figures presented in this document are for IPv4, not IPv6.

2. Header compression

Speech data for IP telephony will most likely be carried by RTP [RFC1889]. A packet will then, in addition to link-layer framing, have an IP header (20 octets), a UDP header (8 octets), and an RTP

header (12 octets) for a total of 40 octets. With IPv6, the IP header is 40 octets for a total of 60 octets. The size of the payload depends on the speech encoding used and the packet rate; it can be as low as 15-20 octets.

From these numbers it is obvious that the header size must be reduced for efficiency reasons. A proposed standard for compressing RTP/UDP/IP headers over low-speed serial links, CRTP, has recently been approved by the IESG [RFC-2508, RFC-2507], together with a way to negotiate parameters for header compression over PPP [RFC-2509]. With CRTP, compressed headers are as small as 2 octets if the UDP checksum is disabled.

CRTP uses delta encoding where compressed headers carry differences from the previous header. The decompressor maintains state, known as the context, that represents what the header looks like, how it is expected to change, etc. The differences carried in each compressed packet updates the context, and thus loss of a packet will bring the context of the decompressor out of sync with the compressor as it is not updated correctly.

CRTPs mechanism for bringing the decompressor context in sync with the compressor relies on messages from the decompressor reporting its state to the compressor. Such CONTEXT_STATE messages cause the compressor to send packets with more information in their headers to update the context of the decompressor: either FULL_HEADER packets with 40 octet headers (60 for IPv6), or COMPRESSED_NON_TCP packets with compressed UDP/IP headers but a complete RTP header. Headers in COMPRESSED_NON_TCP packets are 17 octets if the UDP checksum is disabled, and 19 octets otherwise (15 and 17 octets for IPv6, respectively).

CRTP uses a link sequence number, incremented by one for each packet with a compressed header, to detect lost packets. The link sequence number ranges between 0 and 15. Gaps in the sequence number space triggers the context repair mechanism outlined in the previous paragraph.

High BERs will cause the repair mechanism to be triggered often, causing many FULL_HEADER packets or COMPRESSED_NON_TCP packets to be sent, which consume extra bandwidth. With a long roundtrip time over the link, each damaged packet can cause several subsequent packets to be discarded due to mismatching contexts.

The "Twice" mechanism proposed for compressed TCP headers in [RFC-2507] and also for CRTP in [RFC-2508] can often repair the context and avoid some of the loss caused by mismatching contexts. The assumption behind the "Twice" mechanism is that the delta of a lost

CRTP packet is often the same as the delta of the subsequent packet. An attempt to repair the context by applying the delta twice will therefore often succeed. Successful repairs are detected by a matching transport-layer checksum.

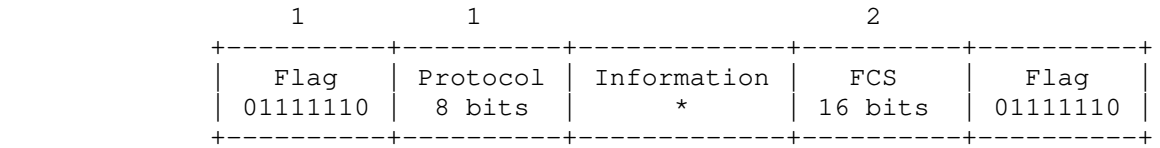
3. Link layers

When evaluating CRTP, the link layer must be considered. We will use two different link layers. One is PPP in HDLC-like framing [RFC-1662], which has a 16/32-bit CRC covering the entire frame. This implies that all damaged frames will be discarded at the link layer since the checksum will fail. It is possible to change the networking code to have such frames delivered, but then it is pointless to have the checksum in the first place and a framing scheme without a checksum would be a better solution.

For header compression purposes it is important that headers are not damaged over the link. As outlined in the introduction, however, damage to the payload is often acceptable to the (speech) decoder of the application. It would therefore make sense to have a checksum which only covers the header part of a packet. That should increase the number of headers seen by the decompressor and improve header compression performance. The second link layer we use for evaluation purposes is an imaginary such link layer, henceforth called the Link-Layer with Partial Checksum (LLPC).

3.1 PPP in HDLC-like framing (HDLC)

PPP typically uses HDLC-like framing [RFC-1662]. With a 16-bit checksum and compressed Address and Control fields, frames carrying CRTP, COMPRESSED_NON_TCP, or FULL_HEADER packets have the following format.

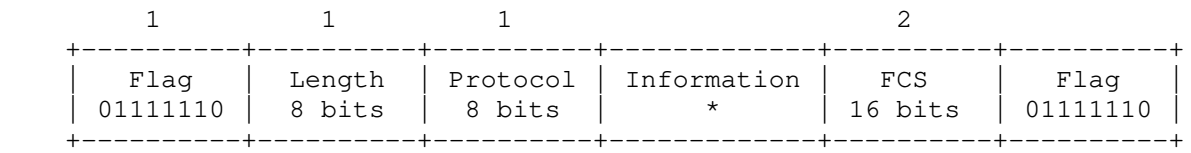


The Flag only occurs once between frames if they are sent back-to-back, so the amortized framing overhead is 4 octets per frame. The checksum (FCS) is calculated over the Protocol field and the Information field (payload), but not the Flags or the checksum itself.

Any errors anywhere in the frame will cause the FCS to fail. The frame will then be discarded.

3.2 Link-layer with partial checksum (LLPC)

This is an imaginary framing scheme derived from the HDLC-format in 3.1 by adding a one-octet Length field.



The Length field indicates how many octets of the payload that are covered by the FCS. It can have values from 0 to 255. The FCS covers the Length and Protocol field plus as many octets in the beginning of the Information field as indicated by the Length field. The value of the Length field must not make the FCS extend over the FCS field. When sending a FULL_HEADER packet, the Length field would have the value 40, since it should protect the IP, UDP, and RTP headers. When sending a minimal COMPRESSED_RTP packet, the Length field would have the value 2. The amortized framing overhead for LPC is 5 octets per frame.

Any errors in the Flag, Length, Protocol, FCS, or the initial Length octets of the Information field will cause the FCS to fail. The frame will then be discarded. Errors in the Information field after the first Length octets will not affect the FCS and will not cause the frame to be discarded.

4. Description of simulations

Section 4.1 describes the simulated scenario and 4.2 elaborates on the properties of the cellular link and the back channel.

4.1. Simulated scenario

A source generates RTP packets containing speech data and sends them across the Internet to an end-system. The end-system is connected to the Internet over a cellular link over which the RTP stream is compressed using CRTP.

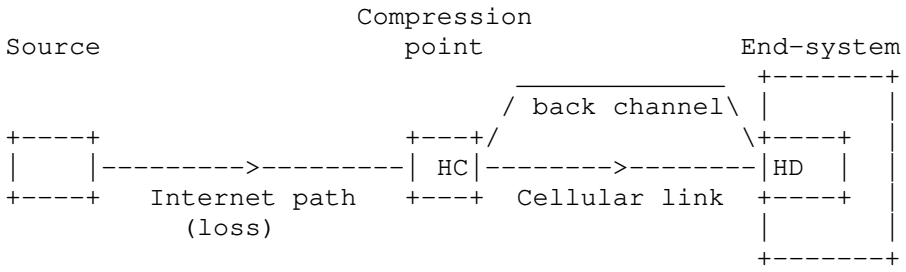


Figure 0: Simulated scenario

Over the Internet path there are uniformly distributed losses which influence the efficiency of CRTP mechanisms, and especially the "Twice" mechanism.

Over the Cellular link one of the framing protocols of section 3 carry the packets. The radio channel of the cellular link is simulated accurately for various BERs and represents fairly bad, but realistic, conditions. The roundtrip time can be varied.

The compressor (HC) at the compression point compresses RTP/UDP/IP headers according to CRTP, and sends them over the cellular link to the decompressor (HD). When HD detects that the context is out of sync, it will send CONTEXT_STATE messages back to HC over the back channel.

The speech source generates packets with payloads of a fixed size, 16 octets (representing the smallest reasonable payload size), at a rate of 50 packets per second (20 ms worth of sound data per packet). Silence suppression is used. The lengths of talk spurts and the silent intervals between them are both exponentially distributed with an expected length of 1 second. Loss over the Internet path due to congestion is uniformly distributed. This loss pattern is reasonably accurate since packet intervals are relatively long compared to congestion related loss events.

4.2. The cellular link and the back channel

The cellular link is simulated accurately using a realistic radio channel model and adding channel coding. The reported bit error rates, BER, are always the BERs after channel coding, i.e., the BER seen by the link layer.

The interesting BERs for cellular systems are in the range between 1e-3 (1/1000) and 1e-6 (1/1000000). Circuit-switched cellular voice

transmission can deliver acceptable speech quality down to around 1e-2, while the systems become expensive at BERs much less than 1e-6.

The compressor repairs the decompressor context after feedback in the form of a CONTEXT_STATE message from the decompressor. This means that the roundtrip time over the link determines the speed of the repair mechanism. The back channel used in our simulations never damages CONTEXT_STATE messages.

5. Frame error rates (FER)

Frames can have errors due to damage over the link. This kind of damage can be further classified into

a) header damage: damage to parts of the frame that are important for header compression purposes. This is the framing plus the compressed or full header.

b) payload damage: damage to other parts of the frame. Such damage may or may not cause the frame to be unusable by the speech decoder, depending on the coding and the location of the damage. Also, it may or may not cause the entire frame to be discarded depending on the framing format.

Frames can also be damaged because the decompressor fails to reconstruct a correct header. That can of course be caused by a), but also by

c) context damage: the context of the decompressor being out of sync with the context of the compressor. This is caused by delta information being lost due to a) or b).

For HDLC, both header damage and payload damage will cause the frame to be discarded, which will increase the rate of frames discarded due to context damage.

For LLPC, payload damage will not cause the frame to be dropped before reaching the decompressor, which will reduce the number of frames discarded due to context damage. Whether or not payload damage causes the frames to be unusable for generating speech is not related to header compression performance. We expect, however, that most speech decoders will be able to utilize information in frames with payload damage.

6. Evaluation of CRTP for cellular radio links

6.1. An ideal header compression scheme

In order to have a reference point, we first simulated an ideal header compression scheme. The ideal header compression scheme can always compress the header down to a total of 2 bytes and will never fail at decompression, i.e., no frames will ever be discarded due to context damage. Such a scheme is probably not achievable, but it gives us something to compare the real CRTP against.

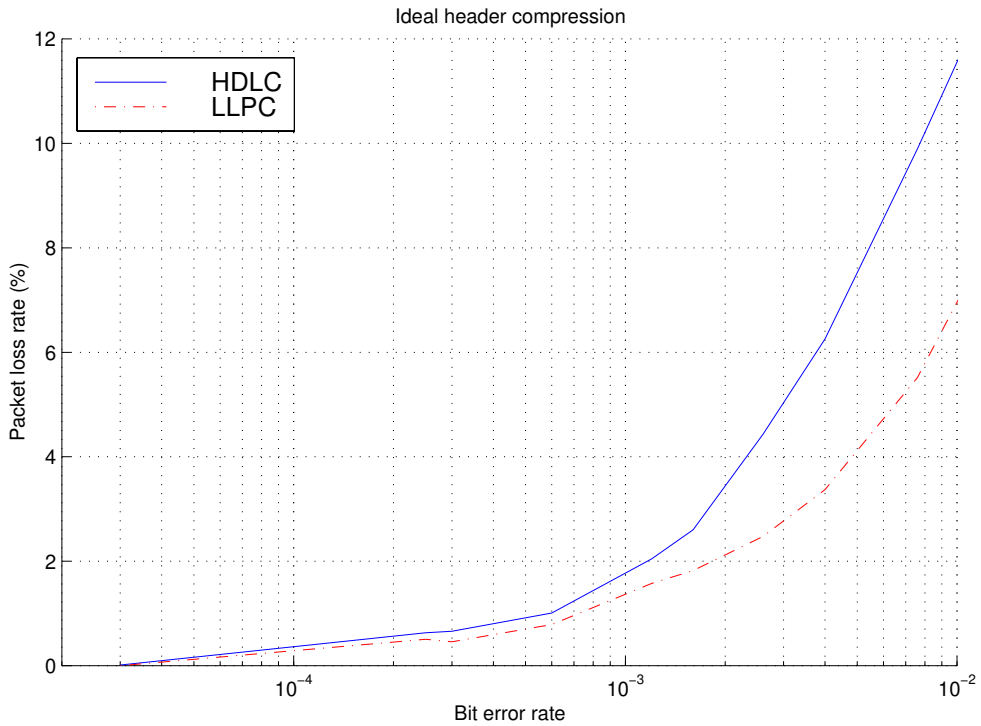


Figure 1: FER for Ideal scheme for HDLC and LLPC

As can be seen in figure 1, for a BER of 1e-3 the FER is 1-2 % for both link layers. LLPC is marginally better. At 5e-3 there is a significant difference between HDLC (7.5% FER) and LLPC (4% FER). Loss over the Internet path does not affect the ideal header

compression scheme at all, and is not included in the reported FER.

There is no context damage for the ideal scheme. The difference between the HDLC and LLPC curves show how many packets with payload damage only there are: around 0.3% for a BER of 1e-3.

With some handwaving and contemplation of packet sizes and checksum coverage, one can argue that LLPC should give a FER which is roughly 7/23 (30%) of the FER for HDLC if errors were uniformly distributed. They are not, however, and it seems that LLPC in fact gives FERs that are 55-60% of the FERs for HDLC.

6.2. CRTP without Twice

With a roundtrip time over the link corresponding to around 120 ms (a realistic value), the slowness of the context repair mechanism will multiply link layer related loss by a large factor. Figure 2 shows CRTP performance for HDLC, while Figure 3 shows CRTP performance for LLPC. The ideal curves have been included for reference. The percent numbers indicate how much loss there were over the Internet path. The plots for CRTP with Twice are discussed in the next subsection.

In figures 2 and 3 one can see that for a BER of 1e-3, CRTP gives a FER of 8% with HDLC while with LLPC the FER is 5%. Given the performance of the ideal scheme, it is clear that most of this loss is due to context damage.

The average header size will increase with increasing loss over the Internet path, since the delta between consecutive packets will then often be different and more data need to be sent to represent the new delta. A single loss over the Internet path will typically cause the following two compressed headers to have three and two extra octets, respectively. When one out of 10 packets are lost over the Internet path, that would add 5 octets to the remaining 9 headers. The average header size then increases with 5/9 octets (0.56 octets).

Figure 4 shows the average header size plotted against BERs, for varying loss over the Internet path. At low BERs, HDLC and LLPC both give an average header size of just over 2 octets when there is no loss over the Internet path. When there is 10% loss over the Internet path, both give an average header size just over 2.5 octets. This is consistent with the expected increases in header sizes due to different deltas after losses over the Internet path.

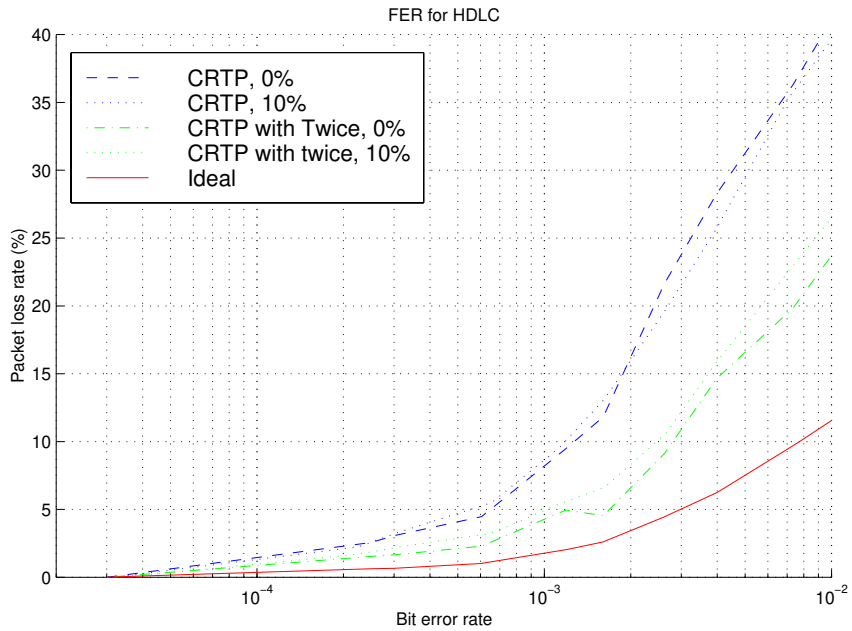


Figure 2: FER for CRTP, CRTP with Twice, and Ideal for HDLC

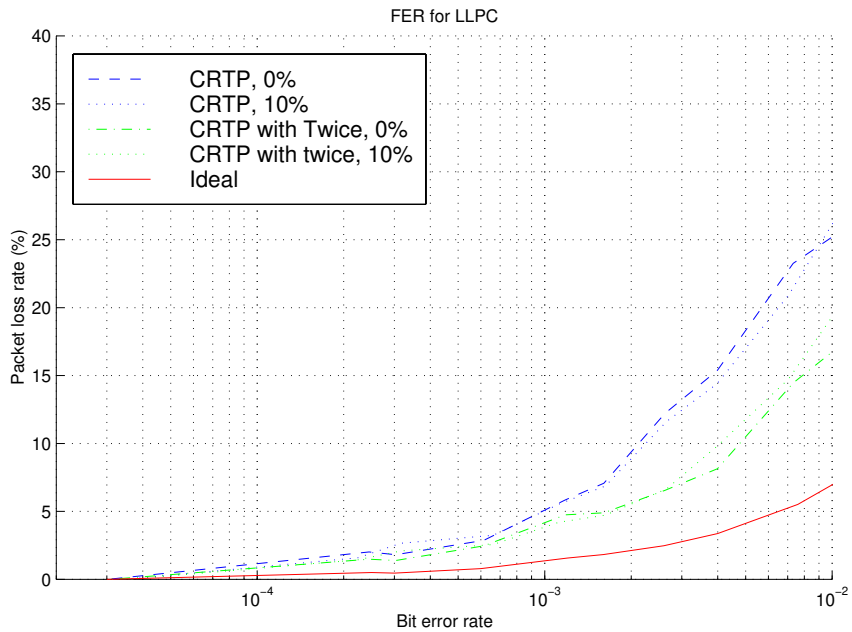


Figure 3: FER for CRTP, CRTP with Twice, and Ideal for LLPC

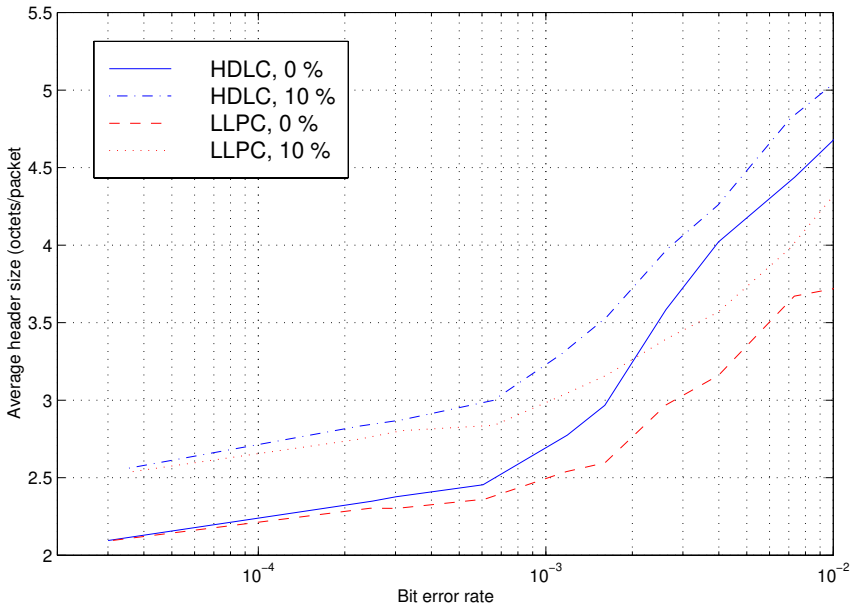


Figure 4: Average header size for CRTP

At higher BERs the average header size is determined by the rate of COMPRESSED_NON_TCP headers (17 octets) sent over the cellular link. CRTP compressors update the context state by sending such headers whenever frames have been discarded over the cellular link. The differences between the HDLC and LLPC curves at high BERs is due to their different FERs. For a BER of 1e-3 and no Internet loss, CRTP with HDLC gives an average header size of 2.7 octets, while CRTP with LLPC gives 2.5 octets. For 10% Internet loss, HDLC gives 3.2 octets and LLPC 3.0 octets.

6.3. CRTP with Twice

The Twice algorithm is a way to repair the context quickly without having to wait for a roundtrip over the link. Twice makes assumptions of what the lost delta was and tries to repair the context according to those assumptions. When using Twice there must be a way to check whether the repair succeeded, typically the UDP checksum is used for that purpose.

The plots in figure 2 show FERs for CRTP, CRTP with Twice, and the Ideal scheme, when HDLC framing is used. The CRTP with Twice curves

really show how successful Twice would be in repairing the context, we have not actually enabled the UDP checksum in our simulations, but instead we determined whether Twice would have succeeded. We wanted to try out LLPC too (figure 3), and as the UDP checksum covers the entire payload and is fairly weak, that scenario wouldn't make much sense using the UDP checksum. Instead we chose to investigate how successful Twice would be if there were some other means to detect a successful repair.

It is evident from figure 2 that Twice improves the FER significantly, although CRTP with Twice is still much worse than the Ideal scheme. At a BER of 1e-3, the FERs are less than 2% for Ideal, about 4% for CRTP with Twice, and 8% for CRTP. More sophisticated implementations of Twice might get closer to the Ideal curve.

Figure 3 shows FERs for CRTP, CRTP with Twice, and the Ideal scheme, when LLPC framing is used. The FERs are lower than for HDLC because fewer frames are discarded at the link layer, but the plots are otherwise similar.

6.4. Loss patterns

For applications such as interactive voice it is not only the loss *rate* that is interesting. Typical voice decoders will reuse earlier frames when a frame is lost, but might decrease the intensity with which that frame is played out. For each successive loss the intensity is decreased such that after a few consecutive lost frames the sound will fade out completely. When only single frames are lost, the tolerable FER might be high. A single burst of lost frames, on the other hand, can cause a very noticeable pause. Figure 5 shows a histogram over the number of consecutive loss bursts of certain lengths for CRTP, with and without Twice, for three different BERs.

It is evident from figures 5a and 5b that the majority of loss events without Twice are such that around 7 consecutive frames are lost. The link roundtrip time in these simulations was 120 ms and the packet rate 50 packets per second, which means that a single discarded frame would cause 6 additional frames to be lost due to context damage. When there is little loss over the Internet path, Twice (or variants) are very efficient since deltas rarely change.

At higher BERs, COMPRESSED_NON_TCP packets are sent often and thus lengths of frame loss bursts are less regular. Updates may be damaged, and an earlier repair may cause an update which repairs new damage.

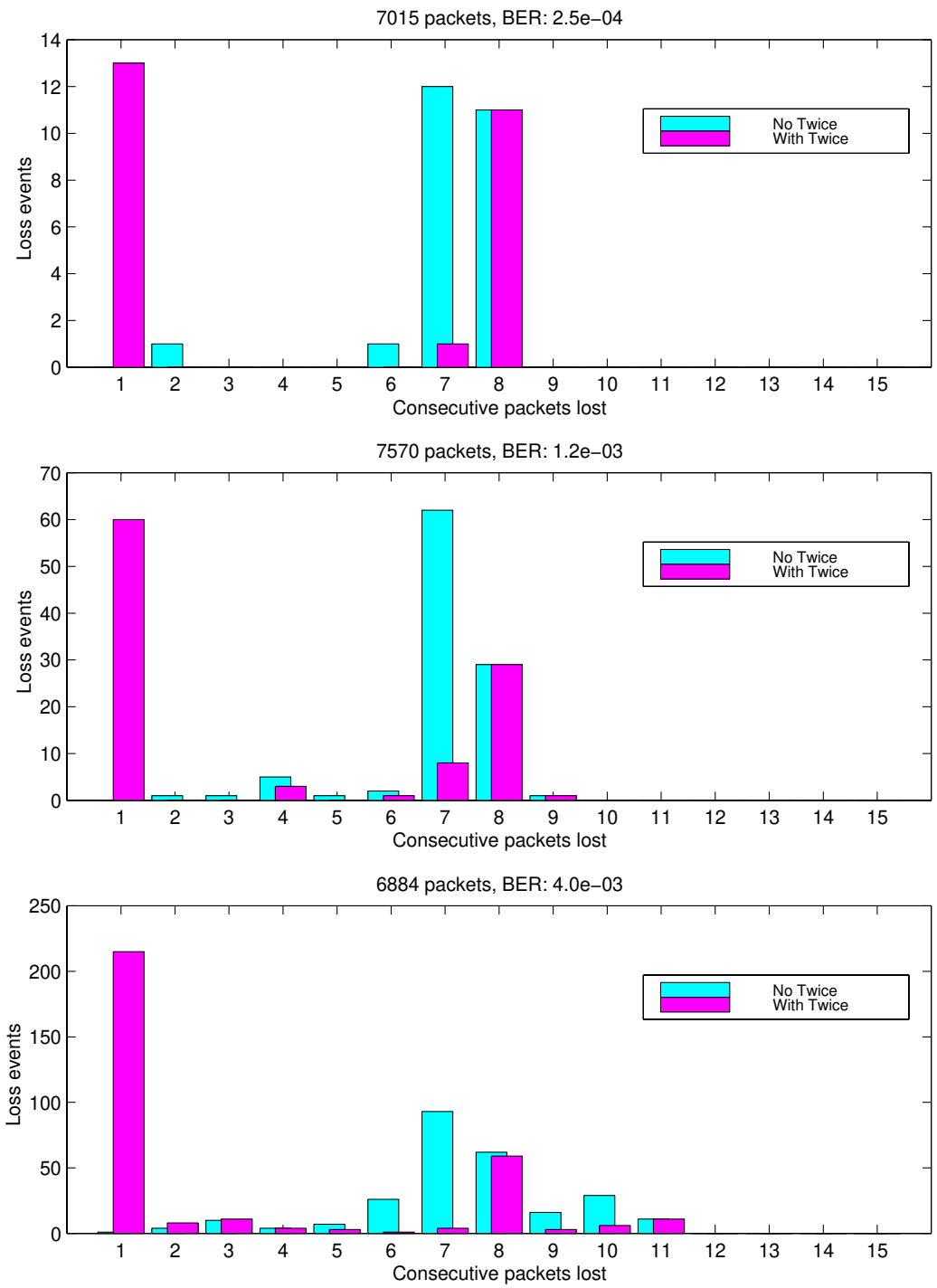


Figure 5a: Lengths of frame loss bursts, HDLC, no IP loss

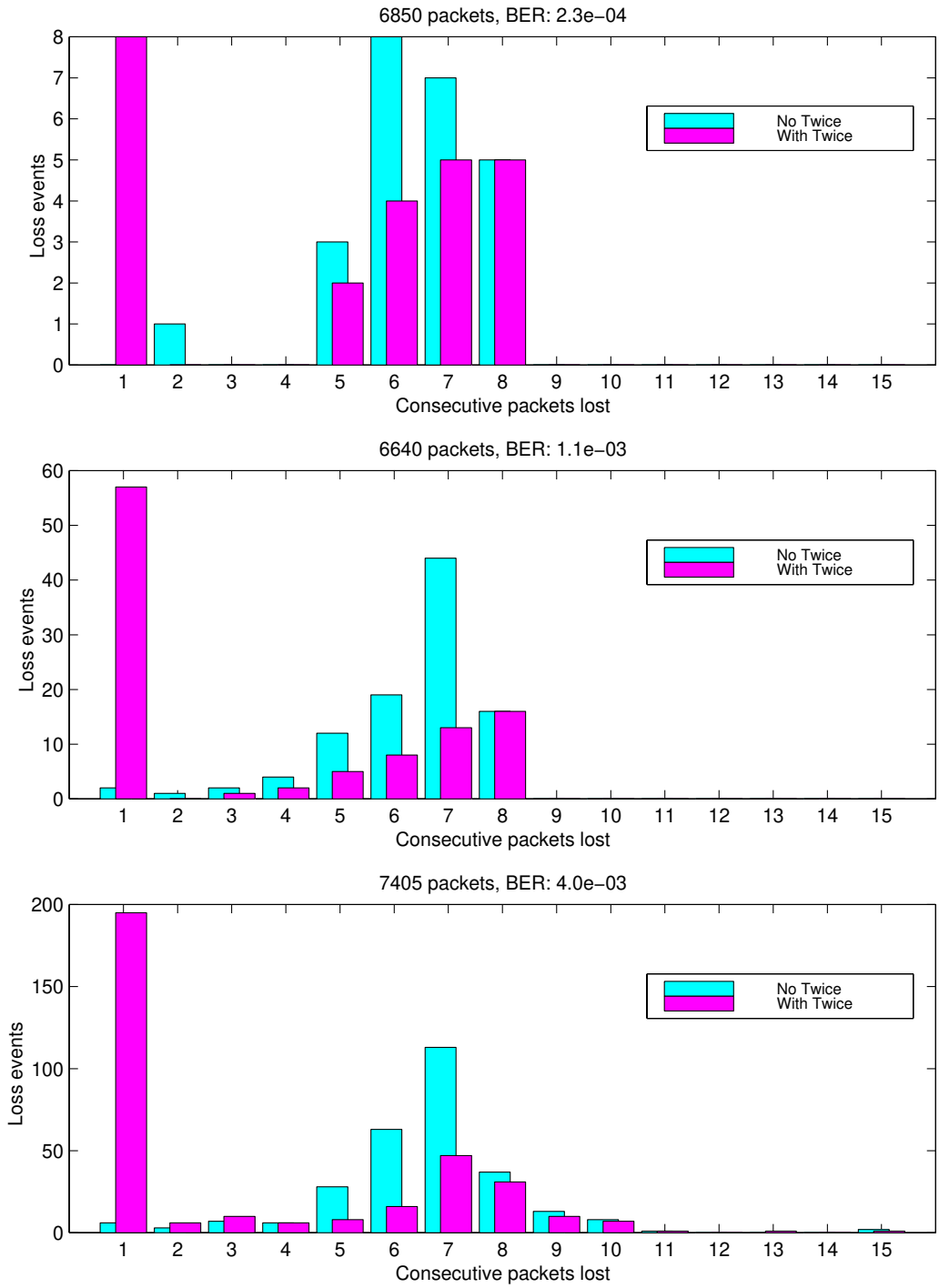


Figure 5b: Lengths of frame loss bursts, HDLC, 10% IP loss

Loss bursts involving 7-8 frames are clearly noticeable for most voice decoders. This is a major disadvantage of using CRTP over high-loss links with nontrivial link roundtrip times. Even if the frame rate was one per 30 ms and the link roundtrip time was only 60 ms the typical loss burst would be 3-4 frames (one discarded at link level, next discovers damage, update requested, update sent), which would decrease the voice quality significantly.

6.5. Using only COMPRESSED_NON_TCP packets

The high FERs for CRTP makes it interesting to compare its performance against sending COMPRESSED_NON_TCP packets only. Their headers are 17 octets. No frames are discarded due to context damage, but on the other hand it is more likely that a packet will be damaged because it is larger.

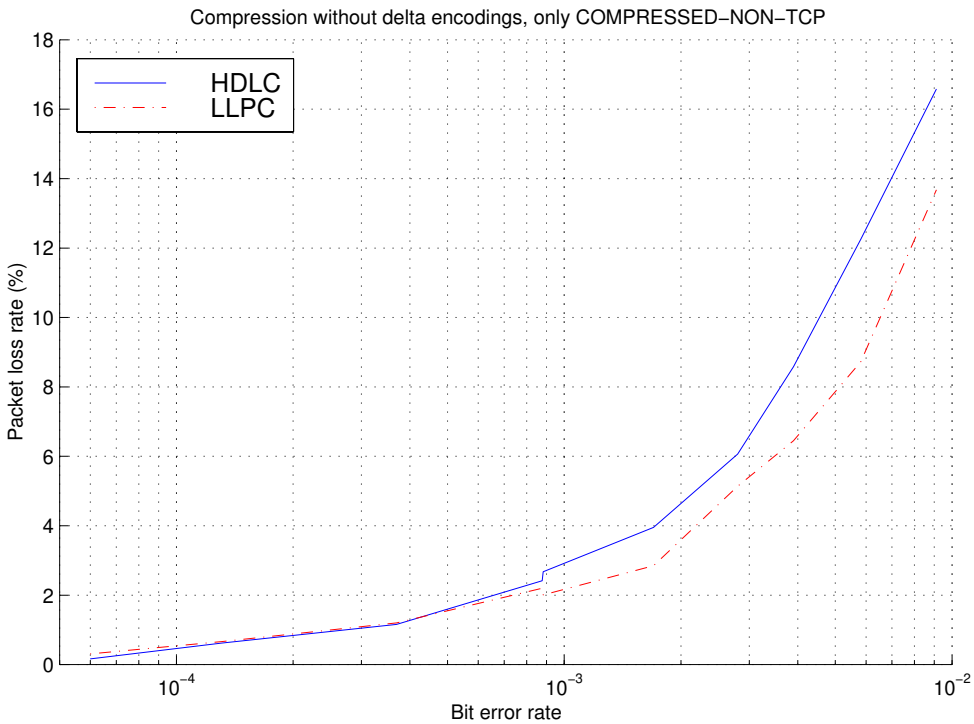


Figure 6: FER for COMPRESSED_NON_TCP only, HDLC and LLPC

Figure 6 shows the FER when sending COMPRESSED_NON_TCP packets only, for HDLC and LLPC. For HDLC, the FER when the BER is 1e-3 is 3%, which is more than for the Ideal scheme (<2%) but less than CRTP with Twice (5%). The FER for LLPC is just over 2% and similarly to HDLC, it is more than the Ideal scheme but less than CRTP with Twice.

7. Conclusions

The packet loss rate of CRTP, CRTP with Twice, and the Ideal header compression scheme is summarized in table 1 for various error rates. The numbers are for HDLC-like framing, i.e., errors in any part of a packet means that it is discarded. The payload is 16 octets. The Ideal scheme discards packets only when the packet itself is damaged.

Bit-error rate	1e-5	1e-4	1e-3	1e-2
Ideal	0	0.4%	1.8%	11%
CRTP+Twice	0	1.0%	5.0%	24%
CRTP	0	1.5%	8.0%	40%

Table 1: Frame loss rates of header compression schemes (HDLC).

It is evident from table 1 that CRTP performs well for BERs less than 1e-5, but not so well for BERs higher than 1e-4. If one considers a scenario where the path of an IP telephony conversation has a cellular link at both ends, the packet loss rates of CRTP and CRTP+Twice become intolerable at high BERs.

The major cause of CRTPs bad performance is that many packets are discarded due to context damage while waiting a link roundtrip time for the repair mechanism.

Twice is a way to repair the context locally. It requires two extra octets of header (the UDP checksum) to verify its repair attempts. These two extra octets make it a less attractive solution. Moreover, the straightforward Twice used in this evaluation does not have a sufficiently high success rate. Combinations of link-loss at a first cellular link and congestion-related loss in the rest of the path will ensure that the compressor at the last cellular link will see many holes in the packet stream. Twice will then fail often. Moreover, the UDP checksum is too weak to reliably determine the success or failure of attempted repairs.

The losses induced by CRTP and its variations are problematic not only because they are high. The loss patterns are such that losses

occur in groups longer than a link roundtrip time. This is problematic for low-bandwidth voice codecs, who cannot mask such long loss events well. Hence, the speech quality will suffer. It is a major disadvantage of CRTP that it causes such long loss events.

It is worth noticing that link layers which protect the header with strong checksums, but not the payload, will decrease the packet-loss rate significantly. Such link-layers will deliver more headers to the decompressor and context damage will be less frequent. Table 2 summarizes the results for such a link-layer.

Bit-error rate	1e-5	1e-4	1e-3	1e-2
Ideal	0	0.3%	1.4%	7%
CRTP+Twice	0	0.8%	4.2%	18%
CRTP	0	1.1%	5.0%	25%

Table 2: Frame loss rates of header compression schemes (LLPC).

Overall, the improvement in FER were around 40% with a payload of 16 octets. When the payload is larger, the improvement will be higher. In addition to the benefits for header compression, speech codecs for lossy links can utilize information in damaged payloads and will deliver higher quality speech when they have access to damaged frames.

To summarize, CRTP does not perform well over lossy links with long roundtrip times. Twice can improve the situation somewhat but the loss is still too high. A disadvantage of using Twice is that it requires that the UDP checksum is enabled, which will double the size of the compressed header. CRTP with Twice performs much worse than the Ideal scheme in terms of packet loss. Because the UDP checksum is fairly weak, Twice should not be extended to attempt a large number of repairs. Because of this, CRTP with Twice cannot approach the performance of the Ideal scheme.

7.1. How to improve CRTP performance.

The link roundtrip time should be kept low. When it is high, local repairs of the contex (without going over the link) is essential. Sophisticated versions of Twice should be considered, which implies that the UDP checksum must be enabled. Unfortunately, that adds 2 octets to the compressed header.

8. Author's Addresses

Mikael Degermark	Tel: +46 920 911 88
Dept of CS & EE, Lulea	Mobile: +46 70 833 89 33
University of Technology	EMail: micke@sm.luth.se
Hans Hannu	Tel: +46 920 20 21 84
Ericsson Erisoft AB	Mobile: +46 70 378 04 73
Lulea, Sweden	EMail: hans.hannu@lu.erisoft.se
Lars-Erik Jonsson	Tel: +46 920 20 21 07
Ericsson Erisoft AB	Mobile: +46 70 365 20 58
Lulea, Sweden	EMail: lars-erik.jonsson@ericsson.com
Krister Svanbro	Tel: +46 920 20 20 77
Ericsson Erisoft AB	Mobile: +46 70 531 25 08
Lulea, Sweden	EMail: krister.svanbro@lu.erisoft.se

9. References

[RFC-768] J. Postel, User Datagram Protocol, RFC 761, August 1980.

[RFC-791] J. Postel, Internet Protocol, RFC 791, September 1981.

[RFC-793] J. Postel, Transmission Control Protocol, RFC 793, September 1981.

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

[RFC-1662] W. Simpson, PPP in HDLC-like framing, RFC 1662, 1994.

[IPv6] S. Deering, R. Hinden, Internet Protocol, Version 6 (IPv6) Specification, RFC 1883, December 1995.

[RFC-2507] M. Degermark, B. Nordgren, S. Pink, IP header compression, RFC 2507, February 1999.

[RFC-2508] S. Casner, V. Jacobson, Compressing IP/UDP/RTP Headers for Low-Speed Serial Links, RFC 2508, February 1999.

[RFC-2509] M. Engan, S. Casner, C. Bormann, IP Header Compression for PPP, RFC 2509, February 1999.

This Internet-Draft expires in December 1999.