Abstract

To make mobile IP telephony over cellular radio systems an economically viable alternative to circuit-switched voice, it is an absolute requirement that the 40-octet IP/UDP/RTP headers on IP telephony packets be reduced in size to conserve bandwidth and radio spectrum. We evaluate the performance of the default header compression scheme for IP/UDP/RTP, CRTP (RFC-2508), over links built on cellular radio access technology. We find that CRTP does not perform adequately over such links, and suggest that a more robust header compression scheme must be developed to make IP telephony over cellular economically viable.

Evaluation of CRTP Performance over Cellular Radio Links

MIKAEL DEGERMARK, LULE UNIVERSITY OF TECHNOLOGY
HANS HANNU, LARS-ERIK JONSSON, AND KRISTER SVANBRO,
ERICSSON ERISOFT AB

With IP telephony gaining momentum and cellular telephony having several hundred million users, it seems inevitable that some future wireless telephony systems will be based on IP technology. What we know today as cellular phones may, in addition to telephony and video, have IP stacks, Web browsers, e-mail clients, networked games, and so on. If based on IP, 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 understood that applications for video, games, and so forth will also have to be supported.

It is vital for cellular phone systems to use 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, voice service should be as good as users expect from the cellular phone systems of today. A lower quality may only be accepted if costs are significantly reduced.

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 those 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 10^{-3} (one in 1000) or even 10^{-2} (one in 100) for some users some of the time.

The only cost efficient way to achieve sufficient voice quality over such channels appears to be 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 automatic repeat request (ARQ) scheme with link-local retransmission. Intolerably 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. Nor 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.

Header Compression

Speech data for IP telephony will most likely be carried by Real-Time Transfer Protocol (RTP) [1]. A packet will then, in addition to link-layer framing, have an IP [2] header (20 octets), a UDP [3] header (8 octets), and an RTP header (12 octets), for a total of 40 octets. With IPv6 [4] 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. A proposed standard for compressing RTP/UDP/IP headers over low-speed serial links, Compressed RTP (CRTP), has recently been approved by the Internet Engineering Task Force (IETF) [5, 6], RFC-2508, together with a way to negotiate parameters for header compression over PPP [7]. With CRTP, the minimal compressed headers are 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 *context*, that represents the current header, how it is expected to change, and so on. The differences carried in each compressed packet update the context; thus, loss of a packet will bring the context of the decompressor out of sync with the compressor since it is not updated correctly.

CRTP's 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 trigger the context repair mechanism outlined in the previous paragraph.

High BERs will cause the repair mechanism to be trig-

gered often, causing many FULL_HEADER packets or COMPRESSED_NON_TCP packets to be sent, which consume extra bandwidth. With a long round-trip 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 [5, 8] and also for CRTP in [6] 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 that 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.

Link Layers

When evaluating CRTP, link-layer framing must be considered. We use two framing schemes. One is Point-to-Point Protocol (PPP) in high-level data link control (HDLC)-like framing [9], which has a 16/32-bit cyclic redundancy check (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 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 framing scheme we use for evaluation purposes is an imaginary such link layer, henceforth called link layer with partial checksum (LLPC).

PPP in HDLC-Like Framing

PPP [9] typically uses HDLC-like framing [10]. With a 16-bit checksum and compressed Address and Control fields, frames carrying CRTP, COMPRESSED_NON_TCP, or FULL_HEADER packets have the format shown in Fig. 1. The Flag only occurs once between frames if they are sent back-to-back, so the amortized framing overhead is 4 octets/frame. The checksum (FCS) is calculated over the Protocol and Information fields (payload), but not the Flags or

■ Figure 1. The HDLC frame format.

■ Figure 2. The LLPC frame format.

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

Link Layer with Partial Checksum

LLPC is an imaginary framing scheme derived from the HDLC format by adding a one-octet Length field. The frame format is shown in Fig. 2. The Length field indicates the number of octets of the payload covered by the FCS. It can have values from 0 to 255. The FCS covers the Length and Protocol fields 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 the FCS 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 LLPC is 5 octets/frame.

Any errors in the Flag, Length, Protocol, FCS, or 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.

Description of Simulations

Here we describe the simulated scenario, and elaborate on the properties of the cellular link and the back channel.

The Simulated Scenario

The simulated scenario is shown in Fig. 3. 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. Over the Internet path are uniformly distributed losses which influence the efficiency of CRTP mechanisms, especially the Twice mechanism. Over the cellular link one of the framing protocols discussed earlier carry the packets. The radio channel of the cellular link represents fairly bad but realistic conditions. The round-trip time over the link 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/s (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 s. 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.

■ Figure 3. Simulated scenario.

■ Figure 4. FER for Ideal scheme for HDLC and LLPC.

The Cellular Link and the Back Channel

The cellular link is simulated accurately using a realistic radio channel model [11], for various BERs and added channel coding. The reported BERs are always the BERs after channel coding (i.e., the BER seen by the link layer). The simulated channel usually damages one or two consecutive frames; only at the highest BERs have three or more consecutive damaged frames been observed.

The interesting BERs for cellular systems are in the range between 10^{-3} and 10^{-6} . Circuit-switched cellular voice can deliver acceptable speech quality down to around 10^{-2} , while the system can become unnecessarily expensive at BERs much less than 10^{-6} .

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

Frame Errors

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

- 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.

- 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. This can, of course, be caused by header damage, but also by:

- Context damage: when the context of the decompressor is out of sync with the context of the compressor. This is caused by delta information being lost due to the first two types of damage.

For HDLC, both header 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 unrelated to header compression performance. Because of the way modern speech decoders are constructed, some bits are less important for intelligibility than others, and damage to those less important bits will not make the damaged frame completely useless. Delivering such frames to the speech decoder will decrease the voice quality less than discarding them.

Evaluation of CRTP Performance

In order to have a reference point, we first simulated what we call the ideal header compression scheme. The ideal scheme is not implementable. It has an overhead we believe is close to the maximum that can be tolerated for cellular IP telephony to be viable. A good header compression scheme should not perform much worse. The performance of the ideal scheme is then compared to the performance of plain CRTP, as well as with the performance of CRTP with a simple mechanism for local context repair called Twice.

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.

As can be seen in Fig. 4, for a BER of 10^{-3} the frame error

■ Figure 5. FER for CRTP, CRTP with Twice, and ideal for HDLC.

■ Figure 6. FER for CRTP, CRTP with Twice, and Ideal for LLPC.

rate (FER) is 1–2 percent for both link layers. LLPC is marginally better. At 5×10^{-3} there is a significant difference between HDLC (7.5 percent FER) and LLPC (4 percent FER). Loss over the Internet path would 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 shows how many packets with payload damage only there are: around 0.3 percent for a BER of 10^{-3} .

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

CRTP without Twice

With a round-trip 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 5 shows CRTP performance for HDLC, while Fig. 6 shows CRTP performance for LLPC. The ideal curves have been included for reference. The percentages indicate how much loss there was over the Internet path. The plots for CRTP with Twice are discussed in the next subsection.

In Figs. 5 and 6 one can see that for a BER of 10^{-3} , CRTP gives a FER of 8 percent with HDLC, while with LLPC the FER is 5 percent. 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 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, 5 octets are added to the remaining 9 headers. The average header size then increases by 5/9 octets (0.56 octets).

Figure 7 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 percent 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.

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 10^{-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 percent Internet loss, HDLC gives 3.2 octets and LLPC 3.0 octets.

CRTP with Twice

The Twice algorithm is a way to repair the context quickly without having to wait for a round-trip over the link. Twice makes assumptions about 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 attempted repair succeeded; typically the transport-layer (UDP) checksum is used for that purpose.

The plots in Fig. 5 show FERs for CRTP, CRTP with Twice, and the ideal scheme when HDLC framing is used.

■ Figure 7. Average header size for CRTP.

The CRTP with Twice plot is unrealistic in the sense that compressed headers in these simulations are only two octets (i.e., the UDP checksum is not passed). The simulator instead determined whether Twice would have been able to repair the context if there was a reliable way to verify a single repair attempt. For HDLC, this means that the Twice plot is slightly too optimistic, since the larger compressed header would have increased the error rate. It is evident from Fig. 5 that Twice improves the FER significantly, although CRTP with Twice is still much worse than the ideal scheme. At a BER of 10^{-3} , the FERs are less than 2 percent for ideal, about 4 percent for CRTP with Twice, and 8 percent for CRTP without Twice. More sophisticated implementations of Twice might get closer to the ideal curve, but on the other hand the UDP checksum is not strong enough to verify repeated repair attempts.

Figure 6 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. Since the UDP checksum covers the entire payload and is fairly weak, a scenario with a partial checksum on the link layer (LLPC) where Twice uses the UDP checksum to verify repair attempts does not make sense. Therefore, the CRTP with Twice plot shows the performance of Twice if there were another way to verify single repair attempts.

Loss Patterns

For applications such as interactive voice, not only the loss rate is interesting. Typical voice decoders will reuse earlier frames when a frame is lost, but might decrease the intensity at which that frame is played out. For each successive loss the intensity decreases 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. Figures 8 and 9 show histograms over the number of consecutive loss bursts of certain lengths for CRTP, with and without Twice, for three different BERs.

It is evident from Figs. 8 and 9 that the majority of loss events without Twice are such that around seven consecutive frames are lost. The link roundtrip time in these simulations was 120 ms and the packet rate 50 packets/s, which means that a single discarded frame would cause at least six additional frames to be lost due to context damage. When there is little loss before the compression point, Twice (or variants) perform rather well 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 8.** Lengths of frame loss bursts, HDLC, no IP loss.

Loss bursts involving seven to eight frames are clearly noticeable for most voice decoders. This is a major disadvantage of using CRTP over high-loss links with nontrivial link round-trip times. Even if the frame rate was 1/30 ms and the link round-trip time only 60 ms, the typical loss burst would be three to four frames (one discarded at link level, next discovers damage, update requested, update sent), which would decrease the voice quality significantly.

Using COMPRESSED_NON_TCP Only

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; on the other hand, it is more likely that a packet will be damaged because it is larger.

Figure 10 shows the FER when sending COMPRESSED_NON_TCP packets only for HDLC and LLPC. For HDLC, the FER when the BER is 10^{-3} is 3 percent, which is more than for the ideal scheme (< 2 percent) but less than CRTP with Twice (5 percent). The FER for LLPC is just over 2 percent and, similar to HDLC, it is more than the ideal scheme but less than CRTP with Twice.

Discussion

The packet loss rates of CRTP, CRTP with Twice, and the ideal header compression scheme are summarized in Table 1 for various error rates. The numbers are for HDLC-like fram-

Bit-error rate	10^{-5}	10^{-4}	10^{-3}	10^{-2}
Ideal	0	0.4%	1.8%	11%
CRTP+Twice	0	1.0%	5.0%	24%
CRTP, no twice	0	1.5%	8.0%	40%

■ **Table 1.** Frame loss rates, HDLC.

ing (i.e., errors in any part of a packet means it is discarded). The payload is 16 octets. The ideal scheme discards packets only when the packet itself is damaged.

It is evident from Table 1 that CRTP performs well for BERs less than 10^{-5} , but not as well for BERs higher than 10^{-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 with Twice become intolerable at high BERs.

The major cause of CRTP's bad performance is that many packets are discarded due to context damage while waiting a link round-trip time for the repair mechanism.

Twice is a way to repair the context locally. It requires 2 extra octets of header (the UDP checksum) to verify its repair attempts. The 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 often fail. Moreover, the UDP checksum is too weak to reliably determine the success or failure of several 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 round-trip time. This is problematic for low-bandwidth voice codecs, which cannot mask such long loss events well. Hence, 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.

Overall, the improvement in FER is around 40 percent 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.

Conclusions

CRTP does not perform well over lossy links with long round-trip times. Twice can improve the situation somewhat, but the loss rate is still too high. A disadvantage of Twice is that it requires that the UDP checksum be 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 several repairs. Because of all this, CRTP with Twice cannot approach the performance of the ideal scheme.

It is evident that a more robust header compression

Bit-error rate	10^{-5}	10^{-4}	10^{-3}	10^{-2}
Ideal	0	0.3%	1.4%	7%
CRTP+Twice	0	0.8%	4.2%	18%
CRTP, no twice	0	1.1%	5.0%	25%

■ Table 2. Frame loss rates, LLPC.

scheme than CRTP is needed over cellular links. Such a scheme must be robust against several consecutive packets being lost between compressor and decompressor. To obtain a viable end-to-end solution, the scheme must also be robust against several consecutive packets being lost before the compression point, i.e., the header compression scheme on the last cellular link must be robust against the losses on the first cellular link.

References

- [1] H. Schulzrinne et al., "RTP: A Transport Protocol for Real-Time Applications," IETF RFC 1889, Jan. 1996.
- [2] J. Postel, "Internet Protocol," IETF RFC 791, Sept. 1981.
- [3] J. Postel, "User Datagram Protocol," IETF RFC 761, Aug. 1980.
- [4] S. Deering and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification," IETF RFC 1883, Dec. 1995.
- [5] M. Degermark, B. Nordgren, and S. Pink, "IP Header Compression," IETF RFC 2507, Feb. 1999.
- [6] S. Casner and V. Jacobson, "Compressing IP/UDP/RTP Headers for Low-Speed Serial Links," IETF RFC 2508, Feb. 1999.
- [7] M. Engan, S. Casner, and C. Bormann, "IP Header Compression for PPP," IETF, RFC 2509, Feb. 1999.
- [8] M. Degermark et al., "Low-loss TCP/IP Header Compression for Wireless Networks," *Proc. 2nd Annual Int'l. Conf. Mobile Comp. and Networking*, 1996, ext. version: ACM/Balzer J. Wireless Networks, vol. 3, no. 5, 1997, pp. 1-14.
- [9] W. Simpson, "PPP in HDLC-Like Framing," IETF RFC 1662, July 1994.
- [10] W. Simpson, "The Point-to-Point Protocol (PPP)," IETF RFC 1661, July 1994.
- [11] "Procedure for Evaluation of Transmission Technologies for FPLMTS," ITU-R TG8-1, 8-1/TEMP/233-E, Sept. 1995.

Additional Reading

- [1] J. Postel, "Transmission Control Protocol," IETF RFC 793, Sept. 1981.
- [2] V. Jacobson, "Compressing TCP/IP Headers for Low-Speed Serial Links," IETF RFC 1144, Feb. 1990.

■ Figure 10. FER for COMPRESSED NON TCP only.

■ Figure 9. Lengths of frame loss bursts, HDLC, 10% IP loss.

Biographies

MIKAEL DEGERMARK (micke@cdt.luth.se) is a visiting assistant professor at the University of Arizona in Tucson. He is on leave from Lulea University of Technology, Sweden. He has a Master's degree in computer science (1987), and a Ph.D. in computer communication (1997), both from Lulea University. He has been active in the IETF since 1994, and currently chairs an IETF Working Group developing robust header compression algorithms suitable for cellular links.

HANS HANNU (hans.hannu@ericsson.com) received an M.Sc. degree in electrical engineering from Lulea University of Technology, Sweden, in 1998. He has been with Ericsson Research since November 1988, working with voice over IP and in particular IP header compression. His research interests also include QoS in IP-based systems.

LARS-ERIK JONSSON (lars-erik.jonsson@ericsson.com) is a researcher at Ericsson in Lulea, Sweden. He received his M.Sc. degree in computer science in 1998, majoring in computer communication. Since then he has been working on the intersection of telecom and computer communication, focusing on IP-based solutions in cellular environments. Most of his research has been on header compression algorithms for error-prone links; he has contributed to the IETF on this topic since 1999.

KRISTER SVANBRO (krister.svanbro@ericsson.com) received his M.Sc. degree in computer science from Lulea University of Technology in 1997, majoring in signal processing. He has been with Ericsson Research since 1997, and initially worked with radio network solutions for generation 2.5 cellular systems. From 1999 his research has focused on enabling efficient voice over IP over wireless for third-generation cellular systems (e.g., WCDMA and EDGE). He currently leads a team developing algorithms enabling efficient realization of real-time IP services in all-IP third-generation cellular systems. Robust header compression is a key algorithm being developed in that group.