IP Header Compression for Media Streaming in Wireless Networks

Tatiana Kozlova Madsen[†], Frank H.P. Fitzek[†], Ramjee Prasad[†], Marcos Katz[‡]
[†] Center for TeleInfrastructure, Aalborg University
Neils Jernes Vej 12, 9220 Aalborg Øst, Denmark,
phone: +45 9635 8632, e-mail: *[tatiana—ff—prasad]@kom.auc.dk*
[‡] Telecommunication R&D Center, Samsung Electronics Co. Ltd., Suwon, Korea

***Abstract*— To efficiently deliver multimedia under bandwidth constrains, header compression schemes should be implemented to reduce overhead when possible. In the paper, we address the issue of header compression for multimedia streaming services. The main focus is on how to keep the synchronization between compressor and decompressor without the use of feedback information. We propose a scheme based on the concept of additional information container (AIC). Usage of AICs helps to make header compression robust against channel errors. In the case of bursty error pattern, interleaving of AICs keeps high performance level. Additionally, discussion on the optimal design of AICs is provided.**

I. INTRODUCTION AND MOTIVATION

The wireless delivery of multimedia services is one of the goals of the next generations mobile communication systems. However, IP-based multimedia applications, including audio- and video-streaming and gaming, require more bandwidth than traditional voice services. Due to bandwidth constraints on wireless links, it is useful to compress multimedia payload and reduce header overhead. Compression of IP packet headers results in significant reduction of overhead information [1].

Header compression algorithms have been developed for over a decade. Analysis of the variations of the field information of the packet flow is used to decide the smallest amount of information needed to reconstruct the header fields on the receiver side. This information is mostly coded as delta values. Compressed headers are comprised of delta values and they are transmitted instead of original headers. One of the first proposed HC schemes was Van Jacobson Header Compression [2], introduced in 1990. Other compression schemes are IPHC [3], CRTP [4] and developed especially for wireless environment ROHC [5].

Header compression (HC) is possible due to redundancy among the header fields of a packet flow. The main concept is as follows. On the sender side, the compressor removes redundancy from the incoming packet using information from the past packets, called the *context* (it is also referred to as a base). The decompressor maintains the context and uses it to reconstruct the header of the incoming packet. The inconsistencies in the contexts of the compressor and the decompressor lead to the loss in synchronization and failure of the decompression procedure. To prevent the context re–synchronization, a context repair mechanism should be applied. A sender will rely on the feedback from a receiver to know when to transmit a packet with an uncompressed header. When the feedback is unavailable, the synchronization of the context is achieved by periodic refreshes of the states. How often the updates should be done, depends a lot on the channel error rate and the propagation environment. For example, Robust Header Compression offers a mode that is designed for links without a return channel, unidirectional mode. In U-mode transitions between compression states are done based on periodic timeouts and irregularities in the header field patterns.

One should not think that ”no feedback” situation is very rare in communication networks. Let consider a multicast scenario when several mobile terminals receive media streaming from one base station or access point. Because multicast service is a one-to-many point service, providing feedback from header decompressors to the compressor is difficult or impossible at all. There can be situations when a return path is undesirable, e.g. for delay-sensitive applications. Additionally, wireless medium is characterized by error-prone links. Because of retransmissions on the MAC layer, we can expect a delayed feedback. If the compressor does not receive the feedback in time, it assumes resynchronization and transmits an uncompressed header. This will lead to loose in compression gain. The feedback should be error-free, otherwise the compression scheme is damaged. Another point is that providing the feedback means a lot of additional signalling, that can lead to spectrum-inefficiency. Not only the bandwidth, but also energy is a limited resource in wireless networks. If the feedback is constantly sent, the power consumption of a mobile terminal is increased. For these reasons, we have chosen to focus on the header compression schemes that do not rely on the feedback information.

The main drawback of the header compression schemes without feedback is their sensitivity to the channel errors. If a packet is lost, it usually results in the loss of the base at the receiver side. All subsequent packets can not be decompressed until a receiver gets a full context update at the beginning on the next frame. It makes header compression schemes based on delta-coding with no feedback very vulnerable to the loss of coded packets.

To prevent the context resynchronization, a scheme based on the cooperative behavior of multiple channels, Cooperative Header Compression (COHC), is proposed in [6]. It has been proven to be efficient when multiple channels are available between a sender and a receiver [7]. In this paper we consider the extension of COHC in the case of a single-channel, single-stream communication. We also present a special case when a number of channels between a source-destination pair varies in time, as it is often a case in mobile ad hoc networks.

The paper is organized as follows. Section II describes the proposed approach. Section III shows the performance evaluation of COHC for single-channel communication both for independent and bursty error patterns. It also contains evaluation for the case of varying number of channels. Section IV elaborates on the design of AICs. Section V gives some concluding remarks and outline for the future work.

II. PROPOSED APPROACH - EXPLOITING COOPERATION IN HEADER COMPRESSION

Header compression schemes are commonly considered in a single-channel, single-flow configuration. Recently, the cooperative header compression (COHC) scheme has been proposed [6]. This scheme proposes to send extra compressed information (additional information container - AIC) through the other channels in piggyback form in order to minimize the desynchronization of compression contexts. The advantage of this scheme is it's simplicity and no need of feedback information from the receiver. A detailed description of COHC can be found in [6].

COHC has proven to perform well when we have a constant number of parallel channels between the sender and the receiver. In general, multi-channel communication offers more flexibility, robustness and higher capacity. For example, increase in system performance can be achieved by exploiting path diversity effect or QoS can be provided in wireless network domain by using multichannel MAC protocols. COHC introduces cooperation between multiple channels to prevent synchronization. Besides sending periodic context updates, COHC maintains the synchronization by using information provided by neighboring channels.

The need for multiple channels can be dictated by applications, e.g. streaming multimedia data that usually consists of several individual RTP-based streams. Clients that receive multimedia (e.g. a streaming video clip) typically receive two RTP-based streams, one for audio and one for video data. Another example is Multiple Description Coding (MDC) and Multi Layered Coding (MLC) for coding a video stream. Encoding process results in splitting a connection into multiple flows, each of which can be transmitted over a separate channel.

Of course, not always multiple channels are available for communication or multiple connections are established between a source and a destination. However, cooperative approach for IP header compression can be used also in the case of a single-channel, single-stream communication. In the following we explain how COHC can be adapted to work with a single-connection.

In the proposed approach, each header compression entity is carrying additional information, an AIC. An AIC is used to repair synchronization loss by reconstructing the context: header of packet k has additional field containing information that is sufficient to reconstruct the base of header $k-1$. If packet k is lost, packet $k+1$ will be readable, because header of packet $k+1$ can be decompressed with the help of base k from the AIC. Comparing with the case of the delta coding, one can easily observe that for delta coding if one packet is corrupted, all the subsequent packets will be undecodable. We refer to this situation as *loss propagation*. In the proposed scheme the context can be repaired (but not the packet itself!) and loss propagation is avoided.

The design of an AIC has a big impact on the overall performance of the header compression scheme. From one hand, appending AICs to the packet, we make the size of compressed headers bigger and, thus, reduce the compression gain. From another hand, using AICs we are able to reduce loss propagation effect and improve the rate of successfully decompressed headers. There is a clear tradeoff between robustness of a scheme and a compression gain. We elaborate on the optimal design of AICs in Section IV.

So far, we consider the case when we have a fixed number of channels between a source and a destination: a fixed number of multiple channels or a single channel. There can occur a situation when the number of channels is varying during the process of communication. This can happen e.g. in ad hoc networks when the nodes are mobile. At the beginning of the communication process, a number of routes is established between a source and a destination; at some later point in time, due to node mobility some of the paths will be broken and later, by using route maintenance mechanism, some new routes will be found. In this case COHC can be also successfully applied. Each time there is a change in the number of channels, we are sending a full context update; otherwise we apply the same cooperation principles as in the case of the constant number of channels.

III. PERFORMANCE EVALUATION

This section provides the performance analysis of the discussed header compression schemes. The most important performance parameters are packet error probability (PEP) and bandwidth efficiency BE, defined as a ration of correctly received payload to the total amount of information sent. Low PEP is an indication of robustness of a scheme. Bandwidth efficiency reflects both the compression gain and the error rate.

In this paper we focus on the impact of the channel errors on the context resynchronization. Therefore, in the following we assume that the changes in the header field are regular and consider only the situation of context periodic updates.

First, we present the results for the case of a single-channel communication, then, using an example of an ad hoc network, we consider the case of varying number of channels.

A. Single-channel communication

1) Uncorrelated error pattern: Assuming independent errors, the following formula can be derived for PEP in the case of simple delta coding:

$$PEP = 1 - \frac{(1-p)(1-(1-p)^N)}{Np}$$

where p is a probability to loose a packet and N is a number of packets in a frame.

The derivation of formula for PEP for the cooperative scheme is somewhat more complicated. First, we find probability to loose a base on k-th place in a frame $BL_k$by the following recursive formula:

$$BL_k = p^2(1-p)(1-\sum_{i=2}^{k-3} BL_i), i = 5, \ldots, N \qquad (1)$$

Using formula (1), we can derive probability PL_k that k-th packet is lost (either due to channel error or due to loss of base):

$$PL_k = p + (1-p)\sum_{i=2}^{k-1} BL_i$$

Finally,

$$PEP = \frac{1}{N}\sum_{k=1}^{N} PL_k$$

Bandwidth efficiency can be found as

$$BE = \frac{ND(1-PEP)}{ND + B_u + (N-1)B_c + (N-1)B_a}$$

where D is a packet payload, B_u, B_c and B_a are the sizes of uncompressed, compressed headers and the size of additional information respectively.

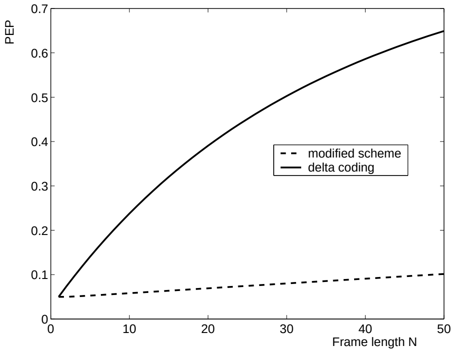

Fig. 1. Analytical results for PEP vs frame length for uncorrelated errors with $p = 0.05$

Figure 1 gives results for PEP versus the frame length. Figure 2 presents bandwidth efficiency versus frame length for the following set of parameters: $D = 1400, B_c = 6, B_u = 40, B_a = 6$ bytes. The figures clearly show that the cooperative scheme outperforms significantly the simple delta coding and it is very efficient in case of uncorrelated errors.

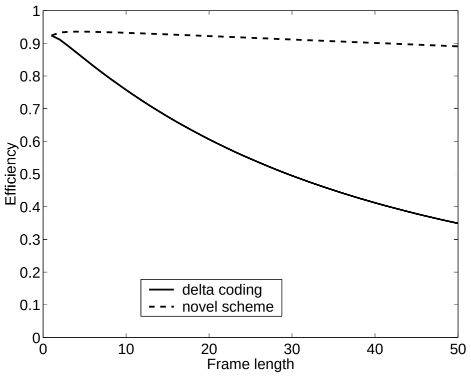

Fig. 2. Efficiency of simple and modified delta coding schemes for uncorrelated errors with $p = 0.05$

2) Correlated error pattern: The cooperative scheme can cope efficiently with uncorrelated errors. But if two packets in a row are corrupted due to channel errors, the base can not be retrieved and the error will propagate until the end of the frame (until a full uncompressed header is transmitted). Bursty error pattern is typical for wireless links. We consider further modifications of the scheme in order to adopt it to bursty transmission errors. One approach is to append extra AICs to a packet header such that the information in header k can be used to recover m previous bases (in this way, a base can ”survive” a burst of length m). The disadvantage of this approach is a very low compression gain.

Another method that can be applied is randomization of bursty errors, widely used in wireless communication, namely interleaving. By interleaving process, a burst of losses on a channel is transformed into a sequence of isolated losses. Applying interleaving in our case, we show that one AIC per packet is enough to compensate for a base loss.

To model correlated errors we use a two-state Markov chain (Gilbert-Elliot model). Figure 3 presents PEP for delta coding, the cooperative scheme and the scheme with interleaving. One can observe that interleaving of AICs improves the performance significantly. Here we can draw an analogy between our approach using interleaving and interleaving combined with error concealment techniques that perform best with small gaps.

The price to pay for reduction in PEP by using interleaving process is increase in delay. We would like to point out that instead of interleaving the packets, we interleave only AICs. This helps to keep the latency low when no errors occur.

B. Varying number of channels

In order to evaluate the performance of COHC in case of varying number of channels, we have performed simulation studies of an ad hoc network using *ns-2* simulation tool [8]. Simulations have been carried out with 50 nodes randomly distributed in an area of 1000 x 1000 m^2. A multi-path extension of Dynamic Source Routing protocol [9] is used to provide multiple routes between one source - destination pair.

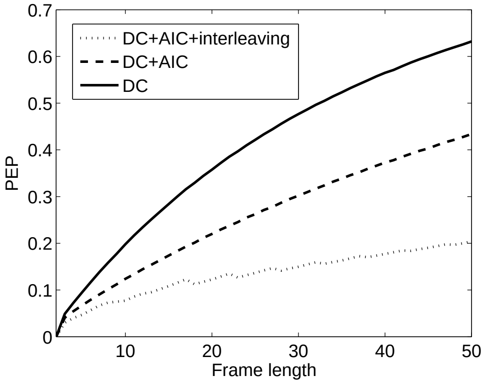

Fig. 3. PEP for correlated error pattern case (Gilbert-Elliot model, $P_{GB} = 0.05, P_{BG} = 0.5$)

A free space propagation model is considered. A constant bit rate stream of data with a payload of 20 bytes is generated.

Figure 4 illustrates the number of available routes in the routing cache. Maximum number of three cooperative channels is used, even if more routes are available. In case there is only one route in cache, for each compressed header two AICs are appended.

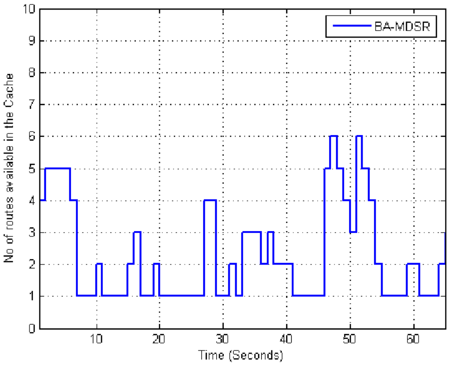

Fig. 4. Number of available routes in an ad hoc network.

Packet delivery and decompression ration (PDDR) versus node mobility is shown in Figure 5. Figure 6 presents the results for bandwidth efficiency. For the comparison reason, lines corresponding to the case when no header compression is used, are also plotted on the figures. Even in the case of high mobility, when the links are broken more often, COHC provides robustness against channel errors and shows PDDR improvement of approx. 50% compared with delta coding. High PDDR leads to the high bandwidth efficiency for COHC (Figure 6).

IV. DESIGN OF AICS

To minimize the size of an AIC, no information belonging to a packet itself should be included, but only the information necessary to recover a base. The easiest solution is to make an AIC just a copy of a compressed header of a previous packet. One can notice that it is not necessary to include UDP checksum in an AIC, since it corresponds to the whole packet content, but is not required to update the base. In this way, the size of an AIC can be reduced by 2 bytes.

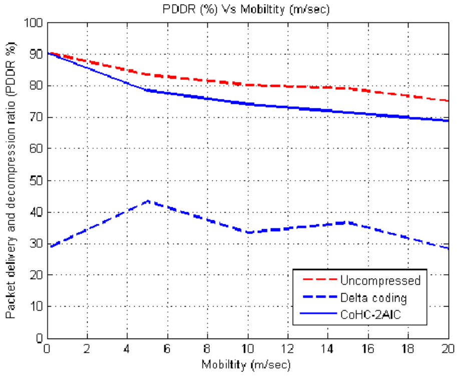

Fig. 5. Packet delivery and decompression ratio.

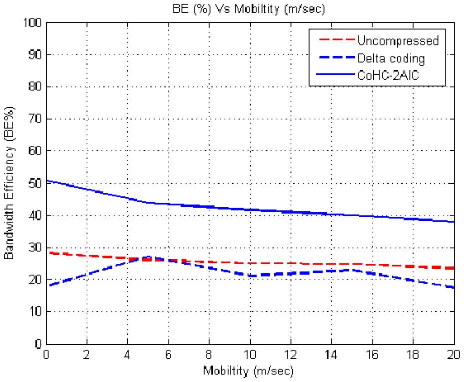

Fig. 6. bandwidth efficiency of different schemes in an ad hoc network.

We have noticed that in some specific cases the size of AICs can be reduced even further. If one considers MDC coded video, the redundancies among headers of different MDC substreams can be exploited to make the design of AICs more optimal. It turns out that there is no need to include the values of some fields of a compressed header in AICs since they can be derived using compressed headers of other MDC substreams.

We illustrate this idea by performing the following test. We have implemented MDC encoder on the basis of Linux tools. Three MDC descriptors were produced; the data was sent over an IP network and captured with Ethereal; the headers were extracted, compressed and interdependencies of compressed header fields were analyzed.

We would like to highlight that the cooperative header compression should be considered as a general approach that can be applied to any IP header compression scheme to improve it's performance. This idea is applicable even to schemes that uses feedback information. In this case cooperation can help to reduce signalling from the decompressor.

In what follows, we use RFC 2508 [4] to compress the headers. Usually, the headers fields are classified into fixed, inferable, delta and random fields. The fields of the later type vary randomly and therefore, they are always transmitted. Our focus is on the delta fields: delta of IP ID, delta of RTP sequence and delta of RTP timestamp. In the observed example the two first delta fields are always +1. It is more interesting to analyze the delta of RTP timestamp field, since its changes for one sub-stream can not be predicted.

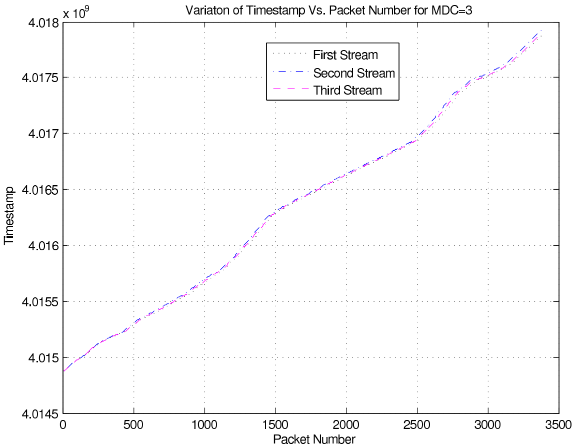

Fig. 7. Variation of RTP timestamps for three MDC substreams.

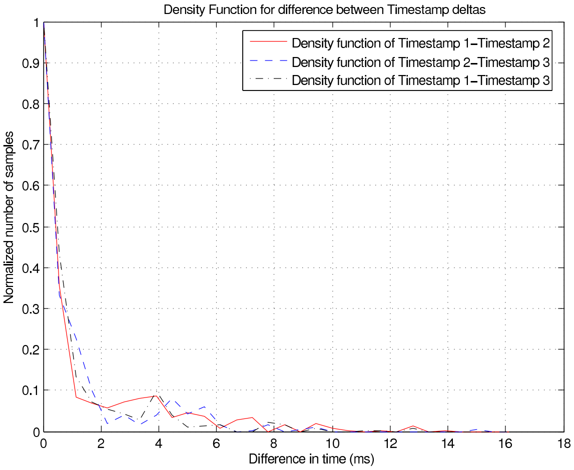

Fig. 8. Density function of differencies between deltas of RTP timestamps.

Figure 7 shows the evolution of RTP timestamps of three MDC sub-streams. One can observe a high correlation between these values. The timestamps from three sub-streams are not the same, but very close: with the resolution used for Figure 7, the three curves almost coincide. To evaluate the correlation between the channels, we consider the compressed headers and plot the density function of difference between the timestamps deltas, that is, $\Delta RTP\ TS_i - \Delta RTP\ TS_j$ where $i \neq j$ (see Figure 8). For the recorded data, in 90% of the observations the ”delta of deltas” is smaller than 1 ms and it is never larger than 18 ms. Thus, if a packet with a compressed header belonging to one sub-stream is lost, we can update the RTP timestamp value by using the corresponding delta field from another MDC sub-stream. By modifying the context update process in this way, we introduce an error to the RTP timestamp, but, as can be seen from Figure 8, this error is marginal and can be tolerated by an application. This error will be removed with the next full context update.

The described above cooperative method allows us to exclude delta fields from AICs in the situations similar to the considered one. This reduces the size of an AIC and improves the efficiency of COHC making this scheme attractive to use in the case of bandwidth-limited links. Overall, COHC is a robust scheme that can withstand a significant amount of channel errors.

V. CONCLUSION AND FUTURE WORK

We have presented a header compression approach based on usage of additional information. It is particular suited for wireless environment since it can cope successfully with bursty channel errors.

We have compared the cooperative scheme with the conventional delta coding and showed that, in spite of loss in compression gain, it has better bandwidth efficiency. In our future work we plan to consider a scheme that has the same compression gain as the AIC-scheme, but instead of appending additional information, stronger error protection is used. The main difference between the AIC-scheme and the compression with error protection is that in the first case we aim at the recovery of the context, while in the second case the goal is to receive the whole packet correctly. Furthermore, a scheme with unequal error protection should be evaluated when excess error protection is given to a compressed header.

VI. ACKNOWLEDGEMENT

The authors would like to thank S. Nethi for his help with the ns-2 simulations.

REFERENCES

[1] F.H.P. Fitzek, S. Hendrata, P. Seeling, M. Reisslein. Chapter in Wireless Internet – Header Compression Schemes for Wireless Internet Access. CRC Press. 2004.

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

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

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

[5] C. Bormann, C. Burmeister, M. Degermark, H. Fukushima, H. Hannu, L-E. Jonsson, R. Hakenberg, T. Koren, K. Le, Z. Liu, A. Martensson, A. Miyazaki, K. Svanbro, T. Wiebke, T. Yoshimura, and H. Zheng, Robust Header Compression: Framework and four profiles, Request for Comments 3095, July 2001.

[6] F.H.P. Fitzek, T. K. Madsen, P. Popovski, R. Prasad and M. Katz. Cooperative IP Header Compression for Parallel Channels in Wireless Meshed Networks. In *IEEE International Conference on Communication (ICC)*, Seoul, Korea, 2005.

[7] T. Madsen, F. Fitzek, Y. Takatori, R. Prasad and M. Katz, ”Cooperative IP Header Compression using Multiple Access Points in 4G Wireless Networks”, in *Proceedings of IST Wireless Summit*, 2005.

[8] Network Simulator - ns-2, http://www.isi.edu/nsnam/ns/.

[9] A. Boursier, S. Dahlen, L. Marie-Francoise, T. S. Marin, S. Nethi, Multipath DSR Protocol for Ad hoc Networks, Report. Aalborg University. January 2005.