Zero–AIC Header Compression with Multiple Description Coding for 4G Wireless Networks

Tatiana Kozlova Madsen[†], Yasushi Takatori[†], Frank H.P. Fitzek[†], Ramjee Prasad[†], Marcos Katz[‡]
[†] Department of Communications Technology, Aalborg University
Neils Jernes Vej 12, 9220 Aalborg Øst, Denmark,
phone: +45 9635 8632, e-mail: *[tatiana|yt|ff|prasad]@kom.auc.dk*
[‡] SAMSUNG Korea

***Abstract*— It is expected that video streaming services will play an important role in 4G wireless systems. Since video communication has typically large and varying bandwidth requirements, high spectrum utilization techniques are needed to support high quality video. In this paper we propose a combination of Cooperative Header Compression (COHC) method with Multiple Description Coding (MDC), which introduces robustness and yields high compression gain for the compression scheme without the need of feedback channel. MDC is a very attractive coding scheme that is suitable for heterogeneous terminals and can benefit from multipath diversity. Unfortunately, MDC introduces overhead in terms of network and coding overhead. Here we focus on the methods to reduce the network overhead. The proposed header compression scheme is based on cooperative behavior of parallel channels. The obtained results show that the proposed method outperforms other combinations of MDC with the header compression schemes in terms of decompression rate.**

I. INTRODUCTION AND MOTIVATION

As new services emerge for wired networks, there arises a user demand to incorporate those services in wireless communication systems. Despite the efforts that were made, the wireless protocol domain is still challenging. The limitations of the currently available technologies have brought researchers to start reflecting on the Fourth Generation (4G). The upcoming 4G is expected to provide a convergence platform for a wide variety of new services, from high-quality voice to high-definition video, through high-data-rate wireless channels. Various visions of 4G have emerged recently among the telecommunication industries, the universities and the research institutes all over the world (see e.g. [1], [2]). And it is anticipating that video streaming will be the dominant force for the success of 4G wireless systems.

Multiple description coding (MDC) has gained a lot of interest lately, because it is an attractive coding scheme for robust transmission over channels with transient bursty error characteristics. MDC allows to split one source of information (audio or video) in multiple entities (called *descriptors*). Different descriptors may be sent over different channels. At the receiver each descriptor is decoded in a stand-alone fashion, that is independently of the other descriptors. In the balanced case, when all descriptors have equal rates and thus carry the same amount of information, the perceived quality at the receiver is proportional to the number of correctly received descriptors. In contrast to multiple layered coding (MLC), where without a base layer the enhanced layers can not be decoded, MDC has no prioritized layers.

The MDC approach has a number of advantages and features that make it attractive for usage in the next generation systems. First of all, MDC can reduce the end-to-end delay because the errors result only in degraded quality instead of frame retransmission. Secondly, MDC enables support for heterogeneous terminals, that is the more advanced the terminal is, the more descriptors is can receive and use to reconstruct the audio or video stream. MDC methods, proposed in [3, 4] are focusing on the heterogeneous terminals scenarios. Thirdly, splitting the connection into the multiple flows offers the possibility to obtain diversity gain by transmitting the flows via different paths. For example, an unbalanced MDC scheme to provide path diversity is proposed in [5].

The major drawback of MDC is generation of the overhead by source splitting process. The overhead can be defined as the amount of data of the splitted streams in comparison with a single stream. The overhead is generated by the encoding process and the IP network. In [6] the measurements for the encoder overhead for twelve different video sequences are presented. It is shown that the overhead depends heavily on the video content, as well as the encoding process and the number of descriptors. It is also shown that the network overhead is not negligible; on the contrary, it comprises the largest part of the total overhead.

The transmission of multiple descriptors to the receiver is mostly done by using the real time protocol (RTP), the user datagram protocol (UDP) and the Internet Protocol (IP). Using RTP/UDP/IP, an overhead of 40 byte per stream has to be taken into account for IPv4. In Figure 1 the encoder and network overhead for the `foreman` video sequence using MDC is given. Three different quantization values (namely 1, 31, and 51) versus the number of parallel descriptors (sub–streams) is given for IP version 4. Low quantization parameters refer to high quality video, while higher quantization values are reducing the video quality. In Figure 1 the curve entitled QP 1, QP 31, and QP 51 gives the overhead that is introduced only by the encoding process. While the QP X + Network gives the overhead of the encoding process with quantization level X in addition to the network overhead. It can be seen that the overhead increases dramatically due to the network overhead and not due to the encoder overhead if large quantization values are used. For small quantization values the impact is less. But for efficient video encoding the quantization values will be large (between 31 and 51) assuring both high video quality and low data rates.

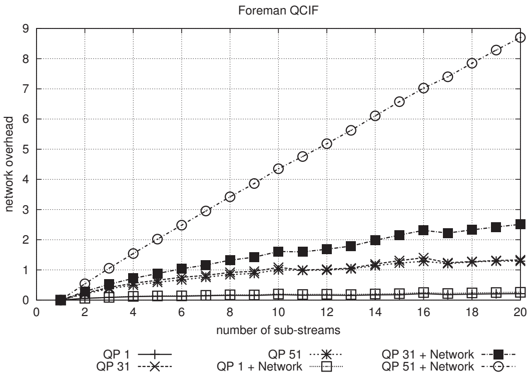

Fig. 1. Network Overhead (RTP/UDP/IPv4) for the foreman video sequence and three different quantization values.

The large overhead translates into the high bandwidth requirements that are difficult to fulfill in the wireless environment. Therefore, to keep MDC attractive for wireless networks, methods to reduce overhead should be considered. Thus, IP header compression becomes very important for MDC streaming. This paper is focusing on header compression techniques and we propose a combination of Cooperative Header Compression (COHC) [7] and MDC for video streaming. COHC works with multiple channels and exploits channel diversity to achieve robust header compression without the need of the feedback from the receiver. To benefit from multi path diversity, compression of different sub-streams is done in the cooperative manner. COHC introduces the concept of an *additional information container* (AIC). AICs generated for a particular sub-stream are appended to other sub-streams and delivered to the receiver via different paths. At the receiver side, AICs are used to keep the header compressor and decompressor synchronized, facilitating decompressing process and preventing loss propagation. We observe that in the case when different streams are MDC sub-streams of one source, the size of the AICs can be reduces to zero. In other words, the compressed headers of other MDC sub-streams will play the role of AICs: they can be used to synchronize the compressors and decompressors of the sub-streams. This leads to a simple and very efficient compression scheme based on delta coding.

II. COOPERATIVE HEADER COMPRESSION

A. *Header Compression for Wireless Communication*

IP header compression mechanisms have always been an important part of saving bandwidth over bandwidth-limited links. Many header compression schemes exists already, but they should be adopted to operation in the wireless environment and designed to withstand loss of packets due to severe propagation conditions. Among the most known are Van Jacobson Header Compression [8], IPHC [9] and CRTP [10]. Many of the header compression schemes are based on *delta coding*: one uncompressed header is sent and followed by a row of compressed headers that carry only the differential information referring to the previous header. This approach does not require the feedback channel but is very sensitive to packet losses: if one packet is lost, the base at the decompressor is not updated and all the subsequent packets, even if received correctly, can not be decompressed. We refer to this situation as *loss propagation*.

Recently, Robust Header Compression [11] was proposed especially developed for wireless multimedia delivery. It defines three states of compression when no context, static or full context are available for decompression. Besides that three mode of operation are introduced: Unidirectional (U-mode), Optimistic (O-mode) and Reliable (R-mode). Only the U-mode works without the feedback, in other modes the transitions between the states is done based on the acknowledgements from the decompressor. Thus, ROHC combines robustness for IP-based data streams and high compression gain due to connection-oriented approach in removing packet redundancies. The price to pay is high complexity of the scheme.

B. *Cooperative Header Compression*

To prevent the context re–synchronization, a context repair mechanism should be applied. When the feedback is unavailable (like in COHC or Unidirectional mode of ROHC), the synchronization of the contents 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. Rare periodic updates can lead to a situation when a packet is received correctly but can not be decompressed because of absence of the correct current base. Frequent updates keep the system robust, but compression gain is decreased. The main idea of cooperative header compression is to have a robust and efficient scheme in case the feedback can not be provided from the decompressor. By using AICs, the updates can be sent less frequently that ensures high compression gain, at the same time maintaining robustness of the scheme. AICs introduce some additional overhead compared with delta coding or ROHC, but the overall bandwidth efficiency of the method is improved significantly.

Figure 2 shows one possible way of AICs construction. Each compressor generates its own compressed header and the related AIC, which is passed to the neighboring compression entities. The neighboring entities in turn send their AICs such that the compressor is able to compose the payload, the compressed header and the AICs for the neighboring channels.

In the case one packet, e.g. CH(1,3), is lost due to the channel errors, the loss propagation can be avoided if one of the packets from other channels is received correctly. Using AIC(1,3) delivered by channel number 3, base of the first channel can be repaired. Only in the case when all packets on all channels at the same time are corrupted, we would have to wait until the beginning of the next frame; until that the packets

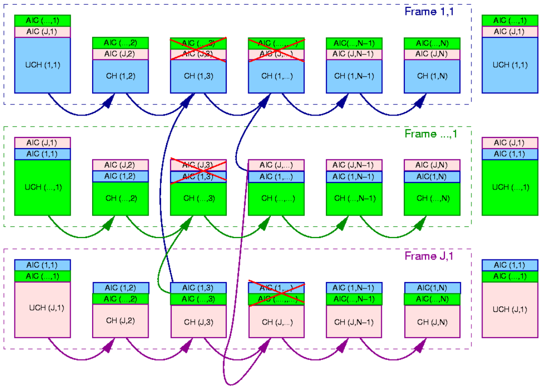

Fig. 2. AICs construction for three cooperative channels.

can not be decompressed. The bigger number of parallel channels are, the smaller is the probability that propagation loss will occur. One should note that here we assume that the packets on different channels are received ’in tact’, that is all channels have the same delay characteristics within the granularity of IP packets. This is a feasible assumption if we consider MDC streaming.

C. Header Compression Scheme for MDC Streaming

As MDC splits the video stream into multiple descriptors, we consider the frame based approach using a splitter entity. The splitter, as given on Figure 3, takes the raw video sequences and split them into J sub-sequnces, such that the i-th sub-sequence contains picture i, $J+i$, $2J+i$, and so on. These J subsequencies are feeded into the video encoder.

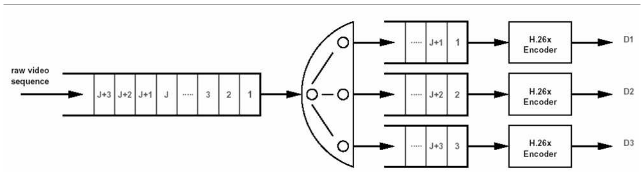

Fig. 3. Construction of multiple substreams

In principle now any header compression scheme can be applied to each of the obtained sub-streams independently. One should notice though that header compression algorithms without feedback would be preferable, since erroneously received packets are discarded by the decoder and not retransmitted. Thus, one choice is delta coding with the periodic update of the context: packets are organized in frames of N; only the first packet in the frame carries uncompressed header, the rest has compressed headers. Alternatively, ROHC U-mode can be used. In U-mode the compressor starts from the initialization and refresh (IR) state (see Figure 4), and then it goes to the second-order (SO) state directly (fast operation) or in normal operation it first moves to the first-order (FO) state and later proceeds to the SO state. The values for the parameters n_1, n_2 and n_3 are not defined in RFC 3095 [11]; they can be chosen e.g. equal to 1. Fast operation is essentially the same as delta-coding. In the normal operation mode the transitions are done in periodic manner between three states, whereas in delta coding approach there are two states. In this paper we limit our considerations only to the case of delta coding.

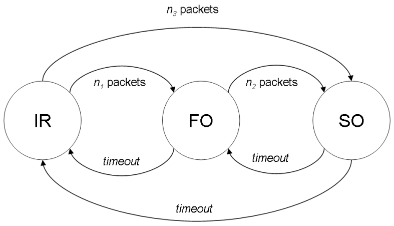

Fig. 4. State machine of the compressor in U-mode of ROHC.

When multiple paths are available between a source and a destination, COHC can be applied. MDC produces multiple streams that are sent over different channels (physical or logical). Therefore, in the case of MDC coded video COHC can be immediately used.

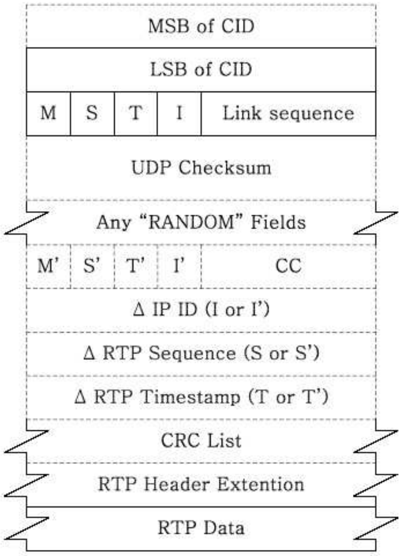

Fig. 5. Compressed header for RFC 2508.

The question remained to answer is how to construct the AICs in this case. The easiest approach is just to take a copy of the compressed header and send it over other channels with other MDC sub-streams. However, due to the redundancies among headers of different MDC sub-streams, the compressed headers themselves can play the role of AICs. If a packet is lost due to the bad channel conditions, the decompressor will use the compressed headers from other sub-streams to reconstruct the base and to decompress the next received packet.

Lets consider RFC 2508 [10] as an example. Figure 5 shows the format of the compressed RTP/UDP/IP header. We have analyzed each of the fields of the compressed header and correlation amoung the fields for different MDC substreams. It was observed that for balanced MDC most of the fields are the same or their delta is constant. E.g. RTP sequence and RTP timestamp of different substreams contain the same information, since the same codec is used for the descriptors (in case of MPEG-2, MPEG-4 and H.264). Of course, such part of compressed headers as UDP checksum will be different for each stream, since it is a packet specific. Thus, it can not be retrieved from the neighboring channels. But this information is only used to check whether the packet is received correctly. Since we do not aim to reconstruct the packet, but the compression base, we do not need this information. Therefore, in the special case of MDC streaming COHC can operate without sending any additional information, still being able to exploit channel diversity and relying on cooperation between different substreams. We call the proposed approach ”`zero-AIC`” solution.

III. PERFORMANCE EVALUATION

Let us consider MDC process that splits an information stream into J sub-streams. The proposed zero-AIC header compression will use COHC with the number of channels equal to J. This section gives a detailed analysis of the performance evaluation of zero-AIC approach in terms of decoding rate. Since this scheme exploits cooperation between the substreams, we have chosen to compare the zero-AIC approach with a combination delta coding + MDC, where the delta coding compression is applied independently to each of the MDC substreams. Furthermore, these schemes are compared with the case when no header compression is used.

As one of the important performance parameters we consider *decompression rate* that we define as the probability to receive and decompress a packet correctly. Thus, decompression rate reflects both the channel propagation conditions and the ability of the decompressor to keep the context synchronized with the compressor.

Let assume that the probability to lose a packet due to the channel conditions for the i-th sub-stream is p_i. Assume there are N packets in a frame. We find the expression for the decompression rate for each sub-stream and then consider it’s average over J streams. The resulting formula has a form:

$$DR_{delta} = \sum_{i=1}^{J} \frac{(1-p_i)\left(1-(1-p_i)^N\right)}{NJp_i} \tag{1}$$

Deriving the decompression rate for COHC, we obtain the following result:

$$DR_{COHC} = \frac{\left(1-\frac{1}{J}\sum_{i=1}^{J} p_i\right)\left(1-(1-\prod_{i=1}^{J} p_i)^N\right)}{N\prod_{i=1}^{J} p_i} \tag{2}$$

If we are sending uncompressed headers always, then the possible errors are only due to the channel errors. Therefore,

$$DR_{uncom} = \frac{1}{J}\sum_{i=1}^{J}(1-p_i) \tag{3}$$

Statement 1. For any $0 \le p_i \le 1$, $N \ge 1$, and $J \ge 1$ we have

$$DR_{uncom} \ge DR_{COHC} \ge DR_{delta} \tag{4}$$

These inequalities are intuitively clear from the physical formulation of the problem. Here we give the mathematical proof.

Proof. We can rewrite formula (1) in the following way:

$$DR_{delta} = \frac{1}{J}\sum_{i=1}^{J}(1-p_i)f(p_i,N) \tag{5}$$

where the function $f(p,N)$ is defined as

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

Furthermore, formula (2) takes the form

$$DR_{COHC} = (1-\bar{p})f(p_T,N) = \frac{1}{J}\sum_{i=1}^{J}(1-\bar{p})f(p_T,N) =$$

$$= \frac{1}{J}\sum_{i=1}^{J}(1-p_i)f(p_T,N) \tag{6}$$

where $\bar{p} = \frac{1}{J}\sum_{i=1}^{J} p_i$ is an arithmetic average of p_i and $p_T = \prod_{i=1}^{J} p_i$ is the geometric average to the power of N.

If no header compression is applied, the decompression rate (3) is

$$DR_{uncom} = \frac{1}{J}\sum_{i=1}^{J}(1-p_i)f(0,N) \tag{7}$$

One can notice that the only difference between formulas (5)-(7) is the arguments p_i, p_T or 1 in the function f. If we fix N, then $f(p,N)$ is monotonically decreasing as a function of p (see Figure 6).When $N \to \infty$, then $f \to f_\infty$:

$$f_\infty = \begin{cases} 1, p = 0 \\ 0, p \ne 0 \end{cases}$$

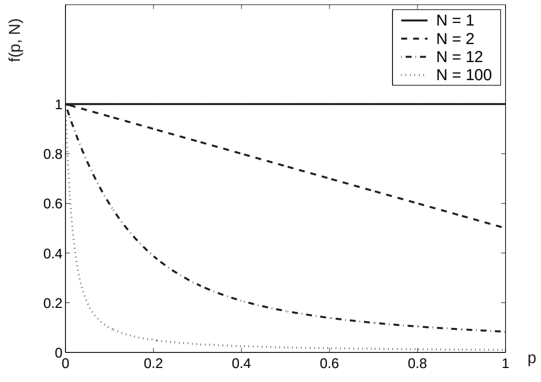

Fig. 6. Function $f(p,N)$ for different values of $N = 1, 2, 12, 100$

Taking into account that $(1-p_i) > 0$ and $f(p,N) > 0$ on the considered interval, from the observation that $0 \le p_T \le p_i$ for any i, the inequalities (4) follow. ■

Statement 2. If $p_i \ll 1$, then $DR_{COHC} = DR_{delta} + \ldots$ where the dots denote terms of order higher than two in p_i. In order words, for very small p_i, COHC performs close to delta coding and the advantage of the cooperative behavior is minimal.

Proof. When $p_i \ll 1$, we can represent the function f as power series in p_i:

$$f(p_i, N) = \frac{1 - (1 - Np_i + O(p_i))}{Np_i} = 1 + O(1) \qquad (8)$$

The same expression is obtained for $f(p_T, N)$. Substituting (8) into (5) and (6), the statement follows.

$\blacksquare$

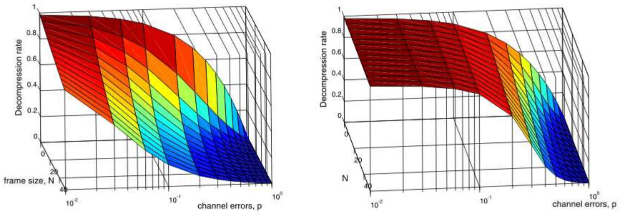

Fig. 7. Decompression rate as a function of N and p for delta coding scheme (left) and COHC scheme when $J = 3$ (right).

Consider a situation when all channels have the same error probabilities: $p_i = p$. Figure 7 present the decompression rate as a function of N and p for delta coding ($J = 1$) and COHC for $J = 3$. One can observe that for small values of p and small N the decompression rate of delta coding and COHC are approximately the same and close to 1. This example illustrates Statement 2. When p is larger than $3 \cdot 10^{-2}$ and $N > 10$, COHC shows significantly better performance even though only 3 cooperative streams are used. For $p < 10^{-1}$ the decompression rate of COHC is quite insensitive to the value of N, that is, we can increase the number of packets sent in a frame (and in this way increase compression gain) while keeping the high rate of decompression. This can not be done if independent compression schemes are applied to sub-streams. Figure 8 shows projection of the considered 3D-graphs when $N = 12$ and $J = 1, 3$ and 5. For comparison we present also the curve corresponding to the uncompressed header transmissions. The higher number of cooperative sub-streams are, the closer corresponding curve lies to the uncompressed curve. One can also notice a big gap between curves for $J = 1$ (delta coding) and $J > 1$. This advocates the usage of cooperation between different MDC sub-streams.

IV. CONCLUSION

In this paper we tried to answer the question how we can effectively reduce network overhead introduced by MDC stream splitting and thus benefit from MDC approach even in the case of bandwidth-limited links. Cooperative Header compression is shown to be a good choice for IP header compression strategy. If in general case COHC has to sent additional information, AICs, to prevent re-synchronization, for the multiple streams produced by MDC, the size of AIC can be reduced to zero. It is possible due to the redundancies in the header fields of different MDC substreams.

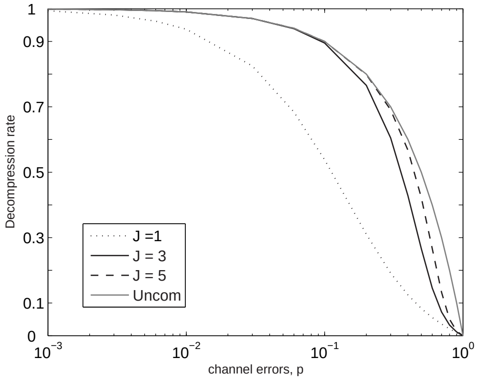

Fig. 8. Decompression rate as a function of p for $N = 12$ when $J = 1, 3, 5$ and when no header compression is applied.

V. ACKNOWLEDGEMENT

This work has been supported by SAMSUNG, Korea.

REFERENCES

[1] S. Frattasi, H. Fathi, F. Fitzek, M. Katz and R. Prasad. A New Pragmatic Methodology to Design 4G. In *International Conference on Networking (ICN)*, Reunion Island, 2005.

[2] J. M. Pereira, Fourth Generation: Now, it is Personal, In *Proceedings of the 11th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, London, UK, September 2000.

[3] M. Schaar and D. S. Turaga. Multiple Description Scalable Coding Using Wavelet-Based Motion Compensated Temporal Filtering. In *IEEE Int. Conf. on Image Processing*, Barcelona, Spain, September 2003.

[4] F.H.P. Fitzek, H. Yomo, P. Popovski, R. Prasad and M. Katz. Source Descriptor Selection Schemes for Multiple Description Coded Services in 4G Wireless Communication Systems. In *The First IEEE International Workshop on Multimedia Systems and Networking (WMSN05)*, Phoenix, USA, 2005.

[5] J.G. Apostolopoulos, S.J. Wee, Unbalanced multiple description video communication using path diversity, In *Proceedings of International Conference on Image Processing*, October 2001.

[6] F.H.P. Fitzek, B. Can, R. Prasad and M. Katz. Overhead and Quality Measurements for Multiple Description Coding for Video Services. In *Proceedings ofWireless Personal Multimedia Communications (WPMC)* Padova, Italy, 2004.

[7] 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.

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

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

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

[11] 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.