

Novel IP Header Compression Technique for Wireless Technologies with Fixed Link Layer Packet Types

Tatiana K. Madsen, Frank H.P. Fitzek, Shekar Nethi, Thomas Arildsen and Gian Paolo Perrucci
Department of Communications Technology, Aalborg University
Niels Jernes Vej 12, 9220 Aalborg Øst, Denmark,
phone: +45 9635 8678, e-mail: *tatiana | ff | snet03 | tha | gpp @kom.aau.dk*

Abstract—This paper focuses on the header compression for wireless technologies with fixed link layer packet types. Exploiting specifics of packet transmission over air in this situation, a novel IP header compression scheme is developed. The novelty of the scheme consists in the unique *CONTEXT* repair mechanisms that ensured high robustness and at the same time does not degrade bandwidth savings. After the general concept of the novel scheme is introduced, it is illustrated by an example of Bluetooth technology. By means of extensive simulation study, we show the performance of the introduced approach. Simulation results verify the robustness of the novel scheme and its ability to deal with characteristics of the wireless channel.

I. INTRODUCTION

Wireless multimedia services are set to play a much greater role in next generation mobile communication systems. With the increasing amount and complexity of services, the requirements for transmission bandwidth are becoming much higher than for the traditional voice services in circuit-switched networks. In order to use the limited resource, bandwidth, in the most efficient way, both IP packet header and packet payload should be compressed. A number of IP header compression schemes have been proposed over the past 15 years, e.g. Van Jacobsen HC (RFC 1114) [1], Compressed RTP (RFC 2508) [2], Enhanced Compressed RTP (RFC 3545) [3], Robust Header Compression (RFC 3095) [4]. These techniques allow compression of a 40-byte IPv4 header down to 2-4 bytes. All these techniques apply compression of an IP packet header based on *CONTEXT* state, but without taking into account the requirements of the link layer. A number of wireless technologies available today use fixed packet sizes at the link level: e.g. GPRS, UMTS, and Bluetooth. Due to the specifics of data packetization implemented in the above-mentioned technologies, IP header compression schemes do not perform efficiently compared to technologies where payload of the link layer packet can vary freely, such as in WLAN IEEE 802.11 standard. Speaking of header compression for packets with variable length, a useful parameter to consider is *bandwidth savings*. In this case, bandwidth savings are calculated as the ratio of the amount of data transmitted using a header compression scheme to the data required to send the same amount of information without applying header compression. Even a small compression of a header leads to some savings. This is not the case when packets of fixed sizes are used on

the link layer. In this situation, the communication is often based on some kind of time division access method: The channel is slotted and a packet transmission can start only at the beginning of the slot. Even if the packet length is smaller than maximally allowed by this packet type, no other packets can be sent in this slot and the bandwidth is wasted. Therefore, if by applying header compression, the size of a packet can not be reduced significantly (such that it will fit in another packet type), no bandwidth savings can be achieved. This problem of header compression inefficiency for fixed packet types has not been addressed in the literature. The only attempt in this direction was made in Zero-byte ROHC [5] that is an additional profile for Robust Header Compression (ROHC). Zero-byte header compression is designed to prevent the single-octet ROHC-header from pushing a packet voice stream into the next higher fixed packet size for the radio. It is designed especially for usage in GPRS air interface. Here, we introduce a novel header compression mechanism that is suitable for situations when the link layer payload size can take values only from a predefined set, that is, when several link layer packet types are supported.

II. HEADER COMPRESSION BASICS

Before we introduce the novel IP header compression concept, we would like to provide insights in the main concept and terminology of header compression (HC) research. Header compression is possible due to some redundancy among the different header fields of different protocol layers and the interdependencies of IP packets. It is typically performed on the headers of the network layer and above. Analysis of the variations in 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. On the sender side, the compressor removes redundancy from the incoming packet using information from the past packets, called the *CONTEXT*. The decompressor maintains the context and uses it to reconstruct the header of the incoming packets.

Compression gain is usually defined as the average compressed header length divided by the uncompressed header size. One should note, though, that this parameter can not be directly related to the *bandwidth savings*, since it does not take into account the cost of the context initialization and

maintenance. Inconsistencies in the contexts of the compressor and the decompressor lead to loss of synchronization and failure of the decompression procedure. To prevent the context re-synchronization, a context repair mechanism should be applied. Under an acknowledgement-based approach, a sender relies on feedback from a receiver to know when to transmit a packet with an uncompressed header. When feedback is unavailable, the synchronization of the context is achieved by periodic refreshes of the states (optimistic approach).

The problem of context updates is crucial for the efficient design of header compression schemes. It was recognized already in the earliest work on header compression. The proposed existing solutions include either excessive signalling from the receiver to the sender or periodic full context updates. Both methods decrease bandwidth savings. Thus, there is a clear trade-off between the compression gain and robustness of the scheme. We propose a novel scheme that allows to keep the context synchronization, thus ensuring robustness. At the same time, the compression gain of the proposed scheme is the same as for conventional header compression schemes.

III. PROPOSED HEADER COMPRESSION SCHEME

A. General description of the scheme

The main idea of the proposed header compression scheme is to switch the compression mechanism on/off adaptively depending on the size of IP packet payload. The compression mechanism will be enabled if by employing HC, a smaller packet size can be used, leading to bandwidth savings. If for sending a packet with compressed and uncompressed header, the same link layer packet type would be used, then the HC mechanism is disabled and the packet is transmitted with a full header. This helps to keep the synchronization between the compressor and decompressor without introducing any additional costs. The following pseudo-code summarizes the novel HC algorithm (U and C are the sizes of uncompressed and compressed headers in bytes, respectively):

- 1) Stream \mathcal{L} of packets
- 2) Take next packet p_i from \mathcal{L} . Payload of p_i is x bytes.
- 3) **If** packet type $[U+x] =$ packet type $[C+x]$,
then
 HC = OFF
else
 HC = ON
- 4) **If** $\mathcal{L} \neq \emptyset$, **then** Goto 2.

The presented scheme can also be explained with the help of a Markov model for the compressor states. In conventional HC schemes, transitions between the states are done either periodically or based on the feedback from the receiver. In the proposed novel scheme, an additional parameter, namely the payload size, is taken into account in order to make a decision about the transition to another state. A decision is made individually for each packet.

The proposed scheme is targeted at the technologies using fixed packet sizes on the link layer, such as GPRS, UMTS for cellular systems and Bluetooth for short range communication. As it is specified in Bluetooth, GPRS or UMTS, a user can access a channel only in an allocated slot and regardless of the size of the data a user has to transmit, the same allocated packet size must be used. Based on this observation, one can see that switching off the header compression does not affect the bandwidth occupancy, assuming that the data will still fit in the allocated slot. At the same time a significant increase in robustness can be achieved. Indeed, every time a full header is sent, the context at the decompressor side is updated. Such proactive maintenance of the context synchronization comes at no cost. Additionally, since no feedback signalling is required, the proposed scheme provides a robust method for compression in case the receiver feedback is unavailable. The novel method can significantly improve the performance of the differential encoding that in the case of wireless environments suffers from the loss propagation problem.

The following properties characterize the novel approach:

- i) The proposed header compression scheme is designed for fixed packet sizes.
- ii) The proposed novel header compression method reduces probability of context re-synchronization compared to the conventional compression schemes.
- iii) It maintains synchronization between the compressor and decompressor without the need of excessive signaling.
- iv) The proposed novel header compression scheme achieves the same bandwidth occupancy as conventional schemes.
- v) The header compression can be tuned such that only a subset of available packet types is used.

The last property is advantageous in the situations when only a subset of available packet types is suitable for data transmission. For example, under bad channel conditions, packets with error protection should be preferred. In this case, by switching header compression on or off for a particular IP packet payload size, we can achieve using only a pre-defined subset of link layer packets.

B. Adaptive Header Compression for Bluetooth

We further illustrate the novel header compression approach by considering Bluetooth technology as an example. The proposed scheme is referred to as *Adaptive Header Compression for Bluetooth*. Bluetooth is a low-cost low-power wireless technology that provides connectivity among devices located within short range. Bluetooth uses controlled scheduled transmissions. The Bluetooth system provides duplex transmission based on slotted time-division duplex (TDD), where the duration of each slot is $625 \mu s$. Asynchronous links support payloads with or without a 2/3-rate FEC coding scheme. In addition, single-slot, three-slot and five-slot packets are available. The Bluetooth packet types are summarized in Table I [6]. From the table one can notice the big difference between user payload of 1-, 3-, and 5-slot packets. The payload length is variable and depends on the available user data. However, the maximum length is specified for each

packet type. The whole slot (or 3 or 5 slots) is reserved for a transmission, even if the packet length is smaller than maximally available.

Type	Payload Header (bytes)	User payload (bytes)	FEC
DM1	1	0-17	YES
DH1	1	0-27	NO
DM3	2	0-121	YES
DH3	2	0-183	NO
DM5	2	0-224	YES
DH5	2	0-339	NO

TABLE I
BLUETOOTH PACKET TYPES

Following the proposed approach, we compress IP headers or not depending on the payload size. Table II shows when to apply header compression (assuming that an uncompressed header is 40 bytes, a compressed header 4 bytes). One should note that if the total size of payload and network and transport headers exceed 339 bytes, segmentation is required.

Payload [Byte]	Header Compression	Packet Type	FEC	Pkt Type Number
0-13	ON	DM1	YES	1
14-23	ON	DH1	NO	2
24-81	OFF	DM3	YES	3
82-117	ON	DM3	YES	4
118-143	OFF	DH3	NO	5
144-179	ON	DH3	NO	6
180-184	OFF	DM5	YES	7
185-299	OFF	DH5	NO	8

TABLE II
PROPOSED ADAPTIVE SCHEME: SWITCHING HC MECHANISM ON/ OFF DEPENDING ON THE PAYLOAD SIZE.

There can occur situations when the link layer specifies that only a subset of available packet types should be used for data transmission. For example, in bad channel conditions, DM packets should be preferred since they have 2/3 FEC protection. In this case the operation of the proposed header compression can be tuned such that automatically, only DM packets are used. The operation of the adaptive scheme is presented in Table III.

Payload (Bytes)	Header Compression	Packet Type	FEC
0-13	ON	DM1	YES
14-81	OFF	DM3	YES
82-117	ON	DM3	YES
118-184	OFF	DM5	YES
185-220	ON	DM5	YES

TABLE III
PROPOSED ADAPTIVE SCHEME: SWITCHING HC MECHANISM ON/ OFF DEPENDING ON THE PAYLOAD SIZE.

IV. PERFORMANCE EVALUATION OF ADAPTIVE HEADER COMPRESSION

A. Simulator and Performance metrics

To evaluate the performance of Adaptive Header Compression scheme, a simulator is developed in Matlab. The simulation model consists of three main parts: i) data and

header generator and header compressor/ decompressor; ii) Bluetooth baseband and iii) channel. Channel is modeled as Additive White Gaussian Noise (AWGN) channel. The header compressor works on the RTP/UDP/IP headers. The payload size of a packet stream is assumed to be i.i.d. over the interval [1, 299] bytes, which is more or less the range of packet video communication with state of the art video compressors. The performance of Adaptive Header Compression is compared to one of the conventional schemes, RFC 2508 [2]. The results when no header compression is applied for data transmission are also presented.

A number of performance metrics has been evaluated. Mainly, we have focused on the metrics that reflect bandwidth savings and robustness. Additionally, parameters that characterize energy-efficiency of the scheme are studied since the minimization of energy consumption is a desired property for small battery-driven devices.

- **Packet Delivery Ratio (PDR):** It is defined as the ratio of the number of successfully decompressed packets to the total number of packets sent. This metric accounts for both packet errors caused by bad channel conditions and failures of the decompression procedure. It reflects robustness of a compression approach.
- **Bandwidth savings:** This metric gives savings of the bandwidth, measured in slots, normalized to the number of slots needed for data transmission if no header compression is applied.
- **Energy Efficiency (Bytes/mJ):** It is the ratio of the user data transmitted over the amount of energy spent. We note that for the energy we only consider the part spent in the RF part and we do not consider the energy needed to compress and decompress packets as this impact depends on the platform where the actual coding is performed. Typical values considered in the simulation for energy usage for transmission, reception, in idle and in in low power modes are seen in table IV [7]. In slots with no transmission (e.g. odd slots in a broadcasting scenario), a node is assumed to be in low power mode.

Power Type	mill Watt
P_{TX}	100
P_{RX}	78
P_{idle}	0.486
P_{lpm}	0.054

TABLE IV
TYPICAL POWER VALUES FOR DIFFERENT STATES

B. Scenarios

Two conceptually different scenarios are considered.

1) *One-to-many communication / No feedback:* The first scenario presents broadcasting/multicasting by a Bluetooth access point (that is assumed to have the role of piconet master) to a number of devices (that have the role of piconet slaves). The master transmits data in even slots using 1-, 3- or 5-slot packets. The slots that will follow after a master transmission (odd slots) are empty: none of the slaves are allowed to transmit in these slots. Thus, the master transmission is left

unacknowledged. We refer to this case as ‘no feedback is available from the receiver’. The context updates consist of periodic transmission of full packet headers.

2) *One-to-one communication / Link layer assisted feedback*: The second scenario focuses on one-to-one communication between two Bluetooth devices. Bluetooth uses stop-and-wait ARQ scheme: DM or DH packets are retransmitted until acknowledgement of a successful reception is returned or timeout is exceeded. The acknowledgement information is included in the header of a return packet. The slave will respond in the first slot following master-to-slave transmission. In the simulation model the number of retransmissions is limited to 5.

Thanks to the ARQ scheme, feedback about the successful transmission of a packet is available at the next slot at the link-layer. Exploiting a cross-layer approach, this information can be passed on to the network layer and to the header compressor. The advantage of the described cross-layer optimization is that the feedback can be delivered to the compressor without the need for additional signalling but purely relying on the cross-layer information exchange. We refer to this situation as ‘link-layer assisted feedback’. For the simulation purposes, we assume that there is no processing delay and if a packet cannot be delivered correctly, already the next IP packet of the stream is sent with an uncompressed header (this allows immediate context re-synchronization). However, if the processing delay is large, the decompressor will drop a number of packets before the update will be received.

The presence of the immediate feedback illuminates the loss propagation problem. In this case the proposed Adaptive Header Compression and RFC 2508 schemes show approximately the same performance. Instead, for our simulation evaluation we have chosen to focus on the case when only a subset of available packet types is used for data transmission. This approach can also be motivated by the following observation: the error-proneness of a header compression scheme depends on the type of link layer packets used for transmission. This is due to the different packet lengths and different error protection schemes used. Figure 1 shows the ‘robustness’ when different Bluetooth packet types are used (for abbreviation see Table II). From Figure 1 one can see that in cases when DM packets are used for data transmission (packet types 1,3,4 and 7), a significantly higher ratio of packets are received and decompressed correctly compared to DH packets. This motivates the proposed header compression approach that works exclusively with DM packets, as explained in Section III-B (see also Table III). It can be anticipated that this scheme gives a performance improvement compared to RFC 2508 when channel conditions are not good. If such channel conditions are detected by a link layer (e.g. bit errors exceed a certain threshold), a request to the compressor to switch to the new scheme can be sent.

V. RESULTS

All considered parameters are plotted versus signal-to-noise ratio (SNR).

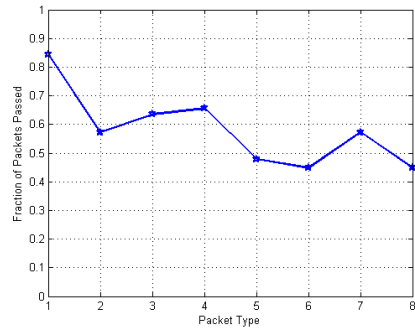


Fig. 1. Fraction of packets received and decompressed correctly for different packet types (PDR).

A. Simulation Results for Scenario 1

Let the periodic context update be sent once every N packets. When N is small, header compression schemes perform very similar to the case of no compression. We have observed bandwidth savings of about 8% for $N = 2$. Figures 4–6 are plotted for $N = 20$. AHC and RFC 2508 achieve the same bandwidth savings of 16% (Fig. 3). Fig. 2 shows a rapid drop in Packet Delivery Ratio when channel conditions degrade. However, AHC is more robust towards channel errors compared to RFC 2508. Speaking of energy efficiency, we observe that under good channel conditions ($SNR > 22$), RFC 2508 is the most energy efficient scheme (Fig. 4). This is due to the high compression gain coupled with no packet losses. But as channel conditions become worse ($SNR < 20$), the efficiency of RFC 2508 deteriorates quickly and already when $SNR = 18$, it falls down to zero. This is due to severe loss propagation problem from which RFC 2508 suffers when $SNR < 18$. On the contrary, AHC still shows high energy efficiency when the channel is not good.

For large N , the context update is done less frequently and the performance of RFC 2508 degrades rapidly. However, the value of N does not have any significant impact on the performance of AHC. This phenomenon can be explained by an efficient context update method of AHC.

Overall, we can conclude for this scenario that for any N AHC is a robust scheme. It is able to keep high PDR without losing in bandwidth savings. On average it consumes slightly more energy compared with the conventional compression schemes, but the overall energy efficiency of AHC is higher when $SNR < 20$.

B. Simulation Results for Scenario 2

We kept the periodic context updates activated, but the choice of N did not have influence on the compression performance: the majority of updates are triggered by the receiver feedback. To underline that only DM packets are used in combination with the AHC scheme, we refer to this scheme as AHC-DM. As expected, AHC-DM achieves the highest PDR, even higher than no HC case (Fig. 5). This is due to the strong error protection of DM packets. The more packets are received correctly, the easier it is to maintain synchronization.

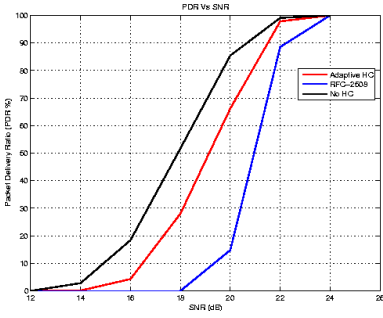


Fig. 2. Scenario 1: Packet Delivery ratio

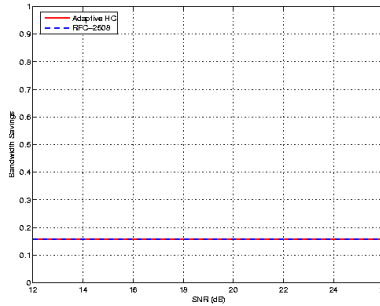


Fig. 3. Scenario 1: Bandwidth Savings

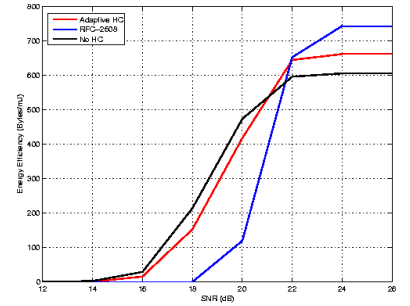


Fig. 4. Scenario 1: Energy Efficiency

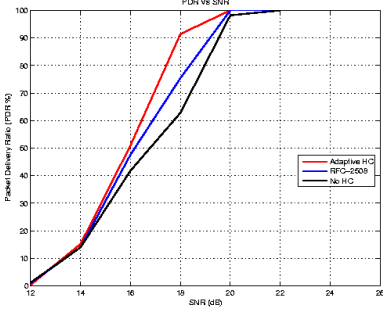


Fig. 5. Scenario 2: Packet Delivery ratio

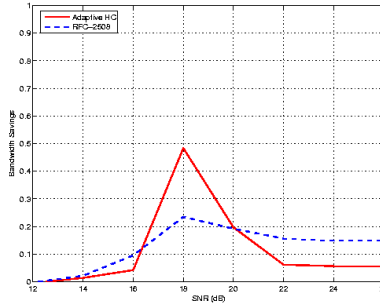


Fig. 6. Scenario 2: Bandwidth Savings

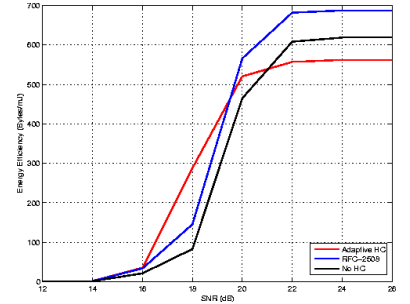


Fig. 7. Scenario 2: Energy Efficiency

Compared to Scenario 1, where the bandwidth savings are constant regardless of SNR, in Scenario 2 the bandwidth saving is not a constant function of SNR (Fig. 6) due to possible packet retransmissions. Under good channel conditions ($SNR > 20$), the bandwidth savings of RFC 2508 are the same as in Scenario 1, 16%, since there is almost no retransmissions in this case. However, for AHC-DM bandwidth savings are lower, about 6%, due to the excessive error protection of DM packets. But when $SNR < 20$, almost 30% of bandwidth can be saved using AHC-DM compared to no HC case. When SNR decreases further, no savings can be achieved. RFC 2508 is the most energy efficient under good channel conditions, but as channel becomes worse, AHC-DM shows the best efficiency and on average it uses less power for packet transmission, since the number of retransmissions is smaller compared to other schemes (Fig. 7). Overall, AHC-DM should be preferred when $SNR < 20$.

VI. CONCLUSIONS

We have introduced a novel header compression scheme that is suitable for usage with wireless technologies operating with fixed link layer packet types. The usage of the general approach is illustrated with an example of Bluetooth technology. The statements on the properties of the proposed scheme are verified by extensive simulations. Two different scenarios are considered: i) broadcasting of data by a Bluetooth access point and ii) data exchange between two Bluetooth devices. The robustness of the scheme is shown for both scenarios. For the first scenario the proposed scheme performs better than the header compression scheme introduced in RFC2508 in terms of packet loss. They perform equally well for the

bandwidth saving. The uncompressed has no bandwidth saving by definition, but lower packet losses than both compression schemes. For the second scenario, the proposed scheme shows better performance than the uncompressed and the RFC2508 with large bandwidth savings, that are only outperformed by RFC2508 for very good channel conditions. To sum up, we have found a novel header compression that offers robustness and compression gain at the same time, at the same complexity as the RFC2508.

ACKNOWLEDGMENT

This work was partially financed by the X3MP project granted by Danish Ministry of Science, Technology and Innovation.

REFERENCES

- [1] V. Jacobson, Compressing TCP/IP Headers for Low-Speed Serial Links, Request for Comments 1144, February 1990.
- [2] S. Casner and V. Jacobson, Compressing IP/UDP/RTP Headers for Low-Speed Serial Links, Request for Comments 2508, February 1999.
- [3] T. Koren and et al, Enhanced Compressed RTP (ECRTP) for links with high delay, packet loss and reordering, Request for Comments 3545, July 2003.
- [4] 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.
- [5] S. Z. Liu, K. Le, Zero-byte support for Bidirectional Reliable mode in Extended Link-layer Assisted Robust Header Compression Profile, Request for Comments 3408, December 2002.
- [6] Bluetooth specification. Version 2.0. www.bluetooth.org
- [7] G. Kuijpers, P. Popovski, H. Yomo, T.K. Madsen, and R. Prasad, Flexible and Energy-Efficient Networking Structure using Bluetooth Park Mode. In *Wireless Personal Multimedia Communications WPMC 2004*, Padova, Italy, September 2004