R-RoHC: a single adaptive solution for header compression

R. Fracchia[a], C. Gomez[a] and A. Tripodi[b]
[a]Thales Communications, 160 boulevard de Valmy, 92740 Colombes, France
[b]Orange Labs, 38 rue du Gal Leclerc, 92130 Issy-Les-Moulineaux, France (previously in Thales Communications)

***Abstract*— In point to multi-point wireless transmissions, header compression protocols like RoHC (Robust Header Compression) can degrade the communication when used on fading channels and, at the same time, introduce long periods of silence, due to decompression problems at the receiver side. This paper presents R-RoHC, a reactive header compression mechanism designed to cope with RoHC misbehavior in case of point to multi-point multimedia streaming over wireless networks. We propose to exploit cross-layer information to trigger a recovery procedure in critical decompression conditions: R-RoHC sends few extra uncompressed packets whenever the static context (i.e., uncompressed and not changing part of the headers) is missing at the receiver side, thus allowing a successful decompression without requiring any modifications of the RoHC state machine and its timeouts.**

I. Introduction

Video streaming, after having flooded the world of fixed communications, is now landing to 3G wireless networks as testified by the every day increasing number of applications to share multimedia content with smart-phones. Moreover, point to multi-point video distribution is increasing in popularity for the diffusion of IPTV or live content in social networks: multicast transmissions to wireless clients are expected to become more and more attractive.

When considering multimedia streaming applications over wireless networks, the problem of bandwidth cost becomes crucial. Indeed, network packetization and transport (e.g., RTP, UDP, IP) add non negligible protocol overhead to the useful video data, being an issue on bandwidth-constrained wireless links. An interesting approach to cope with this problem is to process network headers in order to decrease their inner redundancy. The bandwidth saved in this way can be devoted to the video source bit rate, allowing the transmission of a video with an higher resolution and, as a consequence, a better end-to-end final video quality.

The overhead introduced by the IP protocol has been early identified as a problem on constrained bandwidth links and first solutions appeared in 1984 with the Thinwire protocol [1] and in 1990 with a solution proposed by Van Jacobson [2] concerning TCP/IPv4 compression mechanisms based on information redundancy. Several different solutions followed, including CTCP [3] compression, handling several IPv6 streams, and CRTP [4] extending the previous protocol for RTP header compression. Finally, Robust Header Compression has been standardized in RFC 3095 [5]. Protocols for header compression have been recently considered for inclusion in new wireless communication standards like LTE and WiMAX.

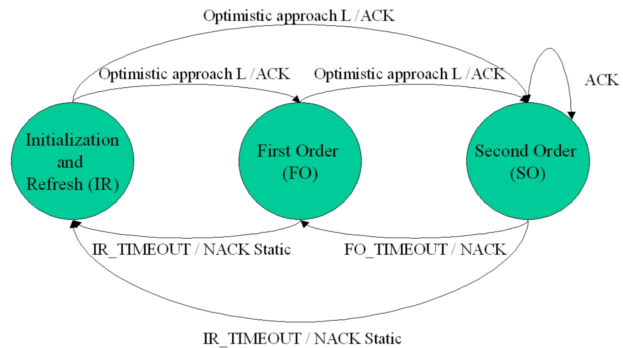

Fig. 1. RoHC state machine

Unfortunately, as explained in Section III, existing header compression solutions require long initialization times in point to multi-point communications. Moreover, in multicast or broadcast transmissions, bad channel conditions could negatively affect header compression performance thus reducing the received throughput. These two problems dramatically limit header compression utilization for next generation services on wireless networks.

We propose in this paper a solution that, exploiting cross-layer information, determines critical RoHC misbehavior and quickly reacts, without significantly increasing the average header size.

The rest of the paper is organized as follows. The RoHC protocol is presented in Section II of this document. Section III overviews RoHC behavior in case of multicast/broadcast transmissions, while Section IV presents the details of the Reactive-RoHC (R-RoHC) approach. Section V reports the performance evaluation of the proposed scheme and Section VI concludes this work.

II. RoHC behavior

The different fields composing the packet header are classified by RoHC into static and dynamic fields: static fields are those not varying during the transmission, as the port numbers and IP addresses, while dynamic ones are the fields varying for every packet, as the sequence number and the timestamp. The compressor state machine, presented in Figure 1, is composed by three states working on these static and dynamic fields:

namely the Initialization and Refresh (IR), the First Order (FO) and the Second Order (SO) states. In the Initialization and Refresh state, packets are sent with an uncompressed header and 3 bytes carrying a context identifier are additionally included in the header: the decompressor stores the received information and associates it to the received identifier, thus creating the context to use for a correct decompression of the following packet headers. In the First Order state, the static fields of the header are no longer transmitted, while the context identifier and the dynamic fields are sent uncompressed. In the Second Order state, a compressed version of the dynamic fields is transmitted together with the dynamic fields that changed, resulting in an even smaller header. Furthermore, RoHC specifies three possible modes of operation: Unidirectional (U), Bidirectional Optimistic (O) and Reliable (R). The difference between the three modes is related to the events controlling the state transitions, as depicted in Figure 1. In the U mode, the compressor sends L packets from each state before moving toward a more compressed state, while timeouts are used to move backward: the FO_TIMEOUT is used to go back from the SO to the FO state and the IR_TIMEOUT to go back from the FO or the SO state to the IR's one. It follows that, for high values of L, the compressor stays longer in the less compressed states, thus resulting in a bigger average header size. On the contrary, for high values of the timeouts, the compressor stays longer in the more compressed states, with the consequence of a smaller average header size but an higher probability of context errors, which can not be corrected until the following context update (done in the IR and in the FO states). When adopting the Bidirectional Optimistic mode, the compressor moves forward as in the U mode and backward when receiving NACKs. Finally, in the R mode the decompressor guides the compressor state transition with the use of acknowledgment (ACKs) and negative acknowledgments (NACKs and Static NACKs, depending on the decompressor state). Defaults values of IR_TIMEOUT, FO_TIMEOUT and L are usually 1000, 100 and 5 packets respectively [6].

III. RoHC FOR POINT TO MULTI-POINT TRANSMISSIONS

In broadcast and multicast transmissions two main issues related to header compression may affect the video distribution over wireless mobile networks. Indeed, the header compression of a packet addressed to several users (i.e., a multicast or a broadcast packet) introduces some peculiar aspects, detailed in the following.

The main problem is due to the compressor state machine and to its state transitions. When working in R mode, RoHC compressor waits for ACKs and NACKs sent by all the clients. The transmission of ACKs by all the receivers introduces in this case an important overhead, using a lot of bandwidth and thus going against the compression principle. Moreover, since the compressor does not know how many terminals are on the multicast list, it can not know how many ACKs it has to wait before changing the compression state. In the O operational mode, instead, a NACK reception triggers the return to a less compressed header generation. Moreover, the transmission of the acknowledgment introduces a latency in the reaction, which increases with the number of retransmissions. Finally, since the different clients may experience different channel conditions, only one bad channel out of many good ones has the power to disable the header compression. In the worst case, the compressor might not move from the IR state, penalizing all the receivers.

It follows that, in case of multicast transmissions, RoHC has to be used in the Unidirectional operation mode. A similar solution has been also adopted by several protocols at different layers of the protocol stack when dealing with multicast transmissions, thus confirming the validity of this choice. As an example, the IEEE 802.11 standard disables the ACK transmission and retransmission of lost packets in case of multicast and broadcast communications.

When working in Unidirectional compression mode, long bursts of errors may affect the communication and uncompressed packets, which are necessary for the static context creation and the following decompression, can be lost or discarded at the receiver side. When all the first uncompressed packets are lost, the static context can not be created and RoHC decompressor is not able to regenerate the compressed header of received compressed packets that are, as a consequence, discarded. Only a successful reception of the IR packets could initiate a successful decompression phase and several seconds of the multimedia stream can be lost in this way.

A fine tuning of RoHC compression parameters (i.e., L, IR_TIMEOUT and FO_TIMEOUT) can reduce the decompression problems at the receiver side in the Unidirectional mode, thus allowing RoHC use without increased losses. However, a reduction of the IR_TIMEOUT would increase the frequency of IR packet transmission by increasing the time spent in IR and FO states. This tuning goes to the detriment of the average header size, thus reducing the advantages introduced by RoHC. Furthermore, the best-suited values of RoHC parameters dramatically change from channel to channel: it follows that, in order to dynamically identify the appropriate values, accurate channel estimations would be required.

Finally, in broadcast and multicast communications, the users start the streaming at different times. Assuming that the compression function is applied at the Base Station, the compressor enters in the IR state as the first video packet is transmitted in that cell. When a new user in the same cell requests the same stream, the compressor may be in the First or Second Order states. In that case, the new joiner won't be able to reconstruct the whole header until the full context is received (i.e., when the compressor goes back to the IR state). The new joiner will thus lose packets for a time period uniformly distributed between 0 and IR_TIMEOUT- L (i.e., the maximum time without static context). Considering the defaults values of IR_TIMEOUT and L (i.e., 1000 and 5 packets respectively), new users would lose on average 500 packets before being able to decompress the packet header. A single solution to cope with this problem appeared in the literature [7]: authors of this work proposed different techniques: i) to send one or two uncompressed packets every

FO_TIMEOUT; ii) to send one or two uncompressed packets in a uniformly distributed period; iii) to go directly back to the IR state at the FO_TIMEOUT expiration. Other proposals in the same paper were to correlate header compression to the video structure, for example by sending one uncompressed header for every packet with the RTP Marker Bit equal to 1 (i.e., before a new image) or to send an extra uncompressed header non associated to any payload before any new image. All these proposals have to be applied for any transmission, unicast or multicast, with good or bad channel conditions. Indeed, the the compressor has not the possibility to know in which conditions it is working. It follows that these actions may be useless, reducing only the effectiveness of the compression.

IV. Reactive-RoHC

The approach presented in this work avoids throughput reductions due to decompression problems and can be triggered only when needed. Moreover, the proposed Reactive Robust Header Compression scheme (R-RoHC) works for both unicast and multicast transmissions and does not require an accurate channel estimation.

As explained above, decompression problems are due to missing or corrupted reception of IR packets. The idea is thus to send few IR packets only when needed, i.e., when the static context is missing at the receiver side, without modifying RoHC state machine and its timeouts. We introduce a context recovery event that triggers the additional transmission of IR packets from any compression state (i.e., IR, FO or SO). After the generation of these IR packets, the compressor restarts the generation of header types corresponding to its actual state, without needing to go back to the IR state.

Two different schemes are considered and evaluated by simulation: i) the transmission of a fixed number of IR packets and ii) the transmission of a varying number of IR packets. In the second case, the number of uncompressed packets is increased with the number of static context recovery attempts. Indeed, we propose to send one IR packet at the first recovery attempt, two IR packets the second time, and so on, in order to increase the probability to receive a correct IR packet while limiting the overhead.

This mechanism, difficult to implement without knowledge of the decompressor status, can be easily introduced in a system architecture offering end-to-end cross-layer signaling. As an example, the triggering engine presented in [8] allows the exchange of information (i.e., triggers) between a trigger source and a trigger consumer. We select as trigger source the RoHC module at the client side and as consumer the RoHC module at the Base Station (i.e., at RoHC compression side). We moreover select as trigger of the static context recovery the number of packets transmitted by the RoHC module to the upper layer. If this number is equal to zero, all the packets are discarded between the source and the destination. Besides link failure, a possible reason is the unavailability of the static context at the RoHC decompression module, with the consequent discarding of all the received packets. It has to be noted that the triggering engine allows the subscription to specific trigger values via the use of filters: the RoHC module at the Base Station can thus subscribe to a value equal to zero for the selected metric. The aim of this filtered subscription is twofold: first, a minimum overhead introduced in the network, since the trigger is transmitted only when really needed; second, a fast reaction of the compressor. Indeed, since the trigger is transmitted to the Base Station only when no packets exit from the RoHC decompressor, the evaluation of this metric can be frequently updated: we decided to compute it every 100ms.

It has to be pointed out that the same scheme can be successfully employed to solve RoHC initialization issues in multicast streaming. As explained above, the RoHC module at the Base Station enters in the IR state as the first video packet is transmitted in that cell and the new joiners are not able to reconstruct the whole header until compressor goes back to the IR state. In that period, zero packets are transferred from the RoHC decoder to the IP layer. By introducing the trigger described above and the R-RoHC protocol, after a short time (i.e., maximum 100 ms) the RoHC module at the client side sends a trigger to the Base Station thus signaling the connection of a new user. At the reception of this trigger, the BS sends the IR packets, enabling in this way the video reception by the new joiner.

V. Simulation Results

In this section, we present the simulation results obtained with the simulator developed on the OMNeT++ framework [9] by the partners of the OPTIMIX project [10]. This tool simulates the transmission - bit by bit - of a precoded H.264/AVC or H.264/SVC video and accurately models all the OSI layers from the application to the physical layer.

Video streaming is started and controlled by RTSP which allows the end-user requesting the desired content (the reference CIF Foreman sequence @30 Hz in the simulation). When the streaming starts, the application layer at the source side transmits the images extracted from an H.264/AVC precoded video. At the client side, a robust decoder transforms the received H.264 frame into an uncompressed yuv video frame which is then displayed. Below the application layer, RTP fragments each image into packets and handles the image reconstruction at the receiver side. At the transport and network layers, the UDPLite transport protocol and IPv6 are used respectively. At the Data Link and Physical layer, an IEEE 802.11g transmission is simulated; data are transmitted with a rate of 6 Mbps, as specified by the standard for multicast and broadcast transmissions. A Rayleigh block fading channel is used to represent the effects of the radio channel on the wireless transmission.

A. R-RoHC recovery for fading channels

We consider first the streaming of a single video (the Foreman reference sequence with a bit rate of 440 kb/s) toward a single user and we study RoHC performance on a Non Line Of Sight scenario, with a mobile client moving at the speed of 2 or 3 Km/h.

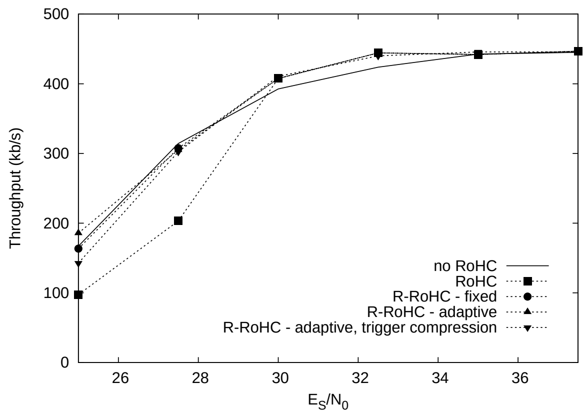

Fig. 2. Average throughput for RoHC and R-RoHC protocols in unicast transmissions

The results presented in the following are obtained by setting the standard deviation of the log-normal shadowing of the Rayleigh block fading channel to 0 and the coherence time to 0.2 s. At the physical layer we perform a 8PSK modulation. No losses are introduced on the wired network. We finally set RoHC parameters as follows: IR_TIMEOUT=1000, FO_TIMEOUT=100, L=5 packets.

In Figure 2, we compare the average throughput obtained at the transport layer (i.e., after RoHC decompression) as a function of the signal-to-noise ratio E_S/N_0 in the following cases: i) no header compression; ii) standard RoHC; iii) R-RoHC with the transmission of three IR packets; iv) adaptive R-RoHC and v) adaptive R-RoHC compression of both data and triggers. The figure shows that, for medium and high signal-to-noise ratio values, performance obtained when RoHC is used supersedes performance without RoHC. Indeed, since UDPLite or DCCP protocols are used at the transport layer, a checksum on the packet header only is computed: it follows that, with few errors, the shorter the header the lower the number of discarded packets and the higher the throughput (we assume to have at the application layer a robust video decoder able to conceal errors). However, as the signal-to-noise ratio decreases, when RoHC is used the achieved throughput is lower than the throughout obtained without compression for the reasons explained in the previous paragraphs. R-RoHC with a transmission of three IR packets brings a significant gain in the achieved performance by offering performance identical to the case without compression even in poor channel conditions. Performance further increases when the adaptive R-RoHC scheme is used, since IR packets are transmitted until the context is recovered. We can finally observe that performance remains very good even if decompression errors may affect triggers and thus feedbacks received by the Base Station.

Moreover, the adaptive R-RoHC mode generates higher throughput while minimizing the number of uncompressed transmitted packets. Header sizes are reported in Table I: we can observe that the average header size in the adaptive case

TABLE I
HEADER SIZE.

E_S/N_0	Uncompressed	RoHC	R-RoHC	Adaptive R-RoHC
32.5	60	6.3	7.55	7.48
30	60	6.3	7.55	7.51
25	60	6.3	7.55	7.62

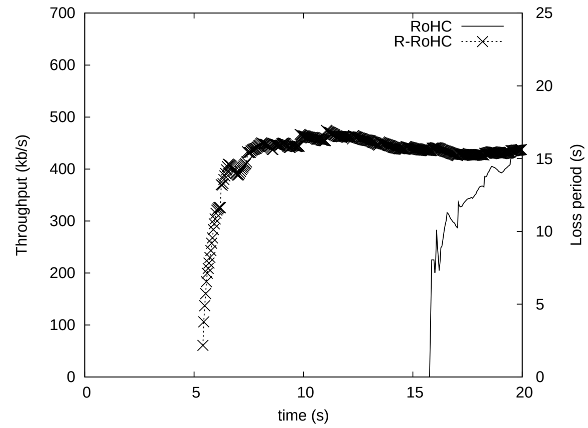

Fig. 3. Instantaneous throughput of the second client.

remains very low for any signal to noise ratio but correctly increases when channel conditions worsen, since more IR packets may be needed for context recovery. As an example, the average header size at E_S/N_0=25dB is 7.62 bytes in the adaptive case versus 7.55 bytes of the fixed case.

B. R-RoHC for multicast streaming

This section presents the results obtained with the adaptive R-RoHC algorithm in a multicast scenario, where two users request the same video stream via an RTSP request: the first user requests the stream at the first second of the simulation while the time of the second user's request is varying. The same channel model of the previous section, with a signal-to-noise ratio E_S/N_0=30dB, is used in the simulations.

Consider first the second user sending the RTSP request at time t=5 seconds. We plot in Figure 3 the instantaneous transport layer throughput of the second client. The compressor is already in the Second Order state when the second RTSP request is received. When standard RoHC is used, the compressor does not send any uncompressed packets until time 15.7 seconds. We can observe that all the packets received before the reception of the IR packets (i.e., between 5 s and 15.7 s) are discarded by the decompressor: video frames arrive at the decoder only starting from 15.7 s. Thanks to the R-RoHC solution, used here in the adaptive mode (i.e., with an increasing number of IR packet transmission), this idle time can be dramatically reduced: indeed, the BS sends an additional IR packet at the reception of a zero throughput trigger, generated every 100 ms, thus allowing a correct decompression and the video display with a negligible delay.

The reduction of the average loss period and the average achieved throughput can be observed in Figure 4 for both RoHC and R-RoHC solutions: we plot in the figure the loss

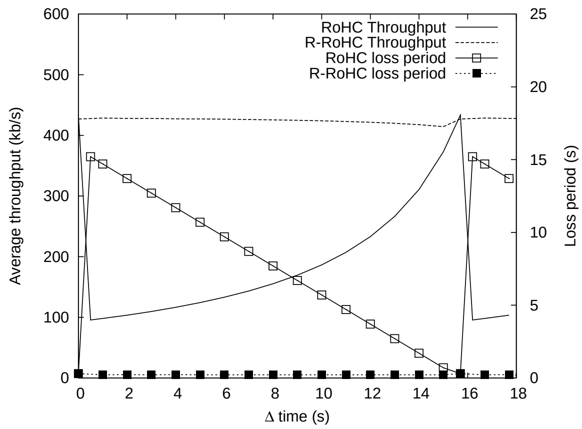

Fig. 4. Average throughput for different connection times of the second client

period (line with points) and the average throughput (lines) as a function of the second versus the first client connection time. When the two clients start the streaming at the same time (i.e., Δ=0) there is no loss of packets: indeed, both clients are connected to the BS when the compressor is in the IR state. The throughput corresponds to the video bit rate in both cases. When the second client requests the streaming immediately after the five IR packets transmission (e.g., Δ=500ms) and standard RoHC is used, the loss period is the maximum one, i.e., 15 s, corresponding to an important degradation in the throughput. The loss period is instead limited by the trigger frequency (i.e., 100 ms) in the R-RoHC case and the throughput remains close to 440 kbps. By further delaying the start time of the video streaming toward the second client, the loss period experienced using RoHC decreases since the waiting time for the following IR transmission is reduced, until reaching zero when the second client connection starts during the following IR packets transmission (i.e., at t=15.7 s). Loss period and achieved throughput are independent of the second client connection time with R-RoHC and close to 50 ms and 440 kbps respectively.

VI. Conclusion

This paper presented a reactive scheme for header compression which, coupled with an end-to-end feedback delivery, allows to recover static context unavailability at the receiver side, phenomenon that could be experienced by mobile devices receiving a point to multi-point data transmission. The proposed solution proved to be able to successfully avoid performance degradation related to RoHC with a minimal increase in the average header side.

Acknowledgements

The research leading to these results has received funding from the European Union's Seventh Framework Programme ([FP7/2007-2013]) under grant agreement no. 214625.

References

[1] Faber, D., et. Al. A Thinwire Protocol for connecting personal computers to the INTERNET, RFC 914, 1984.

[2] V. Jacobson, Compressing TCP/IP Headers, RFC 1144, 1990.

[3] Degermark, M., et. Al., IP Header Compression, IETF RFC 2507, February 1999.

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

[5] Bormann C., et Al., RObust Header Compression (ROHC): Framework and four profiles: RTP, UDP, ESP and uncompressed, IETF RFC 3095, 2001.

[6] ICT FP6 PHOENIX project, D3.4C, Final Specifications and Simulation Results of Data Link Protocols

[7] E. Martinez, A. Minaburo, L. Toutain, ROHC for Multicast Distribution Services, in Proc. of PIRMC05

[8] E. Piri, T. Sutinen and J. Vehkaper, Cross-layer Architecture for Adaptive Real-time Multimedia in Heterogeneous Network Environment, European Wireless 09 conference, Aalborg, Denmark, May 2009.

[9] http://www.omnetpp.org/

[10] http://www.ict-optimix.eu/index.php/OPTIMIXsim