Deployment of IPv6 ROHC profiles 1 and 2

Ana Carolina Minaburo Villar, Laurent Toutain, Loutfi Nuaymi

Deployment of IPv6 Robust Header Compression Profiles 1 and 2

Ana Minaburo*, Laurent Toutain*, Loutfi Nuaymi*

ENST-BRETAGNE, Campus Rennes, BP 78, Rennes, France

{anacarolina.minaburo, laurent.toutain, loutfi.nuaymi}@enst-bretagne.fr

Abstract

The use of interactive services in a low bandwidth link leads to a weak performance because of the large size of encapsulations. The performance could be increased using a header compression algorithm. The IETF ROHC working group has defined a new header compression algorithm used in low bandwidth links with high-level errors. This paper gives a description of the proposed infrastructure for ROHC and the differences between the profile 1 and profile 2 based profiles that are used. The results are based in our IPv6 ROHC implementation.

1. Introduction.

In the third generation of cellular networks, a header compression mechanism can reduce the transmission time and increment the use of scarce resources like the Radio link. The voice over IP services in a cellular network use the RTP (Real-Time Applications Transport Protocol) protocol. The header size in an IPv6/UDP/RTP packet is between 60 to 120 bytes and between 40 to 100 bytes in an IPv4/UDP/RTP packet. The payload is between 15 and 20 bytes using the compression voice algorithms and the real time constraints. The header compression is possible in different OSI/ISO layers but the mechanisms are more efficient in the IP layer where the packet format (and the upper layers) is known. The header compression algorithms eliminate the redundancy in the transmitted information. This work is based on the IETF (Internet Engineering Task Force) standardization results of the ROHC working group and particularly the RFC 3095 [1]. We have developed our implementation in a FreeBSD operating system. This platform is used to measure the response time, the transfer rate and the robustness of the system.

2. Header Compression Algorithms.

The purpose of header compression is to reduce the redundancy in the header information and the redundancy of the headers between consecutive packets. The information that does not change is sent during initialization or updated periodically. The other fields are encoded and a prediction or a dependency mechanism reduces the information transmitted.

The performance problem of IP protocol over a low bandwidth link has been studied since 1990 when Van Jacobson [2] proposed a mechanism based on header redundancy information, which compressed the TCP/IPv4 header. There are other proposals for different protocol headers based on redundancy, such as the CTCP (IP Header Compression) header mechanism [3], and the CRTP (Compressing IP/UDP/RTP Headers for Low-Speed Serial Links) header mechanism [4]. CTCP is used to reduce TCP/IP header flows and CRTP reduces RTP/UDP/IP header flows.

Mechanisms that add a robust and efficient compression scheme are needed for low bandwidth links with a high error rate, such as radio links. Many contributions have been proposed, like adapted header compression for real time multimedia application (ACE), header compression using keyword packets, or header compression based in Checksum (ROCCO). These last three contributions are the basis for the header compression standard ROHC (Robust Header Compression) described in RFC 3095 [1].

3. ROHC protocol.

ROHC header compression algorithm was conceived to reduce the header in an IP packet to be sent through a scarce radio link. The IETF forecasts the transmission of voice and video applications into the cellular links as a future use of the IP technology. The use of the IP protocol on slow links implies to reduce the size of the IP header to improve the performance of this protocol in low speed links.

The difference between ROHC and CRTP is that ROHC mechanism takes into account the characteristics of a cellular link as: high BER (bit error rate), long RTT (round trip time) and residual errors in lower layers.

The first phase of ROHC protocol is a *negotiation*. In this phase the compressor and decompressor learns the different characteristics of the link and the parameters that they will use for compression. ROHC mechanism makes a classification of the header fields; this analysis is based on how the values in the header fields change during a stream.

* This project is made within the research activities of LAFMI.

There are five categories for each header field: INFERRED, STATIC-DEF, STATIC-KNOWN, CHANGING and STATIC. With this classification, ROHC separates the static and dynamic part of the ROHC header packets.

ROHC use a *context* maintained between compressor and decompressor. This context contains the last correct update of the original header and the redundant information in the stream. This context is kept in the compressor and in the decompressor in order to assure the robustness of the mechanism. Each time a value in the context change, the context is updated. If the context is lost, there is no synchronization between compressor and decompressor. The decompressor can request the context through the possible use of acknowledgements, if unidirectional mode is used the decompressor has to wait for a timeout in the compressor to have the context update.

The principle of ROHC is to send the minimal information to achieve the header decompression. The key element here is the CRC (Cyclic Redundancy Check) that is computed before the compression. This gives valid information to the decompressor for the actual information in the decompressor and for the corrupted information that perhaps the decompressor has received due to transmission errors.

ROHC mechanism use *profiles* to identify the flows, *compression levels* to increase the compression rate, three *operation modes* and five transition *modes* to adapt the variable characteristics of the link to the compression. Each operation mode has three levels of compression, and each transition mode works in the fist two level of compression of the actual operation mode using the first two levels until a change mode acknowledgement is received.

3.1. ROHC profiles

The profiles give a definition of the protocol headers that will be compressed in the header stream. Through the profiles the decompressor can learn the IP version, if the stream use RTP or ESP (IP Encapsulating Security Payload) or if there is only a UDP stream. Currently 5 profiles have been defined:

Profile 0 without compression: When this profile is used only the ROHC Context Identifier (CID) is added to each packet to let the decompressor know that the stream is not compressed.

Profile 1, $IP_{v4/v6}$/UDP/RTP header compression: This is the generic profile.

Profile 2, $IP_{v4/v6}$/UDP header compression: This is a variant of profile 1, where the compression is only applied for the UDP/ $IP_{v4/v6}$ headers and then it is the base for the other profiles.

Profile 3, $IP_{v4/v6}$/ESP header compression: This profile compresses the ESP protocol. It uses the profile 2 header format packets.

Profile 4, $IP_{v4/v6}$ header compression: This profile compresses only the $IP_{v4/v6}$ header and is based in profile 2 header format packets.

3.2. Compression Levels and Operation Modes

ROHC has three compression levels: Initiation and Refresh (IR), First Order (FO) and Second Order (SO). The compression level gives the header format packet based on the information the compressor has to send, as shown in table I. Each packet is sent with a specific frequency. This frequency is determined by different factors such as: an acknowledgement receipt, time out, update or in a constant period, base on the confidence the compressor has of radio link reliability. The size of ROHC packets depends on the compression level and the confidence of compressor.

The three operation modes allow the compressor to improve the compression performance because the mechanism could change from one mode to another based on the link characteristics. Each operation mode has its own behavior, the unidirectional (U) mode of operation and bi-directional optimistic (O) mode are more complex than the bi-directional reliable (R) operation mode. This is because neither the compressor nor the decompressor has all the information or all the parameters to perform the compression. The compressor changes the compression level by reducing/increasing the size of the header sent.

TABLE 1. HEADERS FORMAT PACKET USED IN EACH COMPRESSION LEVEL OF EACH OPERATION MODE.

Compression Level	Initiation and Refresh (IR)	First Order (FO)	Second Order (SO)
Header Format Packet used	IR (48-131bytes)	IR-DYN (21-84 bytes), UOR-2 (3-18 bytes),	UO-0 (1 byte), UO-1 (2 bytes), R-0 (1 byte), R-0-CRC (2 bytes), R-1 (2 bytes)

The header format packet will send different information about the header. In the IR packet, the static and dynamic part of the header could be sent. In the IR-DYN, only the dynamic part is sent. The UOR-2 packet sends the encoded timestamp and the sequence number, and the M bit of RTP; it has an option for the ROHC extensions. The encoding timestamp and sequence number, and the M bit of RTP form the type-1 packets. The type-0 packets send only the encoded sequence number.

The U-mode has different regeneration control algorithms. This mode uses two timers and a confidence system (L). L gives the number of times compressor will send the header format packets of a compression level One timer is used to generate the complete context and the compressor restart in the IR level. The other timer is used for the updates between the SO and the FO compression levels. L is based on the behavior of the link error (BER and RTT). In this operation mode, the compressor controls the header size. If there is a change in the header, the compressor can return to a lower compression level in order to give all the necessary information to the decompressor.

The decompressor cannot give any information to the compressor; everything is based on the control mechanism, because the acknowledgements are not enabled. This operation mode has the lowest performance but the mechanism achieves the correct transmission of the headers.

The O-mode is very similar to U-mode but the decompressor can send negative acknowledges. This operation mode does not use the two timers, but it keeps the confidence system (L). The changes of compression level are done according to the confidence level and when the compressor does not receive any negative feedback. There are two kinds of negative acknowledges: NACKS, where compressor goes down one level of compression, and the STATIC-NACKS that give a negative transition to the lowest level of compression. If the compressor receives a packet that will update the entire context, and a SO packet cannot communicate the changes, compressor automatically changes down to the FO level of compression.

The R-mode works only with the acknowledgements received from the decompressor. Each time the compressor receives an ACK/NACK, the compressor change the compression level and goes to the IR state if an STATIC-NACK is received.

3.3. Transition Modes

Each time the decompressor is able to work in a new operation mode, it launches the transition modes, and it sends an ACK/NACK with the new compression level that he wants to work with. Generally, the transition modes work with the first two compression levels of the lastly used operation mode.

All the packets sent in the transitions modes contain a CRC to verify the information. During the transition, each of the compressor and the decompressor keep, two control variables: Transition and Mode that deny the initialization of another transition. If the transition variable has a pending value, the transition mode cannot be release. To finish the transition, an ACK with the sequence number and the operation mode has to be received by the compressor, if it is not the case, the transition variable keeps the pending value and Mode keeps the old value.

The only different transition is between U-mode to O-mode this transition is automatic. To initiate the transition to R, the context has to be established between the compressor and the decompressor; for the others transition the decompressor can start the transition at any moment.

4. ROHC for PPP Links

The ROHC layer is located between the IP layer and the layer two. In the IP layer, each IP interface can have multiple channels each one bi-directional or unidirectional. In each channel a different ROHC CID (Context Identifier) stream could be deployed. A channel is a point-to-point connection between two IP interfaces of two nodes. Each channel has a compressor and a decompressor at both sides of the channel.

The platform of ROHC can have many ROHC channels running different streams at the same time. For each instance of ROHC there is a ROHC channel and there could be a feedback channel. The Context Identifier (CID) of ROHC mechanism identifies each stream. There is a maximum number of CID, negotiated initially. In a Node, there can be as many different compressors or decompressors as MAX-CID free exist, running at the same time with different consecutives streams.

4.1. ROHC INFORMATION FLOW

4.1.1. Negotiation: In a PPP connection the process of negotiating the network parameters is handled by the network control protocols (NCPs); in this process, each end of the link must agree on a set of configuration parameters for the compression.

There are different parameters for IPv4 and for IPv6, so there are specific values for each one in the NCP protocol define in [5], see figure 1.

When the negotiation has been done, the compressor can start working in unidirectional mode of operation in the initialization state.

4.1.2. Transition Modes: In ROHC there are five different transition modes. When the decompressor starts a transition mode, there are two-control variable, which help to establish the new operation mode. The Transition variable gives the status of the transition. If the Transition variable is not in D (done) state any other transition can be initialized see figure 1.

The transition from U-mode to O-mode is automatic; only a correct acknowledgement from Decompressor (D) will trigger to the new mode of operation and then the Compressor (C) starts working in the O-mode.

The transitions from O to R, from U to R, from R to O and from R/O to U are very similar. It is possible to start this transition only after at least one packet has been correctly decompressed. An acknowledgement from

D with the new mode and carrying the CRC is sent to C. In the transition C has to use packets from the two first levels of compression with the new mode of operation until an ACK is received with the correct Sequence Number (SN), then C can start sending Second Order (SO) level packets and the transition is finished.

An example of the information flow for ROHC is shown in Figure 1. The detailed information between the Compressor and Decompressor can be deduced from [1]. The following example shows a typical procedure:

PPP opens a link

NCP negotiates ROHC parameters. C sends a negotiation packet and D responses accepting this parameters

C starts sending ROHC packets in U-mode

D triggers a transition from U to O

C starts sending ROHC packets in O-mode

D triggers a transition from O to R. A handshake is taking place using the first two level of compression of O-mode, when transition is finished C sends SO order packets.

5. Implementations Results

Our implementation uses the profile 1 and profile 4. The profile 1 experimental system consists in a video application platform in v6 and a PPPoE v6 based on FreeBSD4.5 with Kame. For profile 4, an IPv6 flow is used. The client receives through the PPPoE the two header compressed flows with different CID that will be decompressed by the ROHC decompressor in the other node. At the beginning node A is in U-mode. No feedback is sent by node B until the acknowledgement sent by B changes the operation mode of compressor.

The results are positive and in favor to use ROHC. In the graph of figure 2, we show the performance of both profile 1 and 4. The curve for profile 4 presents a little lag due to the differences of the header format packets used explained in the next section.

For each profile, we can see the importance of the feedback channel in R-mode. In addition, we notice the differences among U-mode, O-mode and R-mode, it consist in an increase use of the downlink channel, the refreshing compression parameters and the confident system. R-mode is interesting because the confidence system is not used, the larger headers are only sent at the beginning or when the context is lost and most of the time, SO packets are used. O-mode keeps the confident system but use the negative acknowledgements to refresh the context.

The performance in the uplink is improved in R-mode because the smallest headers are used more frequently than the larger ones. U-mode is the less performing because of the use of timers that make periodic refreshes of the complete header and the confidence variable that produce a repetition of the same header format packet L times.

It is important to mention that Figure 2 is the evolution of the compression without error. When this happens, the performance of O-mode and R-mode are similar in the uplink but different in the downlink. This could leave the idea that O-mode performance is better because the downlink is less used; but in a noisy link where link parameters vary the use of downlink in R-mode gives a better feedback than a confidence system, which is based on fixed initial values.

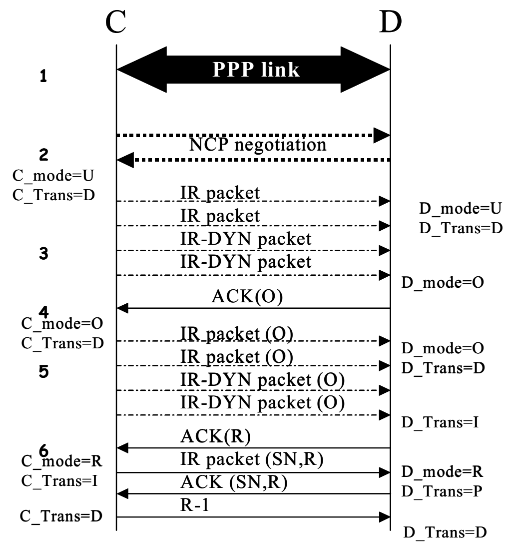

Figure 1. ROHC Information Flow. After the PPP channel is opened the NCP negotiated able the ROHC compression. Two variables are kept in each side Transition and Mode.

The performance of ROHC mechanism with different error rates is shown in figure 3, the percentage of error affecting more often the payload and not the header information. This is because with the use of ROHC the header is smaller and the probability to have an error in the header has been reduced then the error is in the payload.

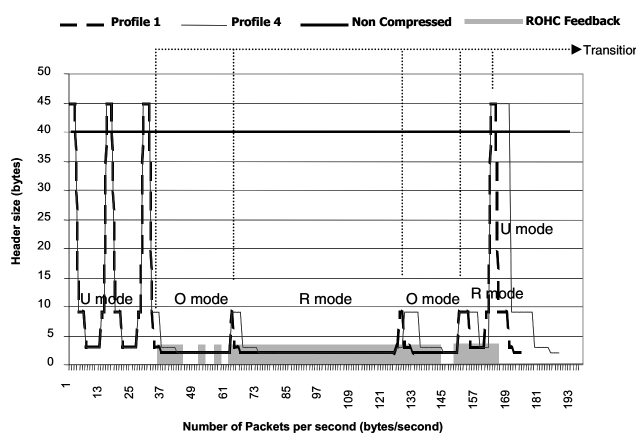

Figure 2. ROHC compression for Profile 1 and 4 without error. The compressor is using the three Operation Modes (U, O, R), the five Transition Modes and the use of feedback in the bi-directional modes.

The implementation is working with IPv6 then the UDP checksum is mandatory, when the BER is high and the error is in the header, ROHC tries to correct the error. The recovery schemes are working well some packets are corrected and sent to the upper layer but in some cases we have noticed that the application, as it uses the UDP checksum, drop the packet in case of error, as shown in Figure 3, then packet loss is due to the UDP checksum. Even though there are some packets affected by the BER in the header.

ROHC can support some isolated error, as we can see the behavior of ROHC in $1x10^{-3}$, $4x10^{-3}$ and $5x10^{-3}$ error rates, the losses in ROHC has not change because the protocol is able to correct the header packets but when there are consecutive errors, the decompressor cannot assure the confidence of the information and the packet is drop by ROHC. The worst case is taken when the error arrives to $5x10^{-2}$ where ROHC cannot recovery the error, the compressor downwards to the IR state to update the lost context. Some times when the error is recovery the packet is dropped because of UDP checksum.

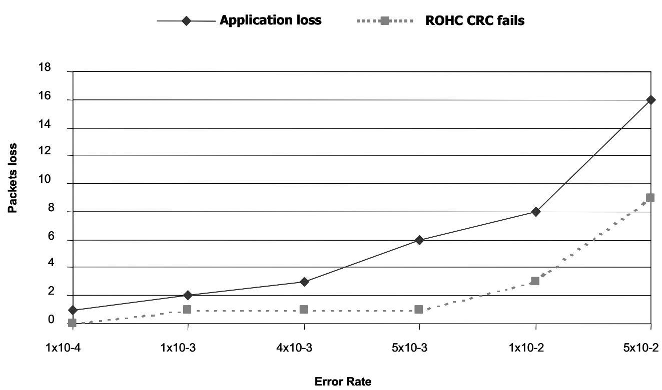

Figure 3. Percentage of Error in Payload vs. Error in Header

If several consecutive errors arrive then ROHC reduces the level of compression. When the context is lost, ROHC goes down to the lowest level of compression where largest headers are sent. In O-mode and R-mode ROHC will loose all the packets in a RTT until an IR packet arrives. In U-mode, all the packets will be dropped until a timer refresh the context.

6. Differences between Profile 1 and Profile 2-based profiles

During the deployment of the test for both profiles we have found some differences between the Profile 1 and Profile2-based profiles (Profiles 2, 3, and 4 of ROHC). These differences can allocate a bad performance and the uselessness of the mechanisms for any compression using IPv6 as we can see in figure 2. The problems were mentioned to the IETF working group.

6.1. Control variable Error transmission.

A bad performance is shown for ROHC when other profile different than Profile 1 is used. In profile 1 (UDP/RTP/IP), ROHC use the RTP sequence number bits for the compression control. ROHC adds some bits to the dynamic part of the original RTP header in order to send the compression information to decompressor. This information is the operation mode and some flags to known the encoding method used for Timestamp and if extensions of IP are present.

The compression for UDP/IP, ESP/IP and IP (Profiles 2,3 and 4) ROHC mechanism adds a sequence number randomly created for compressor to control the transmission. This sequence number is placed out of the original header and the ROHC control variables are not added, as shown in figure 4.

The CRC in ROHC is computed over the original header for the IR and FO level compression packets but it is calculated over the compressed header in the SO level compression. In CRC calculation for profile 2,3 and 4 the sequence number is not taken into account. When there is an error in the sequence number, decompressor cannot resolve if there is an error in the header, in IR level of compression it cannot use the recovering schemes and the packet is lost (see figure 4).

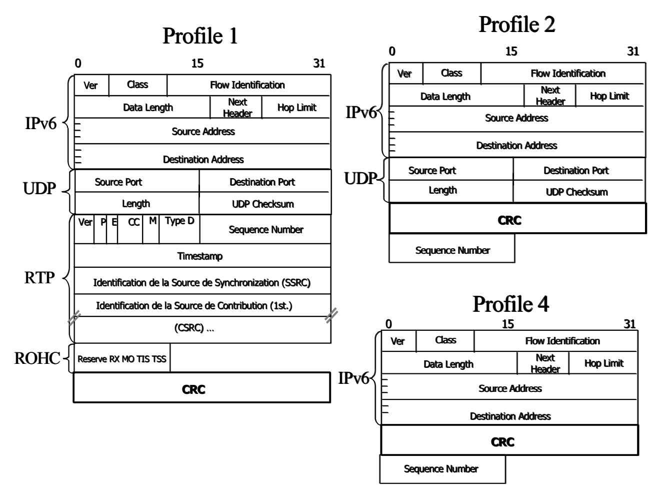

Figure 4. The computing of CRC for ROHC profiles

6.2. Mode Transition Duration.

The two bits for operation mode added in the dynamic part of the original RTP header in profile 1 and that are not added for the other profiles generate some problems in the transition mode of operation unless for transition from U to O that is direct, see figure 2.

When decompressor initiated the transition it stays in the initial point until the reception of a packet with the sequence number and the operation mode, see figure 5. The only packet that could be used to give this information to decompressor is UOR2+extension3 that is used when an update in the dynamic part of the header is needed. This does not perturb the compression but the transition will not be use. The mechanism will transit to O-mode to perform the best. In the implementation we have observed that transitions other than U to O in profile 2 the system takes more than two additional RTT to conclude the transition (see figure 5).

6.3. Useless bits sent for IPv6.

In profile 1 there are different header format packets for each version of IP. For the other profiles, there is only one header format packet for both version of IP. IP-ID (2 bytes in original value and 6-7 bits in encoding value) of IPv4 is sent in every packet format due to the variability of its value. The original IPv6 header does not have the IP-ID field but it is present in ROHC profile 2. There is non-consensus in the working group to eliminate these bits for the IPv6 header, but the performance for IPv6 could be beneficiated because it is mandatory to use the UDP checksum (2 bytes).

7. Proposed Packet Format for Profile 2 Based Profiles

In the ROHC fields classification for profile 1 the mechanism used the RTP sequence number as a control variable of the protocol, it also add some bits for ROHC as part of the dynamic header of RTP header. Moreover, there is a specific definition of packets for each version of IP. When other profile is used, the packet format is based on profile 2 packets where there is only one packet format definition for both IP versions.

The necessity of the ROHC bits is important for the performance of the protocol, as we have seen in figure 5. We proposed to add this bits in the dynamic part of the IP header and not in the dynamic part of RTP header as proposed profile 1. As IP is the based protocol for all the profiles is the best place to have the ROHC bits and the artificial Sequence Number for Profile 2-based profiles needed for control. This artificial sequence Number will be included in the CRC computing to reduce the ambiguities in decompressor. It is important also to define a new header format packet for IPv6 in profile 2 based profiles to eliminate the use of IP-ID, reducing the size of the header.

8. Conclusions

The ROHC compression architecture profile 1 presented in this paper allows the Internet client to benefit from the advantages of a network offering a robust compressor mechanism to download a multimedia service. Modifications need to be done for profile 2-based profiles to give the same performance as profile 1.

ROHC header compression reduces the number of bits sent in a link, it assures the reduction of packets with error, per error rate. When compression is used, link error can lead to consecutive loss, the system can loose at least the number of packets in a RTT if IR packets are lost.

When IPv6 is used, it will be interesting to test the protocol with UDP-lite to reduce the number of dropped packets but there is not an standard for this protocol and ROHC standard does not include this option.

We will continue analyzing the different parameters in ROHC mechanism. They will give the next steps to define which could be the best mode of operation in noisy links or in high-speed applications.

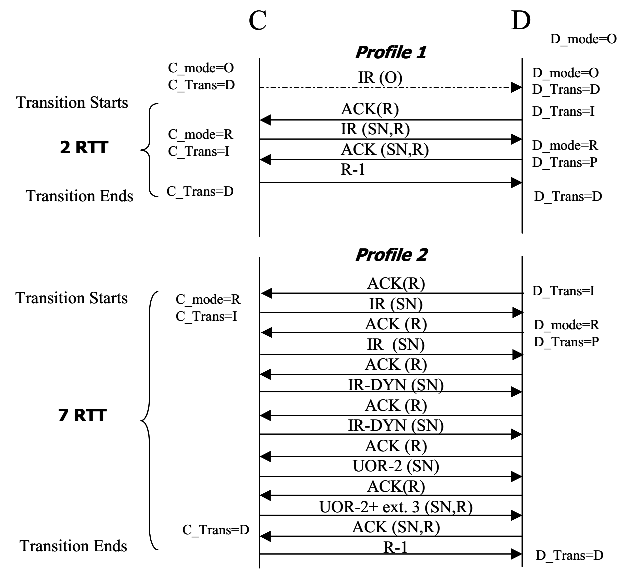

Figure 5. Mode Transition in Profile 1 and Profile 2 based profiles.

9. References

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

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

[3] Degermark, M. et. al. "IP Header Compression", RFC 2507, 1999.

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

[5] Bormann, C., “ROHC over PPP”, RFC 3241, 2002.