

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO
Departamento de Engenharia Electrotécnica e de Computadores

Robust Header Compression in 4G Networks

António Pedro Freitas Fortuna dos Santos

Dissertação submetida para satisfação parcial dos
requisitos do grau de mestre
em
Redes e Serviços de Comunicação

Dissertação realizada sob a supervisão do
Professor Doutor Manuel Alberto Pereira Ricardo,
do Departamento de Engenharia Electrotécnica e de Computadores
da Faculdade de Engenharia da Universidade do Porto

Porto, Maio de 2007

Abstract

The advent of the 4th generation of mobile networks (4G) may provide users with powerful and bandwidth demanding multimedia applications. At the wireless access networks, the transport of high bitrates may be obtained by using more spectrum, small cells, and new radio techniques such as adaptive modulations. It may also be obtained by optimizing the data transmissions and the QoS management processes. The transport of the applications data using the IP stack of protocols increases the amount of information to be transmitted, mainly due to the requirement of transmitting the IP protocols headers. The applications generating the smallest payloads, namely Voice over IP, will be responsible for the highest relative overheads; in this case, the amount of IP overhead may reach 75% of the total packet size. Assuming the voice traffic transported by GSM systems migrates to 4G packet switched networks, the amount of VoIP traffic is expected to become significant in the later networks. These two assumptions combined, significant amounts of VoIP traffic and large relative overheads, justify the study of header compression techniques and of their value in 4G networks.

The Robust Header Compression (RoHC) standard describes a set of header compression mechanisms, and addresses a set of scenarios including wireless links. But there is not, currently, a solution which enables RoHC to function properly across heterogeneous wireless networks. The integration of RoHC with 4G-like wireless networks is an open research topic.

The IST DAIDALOS I project aimed at delivering end-to-end services across heterogeneous networks, aligned with the current 4G networks vision. To accomplish this, the project has developed an architecture enabling the integration of heterogeneous wireless networks which also provides end-to-end QoS support. The QoS Abstraction Layer (QoS-AL) is the component responsible for setting the L2 QoS reservations in the wireless access networks.

In this dissertation we extend the QoS-AL in order to integrate the RoHC functionality. It has required the development of a new solution for transporting RoHC packets over IEEE 802 networks; this solution uses the Ethernet DIX encapsulation which is supported by most Ethernet equipment, and adds a L2.5 header containing a *Length* field, not present in the DIX header. Using this field, the bridges/APs can evaluate if padding was added to the frame. A scheme for the negotiation of RoHC Channels was also proposed.

The RoHC mechanisms were initially defined for wireless links having high BERs. In opposition to it, the IEEE 802.11 offers a confirmed frame delivery service which reduces the transmission errors to insignificant values. This means that the robustness mechanisms included in RoHC are not adequate to 802.11 links, although the compression itself seems to be of value. Thus, we performed a simulation to study

the performance of the RoHC U-mode over IEEE 802.11. The U-mode's parameters were also studied. Based on the experiments made, we proposed values for the U-mode's parameters.

Resumo

A quarta geração de redes móveis (4G) poderá proporcionar aos utilizadores aplicações multimédia avançadas, com requisitos exigentes em termos de largura de banda. Nas redes de acesso sem fios, podem-se obter maiores débitos utilizando mais espectro, células reduzidas, ou novas técnicas de rádio como adaptações de modulação. Maiores débitos também poderão ser obtidos otimizando as transmissões de dados e os processos de gestão de QoS. O transporte de dados aplicativos utilizando a pilha de protocolos IP aumenta a quantidade de informação a ser transmitida, em grande parte devido ao requisito da transmissão dos cabeçalhos do protocolo IP. As aplicações que geram menores conteúdos, nomeadamente o VoIP, serão responsáveis pelos maiores *overheads* relativos; neste caso, a quantidade de *overhead* poderá atingir 75% do tamanho do pacote a transmitir. Assumindo que a voz transportada por GSM migrará para redes de comutação de pacotes 4G, espera-se que a quantidade de tráfego VoIP nas redes 4G venha a ser significativa. Em conjunto, estes dois pressupostos, quantidades significativas de tráfego VoIP e maiores *overhead* relativos, justificam o estudo de técnicas de compressão de cabeçalhos e a sua avaliação no contexto das redes futuras de 4G.

O *Robust Header Compression* (RoHC) descreve um conjunto de mecanismos de compressão de cabeçalhos, pensados para um conjunto de cenários no qual se incluem ligações sem fios. Correntemente, não existe uma solução que permita ao RoHC funcionar correctamente sobre redes heterogéneas sem fios. A integração do RoHC com redes de 4G sem fios é um tópico de investigação em aberto.

O projecto IST DAIDALOS I teve como objectivo oferecer serviços extremo-a-extremo sobre redes heterogéneas sem fios, alinhados com as actuais visões de redes de 4G. Para tal, o projecto desenvolveu uma arquitectura que integra redes heterogéneas sem fios, com suporte para QoS extremo-a-extremo. O *QoS Abstraction Layer* (QoS-AL) é o componente responsável por efectuar as reservas de QoS de nível 2 nas redes de acesso sem fios.

Nesta dissertação, extendemos o QoS-AL por forma a integrar funcionalidade RoHC. Para tal foi necessário o desenvolvimento de uma nova solução para o transporte de pacotes RoHC sobre redes IEEE 802; esta solução utiliza encapsulamento Ethernet DIX, suportado pela maioria do equipamento Ethernet, e adiciona um cabeçalho nível 2.5 contendo um campo *Length*, não presente no cabeçalho DIX. Através deste campo, os comutadores/APs conseguem verificar se foi adicionado *padding* à trama. Foi proposto um esquema para a negociação de canais RoHC.

Inicialmente, os mecanismos do RoHC foram concebidos para ligações sem fios com altas taxas de erro. Contrariamente a isso, a norma IEEE 802.11 oferece um serviço de entrega de tramas confirmado, que reduz a transmissão de erros para valores

insignificantes. Isto significa que a robustez dos mecanismos incluídos no RoHC não é adequada para ligações 802.11, apesar de a compressão dos cabeçalhos ser proveitosa. Assim, conduzimos um estudo da performance do *U-mode* do RoHC sobre IEEE 802.11. Os parâmetros do *U-mode* também foram estudados. Com base nas experiências efectuadas, propusemos valores para os parâmetros do *U-mode*.

Acknowledgements

The author would like to thank a number of people for making this work possible. First, I would like to thank my supervisor, Prof. Manuel Ricardo, for his guidance, invaluable advice and support, and for always being available.

I would also like to thank my colleagues at INESC Porto for their support, in particular to Filipe Abrantes for his interest in my work and useful advices, and to Gustavo Carneiro for his help and generosity. The INESC Porto institution also deserves my appreciation, for providing me with excellent work conditions and an inspiring research environment where this work was conducted.

Thanks also to Nuno Pereira, for a never ending exchange of ideas and interesting discussions on computers and networks.

Last but not least, I would like to express my gratitude to my family, for all the patience and support during the elaboration of this dissertation, and to Sónia Couto, for standing next to me, and for all the emotional support.

Contents

1. Introduction	1
1.1. Reference Scenario and Scope	2
1.2. Objectives.....	3
1.3. Contributions.....	3
1.4. Results	4
1.5. Organization of the Dissertation	4
2. Robust Header Compression.....	5
2.1. Introduction	5
2.2. Former Header Compression Mechanisms	5
2.2.1. Thinwire-II	5
2.2.2. cTCP.....	6
2.2.3. IPHC.....	6
2.2.4. cRTP.....	7
2.2.5. ECRTP	8
2.3. Robust Header Compression (RoHC)	8
2.3.1. RoHC Architecture.....	8
2.3.2. Header Compression Profiles.....	11
2.3.3. Lower Layer Requirements.....	12
2.3.4. RoHC Channel Negotiation	13
2.3.5. Header Compression States.....	13
2.3.6. RoHC Packets	14
2.3.7. Operation Modes	15
2.4. Conclusions	19
3. Header Compression in Cellular Networks	21
3.1. Introduction	21

3.2.	GPRS.....	21
3.2.1.	GPRS Architecture.....	21
3.2.2.	Protocol Architecture	23
3.2.3.	Header Compression in GPRS	24
3.3.	UMTS.....	26
3.3.1.	UMTS Architecture.....	26
3.3.2.	Protocol Architecture	27
3.3.3.	Header Compression in UMTS	28
3.4.	Conclusions	29
4.	QoS Abstraction Layer	31
4.1.	Introduction	31
4.2.	IST-DAIDALOS	31
4.3.	QoS Abstraction Layer.....	33
4.3.1.	QoS-AL Requirements.....	33
4.3.2.	QoS-AL Specification	34
4.4.	Conclusion.....	40
5.	AL-RoHC Specification	41
5.1.	Introduction	41
5.2.	Requirements.....	41
5.2.1.	RoHC Requirements on Lower Layers	41
5.2.2.	AL-RoHC Requirements.....	42
5.2.3.	RoHC and QoS-AL integration Requirements.....	43
5.3.	RoHC over IEEE 802 Networks	43
5.4.	The Ethernet Minimum Frame Size	44
5.5.	RoHC over QoS-AL Access Networks.....	46
5.5.1.	RoHC Channel Negotiation	46
5.5.2.	MTU considerations.....	48
5.5.3.	Architecture.....	48
5.5.4.	Modules.....	50
5.5.5.	Interfaces	51
5.5.6.	AL-RoHC Stages.....	51
5.6.	Specification and Requirements.....	54
5.7.	Conclusion.....	54
6.	RoHC over IEEE 802.11	55
6.1.	Introduction	55

6.2.	Simulation Scenario and Assumptions	55
6.3.	Simulation Model	56
6.3.1.	VoIP Modelling	57
6.3.2.	Simulation Parameters	58
6.4.	Experiments and Results	59
6.4.1.	Throughput Experiments	59
6.4.2.	RoHC Performance Experiments	61
6.4.3.	RoHC U-mode Parameters Experiments	71
6.5.	Validation	74
6.5.1.	Test case parameters	74
6.5.2.	Analysis	75
6.6.	Conclusions	76
7.	Conclusions	77
7.1.	Conclusions	77
7.2.	Original Contributions	78
7.3.	Results	78
7.4.	Future Work	79
Annex A - QoS-AL Service Primitives	81	
A.1.	Data types used in the Primitives	81
A.2.	QoS Reservation Primitives	81
A.3.	Resource Querying Primitives	83
A.4.	QoS Notification Primitives	83
A.5.	Mobility Primitives	84
A.6.	Multicast Primitives	84
Annex B - QoS-AL Driver Primitives.....	85	
B.1.	Driver Registration Primitives	85
B.2.	QoS Reservation Primitives	85
B.3.	Resource Querying Primitives	86
B.4.	QoS Notification Primitives	87
B.5.	Multicast Primitives	87
Annex C - AL-RoHC Details	89	
C.1.	AL-RoHC PDUs	89
C.2.	AL-RoHC Interfaces	90
Annex D - IEEE 802.11	93	

List of Figures

Figure 1 – QoS-AL Reference Scenario.....	2
Figure 2 - Channel Abstraction (using point-to-point links)	9
Figure 3 - Channel Abstraction (using point-to-multipoint links).....	9
Figure 4 - RoHC Channels and Peers.....	10
Figure 5 - RoHC Compressor state machine	13
Figure 6 - RoHC Decompressor state machine	14
Figure 7 – General RoHC Packet Structure	15
Figure 8 - RoHC Operation Modes	16
Figure 9 - Compressor states and transitions in U-mode	16
Figure 10 - Decompressor states and transitions in U-mode.....	17
Figure 11 - Compressor states and transitions in O-mode	18
Figure 12 - Compressor states and transitions in R-mode.....	18
Figure 13 – Operation mode transitions	19
Figure 14 - GSM Architecture.....	22
Figure 15 - GPRS Network Architecture	22
Figure 16 - End-to-end GPRS protocol stack between the MS and the GGSN (user plane)	23
Figure 17 - SNDCP XID negotiation packet structure	24
Figure 18 – UMTS Network Architecture.....	27
Figure 19 - End-to-end UMTS protocol stack between the UE and the GGSN (user plane)	27
Figure 20 - DAIDALOS Network Architecture	32
Figure 21 – DAIDALOS E2E QoS Concept	32
Figure 22 - QoS-AL Architecture.....	34
Figure 23 - QoS Connection Activation Example (from [44]).....	38
Figure 24 - Example of a Resource Query with an AP as target (from [44]).....	39
Figure 25 – Ethernet DIX and IEEE 802.3/802.2 frames.....	44

Figure 26 - The use of padding in Ethernet links	45
Figure 27 – Padding propagation to non-Ethernet links example	45
Figure 28 - RoHC encapsulation in DIX frames	46
Figure 29 - RoHC encapsulation in QoS-AL Access Networks	46
Figure 30 – Standalone negotiation - Simple Mode signalling (Unidirectional)	47
Figure 31 - Standalone negotiation - Quick Mode signalling (Bidirectional).....	47
Figure 32 - AL-RoHC negotiation - Quick mode piggybacked (Bidirectional).....	48
Figure 33 - AL-RoHC Architecture.....	49
Figure 34 – RoHC Channel Negotiation (using Quick mode piggybacked).....	52
Figure 35 - RoHC Packets Exchange	53
Figure 36 - RoHC Channel Termination	53
Figure 37 - Simulation Scenario.....	56
Figure 38 - Simulation Model	57
Figure 39 – Maximum Throughput Experiment.....	59
Figure 40 – Maximum Throughput Experiment (1500 byte packets and RTS/CTS Threshold of 1000 bytes).....	60
Figure 41 - Maximum Throughput Experiment (2304 byte packets and RTS/CTS disabled)	60
Figure 42 – RoHC Throughput Experiment.....	60
Figure 43 – RoHC Throughput Experiment.....	61
Figure 44 - RoHC Throughput Experiment.....	61
Figure 45 – RoHC Performance Simulation Sets.....	62
Figure 46 – RoHC’s Gain (without background traffic)	64
Figure 47 – RoHC Compression Ratio (without background traffic)	64
Figure 48 – Throughput (without background traffic)	64
Figure 49 – Average Delay (without background traffic)	64
Figure 50 – Jitter (without background traffic)	64
Figure 51 – Packet Loss (without background traffic)	64
Figure 52 – RoHC Compression Ratio (RoHC + background TCP traffic).....	66
Figure 53 – RoHC’s Gain (RoHC + background TCP traffic – 1500 byte TCP packets)	66
Figure 54 – RoHC’s Gain (with background TCP traffic - 80 byte TCP packets).....	66
Figure 55 – RoHC + TCP Throughput.....	66
Figure 56 - RoHC Throughput (with background TCP traffic).....	66
Figure 57 – TCP Throughput (with RoHC traffic).....	66
Figure 58 – Average delay (RoHC + background TCP traffic)	67

Figure 59 – Jitter (RoHC + background TCP traffic).....	67
Figure 60 – RoHC’s Gain (with background UDP traffic)	68
Figure 61 – Packet Loss and Loss Propagation (RoHC + background UDP traffic)	68
Figure 62 – RoHC Packet Loss and Loss Propagation (with background UDP traffic)	69
Figure 63 – UDP Packet Loss (with RoHC traffic).....	69
Figure 64 – RoHC + UDP Throughput	69
Figure 65 – RoHC Throughput (with background UDP traffic)	69
Figure 66 – UDP Throughput (with RoHC traffic)	70
Figure 67 – Average Delay (RoHC + background UDP traffic).....	71
Figure 68 – Jitter (RoHC + background UDP traffic).....	71
Figure 69 – RoHC’s gain with 1-50 VoIP calls and background UDP traffic	73
Figure 70 – RoHC’s gain with 60 VoIP calls and background UDP traffic	73
Figure 71 – RoHC’s gain with 100 VoIP calls and background UDP traffic	73
Figure 72 – RoHC’s gain.....	74
Figure 73 – Average Delay.....	74

List of Tables

Table 1 - Header Compression Mechanisms supported by SNDCP	24
Table 2 - cTCP header compression parameters (PCOMPs)	25
Table 3 - IPHC header compression parameters	25
Table 4 - RoHC header compression parameters	26
Table 5 - Mappings of PID values for IPHC	28
Table 6 - Mappings of PID values for RoHC (CIDs in PDCP headers)	29
Table 7 - Mappings of PID values for RoHC (CIDs in RoHC packets)	29
Table 8 - DAIDALOS QoS Classes and Parameters (from [44]).....	35
Table 9 - Data plane QoS mapping rules (from [44]).....	36
Table 10 - AL PDUs.....	39
Table 11 - RoHC Channels Table example	50
Table 12 - Identifiers Map Table example	50
Table 13 - VoIP Codecs candidates.....	57
Table 14 – Throughput in Simulation Set B (1500 byte TCP packets).....	67
Table 15 – Average Delay in Simulation Set B.....	67
Table 16 – Jitter in Simulation Set B.....	67
Table 17 - Packet Loss and Loss Propagation in Simulation Set C	69
Table 18 – Throughput in Simulation Set C.....	70
Table 19 – Average Delay in Simulation Set C.....	71
Table 20 – Jitter in Simulation Set C.....	71
Table 21 - Traffic produced by a single VoIP source in 20 Compression Cycles	75

1. Introduction

The advent of the 4th generation of mobile networks (4G) may provide users with powerful and bandwidth demanding multimedia applications, which may have real-time requirements under mobility scenarios. Users may expect to execute in their light mobile terminals the same tasks they currently do on their desktop computers at home or office. 4G network improvements with respect to 3G networks include: data rates up to 100 Mbit/s in Wide Area Networks (WAN) and 1 Gbit/s in Wireless LANs; IP-based transport of voice, video, and data; seamless mobility between heterogeneous wireless access networks, such as IEEE 802.11, IEEE 802.16, or Bluetooth. The research areas associated with 4G are broad, from radio technologies to networking and applications.

Multimedia applications will demand the transport of high bitrates. At the wireless access networks, the transport of high bitrates may be obtained by using more spectrum, small cells, and new radio techniques including adaptive modulations and Ultra Wide Band (UWB). It may also be obtained by optimizing the data transmissions or the QoS management processes. The transport of the applications data using the IP stack of protocols increases the amount of information to be transmitted, mainly due to the requirements of transmitting the IP protocol headers. The applications generating the smallest payloads, namely Voice over IP (VoIP), will be responsible for the highest relative overheads. A typical IPv6 VoIP packet includes a 40 byte IP Header, an 8 byte UDP header, a 12 byte RTP header and a 20 byte payload. In this case, the total IP overhead is about 3 times the size of the payload, or 75% of the total packet size. Assuming the regular voice traffic transported by GSM systems migrates to 4G packet switched networks, the amount of VoIP traffic is expected to become significant in the later networks. These two assumptions combined, significant amounts of VoIP traffic and large relative overheads, justify the study of header compression techniques and their value in future 4G networks.

The Robust Header Compression (RoHC) standard [1] describes a set of header compression mechanisms, and addresses a set of scenarios including wireless links. The 3GPP has recognized the value of using header compression, and it has included RoHC in the Release 6 of the UMTS [30]. There is not, currently, a solution which enables RoHC to function properly across heterogeneous wireless networks. The integration of RoHC with 4G-like wireless networks seems to be an open and interesting research topic.

1.1. Reference Scenario and Scope

Part of the work described in this dissertation uses the network architecture provided by the IST DAIDALOS I project [43] as reference scenario. The main objective of the DAIDALOS I project was to deliver end-to-end services across heterogeneous networks, aligned with the current 4G networks visions.

In DAIDALOS networks, the Mobile Terminal (MT) is connected to an operator's network by means of an Access Router (AR). However, the AR does not offer direct connectivity to the mobile terminal. MTs are connected to the AR through a L2 access network, which may consist of concatenated wireless links of multiple technologies such as IEEE 802.11, IEEE 802.16 and IEEE 802.3.

In order to provide end-to-end QoS guarantees to applications, the DAIDALOS framework includes a L3 QoS architecture based on QoS Brokers which handle the QoS reservations between ARs. The provisioning of end-to-end QoS guarantees demands also QoS guarantees at the L2 links, in the access networks. This is achieved with the help of a new QoS Abstraction Layer (QoS-AL) [44]. The QoS-AL handles the QoS parameters of multiple L2 technologies and offers an abstract interface (AI) to the L3 QoS modules, thus providing L2 QoS transparency. In DAIDALOS I the QoS reservations are initiated mainly by the network side. The QoS negotiation between the MT and the AR is made using the QoS-AL protocol. Figure 1 illustrates the DAIDALOS I access network reference scenario, in which the QoS-AL is used.

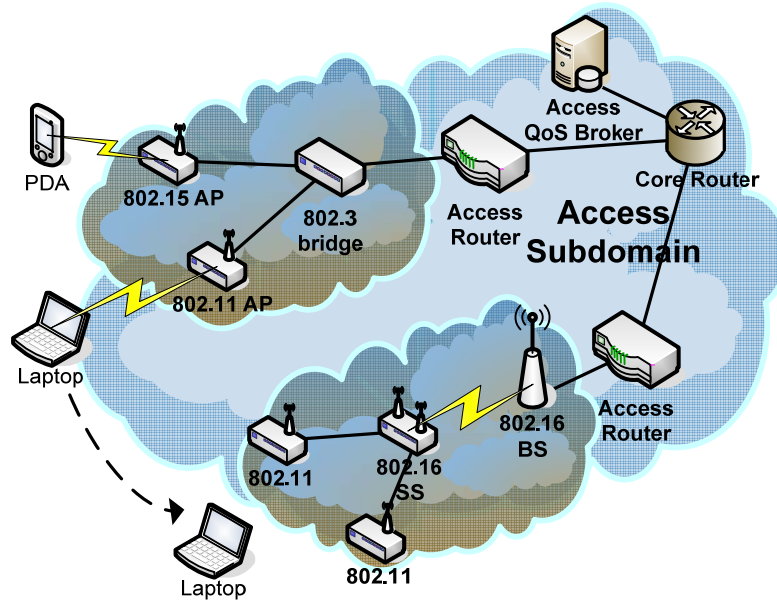


Figure 1 – QoS-AL Reference Scenario

As mentioned above, the DAIDALOS L2 access networks may consist of concatenated wireless and wired links. QoS reservations along the L2 path between a MT and the AR are usually constrained by wireless resources, which are limited and vary along the time. Assuming this reference model, the most viable scenario for using RoHC is to use it between the MT and the AR. Adding RoHC to this architecture poses some challenges:

- before two hosts are able to exchange header compressed packets, they have to establish a *RoHC Channel*. The establishment of this channel demands signalling;

- the RoHC standard provides just a solution for transporting RoHC packets over PPP links. Therefore there is not, currently, a solution for transporting RoHC packets over IEEE 802 Networks;
- the QoS framework should be aware of how much bandwidth is saved by RoHC. In this way, it could better estimate the resources left unused in the network;

The first problem addressed by this dissertation is how to extend the QoS-AL framework in order to enable it to support RoHC. A solution was proposed, the AL-RoHC, which enables the transport of RoHC packets over heterogeneous links, taking advantage of the existing QoS-AL reservation primitives.

RoHC mechanisms were initially defined for wireless links having high BERs. In opposition to it, the IEEE 802.11 offers a confirmed frame delivery service which reduces the transmission errors to insignificant values. This means that the robustness mechanisms included in RoHC are not appropriate to 802.11 links, although the compression itself seems to be of value. This calls for a study of the implications of transporting RoHC over 802.11, which is expected to be common in 4G networks. Therefore, the second problem addressed by this dissertation is the characterization of the RoHC performance over IEEE 802.11.

1.2. Objectives

The objectives defined for this dissertation are divided in two groups:

- To specify a solution for integrating RoHC capabilities into the DAIDALOS QoS-AL framework. This includes:
 - designing an extension to the QoS-AL framework which enables the negotiation of RoHC channels over heterogeneous access networks; it shall be mobility friendly and live with fast handovers;
 - defining a solution for supporting RoHC over IEEE 802 networks; it shall address IEEE 802.3, IEEE 802.11, IEEE 802.15, and IEEE 802.16;
- To study the performance of RoHC over a reliable wireless transmission service, such as the 802.11.

1.3. Contributions

The following original contributions have been produced during the course of this work:

1. An encapsulation solution for transporting RoHC packets over IEEE 802 Networks.
2. A mechanism for negotiating RoHC channels.
3. An extension to QoS-AL framework, including new primitives, modules and interfaces, which enables the integration of RoHC in the QoS-AL.
4. A performance study of RoHC over IEEE 802.11, which characterizes quantitatively the value of RoHC in wireless LAN scenarios.

1.4. Results

A simulation model of RoHC's U-mode has been produced using the Network Simulator 2 (NS2) [31]. This simulation model was used to study the RoHC performance over IEEE 802.11.

1.5. Organization of the Dissertation

This dissertation is organised as follows. Chapter 2 describes the state of the art in header compression, culminating with in the detailed description of RoHC. Chapter 3 provides the details of header compression in 3GPP specifications. Chapter 4 describes the QoS Abstraction Layer, including its architecture, interfaces, and protocol. Chapter 5 enumerates AL-RoHC requirements, proposes a solution for transporting RoHC packets, and specifies the AL-RoHC solution. Chapter 6 reports the study of RoHC performance over IEEE 802.11. Chapter 7 concludes the dissertation with final remarks about the results and contributions obtained, and directions for future work.

2. Robust Header Compression

2.1. Introduction

The motivations for performing IP header compression arose from the need to maximize the amount of data transferred over narrow band links. Data transferred in IP packets has overheads associated which reduce the transmission efficiency. This (in)efficiency can be described as the ratio between the packet header size (overhead) and the total packet size. A typical VoIP packet has an IP header of 40 or 60 bytes, depending if it is IPv4 or IPv6, an UDP header of 8 bytes, an RTP header of 12 bytes, and a small payload. If we consider a 20 byte payload and an IPv6 header, it results in a 100 byte VoIP packet, of which 80% is overhead. In these circumstances, header compression seems to be profitable.

Header compression uses mainly field suppression. This suppression consists in eliminating redundant fields, which can be inferred or calculated from other fields or other packets.

This chapter provides an overview of the evolution of the header compression mechanisms preceding RoHC, the main header compression protocol addressed in this dissertation.

2.2. Former Header Compression Mechanisms

2.2.1. Thinwire-II

Header compression (HC) was first explored by the Thinwire-II protocol [2], which was developed in 1984 to address the transmission inefficiency of low speed internet access. It was the first header compression scheme taking advantage of the inter-packet redundancy within a TCP packet flow, by transmitting only the differences in headers of consecutive packets. Both the Compressor and the Decompressor (i.e. the sender and the receiver) cache fields from previous packet headers, and store them locally, in a per flow context. This enables the Compressor to assess which fields need to be sent in the next packet (the fields that changed), and allows the Decompressor to reconstruct the header by joining the transmitted fields with the cached from that flow.

The first packet is always transmitted with a full header to build the per-context cache at both sides. Full header packets are uncompressed packets which in addition carry a context identifier. Subsequent packets carry a bitfield indicating which fields have changed, followed by the changed fields. This operation succeeds only if the context information is synchronized between the sender and receiver. Otherwise header decompression fails. In order to minimize packet loss due to header decompression failure, a full header is transmitted every N th packet, thus refreshing the cached

information. Despite being at the time an innovative scheme, it offered only mild compression rates; 40 byte IPv4/TCP headers were typically reduced to 13 bytes. Nevertheless, this scheme constituted the basis of today's header compression.

2.2.2. cTCP

In 1990, using the same concept Van Jacobson devised a scheme for compressing the headers of TCP traffic [3], later baptized as cTCP, which offered compression rates better than Thinwire-II; 40 byte IPv4/TCP headers were successfully reduced to 3 bytes. Thinwire-II mostly suppressed the static unchanged fields. Van Jacobson optimized compression by observing that often, non-static fields would vary by a constant factor, or at least by a small amount. Based on this assumption, instead of transmitting changing fields, cTCP transmitted only the difference between the changing fields of consecutive packets; this method was called delta encoding. To reconstruct a packet, the Decompressor adds the transmitted delta to the previous packet whose value is stored in the context. This improvement saves a considerable amount of bandwidth by removing 10 bytes to the average compression rate offered by the Thinwire-II. Furthermore, the cTCP algorithm was very simple, encouraging its implementation in hardware.

cTCP also introduced an error recovery mechanism in which the Compressor detects transport level retransmissions. When it happens, the Compressor knows that a packet has not been delivered and it sends a TCP packet with full header to synchronize the flow context at both endpoints.

The author tried to apply the same concept to UDP headers, but he concluded that changes were infrequent and that consecutive datagrams were often incoherent.

2.2.3. IPHC

Later on, the IP Header Compression (IPHC) [4], a general purpose IP header compression scheme, brought some improvements to cTCP and introduced the possibility of compressing IP headers (standalone, both IPv4 and IPv6), UDP headers, and headers from other layers, such as the RTP header [12]. IPHC can compress headers of UDP and TCP packets to 4-7 bytes, including two octets of UDP or TCP checksum.

When compressing non-TCP headers (e.g. UDP), IPHC does not use delta encoding. Every time a field changes, the Compressor sends a full header to build the new context information in the Decompressor. To prevent situations where the loss of a full non-TCP header packet would result in subsequent non-TCP headers being incorrectly decompressed, every version of the context of a non-TCP header is identified by a generation value, that is, a small number carried out by the full header. Note that non-TCP packets with full header also carry the context identifier. Compressed non-TCP headers carry the generation value of the context used to compress them. If a Decompressor sees a non-TCP header with a different generation value, it assumes the context is not up to date and discards or stores the packet until the next full header re-establishes the context. Differential coding is not used, and compressed non-TCP headers do not change the context. It means that the loss of a compressed non-TCP header does not invalidate subsequent packets. Periodically the Compressor sends a non-TCP full header to enable the Decompressor to recover quickly. The Compressor uses a slow start mechanism in which a non-TCP full header packet is sent with an exponentially increasing interval after a change in the context.

IPHC uses the same error recovery mechanisms of cTCP [3], but it also introduces two new mechanisms:

- TWICE, a context repair mechanism that synchronizes locally context information by adding the delta to the context twice (or more), in order to estimate the change that would occurred if the missing packets had arrived. The guess is verified by decompressing the packet and checking the checksum. This mechanism may accommodate some packet loss, but it is inadequate when the number of lost packets is high. Moreover, TWICE demands the use of UDP checksum, which may be problematic because some multimedia protocols actually tolerate transmission errors, and may need damaged packets to be delivered. The checksum also adds 2 bytes to the compressed header, thus reducing the compression efficiency.
- a link-level negative acknowledgment (NACK) to signal out of synchronization when compressing TCP. When the Decompressor fails to repair the context after a loss, it may optionally request a full header from the Compressor (by sending a `CONTEXT_STATE` message). This repair mechanism improved TCP throughput over links with higher bit error ratios (i.e. wireless links). The TCP checksums also contributed to improve the error detection ability.

2.2.4. cRTP

IPHC does not compress RTP, but can be extended to do so. Based on IPHC, and specified at the same time, Compressed RTP (cRTP) [5] was defined to compress the headers of IP/UDP/RTP flows. cRTP is able to compress the RTP headers to a minimum of 2 bytes if the UDP checksum is disabled, and to 4 bytes if enabled. cRTP uses the same techniques applied to TCP traffic; it suppresses static fields, applies differential encoding to changing fields, and eliminates changing fields whose variance is predictable (the same delta).

For error detection, it adds a 4-bit sequence number which detects packet loss. When the 4-bit sequence number for a context increments by other value than 1, the Decompressor invalidates the context, and sends a negative acknowledgement (NACK) to the compressor. All the compressed packets are then ignored until the context is repaired. This feedback mechanism is limited by the round-trip time speed. If the link has a long round trip time and if the link bandwidth is high, many consecutive packets may be discarded before the context is repaired. Another problem that may result in context corruption is the reordering of packets; cRTP assumes that the lower layer guarantees the packet order. The TWICE mechanism may also be used. When TWICE succeeds, NACK is avoided. In lossy links where bursts of errors are common, TWICE will only succeed if applied several times. However, due to UDP checksum's weakness, it is not advisable to use TWICE several times to repair a flow. Moreover, TWICE requires the use of the checksum, which adds 2 bytes to the compressed header.

A recent study [6] shows that cRTP is not robust against several consecutive lost packets between the Compressor and the Decompressor, and does not perform well on lossy or long round trip times links (e.g. wireless links). Thus, cRTP is not recommended for IP telephony over wireless links.

2.2.5. ECRTTP

The Enhanced Compressed RTP (ECRTP) [7] was developed to address cRTP loss multiplication caused by context corruption, by increasing robustness to both packet loss and misordering. To achieve this, it allows the compressor to send selective updates of the absolute values of the context, in addition to the existing updates of the delta values. ECRTP becomes reliable by sending the changes multiple times (N) in order to keep the Compressor and Decompressor synchronized, instead of sending full header packets. N represents the quality of the link between the Compressor and Decompressor, so the probability of more than N consecutive packets getting lost is small. As long as less than $N+1$ consecutive packets are lost, the context at the Decompressor is guaranteed to be synchronized with the context at the Compressor, and the use of TWICE successfully recovers from packet loss and reconstructs the compressed headers. If more than $N+1$ consecutive packets are lost, the context is invalidated, and a CONTEXT_STATE message is sent to the Decompressor.

2.3. Robust Header Compression (RoHC)

The header compression concepts and mechanisms for error detection and recovery described in the previous section are not considered robust because they:

- do not perform well on links with high error ratios and/or long round trip times (e.g. wireless links);
- do not take into account that some applications may actually benefit from delivering packets with errors;
- assume that the lower link will not reorder packets (only ECRTP handles this problem);

RObust Header Compression (RoHC) [1] emerged from the need to standardize a single, solid and extendable header compression protocol that performed well over links with high error rates and long link round trip times, taking into account the problems shown by its predecessors.

2.3.1. RoHC Architecture

RoHC is a protocol for compressing the headers of IP-based protocols. Similarly to the other header compression schemes, the RoHC function is distributed. Each RoHC Instance performs one of two functions: compressing and decompressing. The instances responsible for compressing are called *RoHC Compressor Instances*, and decompressing is accomplished by the *RoHC Decompressor Instances* [11].

The end-to-end path between two RoHC instances is referred to as *Channel*. A Channel is a communication path abstraction between two IP interfaces (see Figure 2 and Figure 3). The nature of the channel depends on its directionality (unidirectional or bidirectional), on the number of channels available between two interfaces, on the type of links (point-to-point or point-to-multipoint), and on the management of the channels.

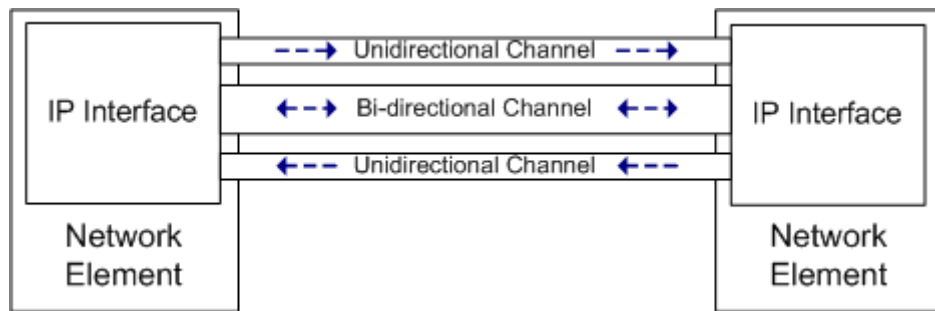


Figure 2 - Channel Abstraction (using point-to-point links)

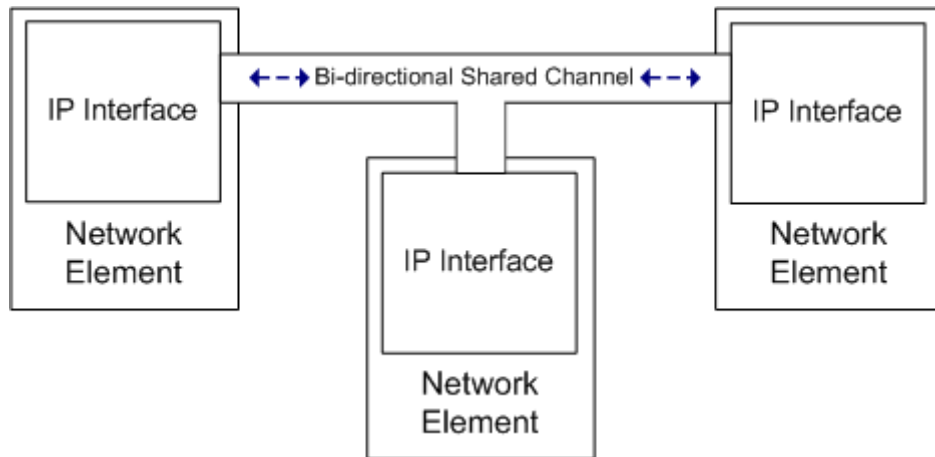


Figure 3 - Channel Abstraction (using point-to-multipoint links)

An IP interface can have multiple RoHC Compressor and Decompressor Instances. The RoHC Compressor and Decompressor engaged in a RoHC conversation are called *Corresponding RoHC Instances*. If a link transports packet streams in both directions, it may be desirable to compress packets in both directions. Thus, both a RoHC Compressor and Decompressor Instances are needed at each end of the link. The aggregation of both instances in a RoHC peer is called a *RoHC Co-located Compressor/Decompressor*.

A RoHC Compressor Instance receives and sends data through three inputs and one output:

- Uncompressed Input (UI): the higher layers feed the RoHC Compressor with uncompressed packets.
- Compressed Output (CO): this is the result of the RoHC header compression.
- Feedback Input (FI): the Compressor may optionally receive feedback packets from the corresponding RoHC Decompressor Instance. This is possible only if there is a feedback channel between the RoHC Decompressor and the RoHC Compressor Instances.
- Piggybacked Input (PI): if the Compressor is associated with a co-located Decompressor, feedback packets may optionally travel piggybacked on the RoHC packets flowing in the opposite direction.

A RoHC Decompressor Instance receives and sends data through one input and three outputs:

- Compressed Input (CI): where the compressed packets are received by the Decompressor. It is always connected to the CO of the Compressor.
- Decompressed Output (DO): the result of decompressing compressed packets. The decompressed packets are delivered to higher layers.

- Feedback Output (FO): the Decompressor may optionally return feedback to the Compressor using a feedback channel. There must be a feedback channel available.
- Piggyback Output (PO): the Decompressor may optionally return feedback to the Compressor by piggybacking feedback headers in RoHC packets traveling between both RoHC peers, but in the reverse direction. This is only possible if the underlying channel is bidirectional.

For simplicity, the RoHC Compressor Instance and RoHC Decompressor Instance will be hereafter referred as RoHC Compressor/RoHC Decompressor, or simply as Compressor/Decompressor.

A *Channel* is a point-to-point bidirectional connection between IP interfaces of two network elements (NE), which means that the two NEs are L3 next-hops. The RoHC Channel is unidirectional, which means that two RoHC Channels are needed to exchange RoHC packets in both directions of a bidirectional Channel. The RoHC Channel is associated with a single RoHC Compressor Instance CO's and to a single RoHC Decompressor Instance CI's. The Feedback Channel is associated with the FO or PO of a single RoHC Decompressor Instance and with the FI or FO of a single RoHC Compressor Instance.

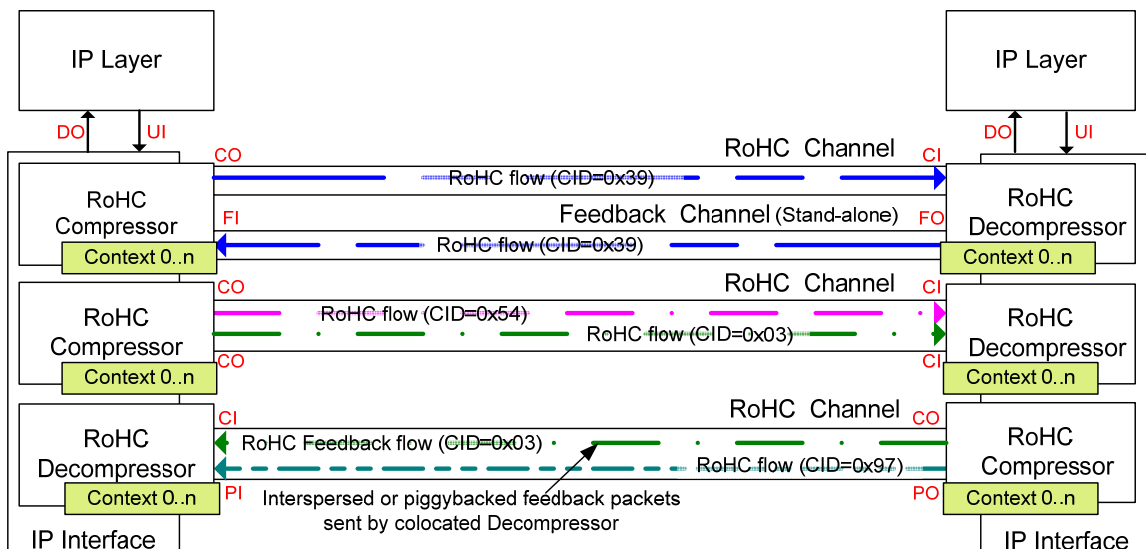


Figure 4 - RoHC Channels and Peers

A limited number of packet streams, known as RoHC flows, are contained within each RoHC Channel. For each flow a context is maintained by both the Compressor and the Decompressor, which contains a set of static and inferable fields. Each context is identified by a Context ID (CID) which is included in the header of all RoHC packets.

The first RoHC packet in a RoHC flow is always sent with full header to allow both the Compressor and Decompressor to build the context for the flow. After this, the Compressor may suppress static and inferable fields from subsequent packets, transmitting only dynamic fields and delta values (only in cases where delta encoding is used).

A packet that arrives at the Compressor is first classified and associated with an existing CID. The classification is made by grouping packets that share some properties, such as having the same values for some fields (e.g., IP addresses, ports, etc). However, the RoHC standard [1] does not define how packet classification is performed and, thus,

any RoHC implementation may choose the packet classification method it finds more appropriate. After the classification, the packet header is compressed, a RoHC CID field of one or two bytes is added, and the RoHC packet is sent through the RoHC channel towards the Decompressor.

If a bidirectional header compressed communication between the two nodes is desired, then another header compressed flow is needed in the reverse direction, using the same or a different channel. In this case the nodes switch functions, i.e. the Compressor node in one flow direction becomes the Decompressor node in the reverse direction, and vice-versa. RoHC uses a distinct context identifier space per channel. If the two flows share the same channel, they must have distinct CIDs. This restriction does not apply if they use separate channels. Within a node, a RoHC flow is identified by the tuple (RoHC channel, CID). Although a pair of RoHC flows may be used by a single bidirectional service (e.g. an FTP session), there is no correlation between them. They are independent RoHC flows with separate CIDs and context information.

There are two types of error propagation: damage propagation, and loss propagation. The delivery of incorrect decompressed headers, due to errors in previous headers or feedback is named damage propagation; the loss of headers due to errors in previous headers or feedback is called loss propagation.

2.3.2. Header Compression Profiles

Efficient header compression for a RoHC flow requires optimizations that consider the variance of each header field and knowledge of the protocols involved. The variance of a field defines the probability of that field change between consecutive packets of the same RoHC flow. Fields that change often/unpredictably must be sent frequently in order to maintain the context of the Decompressor synchronized; Predictable variations can be sent less frequently.

Three dimensions of redundancy are explored: 1) redundancy between header fields of the same protocol; 2) redundancy between header fields of different protocols transported together; and 3) redundancy between header fields of consecutive packets. This multi-header, multi-packet aware compression scheme is what makes RoHC more robust, and more compression efficient than its predecessors. In RoHC, the strategy for compressing specific traffic types is specified in *Header Compression Profiles* (HCP).

The main RoHC specification [1] defines four HCPs:

- Profile 0x0000, for sending uncompressed IP packets;
- Profile 0x0001, for sending compressed IP/UDP/RTP packets. This is the main RoHC compression profile;
- Profile 0x0002, for IP/UDP compression (the first 12 octets of the UDP payload are not compressed);
- Profile 0x0003, for sending IP/ESP compressed packets;

There are profiles defined in other IETF RoHC Working Group documents:

- Profile 0x0004, for plain IP header compression [15];
- Profile 0x0005, a Link-Layer Assisted (LLA) profile, an extension to profile 0x0001 to provide header free compression for channels that cannot accommodate packets of arbitrary sizes [17];
- Profile 0x0006, for IP/TCP compression [19];
- Profile 0x0007, for IP/UDP Lite/RTP compression [16];
- Profile 0x0008, for IP/UDP Lite compression [16];
- Profile 0x0105, an extension to Profile 0x0005, adding 0 byte support for R-mode [18]

An HCP classifies each header field of the compressed headers in one of the following variance classes:

- **STATIC:** fields which are expected to be constant throughout the lifetime of the RoHC flow. There are two subtypes of static fields: those who are always known and are never transmitted; those who must be transmitted at least once, to initialize the contexts.
- **CHANGING:** fields which are expected to have some level of variance. It could be random or contained within a limited value set or range.
- **INFERRED:** fields whose value can be inferred or calculated using other values, and thus do not have to be handled by RoHC.

The HCP also defines how and when transitions occur in the RoHC state machines.

2.3.3. Lower Layer Requirements

The HCP does not specify how the RoHC packets are transported along the end-to-end layer 2 path. There is some functionality that must be provided by the underlying layer(s) and, thus, there are some details that are lower-layer specific.

The minimal set of functions that RoHC requires from the lower-layers [9] is the following:

- **Error detection:** almost every header compression scheme relies on lower layers for error detection [3][4][5]. This is usually achieved through the use of a lower-layer checksum mechanism.
- **Packet length indication:** RoHC must know the size of each packet, to infer fields such the IPv4 Packet Length or the IPv6 Payload Length. These fields are marked as inferable by HCP, and are never transported. Therefore, the underlying lower layer must be able to provide this information.
- **Handling of different packet sizes:** each packet may have a different header compression ratio and payload size. The lower-layer must be able to support the transportation of multiple packet sizes.
- RoHC also relies on the lower-layer not to duplicate or reorder packets between the RoHC Compressor and Decompressor.

Optionally, the lower-layer may also provide:

- **Packet type indication:** if the RoHC packets are mixed with other type of traffic, the lower layer must provide packet type indication. A single lower-layer packet type is sufficient since RoHC already includes its own packet identifier.
- **Other inferred fields:** if some fields can have their values inferred from lower-layer fields, this information is considered to achieve higher compression efficiency.
- **Mobility and handover enabling mechanisms:** if RoHC is used in a mobile environment (i.e. radio links), it will require certain mechanisms and information from lower-layers, such as the signal strength.

The lower layer may also impose some requirements on RoHC, such as the path Maximum Transmission Unit (MTU). RoHC includes a simple segmentation and reassembly mechanism which enables it to comply with any MTU size. This mechanism does not include sequence numbers, and thus, it relies on the lower layer being able to

deliver packets in order. The required changes for RoHC Decompressors to work correctly over channels that reorder packets are defined in [14].

The fulfilment of these requirements is addressed by specific documents *RoHC-over-X*, where *X* is the underlying technology. Each document also specifies the channel negotiation, feedback mechanisms, and other optimization aspects. Up to now the IETF RoHC Working Group has defined only RoHC-over-PPP [13].

2.3.4. RoHC Channel Negotiation

Before exchanging RoHC packets, the two RoHC peers must negotiate a *channel state*. The RoHC standard recommends that the link layer shall provide an out-of-band mechanism to support this negotiation. If this is not possible, the channel state can also be defined statically at both endpoints, which is useful in scenarios having no feedback channel.

The parameters of the negotiated channel state [1] are:

- MAX_CID: the maximum Context Identifier number that can be used by the Compressor.
- LARGE_CIDS: the size of CID field to be used (1 or 2 bytes).
- Set of header compression profiles accepted by the Decompressor.
- MRRU, or Maximum Reconstructed Reception Unit: the maximum size in octets of packet, including CRCs, reconstructed from RoHC segments allowed by the Decompressor.
- Optionally, a reference to a channel in the reverse direction that can be used to send feedback.

There are also some RoHC flow parameters which are stored in a per-context state:

- The HCP currently in use for the RoHC flow.
- Predefined values for some fields stored in the context (optional).

2.3.5. Header Compression States

The Compressor and the Decompressor behaviours are described by two state machines, each containing three states. Both machines start in the lowest compression state and advance to higher compression states. Transitions are based on events, which are not necessarily synchronized between the two machines. The state transitions are caused by the reception of feedback messages from the Decompressor, error detection, variations in the packet header fields, and timeouts. Transitions also depend on the HCP in use.

2.3.5.1. Compressor States

The RoHC Compressor state machine contains the *Initialization and Refresh* (IR) state, the *First Order* (FO) state, and the *Second Order* (SO) state, as shown in Figure 5. The IR State is the lowest compression state, and the SO State is the highest compression state.

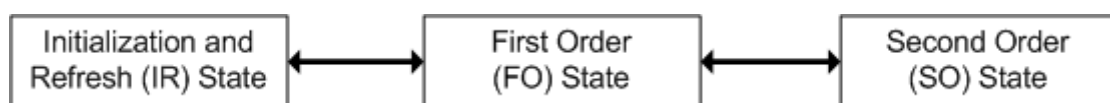


Figure 5 - RoHC Compressor state machine

When in the IR state, the Compressor sends a full header RoHC packet, including uncompressed static and non-static fields, and other information such as the RoHC CID. Its purpose is to initialize the context at the Decompressor or to recover after a failure. After being confident that the context was correctly built at the Decompressor, the Compressor steps forward to the FO state. This confidence level is defined by the specific HCP in use. It is usually expressed in number of packets sent in the IR state.

In the FO state the Compressor communicates the variations in the RoHC flow. The static fields are usually not sent, and most of the dynamic fields are sent in a compressed form, usually using delta encoding. The compressor enters this state from the IR state, or from the SO state when the headers of the flow do not conform to their previous pattern. There are some fields which are always irregular (e.g. CRC) and therefore are always sent. The Compressor remains in the FO state until it is confident that the Decompressor was able to learn the parameters of the new pattern. If an error is detected, it reverts back to IR state.

Once in the SO state, the Compressor achieves the maximum compression ratio. The Compressor enters in this state only when the headers show an uniform pattern and are fully predictable. Using the context information, the irregular header fields and a sequence number used as an index for the progression of the header field values, are transmitted. When the uniform pattern breaks, the Compressor falls back to the FO state.

The Compressor takes the initiative of changing between compression states when it detects variations in the header fields, and it anticipates possible synchronization errors in the Decompressor. Nevertheless, there are additional mechanisms for error detection. In certain cases, if the Decompressor detects an error it may send feedback to the Compressor (i.e. ACK or NACKs) to signal incorrect packet decompression. This enables faster error recovery. If a feedback channel is unavailable, the Compressor periodically reverts back to lower compression states, which allows the Decompressor to synchronize again.

2.3.5.2. Decompressor States

The RoHC Decompressor state machine contains the *No Context*, *Static Context* and *Full Context* states, shown in Figure 6.



Figure 6 - RoHC Decompressor state machine

The Decompressor always starts in the No Context state. The context is built upon on the correct reception of a full header packet, sent by the Compressor in IR state. When this happens, the Decompressor advances to Static Context state, and from there, to Full Context state. The Decompressor will remain in Full Context state unless repeated failures occur; in this case, it goes back to Static Context state where the reception of a packet sent in the FO state enables the transition to the Full Context state. If the transmission of several packets in FO state fails, then the Decompressor reverts to the No Context state.

2.3.6. RoHC Packets

The general structure of a RoHC packet is illustrated in Figure 7.



Figure 7 – General RoHC Packet Structure

Padding is sometimes needed because some link layer technologies impose a minimum packet size; Ethernet, for instance, has a minimum frame size of 64 bytes. The padding section contains 0 or more padding octets containing value 0xE0. RoHC also includes a simple segmentation and reassembly mechanism.

After Padding, a Feedback, an Header, or both sections, must be present. RoHC feedback information can travel piggybacked on a normal RoHC packet, or in a standalone RoHC Feedback packet. A feedback packet can be one of the following types: ACK, to acknowledge the correct reception and decompression of a compressed header packet; NACK, to signal that the dynamic context of the Decompressor is out of synchronization; STATIC-NACK, to signal that the static context of the Decompressor is invalid. The feedback can also include profile-specific information.

A CID of 1 or 2 octets is embedded in the Header. Header can be an IR, an IR-DYN, or a specific profile header. The RoHC IR packet is used to transmit the static parts of the context; it associates a CID with a compression profile and initializes the context. The RoHC IR-DYN is similar to the IR but cannot be used to initialize a context, since it transmits only the dynamic part of the context; it can be used to redefine the profile associated with the context.

In order to maximize compression efficiency, each compression profile has its own packet types, each one used only in specific states. Each compression profile adds three packet types: Packet Type 0, Packet Type 1 and Packet Type 2. The name indicates the amount of information transmitted. The higher the number, the higher is the number of bytes sent. Packet Type 0 is used only for higher compression states. The format of the packets [1] depends on the compression profile.

The RoHC payload includes the IP-based headers not considered by the HCP in use, and the IP payload.

2.3.7. Operation Modes

RoHC can operate in three modes: Unidirectional Mode (U-mode), Bidirectional Optimistic (O-mode), and Bidirectional Reliable (R-mode). States and modes are orthogonal. The states are the same for all modes of operation; the modes control the transitions between the states, and specify the actions to perform in each state.

An operation mode can be used at anytime; the mode depends on the channel characteristics. The channel characteristics are the static properties of the channel such as bidirectionality, and error probability, but the later may change with the time. Simultaneously to the state machine transitions, and based on the same events that cause them, RoHC can also switch between operation modes.

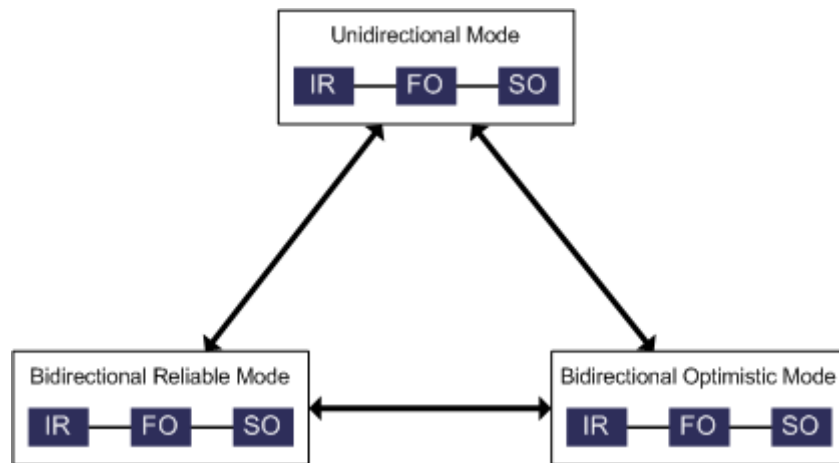


Figure 8 - RoHC Operation Modes

2.3.7.1. Unidirectional Mode (U-mode)

When RoHC is operating in the Unidirectional Mode only the Compressor to the Decompressor direction is used. This mode is appropriate for RoHC over links where a return path from the Decompressor to the Compressor is unavailable or undesirable.

A Compressor always starts in the IR state, but can perform the transition to the FO state as soon as it is confident that the Decompressor was able to receive correctly and to decompress a packet. Similarly, the Compressor can advance to the SO state when it feels that the Decompressor could receive correctly the compressed packets and could detect a pattern in the header fields of the flow. In both cases, the confidence is expressed by the number of packets sent in the previous compression state. The RoHC standard does not specify values for these parameters.

A single direction is used, so the Decompressor cannot feedback the Compressor to initiate error recovery; as a consequence the Compressor cannot detect if the Decompressor is out of synchronization. In order to overcome this problem, the Compressor analyzes the headers of the source flow and, in case of irregularities in the header pattern, makes a transition to a lower compression ratio state, forcing a context update. The Decompressor can get out of synchronization by other reasons, such as transmission errors, or packet loss. In order to assure that the errors do not result in significant losses of traffic, the Decompressor reverts periodically to a lower compression ratio state, and forces the refreshment of the context information.

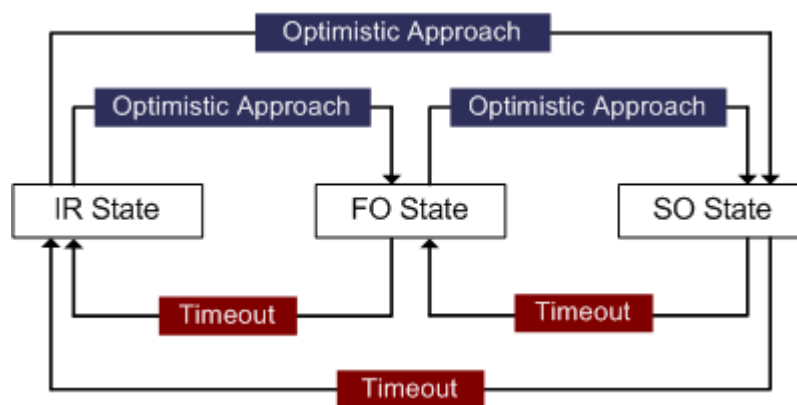


Figure 9 - Compressor states and transitions in U-mode

Figure 9 illustrates the Compressor states and transitions when operating in U-mode. It uses optimistic assumptions to make the transition to higher compression states, and uses context updates and timeouts to make transitions to lower compression ratio states.

Figure 10 represents the Decompressor states and transitions when operating in the U-mode. The correct reception of a compressed packet moves the Decompressor forward to the Full Context state; in the case of a decompression failure, it makes a transition back to a lower compression state.

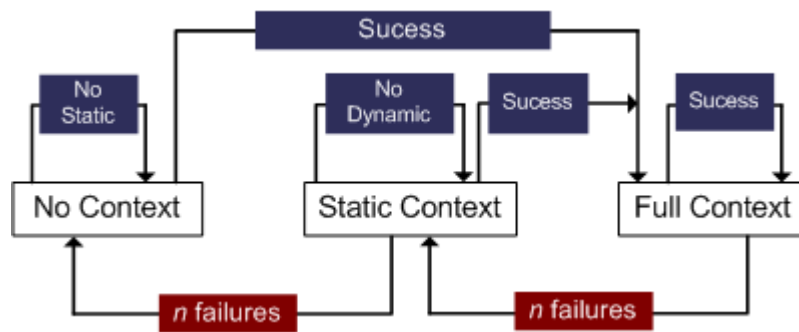


Figure 10 - Decompressor states and transitions in U-mode

This mode enables the use of RoHC header compression in unidirectional environments, but also makes RoHC less efficient and more vulnerable to loss propagation than in the bidirectional modes. The compressor always starts in the U-mode, but it transits to a bidirectional mode as soon as it receives a feedback packet from the Decompressor. When an unidirectional channel is used, the Compressor remains in U-mode.

2.3.7.2. Bidirectional Optimistic (O-mode)

The Bidirectional Optimistic mode (O-mode) differs from the U-mode only by the bidirectionality. The Decompressor sends error recovery requests (NACK) and, optionally, acknowledgments (ACK) of significant context updates through a feedback channel. The Compressor transits to a lower compression state when needed, eliminating the need for periodic refreshes, minimizing the use of the feedback channel and, as a result, maximizing the compression efficiency. In the presence of burst errors, the O-mode is more vulnerable to invalid contexts. In Figure 11 we can observe that the O-mode may still use an optimistic approach to move to higher compression state; but it does not use timeouts in the downward transitions. When a context update is needed, the Decompressor sends a NACK or a STATIC-NACK. ACKs trigger transitions upwards.

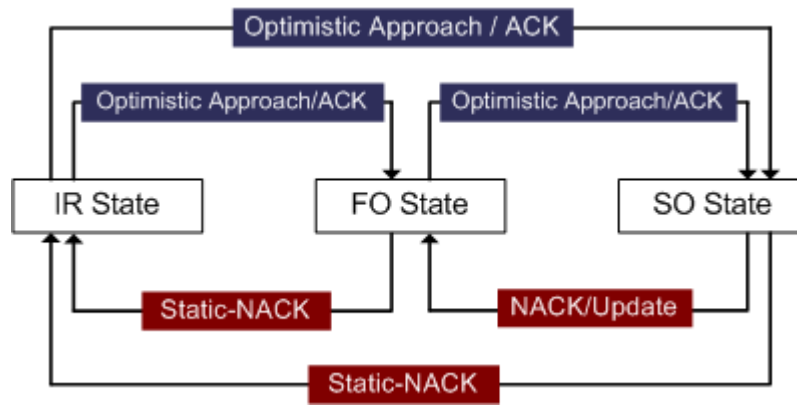


Figure 11 - Compressor states and transitions in O-mode

The Decompressor states and transitions in the O-mode are similar to those in the U-mode.

2.3.7.3. Bidirectional Reliable (R-mode)

The Bidirectional Reliable mode (R-mode) offers further protection against loss and damage propagation. It makes an intensive usage of the feedback channel, as all context updates are acknowledged. Under high BERs loss of context synchronization between the Compressor and the Decompressor may still occur, and a larger number of damaged headers may be delivered.

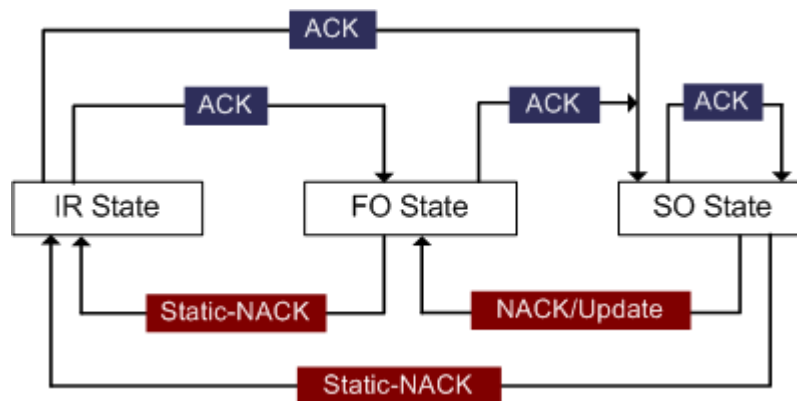


Figure 12 - Compressor states and transitions in R-mode

The Compressor states and transitions in the R-mode are different from the other modes. Upwards transitions are similar to those in U-mode, but the confidence is obtained only by the ACKs sent by the Decompressor. This guarantees that the context will not get out of synchronization. Downward transitions make use of the feedback channel as in the O-mode. Similarly to the O-mode, NACKs are sent when the context gets out of synchronization, which in R-mode is a rare event. The Decompression logic is similar to the U-mode's logic, except that feedback is used, which guarantees that only decompressible packets are sent to the Decompressor.

2.3.7.4. Mode Transitions

The decision of changing the operation mode is always taken by the Decompressor. The reception and correct decompression of a packet by the Decompressor may cause the Decompressor to send a feedback packet requesting a change of operation mode. Figure 13 illustrates this concept. The operation mode is

often included in the RoHC and feedback packets. The Compressor and the Decompressor always start in U-mode.

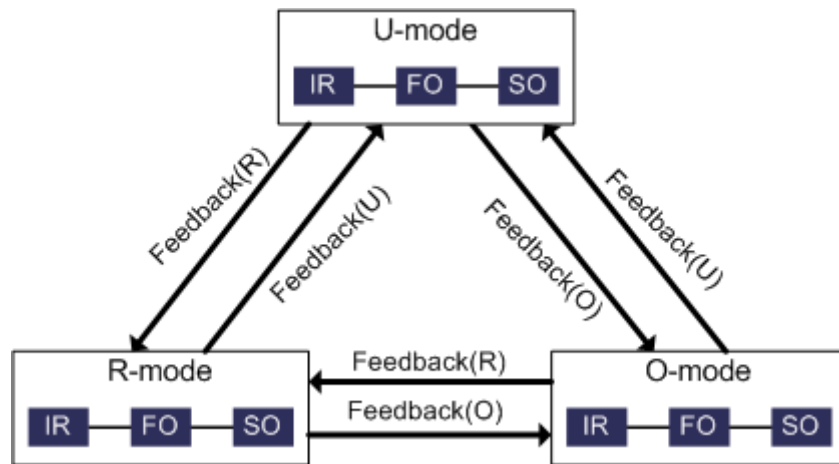


Figure 13 – Operation mode transitions

During mode transitions the behaviour of the Compressor and Decompressor are restricted to a safe set of actions in order to minimize the probability of they get out of synchronization. For instance, the feedback packets sent during an operation mode transition must be protected by a CRC; if not, the transition request is ignored by the Compressor. During a transition the Compressor uses only packet types which are common to all operation modes.

2.4. Conclusions

This chapter has presented the state of the art in IP header compression. This included a brief description of early header compression protocols, namely Thinwire-II, cTCP, IPHC, CRTP and ECRTTP. These protocols do not perform well on links characterized by high error ratios or long round trip times and, as such, they are not considered robust. RoHC emerged from the need to standardize a single and extendable header compression protocol that performed well over such links.

Before exchanging RoHC packets, the two RoHC peers must negotiate a RoHC Channel. Context information stored in the RoHC endpoints is identified by a CID. This CID is also used to identify individual flows in a RoHC Channel. The strategy for compressing specific traffic types is specified in *Header Compression Profiles* (HCP). An HCP also determines the RoHC packet structure, and the behaviour of the Compressor and of the Decompressor state machines. The main HCPs defined in the RoHC standard are the IP/UDP/RTP, IP/UDP, IP/ESP and IP/TCP. RoHC specifies three operation modes: the U-mode, the O-mode and the R-mode. The modes control the transitions between the states, and specify the actions to be executed in each state.

RoHC is considered robust under high BERs, which may be typical of some wireless links. RoHC is profitable in wireless networks, and for traffic characterized by small payloads. These are some of the characteristics of 4G networks, where terminals may use multiple L2 technologies, some with limited resources available. Currently, there is not a single solution to transport RoHC packets over wireless and wired heterogeneous technologies.

3. Header Compression in Cellular Networks

3.1. Introduction

Cellular networks have grown intensively used during the last years. Voice services still represent the significant portion of the traffic exchanged, but the usage of data services is increasing. The Third Generation Partnership Program (3GPP) [23] is responsible for the standardization of European cellular networks, namely, GSM/GPRS (2G/2.5G) and UMTS (3G). 3GPP has been working, among other topics, to improve the transmission efficiency. Header compression, contributes to bandwidth savings, and has been integrated in the 3GPP specifications since early stages.

This chapter describes the evolution of 3GPP networks, from GSM to UMTS, and presents the header compression mechanisms used in these technologies.

3.2. GPRS

The General Packet Radio Service (GPRS), commonly referred as a 2.5G technology, extends the existing General System for Mobile communication (GSM) [22] networks with the possibility to transport packet based services.

3.2.1. GPRS Architecture

The GPRS is an extension to the GSM standard, and so it inherits and extends its architecture. The GSM architecture is composed by the Base Station Subsystem (BSS) and the Network and Switching Subsystem (NSS) (see Figure 14). The BSS includes the Mobile Stations (MS), the Base Transmission Stations (BTS), and the Base Station Controllers (BSC). The BSC controls one or more BTS, and each BTS may serve one or more cell areas. The NSS includes Mobile Switching Centers (MSC), to whom the BSC are connected to, the Gateway Mobile Switching Center (GMSC), which is the gateway to external networks, the Home Location Register (HLR), the Visited Location Register (VLR), the Authentication Center (AuC), and the Equipment Identity Register (EIR).

The HLR is a central database which contains data from the users of a given operator, such as the user profiles. To optimize the access to this information, upon call setup, this data is cached in the nearest VLR. The VLR also stores temporary identification which is attributed to the user before authentication. The AuC generates and stores security-related data such as keys used for authentication and encryption. The EIR is responsible for authenticating the user equipment..

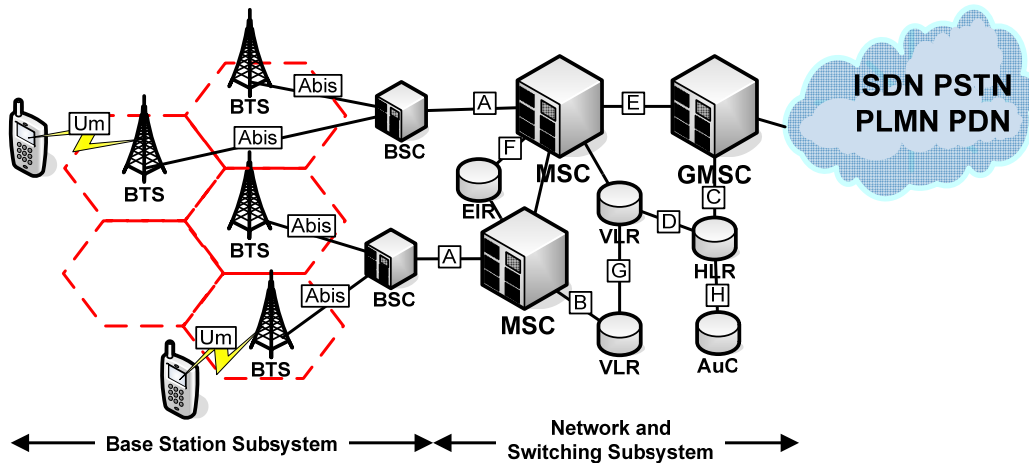


Figure 14 - GSM Architecture

GPRS introduced packet switching capabilities to the GSM Networks. GPRS extends the GSM architecture by adding two new network elements: the Serving GPRS Support Node (SGSN), and the Gateway GPRS Support Node (GGSN), shown in Figure 15. The SGSN is responsible for the delivery of data packets to the mobile stations within the area it serves; it also handles mobility management, and authentication and charging. The SGSNs are interconnected by a GPRS backbone where packets are routed. The GGSN acts as an interface between the GPRS backbone and external Packet Data Networks (PDN). It is responsible for converting Packed Data Protocol (PDP) formats.

The SGSN is connected to BSC using Frame Relay (FR), while in the backbone network, SGSNs and GGSNs are interconnected by an IP network. Networks from different providers can be interconnected by Border Gateways (BG), which are responsible for employing policies and exchanging routing data.

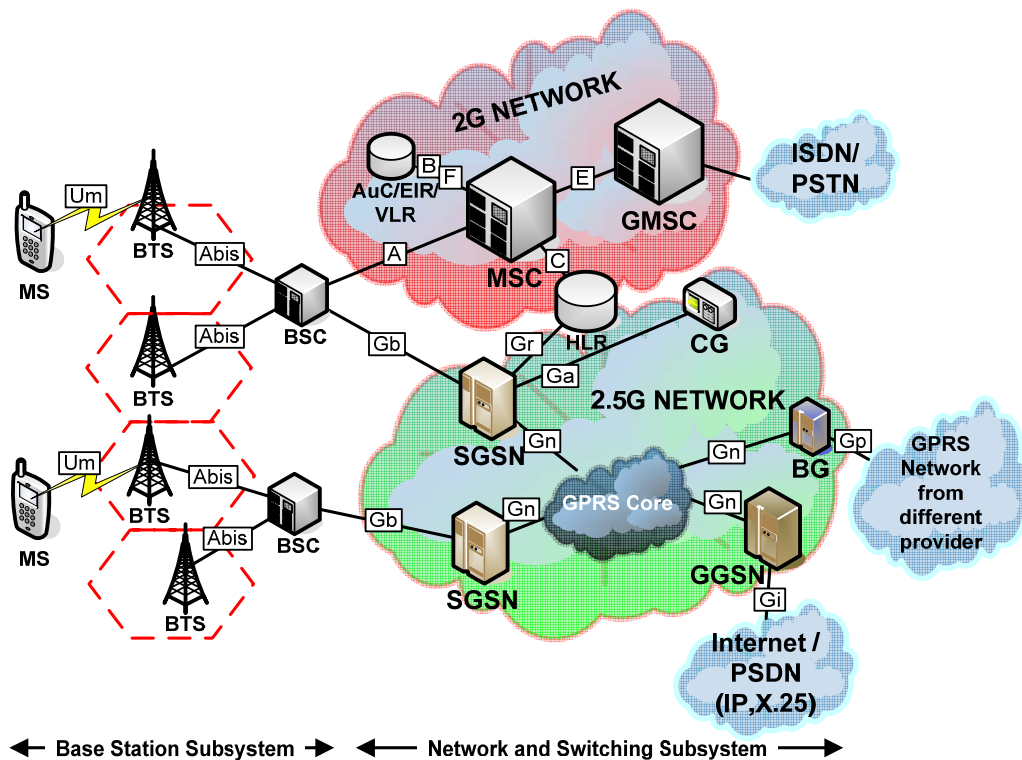


Figure 15 - GPRS Network Architecture

3.2.2. Protocol Architecture

Figure 16 illustrates the end-to-end GPRS protocol stack between the MS and the GGSN in the user plane. The BSS and the NSS parts are interconnected by the Gb interface. The BSS can also be regarded as having two independent subsections: the wireless part, between the MS and the BTS, and the wired part which connects the BTS to the BSC and then to the SGSN.

When considering the end-to-end transmission of data packets, convergence is provided at network level, by IP, although X.25 can also be used. Below IP, different transport protocols are used within the BSS and the NSS. In the BSS, the SubNetwork Dependant Convergence Protocol (SNDCP) [24] transports IP packets between the MS and the SGSNs. Within the core GPRS Network, IP packets are encapsulated by a tunnelling protocol named GPRS Tunnelling Protocol (GTP). The SNDCP is responsible for the aggregation of multiple network connections, named Packed Data Protocol (PDP) Contexts, into one virtual logical connection which is then carried by the data link layer. To be able to send packets, a MS needs first to attach to the GPRS network. Then, it must create and activate a PDP Context to the GGSN, specifying a local *connection ID* (CID), a *PDP_Type* (IP/X.25), and the address of the GGSN. This PDP can be seen as a tunnel between the MS and the GGSN. The SNDCP at the MS is able to manage multiple simultaneous PDP Contexts, to the same or different GGSN, with the same or different QoS profiles. The SNDCP is also responsible for the data and header compression function.

The Data Link Layer between the MS and the BSS is divided in two sub layers: the Logic Link Control layer (LLC) [25], between the MS and the SGSN; and below, the Radio Link Control (RLC)/MAC [26][29] between the MS and the BSS, and the BSS GPRS Application Protocol (BSSGP) between the BSCs and its assigned SGSN. The LLC provides a reliable logical link. It is based on the well known HDLC protocol [32] and includes sequence control, ordered delivery, flow control and detection and retransmission of erroneous frames. Data confidentiality can also be provided by using LLC's ciphering functions. LLC supports both acknowledged and unacknowledged data transmission modes and variable length frames. The main purpose of the RLC/MAC layers at the air interface is to establish a reliable link between the MS and the BSS. This includes the segmentation and reassembly of LLC frames into RLC data blocks, and retransmission of erroneous data blocks. Below MAC, at physical layer, the GSM Radio Frequency (GSMRF) protocol is used for wireless access. The BSSGP handles the flow of LLC PDUs between the BSS and the SGSN. The underlying network service is based on the Frame Relay protocol [33].

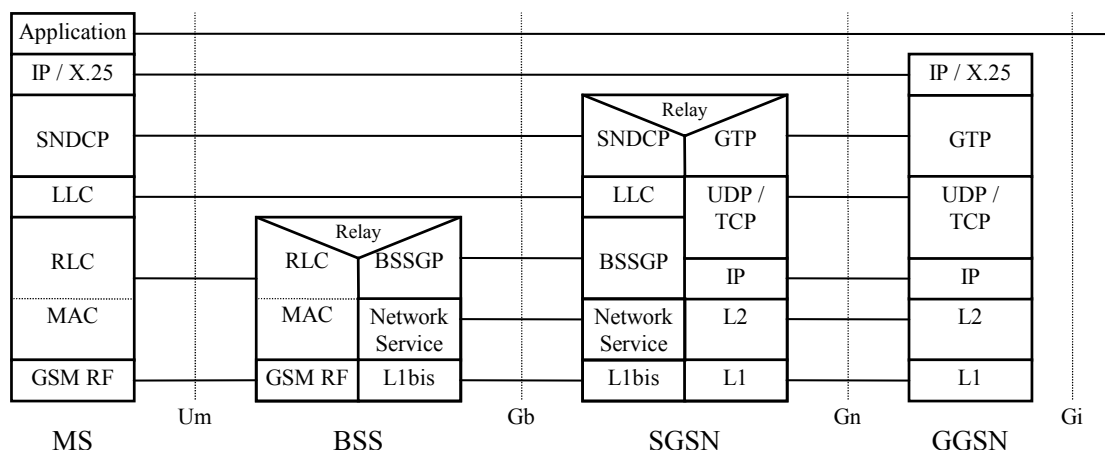


Figure 16 - End-to-end GPRS protocol stack between the MS and the GGSN (user plane)

In the core GPRS network, GTP packets are transported by IP packets upon an arbitrary L2 technology. In early versions of GPRS, ATM [34] was the main L2 technology, but it was replaced later on by Ethernet [35] or even MPLS [38] over Ethernet, due to Ethernet's reduced cost and MPLS flexibility.

3.2.3. Header Compression in GPRS

Header compression in GPRS is performed at the SNDCP layer, between the MS and the SGSN, as shown in Figure 16. The SNDCP is responsible for multiplexing multiple PDPs, compressing and decompressing headers and data, segmenting Network PDUs (N-PDU) into LLC PDUs, and reassembling LLC PDUs into N-PDUs.

Before being able to use header compression, the MS and the SGSN negotiate if header compression can be applied, and the header compression protocols to be used. The negotiation of one or more header compression protocols, designated as *compression entities* in SNDCP, is carried out using the SNDCP XID parameter negotiation stage. The initiating entity builds a list of supported header compression protocols and, for each, the header compression parameters desired. The list is sent to its SNDCP peer, which responds with another list. If multiple header compression mechanisms are proposed, the receiving peer chooses the most appropriate.

Bit	8	7	6	5	4	3	2	1
Octet 1	P	X	X	Entity number				
Octet 2	X	X	X	Algorithm identifier				
Octet 3	Length=n-3							
Octet 4	PCOMP1				PCOMP2			
...	...				...			
Octet n	...							

Figure 17 - SNDCP XID negotiation packet structure

Figure 17 illustrates the structure of the SNDCP XID negotiation packet. The X bits, or *Spare bits*, should be set to 0, and it is ignored by the receiving SNDCP entity. The P, or *Propose bit*, is set to 1 if a new compression entity is being proposed; otherwise it is set to 0. If the P bit is set to 1, then all of the octets in Figure 17 must be included. The PCOMP (Protocol control information COMPression) values are header compression parameters. The number of parameters depends on the used header compression mechanisms. If an odd number of PCOMP values are needed, then a PCOMP value should be set to 0, and ignored by the receiving SNDCP entity. The entity number is a value between 0 and 31 used to identify the compression entity; this effectively limits to 32 the number of different compression protocols that can be used.

Entity Number	Compression mechanism
0	cTCP
1	IPHC
2	RoHC
Other values reserved	-

Table 1 - Header Compression Mechanisms supported by SNDCP

Table 1 enumerates the header compression mechanisms currently supported by SNDCP. Header compression was introduced in GPRS in the release 98, when cTCP

support was added. IPHC was introduced in GSM/GPRS release 99, and RoHC later on in release 6. Note that these header compression mechanisms apply only to IP traffic.

3.2.3.1. cTCP for GPRS

Table 2 contains the parameters defined for cTCP compression, which may be negotiated with SNDCP XID.

Parameter Name	Format	Range	Default Value
Applicable NSAPIs	bbbbbbbbb bbb00000	0, 32, 64, ... 65504	0
$S_0 - 1$	bbbbbbbbb	0 through 255	15
Uncompressed TCP	bbbbbbbbb	0 through 255	1
Compressed TCP	bbbbbbbbb	0 through 255	2

Table 2 - cTCP header compression parameters (PCOMPs)

The first parameter, Applicable NSAPIs, specifies the Network Service Access Point Identifiers (NSAPI) that can be used by SNDCP clients. The second parameter, $S_0 - 1$, represents the number of available state slots, i.e. the number of saved packet headers that cTCP maintains in the header compression context. cTCP needs to distinguish among three possible types of compressed N-PDUs defined in [3]: Type IP (0), Uncompressed TCP (1), and Compressed TCP (2). Type IP N-PDUs is associated with the value 0. The values of other types are negotiated via the “Uncompressed TCP” and “Compressed TCP” PCOMPs, respectively.

LLC provides two frame delivery services: acknowledge and unacknowledged frame delivery. If the former is used, when LLC frames are dropped the compression entity may be informed, so that error recovery procedure can be initiated, as described in Section 4.2 of [3].

3.2.3.2. IPHC and GPRS

Table 3 contains the parameters defined for the IPHC compression:

Parameter Name	Format	Range	Default Value
Applicable NSAPIs	bbbbbbbbb bbb00000	0, 32, 64, ... 65504	0
F_MAX_PERIOD	bbbbbbbbb bbbbbb	1-65535	256
F_MAX_TIME	bbbbbbbbb	1-255	5
MAX_HEADER	bbbbbbbbb	60-255	168
TCP_SPACE	bbbbbbbbb	3-255	15
NON_TCP_SPACE	bbbbbbbbb bbbbbb	3-65535	15
Full Header	bbbbbbbbb	1-255	1
Compressed TCP	bbbbbbbbb	1-255	2
Compressed TCP non-delta	bbbbbbbbb	1-255	3
Compressed non-TCP	bbbbbbbbb	1-255	4
Context State	bbbbbbbbb	1-255	5

Table 3 - IPHC header compression parameters

The first parameter has the same meaning of the previous subsection. The other parameters are IPHC specific. F_MAX_PERIOD is the largest number of compressed non-TCP headers sent between full header packets. F_MAX_TIME is the maximum time compressed headers may be sent, before another full header packet is sent. MAX_HEADER specifies the largest header size in octets eligible for compression. TCP_SPACE and TCP_NON_SPACE indicate the maximum CID value for TCP and

Non-TCP traffic. IPHC needs to distinguish among five possible types of compressed N-PDUs: Type IP (0), Full Header (1), Compressed TCP (2), Compressed TCP non-delta (3), Compressed non-TCP (4) and Context State (5). Type IP N-PDUs is associated with the value 0. The values of other types are negotiated via the “Full Header”, “Compressed TCP”, “Compressed TCP non-delta”, “Compressed non-TCP” and “Context State” PCOMPs, respectively.

IPHC can also take advantage of LLC notifications.

3.2.3.3. RoHC and GPRS

RoHC was integrated in the 3GPP release 6. GPRS may have PPP above SNDCP. Instead of developing a new RoHC-over-X document and negotiation scheme, RoHC-over-PPP [13] was adopted. It is used to negotiate the CID size.

Table 4 contains the parameters defined for RoHC compression. As before, the first parameter specifies the applicable NSAPIs. MAX_CID indicates the maximum number of CIDs that can be used by the compressor. The MAX_HEADER indicates the maximum number of octets that can be compressed. Not every RoHC profile makes use of this parameter. The PROFILE parameters specify the RoHC profiles supported by the peer which initiated the negotiation.

LLC notifications in acknowledged mode are not fundamental since RoHC has its own feedback-based channel

Parameter Name	Format	Range	Default Value
Applicable NSAPIs	bbbbbbbbb bbb00000	0, 32, 64, ... 65504	0
MAX_CID	00bbbbbb bbbbbb	1-16383	15
MAX_HEADER	00000000 bbbbbb	60-255	168
PROFILE 1	bbbbbbbbb bbbbbb	0-65535	0
PROFILE 2	bbbbbbbbb bbbbbb	0-65535	0
...	...	...	...
PROFILE 16	bbbbbbbbb bbbbbb	0-65535	0

Table 4 - RoHC header compression parameters

3.3. UMTS

The Universal Mobile Telecommunications System (UMTS) [23] emerged from the necessity of providing high bandwidth to data. It was also required to reuse the concepts and equipments of the core network of PLMNs.

3.3.1. UMTS Architecture

When GPRS emerged, it has introduced mainly architectural changes in the NSS, enabling it for packet routing in cellular network cores. The opposite situation occurred with UMTS, where the radio access part of the network suffered most of the transformations. In order to distinguish GPRS and UMTS radio access networks, the later was named *UMTS Terrestrial Radio Access Network* (UTRAN).

UTRAN introduced two new network elements, which replaced BTSs and BSCs: the Node B's, which is functionally equivalent to the BTS; and the Radio Network Controllers (RNC), which is functionally equivalent to the BSC.

Figure 18 illustrates the integration of GSM/GPRS and UMTS network elements. The network can be divided in two main groups: the Radio Access Network (RAN), where GSM/GPRS MS and UMTS User Equipment (UE) are supported; and

the Core Network (CN) where both domains, circuit switching and packed switching, may coexist.

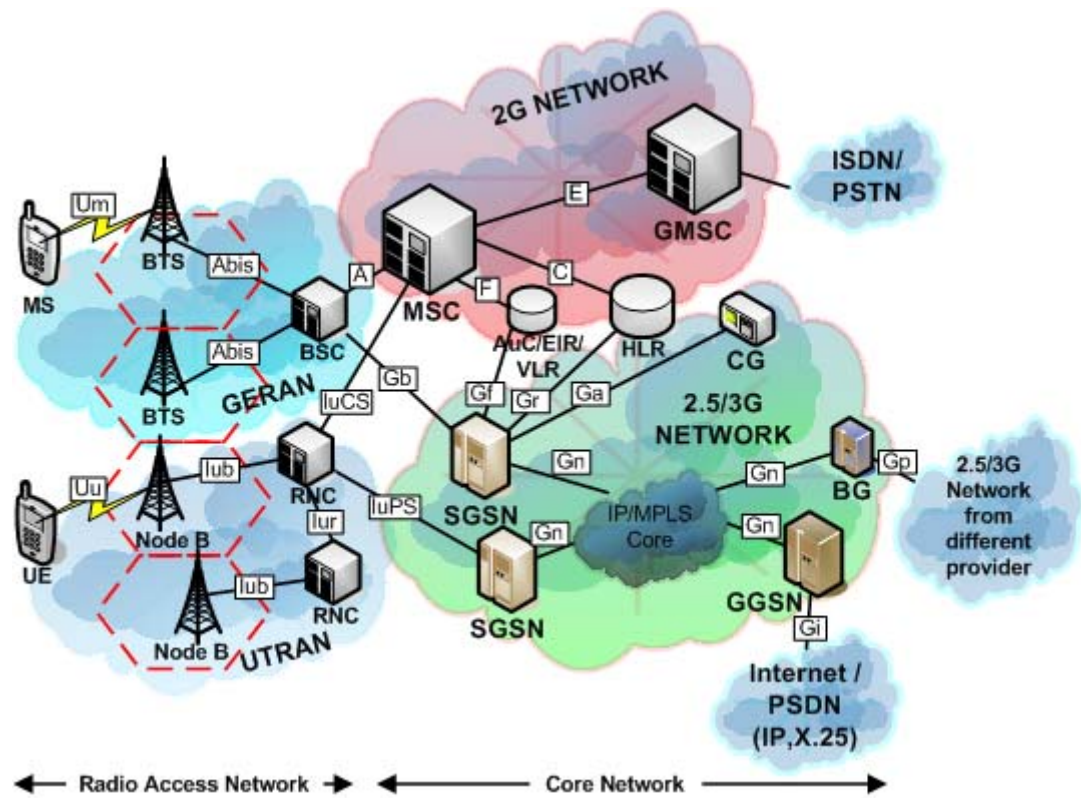


Figure 18 – UMTS Network Architecture

3.3.2. Protocol Architecture

The UMTS protocol stack at the user plane has some differences compared with the GPRS protocol stack. In the UTRAN, the SMDCP was replaced by the Packet Data Convergence Protocol (PDCP). PDCP is responsible for the ordered delivery of user data between the UE and the UTRAN (e.g. interface *Uu*).

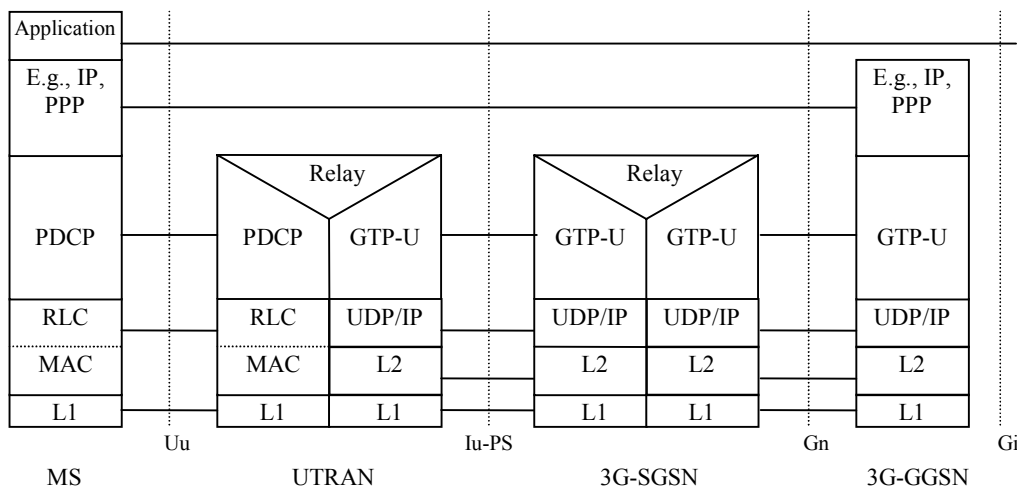


Figure 19 - End-to-end UMTS protocol stack between the UE and the GGSN (user plane)

There are 3 types of PDCP PDUs: PDCP-No-Header PDUs, PDCP Data PDUs, and PDCP SeqNum PDUs. The first does not add any overhead to the PDCP SDUs; the second is used to convey data containing uncompressed PDCP SDUs, header compression signalling, or PDCP SDUs after header compression; the later is used to convey numbered PDCP data PDUs.

Each radio bearer (RB) is associated with one PDCP entity. Each PDCP entity is associated with one or two RLC entities, depending if the RB is unidirectional or bidirectional.

3.3.3. Header Compression in UMTS

Header Compression may be performed at PDCP layer, using either IPHC or RoHC. Each PDCP entity may use zero or more header compression protocols.

All PDCP PDU types, but the PDCP-No-Header, carry a three bit field named *PDU Type*, and a five bit PID (protocol ID) field. The later indicates the header compression protocol type, and the header compression packet type or CID type. PID values are independent of the header compression mechanisms. The PID value “0” is mapped to “no compression”.

3.3.3.1. IPHC and UMTS

According to [28], the IPHC error recovery and packet reordering mechanisms must be supported by PDCP implementations. Table 5 contains the PID mappings for IPHC. “*n*” represents the number of PID values previously allocated to other header compression mechanisms within the same PDCP entity. If no compression mechanisms were negotiated, then *n* has the value of 0.

PID Value	HC mechanism	Packet type
0	No header compression	-
n+1	IPHC	Full Header
n+2	IPHC	Compressed TCP
n+3	IPHC	Compressed TCP non-delta
n+4	IPHC	Compressed non-TCP
n+5	IPHC	Context state

Table 5 - Mappings of PID values for IPHC

The lower layer packet loss indications may be used by IPHC to decide when to send full header packets.

3.3.3.2. RoHC and UMTS

RoHC contexts are identified by the CID. This field is carried in the PDCP header, or in the RoHC packet. The decision between the two alternatives, is made by the upper layers, and affects the mapping of PID values into RoHC. If the CIDs are transported in the PDCP headers, then Table 6 PID mappings apply. Otherwise, CIDs are carried in RoHC packets, and only one PID value is needed, as shown in Table 7.

In Table 6, the “*n*” represents the number of PID values previously allocated to other header compression mechanisms, and “*x*” represents the number of CID values used by RoHC.

PID Value	HC mechanism	Packet type
0	No header compression	-
n+1	RoHC	CID=0
n+2	RoHC	CID=1
...	RoHC	...
n+x+1	RoHC	CID=x

Table 6 - Mappings of PID values for RoHC (CIDs in PDCP headers)

PID Value	HC mechanism	Packet type
0	No header compression	-
n+1	RoHC	RoHC packet format

Table 7 - Mappings of PID values for RoHC (CIDs in RoHC packets)

3.4. Conclusions

This chapter has presented the 3GPP cellular networks. GSM was used mainly for voice conversations, and it is based on circuit switching. The increasing demands on the transport of data lead to new requirements and to the usage of packet switching in the PLMN. GPRS emerged. At this point the first header compression mechanisms appeared in 3GPP. First, only cTCP and IPHC were available, but later RoHC was added.

The ever increasing bandwidth demands and the convergence towards All-IP networks, lead to the appearance of UMTS. Both IPHC and RoHC are currently supported by UMTS.

The support of RoHC by GPRS and UMTS was analyzed. In GPRS, the header compression negotiation is performed by using the SNDCP XID negotiation mechanism. In the case of RoHC over GPRS, the RoHC-over-PPP standard was adopted. In UMTS, no specific negotiation mechanism was recommended, so every mechanism can be used.

Details about RoHC transportation in these networks are relevant for the RoHC solution presented in this dissertation. There is not currently a standard solution for transporting and negotiating RoHC over IEEE 802 networks, and the 3GPP solution can be considered as a case study.

4. QoS Abstraction Layer

4.1. Introduction

The first discussions regarding the Fourth Generation (4G) of Mobile Networks emerged in the late nineties [39][40]. Although there are multiple visions for 4G Networks [41][42], they converge in many aspects such as seamless mobility and integration of wireless heterogeneous networks, IP-based, automatic configuration, and user-focused network services.

The IST DAIDALOS Project I (**D**esigning **A**dvanced **I**nterfaces for the **D**elivery and **A**ministration of **L**ocation independent **O**ptimised personal **S**ervices) aimed to develop an architecture for 4G wireless networks with Quality Of Service (QoS) support. The QoS Abstraction Layer (QoS-AL) [44] is a service for resource reservation in heterogeneous access technologies, and it is part of the DAIDALOS architecture. The RoHC solution presented in this dissertation aims at improving the QoS-AL with RoHC mechanisms.

This chapter describes the QoS-AL architecture. In Section 4.2 an overview of the DAIDALOS architecture is given and, in Section 4.3, the QoS-AL architecture is presented.

4.2. IST-DAIDALOS

The DAIDALOS Project I has developed a 4G network architecture supporting heterogeneous wireless technologies (WLAN/802.11, WMAN/802.16, (TD)-CDMA, and Bluetooth) with requirements which include auto-configuration transparency of the access network for the users, service transparency, mobility, and end-to-end QoS support. Figure 20 represents the DAIDALOS network architecture. This network is divided in Administrative Domains, each belonging to a Network Operator/Service Provider. Each Administrative Domain is further organized in one Core Subdomain and several Access Subdomains. The Mobile Terminals (MTs) (e.g. laptops, PDAs) are connected via wireless Access Points (APs) and L2 bridges to Access Routers (ARs). Wireless APs and bridges are not represented in Figure 20 for simplification.

The DAIDALOS QoS architecture is composed of QoS brokers (AQoSB) in the Access Subdomains, and a Core QoS Broker (CQoSB) in the Core Subdomain. Each AQoSB is responsible for managing the resources in the Access Subdomain, while the CQoSB has a global view of the resources available in the Administrative Domain. An E2E QoS reservation may interconnect two MTs in the same Administrative Domain, or interconnect MTs in two Administrative Domains. A QoS reservation is made in three sections: two L2 reservations between the end nodes and the respective ARs, and a L3 reservation between the ARs (see Figure 21).

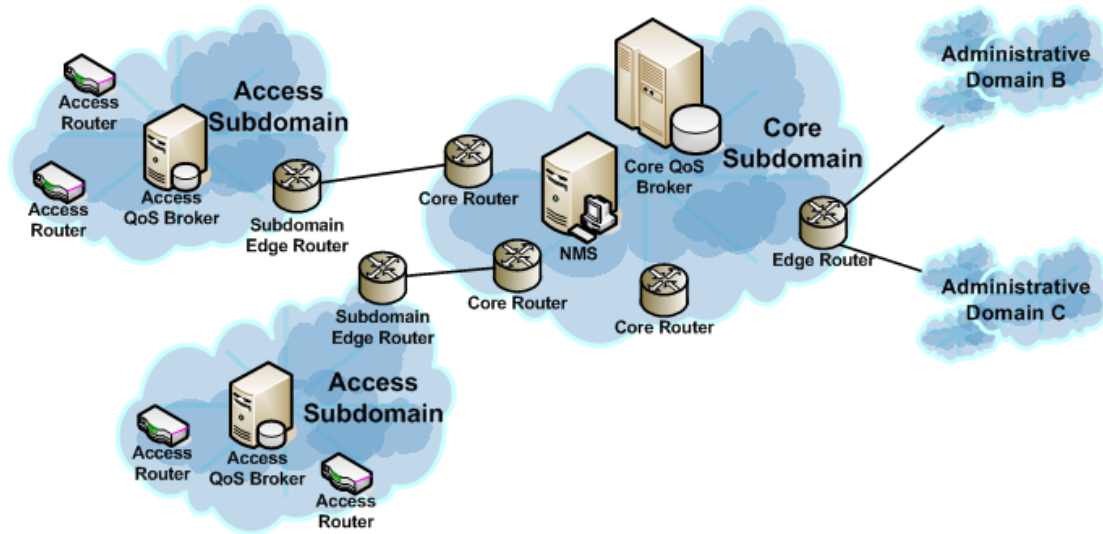


Figure 20 - DAIDALOS Network Architecture

The DAIDALOS E2E QoS architecture uses a distributed QoS Brokers scheme, which is also responsible for admission control. During the setup of a new end-to-end reservation, a QoS reservation request must be sent to the AQoS Broker. The AQoS Broker verifies if there are resources available in the Access Subdomain by contacting the relevant Access Router(s). Then it forwards the request towards the chain of QoS Brokers which may include the CQoS Broker and other AQoS Brokers along the path. If there are resources available, the AQoS Broker sends a reservation request to the same *QoS Manager* and also towards the chain of QoS Brokers, in order to establish the end-to-end path reservation. QoS Brokers communicate using the Common Open Policy Service (COPS) protocol [47].

The DAIDALOS QoS Architecture supports multiple QoS reservation strategies. The QoS requests may be initiated by the MTs, via the ARs, or by Application Servers/Service Proxies. In the former case the request may be: 1) an explicit QoS Request sent to the AR, and then forwarded to the AQoS Broker, or 2) legacy application signalling sent by the MT and intercepted by the AR, which is translated to a QoS Request and sent to the AQoS Broker. In the later case, the QoS reservation is made by the network.

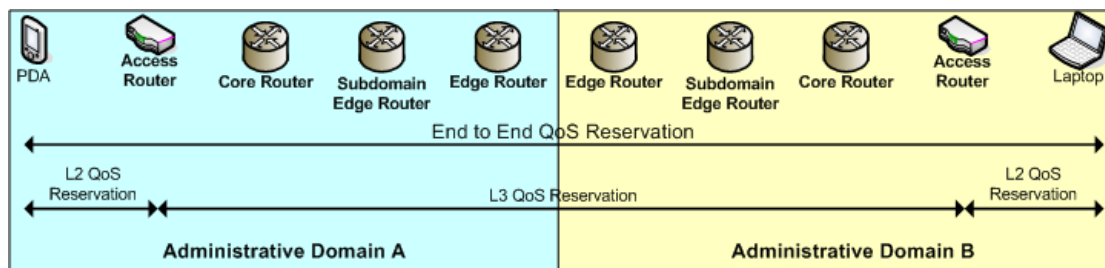


Figure 21 – DAIDALOS E2E QoS Concept

The Core Routers contain a *QoS Manager* module which is responsible for implementing a DiffServ QoS model. The *QoS Manager* module configures the following tasks: 1) policing, shapping, and scheduling, according to the policies defined by the QoS Broker, 2) Diffserv Code Point (DSCP) (re)marking, 3) load monitoring, 4) and L2 QoS resource reservation. The MTs have a *QoS Client* module responsible for:

- **controlling the QoS Signalling:** the QoS Client can act on behalf of the applications, by managing the signalling of particular flows;
- **supporting legacy applications:** the QoS Client contains a list of applications/ports and recommended QoS parameters, and automatically initiates resource reservation process when these applications start sending the traffic;
- **marking packets:** The QoS-AL expects that packets are marked with a connection identifier in the IPv6 Flow Label field [51], to identify the L2 flows. The QoS Manager marks downlink flows and the QoS Client configures the traffic control mechanisms so they can use the IPv6 flow label field conveniently;
- **reacting to QoS level changes:** The QoS Client reacts to QoS degradation notifications and, for instance, notifies the Applications.

4.3. QoS Abstraction Layer

The QoS-AL is responsible for reserving resources along the L2 path between the MTs and the ARs. It consists of There three types of modules, one for each type of network element: MT, AR, and access point.

4.3.1. QoS-AL Requirements

The QoS-AL requirements can be divided in four classes [44]: a) QoS requirements; b) mobility requirements; c) auto-configuration requirements; and d) other requirements.

4.3.1.1. QoS Requirements

A.1 – L2 QoS reservations: setup and tear down of the QoS reservations must be possible.

A.2 – QoS Abstraction: QoS-AL must be used in heterogeneous environments, and therefore, it should offer an abstract QoS interface and generic QoS parameters.

A.3 – Flexibility: QoS-AL must be flexible to allow its integration in multiple end-to-end QoS architectures, namely IntServ [48] and DiffServ [50].

4.3.1.2. Mobility Requirements

B.1 – Smooth handover support: the QoS-AL reservation should not be significantly degraded when the terminal moves between APs or ARs; therefore, smooth handover QoS aware mechanisms [46] must be supported.

B.2 – Triggers: the QoS-AL should detect network overloads and QoS degradation, and inform the network about it.

4.3.1.3. Auto-configuration Requirements

C.1 – Auto-routing: prior knowledge of the network topology should not be required by the MTs and the ARs.

C.2 – Concatenated networks: 4G scenarios will feature a multitude of wireless networks with different ranges. QoS-AL must perform QoS reservation across an arbitrary number of concatenated networks.

C.3 – Dynamic topology adaptation: the L2 topology is subject to changes throughout time. There are many possible reasons: links going down, a bridge/AP stop responding,

and the actual mobility of the terminal. The QoS-AL should continue to function after topology changes, without manual reconfiguration.

C.4 – Legacy equipment: the QoS-AL protocol must work transparently even when one or more L2 nodes do not support it. In those cases, QoS reservations should still be established on the segments supporting the QoS-AL protocol.

4.3.1.4. Other Requirements

D.1 – Event Generation: The wireless links conditions vary along the time. The QoS-AL shall provide events/indications to other layers, in order to enable adaptations such as transport and application.

D.2 – Robustness: the QoS-AL must address signal fading in wireless environments, which may result in periods of no connectivity.

D.3 – Multicast: the QoS-AL must be able to support QoS reservations for multicast flows.

D.4 – Modularity: the QoS-AL should be modular to incorporate upcoming network technologies.

4.3.2. QoS-AL Specification

4.3.2.1. QoS-AL Architecture

The QoS-AL consists of two components: 1) the *QoS-AL*, which implements the functionality common to all the technologies and, 2) the *AL Drivers*, which receive abstract primitives from the QoS-AL, and translate them into technology primitives.

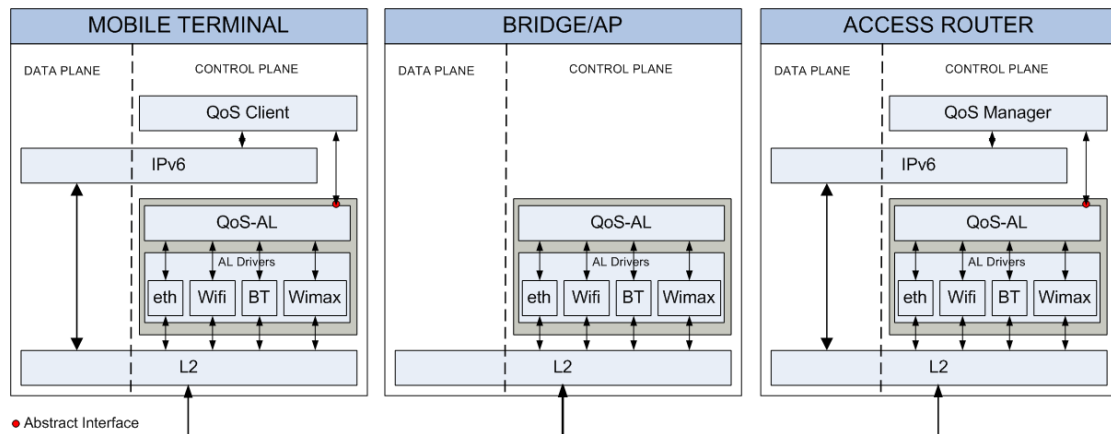


Figure 22 - QoS-AL Architecture

As described above and illustrated in Figure 22, the QoS-AL may be located in three types of network elements: MT, AP and AR. The *QoS-AL* modules in the end-nodes, MT and AR, provide their services to the *QoS Client* and *QoS Manager* through an *Abstract Interface* (AI). *QoS-AL* in the APs does not offer the AI. The *QoS Manager* delegates in the *QoS-AL* the responsibility of providing L2 QoS guarantees to flows traversing the heterogeneous wireless access network.

The *QoS-AL* is connection-oriented. Individual traffic flows are associated with L2 Connections, known as *QoS Connections*. The IPv6 Flow Label field is used to identify the connections and packets of a flow. All the connections have a connection identifier, known as CnxID.

Only the QoS Manager can initiate L2 QoS reservations. The DAIDALOS architecture includes L3 mechanisms which enable the QoS Client to contact the QoS Manager and trigger the activation of a L2-QoS connection. When the QoS Manager performs a QoS reservation, it uses a primitive of the QoS-AL in the AR. The L2 QoS reservation is signalled by the AL protocol. If the QoS reservation succeeds, the QoS-AL determines a CnxID and returns it to the QoS Manager. From this point after, the packets belonging to the flow for which resources have been reserved, are marked with the CnxID value at the MT or at the AR. Packets received with CnxID=0 receive best-effort service.

An AL driver needs to be developed for each L2 technology. Each AL Driver interfaces with the QoS-AL layer and with the driver of the L2 technology it represents. The AL Driver provides an unified interface to the QoS-AL and configures the driver of its technology.

L3-L2 QoS Mappings

The QoS mapping between the DAIDALOS L3 QoS framework and the QoS-AL can be characterized in the control plane and data plane. In the control plane the mapping is made by correlating QoS parameters. The QoS-AL parameters are roughly the same used in the L3 QoS, namely the *Class Identifier*, *Reserved Bitrate*, and *TSpec*. The *Class Identifier* indicates the type of traffic that will be transported in the QoS Connection. Table 8 lists the QoS Classes available in DAIDALOS. Each of them is characterized by end-to-end delay, delay in the L2 access network, and the packet loss. The *Reserved Bitrate* indicates the amount of bandwidth reserved in the L2 access network. The *TSpec* parameter has the same meaning as in RSVP/IntServ [52][48], and it specifies the flow requirements in terms of required bandwidth and delay.

Parameters	Class 0 Conversational	Class 1 Transactional	Class 2 Streaming	Class 3 Best Effort
Mean Delay (end-to-end)	150 ms	400 ms	1 s	Unspecified
Mean Delay (L2 access network)	40 ms	100 ms	250 ms	Unspecified
Loss probability	1×10^{-3}	1×10^{-3}	1×10^{-3}	Unspecified
Example Services	Interactive voice and video (e.g. audio and video conferencing)	Transaction data, interactive (e.g. Web browsing, telnet, e-commerce)	Short transactions, bulk data, video streaming (e.g. video on demand, ftp)	Legacy applications / budget services

Table 8 - DAIDALOS QoS Classes and Parameters (from [44])

The QoS-AL exists only in the control plane of the communications stack. In the data plane the DSCP field is used to identify signalling packets. The L2 access network uses the CnxID/Flow Label field for flow identification. When a new connection is about to be set up, i.e. a packet arrives to the QoS Manager without belonging to an existing L2 QoS Connection, the QoS Manager determines the QoS requirements of this flow by inspecting the packet DSCP field. Table 9 lists the rules for classifying packets in the data plane. In DAIDALOS even the best effort traffic needs a QoS reservation in the L2 access network.

Flow Label	DSCP	Packet Type / Treatment
$\neq 0$	Don't care	Application flow with reservation. Apply the QoS of the connection identified by <i>FlowLabel</i> = <i>CnxID</i>
= 0	$\neq \text{signalling}$	Flow without reservation – some residual bandwidth reserved for it.
= 0	= <i>signalling</i>	Signalling packet. Some residual bandwidth reserved for it.

Table 9 - Data plane QoS mapping rules (from [44])

4.3.2.2. Service Interface

The QoS Abstraction Layer provides a service interface, called *Abstract Interface* (AI), which provides services which can be grouped in five categories: 1) QoS reservation; 2) resource querying; 3) QoS notification; 4) mobility; 5) multicast.

The QoS reservation service consists in the creation of QoS Connections between the AR and the MT. Within this group, there are primitives to create, modify, and deactivate QoS connections. The primitives AL-CNX-ACTIVATE-REQ (Annex A2.1) and AL-CNX-ACTIVATE-RESP (Annex A2.2) are used in the establishment/activation of a new QoS connection. The former initiates the QoS reservation signalling in one direction and, if successful, the later is returned to the calling entity (i.e. QoS Manager) with the CnxID. The calling entity becomes from that moment on responsible for marking the *Flow Label* of packets belonging to the QoS Connection with that CnxID. The same identifier can be used to modify or deactivate the QoS connection, and it is included in QoS degradation notification messages. When a connection is activated, the destination end point is notified by receiving the primitive AL-CNX-INDICATION (Annex A4.1). This primitive is also for notification of QoS degradation. The primitives AL-CNX-MODIFY-REQ (Annex A2.3) and AL-CNX-MODIFY-RESP (Annex A2.4) are used to modify connection reservations. The QoS-AL implements a *makeBeforeBreak* mechanism; if for some reason a QoS modification fails, the QoS settings are maintained as they were before the modification. The attempts to decrease the QoS levels are always successful. The AL-CNX-DEACTIVATE (Annex A2.5) is used to deactivate QoS Connection.

The AL-RESOURCE-QUERY (Annex A3.1) is used to request a report of the resources available along a L2 path. This primitive is used by the QoS Manager for L2 admission control, before making the QoS reservations. In response to the AL-RESOURCE-QUERY, an AL-RESOURCE-INDICATION (Annex A3.2) is received, containing the bandwidth available, which is the minimum bandwidth available in the L2 links which make the L2 paths. The AL-RESOURCE-INDICATION can also be sent spontaneously by the QoS-AL, either periodically or when significant changes in the resources available occur. The AL-RESOURCE-DEGRADATION (Annex A3.3) is sent when the bitrate available in any of the APs drops below 25% of its maximum bitrate. The AL-CNX-INDICATION (Annex A4.1) is issued by the AR and the MT to indicate that the QoS-AL was forced to modify the QoS due to changes in the resources available. This information may be used by higher layers to support handover or adaptation decisions. It is also used to notify the QoS Client about new QoS Connections.

The QoS-AL has support for mobility scenarios. When a handover is about to take place, the new AR cannot activate a regular QoS Connection because the MT has not yet arrived. In these cases, the new AR sends an AL-CNX-ACTIVATE-REQ with the *handoff* parameter set to True. This parameter postpones the activation of the new QoS Connection until the MT connects to the new AR/APs and calls the AL-

HANDOVER-EXECUTE (Annex A5.1) primitive. This sets off an equivalent PDU, which is sent to the AR, requesting it to proceed with a regular activation.

A multicast QoS Connection is set up in two stages. First, a multicast session is created. Then membership management takes place. The primitive AL-MULTICAST-JOIN (Annex A6.1) is used by the AR to add a given MT to the multicast session. The primitive AL-MULTICAST-LEAVE (Annex A6.2) is used to indicate that the MT is leaving the multicast session.

4.3.2.3. Driver Interface

The interface between the QoS-AL and the AL drivers is named *Driver Interface* (DI), and it exists in every L2 node with QoS-AL support. The DI primitives are invoked by the QoS-AL process. The AL driver processes the call and returns the results to the QoS-AL.

The AL Driver must register in the QoS-AL. For that purpose, it sends the AL-DRIVER-REGISTER primitive (Annex B1.1), declaring the intention to control a driver for a certain network interface. Each AL driver instance may handle several network interfaces.

The DI primitives that handle QoS reservations and modifications, i.e. AL-CNX-ACTIVATE-REQ (Annex B2.1), AL-CNX-ACTIVATE-RESP (Annex B2.2), AL-CNX-DEACTIVATE (Annex B2.3), AL-CNX-MODIFY-REQ (Annex B2.4), AL-CNX-MODIFY-RESP (Annex B2.5), and AL-CNX-DEACTIVATE (Annex B2.5), are similar to the service interface primitives. There are minor differences, such as the driver interface supports only L2 addresses, and there is no context information parameters.

4.3.2.4. AL Protocol

The AL is a L2.5 protocol, and exists at the control plane. The QoS-AL PDUs are transmitted as IEEE 802 frames with a dedicated *ethertype* number (0x1234). For most of the AI primitives there is a corresponding PDU.

The protocol specification was inspired on the RSVP protocol [52]. It uses in-band signalling and soft-state [54], and the PDUs use the same channels and paths of the data plane packets. QoS reservations have a TTL and therefore, they have to be periodically refreshed, or else they expire and the associated resources are freed.

Figure 23 illustrates an example of a QoS Connection activation, using the AL protocol. Before initiating the activation, the AR must generate a CnxID for the new connection. Then it sends the AL-CNX-ACTIVATE-REQ to the MT's MAC address, carrying as parameters, the CnxID and the QoS. There is no need to have previous knowledge of the path since the path discovery is guaranteed by the IEEE 802.1D Learning Bridges [55]. Intermediary bridges along the path capture packets with AL's *ethertype*, process and re-queue them.

QoS reservations are only committed when the QoS-AL enabled nodes receive the corresponding AL-CNX-ACTIVATE-RESP. One of the reasons a response frame is needed, has to do with the nature of the Learning Bridges. When a Learning Bridge ignores the location of a certain MAC address, it broadcasts the frame to every port. When a frame with that MAC address as source address arrives at the bridge, it learns the port associated with that MAC address. The problem may arise when AL-CNX-ACTIVATE-REQ frames are broadcasted (see step 4 in Figure 23). If a response

message was not used, this would set up two QoS reservations, one in the 802.11 AP, and the other in the 802.15 NAP.

A AL-CNXX-ACTIVATE-REQ/RESP pair is used to refresh a QoS reservation. As the AL-CNXX-ACTIVATE-RESP frame passes through the nodes along the path, they refresh the timer associated with that CnxID.

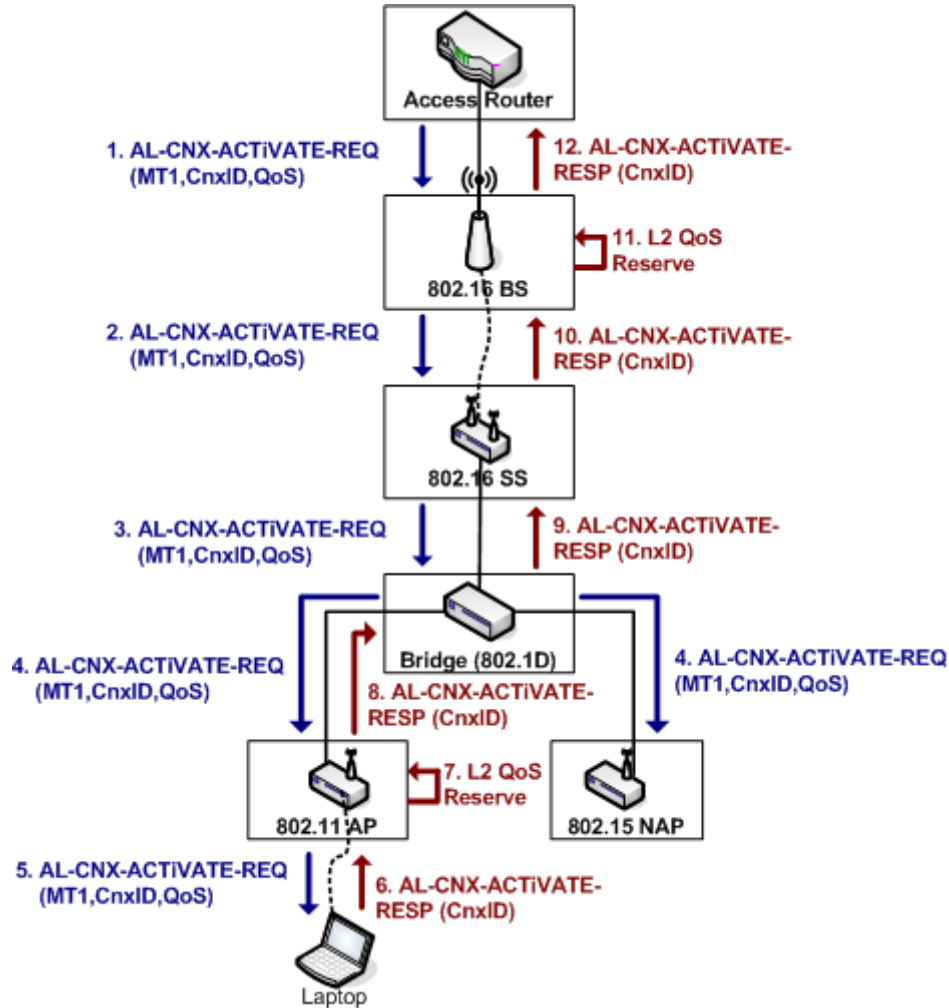


Figure 23 - QoS Connection Activation Example (from [44])

Figure 24 represents a resource query directed to an AP. Resource queries are used to know what are the resources available along a L2 path. In the example of Figure 24, the AR sends an AL-RESOURCE-QUERY directed to AP with MAC address AP2b. The frame is forwarded towards the destination AP. There, the QoS-AL asks the AL Driver about the resources available which returns the value 800 kbit/s. An AL-RESOURCE-INDICATION with that value is returned. Intermediate bridges in the return path verify if they have sufficient resources to satisfy the bandwidth reported. If not, they replace that value with the amount of unreserved bandwidth they still have left. When the AL-RESOURCE-INDICATION reaches the Access Router, it will contain the maximum value that all nodes are able to provide for a new QoS Reservation.

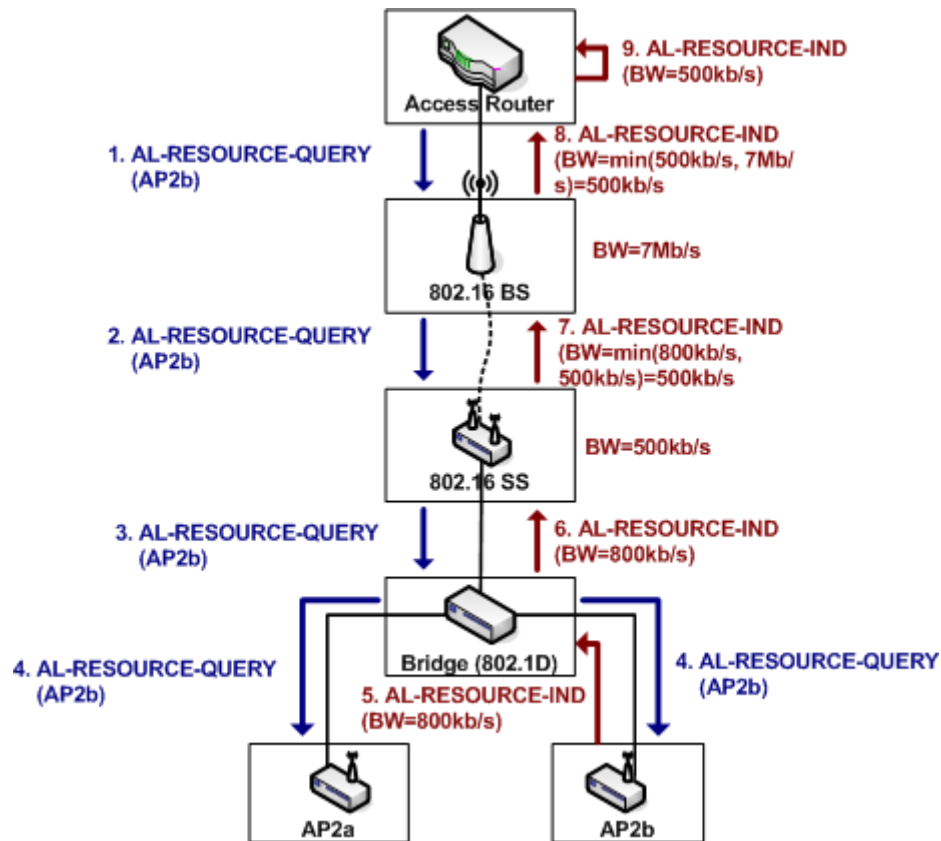


Figure 24 - Example of a Resource Query with an AP as target (from [44])

The AR needs to discover automatically what APs are reachable from its interfaces, and the bandwidth available towards that AP. QoS-AL enabled bridges announce their presence with an AL-ANNOUNCE PDU. This message carries as parameter a list of intermediary nodes. They are periodically broadcasted by bridges and received by the ARs. Thus, by receiving these messages the AR can build the topology of the Access Network and store it in a local database. These database entries are also soft-state, so they are deleted if not refreshed in due time. Thus, the AR is able to find out when bridges are disconnected. Another advantage of using these messages is that by sending them, they also update the MAC cache of intermediary bridges between the sending node and the ARs.

Table 10 summarizes the PDUs that are used in the AL protocol.

PDUs	Parameters
AL-CNXX-ACTIVATE-REQ	dst, src, cnxid, tx_tspec, tx_rspec, rx_tspec, rx_rspec, retention_prio, ctx_info
AL-CNXX-ACTIVATE-RESP	dst, src, cnx_id, result
AL-CNXX-DEACTIVATE	dst, src, cnx_id
AL-CNXX-INDICATION	dst, src, cnxid, tx_tspec, tx_rspec, rx_tspec, rx_rspec, ber, ctx_info
AL-RESOURCE-QUERY	dst, src
AL-RESOURCE-INDICATION	dst, src, bandwidth
AL-ANNOUNCE	dst, src, aplist

Table 10 - AL PDUs

4.4. Conclusion

This chapter has described the QoS-AL architecture. The QoS-AL is a framework developed within the IST DAIDALOS project, for QoS reservation in L2 heterogeneous access networks. It provides an abstract QoS interface. The QoS-AL can be integrated with various L3 QoS models, namely RSVP/Intserv or Diffserv.

The QoS reservations, also known as QoS Connections, are set up between ARs and MTs. They are initiated by the AR which sends an AL-CNX-ACTIVATE-REQ to the MAC address of the MT. The QoS reservation is committed by the AL-CNX-ACTIVATE-RESP travelling in the return path. Because the QoS-AL nodes are IEEE 802.1D Learning Bridge compatible nodes, there is no need of previous knowledge about the access network topology. The QoS-AL nodes periodically announce themselves by sending AL-ANNOUNCE PDUs to the ARs. These factors account for the auto-configuration requirement of the QoS-AL framework.

The QoS-AL is a modular architecture. The QoS-AL contains a SAP, the *Abstract Interface*, which consists of primitives with abstract QoS parameters. Below the QoS-AL process, there are *AL Driver* modules which receive abstract QoS commands from the QoS-AL and translate them into L2 specific QoS parameters. Each AL Driver controls a L2 hardware driver. Examples of different AL Drivers can be found in [45].

The QoS-AL has support for mobility, and it provides triggers for handover execution. The AI primitives supporting the handover are the AL-RESOURCE-INDICATION, and AL-RESOURCE-DEGRADATION. Both report changes to the free QoS resources, and provide information for higher layer handover decisions. New QoS reservations are prepared, and committed when the MT signals its arrival to the new AP.

5. AL-RoHC Specification

5.1. Introduction

This chapter presents the AL-RoHC, which is a solution for integrating RoHC in the QoS-AL framework described in the previous chapter. There are 2 goals associated to QoS-AL: to achieve good transmission efficiencies, and to enable the RoHC transportation over heterogeneous and concatenated L2 networks.

This chapter starts by identifying the AL-RoHC functional requirements in Section 5.2, followed by the actual specification. The AL-RoHC specification is presented in two parts: first, a generic solution for RoHC transportation over IEEE 802 networks is proposed in Sections 5.3 and 5.4; then, in Section 5.5, this scheme is adjusted to the QoS-AL. Section 5.7 provides the conclusions of the chapter.

5.2. Requirements

5.2.1. RoHC Requirements on Lower Layers

The RoHC requirements can be divided in two groups: 1) requirements that were established as design goals in the specification of RoHC [8][10]; 2) requirements that RoHC imposes on the lower layers. The later are defined within the IETF in *RoHC-over-X* documents, where *X* is a lower layer (e.g. RoHC-over-PPP [13]). Every *RoHC-over-X* specification must respect the requirements defined in [9] and described below:

R1.1 - Error detection: the lower layers must provide error detection capabilities. The residual bit error probability of compressed headers should be close to zero.

R1.2 - Inferred header fields: some RTP/UDP/IP fields are classified as inferred by RoHC. Their values are usually obtained from the underlying link layer. RoHC requires that at least the following header fields can be inferred from the lower layer: 1) Packet Length (IPv4) and Payload Length (IPv6); 2) UDP Length.

R1.3 - Variable header sizes: it is desired that packet size variations are minimized. However, since header compression efficiency varies, RoHC requires that the underlying layers support headers from 1 to 60 bytes and packet sizes of different sizes. Nevertheless RoHC can still function when limitation of the number of packet size exists, since it includes padding schemes.

R1.4 - Negotiation: before RoHC may be used it needs to configure some parameters. Those parameters include header compression profiles allowed, and type of CID to use.

For unidirectional links this configuration may be performed out-of-band or a priori. AL-RoHC must provide a mechanism to negotiate RoHC parameters.

R1.5 - Flow multiplexing: RoHC flows can be multiplexed into a link layer channel or, in case the link layer supports it, multiplexed on logical channels. In the later case, the need for context identification is not so clear, as each logical channel is enough to isolate a context.

R1.6 - Packet type identification: RoHC includes a packet type identifier to distinguish between different RoHC packet types. However, link layer identification for RoHC packets is needed; for instance to distinguish RoHC packets from regular IP packets.

R1.7 - Duplication and Reordering: RoHC is able to cope with packet duplication and reordered packets before the compression point, but not between the RoHC Compressor and RoHC Decompressor. The RoHC Decompressor expects to receive packets in the same order they were compressed.

R1.8 - Feedback: in order to support the O-mode and the R-mode, the lower layers must be able to transport feedback packets from Decompressor to Compressor. For optimal compression efficiency, the feedback packets should be delivered to the compressor as fast as possible.

R1.9 - Handovers: the packet loss due to handover should be minimized. When handovers are performed, the RoHC Compressor should be notified to take appropriate measures and in order to prevent the consequences associated with long loss of events. RoHC includes a mechanism for context relocation [20]. If this mechanism is not available or can not be used, the context may be built by means of context refresh.

5.2.2. AL-RoHC Requirements

R2.1 - Heterogeneity: the solution must function transparently over IEEE 802 networks such as 802.3, 802.11, 802.15, and 802.16.

R2.2 - Flow granularity: RoHC should be used in QoS-AL reservations. QoS-AL reservations are usually set for individual application flows, so two (unidirectional) RoHC channels for each (bidirectional) QoS reservations should be employed.

R2.3 - RoHC transportation over IEEE 802 networks: AL-RoHC needs to define means to transport RoHC packets over IEEE 802.x networks. RoHC requirements on lower layers must be considered. For minimum overhead, this transportation should use direct L2 encapsulation.

R2.4 - RoHC negotiation: a RoHC negotiation mechanism over IEEE 802.x is needed. The AL protocol should be extended for this purpose.

5.2.3. RoHC and QoS-AL integration Requirements

R3.1 - IPv6 only: QoS-AL supports only IPv6. Therefore, AL-RoHC must support the RoHC profiles based on IPv6.

R3.2 - Optional RoHC utilization: each QoS-AL reservation may optionally choose to use RoHC in the context of that reservation.

R3.3 - Access to CnxIDs: QoS-AL flows transport the CnxIDs in the IPv6 flow label. RoHC may compress this field. So the AL-RoHC must provide a way for QoS-AL nodes to access this value in data packets.

R3.4 - Reservations Initiator: in QoS-AL access networks, the reservations are triggered by the QoS Manager in the AR. Thus, the setup of two RoHC channels for that reservation must also be initiated by the AR.

5.3. RoHC over IEEE 802 Networks

RoHC requires access to some information which is commonly found in lower layer headers. The IEEE 802 technologies offer a MAC service interface, hiding the physical details of the technology. Most of the IEEE 802 equipments support Ethernet II/802.3 frame encapsulation. This suggests that developing a single RoHC transportation solution over IEEE 802 networks would be a good choice, as opposed to developing solutions for every technology.

This dissertation proposes an unified solution for transporting RoHC packets over IEEE 802 networks, a combination for which there is no standard defined. There was a previous effort, in the form of an IETF internet draft, to specify RoHC over 802 networks [21]. Some directions were pointed out, but a final solution was not proposed.

A solution based on the existing RoHC-over-PPP specification [13] could be reused since PPP can also be transported over IEEE 802 networks. However, as mentioned in [21], the overhead and the additional complexity discourage its use. The solution should add as little overhead as possible. Hence, solutions based on the direct transportation of RoHC packets over Ethernet encapsulation seem to be preferable. There are five types of Ethernet frames:

- Ethernet I frames: the original type of Ethernet frames, specified by Ethernet version 1 [35]. This type of framing is no longer supported by most equipment. It included a source and destination MAC addresses (SA and DA), and 2 bytes *length* field.
- Ethernet II frames: commonly known as DIX frames, these frames are named after the major participants in the specification of Ethernet: **D**igital Equipment Corporation, **I**ntel and **X**erox. In this encapsulation, the 2 byte field following the DA field is interpreted as the *ethertype* field, which is used to identify the upper-layer protocol. In order to avoid confusing this field with the *length* field, only *ethertype* numbers over 0x600 are supported (1536 in decimal notation). Frames containing *length* fields have a maximum value of 1500 bytes [36].
- IEEE 802.3/802.2 LLC frames [37]: supported by the 802.3 standard. The 802.2 frames may contain an extra 3 bytes LLC field which can be used to multiplex LLC protocols and provide some basic flow control service. These frames are *length*-based, and do not provide directly upper-layer protocol identification.

- IEEE 802.3/802.2 LLC/SNAP frames [37]: the 802.2 frames may optionally also include a 5 byte SNAP header, containing a 3 byte *vendor code* field, and a 2 byte *local vendor* field. The later is typically used to transport the *ethertype* field, missing in the base header.
- RAW 802.3 frames: a rare variation of the 802.3 used by Novell, no longer supported.

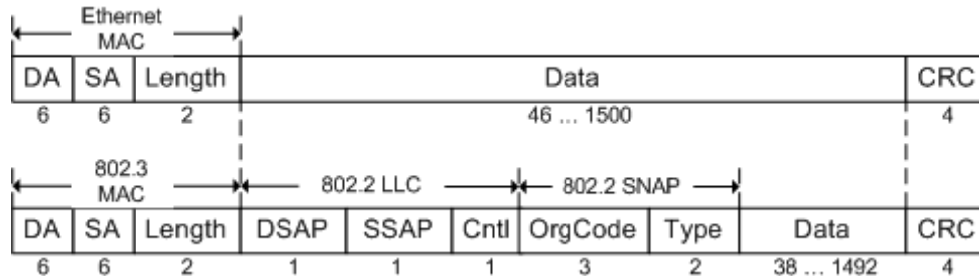


Figure 25 – Ethernet DIX and IEEE 802.3/802.2 frames

These types of frames can coexist in the same physical medium, but typically, only Ethernet II frames and 802.3/802.2 variants are supported. These two encapsulations can be distinguished by the *length/ethertype* value. The Ethernet II/DIX frame has less overhead, and is the most common type of frame. Hence, it is more appropriate for the solution proposed by this dissertation.

There are two main challenges that have to be addressed in order to support the efficient transport of RoHC packets in DIX frames: 1) the Ethernet minimum frame size problem, which will be described next; and 2) the mechanism required for negotiating RoHC channels in IEEE 802 networks.

5.4. The Ethernet Minimum Frame Size

Ethernet uses CMA/CD (Carrier Sense Multiple Access with Collision Detection) for medium access control. In order to enable the frame collisions to be detected, the frames must have a minimum size of 64 bytes. 14 bytes are used for the MAC header and 4 bytes are used for the MAC trailer, leaving 46 bytes as a minimum value for the payload. When an host sends an IP packet smaller than 46 bytes, the Ethernet interface must add padding until the 46 bytes are achieved. Other IEEE 802 technologies may use other access methods which do not impose minimum frame sizes.

Bridges which interconnect Ethernet links with non-Ethernet links must also add and remove padding to frames in transit, as illustrated in Figure 26. In order to remove padding, the bridges must first verify if padding is present. In the 802.3/802.2 frames, this is done simply by checking the *Length* field in the L2 header. However, when using DIX frames the bridges have to snoop the *Length* information from upper layer headers, such as IPv4's *Total Length* or IPv6's *Payload Length*.

The use of RoHC introduces one problem. RoHC usually compresses these fields, and the bridges are unable to determine if padding is present in the DIX frames carrying RoHC packets and, therefore, bridges are unable to remove padding before forwarding the frames towards the non-Ethernet links, where padding is not needed. This extra overhead may negate some of the RoHC's gain.

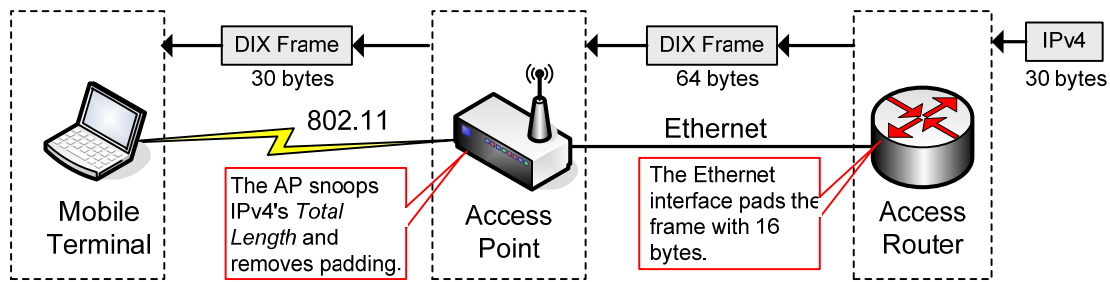


Figure 26 - The use of padding in Ethernet links

Figure 27 illustrates the problem. The AR sends a 70 byte VoIP packet to the MT, of which 40 bytes are for the IPv4 header, 8 bytes are for the UDP header, 12 bytes are for the RTP header, and the remaining 10 bytes are the voice information. Before sent, the VoIP packet is header compressed and reduced to 14 bytes. Before sent to the Ethernet link, 32 bytes of padding are added in order to achieve the 46 byte minimum size. This results in 35% of compression ratio ($46 / 70$). The RoHC packet is then encapsulated in a DIX frame and sent towards the MT. When the frame reaches the AP, it can not see if the frame contains padding because the *IP Total Length* was suppressed. The AP changes the L2 encapsulation and sends the RoHC packet and Ethernet padding to the MT. As a consequence, the compression ratio in the IEEE 802.11 link remained at 35% where it should reach 80% ($14 / 70$) if padding was not propagated. The example contains two 802 links.

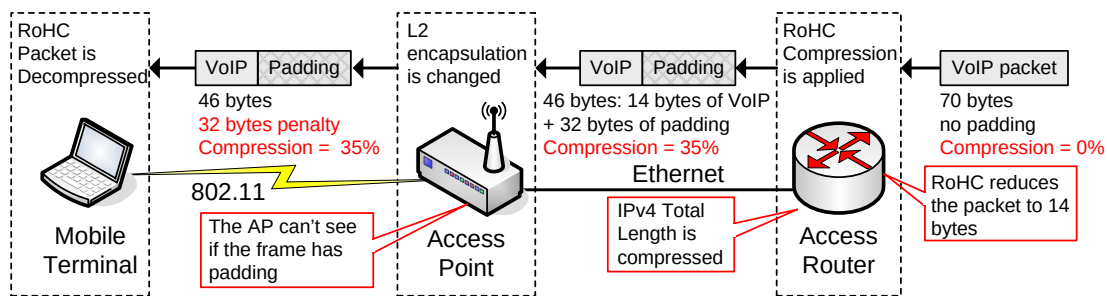


Figure 27 – Padding propagation to non-Ethernet links example

If 802.3 links (802.3 frames) were mixed with Ethernet links (DIX frames), another problem could occur. There are some 802.3 bridges which change the encapsulation of DIX frames to 802.3/802.2 frames. This would result in more overhead. The solution proposed by this dissertation assumes that these types of bridges are rare and this case is not addressed.

In [21] Bormann outlines three possible approaches, all of them based on 802.3/802.2 Ethernet frames. Using this type of frames the padding problem is solved since they carry a *length* field. The proposals focus on ways to provide indication that the frames carry RoHC packets. The first proposal consists in allocating two extra SAP values for representing Low and High-CID RoHC packets. If, by some reason, it is not possible to allocate new SAP values, a second proposal recommends the use of restricted SAP values for the same end. The third option makes use of the optional SNAP header.

These proposals are not ideal because 1) they only work on equipment that supports 802.3/802.2 headers, and 2) they are not so resource efficient as they could be, because they add 3 or 8 bytes, when 2 bytes are sufficient to hold *length* or *ethertype*

information, depending of the type of Ethernet frame in use. The SNAP-based proposal is as inefficient as the PPP-based solution, since both add 8 bytes of overhead.

Figure 28 presents our solution. The solution is based on DIX frames, to which a *length* field is added between the L2 and the L3 headers. Bridges/APs must inspect this field to infer the L2 payload size and, thus, detect the presence of padding. Compared with Bormman's proposals, this solution only adds 2 bytes to the L2 payload.

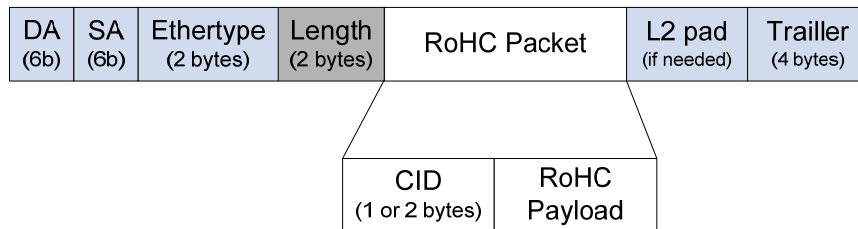


Figure 28 - RoHC encapsulation in DIX frames

Two new *ethertype* values are needed to indicate the type of CID (Low or High-CID) used in the RoHC packets. Note that in non-Ethernet links the encapsulation of RoHC packets does not need an extra *Length* field, since in those links padding is never added.

This solution requires changes in the network equipment. This is not a problem since the next generation equipments required for 4G networks is expected to suffer hardware/software adjustments.

5.5. RoHC over QoS-AL Access Networks

Integrating RoHC in the QoS-AL introduces the problem of RoHC compressing the IPv6 headers, making the IPv6 Flow Label inaccessible to the QoS-AL modules. Bridges and APs along the path between the AR and the MTs need to access the CnxID of each packet for scheduling purposes. In order to overcome this problem, and since a *length* field was already introduced, a CnxID field will be placed between the L2 header and the RoHC packet. This requires a change to the QoS-AL, so that its modules can look for the CnxID after the L2 Header, instead of the *IPv6 Flow Label*. As a side effect, the QoS-AL becomes no longer dependant on the availability of the IPv6 headers and, therefore, it becomes able to transport other payloads types such as IPv4. A 4-bit reserved bits field was added to maintain the frame aligned to byte lengths.

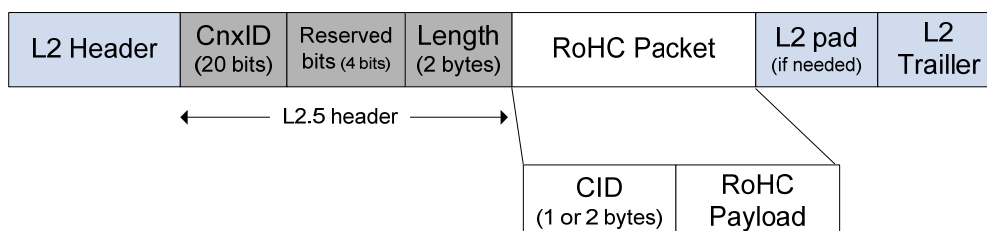


Figure 29 - RoHC encapsulation in QoS-AL Access Networks

5.5.1. RoHC Channel Negotiation

The RoHC channel negotiation is conducted between the AR and the MT. In its simplest form the negotiation is unidirectional. Two independent negotiations are

needed for bidirectional RoHC conversations between two RoHC peers. A bidirectional negotiation scheme is provided as a signalling optimization, but there is always an independent negotiated set of RoHC parameters per traffic direction.

The negotiation is performed by two new AL protocol PDUs: 1) AL-RoHC-REQUEST-PARAMETERS; and 2) AL-RoHC-PARAMETERS. The former is used by the RoHC entity which initiated the request of parameters; and the later is used to answer the request. The description of these messages can be found in Annex C.1.

Two negotiation schemes are proposed: 1) the standalone negotiation scheme, which can be used in any network scenario; and 2) the AL-RoHC negotiation scheme, optimized for QoS-AL access networks. Only the later is supported by the QoS-AL. In the standalone negotiation there are two possible signalling modes, the Simple Mode, and the Quick Mode. The Simple Mode is unidirectional. In Figure 30 Node1 sends an AL-RoHC-REQUEST-PARAMETERS PDU to Node2 and receives an AL-RoHC-PARAMETERS as response. At the same time, Node2 may also initiate Simple Mode negotiation with Node1. The Quick Mode is an optimized signalling mode in which the initiating mode, Node1 in Figure 31, sends its RoHC parameters piggybacked in the RoHC parameters request. Thus, a single pair of signalling messages is used instead of two. In the AL-RoHC negotiation scheme, the Quick Mode is mandatory. The AL-RoHC negotiation messages may be piggybacked on AL-CNXX-ACTIVATE and AL-CNXX-ACTIVATE-RESP messages, as shown in Figure 32, or sent separately.

A new parameter, called RoHC Flag (RF) was also introduced in AL-CNXX-ACTIVATE and AL-CNXX-ACTIVATE-RESP primitives. This flag can be used by any of the network elements in the path of a QoS reservation to indicate that RoHC must not be used in the reservation which is about to be created. All nodes in the path must agree on using RoHC, for RoHC to be enabled in a given reservation.

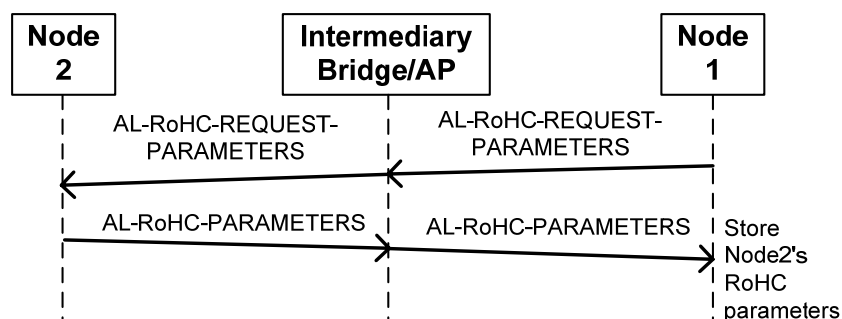


Figure 30 – Standalone negotiation - Simple Mode signalling (Unidirectional)

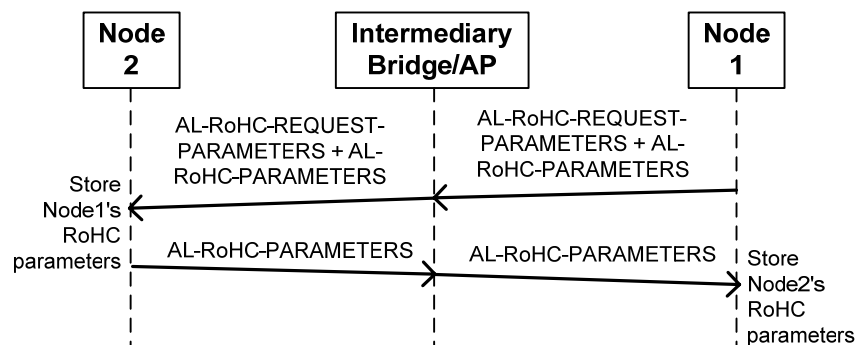


Figure 31 - Standalone negotiation - Quick Mode signalling (Bidirectional)

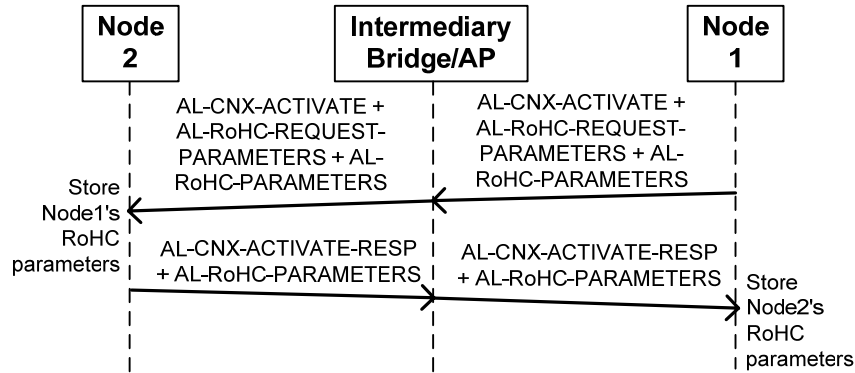


Figure 32 - AL-RoHC negotiation - Quick mode piggybacked (Bidirectional)

Once the negotiation is complete, multiple individual RoHC flows are established without the need to renegotiate the RoHC channel parameters. If there are any changes in a host RoHC parameters, the channel parameters may be renegotiated considering that:

- in QoS-AL scenarios only the AL-RoHC negotiation scheme is supported;
- when using the AL-RoHC negotiation scheme, the piggybacked Quick Mode can only be used when AL-CNXX-ACTIVATE messages are sent. These messages are sent initially to set up the QoS reservation and when the QoS-AL needs to be refreshed.

The RoHC parameters are stored with a TTL to guarantee the freshness of this information and save resources. When the RoHC channel expires, a new one must be set. Thus, to prevent traffic disruption the nodes must periodically refresh the RoHC parameters. Because RoHC parameters do not change often, the TTL could be set to an high value.

5.5.2. MTU considerations

The Ethernet DIX frames support a maximum transmission unit (MTU) of 1500 bytes.. The solutions proposed for carrying RoHC packets are DIX-based, and add L2.5 headers of 2 and 5 bytes, respectively to RoHC-over-802 and RoHC-over-QoS-AL access networks. This leaves 1498 bytes and 1495 bytes available for payload.

Other types of IEEE 802 links support larger MTUs. The IEEE 802.11b/a/g supports a maximum of 2304 bytes, which means that 802.11 APs may need to perform fragmentation when forwarding frames to 802.3 links. In practice, this is avoided by setting the default MTU in the AP to 1500 bytes.

5.5.3. Architecture

Figure 33 presents the general architecture of the AL-RoHC. This architecture is based on the QoS-AL architecture presented in Section 4.3.2.1.

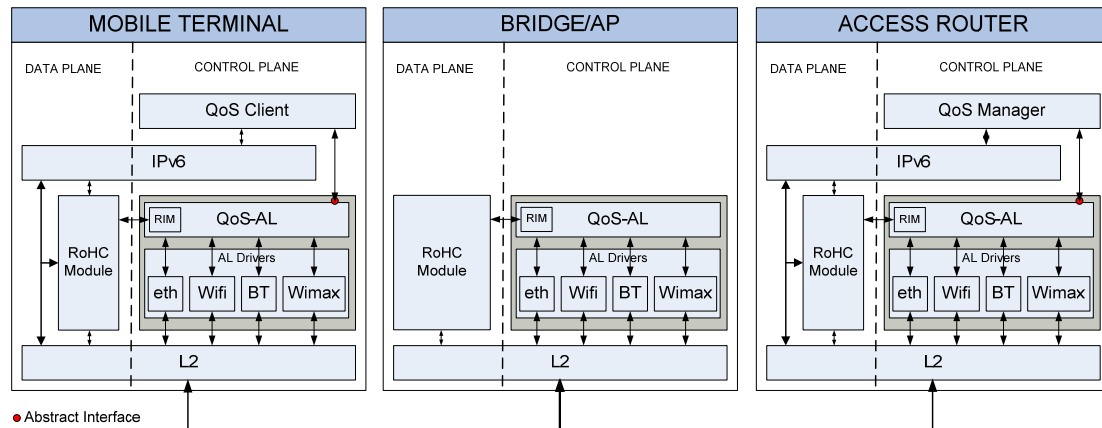


Figure 33 - AL-RoHC Architecture

Figure 33 represents a one hop IP link consisting of one MT, one bridge/AP and one AR. Multiple bridges/APs between the AR and the MTs can also be supported. The AL-RoHC protocol stack is split in control and data planes. In the data plane a RoHC Module (RM) was placed between the TCP/IP stack and the L2 drivers. This module is responsible to apply RoHC to data packets. In the control plane the QoS-AL module was extended with an embedded module called RoHC Interface Module (RIM). The RIM is responsible for RoHC Channel negotiations and for configuring and interfacing with the RM. There are different instantiations of these modules for endpoints (AR, MT) and intermediary nodes (bridges, APs). The differences will be described in Section 5.5.4.

When a reservation is requested to the QoS Manager, the QoS-AL module inquires the RIM if a RoHC channel to the destination host (e.g. the MT) is already set. If not, the QoS-AL instructs the RIM to set the channel. The RIM forwards the query to the RM which returns a true/false value. In case the answer is negative, the RIM sends an AL-RoHC-REQUEST-PARAMETERS to the MT. By default, the Quick Mode negotiation should be used. Hence, the AL-CNX-ACTIVATE primitive is sent with the AL-RoHC-REQUEST-PARAMETERS piggybacked, in addition to the AR's own AL-RoHC-PARAMETERS primitive. The RF field in AL-CNX-ACTIVATE is always sent with value "1", which means RoHC utilization on the new QoS reservation is desired. If some intermediary bridge does not support RoHC, along the path between the AR and the MT, it would change the RF field to "0", meaning that RoHC must not be used for that QoS reservation. When the message reaches the MT, the AR RoHC parameters are stored, the QoS reservation is set up, and the MT returns an AL-CNX-ACTIVATE-RESP primitive, maintaining the RF field set to "1". Note that the RoHC usage can be disabled in one direction and enabled in the other direction. This message is piggybacked with the MT's AL-RoHC-PARAMETERS. When both peers have exchanged RoHC parameters, the RoHC channel is considered set.

At both ends the RoHC parameters are received by the RIM which forwards them to the RM, that stores them. When the RF field in the reservation messages reaches an endpoint set to "1", the local RIM informs the RM that IP packets associated with the CnxID are to be RoHC compressed. From that moment after, outgoing IPv6 packets containing that CnxID are intercepted by the RM which then performs the RoHC function, builds the L2.5 header, and sends the frame to the network interface driver.

Bridges along the path recognize frames carrying RoHC packets by the two new *ethertypes*. They exchange the L2 encapsulation and forward them, leaving the L2.5

header and RoHC packet unchanged. As packets reach the destination host they are again intercepted by the RoHC module, which decompresses the RoHC packets and delivers the IPv6 packets to the host TCP/IP stack. QoS reservations can be deactivated using the AL-CNX-DEACTIVATE primitive. However, the RoHC channels persist until its TTL expires, and may be used by other QoS reservations.

5.5.4. Modules

The AL-RoHC architecture is composed of modules from the QoS-AL. The AL-RoHC has added the RoHC Module (RM) in the data plane, and the RoHC Interface Module (RIM) in the control plane. There are different instantiations of these modules for the endpoints and intermediary nodes.

5.5.4.1. RoHC Module (RM)

RM at the RoHC Channel endpoints

The RM at RoHC channel endpoints (AL_RoHC_MOD_AR and AL_RoHC_MOD_MT) intercepts incoming and outgoing data packets, and applies RoHC compression/decompression to the packets belonging to reservations for which the RoHC utilization has been previously negotiated. In addition, the RM is also responsible for adding/removing the L2.5 header. In order to identify the packets to which apply RoHC, this module maintains two state tables: the *RoHC Channels Table* (RCT), and the *Identifiers Map Table* (IMT). The RCT holds information regarding negotiated RoHC channels, namely the Decompressor MAC address, its RoHC parameters, including supported RoHC profiles. Each RoHC channel has an associated TTL value to force updates. The negotiation is handled by the RIM at the AR or MT, but the negotiation results are sent to the RM. Note that in QoS-AL the QoS reservations are always initiated by the AR. It is possible that RoHC channel negotiation is initiated by the MT, but the AL-RoHC negotiation scheme can only be used in standalone mode. Table 11 shows an example of the *RoHC Channels Table*.

Mac Address	RoHC Parameters	Timestamp	TTL
00:00:86:47:A1:83	MaxCid=...;RoHC Profiles=... (...)	07/02/2007 12:23:10	5 hours
00:00:74:C2:31:56	MaxCid=...;RoHC Profiles=... (...)	07/02/2007 15:06:24	4 hours

Table 11 - RoHC Channels Table example

The *Identifiers Map Table* maps RoHC Context Identifiers (CID) to QoS reservations (i.e. CnxIDs). This information allows the host to know which context information can be deleted when a certain reservation is tear down. A new record is inserted in the IMT when a RoHC compression operation results in a new CID. Records are deleted when the associated QoS connections are tear down.

CnxID	CID
3	12
3	8
4	7

Table 12 - Identifiers Map Table example

RM at the intermediary bridges

The RM at the intermediary bridges only performs encapsulation functions. It does not compress/decompress headers. When the packets are about to be forwarded into an Ethernet link, this module checks if the total frame size is at least 64 bytes. If not, it adds RoHC padding avoiding the Ethernet driver to add L2 padding. When the frame reaches the next AL-RoHC node, unless the frame is to be forwarded into another Ethernet link, the RM may remove the RoHC padding.

The RM at intermediary bridges does not maintain RoHC state, that is, the RCT and the IMT tables.

5.5.4.2. RoHC Interface Module (RIM)**RIM at the RoHC Channel endpoints**

The RIM is embedded in the QoS-AL and it is responsible for negotiating RoHC Channels and for establishing communication with the RM. The RIM provides an abstract interface to RM, hiding the QoS-AL. This transparency enables the possibility of adapting this to other QoS signalling frameworks without changing the RM.

The RIM at RoHC Channel endpoints (AL_RoHC_IMOD_AR and AL_RoHC_IMOD_MT) can be used to exchange the following information:

1. New RoHC channels negotiated and their parameters (from RIM to RM);
2. CnxIDs of reservations that are about to be tear down (from RIM to RM);
3. Compression ratio (from RM to RIM);
4. RoHC compression statistics (from RM to RIM);

RIM at the RoHC at intermediary bridges

The RIM at the intermediary bridges only exchange items 1 and 2 from the above list.

5.5.5. Interfaces

There are some differences in the interfaces of the MTs, ARs and intermediary bridges. The description of the AL-RoHC's interfaces can be found in Annex C.2.

5.5.6. AL-RoHC Stages

The AL-RoHC consists of three stages: 1) the RoHC Channel Negotiation; 2) the RoHC Packet Exchange; 3) RoHC Channel Termination.

5.5.6.1. RoHC Channel Negotiation

A RoHC channel establishment is triggered when the QoS Manager receives a request to setup a QoS reservation between the AR and the MT. The RoHC channel negotiation is performed by the AL-RoHC Quick Mode piggybacked negotiation scheme. This scheme consists in the AR sending AL-RoHC-REQUEST-PARAMETERS and AL-RoHC-PARAMETERS messages piggybacked on the AL-CNX-ACTIVATE message. As the PDU passes through intermediary bridges, the local QoS-AL module asks the RIM if RoHC is enabled. If not, it sets the RF flag in the AL-CNX-ACTIVATE PDU to 0. After receiving the message, the MT stores the AR RoHC parameters, checks if there is a common RoHC profile in MT and AR. If the AL-CNX-ACTIVATE RF flag is set to 1, and prepares its own AL-RoHC-PARAMETERS

message. In case the QoS reservation is accepted (resources available), a CnxID is generated and included in the response message, which is returned to the AR, and then to the QoS Manager, having the AL-RoHC-PARAMETERS message piggybacked.

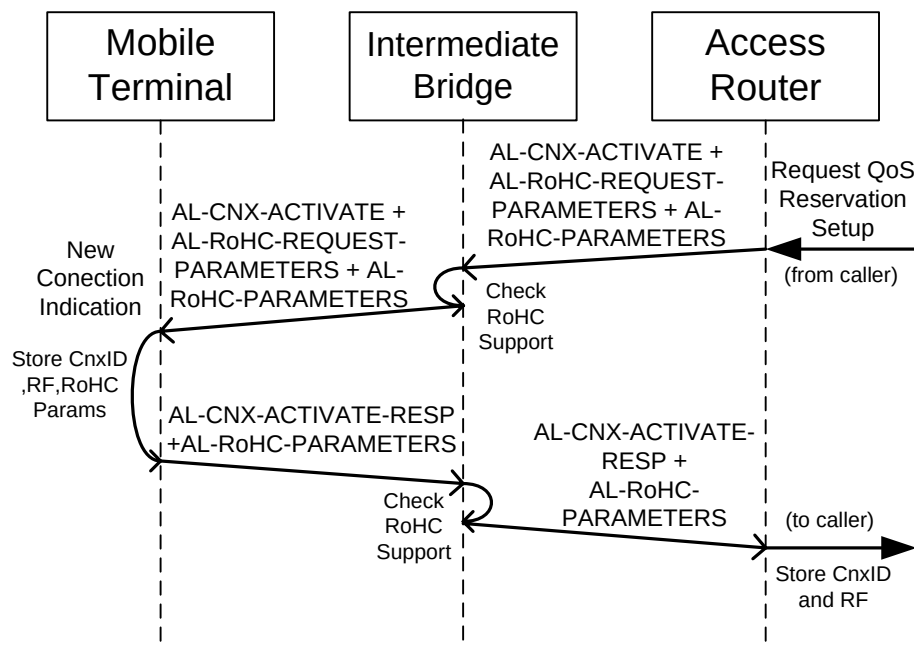


Figure 34 – RoHC Channel Negotiation (using Quick mode piggybacked)

5.5.6.2. RoHC Packets Exchange

After a successful RoHC channel establishment and a RoHC-enabled QoS reservation, both the AR and the MT may exchange compressed RoHC packets. At the RoHC channel endpoints, the outgoing IPv6 packets are intercepted and compressed by the RM. In intermediary bridges, the packets are intercepted, the L2 encapsulation changed, and padding added/stripped, if necessary. When a RoHC packet reaches the RoHC Decompressor, it is captured by the RM and decompressed. A RoHC feedback packet may be returned to the RoHC Compressor, depending on the operation mode.

In wireless communications, where link conditions vary with time, the compression ratio may vary also. Periodically the compression statistics are returned by the RM to QoS-AL, which decides if the QoS reservation needs to be modified. In order to avoid frequent changes, thresholds and a hysteresis mechanism may be used.

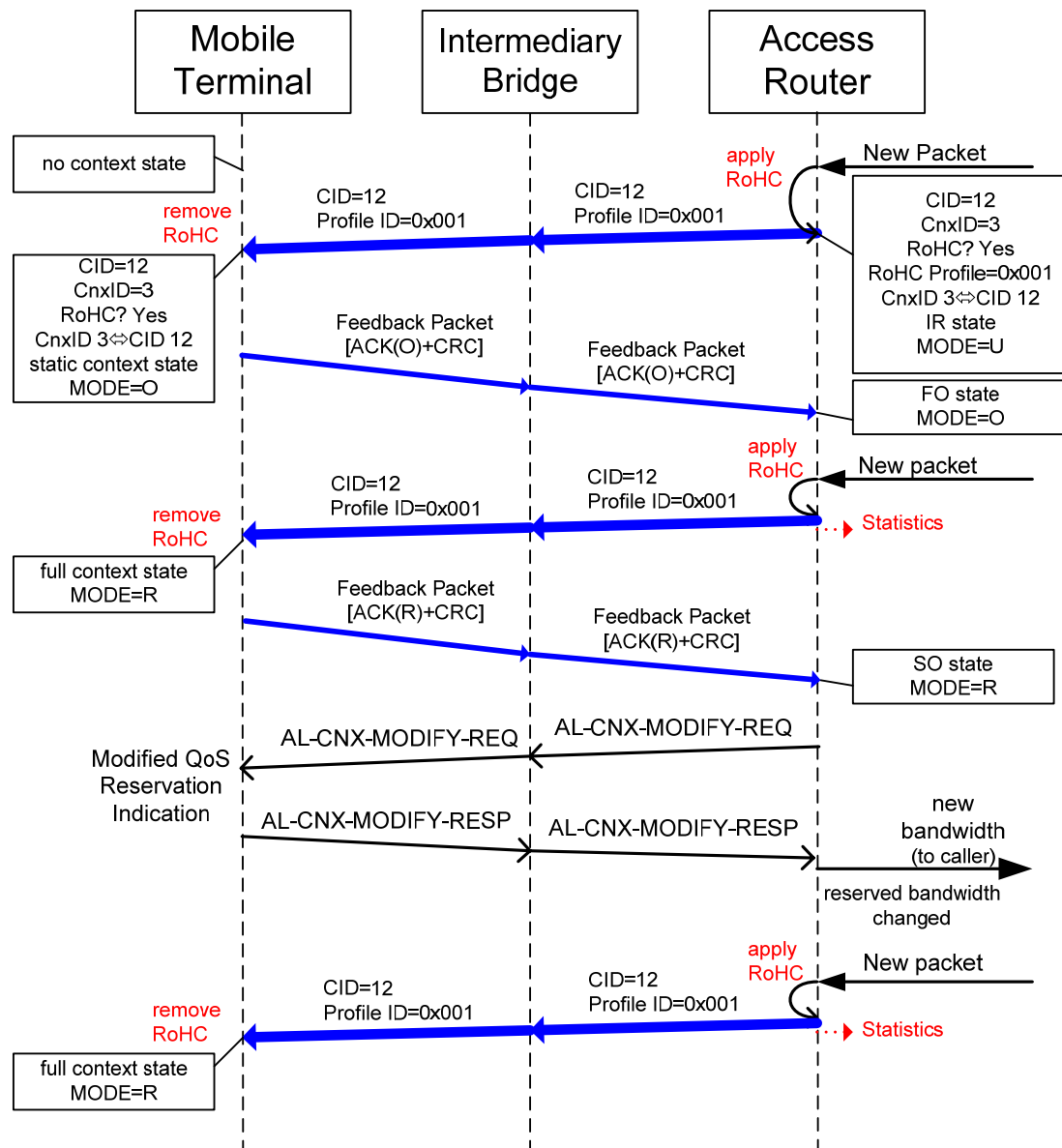


Figure 35 - RoHC Packets Exchange

5.5.6.3. RoHC Channel Termination

The RoHC channel termination can be initiated in both the AR or the MT, by sending a QoS Reservation Termination message (AL-CNXX-DEACTIVATE). As the frame is received, the action is always successful, so it does not require an answer. It deactivates the reservation and the associated RoHC mechanisms.

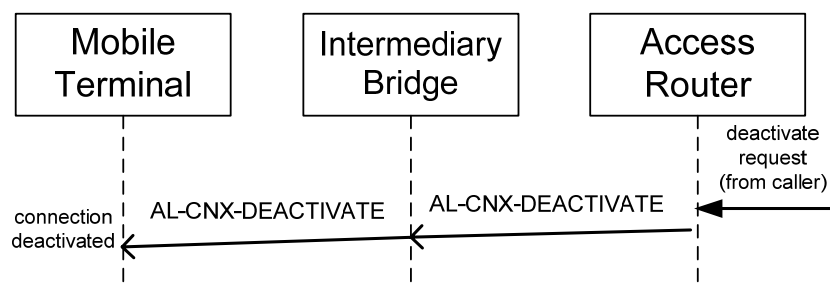


Figure 36 - RoHC Channel Termination

The RoHC channel parameters are kept by each endpoint on its RCT, and need not to be exchanged in the next QoS-AL reservation request.

5.6. Specification and Requirements

The proposed specification fulfills the requirements specified in Section 5.2. The solution uses DIX frames, which is supported by all IEEE 802 equipment (R2.1, R2.3, R1.1, R1.7), minimizes the L2 overhead (R2.3), provides packet type identification (R1.6), and supports variable packet sizes (R1.3) (except in 802.3 where a minimum of 46 bytes exists).

The generic encapsulation solution contains a L2.5 header containing a *Length* field, which can be used by bridges to infer the payload size (R1.2), and verify if padding is present (R2.3). The encapsulation proposed for the QoS-AL access networks adds a CnxID field (R3.3) to the L2.5 header. RoHC IPv6 profiles and RoHC feedback is supported (R3.1, R1.8).

The proposed RoHC negotiation mechanism (R1.4, R2.4) can be used standalone or integrated in the AL connection activation messages (R2.2, R3.4, R1.9). The utilization of RoHC in the QoS-AL reservations is optional, signalled by the RF flag (R3.2). RoHC flows are identified by the CID (R1.5), and mapped with QoS-AL connections in the IMT, at the RoHC endpoints.

5.7. Conclusion

This chapter described the AL-RoHC solution, aimed at integrating RoHC in the QoS-AL framework. This solution solves an open problem - it enables RoHC to be transported over heterogeneous networks, by adding a length field to L2 payload, thus also solving the 802.3 padding issue. It also provides a RoHC negotiation mechanism which takes advantage of the QoS-AL framework. The negotiation information is piggybacked on the QoS-AL reservation messages, what reduces the setup period to one round-trip-time, suitable for mobility scenarios. The addition of RoHC capabilities to the QoS-AL contributes to the efficient usage of the radio resources in 4G networks.

6. RoHC over IEEE 802.11

6.1. Introduction

The IEEE 802.11 is an important wireless technology, used in access networks, and it will play an important role in future 4G access networks.

The performance of RoHC over UMTS [60][57] and over wireless links modelled using randomly distributed errors [56][58][61] has been studied, but there are no studies of RoHC over IEEE 802.11. The IEEE 802.11 has some features which makes it different from the other technologies for which RoHC has been studied namely a shared medium controlled CSMA/CA, and an almost reliable frame transmission service. These characteristics have an impact on RoHC's performance which is studied in this chapter.

The performance evaluation of RoHC over IEEE 802.11 **Error! Reference source not found.** is carried out through simulation, using a special branch of Network Simulator 2 (NS2) [31] which includes an implementation of IEEE 802.11a [70].

The remainder of this chapter is structured as follows: in Section 7.2 the simulation scenario and the simulation assumptions are presented. In Section 7.3 the simulation model is introduced, including the simulation modules and the simulation parameters. The experiments carried out and the results obtained are described in Section 7.4. Then, in Section 7.5, we validate results obtained. The conclusions of the chapter are presented in Section 7.6..

6.2. Simulation Scenario and Assumptions

The main goal of this simulation is to study the RoHC performance over IEEE 802.11 wireless LANs. More precisely, to study how the RoHC performance is influenced by 1) the Distributed Coordination Function (DCF) which implements the distributed CSMA/CA mechanism, and 2) by the acknowledged delivery service of IEEE 802.11.

This RoHC simulation was limited to the U-mode. The other RoHC modes, namely the R-mode and the O-mode, would provide extra robustness and compression efficiency, but are considerably harder to implement. Furthermore, U-mode is more suitable for real-time traffic and for mobile environments because it avoids feedback packets.

In the simulation experiments, RoHC is applied to VoIP traffic. VoIP was chosen because: 1) its use in future 4G networks is expected to be significant, 2) it generates small IP packets, the type of traffic which benefits from header compression, and 3) it consists of real-time traffic, which is sensible to delay and jitter. Because VoIP traffic demands little bandwidth, some simulations also include other traffic flows as

background traffic. Using it we can easily increase the network loads, and study the RoHC performance when the network is more close to its limit capacity. Only the RoHC UDP/RTP profile is supported because RoHC was only applied to VoIP traffic.

Figure 37 illustrates the simulated network architecture, which represents an IEEE 802.11 access network. A number of Mobile Terminals (MTs) are connected to an Access Router (AR) through an Access Point (AP). The AP and the AR are connected by an Ethernet link. Each MT accesses the Internet via the AR.

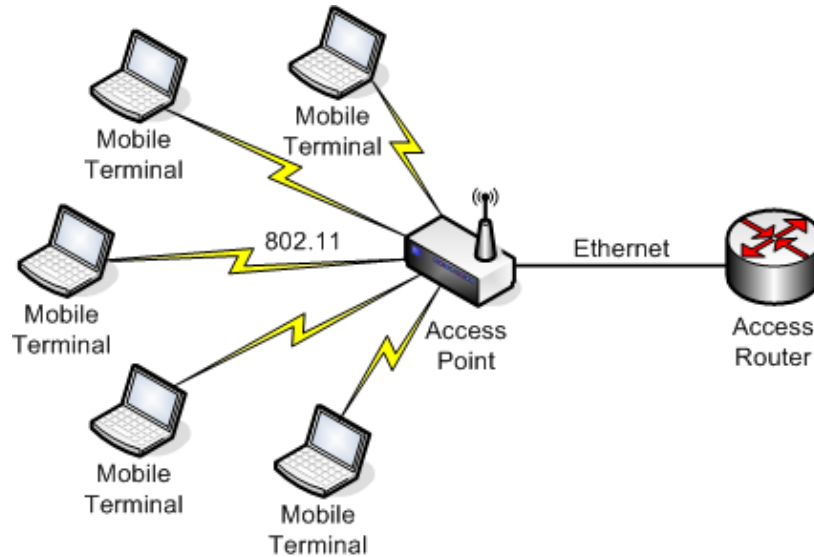


Figure 37 - Simulation Scenario

6.3. Simulation Model

The model used for simulating RoHC over IEEE 802.11 was implemented in an NS2 branch, which includes the IEEE 802.11a model developed by Mathieu Lacage [70]. NS2 was chosen because it is a well known and accepted network simulator. Mathieu Lacage's model was chosen over NS2's default IEEE 802.11 model because it implements IEEE 802.11a and supports multi-rate. It also implements IEEE 802.11e which was not used in this work, but may be important for future work.

The Simulation architecture is presented in Figure 38. From top to bottom, a *RoHC Node* contains an application layer which can generate VoIP, UDP-based traffic with exponential interarrival times, and FTP/TCP traffic. At the network layer, either the UDP or the TCP agent is used over IPv6. If the traffic is VoIP, then it is forwarded to the RoHC module, where it is header compressed before being sent to the 802.11a MAC Layer. On the reverse direction, RoHC packets arriving from the 802.11a MAC Layer, are handled by the RoHC module where they are decompressed. Afterwards, they are sent to the UDP Agent which delivers them to the VoIP application.

The RoHC module handles the RoHC flow information and operation, including the maintenance of RoHC contexts, compression/decompression, and the gathering of statistics. The *NetNode* contains the functionality required by the MT to exchange packets with other MTs/APs, including the MAC layer protocol, the 802.11a physical layer modelling, and the radio link model.

The 802.11a MAC layer contains an upstream queue. This model does not limit the size of this queue. In order to set up an upper bound in the traffic delay, we have limited the upstream queues of all the nodes to 400 packets.

To model the VoIP codec, we used a source which generates exponentially distributed traffic.

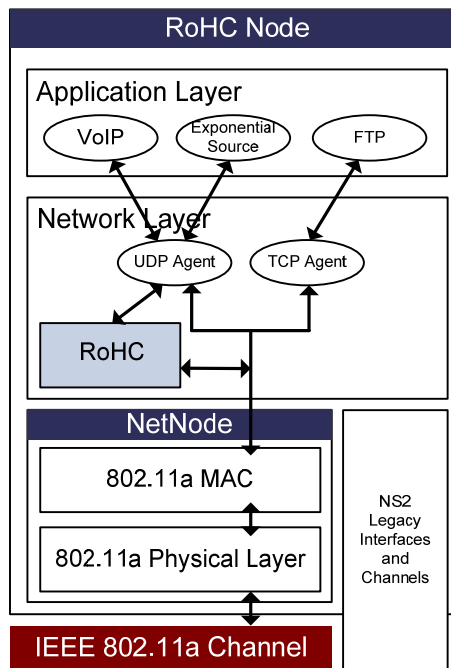


Figure 38 - Simulation Model

6.3.1. VoIP Modelling

The VoIP codecs candidates for the simulation experiments were selected based on the following criteria: 1) codecs should generate packets having small payload sizes, the type of traffic which benefits from header compression; 2) codecs should be supported by most of the VoIP equipment and software; 3) codecs should demand low processing power; 4) codecs should not have a Voice Activity Detection (VAD) mechanism, because this feature would create an additional variable which would make the analysis of the RoHC performance more complex to evaluate.

The ITU-T G.729 [67], the ITU-T G.711 [68] and the ITU-T G.723.1 [69] are popular VoIP codecs [65]. Table 13 presents their characteristics.

Codec	Payload Size (bytes)	Sample Period (ms)	Codec Bandwidth (kbit/s)	CPU demand
ITU-T G.711 (PCM)	160	20	64	Low
ITU-T G.723.1 (MP-MLQ)	24	30	6.3	High
ITU-T G.723.1 (AC-ELP)	20	30	5.3	High
ITU-T G.729A (CS-CELP)	10	10	8	Low
ITU-T G.729A (CS-CELP)	20	20	8	Low

Table 13 - VoIP Codecs candidates

The G.711 codec does not meet the criteria because it generates large payloads. The G.729A codec was preferred over the G.723.1 codec because the former has smaller payload size and demands lower CPU consumption. The G.729A/20 byte payload generates the same throughput as the G.729A/10 byte version, but half the number of packets, because each packet contains two data frames of 10ms. The maximum throughput of an IEEE 802.11 WLAN depends on the average frame size.

The lower the average frame size is, the lower will be the maximum throughput of the 802.11 WLAN. Considering also this factor, we selected the G.729A/20 byte codec.

6.3.2. Simulation Parameters

The simulation parameters are classified into three groups: RoHC parameters, VoIP parameters, and other parameters.

6.3.2.1. RoHC Parameters

The U-mode state machine was presented in Figure 9 and Figure 10. This State machine has 3 configurable parameters: 1) the IR_TIMEOUT, which is the maximum time the RoHC Compressor state machine may stay in the FO or the SO states before returning to the IR state; 2) the FO_TIMEOUT, which is the maximum time the RoHC Compressor state machine may stay in the SO state before returning to the FO state; 3) the L_VAR, which is the number of packets that must be sent in the IR and in the FO state before the state machine advances to the next compression state. For better compression efficiency, these parameters have to be adjusted to the low packet loss characteristics of IEEE 802.11.

These parameters are defined in the RoHC standard [1] as implementation dependent and no values are recommended. Some publications [56][57][59] proposed values for these parameters but their values were obtained for RoHC over UMTS or abstract wireless links with random errors. The unreliability of these transmission media require lower U-mode timeout intervals, in order to contain the loss propagation events. However the acknowledged frame transmission service of IEEE 802.11 makes packet loss a rare event. Thus, the U-mode timeout intervals may be set to large values. Therefore, the parameters suggested in the literature may not be the optimum values for 802.11 links. Nevertheless, the departing values were assumed based on results published in [56], that is:

- IR_TIMEOUT: 5s
- FO_TIMEOUT: 3s
- L_VAR: 2 packets

6.3.2.2. Other Parameters

As mentioned in Section 6.3.1, the ITU-T G.729A/20 byte payload version was the VoIP codec selected. According to the characteristics of this codec, the following parameters and corresponding values were configured:

- Generated bitrate: 8 kbit/s
- Packet payload size: 20 byte

6.3.2.3. General Parameters

Our simulation scenarios use IPv6 at the network layer. Therefore the G.729 VoIP packets at the network layer will have a size of 80 byte: 40 byte for the IPv6 header, 12 byte for the UDP header, 8 byte for the RTP header, plus 20 byte for the G.729A payload.

The simulations were executed for an 802.11 Request-To-Send (RTS) / Clear-To-Send (CTS) CTS threshold set to 1000 bytes. This means that all the frames having a size over 1000 byte must be preceded by a RTS/CTS frame sequence, used to prevent the hidden terminal problem [71]. However, this mechanism decreases the 802.11 link

goodput. In some particular experiments this mechanism was disabled by setting the RTS/CTS Threshold to 2304 bytes, the maximum frame size allowed in IEEE 802.11.

6.4. Experiments and Results

Three groups of simulation experiments were carried out: 1) throughput experiments, to measure IEEE 802.11a network throughput; 2) RoHC performance experiments, to determine the RoHC U-mode performance over IEEE 802.11; 3) RoHC U-mode parameters experiments, to obtain the optimum values for RoHC U-mode parameters over IEEE 802.11.

6.4.1. Throughput Experiments

The throughput experiments are divided in two types: Maximum Throughput Experiment, and RoHC Throughput. Their goals are respectively to determine the maximum throughput of the IEEE 802.11a link, and to measure the reference scenario throughput.

6.4.1.1. Maximum Throughput Experiment

As illustrated by Figure 39, the Maximum Throughput Experiments consist of a single unidirectional UDP flow between the Access Router (AR) and a Mobile Terminal (MT) connected through the 802.11 AP. To determine the maximum throughput of the network, the bitrate of the UDP flow was increased gradually until the further increase of the UDP flow bitrate did not result in higher throughputs. Since the 802.11 link provides a bandwidth smaller than the Ethernet link, the maximum throughput obtained in the simulation characterizes the 802.11 link maximum bandwidth.

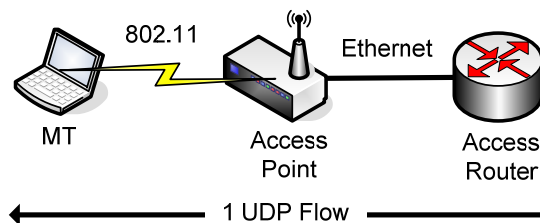


Figure 39 – Maximum Throughput Experiment

Figure 40 shows that the maximum throughput obtained through this experiment was about 16.3 Mbit/s when using a RTS/CTS threshold of 1000 bytes, and 22.3 Mbit/s when disabling the RTS/CTS mechanism. In the first case, the UDP flow consisted of 1500 byte packets and, in the second case, 2304 byte packets. An 802.11a link is capable of sustaining 54 Mbit/s at physical layer. However, a significant part of it is spent on L1/L2 overheads and control frames

UDP flows consisting of smaller payloads would result in smaller maximum throughputs. The throughput of an 802.11 link is maximized when maximum size packets are sent. As shown in Figure 41 the disabling of RTS/CTS mechanism has resulted in 36.8% gain $\left(\frac{22.3 - 16.3}{16.3} \right)$ over the previous (Figure 40) maximum throughput.

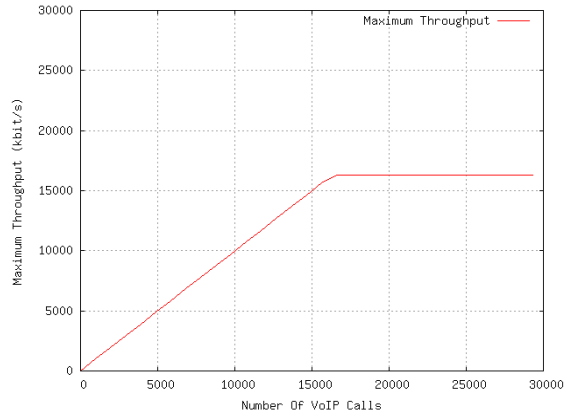


Figure 40 – Maximum Throughput Experiment (1500 byte packets and RTS/CTS Threshold of 1000 bytes)

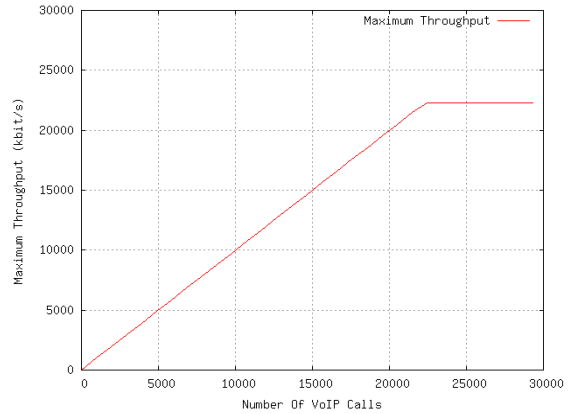


Figure 41 - Maximum Throughput Experiment (2304 byte packets and RTS/CTS disabled)

6.4.1.2. RoHC Throughput Experiment

The objective of the RoHC Throughput Experiments was to determine the maximum throughput of the network using the same scenarios of the RoHC Performance experiments, using UDP as background traffic. This scenario is illustrated by Figure 42. Now multiple nodes and flows are considered.

A number of RoHC Compressed VoIP calls are established between the AR and the MTs. Additionally, an unidirectional UDP flow is established between the AR and a MT (*MT_load*), to represent background traffic. Between simulation runs the number of VoIP calls is varied between 1 and 100. Each call generates two RoHC flows, one for each direction. Each MT generates one call.

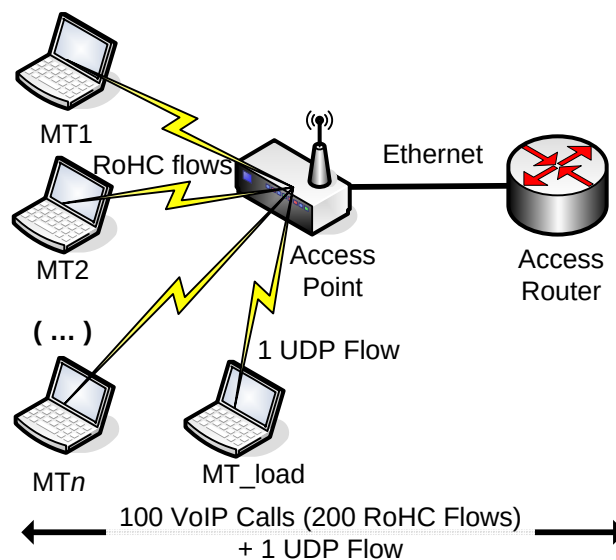


Figure 42 – RoHC Throughput Experiment

Figure 43 shows that the maximum throughput is obtained in the simulation run with a single VoIP call. This confirms that the efficiency of the 802.11 link decreases with the number of MTs accessing to the shared medium. Using a single RoHC compressed VoIP call and the UDP background flow, a maximum throughput of 16.2 Mbit/s was obtained. But as more VoIP calls are added, the maximum throughput decreases. With 100 VoIP calls, the maximum throughput is only about 12.6 Mbit/s, which represents a reduction of 22.2% $\left(\frac{16.2-12.6}{16.2}\right)$ relative to the maximum throughput obtained in the simulation run using a single VoIP call. As more VoIP calls are added to the network, smaller packets are introduced into the 802.11 link and, consequently, smaller throughputs are obtained.

In this scenario only the UDP flow frames are preceded by a RTS/CTS sequence, since the UDP flow consists of 1500 byte packets. The header compressed VoIP packets are small and do not exceed the RTS/CTS threshold of 1000 bytes. As shown by Figure 44, if the RTS/CTS is disabled and the UDP flow is configured to use 2304 byte packets, the maximum throughput obtained is 22.2 Mbit/s, which corresponds to an 37% increase $\left(\frac{22.2-16.2}{16.2}\right)$ of the maximum throughput.

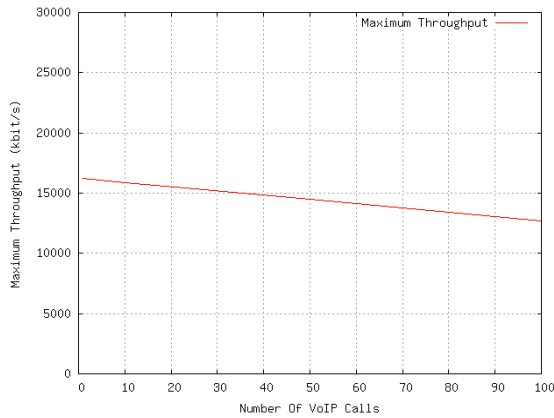


Figure 43 – RoHC Throughput Experiment
(1500 byte UDP packets and RTS/CTS threshold of 1000 bytes)

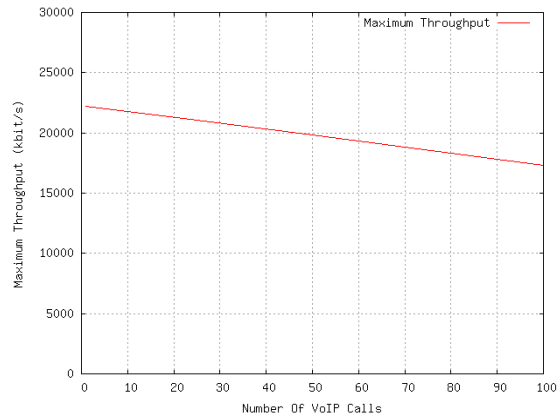


Figure 44 - RoHC Throughput Experiment
(2304 byte UDP packets and RTS/CTS disabled)

6.4.2. RoHC Performance Experiments

The RoHC Performance Experiments aim to characterize the performance of RoHC U-mode in terms of gain and additional delay and jitter, when assuming VoIP traffic over an IEEE 802.11 link.

This dissertation defines *RoHC's gain* as the amount of bandwidth that becomes available for other flows after applying RoHC, and it is quantified as a percentage of the offered load, before compression. Reducing the bitrate of a flow by a certain amount does not mean that the amount will be available for other flows. This is the case of IEEE 802.11, where the L1 and L2 overheads are significant. Thus, the RoHC's gain can be calculated using two methods: 1) considering only the L3 header and payload sizes (i.e. MAC payload); 2) considering the L1/L2/L3 and the medium access protocol overheads. The later can be considered more realistic.

Three simulation sets were performed, using the scenario illustrated in Figure 45:

- a) **No background traffic:** this simulation aims at characterizing the RoHC's gain assuming no background traffic. In this case, only the MAC payload is considered to calculate the RoHC's gain.
- b) **Background TCP traffic:** this simulation calculates the RoHC's gain using background TCP traffic. In particular, the bandwidth freed by RoHC is evaluated. The TCP source has a very large file to transmit, what causes the use of all the bandwidth available, according to the TCP congestion control mechanisms, and during all the simulation. Using this method, we calculate the RoHC's gain taking into consideration L1/L2 effects.
- c) **Background UDP traffic:** this simulation set adds an UDP flow as background traffic, with a bitrate near the maximum network throughput. In opposition to TCP, UDP does not include congestion control mechanisms. Therefore, as more VoIP traffic is generated, the network reaches a congestion point, resulting in packet losses. Thus, the main goal of this simulation set is to study RoHC's gain under the presence of packet loss. In this simulation set, we calculate the RoHC's gain only using the MAC payload sizes.

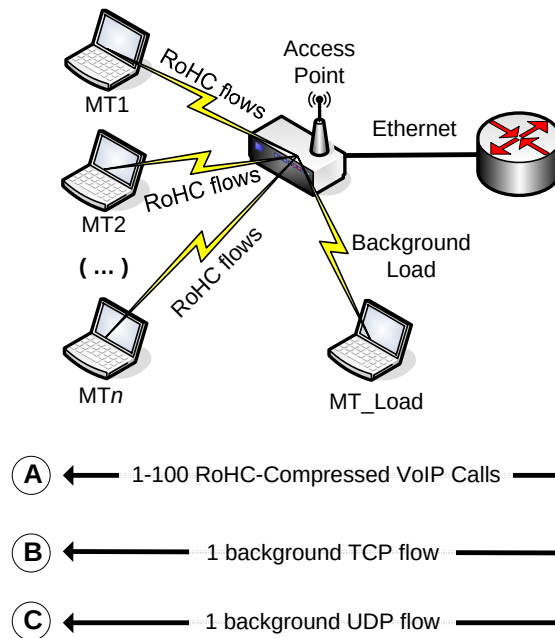


Figure 45 – RoHC Performance Simulation Sets

All the simulations were executed with 1500 byte TCP and UDP packets, and a RTS/CTS Threshold of 1000 bytes, except when explicitly stated otherwise.

The compression ratio of a RoHC packet is calculated by subtracting to one, the quotient of dividing the size of the resulting RoHC packet after compression by the size of the uncompressed IP packet, including header and payload.

$$\text{CompressionRatio} = 1 - \frac{\text{CompressedIPPacketSize}}{\text{UncompressedIPPacketSize}}$$

The delay includes the time the packet waits in the sender queue plus the transmission and propagation times. Jitter is calculated as the average difference between the delays of two consecutive packets.

When a RoHC packet reaches the Decompressor, if the associated context is out of synchronization, the packet is dropped. These packet losses are designated as Loss Propagation and only occur because RoHC was used.

6.4.2.1. RoHC Performance With No Background Traffic

In this experiment only RoHC compressed VoIP traffic is used. The RoHC's gain is calculated considering only the L3 header and payload sizes, and it is the amount of saved bandwidth when applying RoHC to the VoIP traffic, considering the effect of the loss propagation. The RoHC's gain is normalized to the throughput of the VoIP traffic without compression. This is described by the following equation:

$$\text{(Equation 1) } RoHCGain = \frac{(VoIPNoCompression - VoIPCompression) - LossPropagation}{VoIPNoCompression}$$

VoIPNoCompression: the mean bitrate of the VoIP traffic when RoHC is not applied, in kbit/s.

VoIPCompression: the mean bitrate of the VoIP traffic when RoHC is applied, in kbit/s.

LossPropagation: additional packet loss, in kbit/s, due to the Decompressor context synchronization.

In this experiment the RoHC's gain is 71% and remains constant regardless of the number of VoIP calls in the network, as shown in Figure 46. We can also see that in this experiment the RoHC's gain is similar to the RoHC compression ratio, illustrated in Figure 46 and Figure 47, respectively. This happens because: 1) only the MAC payload was considered in the RoHC's gain calculation; 2) the traffic load is low and, as such, the packet loss ratio is nearly zero.

Figure 51 shows the packet loss; it is inferior to 0.1%. This packet loss results from full queues in the sender, and not from transmission errors. However, this small amount of packet loss resulted in 2% of loss propagation in the simulation run with 100 VoIP calls. Nevertheless, the difference in the RoHC's gain with 90 VoIP calls and 100 VoIP calls is not significant, as shown in Figure 46.

In Figure 48, we can see that the bitrate of the VoIP traffic. It increases linearly with the number of VoIP calls, but the line representing uncompressed VoIP traffic has a larger derivative. The difference between the two lines for a given number of VoIP calls represents the profit of using RoHC, in kbit/s. Without RoHC compression, 100 VoIP calls register a throughput of 1.56 Mbit/s, whereas with RoHC compression only 0.44 Mbit/s is required. These values are much lower than the 12.7 Mbit/s, the maximum throughput of the network, as shown in Section 6.4.1.2.

Figure 49 and Figure 50 show that RoHC does not have a significant impact on delay or jitter. Due to smaller packet sizes, RoHC slightly improves delay parameters, benefiting from reduced propagation times and 802.11 collisions.

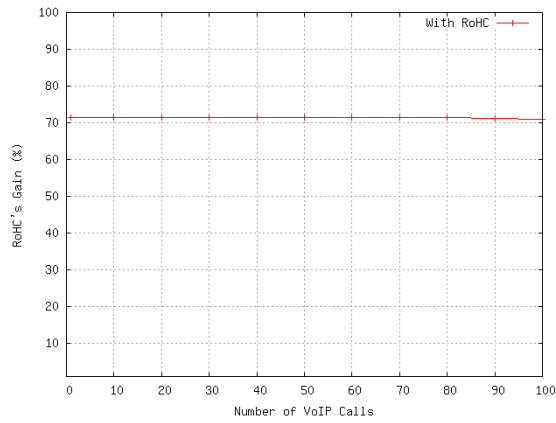


Figure 46 – RoHC's Gain (without background traffic)

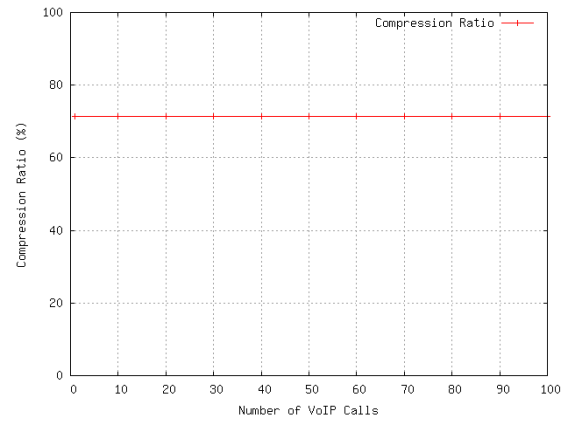


Figure 47 – RoHC Compression Ratio (without background traffic)

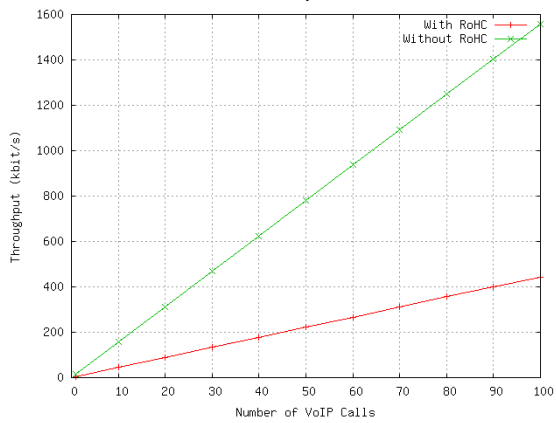


Figure 48 – Throughput (without background traffic)

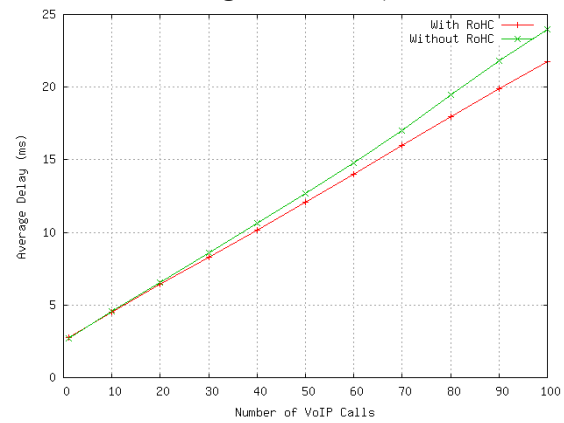


Figure 49 – Average Delay (without background traffic)

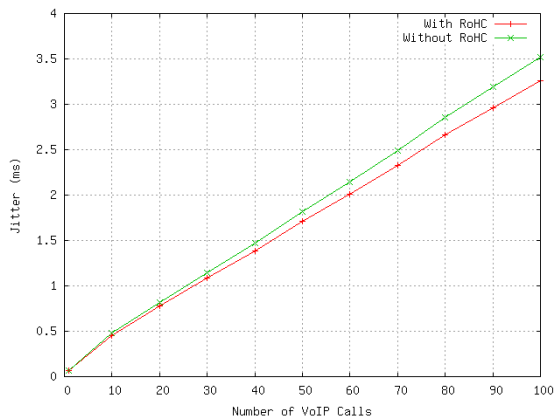


Figure 50 – Jitter (without background traffic)

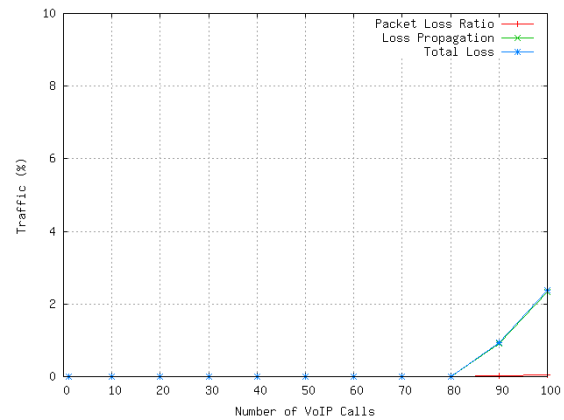


Figure 51 – Packet Loss (without background traffic)

6.4.2.2. RoHC Performance With TCP as Background Traffic

The introduction of background traffic in the previous experiment makes the scenario more realistic. Since more bandwidth is used the network utilization approaches the maximum network throughput, and the packet loss becomes relevant. However, because TCP was used as background traffic, this experiment will not register significant packet loss ratios, since the TCP sender adjusts the flow rate to the bandwidth available.

The goal of this simulation set was to evaluate how packet loss increases when adding TCP background traffic, and to use this TCP traffic to measure the RoHC's gain taking into consideration also the L1/L2 overheads. By measuring the difference in the TCP throughput between the simulation using RoHC compressed VoIP calls and background TCP, and the simulation with uncompressed VoIP traffic and background TCP, we can evaluate how much bandwidth is freed. Thus, considering also the L1/L2 overheads, we define the RoHC's gain as:

$$\text{(Equation 2) } RoHCGain = \frac{(TCPCompression - TCPNoCompression) - LossPropagation}{VoIPNoCompression}$$

TCPCompression: the mean bitrate of the TCP flow, when the VoIP traffic is RoHC compressed, in kbit/s.

TCPNoCompression: the mean bitrate of the TCP flow, when the VoIP traffic is not RoHC compressed, in kbit/s.

VoIPNoCompression: the mean bitrate of the VoIP traffic, when RoHC is not applied, in kbit/s.

LossPropagation: additional packet loss, in kbit/s, due to the Decompressor context being out-of-sync.

When RoHC compression is applied to the VoIP traffic, the TCP sender is able to send more bytes. This means that *TCPCompression* will be higher than *TCPNoCompression*, which is the reason because in Equation 2, the terms *TCPCompression* and *TCPNoCompression* are inverted when compared to Equation 1.

The RoHC's compression ratio, presented in Figure 52, resulted in 71%, similar to the value obtained in the previous experiment. It puts in evidence two facts: 1) the RoHC's compression ratio remains constant, regardless of the packet loss ratio; 2) using the compression ratio as gain metric seems to be unrealistic.

Considering the IEEE 802.11 L1/L2 overheads, as presented in Figure 53, the RoHC's gain is 30%. In this case, the TCP background traffic was composed by 1500 byte packets. The difference between the two metrics is explained by two factors: 1) the RoHC Gain metric takes into consideration the Loss Propagation, which is a consequence of packet loss; 2) the RoHC Gain metric takes into consideration the 802.11 L1/L2 overheads.

The 802.11 L1 PLCP header and preamble uses 16 bytes, the L2 Header needs 34 bytes, including the CRC; the IPv6/UDP/RTP headers use 60 bytes. This results in 110 bytes of headers per VoIP payload. Moreover, the L1 header is always transmitted in basic rate, which in IEEE 802.11a is 6 Mbit/s. Finally, the CSMA/CA nature of DCF and the use of RTS/CTS introduce additional overhead.

If, instead of using 1500 byte packets in the TCP flow, 80 byte packets are used the percentage of small IP packets would increase. Although 80 byte packet TCP flows is not realistic, it is useful for our studies since this traffic can "emulate" VoIP packets. The RoHC's gain reflects how much additional VoIP traffic we can add to the 802.11 medium. Figure 54 illustrates the results obtained with 80 byte TCP packets. In this case, the RoHC's gain is about 5%.

The compression ratio is higher with smaller application payloads, but if the number of small packets in the network is high, the overall 802.11 transmission efficiency suffers due to the L1/L2 effects described above.

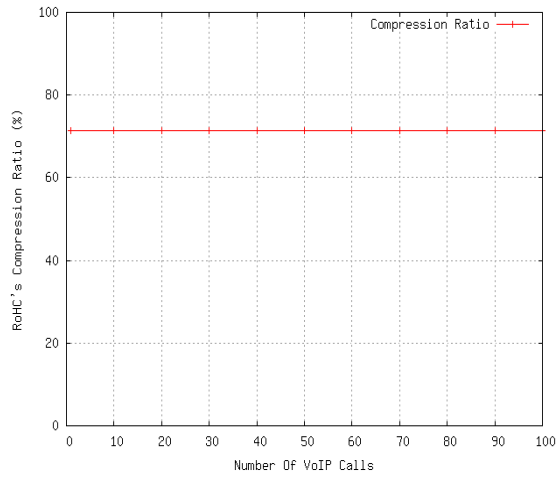


Figure 52 – RoHC Compression Ratio (RoHC + background TCP traffic)

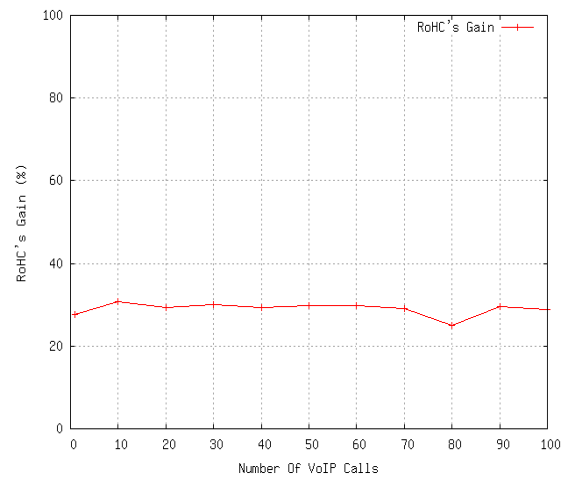


Figure 53 – RoHC's Gain (RoHC + background TCP traffic – 1500 byte TCP packets)

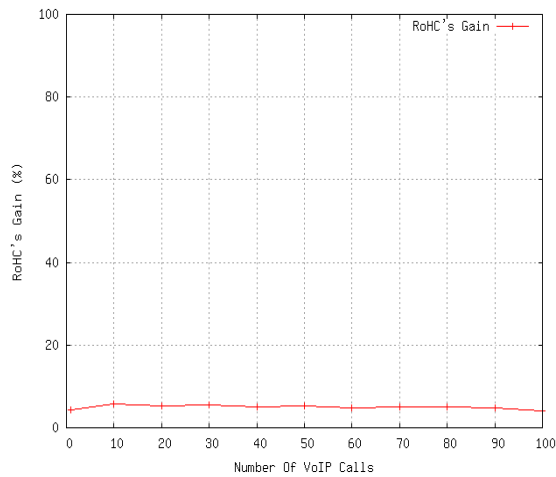


Figure 54 – RoHC's Gain (with background TCP traffic - 80 byte TCP packets)

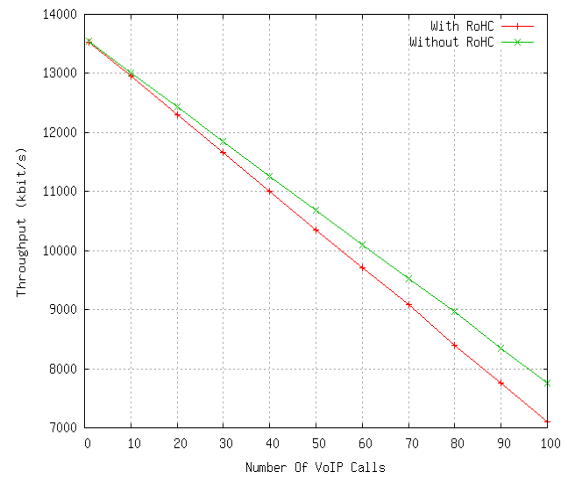


Figure 55 – RoHC + TCP Throughput

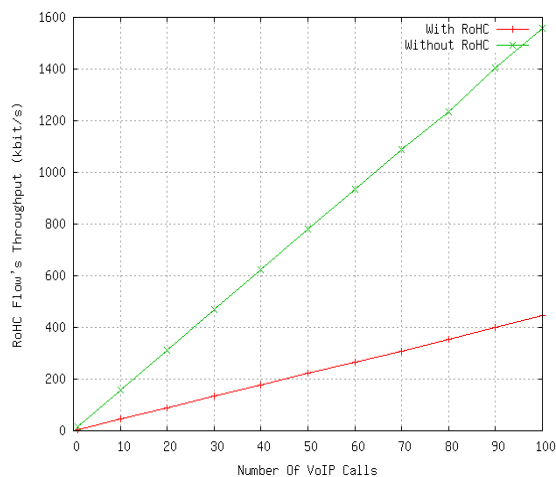


Figure 56 - RoHC Throughput (with background TCP traffic)

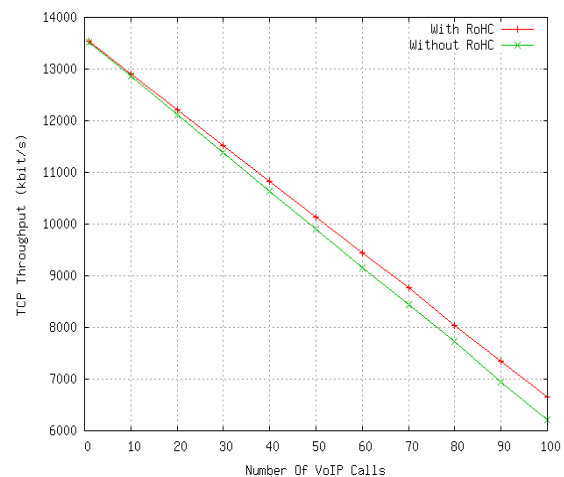


Figure 57 – TCP Throughput (with RoHC traffic)

In Figure 55, the throughput decreases linearly with the number of VoIP calls in the network. The TCP flow has its bitrate reduced when the overall VoIP traffic increases in the network. Note that the amount of the TCP traffic is several times the amount of VoIP traffic. Figure 57 shows that even with 100 VoIP calls in the network,

TCP traffic still uses around 6.5 Mbit/s while, in Figure 56, it is shown that even without compression, 100 VoIP calls only uses 1.6 Mbit/s. The weight of total TCP traffic explains why Figure 55 resembles Figure 57. Nevertheless, as shown in Figure 55, using RoHC always saves bandwidth. Table 2 resumes the throughput of Simulation Set B.

	1 Uncompressed VoIP Call (Mbit/s)	100 Uncompressed VoIP Calls (Mbit/s)	1 Compressed VoIP Call (Mbit/s)	100 Compressed VoIP Calls (Mbit/s)
TCP throughput	13.524	6.654	13.529	6.654
VoIP throughput	0.015	1.558	0.004	0.444
Total throughput	13.540	7.762	13.533	7.099

Table 14 – Throughput in Simulation Set B (1500 byte TCP packets)

Table 15 and Table 16, and Figure 58 and Figure 59 confirm that RoHC does not worsen average delay and jitter values; it improves them slightly.

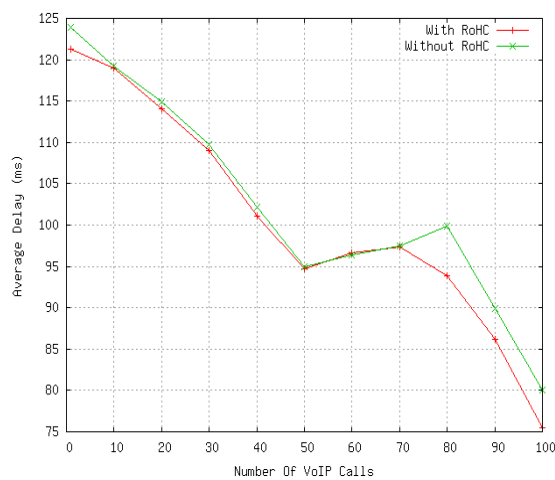


Figure 58 – Average delay (RoHC + background TCP traffic)

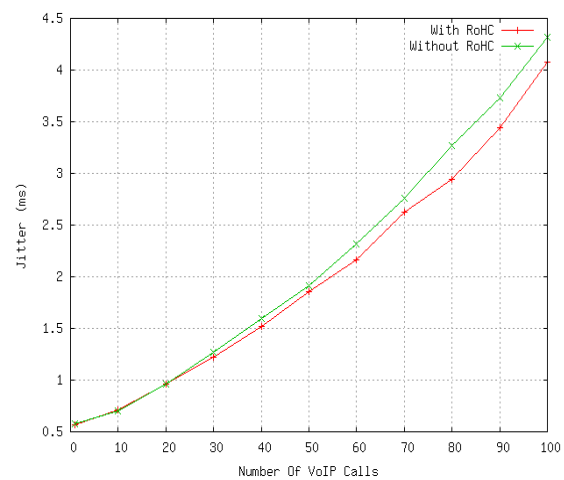


Figure 59 – Jitter (RoHC + background TCP traffic)

	1 Uncompressed VoIP Call (ms)	100 Uncompressed VoIP Calls (ms)	1 Compressed VoIP Call (ms)	100 Compressed VoIP Calls (ms)
TCP average delay	123.99	74.10	121.34	70.50
VoIP average delay	112.47	82.45	113.11	77.59
Total average delay	123.86	80.03	121.30	75.43

Table 15 – Average Delay in Simulation Set B

	1 Uncompressed VoIP Call (ms)	100 Uncompressed VoIP Calls (ms)	1 Compressed VoIP Call (ms)	100 Compressed VoIP Calls
TCP average jitter	0.55	1.36	0.55	1.26
VoIP average jitter	2.75	5.53	2.68	5.32
Overall network average jitter	0.58	4.32	0.57	4.08

Table 16 – Jitter in Simulation Set B

The packet loss ratio has remained below 0.1%, due to the TCP congestion control mechanism. In the simulation run with higher packet loss ratios (i.e. 70 VoIP calls), the loss propagation reached 2%.

6.4.2.3. RoHC Performance Experiments With UDP as Background Traffic

The Simulation Set C uses UDP as background traffic. Since UDP does not have any congestion mechanism, when the offered throughput exceeds the 802.11a link capacity, the upstream queues start to get full. When the queues reach the length of 400 packets, the new arriving packets are dropped. This simulation set aims at studying the impact of these packet losses in the RoHC's gain.

In order to experience a considerable amount of packet loss, a background UDP flow characterized by a mean bitrate of 14 Mbit/s was selected. This value was selected to be between the maximum throughputs obtained with one and hundred VoIP calls, respectively 16.2 Mbit/s and 12.6 Mbit/s, as shown in Figure 43. Having this UDP stream as background UDP traffic, we can observe that the packet loss ratio increases exponentially with the number of VoIP calls, as shown in Figure 61. Packet loss was observed for more than 50 VoIP calls.

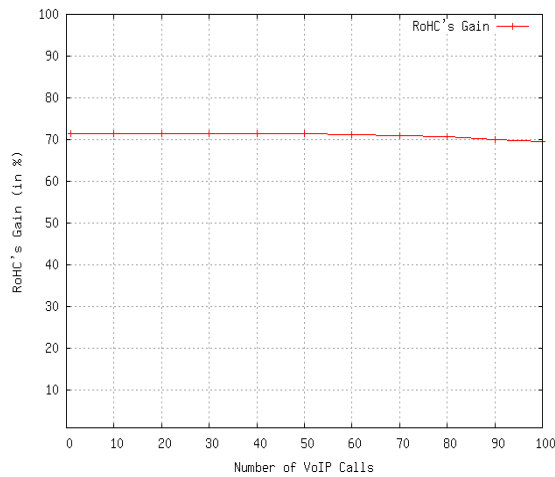


Figure 60 – RoHC's Gain (with background UDP traffic)

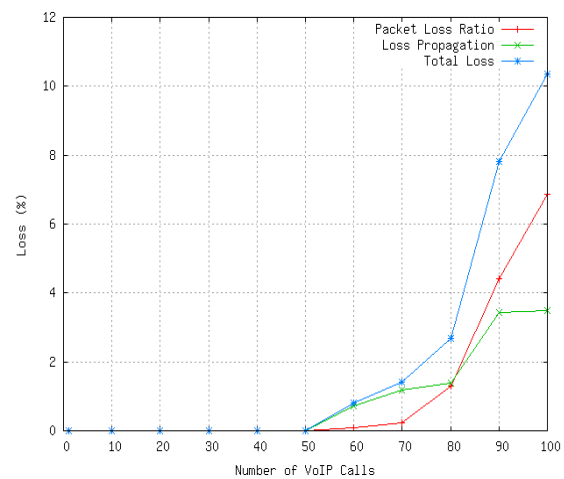


Figure 61 – Packet Loss and Loss Propagation (RoHC + background UDP traffic)

Figure 60 presents the RoHC's gain according to Equation 1. As expected the gain is constant for simulation runs where no packet loss is observed; it starts dropping for simulations with more than 50 VoIP calls. With packet loss ratios of 6.8% and loss propagation of 3.8%, the RoHC's gain has dropped about 2%, from 71% to 69%.

If the packet loss was uniformly distributed, such high packet loss ratios would be enough to reduce substantially the RoHC's gain. In IEEE 802.11 links, the packet losses are caused mostly by the packet drops at the sender queues, and not from transmission errors. In this case, the packet drops occur in short bursts. This explains why, in Figure 62, the loss propagation is not higher.

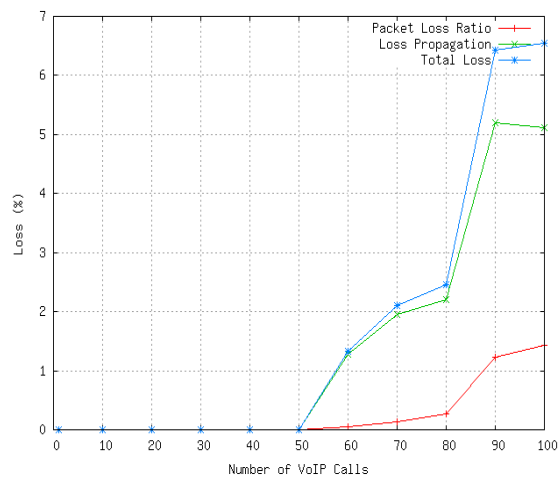


Figure 62 – RoHC Packet Loss and Loss Propagation (with background UDP traffic)

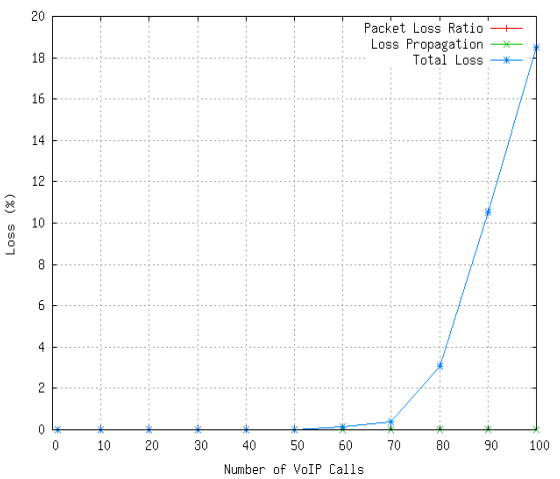


Figure 63 – UDP Packet Loss (with RoHC traffic)

	Packet Loss with RoHC (%)			Loss Propagation with RoHC (%)		
	1 call	80 calls	100 calls	1 call	80 calls	100 calls
UDP	0%	3.07%	18.52%	-	-	-
VoIP	0%	0.26%	1.43%	0%	2.20%	5.11%
Total traffic	0%	1.29%	6.87%	0%	1.39%	3.49%

Table 17 - Packet Loss and Loss Propagation in Simulation Set C

Figure 62 and Figure 63 and in Table 17 represent percentages of the offered traffic.

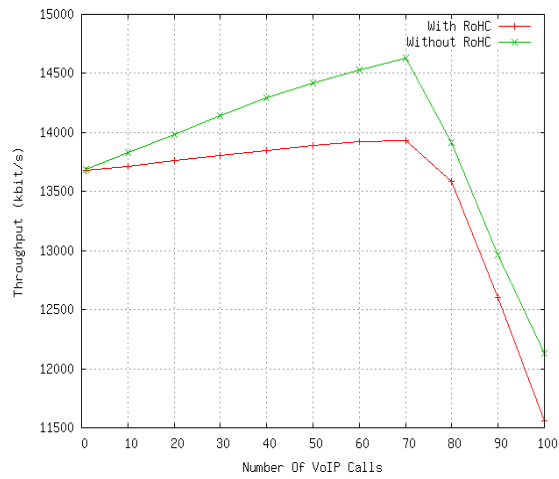


Figure 64 – RoHC + UDP Throughput

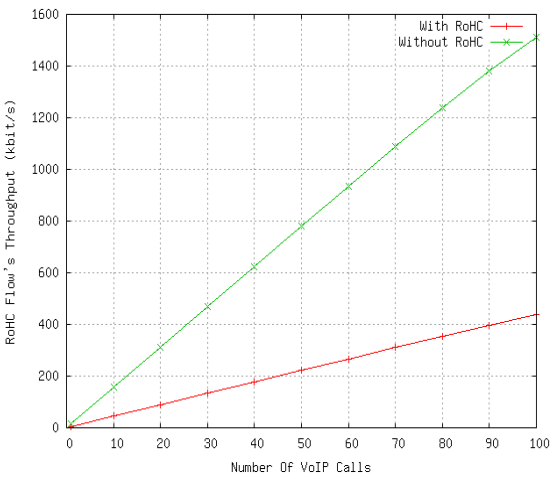


Figure 65 – RoHC Throughput (with background UDP traffic)

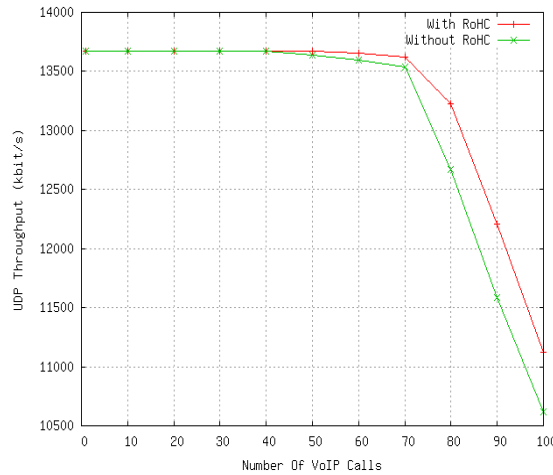


Figure 66 – UDP Throughput (with RoHC traffic)

As shown in Figure 62, Figure 63, and in Table 17, with 80 calls the RoHC compressed VoIP traffic experiences around 0.26% of packet loss, resulting in 2.20% of loss propagation. This adds up to 2.46% (0.26% + 2.20%) total packet loss, which may be considered bearable. But with 100 VoIP calls, the total packet loss reaches 6.54%, which is a significant value, for which the VoIP quality is affected.

In Figure 64 we can see that the network throughput increases linearly until the 70 VoIP call simulation run. From here after, the offered traffic exceeds the maximum throughput of the network, which results in packet loss in the sender queues. Using RoHC attenuates this problem by reducing the VoIP throughput. More VoIP traffic is needed to use all the link capacity.

Figure 65 is similar to Figure 48 and Figure 56 although, in this case, the lines are slightly bended in the end due to some packet loss. In Figure 66, it is shown that by reducing the size of VoIP packets, the background UDP flow achieves higher throughput. Table 18 resumes the observed throughputs.

	Uncompressed VoIP Calls (Mbit/s)			Compressed VoIP Calls (Mbit/s)		
	1 call	50 calls	100 calls	1 call	50 calls	100 calls
UDP throughput	13.671	13.641	10.618	13.671	13.670	11.120
VoIP throughput	0.015	0.780	1.509	0.004	0.222	0.437
Overall throughput	13.687	14.421	12.128	13.676	13.892	11.558

Table 18 – Throughput in Simulation Set C

It was shown that using RoHC slightly attenuates congestion, and this effect is also visible in the delay figures.

Figure 67 and Figure 68 show that when using RoHC, the average delay and jitter is slightly lower than when not using it. However, when loss propagation is high RoHC slightly increases the jitter. The delay grows slightly up to with 70 VoIP calls, and reaches the highest delay for 80 VoIP calls. Here, the packet loss ratio increases significantly, making the overall throughput to decrease, and stopping delay from increasing even more.

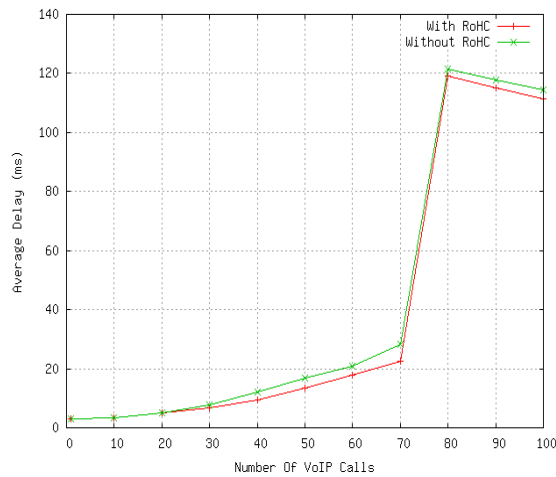


Figure 67 – Average Delay (RoHC + background UDP traffic)

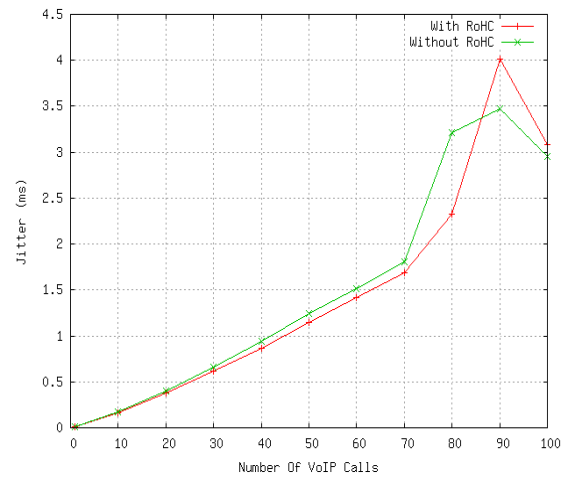


Figure 68 – Jitter (RoHC + background UDP traffic)

	Uncompressed VoIP Calls (ms)			Compressed VoIP Calls (ms)		
	1 call	50 calls	100 calls	1 call	50 calls	100 calls
UDP average delay	2.92	17.27	171.49	2.92	13.12	166.31
VoIP average delay	2.88	16.10	92.80	2.88	13.78	90.08
Total average delay	2.92	16.66	114.27	2.92	13.46	111.29

Table 19 – Average Delay in Simulation Set C

	Uncompressed VoIP Calls (ms)			Compressed VoIP Calls (ms)		
	1 call	50 calls	100 calls	1 call	50 calls	100 calls
UDP jitter	0.009	0.46	0.45	0.008	0.43	0.47
VoIP jitter	0.22	1.97	3.88	0.22	1.80	4.07
Overall jitter	0.013	1.24	2.95	0.012	1.14	3.08

Table 20 – Jitter in Simulation Set C

6.4.3. RoHC U-mode Parameters Experiments

In the previous experiments the U-mode parameters IR_TIMEOUT, FO_TIMEOUT and L_VARS were defined as simulation constants. As demonstrated in the RoHC performance experiments, in the presence of network congestion the U-mode originates loss propagation.

Under network congestion the packet loss ratio is expected to be correlated with the U-mode parameters, since they define the timeouts which make the RoHC Compressor state machine going back to the IR state. In the presence of packet loss the return to the IR State is unavoidable, since the contexts may need to be synchronized. Returning to the IR state decreases the compression ratio, and consequently the RoHC's gain is decreased.

The goal of this Section is to find the values for these parameters which maximize the RoHC's gain when RoHC is transported over an 802.11 link. The experiments were divided into two simulation sets. In the first set the IR_TIMEOUT

and FO_TIMEOUT are varied between 1 and 10 and the L_VARS parameter was fixed to 2 packets. In the second set, the IR_TIMEOUT and FO_TIMEOUT are fixed to 3 and 5, which are the same values used in the previous experiments, and the L_VARS is varied between 1 and 3.

The simulation runs were executed using an UDP flow of 14 Mbit/s as background traffic. The RoHC standard restriction which states that FO_TIMEOUT should always be lower than IR_TIMEOUT was also enforced. It means that if IR_TIMEOUT is 10, possible values of FO_TIMEOUT are the set 1 to 9.

In these experiments, the RoHC's gain was calculated using Equation 1, presented in Section 6.4.2.1.

6.4.3.1. IR_TIMEOUT and FO_TIMEOUT Experiment

As shown in Figure 61, in the RoHC performance experiments with UDP background traffic, the packet loss occurs for more than 50 VoIP calls. Below that, as long as there is no packet loss, and assuming a regular VoIP stream in the sense that fields variations are predictable, the contexts do not get out of synchronization and, therefore, the IR_TIMEOUT and the FO_TIMEOUT should be as high as possible. In this way, the RoHC Decompressor state machine spends more time in the Full Context state, maximizing the compression ratio. The results illustrated in Figure 69 show that with 1-50 VoIP calls, the optimum combination of these parameters is 10 and 9 respectively.

With the increase of packet loss, the RoHC Compressor returns more frequently to the IR state, in order to prevent loss propagation from annulling the RoHC's gain. Figure 70 shows the RoHC's gain obtained for different values of the timeout values when simulating with 60 VoIP calls. In this case, the optimal combination for the IR_TIMEOUT and FO_TIMEOUT is 9 and 4 respectively. In the simulation run with 100 VoIP calls the optimal values for the IR_TIMEOUT and FO_TIMEOUT are 6 and 3, as shown in Figure 71.

The optimal values for these parameters depend on the degree of the network utilization. Assuming the RoHC traffic has regular fields, if RoHC is to be applied to a network with high utilization, the timer values should be low; if the network is underutilized, and packet loss is practically inexistent, then these values should be high.

In IEEE 802.11, the bandwidth available depends on a number of factors including the distance between communications and the number of stations.

A set of values for IEEE 802.11 could be those obtained in the simulation with 60 VoIP calls, which correspond to the network congestion point and present a small but significant percentage of dropped packets. The values for this case are 9 and 4 seconds respectively for the IR_TIMEOUT and FO_TIMEOUT timeouts.

theoretically achieve better compression ratios; but in the scenarios simulated the packet loss has occurred in bursts due to the queues.

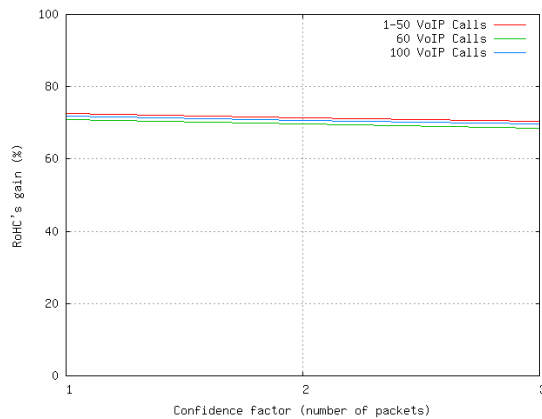


Figure 72 – RoHC's gain

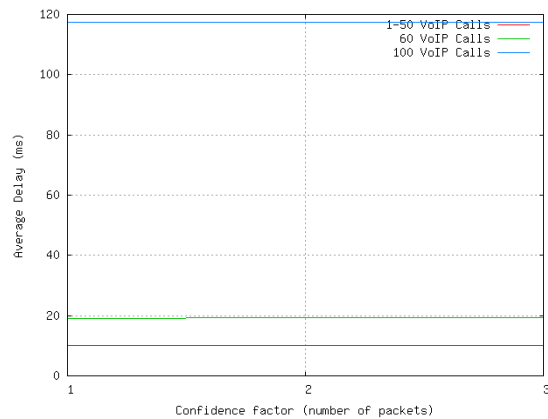


Figure 73 – Average Delay

6.5. Validation

The goal of this section is to validate the RoHC's U-mode simulation model used in the experiences described above. The validation is performed through analysis of a single well-defined test case. The numbers obtained are then compared with the results obtained from running the test-case in the RoHC model developed in NS2.

In order to minimize the influence of the IEEE 802.11 MAC, the test case will be composed by a single unidirectional UDP flow sent between two RoHC-enabled wireless nodes. In this case we can elaborate a simple analysis without the need of modelling the IEEE 802.11 MAC.

6.5.1. Test case parameters

The test case has the following parameters:

- IR_TIMEOUT=5 seconds, FO_TIMEOUT=3 seconds, and L_VAR=2 packets.
- The uncompressed IP packet, not including L2 headers, has a total of 80 bytes: 40 byte IPv6 header, 12 byte UDP header, 8 byte RTP header, and 20 byte G.729A payload.
- When a packet is header compressed in IR State it has 75 bytes.
- When a packet is header compressed in FO State it has 22 bytes.
- When a packet is header compressed in SO State it has 21 bytes.
- The VoIP codec tries to send a packet every 80 ms.
- The VoIP is regular, which means that most fields do not change, and the others change predictably. Thus, once the RoHC Compressor is in a higher compression state, it will only return to a lower compression state through the action of the U-Mode timers.
- Simulation duration lasts 100.81 seconds; this value was set to help the analysis, as explained in Section 6.5.2.
- Other rules include:
- The IR_TIMEOUT is set or reset when the first packet is about to be sent, after changing to IR State, and not when the previous IR_TIMEOUT expires.

- The FO_TIMEOUT is set or reset when the first packet is about to be sent, after changing to FO State, and not when the previous FO_TIMEOUT expires.

6.5.2. Analysis

RoHC's U-mode state machine is cyclic and commanded by the IR_TIMEOUT and the FO_TIMEOUT. A compression cycle starts when the RoHC Compressor returns to IR State. The cycle duration is calculated as the number of packets the RoHC Compressor sends before returning again to the IR State. The IR_TIMEOUT is 5000 ms and the VoIP packet inter-arrival time is 80ms. If we divide 5000ms by 80ms, we get 62.5 packets. This means that the 5000 ms timeout is going to expire after the 63th packet is sent, and before the 64th. In fact, the 64th will be sent in IR State. Assuming that the first packet is sent at time 0, then a new cycle starts every $(64-1) * 80 = 5040$ ms, and consists of 63 packets. Based on this, each cycle has:

- 2 packets sent in the IR State, the first two packets. After that the compressor moves to the FO State.
- a total of 4 packets sent in the FO State. The third and forth packet are sent in FO State, plus a pair for each time the FO_TIMEOUT forces the compressor state machine to move to FO State. Because the IR_TIMEOUT is 5 seconds and the FO_TIMEOUT is 3 seconds, then this only happens once.
- a total of 57 packets sent in the SO State. This is deduced from the number of packets sent in IR and FO States, and the total number of packets sent in each cycle.

The next step is to calculate how many cycles fit in the simulation time and how many bytes are sent. In order to simplify this validation, the simulation time was intentionally set to be a multiple of a cycle plus 10ms to allow the last packet to reach the RoHC decompressor and be accounted. 100.08 seconds are exactly 20 cycles. Table 21 contains the traffic details produced by the VoIP source in 20 cycles.

	Packets/ Cycle	Packets in 20 Cycles	Bytes per packet without RoHC	Total Bytes without RoHC	Bytes per packet with RoHC	Total Bytes with RoHC
IR State	2	$2*20 = 40$	80	3200	75	3000
FO State	4	$4*20 = 80$	80	6400	22	1760
SO State	47	$47*20 = 1140$	80	91200	21	23940
Total	63	1260		100800		28700

Table 21 - Traffic produced by a single VoIP source in 20 Compression Cycles

The total byte reduction is $100800 - 28700 = 72100$ bytes. In these circumstances, RoHC compresses a bandwidth of 7.81 kbit/s to 2.22 kbit/s. This amounts to 71.528% of compression ratio. Exactly the number that results from running the simulation with the same parameters of this analytical model.

6.6. Conclusions

This chapter has presented an evaluation of the RoHC's U-mode performance over IEEE 802.11, a wireless technology which will have an important role in the future 4G wireless access networks.

RoHC's U-mode has been studied for other wireless access technologies such as UMTS. The IEEE 802.11 technology has particular characteristics such as the CSMA/CA nature of the DCF, and a reliable frame transmission service. The impact these features have on the performance of RoHC over 802.11 links constitutes a new view on the study of RoHC. The study was restricted to RoHC's U-mode due to its adequacy for real-time traffic and mobile environments. An important aspect which must be stressed is that the size of IP packets influences the transmission efficiency of IEEE 802.11 networks. It was shown that the maximum efficiency was achieved by using full size 802.11 frames of 2304 bytes. Smaller frame sizes reduce the maximum throughput, because the percentage of L1/L2/L3 overhead over the transported payload increases.

In the RoHC performance experiments, we demonstrated that RoHC saves bandwidth, the most important network resource in wireless networks. Results showed packet compression ratios up to 71.5%. If only the L3 overhead is considered, this value is the RoHC's gain. When all the L1/L2/L3 overheads are considered, the RoHC's becomes more modest, particularly when small packets are predominant in the network. The RoHC performance experiment with full size TCP packets as background packets showed that RoHC's gain is about 30%; with small size TCP packets about 5%.

RoHC performance experiments demonstrated that RoHC is delay and jitter friendly, that it helps reducing delay and jitter. In the presence of congestion and high packet loss, RoHC can slightly worsen the jitter, as a consequence of high loss propagation. By making packet smaller RoHC optimizes the access to the shared medium, which helps lowering average queue size.

The reliability of the IEEE 802.11 frame transmission service reduces packet loss ratio to insignificant numbers. An 802.11 packet loss still may occur due to full queues, a consequence of network congestion. Moreover, in IEEE 802.11 links, the available bandwidth varies with the link conditions, and network congestion can occur suddenly. These aspects imply that RoHC U-mode parameters, IR_TIMEOUT and FO_TIMEOUT timeouts, and L_VAR parameter, should be configured to maximize the RoHC's gain. This was carried out by the RoHC's U-mode experiments. Based on these experiments, we recommended values for the U-mode's parameters that resulted in the highest RoHC gain for a scenario where the offered load was near the network congestion point.

7. Conclusions

7.1. Conclusions

This dissertation started by presenting the state of the art on IP header compression. This included a description of early header compression protocols such as cTCP, IPHC or CRTP. These protocols do not perform well on links having high error ratios or long round trip times. RoHC emerged from the need to standardize a robust IP header compression protocol that performed well also over such links. RoHC is profitable in wireless networks and for traffic characterized by small payloads and large relative headers. These are characteristics of the 4G scenarios.

Chapter 3 described how header compression capabilities were integrated in 3GPP cellular networks. Header compression was first introduced in GPRS. At first, only cTCP and IPHC were supported, but in the 3GPP Release 6 RoHC support was added. These mechanisms were also integrated in the UMTS specification. The details of RoHC integration in UMTS were relevant for the RoHC solution presented in this dissertation. There is not, currently, a standard solution for transporting and negotiating RoHC over IEEE 802 networks, so the 3GPP solution was used as a case study.

The architecture of the QoS-Abstraction Layer (QoS-AL) was presented in Chapter 4. The QoS-AL is a framework developed in the IST DAIDALOS project, for performing QoS reservations over L2 heterogeneous access networks. It provides an abstract QoS interface which receives abstract QoS parameters. The QoS-AL is generic and can be integrated with various L3 QoS models (such as RSVP/Intserv or Diffserv) to provide E2E QoS guarantees. The QoS reservations, also known as QoS Connections, are set up between Access Routers (AR) and Mobile Terminals (MT). They are initiated by the AR which sends an AL-CN-activate-req to the MAC address of the MT. The QoS reservation is committed by the AL-CN-activate-resp traveling in the return path. Because the QoS-AL nodes are IEEE 802.1D Learning Bridge compatible nodes, there is no need of previous knowledge about the access network topology.

In Chapter 5 the AL-RoHC solution was presented. First, an encapsulation solution for the transportation of RoHC packets was provided. This solution uses DIX frames and includes a L2.5 header containing a *Length* field to describe the *Ethernet Minimum Frame Size* issue. In an Ethernet link, a frame must have a minimum size of 64 bytes, or a 46 byte payload. If an IP packet is too small, padding is introduced. Usually the L2 equipment snoop the IP header's IP Total Length field, to evaluate if padding is present. Because RoHC suppresses this field the 802.11 Access Points are not able to detect and remove padding. This penalty may influence negatively the RoHC profit. By looking at the *Length* field in the L2.5 header, this problem was solved. The encapsulation solution was integrated in the QoS-AL. The negotiation mechanism was piggybacked on the QoS-AL reservation messages, what has limited the setup time to

one round-trip-time, which is adequate for mobility scenarios. The addition of RoHC capabilities to the QoS-AL enables the later to use efficiently the resources of 4G networks.

Chapter 6 presented a study of the RoHC's U-mode performance over IEEE 802.11, a wireless technology which will play an important role in the future 4G access networks. The IEEE 802.11 technology has characteristics such as the CSMA/CA and a reliable transmission service. The impact of these features on the RoHC's performance was evaluated. The study was limited to the RoHC's U-mode since it seems more adequate for real-time traffic and mobile environments than the other modes. RoHC studies showed that applying RoHC to VoIP traffic saves bandwidth but, if the L1/L2 overheads is considered, RoHC's gain can be modest. When large frames are used, the RoHC's gain can reach 30%; when small frames are used, the gain is reduced to 5%. RoHC was found to be delay and jitter friendly. By reducing the packet size, RoHC helps reducing the delay and the jitter. However, in the presence of congestion and high packet loss, RoHC can slightly worsen the jitter, as a consequence of high loss propagation. The particular characteristics of the 802.11 also motivated the study of the optimal values for the RoHC's U-mode parameters, the IR_TIMEOUT and FO_TIMEOUT timeouts, and the L_VAR parameter. Based on experiments we recommended values for the U-mode's parameters.

7.2. Original Contributions

The original contributions of the work associated with this dissertation are:

1. **An encapsulation solution for transporting RoHC packets over IEEE 802 Networks.** This proposal avoided the use of LLC/SNAP headers, but added a L2.5 header containing a *Length* field. This is required to overcome the *Ethernet Minimum Frame Size* problem.
2. **A RoHC channel negotiation mechanism over IEEE 802 Networks.** This negotiation mechanism complements the proposed RoHC encapsulation solution.
3. **An extension which integrates RoHC capabilities in the DAIDALOS QoS-AL framework.** The solution specified integrates the proposed encapsulation and RoHC channel negotiation mechanisms into the QoS-AL. Two new AL protocol PDUs were added: AL-RoHC-REQUEST-PARAMETERS, and AL-RoHC-PARAMETERS.
4. **A performance study of RoHC's U-mode over IEEE 802.11.** The IEEE 802.11 technology has characteristics such as the CSMA/CA and the reliable transmission service. The impact these features have on the RoHC's performance constitutes a new view on the study of RoHC. The study addresses mainly VoIP traffic. The RoHC U-mode parameters, such as the IR_TIMEOUT and FO_TIMEOUT timeouts, and the L_VAR parameter, were also studied and values were proposed for them.

7.3. Results

A simulation model has been developed which implements RoHC's U-mode in NS2. This simulation model adds new functionality to the NS2 simulator: RoHC header compression; a flow manager capable of measuring network throughput, delay, jitter, and packet loss; and a new traffic type, to model the G.729 codec. Most of the source

code was written in C++ and oTCL. The simulation batch scripts were written in *bash*, traces stored as XML, and the post log processing was made in Perl and *awk*.

7.4. Future Work

Future work based on the contributions produced during this dissertation includes:

RoHC implementation: an implementation of RoHC as a Linux Kernel module is currently underway. In a first phase this implementation will address only the RoHC U-mode and IPv4. Then it may be complemented with the other RoHC modes and with IPv6 support.

AL-RoHC implementation: once the RoHC prototype is complete, the AL-RoHC can be implemented and integrated in the QoS-AL.

Dynamic QoS reservation adjustment: based on the compression feedback provided by AL-RoHC, the QoS-AL could adjust the resources reserved for each flow. This would provide new means to the resource management modules.

More validation: the simulation model was tested and validated based on the simulated scenarios. Performing more simulations, using perhaps different scenarios, would certainly contribute to the fine-tuning of the simulation model and to a better validation.

MPLS: this dissertation proposed an encapsulation solution for transporting RoHC packets over 802 Networks. A similar solution could be devised for MPLS. Although MPLS is a technology mainly used in large core networks, it has been conquering some space in the access networks. Core or access operators that use MPLS could make use of RoHC to increase their profits.

Cross-layer queue drop notifications: Even on networks where packet loss is rare there is a need for the U-mode's Compressor to return periodically to a no compression state. This is caused by packet loss occurring in the sender queue, due to queue overflow. This type of loss is different from the packet loss resulting from regular transmission errors, in the sense that the Compressor can be aware that it is dropping packets. Thus the RoHC's U-mode performance could be improved if queue drops were known. Having this information, the Compressor could go immediately to the IR state. This would eliminate the loss propagation almost completely.

Annex A - QoS-AL Service Primitives

A.1. Data types used in the Primitives

Addr	Can be either a L3Addr or a L2Addr;
L3Addr	An IPv6 address;
L2Addr	An IEEE 802 MAC address (e.g. 802.11 AP) or a generic link-layer address (e.g. UMTS IMEI);
Result	Can be either ACCEPT or REJECT;
CnxID	An unsigned 20-bit integer number that univocally identifies a QoS-AL connection;
TSpec	Contains the same parameters of IntServ/RSVP's TSpec [49]: • Peak Bitrate (p) • Average Bitrate (r) • Maximum Burst Size (b) • Maximum Transmission Unit (M) • Minimum Policed Unit (m)
RSpec	Contains the set of service attributes that the L2 promises to deliver, as long as the application does not violate the TSpec. In QoS-AL, the RSpec includes a Class Identifier and Reserved Bitrate. Other parameters, such as BER and delay, are determined implicitly based on the Class Identifier;
Bitrate	Transmitted information per time unit, in bits per second;
ContextInfo	A block of binary data that contains information not used by the AL protocol, but by upper layers. This information is piggybacked in AL PDU's from the AR to the MNs, and from there passed to the upper layers;
BER	The Bit Error Ratio is the ratio between erroneous bits and the total number of bits in a frame/packet, usually represented as a single precision floating-point number. A negative value means that BER information is not available;
Priority	An integer number representing the priority of a packet, which ranges from -128 (lowest priority) to +127 (highest priority);
Boolean	A boolean value (true/false).

A.2. QoS Reservation Primitives

A.2.1 AL-CNX-ACTIVATE-REQ Primitive		
Description: Used by the QoS Manager to request the activation of a new QoS connection.		
Direction: Downcall		
Parameters:		
Type	Name	Description
Addr	dest_addr	The destination address of the MT for which we wish to make a reservation. This is a L3 address, except when the reservation is for multicast traffic, in which case it is a L2 address.

TSpec	tx_tspeg	Traffic Specification for the transmitted flow
RSpec	tx_rspeg	Reservation Specification for the transmitted flow
TSpec	rx_tspeg	Traffic Specification for the received flow
RSpec	rx_rspeg	Reservation Specification for the received flow
Priority	retention_prio	Retention priority. This parameter is similar to RSVP-TE's Hold Priority [53]. Reservations with lower priority are selected for QoS degradation when resources become scarce.
Boolean	handoff	This parameter is set to True if mobility scenarios are to be supported.
ContextInfo	ctx_info	Context information attached to the QoS Connection.

A.2.2 AL-CNX-ACTIVATE-RESP Primitive

Description: If a QoS reservation is successfully activated, this primitive is returned to the caller, usually the QoS Manager in the AR.

Direction: Upcall

Parameters:

Type	Name	Description
CnxID	cnx_id	The identifier of the QoS reservation that was created.
Result	result	Indicates if the reservation was successful or not.

A.2.3 AL-CNX-MODIFY-REQ Primitive

Description: Used to request modification of the QoS parameters of an existing QoS Connection.

Direction: Downcall

Parameters:

Type	Name	Description
CnxID	cnx_id	The identifier of the QoS reservation that was created.
TSpec	tx_tspeg	New Traffic Specification for the transmitted flow
RSpec	tx_rspeg	New Reservation Specification for the transmitted flow
TSpec	rx_tspeg	New Traffic Specification for the received flow
RSpec	rx_rspeg	New Reservation Specification for the received flow

A.2.4 AL-CNX-MODIFY-RESP Primitive

Description: If the parameters of a QoS reservation are successfully modified, this primitive is returned to the caller.

Direction: Upcall

Parameters:

Type	Name	Description
CnxID	cnx_id	The identifier of the QoS reservation for which a modification was requested.
Result	result	Indicates if the modification was successful or not.

A.2.5 AL-CNX-DEACTIVATE Primitive

Description: Can be called by either the QoS Client (at the MT) or by the QoS Manager (at the AR) to request the deactivation of an existing QoS Connection.

Direction: Downcall/Upcall

Parameters:

Type	Name	Description
------	------	-------------

CnxID	cnx_id	The identifier of the QoS reservation for which the deactivation was requested.
-------	--------	---

A.3. Resource Querying Primitives

A.3.1 AL-RESOURCE-QUERY Primitive		
Description: This primitive can be used to request information about the available resources along a given L2 network path.		
Direction: Downcall		
Parameters:		
Type	Name	Description
DestAddr	Target	Address of the destination endpoint of the L2 path. The target can be MT, an AP, or a broadcast address.

A.3.2 AL-RESOURCE-INDICATION Primitive		
Description: Reports information about the available resources in a L2 path, as a result of a previously AL-RESOURCE-QUERY request.		
Direction: Downcall		
Parameters:		
Type	Name	Description
L2Addr	Target	Address of the destination endpoint of the L2 path.
Bitrate	Bandwidth	Estimation of available bandwidth

A.3.3 AL-RESOURCE-DEGRADATION Primitive		
Description: Notification of degradation of overall resources.		
Direction: Upcall		
Parameters:		
Type	Name	Description
List of Tuple [L2Addr, Bitrate]	Resources	List of bridges/AP address/bandwidth pairs.

A.4. QoS Notification Primitives

A.4.1 AL-CNX-INDICATION Primitive		
Description: Provides notification of a new connection activated or a QoS change on a connection.		
Direction: Upcall		
Parameters:		
Type	Name	Description
CnxID	cnx_id	Identifies the connection
TSpec	tx_tspec	New Traffic Specification for the transmitted flow
RSpec	tx_rspec	New Reservation Specification for the transmitted flow
TSpec	rx_tspec	New Traffic Specification for the received flow
RSpec	rx_rspec	New Reservation Specification for the received flow
BER	ber	Optional BER information
ContextInfo	ctx_info	Context information attached to the QoS Connection.

A.5. Mobility Primitives

A.5.1 AL-HANDOVER-EXECUTE Primitive		
Description:	This primitive should be sent right after the Handover has occurred; The AR should proceed with any pending connection activation(s).	
Direction:	Downcall (MT side only)	
Parameters:		
Type	Name	Description
Addr	dst	Address of the new AR the MT has just connected to.

A.6. Multicast Primitives

A.6.1 AL-MULTICAST-JOIN Primitive		
Description:	Indicates that a terminal is joining a multicast session.	
Direction:	Downcall	
Parameters:		
Type	Name	Description
Addr	mt_addr	Address of the MT that is joining the multicast session.
CnxID	session	Connection identifier of the multicast session.

A.6.2 AL-MULTICAST-LEAVE Primitive		
Description:	Indicates that a terminal is leaving a multicast session.	
Direction:	Downcall	
Parameters:		
Type	Name	Description
Addr	Dst	Address of the MT that is leaving the multicast session.
CnxID	session	Connection identifier of the multicast session.

Annex B - QoS-AL Driver Primitives

B.1. Driver Registration Primitives

B.1.1 AL-DRIVER-REGISTER Primitive		
Description: Used by an AL Driver to register itself in the main QoS-AL process.		
Direction: Upcall		
Parameters:		
Type	Name	Description
String	netdevice	Network device identifier that the driver wishes to handle.
Bitrate	bwmax	Maximum bandwidth (theoretical value) available on the interface.

B.2. QoS Reservation Primitives

B.2.1 AL-CNX-ACTIVATE-REQ Primitive		
Description: Used by the QoS Manager to request the activation of a new QoS connection. Called by the AI's AL-CNX-ACTIVATE-REQ primitive.		
Direction: Downcall		
Parameters:		
Type	Name	Description
L2Addr	dest_addr	The L2 address of the MT.
TSpec	tx_tspec	Traffic Specification for the transmitted flow
CnxID	cnxid	Connection Identifier. The driver must configure the L2 to map packets to the connection being activated by comparing the IPv6 Flow Label field with CnxID's.
TSpec	tx_tspec	Traffic Specification for the transmitted flow
RSpec	tx_rspec	Reservation Specification for the transmitted flow
TSpec	rx_tspec	Traffic Specification for the received flow
RSpec	rx_rspec	Reservation Specification for the received flow
Priority	retention_prio	Retention priority.

B.2.2 AL-CNX-ACTIVATE-RESP Primitive		
Description: Result of QoS Connection activation request. Called by the AI's AL-CNX-ACTIVATE-RESP primitive.		
Direction: Upcall		
Parameters:		
Type	Name	Description
CnxID	cnx_id	The identifier of the QoS reservation that was created.
Result	result	Indicates if the reservation was successful or not.

B.2.3 AL-CNX-MODIFY-REQ Primitive		
Description:	Used to request modification of the QoS parameters of an existing QoS Connection. Called by the AI's AL-CNX-MODIFY-REQ primitive.	
Direction:	Downcall	
Parameters:		
Type	Name	Description
CnxID	cnx_id	The identifier of the QoS reservation that was created.
TSpec	tx_tspeg	New Traffic Specification for the transmitted flow
RSpec	tx_rspec	New Reservation Specification for the transmitted flow
TSpec	rx_tspeg	New Traffic Specification for the received flow
RSpec	rx_rspec	New Reservation Specification for the received flow

B.2.4 AL-CNX-MODIFY-RESP Primitive		
Description:	If the parameters of a QoS reservation are successfully modified, this primitive is returned to the caller. Called by the AI's AL-CNX-MODIFY-RESP primitive.	
Direction:	Upcall	
Parameters:		
Type	Name	Description
CnxID	cnx_id	The identifier of the QoS reservation for which a modification was requested.
Result	result	Indicates if the modification was successful or not.

B.2.5 AL-CNX-DEACTIVATE Primitive		
Description:	Can be called by either the QoS Client (at the MT) or by the QoS Manager (at the AR) to request the deactivation of an existing QoS Connection. Called by the AI's AL-CNX-DEACTIVATE primitive.	
Direction:	Downcall/Upcall	
Parameters:		
Type	Name	Description
CnxID	cnx_id	The identifier of the QoS reservation for which the deactivation was requested.

B.3. Resource Querying Primitives

B.3.1 AL-RESOURCE-QUERY Primitive		
Description:	This primitive can be used to request information about the available resources in the network interface controlled by the AL Driver. Called by the AI's AL-RESOURCE-QUERY primitive.	
Direction:	Downcall	
Parameters:		
Type	Name	Description
-	-	-

B.3.2 AL-RESOURCE-INDICATION Primitive		
Description:	Reports information about the available resources in the network interface. The information provided by this primitive is used by the QoS-AL process to generate AL-RESOURCE-DEGRATION when the available bandwidth	

Direction: drops bellow 25% of the maximum theoretical bandwidth. Parameters: Upcall		
Type	Name	Description
Bitrate	Bandwidth	Estimation of available bandwidth in the network interface.

B.4. QoS Notification Primitives

B.4.1 AL-CNX-INDICATION Primitive		
Description: Provides notification of a QoS change on a connection.		
Direction: Upcall Parameters:		
Type	Name	Description
CnxID	cnx_id	Identifies the connection
TSpec	tx_tspeg	New Traffic Specification for the transmitted flow
RSpec	tx_rspec	New Reservation Specification for the transmitted flow
TSpec	rx_tspeg	New Traffic Specification for the received flow
RSpec	rx_rspec	New Reservation Specification for the received flow
BER	ber	Optional BER information

B.5. Multicast Primitives

B.5.1 AL-MULTICAST-JOIN Primitive		
Description: Indicates that a terminal is joining a multicast session.		
Direction: Downcall Parameters:		
Type	Name	Description
L2Addr	mt_addr	Address of the MT that is joining the multicast session.
CnxID	session	Connection identifier of the multicast session.

B.5.2 AL-MULTICAST-LEAVE Primitive		
Description: Indicates that a terminal is leaving a multicast session.		
Direction: Downcall Parameters:		
Type	Name	Description
L2Addr	Dst	Address of the MT that is leaving the multicast session.
CnxID	session	Connection identifier of the multicast session.

Annex C - AL-RoHC Details

C.1. AL-RoHC PDUs

C1.1 AL-RoHC-REQUEST-PARAMETERS Primitive		
Description: Initiates a RoHC channel negotiation.		
Direction: Downcall		
Parameters:		
Type	Name	Description
-	-	-

C1.2 AL-RoHC- PARAMETERS Primitive		
Description: RoHC channel parameters.		
Direction: Upcall		
Parameters:		
Type	Name	Description
Result	result	Indicates if the inquired node supports RoHC. The subsequent fields are only present if RoHC is supported.
String	ChannelName	This field represents the administrative name of this RoHC channel and is intended for management purposes. The link name may be reused as RoHC channel.
Number16	MAX_CID	The maximum value of a context identifier. The default value is 15. It must be at least 0 (a single context) and at most 16383. This value is constrained by LARGE_CIDS parameter.
Boolean	LARGE_CIDS	If false, the short CID representation (0 or 1 bytes) is used (maximum of 16 CIDs). If true, the high CID representation (1 or 2 bytes) is used (maximum of 16383 CIDs).
List of Number16	PROFILES	The set of RoHC profiles supported by the sender. The first octet contains the main Profile ID, and the second octet contains the supported variant of that profile.
String	FEEDBACK_FOR	An optional reference to a channel in the reverse direction. If provided, this parameter indicates which channel should be used to send feedback.
Number16	MRRU	Indicates the maximum reconstructed reception unit, or the size of the largest reconstructed unit in octets

Number8	MAX_HEADER	that the decompressor is allowed to reassemble from segments. If the MRRU is set to 0, RoHC segmentation is disabled. The default suggested value is 0. The largest header size in octets that may be compressed. This value should be large enough to enable the compression of headers in all supported profiles. The suggested default value is 168 octets.
---------	------------	---

C.2. AL-RoHC Interfaces

Interfaces at RoHC Channel endpoints

AL_RoHC_RIM_QOS-AL_AR and AL_RoHC_RIM_QOS-AL_MT

C2.1 AL-RoHC-GET-LOCAL-RoHC-PARAMETERS

Description:	Can be used by the QoS-AL to request the local node's RoHC parameters.
Modules:	AL_RoHC_IMOD_AR & AL_RoHC_IMOD_MT
Input	-
Parameters:	
Returns:	Returns NULL if this node does not support RoHC, or if it is administratively disabled. If not, it returns an object containing the MAX_CID, LARGE_CIDS, PROFILES tuple, MRRU, and MAX_HEADER. These fields have the same meaning as in C1.2.

C2.2 AL-RoHC-CONFIGURE-LOCAL-RoHC-PARAMETERS

Description:	Can be used by the QoS-AL to configure the local node's RoHC parameters.
Modules:	AL_RoHC_IMOD_AR & AL_RoHC_IMOD_MT
Input	MAX_CID, LARGE_CIDS, PROFILES tuples, MRRU and
Parameters:	MAX_HEADER. See C1.2 for details on these parameters.
Returns:	Returns True if succeeds. Returns False if this node does not support RoHC, or if it is administratively disabled.

C2.3 AL-RoHC-STORE-HOST-PARAMETERS

Description:	This primitive is called by the QoS-AL to request the storage of the RoHC parameters of a remote host (i.e. a MT).
Modules:	AL_RoHC_IMOD_AR & AL_RoHC_IMOD_MT
Input	L2Address of the Remote Host, MAX_CID, LARGE_CIDS, PROFILES
Parameters:	tuples, MRRU and MAX_HEADER.
Returns:	Returns -1 if the operation fails (e.g. not enough memory).

C2.4 AL-RoHC-HAVE-HOST-PARAMETERS

Description:	This primitive is called by the QoS-AL to question the RM through RIM, if the parameters for a given remote host are already known. If not, it may be followed by a call to AL-RoHC-STORE-HOST-PARAMETERS.
Modules:	AL_RoHC_IMOD_AR & AL_RoHC_IMOD_MT
Input	L2Address of the Remote Host.

Parameters:

Returns: Returns **true** if the RoHC parameters for L2Address are known and **false** otherwise.

C2.5 AL-RoHC-ENABLE-ROHC-FOR-CNX

Description: This primitive is called by the QoS-AL to request the activation of RoHC for a given QoS-AL connection.

Modules: AL_RoHC_IMOD_AR & AL_RoHC_IMOD_MT

Input CnxID

Parameters:

Returns: Returns **true** if successful, **false** otherwise.

C2.6 AL-RoHC-DISABLE-ROHC-FOR-CNX

Description: This primitive is called by the QoS-AL to request the deactivation of RoHC for a given QoS-AL connection.

Modules: AL_RoHC_IMOD_AR & AL_RoHC_IMOD_MT

Input CnxID

Parameters:

Returns: Returns **true** if successful, **false** otherwise.

Interfaces at Intermediary RoHC Channel Nodes

AL_RoHC_RIM_QOS-AL_BR

C2.7 AL-RoHC-IS-ROHC-AVAILABLE

Description: This primitive is called by the QoS-AL to check if a RoHC Module, in its Intermediary Bridge Version, is present. If not, RoHC usage will not be possible for the flows passing in this intermediary bridge.

Modules: AL_RoHC_IMOD_802.16_BS, AL_RoHC_IMOD_802.16_SS, AL_RoHC_IMOD_802.15_AP and AL_RoHC_IMOD_802.11_AP

Input -

Parameters:

Returns: Returns **true** if the RoHC Module is present, **false** otherwise.

C2.8 AL-RoHC-SET-ROHC-AVAILABLE

Description: This primitive may be called to configure the RIM module, indicating if the RoHC Module, in its Intermediary Bridge Version, is present.

Modules: AL_RoHC_IMOD_802.16_BS, AL_RoHC_IMOD_802.16_SS, AL_RoHC_IMOD_802.15_AP and AL_RoHC_IMOD_802.11_AP

Input Boolean value

Parameters:

Returns: Returns **true** if successful, **false** otherwise.

Annex D - IEEE 802.11

The IEEE 802.11 is a set of standards for Wireless LAN developed by the IEEE LAN/MAN Standards Committee (IEEE 802) [62]. There are three radio specifications defined as amendments to the original standard: the 802.11a [63], the 802.11b [64] and the 802.11g [65]. The 802.11b and the 802.11g use the 2.4 GHz band and the 802.11a uses the 5 GHz band. The differences between these amendments are mainly at the physical layer; the 802.11b has a maximum throughput of 11Mbit/s; the 802.11a and the 802.11g have bitrates up to 56 Mbit/s. At MAC level, the 802.11 use the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA).

The 802.11 MAC sub layer includes two medium access mechanisms: the Distributed Coordination Function (DCF) and the Point Coordination Function (PCF). The PCF is optional and can only be used when the WLAN is configured in infrastructure mode; the former is mandatory and can be used in both infrastructure and ad-hoc mode. When using the DCF, all stations apply a distributed algorithm to determine which station accesses the medium next. When PCF is used, a central node, usually the Access Point, schedules the medium access time. When both mechanisms are used, each super-frame is divided into two parts: a contention-free period (CFP), where the PCF is in charge; and a contention period (CP) where the DCF determines which gain access to the medium. The PCF is usually not available in 802.11 equipment.

In DCF each station “senses” the medium before trying to send frames. If the medium is busy then the station uses a random backoff procedure. When a station gets access to the medium it transmits the frames it has since queued for delivery. The transmitting station always waits for a positive acknowledgement (ACK) for each frame it sends. If an ACK is not received shortly by the station, then it reschedules a retransmission of the frame. Each frame can be retransmitted up to 7 times before being discarded permanently. This scheme reduces significantly packet losses due to transmission errors.

List of Acronyms

3G	3rd Generation of mobile communications	CP	Contention Period
3GPP	3rd Generation Partnership Program	CQoS	Core QoS Broker
4G	4th generation of mobile communications	CRC	Cycle Redundancy Code
ACK	Acknowledgments	cRTP	Compressed RTP
AI	Abstraction Interface	CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
AL	Abstraction Layer	cTCP	Compressed TCP
AL-RoHC	Abstraction Layer RoHC	CTS	Clear To Send
AN	Access Network	DA	Destination Address
AP	Access Point	DAIDALOS	Designing Advanced network Interfaces for the Delivery and Administration of Location independent, Optimised personal Services
AQoS	Access QoS Broker	DCF	Distributed Coordination Function
AR	Access Router	DI	Driver Interface
ATM	Asynchronous Transfer Mode	DiffServ	Differential Services
AuC	Authentication Center	DIX	DEC IBM Xerox (encapsulation)
BG	Border Gateway	DSCP	DiffServ Code Point
BSS	Base Station Subsystem	DO	Decompressed Output
BSSGP	BSS GPRS Application Protocol	E2E	End-to-end
BTS	Base Transmission Station	ECRTP	Enhanced Compressed RTP
BER	Bit Error Ratio	EDGE	Enhanced Data rates for GSM Evolution
CDMA	Code Division Multiple Access	EIR	Equipment Identity Register
CFP	Contention-free Period	ETSI	European Telecommunications Standards Institute
CG	Charging Gateway	FI	Feedback Input
CI	Compressed Input	FO	First Order state; Feedback Output
CID	Context ID	FR	Frame Relay
CN	Core Network	FTP	File Transfer Protocol
CnxID	Connection ID		
CO	Compressed Output		
COPS	Common Open Policy Service		

GERAN GPRS EDGE Radio Access Network	NSAPI Network Service Access Point Identifiers
GGSN Gateway GPRS Support Node	NS2 Network Simulator 2
GMSC Gateway Mobile Switching Center	NSS Network Switching Subsystem
GPRS General Packet Radio Service	PCF Point Coordination Function
GSM General System for Mobile communication	PDA Personal Digital Assistant
GSMRF GSM Radio Frequency	PDCP Packet Data Convergence Protocol
GSN GPRS Support Node	PDP Packet Data Protocol
GTP GPRS Tunnelling Protocol	PDN Packet Data Network
HCP Header Compression Profile	PDU Protocol Data Unit
HDLCL High-level Data Link Control	PI Piggybacked Input
HLR Home Location Register	PID Protocol ID
IANA Internet Assigned Number Authority	PLCP Physical Layer Convergence Procedure
IEEE Institute of Electrical & Electronics Engineers	PLMN Public Land Mobile Network
IETF Internet Engineering Task Force	PO Piggybacked Output
IMT Identifiers Map Table	PPP Point-to-point Protocol
IntServ Integrated Services	PSTN Plain Simple Telephone Network
IP Internet Protocol	O-MODE Bidirectional Optimistic Mode
IPHC IP Header Compression	QoS Quality of Service
IR Initialization and Refresh state	QoS-AL QoS Abstraction Layer
ISDN Integrated Services Digital Network	RB Radio Bearer
ITU-T International Telecommunication Union Telecommunication Standardization Sector	R-MODE Bidirectional Reliable Mode
L2 Layer 2	RAN Radio Access Network
L3 Layer 3	RCT RoHC Channels Table
LAPDm Link Access Protocol on the Dm Channel	RIM RoHC Interface Module
LLC Logic Link Control	RF RoHC Flag
MAC Medium Access Control	RLC Radio Link Control
MPLS Multi Protocol Label Switching	RM RoHC Module
MRRU Maximum Reconstructed Reception Unit	RNC Radio Network Controller
MS Mobile Station	RoHC RObust Header Compression
MSC Mobile Switching Center	RSPEC Reservation Specification
MT Mobile Terminal	RSVP ReSerVation Protocol
MTU Maximum Transmission Unit	RSVP-TE RSVP with Traffic Engineering
NACK Negative Acknowledgment	RTP Real-time Transport Protocol
	RTS Request To Send
	SA Source Address
	SAP Service Access Point
	SDU Service Data Unit

SGSN Serving GPRS Support Node
SO Second Order state
SNAP Standard Network Access Protocol
SNDCP SubNetwork Dependant Convergence Protocol
TCP Transmission Control Protocol
TDD Time Division Duplex
TDMA Time Division Multiple Access
TSPEC Traffic Specification
TTL Time to live
U-MODE Unidirectional Mode
UDP User Datagram Protocol
UE User Equipment
UI Uncompressed Input
UMTS Universal Mobile Telecommunications System
UTRAN UMTS Terrestrial Radio Access Network
UWB Ultra Wide Band
VAD Voice Activity Detection
VLR Visited Location Register
VoIP Voice over IP
WAN Wide Area Network
WMAN Wireless Metropolitan Area Network
WLAN Wireless Local Area Network

Index

3

3GPP 1, 4, 21, 26, 29, 77, 95, 102

4

4G 1, 3, 1, 2, 3, 19, 31, 33, 46, 54, 55, 76, 77, 78, 95, 103, 104

8

802 Networks 3, 43, 78, 79, 103
 802.11 3, 1, 2, 3, 4, 31, 38, 42, 45, 48, 55, 56, 57, 58, 59, 61, 63, 65, 68, 71, 72, 74, 76, 77, 78, 81, 91, 93, 105
 802.15 3, 38, 42, 91
 802.16 1, 2, 3, 31, 42, 91
 802.3 2, 3, 42, 43, 44, 45, 48, 54

A

AL-RoHC 4, 41, 42, 43, 47, 48, 49, 50, 51, 54, 77, 78, 79, 89, 90, 91, 95
 AP 33, 34, 38, 39, 40, 45, 48, 49, 56, 59, 81, 83, 91, 95
 AR 2, 32, 33, 34, 35, 36, 37, 38, 39, 40, 43, 45, 46, 49, 50, 51, 52, 53, 56, 59, 60, 77, 81, 82, 84, 86, 90, 91, 95

B

BER 3, 81, 83, 87, 95
 Bidirectional Optimistic 15, 17, 96, *See* O-mode
 Bidirectional Reliable 15, 18, 96, 102, *See* R-mode
 Bluetooth 31
 Bluetooth 1
 bridge 33, 37, 49, 91

C

CID 10, 11, 13, 14, 15, 19, 23, 25, 26, 28, 29, 41, 45, 46, 50, 54, 89, 90, 95
 CnxID 34, 35, 36, 37, 38, 46, 49, 50, 52, 54, 81, 82, 83, 84, 85, 86, 87, 91, 95
 Codec 57
 Compressed RTP 7, 8, 95, 101, *See* cRTP
 compression ratio 12, 14, 16, 45, 52, 62, 63, 65, 71, 72, 74, 75, 76
 Compressor 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 16, 17, 18, 19, 42, 52, 58, 71, 72, 74, 75, 79

Context ID 10, 95, *See* Context ID
 cRTP 7, 8, 95
 cTCP 6, 7, 19, 24, 25, 29, 77, 95

D

DAIDALOS 2, 3, 31, 32, 35, 40, 77, 78, 95, 103
 damage propagation 11, 18
 DCF 55, 65, 76, 93, 95
 Decompressor 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 42, 50, 52, 63, 65, 72
 delay 35, 55, 56, 61, 62, 63, 67, 70, 71, 73, 76, 78, 81
 Distributed Coordination Function 93, 95
 DIX 3, 43, 44, 45, 46, 48, 54, 77, 95

E

ECRTP 8, 19, 95
 encapsulation 3
 Enhanced Compressed RTP 8, 95, 101, *See* ECRTP
 error propagation 11
 Ethernet 3, 15, 24, 43, 44, 45, 46, 48, 51, 56, 59, 77, 78, 103

F

feedback 7, 9, 10, 11, 13, 14, 15, 16, 17, 18, 19, 26, 42, 52, 54, 55, 79, 89
First Order 13, 95, *See* FO
 FO 10, 13, 14, 16, 58, 71, 72, 73, 74, 75, 76, 78, 95
Full Context 14, 17, 72
 full header 5, 6, 7, 8, 10, 14, 25, 28

G

GPRS 21, 22, 23, 24, 25, 26, 27, 29, 77, 95, 96, 97, 102
 GSM 21, 22, 25, 26, 29, 95, 96, 102

H

HCP 11, 12, 13, 14, 15, 19, 96, *See* Header Compression Profile
 Header Compression Profile 11, 19, 96
 heterogeneous wireless networks 3, 1
 High-CID 45, 46

I

Identifiers Map Table 50, 96
 IETF 11, 13, 41, 43, 96, 101, 102, 103, 104
 IMT 50, 51, 54, 96, 103
Initialization and Refresh 13, 96, *See* IR
 IP Header 1, 6, 96, 101
 IP Header Compression 6, 96, 101
 IPHC 6, 7, 19, 24, 25, 26, 28, 29, 77, 96, *See* IP Header Compression
 IR 13, 14, 15, 16, 58, 71, 72, 73, 74, 75, 76, 78, 79, 96
 IR-DYN 15

J

jitter 55, 61, 63, 67, 70, 71, 76, 78

L

LAN 57, 93
 loss propagation 11, 17, 58, 63, 68, 70, 71, 72, 76, 78, 79

M

MAC 23, 37, 38, 39, 40, 43, 44, 50, 56, 61, 62, 63, 74, 77, 81, 93, 96, 102
 MT 2, 32, 33, 34, 35, 36, 37, 40, 45, 46, 49, 50, 51, 52, 53, 56, 59, 60, 77, 81, 82, 83, 84, 85, 86, 87, 90, 91, 96

N

NACK 7, 15, 17, 96
 Network Simulator 2 4, 55, 96, 103
No Context 14
 NS2 4, 55, 56, 74, 78, 96, 105, *See* Network Simulator 2

O

O-mode 15, 17, 18, 19, 42, 55

P

packet loss 5, 7, 8, 16, 28, 35, 42, 58, 62, 63, 64, 65, 68, 70, 71, 72, 73, 76, 78, 79, 93
 padding 3, 15, 41, 44, 45, 46, 51, 52, 54, 77
 PCF 93, 96
 Point Coordination Function 93, 96

Q

QoS 3, 1, 2, 3, 4, 23, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 46, 47, 48, 49, 50, 51, 52, 53, 54, 77, 78, 79, 81, 82, 83, 85, 86, 87, 90, 91, 95, 96, 103
 QoS Abstraction Layer 3, 2, 4, 31, 33, 36, 96, 103
 QoS reservation 3, 2, 31, 32, 33, 34, 35, 36, 37, 38, 40, 42, 47, 48, 49, 50, 51, 52, 77, 79, 82, 83, 85, 86

QoS-AL 3, 2, 3, 31, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 46, 47, 48, 49, 50, 51, 52, 54, 77, 78, 79, 81, 85, 86, 90, 91, 96

R

RCT 50, 51, 54, 96
 RF 47, 49, 51, 54, 96
 RIM 49, 50, 51, 90, 91, 96
 RM 49, 50, 51, 52, 90, 96
 R-mode 11, 15, 18, 19, 42, 55, 102
 Robust Header Compression 1, 3, 1, 5, 8, 101, 102, 104
 RoHC 3, 1, 2, 3, 4, 5, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 19, 24, 25, 26, 28, 29, 31, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 89, 90, 91, 95, 96, 102, 104
 RoHC Channel 3, 2, 10, 13, 19, 46, 49, 50, 51, 52, 53, 90, 91, 96
 RoHC Channels Table 50, 96
 RoHC Compressor 8, 9, 10, 12, 13, 42, 52, 58, 71, 72, 74, 75, *See* Compressor
 RoHC Decompressor 8, 9, 10, 13, 14, 42, 52, 72, *See* Decompressor
 RoHC Interface Module 49, 50, 51, 96
 RoHC Module 49, 50, 91, 96
 RoHC-over-X 13, 26, 41
 RTP header 1, 5, 6, 7, 45, 58, 65, 74, 101
 RTS/CTS threshold 59, 61

S

Second Order 13, 97, *See* SO
 SO 13, 14, 16, 58, 74, 75, 97
Static Context 14
 STATIC-NACK 15, 17

T

Thinwire-II 5, 6, 19
 TWICE 7, 8

U

UDP header 1, 5, 6, 45, 58, 74
 U-mode 4, 15, 16, 17, 18, 19, 55, 58, 59, 61, 71, 74, 75, 76, 78, 79
 UMTS 1, 21, 26, 27, 28, 29, 55, 58, 76, 77, 81, 97, 102, 104
 Unidirectional Mode 15, 16, 97, 104, *See* U-mode

V

VoIP 3, 1, 5, 45, 55, 56, 57, 58, 60, 61, 62, 63, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 78, 97, 105

W

Wireless LAN 1, 93

Bibliography

- [1] C. Borman et al, “RObust Header Compression (ROHC): Framework and four profiles: RTP, UDP, ESP, and uncompressed”, RFC 3095, IETF, July 2001.
- [2] Farber, D. J., Delp, G. S., and Conte, T. M., “A Thinwire Protocol for connecting personal computers to the Internet”, Arpanet Working Group Requests for Comment, DDN Network Information Center, SRI International, Menlo Park, CA, Sept. 1984. RFC-914, IETF.
- [3] V. Jacobson, “Compressing TCP/IP Headers for Low-Speed Serial Links”, RFC 1144, IETF, February 1990.
- [4] M. Degermark, B. Nordgren, S. Pink, “IP Header Compression“, RFC 2507, IETF, February 1999.
- [5] S. Casner, V. Jacobson, “Compressing IP/UDP/RTP Headers for Low-Speed Serial Links”, RFC 2508, IETF, February 1999.
- [6] M. Degermark, H. Hannu, L.E. Jonsson, K. Svanbro, “Evaluation of CRTP Performance over Cellular Radio Networks”, IEEE Personal Communication Magazine, Volume 7, number 4, pp. 20-25, August 2000
- [7] T. Koren, S. Casner, J. Geevarghese, B. Thompson, P. Ruddy, “Enhanced Compressed RTP (CRTP) for Links with High Delay, Packet Loss and Reordering”, RFC 3545, IETF, July 2003.
- [8] M. Degermark, “Requirements for robust IP/UDP/RTP header compression”, RFC 3096, IETF, July 2001.
- [9] K. Svanbro, “Lower Layer Guidelines for Robust RTP/UDP/IP Header Compression”, RFC 3409, IETF, December 2002.
- [10] L-E. Jonsson, “RObust Header Compression (ROHC): Requirements on TCP/IP Header Compression”, RFC 4163, IETF, August 2005.
- [11] L-E. Jonsson, “RObust Header Compression (ROHC) Terminology and Channel Mapping Examples”, RFC 3759, IETF, April 2004.
- [12] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, “RTP: A Transport Protocol for Real-Time Applications”, RFC 3550, IETF, July 2003.
- [13] C. Bormann, “Robust Header Compression (ROHC) over PPP”, RFC 3241, IETF, April 2002.

- [14] G. Pelletier, L-E. Jonsson, K. Sandlund, "Robust Header Compression (ROHC): ROHC over Channels That Can Reorder Packets", RFC 4224, IETF, January 2006.
- [15] L-E. Jonsson, G. Pelletier, "Robust Header Compression (ROHC): A Compression Profile for IP", RFC 3843, IETF, June 2004.
- [16] G. Pelletier, "Robust Header Compression (ROHC): Profiles for User Datagram Protocol (UDP) Lite", RFC 4019, IETF, April 2005.
- [17] L-E. Jonsson, G. Pelletier, K. Sandlund, "Robust Header Compression (ROHC): A Link-Layer Assisted Profile for IP/UDP/RTP", RFC 4362, IETF, January 2006.
- [18] Z. Liu, K. Le, "Zero-byte Support for Bidirectional Reliable Mode (R-mode) in Extended Link-Layer Assisted RObust Header Compression (ROHC) Profile", RFC 3408, IETF, December 2002.
- [19] G. Pelletier, L. Jonsson, K. Sandlund, M. West, "RObust Header Compression (ROHC): A Profile for TCP/IP (ROHC-TCP)", Internet Draft, January 2006.
- [20] G. Pelletier, "RObust Header Compression (ROHC): Context Replication for RoHC Profiles", RFC 4164, IETF, August 2005.
- [21] C. Bormann, "Robust Header Compression (ROHC) over 802 networks", Internet Draft, February 2005.
- [22] ETSI, General Packet Radio Service (GPRS); Service Description; State 2 (Release 1998); EN 301 344 v7.4.1, September 2000.
- [23] 3GPP, General Packet Radio Service (GPRS); Service Description; Stage 2 (Release 7); TS 23.060 v7.0.0, March 2006.
- [24] 3GPP, Mobile Station (MS) – Serving GPRS Support Node (SGSN); Subnetwork Dependent Convergence Protocol (SND CP) (Release 6); TS 44.065 v6.5.0, September 2005.
- [25] 3GPP, Data Link (DL) layer; General aspects (Release 7); TS 44.005 v7.0.0, November 2005.
- [26] 3GPP, General Packet Radio Service (GPRS); Mobile Station (MS) – Base Station System (BSS) interface; Radio Link Control/Medium Access Control (RLC/MAC) protocol (Release 7); TS 44.060 v 7.3.0, January 2006.
- [27] 3GPP, Technical Specification Group GSM/EDGE; Radio Access Network; Overall description – Stage 2 (Release 6); TS 43.051 v6.0.0, November 2003.
- [28] 3GPP, Packet Data Convergence Protocol (PDCP) specification (Release 7); TS 25.323 v7.0.0, March 2006.
- [29] 3GPP, Radio Link Control (RLC) protocol specification (Release 7); TS 25.322 v7.0.0, March 2006.
- [30] 3GPP, General Universal Mobile Telecommunications System (UMTS) architecture (Release 6); TS 23.101 v6.0.0, December 2004.

- [31] Network Simulator 2, <http://www.isi.edu/nsnam/ns/>
- [32] ISO/IEC 3309:1991(E), "Information Technology - Telecommunications and information exchange between systems - High-level data link control (HDLC) procedures – Frame structure", International Organization For Standardization, Fourth edition 1991-06-01.
- [33] ITU-T recommendation I.122, Framework for frame mode bearer services, March 1993.
- [34] ITU-T recommendation I.150, B-ISDN asynchronous transfer mode functional characteristics, February 1999.
- [35] Ethernet, a local area network: data link layer and physical layer specifications version 1.0. Digital Equipment Corporation, Intel Corporation, Xerox Corporation, Sept. 1980.
- [36] C. Hornig, "A Standard for the Transmission of IP Datagrams over Ethernet Networks", RFC 894, IETF, April 1984
- [37] J. Postel, J. Reynolds, "A Standard for the Transmission of IP Datagrams over IEEE 802 Networks", RFC 1042, IETF, February 1988.
- [38] E. Rosen, A. Viswanathan, R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, IETF, January 2001.
- [39] F. Williams, "Fourth Generation Mobile", ACTS Mobile Summit 99, Sorrento, Italy, June 1999
- [40] João da Silva, "Beyond IMT-2000", European Commission, ITU-R WP8F Workshop, Geneva, March 2000
- [41] J. M. Pereira, "Fourth generation: now, it is personal", in Proc. 11th IEEE International Symposium of Personal, Indoor and Mobile Radio communications, London, UK, September 2000, Vol. 2, pp. 1009-1016.
- [42] B.G. Evans and K. Baughan, "Visions of 4G", Electronics & Communication Engineering Journal, Vol. 12, No. 6, pp. 293-303, December 2000.
- [43] DAIDALOS IST Project: DAIDALOS: "Designing Advanced Interfaces for the Delivery and Administration of Location independent Optimized personal Services". (FP6-2002-IST-1-506997), <http://www.ist-DAIDALOS.org>
- [44] G.Carneiro, "QoS Abstraction Layer in 4G Access Networks", *MSc Thesis*, November 2005
- [45] G. Carneiro, C. Garcia, P. Neves, Z. Chen, M. Wetterwald, M. Ricardo, P. Serrano, S. Sargento, and Albert Banchs, "The DAIDALOS Architecture for QoS over Heterogeneous Wireless Networks", In Proc. of the 14th IST Summit, June 2005
- [46] R. Koodli, "Fast Handovers for Mobile IPv6", RFC 4068, IETF, July 2005.
- [47] D. Durham, J. Boyle, R. Cohen, S. Herzog, R. Rajan, and A. Sastry, "The COPS (Common Open Policy Service) Protocol", RFC 2748, IETF, January 2000.

- [48] R. Braden, D. Clark, and S. Shenker, "Integrated services in the Internet architecture: an overview", RFC 1633, IETF, June 1994.
- [49] S. Shenker, J. Wroclawski, "General Characterization Parameters for Integrated Service Network Elements", RFC 2215, IETF, September 1997.
- [50] Carlson, E. Davies, Z. Wang, and W. Weiss, "An architecture for differentiated services", RFC 2475, IETF, December 1998.
- [51] J. Rajahalme, A. Conta, B. Carpenter, S. Deering, "IPv6 Flow Label Specification", RFC 3697, IETF, March 2004.
- [52] R. Braden, L. Zhang, S. Berson, S. Herzog, S. Jamin, "Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification", RFC 2205, IETF, September 1997.
- [53] D. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan, G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, IETF, December 2001.
- [54] P. Ji, Z. Ge, J. Kurose, and D. Towsley, "A Comparison of Hard-state and Soft-state Signaling Protocols", *In Proc. of SIGCOMM'03*, August 2003, Karlsruhe, Germany.
- [55] Information Technology Telecommunications, ISO/IEC final dis 15802-3 (IEEE P802.1D/D17), 25 May 1998.
- [56] A. Minaburo, K. Singh, L. Toutain, L. Nuaymi, "Proposed Behavior for Robust Header Compression over a radio link", *in IEEE International Conference on Communications (ICC)*, June 2004, Paris, France.
- [57] A. Minaburo, L. Nuaymi, K.D. Singh, L. Toutain, "Configuration and Analysis of Robust Header Compression in UMTS", *in Proc. 14th IEEE International Symposium of Personal, Indoor and Mobile Radio communications*, Beijing, China, September 2003, Vol. 3, pp. 2402-2406.
- [58] A. Yuannakoulis, E. Kuehn, "Evaluation of Header Compression Schemes for IP-Based Wireless Access Systems", *in IEEE Wireless Communications*, February 2005, Vol. 12, Issue 1, pp. 68-74.
- [59] A. Couvreur, L. Le Ny, A. Minaburo, G. Rubino, B. Sericola, L. Toutain, "Performance Analysis of a Header Compression Protocol: the RoHC Unidirectional Mode", *in Telecommunication Systems Journal*, January 2006, Vol. 31, Issue 1, pp. 85-98.
- [60] B. Wang, H.P. Schwefel, K.C. Chua, R. Kutka, C.Schmidt, "On Implementation and Improvement of Robust Header Compression in UMTS", *in Proc. 13th IEEE International Symposium of Personal, Indoor and Mobile Radio communications*, Lisboa, Portugal, September 2002, Vol. 3, pp. 15-18.
- [61] F.H.P. Fitzek, S. Rein, P. Seeling, and M. Reisslein, "Robust Header Compression (RoHC) Performance for Multimedia Transmission over 3G/4G Wireless Networks", *in Wireless Personal Communications Journal*, January 2005, Vol. 32, Issue 1, pp. 23-41.

- [62] IEEE 802.11, 1999 Edition (ISO/IEC 8802-11: 1999) IEEE Standards for Information Technology - Telecommunications and Information Exchange between Systems – Local and Metropolitan Area Network – Specific Requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.
- [63] IEEE 802.11a-1999 (8802-11:1999/Amd 1:2000(E)), IEEE Standard for Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications—Amendment 1: High-speed Physical Layer in the 5 GHz band.
- [64] IEEE 802.11b-1999 Supplement to 802.11-1999, Wireless LAN MAC and PHY specifications: Higher speed Physical Layer (PHY) extension in the 2.4 GHz band.
- [65] IEEE 802.11g-2003 IEEE Standard for Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications—Amendment 4: Further Higher-Speed Physical Layer Extension in the 2.4 GHz Band
- [66] VoIP Codecs, <http://www.ozvoip.com/voip-codecs/>
- [67] ITU-T G.729a codec, <http://www.itu.int/rec/T-REC-G.729/en>
- [68] ITU-T G.711 codec, <http://www.itu.int/rec/T-REC-G.711/en>
- [69] ITU-T G.723.1 codec, <http://www.itu.int/rec/T-REC-G.723/en>
- [70] NS2 Mathieu Lacage 802.11 model, <http://yans.inria.fr/ns-2-80211/>
- [71] F. Tobagi, L. Kleinrock, “Hidden Terminal Problem in Carrier Sense Multiple-Access and the Busy-Tone Solution”, in IEEE Transactions on Communications, Vol. Com-23, no. 13, December 1975.