



ESTUDIO Y EVALUACIÓN DEL MECANISMO DE COMPRESIÓN DE CABECERAS ROHC EN REDES INALÁMBRICAS

Rafael Pires Orozco

Tutor: Juan Carlos Guerri Cebollada

Cotutor: Kalle Ruttik

Trabajo Fin de Grado presentado en la Escuela Técnica Superior de Ingenieros de Telecomunicación de la Universitat Politècnica de València, para la obtención del Título de Graduado en Ingeniería de Tecnologías y Servicios de Telecomunicación

Curso 2018-19

Valencia, 25 de July de 2019



Resumen

Este proyecto se inicia con un estudio exhaustivo de los principios y propuestas más relevantes en el uso de la compresión de cabeceras IP aplicadas a las comunicaciones de redes inalámbricas. A continuación, ofrece una descripción detallada del Mecanismo Robusto de Compresión de Cabeceras (ROHC: por sus siglas en inglés) y su evolución. Esta descripción incluye una sección dedicada al reciente estándar del mecanismo TCP ROHC de IETF.

A continuación, se describe la plataforma experimental y herramientas utilizadas. La plataforma de hardware ha consistido en una Raspberry Pi 3 operando bajo el sistema operativo Raspbian y equipada con una interfaz WiFi. Se ha utilizado una implementación de ROHC de código abierto como base para llevar a cabo la evaluación experimental del mecanismo de compresión. Se han utilizado dos herramientas ROHC complementarias para recopilar estadísticas y diferentes modos de funcionamiento del mecanismo de compresión: relaciones de compresión globales para longitudes de paquetes y cabeceras, relaciones de compresión basadas en tipos de paquetes ROHC y modo de funcionamiento de ROHC. Los experimentos han incluido el uso de trazas capturadas a través de la interfaz WiFi utilizando las herramientas Sniffer y Stat de la librería ROHC, herramientas estadísticas haciendo uso de trazas de paquetes disponibles en Internet. Se han analizado varios perfiles ROHC correspondientes a diferentes configuraciones de pila de protocolos IP, por ejemplo, RTP/UDP/IP, UDP/IP, TCP/IP.

Los resultados incluyen las relaciones de compresión de cabecera y de paquetes, la compresión lograda por tipos de paquetes ROHC y los modos de operación ROHC. Los resultados muestran que el mecanismo ROHC es particularmente útil para reducir la sobrecarga introducida por los protocolos IP en aplicaciones caracterizadas por cargas útiles cortas, como Voz sobre IP.



Resum

Aquest projecte s'inicia amb un estudi exhaustiu dels principis i propostes més rellevants en l'ús de la compressió de capçaleres IP aplicades a les comunicacions de xarxes sense fils. A continuació, ofereix una descripció detallada del Mecanisme de Robust de Compressió de Capçaleres (ROHC: per les seues sigles en anglés) i la seua evolució. Aquesta descripció inclou una secció dedicada al recent estàndard del mecanisme TCP ROHC de IETF.

A continuació es descriu la plataforma experimental i eines utilitzades. La plataforma de maquinari ha consistit d'una Raspberry Pi 3 de sistema operatiu Raspbian i equipada amb una interfície Wifi. S'ha utilitzat una implementació de ROHC de codi obert com a base per a dur a terme l'avaluació experimental del mecanisme de compressió. S'han utilitzat dues eines ROHC complementàries per a recopilar estadístiques i diferents maneres de funcionamiento del mecanisme de compressió: relacions de compressió globals per a longituds de paquets i capçaleres, relacions de compressió basades en tipus de paquets ROCH i manera de funcionament de ROHC. Els experiments han inclòs l'ús de traces capturades a través de la interfície Wifi utilitzant el sniffer ROCH, eines estadístiques fent ús de traces de paquets disponibles en Internet. S'han analitzat diversos perfils ROHC corresponents a diferents configuracions de pila de protocols IP, per exemple, RTP/UDP/IP, UDP/IP, TCP/IP.

Els resultats inclouen les relacions de compressió de capçalera i de paquets, la compressió reeixida per tipus de paquets ROCH i les maneres d'operació ROHC. Els resultats mostren que el mecanisme ROHC és particularment útil per a reduir la sobrecàrrega introduïda pels protocols IP en aplicacions caracteritzades per càrregues útils curtes, com la Veu sobre IP.



Abstract

This project starts by an exhaustive study of the most relevant principles and proposals in the use of compression of IP headers applied to wireless network communications. It then provides an in-depth description of the Robust Header Compression Mechanism (ROHC) and its evolution. This description includes a section dedicated to one of the latest IETF ROHC TCP Standard.

Subsequently, the experimental setup and tools are described. The hardware platform consisted of a Raspberry Pi 3 running under the Raspbian operating system and equipped with a WiFi interface. An open-source ROHC implementation has been used as a basis to carry the experimental evaluation of the compression mechanism. Two accompanying Sniffer and Stat ROHC tools have been used to collect different statistics and operating conditions: overall compression ratios for both packet and headers lengths, compression ratios based on ROHC packet types and operating ROHC mode. The experiments have included the use of traces captured through the WiFi interface using the ROHC sniffer and statistical tools and packet traces available in the Internet. Various ROHC profiles corresponding to different IP protocol stack configurations, e.g., RTP/UDP/IP, UDP/IP, TCP/IP, have been analyzed.

The results include the header and packet compression ratios, compression achieved per ROHC-packet types, ROHC operating modes. The results show that the ROHC mechanism is particularly useful on reducing the overhead introduced by the IP protocols in applications characterized by short payloads, such as, Voice over IP.



ÍNDICE

Capítulo 1.	INTRODUCCIÓN	1
1.1	Motivación y beneficios de la compresión de las cabeceras.....	1
1.1.1	Motivación.....	1
1.1.2	Beneficios	2
1.2	Objetivos.....	2
1.3	Metodología.....	2
1.4	Organización de la memoria	4
Capítulo 2.	INTERNET Y COMPRESIÓN DE CABECERAS.....	5
2.1	Internet Protocol Architecture.....	5
2.1.1	Pila de protocolos TCP/IP	5
2.1.2	Cabeceras de protocolos IP.....	7
2.2	Estándares de compresión: principios de funcionamiento.....	8
2.3	Estándares de compresión de cabeceras: evolución histórica	9
2.3.1	Compressed Transport Control Protocol (CTCP).....	9
2.3.2	Mejoras al mecanismo CTCP	9
2.3.3	IP Header Compression (IPHC).....	11
2.3.4	Compressed Real Time Protocol (CRTP)	12
2.4	Resumen de los principales estándares de compresión de cabeceras IETF.....	12
Capítulo 3.	COMPRESIÓN DE CABECERAS ROBUSTA (ROHC).....	14
3.1	Introducción.....	14
3.2	Contexto y Estados.....	15
3.2.1	Parámetros de canal	16
3.2.2	Perfiles.....	17
3.2.3	Contextos e identificadores de contexto.....	17
3.3	Diagrama de estados del compresor	17
3.4	Diagrama de estados del descompresor.....	18
3.5	Modos de Transmisión	18
3.6	Métodos de Codificación.....	21
3.7	Compresión RTP.....	22
3.8	Paquetes ROHC	22
3.8.1	Tipos de paquetes ROHC	22
3.8.2	Paquetes ROHC feedback	23
3.9	El protocolo ROHC.....	24



3.9.1	Estado de inicialización y reinicialización	24
3.9.2	Estado de compresión de primer orden	27
3.9.3	Estado de compresión de Segundo Orden	27
3.9.4	Transición entre modos	28
3.10	Estándares ROHCv2 y ROHC TCP	28
3.10.1	ROHCv2.....	29
3.10.2	ROHC-TCP	29
Capítulo 4.	ROHC Y HERRAMIENTAS SOFTWARE	32
4.1	LIBRERÍA ROHC	32
4.2	ROHC STATS	33
4.3	ROHC SNIFFER.....	34
4.4	WIRESHARK.....	35
4.5	Herramientas de procesamiento y visualización	35
Capítulo 5.	PLATAFORMA EXPERIMENTAL Y EVALUACIÓN	37
5.1	Descripción de la plataforma	37
5.1.1	Hardware y software	37
5.1.2	Métricas.....	39
5.2	Evaluación de prestaciones – ROHC STAT	39
5.2.1	Datos experimentales	39
5.2.2	Video streaming - Youtube.....	39
5.2.3	Video retrieval - YouTube.....	42
5.2.4	Reproducción de música - Spotify	43
5.2.5	Video conferencia – Hangouts.....	45
5.2.6	VoIP G.711.....	47
5.3	Evaluación de prestaciones – ROHC Sniffer	48
5.3.1	Escenario de experimentación	49
5.3.2	Audio conferencia – Google Hangouts	49
5.3.3	Video conferencia – Hangouts.....	52
5.4	Resumen de resultados	55
Capítulo 6.	CONCLUSIONES Y TRABAJOS FUTUROS.....	56

LISTA DE FIGURAS

Figura 1.1 Red Inalámbrica de Acceso	1
Figura 2.1 Pila de protocolos TCP/IP y unidades de datos	6
Figura 2.2 Cabecera RTP	7
Figura 2.3 Cabecera UDP.....	7
Figura 2.4 Cabecera IPv4.....	8
Figura 2.5 Partes estáticas y dinámicas de la cabecera UDP.....	8
Figura 2.6 Compresión de cabeceras - codificación delta.....	9
Figura 2.7 Esquema de compresión con marco de referencia [4].....	10
Figura 2.8 Mecanismo propuesto por Rossi et al. [6].....	10
Figura 3.1 ROHC dentro de la pila de protocolos TCP/IP.....	15
Figura 3.2. Contexto ROHC.....	16
Figura 3.3 Diagrama de estados del Compresor [2].....	18
Figura 3.4 Diagrama de estados del descompresor.....	18
Figura 3.5 Transiciones entre Modos.....	19
Figura 3.6 Transiciones del compresor - Unidireccional mode [2].	19
Figura 3.7 Transiciones del compresor - Bidirectional Optimistic mode [2]	20
Figura 3.8 Transiciones del compresor - Bidirectional Reliable mode [2].....	20
Figura 3.9 Intervalo utilizado por la codificación LSB [2]	21
Figura 3.10 Formato de paquete ROHC type	23
Figura 3.11 Formato del paquete ROHC feedback Type 2	24
Figura 3.12 Protocolo ROHC en modo unidireccional	25
Figura 3.13 ROHC state transition in bidirectional mode.....	25
Figura 3.14 Formato del paquete IR	26
Figura 3.15 Formato del paquete IR DYN	26
Figura 3.16 Campos estáticos y dinámico de la cabecera IPv4	27
Figura 3.17 Paquetes R0-CRC and RO.....	28
Figura 3.18 Cambio de modo	28
Figura 4.1 ROHC library API to create a compressor	33
Figura 4.2 Resultados de la herramienta ROHC STATS	33
Figura 4.3 Resultados de la herramienta ROHC Sniffer	34
Figura 5.1 Plataforma experimental - hardware	38
Figura 5.2 Plataforma experimental -software	38
Figura 5.3 Video y audio streaming – pantalla Wireshark.....	39
Figura 5.4 Video streaming – cabeceras	40



Figura 5.5 Video y audio streaming – ganancia de compresión por tipo de paquete.....	41
Figura 5.6 Video y audio streaming – porcentajes por tipo de paquetes ROHC	41
Figura 5.7 Stored video retrieval - trazas	42
Figura 5.8 Stored video retrieval – ganancia de compresión.....	42
Figura 5.9 Stored video retrieval – porcentaje por tipo de paquetes ROHC	43
Figura 5.10 Stored music retrieval.....	43
Figura 5.11 Spotify – ganancia de compresión.....	44
Figura 5.12 Spotify – porcentaje por tipo de paquetes ROHC	44
Figura 5.13 Video conferencing Hangouts – pantalla Wireshark.....	45
Figura 5.14 Video conferencia Hangouts – trazas	45
Figura 5.15 Video conferencing Hangouts – ganancia por tipos de paquete ROHC	46
Figura 5.16 Video conferencia Hangouts – porcentaje por tipo de paquete ROHC	46
Figura 5.17 VoIP G.711 – pantalla Excel	47
Figura 5.18 VoIP G.711 – trazas	47
Figura 5.19 VoIP G7.11 – ganancia de compresión por tipo de paquete ROHC	48
Figura 5.20 VoIP G.711 – porcentaje por tipo de paquetes ROHC	48
Figura 5.21 Audio conferencia Hangouts – resultados ROHC Sniffer - ROHCv1.....	50
Figure 5.22 Audio conferencia Hangouts – resultados ROHC Sniffer– ROHCv2.....	51
Figura 5.23 Video conferencia – resultados ROHC Sniffer - ROHCv1	53
Figura 5.24 Video conferencia Hangouts – resultados ROHC Sniffer - ROHCv2.....	54



LISTA DE TABLAS

Tabla 1.1 Ganancia de compresión [1].....	2
Tabla 1.2 Cronograma del TFG.....	4
Tabla 2.1 Familia de protocolos Internet.....	6
Tabla 2.2 Principales estándares de compresión de cabeceras de la IETF [1]	13
Tabla 3.1 Parámetros de canal [2].....	16
Tabla 3.2 Perfiles	17
Tabla 3.3 Tipos de paquetes ROHC [2]	23
Tabla 3.4 Ganancia de compresión esperadas – norma ROHC-TCP [24]	29
Tabla 3.5 Tipos de paquetes del perfil ROHC TCP [24].	30
Tabla 3.6 Principales perfiles definidos por las diferentes normas ROHC	31
Tabla 4.1 ROHC profiles supported by ROHC library release 2.2 [25].	32
Tabla 5.1 Plataforma experimental	37
Tabla 5.2 Experimental data traces	39
Tabla 5.3 Resultados audio conferencia Hangouts – ROHC Sniffer	49
Tabla 5.4 Video conferencia– Sniffer tool	52
Tabla 5.5 Ganancias por aplicación	55



ACRÓNIMOS

ACK	Acknowledgement
CH	Compressed Header
CID	Context Identifier
CIoT	Cellular Internet of Things
CRTP	Compressed Real Time Protocol
CTCP	Compressed Transport Control Protocol
DASH	Dynamic Adaptive Streaming over HTTP
DC	Dynamic Context
DF	Don't Fragment
DSL	Digital Subscriber Line
DYN	Dynamic
ECRTP	Enhanced Compressed RTP
ESP	Encapsulating Security Payload
FC	Full Context
FO	First Order
FTP	File Transfer Protocol
IETF	Internet Engineering Task Force
HTTP	Hypertext Transfer Protocol
ICMP	Internet Control Message Protocol
IGMP	Internet Group Management Protocol
HIS	Internet Header Length
IoT	Internet of Things
IPHC	Internet Protocol Header Compression
IPv4	Internet Protocol – Version 4
IPv6	Internet Protocol – Version 6
IR	Initialization and Refresh
KPI	Key Performance Indicator
LSB	Least-significant Bits
LZW	Lempel-Ziv-Welch
MF	More Fragments
MRRU	Maximum reconstructed reception unit.
NACK	Negative Acknowledgment
NB-IoT	Narrow Band IoT



NC	No Context
O-Mode	Optimistic Mode
PPP	Point-to-Point Protocol
RF	Request Flag
RFC	Request For Comments
ROCCO	RObust Checksum-based COmpression
ROHC	Robust Header Compression
RTP	Real-Time Transport Protocol
SC	Static Context
SCTP	Stream Control Transmission Protocol
SMTP	Simple Mail Transfer Protocol
SN	Sequence Number
SO	Second Order
TCP	Transmission Control Protocol
TOS	Type Of Service
TS	Timestamp
TTL	Time To Live
UCH	Uncompressed Header
UDP	User Datagram Protocol
U-Mode	Unidirectional Mode
VoIP	Voice over IP
WiFi	Wireless Fidelity
W-LSB	Windowed Least Significant Bits

Capítulo 1. INTRODUCCIÓN

1.1 Motivación y beneficios de la compresión de las cabeceras

El desarrollo de nuevas aplicaciones y el crecimiento exponencial de los servicios móviles basados en Internet hacen necesaria la utilización de mecanismos que permitan hacer un mejor uso del ancho de banda del canal de comunicaciones. Esto es particularmente interés en el caso de los enlaces inalámbricos caracterizados por su limitado ancho de banda y su elevada tasa de errores [1].

1.1.1 Motivación

A medida que aumenta la demanda de servicios de comunicaciones móviles, la gestión eficiente de los recursos de la red requiere el despliegue de múltiples mecanismos que permitan reducir la cantidad de datos a transmitir. Dado que las cargas útiles (payload en inglés) de los paquetes son en ocasiones más cortas que las cabeceras de los paquetes [1][2], se viene investigando el uso de mecanismos de compresión de las cabeceras de los paquetes. Un gran número de aplicaciones en la actualidad, como VoIP, juegos interactivos y audio streaming, se caracterizan por tener cargas útiles cortas, véase la Figura 1.1. Esta característica se traduce en altos índices de sobrecarga. Además, el diseño y tamaño de los campos de cabecera de los distintos protocolos no han sido diseñados teniendo en cuenta las características de los enlaces de comunicaciones inalámbricas. A través de la conexión de extremo a extremo, compuesta por múltiples saltos, las cabeceras de protocolo son utilizadas por las distintas capas de protocolo para realizar los distintos servicios de red: enrutamiento, control de flujo, control de errores, control de calidad entre otros. Sin embargo, en el enlace de acceso inalámbrico, las cabeceras de los protocolos de red y superiores no realizan ninguna función. Esta es la motivación detrás del diseño de mecanismos de compresión de las cabeceras de los protocolos de la capa red y de las capas superiores. Según algunos estudios, estos esquemas de compresión pueden proporcionar en muchos casos un ahorro superior al 90%, véase la Tabla 1.1[1].

El mecanismo de compresión robusta de cabeceras (ROHC: por sus siglas en inglés) es un estándar de compresión de cabeceras especialmente diseñado para ser utilizado en enlaces inalámbricos propensos a errores y largas latencias (RTT: en inglés) [2].

Este proyecto estudia el uso del mecanismo de ROHC definido por el estándar RFC 3095 publicado el año 2001. También se aborda el estudio del estándar RFC 6846 introducido en el año 2013. Este último estándar es de particular interés dado que especifica el mecanismo ROHC del Protocolo de Control de Transmisión (TCP: por sus siglas en inglés). Debido al amplio uso de TCP por muchas aplicaciones de Internet, como YouTube, la compresión de las cabeceras TCP debería reducir eficazmente la carga a la que se exponen los enlaces de acceso inalámbrico.

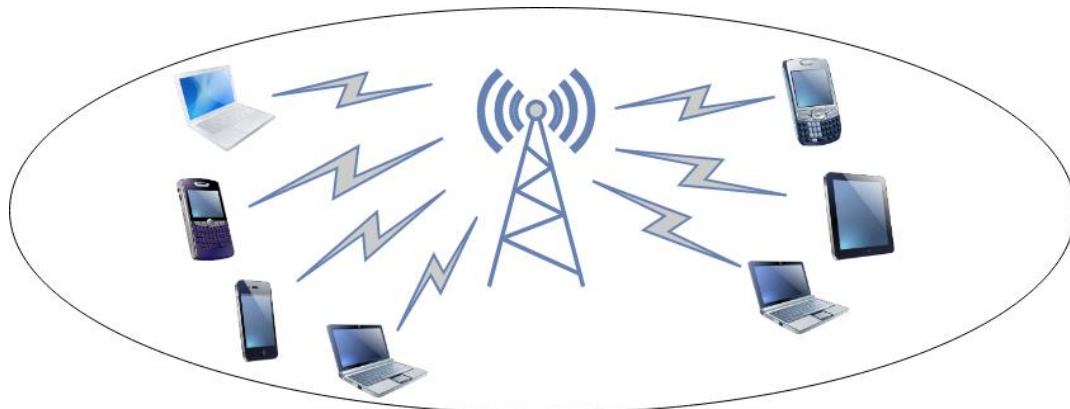


Figura 1.1 Red Inalámbrica de Acceso

Protocolo	Tamaño de la cabecera original (Bytes)	Tamaño de la cabecera comprimida (Bytes)	Ganancia (%)
IP4/TCP	40	4	90
IP4/UDP	28	1	96.4
IP4/UDP/RTP	40	1	97.5
IP6/TCP	60	4	93.3
IP6/UDP	48	3	93.75
IP6/UDP/RTP	60	3	95

Tabla 1.1 Ganancia de compresión [1]

1.1.2 Beneficios

Los principales beneficios que aportan los mecanismos de compresión de cabeceras pueden resumirse brevemente de la siguiente manera [1]:

- Disminución de la sobrecarga introducida por las cabeceras de paquetes (ahorro de ancho de banda). Esta disminución es particularmente relevante en el caso de aplicaciones con cargas útiles cortas, por ejemplo, aplicaciones de voz. En los últimos años, el desarrollo de aplicaciones Internet de las Cosas (IoT) también ha motivado el interés por el uso del esquema de compresión de las diversas cabeceras de los protocolos Internet.
- Reducción de la pérdida de paquetes. Dado que los canales inalámbricos son propensos a errores, la reducción de la longitud del paquete puede ayudar a reducir la probabilidad de pérdida de paquetes debido a errores.
- Mejor tiempo de respuesta interactiva. Se espera que la compresión de las cabeceras de paquetes tenga un impacto positivo en la reducción de la carga del canal. Esto se traduce en una reducción del tiempo de transmisión y, finalmente, en el tiempo de respuesta global.
- Disminución del coste de infraestructura, más usuarios por canal de ancho de banda significa menos costes de despliegue de infraestructura. La reducción de la carga de trabajo de la red se traduce en una reducción de los recursos que deben desplegarse.

1.2 Objetivos

Los principales objetivos del proyecto se resumen de la siguiente manera:

- Estudio de los mecanismos de ROHC [2]. Estudio de los principios de los mecanismos de compresión y las estructuras de datos necesarias para realizar un seguimiento adecuado de los procesos de compresión/descompresión.
- Desarrollo de una plataforma experimental y realización de experimentos para evaluar las principales funcionalidades del mecanismo de compresión ROHC.
- Recomendaciones. Realizar un análisis detallado de los resultados de los mecanismos de ROHC obtenidos de los experimentos. Identificar las capacidades y limitaciones de los mecanismos de ROHC cuando se utilizan en diferentes aplicaciones: video streaming, comunicaciones VoIP, etc. Se analizarán los beneficios aportados en términos de la ganancia y ratio de compresión obtenidos para cada una de las aplicaciones.

1.3 Metodología

La Tabla 1.2 muestra la planificación del proyecto. Como se puede observarse de la tabla, las actividades del proyecto se organizaron en tres partes:

- Estudio del estado del arte, estándares y la librería ROHC. Esta actividad se inició en febrero. Se dividió en cuatro tareas:
 - Definición del tema y elaboración del plan de trabajo. El tema del proyecto se definió en mayo de 2018. El plan inicial del proyecto fue presentado e incluido en el *Learning Agreement* firmado entre la UPV y la Universidad Aalto de Helsinki donde realicé una estancia de movilidad. A finales de febrero de 2019 se preparó un primer borrador del plan de trabajo.
 - Estudio del estado de arte. Una vez definido el plan de trabajo, mi primera tarea consistió en la búsqueda de bibliografía. Mi co-director en Aalto me sugirió centrar mi estudio en el mecanismo ROHC. En la Universidad de Aalto existe mucho interés en este mecanismo ya que ahí se vienen realizando importantes desarrollos en el área de las comunicaciones 5G. Uno de los principales temas que se vienen abordando incluye el despliegue de aplicaciones Internet de las Cosas en redes móviles (C-IoT por sus siglas en inglés).
 - Estudio de los estándares. El estudio de la norma RFC 3095 fue una de las primeras tareas a fin de familiarizarse con la terminología de los diferentes mecanismos. También se consultó el RFC 6846, ya que muchas aplicaciones utilizan el protocolo TCP.
 - Estudio de la librería y herramientas de ROHC. La librería y las herramientas de ROHC se han utilizado en el desarrollo y evaluación experimental. La librería es una implementación de código abierto de los mecanismos de ROHC desarrollada por la empresa Viveris.

- Plataforma experimental y pruebas
 - Configuración de la plataforma hardware y software. Para el desarrollo de la plataforma experimental se adquirió una Raspberry Pi3 equipada con una interfaz WiFi a la que se le instaló el sistema operativo Raspbian y las aplicaciones descritas en los siguientes puntos.
 - Instalación de la librería y herramientas de ROHC. Esta tarea requirió familiarizarse con el software e identificar las diversas herramientas disponibles para llevar a cabo la evaluación experimental. Como parte de esta tarea, fue necesario revisar el tutorial, estudiar parte del código y las funcionalidades de las diversas herramientas y revisar las limitaciones de la implementación del código.
 - Planificación de la evaluación experimental. Se preparó una primera planificación de las tareas. La planificación tuvo que ser revisada y las tareas de desarrollo tuvieron que modificarse debido a las limitaciones del software y a la complejidad del desarrollo de un sistema de compresión ROHC.
 - Evaluación experimental. La tarea se llevó a cabo utilizando varias aplicaciones haciendo uso de diferentes configuraciones de pila de protocolos.

- Análisis de datos e informes
 - Introducción y conclusión. Estos dos capítulos de la memoria presentan y resumen los principales resultados del proyecto, respectivamente.
 - Revisión de los esquemas de compresión. El estudio y redacción de este capítulo fue una de las tareas más difíciles y largas del proyecto. El estudio de una norma requiere familiarizarse con muchos términos y mecanismos de protocolo nuevos.
 - Análisis de los resultados. En esta tarea se analizan primeramente las prestaciones de los mecanismos ROHC. Las dos principales métricas de interés son la ganancia y la ratio de compresión obtenidos para las diferentes aplicaciones utilizadas en las pruebas experimentales.
Dado que uno de los objetivos de este proyecto es el identificar los beneficios que puede aportar el uso del mecanismo, también se incluye un análisis detallado de las prestaciones de la librería ROHC. En concreto se analiza en detalle los resultados de

las métricas de compresión teniendo en cuenta los mecanismos empleados por las pilas de protocolos empleados por cada aplicación.

ACTIVIDAD	FEBRERO	MARZO	ABRIL	MAYO	JUNIO
BÚSQUEDA BIBLIOGRÁFICA Y SOFTWARE					
Tema y plan de trabajo	■				
Búsqueda bibliográfica	■	■	■	■	
Estudio de estándares		■	■	■	
Librería ROHC y herramientas		■	■	■	
PLATAFORMA EXPERIMENTAL Y PRUEBAS					
Configuración de la plataforma		■	■	■	
Instalación de la librería ROHC y herramientas			■	■	
Planificación de las pruebas			■	■	
Experimentos				■	■
ANÁLISIS DE RESULTADOS Y MEMORIA					
Introducción					■
Mecanismos de compresión		■	■	■	
Descripción de la plataforma			■	■	
Análisis de resultados				■	■
Conclusiones					■

Tabla 1.2 Cronograma del TFG

1.4 Organización de la memoria

El capítulo 2 introduce la arquitectura del protocolo de Internet y motiva la necesidad de estándares de compresión de encabezados IP. Incluye una revisión de los principales esfuerzos de estandarización de los mecanismos de compresión de cabeceras desarrollados en los últimos años.

En el capítulo 3 se estudia de manera exhaustiva las normas ROHC. Primeramente, se inicia una introducción a la terminología, las estructuras de datos y los diferentes formatos de paquetes utilizados por el mecanismo de compresión de cabeceras. El funcionamiento del compresor y del descompresor se describe utilizando los diagramas de estado y diagramas temporales. También se ha incluido una sección sobre el perfil TCP redefinido por el RFC 6846.

El capítulo 4 presenta el software de código abierto de ROHC y las herramientas de software asociadas utilizadas en la configuración y evaluación experimental del desempeño del esquema de ROHC.

El capítulo 5 describe la plataforma experimental y los resultados experimentales. El análisis de los resultados se realiza para cada aplicación. También se incluye una comparación general de la ganancia y la ratio de compresión conseguidas para las diferentes aplicaciones. Se incluye una comparativa de los resultados donde se resalta el uso de las diferentes pilas de protocolos y características de las aplicaciones: tamaño de paquetes, tamaños de cabeceras y mecanismos de los protocolos.

El capítulo 6 enumera las conclusiones, lecciones aprendidas y principales desafíos en el desarrollo de esquemas de compresión de encabezados

Capítulo 2. INTERNET Y COMPRESIÓN DE CABECERAS

El objetivo principal de este capítulo es describir el funcionamiento de la pila de protocolos de Internet centrándose en la sobrecarga introducido durante el proceso de encapsulación de los paquetes utilizados para la transferencia de datos. A continuación, se motiva el uso del mecanismo de compresión de cabeceras como un medio para reducir la sobrecarga introducida por las cabeceras de los diversos protocolos con el fin de reducir la sobrecarga.

El capítulo está organizado en tres apartados. El primer apartado presenta la familia de protocolos TCP/IP. Se explica brevemente el papel de cada capa. A continuación, se explica el proceso de encapsulación y se describen las cabeceras introducidas por algunos de los protocolos más populares. La sección 2.2 motiva la necesidad de comprimir los campos de cabecera como medio para reducir la sobrecarga introducida por los distintos campos. Por último, en la sección 2.3 se describen brevemente los principales mecanismos de compresión de cabeceras introducidos a lo largo de los años resaltando sus debilidades y aportaciones al diseño de mejores mecanismos. En el Capítulo 3 se ofrece una visión general detallada del estándar de compresión ROHC de la IETF.

2.1 Internet Protocol Architecture

Las diferentes capas de Internet realizan una función distinta en las redes de comunicación. Cada capa es responsable de realizar un conjunto determinado de servicios necesarios para hacer posible la comunicación con otro sistema. Cada capa depende de los servicios prestados por la capa inferior y proporciona una serie de servicios a la capa superior. Para ello, cada capa debe añadir información adicional al mensaje. La información añadida se pone en una cabecera. En el receptor, el host procesará la información de la cabecera y tendrá que retirarlo antes de pasar el mensaje a la capa superior. El proceso de agregar y quitar cabeceras se conoce como los procesos de encapsulación y desencapsulación, respectivamente. La siguiente sección describirá la pila de protocolos TCP/IP y el proceso de encapsulación.

2.1.1 Pila de protocolos TCP/IP

La pila de protocolos TCP/IP es la arquitectura de protocolos utilizada en Internet. Consta de cinco capas (véase la tabla 2.1). Los principales servicios proporcionados por los protocolos de cada capa pueden describirse de la siguiente manera:

- Capa 1: Capa física. Esta capa proporciona las especificaciones físicas, eléctricas, bandas de radio y frecuencias utilizadas por los diferentes dispositivos de red.
- Capa 2: Capa de enlace. Esta capa comprende los protocolos necesarios para la entrega fiable, sin errores y sin pérdidas de los datos de un dispositivo de red a otro conectado mediante un enlace directo. El control de error y de flujo son los dos servicios principales que proporcionan los protocolos de enlace de datos. Las especificaciones sobre las direcciones de red y el tamaño máximo de paquetes que puede soportar una red, los protocolos utilizados para controlar el acceso al medio subyacente y el direccionamiento de hardware forman parte de la capa de enlace de datos.
- Capa 3: Capa de red. Los principales servicios que proporciona la capa de red son: direccionamiento y encaminamiento (routing). Los protocolos de esta capa hacen posible la entrega de datos desde un host a uno o varios hosts a través de Internet, es decir, a través de múltiples redes interconectadas. La capa de red define la estructura de direcciones de Internet, así como su gestión, los campos de los paquetes de Internet y los protocolos de enrutamiento.
- Capa 4: Capa de transporte. La capa de transporte define los protocolos para la comunicación de extremo a extremo. El Protocolo de datagramas de usuario (UDP) y el Protocolo de control de transmisión (TCP) son los dos principales protocolos de transporte definidos por el IETF. El primero es un protocolo sin conexión, mientras que el protocolo TCP es un protocolo orientado a la conexión.

- Capa 5: Capa de aplicación. Los protocolos de esta capa interactúan directamente con las aplicaciones del usuario final. Los protocolos de aplicación especifican el formato y el significado de los mensajes. La capa de aplicación proporciona los protocolos para el intercambio de correo electrónico, la transferencia de archivos, la navegación web, los servicios telefónicos y las aplicaciones de videoconferencia.

Dado que hay muchas aplicaciones y muchas tecnologías, cada capa está compuesta por protocolos diferentes. La Tabla 2.1 proporciona una lista de algunos protocolos comunes utilizados en cada una de las distintas capas de los protocolos de Internet. Por ejemplo, la capa física incluye protocolos para los diferentes medios de transmisión, como las líneas DSL (Digital Subscriber Lines), 10Base-T y otros. La capa de aplicación tendrá que proporcionar servicios para muchas y diversas aplicaciones.

Capa	Protocolos
Aplicación	HTTP, FTP, Telnet, RTP, SMTP, ...
Transporte	TCP, UDP, SCTP,...
Red	IPv4, IPv6, ICMP, IGMP,...
Enlace de datos	Ethernet, WiFi, PPP,...
Física	RS-232, DSL, 10Base-T,...

Tabla 2.1 Familia de protocolos Internet

La Figura 2.1 muestra la pila de protocolos TCP/IP y los Protocol Data Units (PDUs) utilizadas para realizar el proceso de transferencia de datos. Como se puede ver en la figura, el proceso comienza en capa de aplicación, el mensaje de datos. La capa de aplicación añade una cabecera al mensaje, representado por el cuadrado gris oscuro de la figura. Los protocolos de la capa de aplicación utilizarán la información insertada en la cabecera para realizar las acciones descritas anteriormente correspondientes a cada capa de la arquitectura. A continuación, el mensaje pasa a la capa de transporte. Esta capa también añadirá una cabecera con la información necesaria para realizar los servicios de los protocolos. El proceso de encapsulación se repite pasando de una capa superior a una inferior su unidad de datos. Finalmente, La capa física enviará los datos al otro sistema. Como se puede ver en la figura, la sobrecarga introducida en cabeceras puede ser un problema.

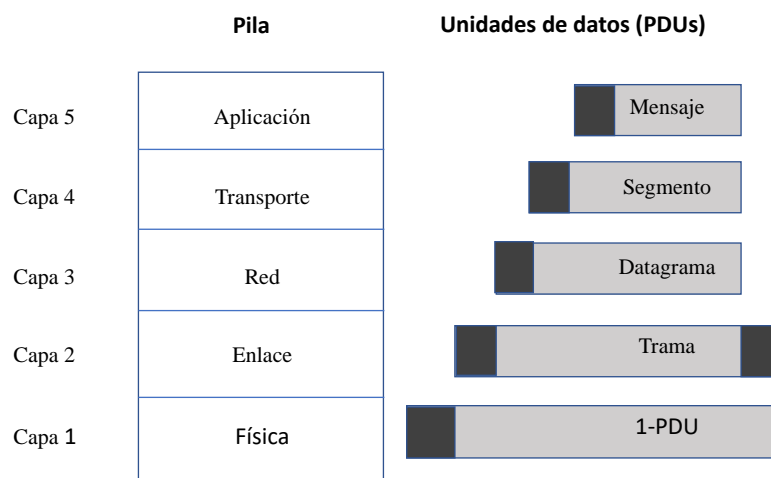


Figura 2.1 Pila de protocolos TCP/IP y unidades de datos

2.1.2 Cabeceras de protocolos IP

Como ya se indicó, la información incluida en las cabeceras de las distintas unidades de datos (PDUs en inglés) es utilizada por los protocolos para llevar a cabo distintos servicios. La Figura 2.2 muestra el formato de cabecera del protocolo RTP. Como se puede ver en la figura, el tamaño mínimo total de la cabecera RTP es de 12 octetos. Como se ha explicado anteriormente, el mensaje se encapsulará en un segmento UDP. La Figura 2.3 muestra el formato de cabecera UDP. En este caso, se añadirán otros ocho bytes. Como el segmento UDP se encapsulará en un paquete IP, véase Figura 2.4, se añadirán al menos otros 20 octetos antes de enviar el paquete a la capa de enlace de datos. En total se utilizarán 40 bytes para transmitir las cabeceras.

El protocolo IP hará uso de las direcciones IP para enrutar a través de las redes los paquetes. El protocolo UDP utilizará la identificación del puerto de destino para entregar los datos a la aplicación de destino.

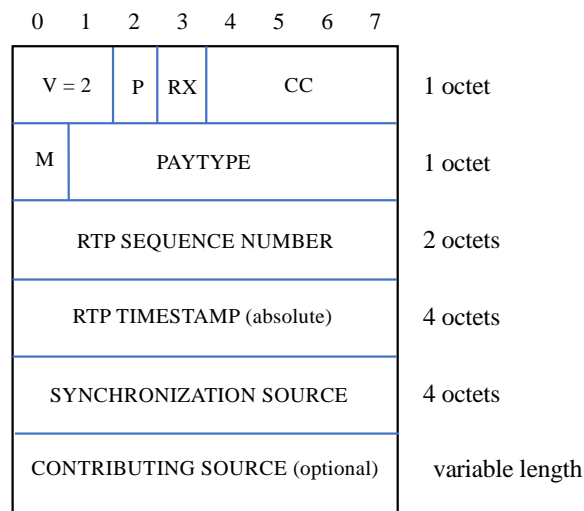


Figura 2.2 Cabecera RTP

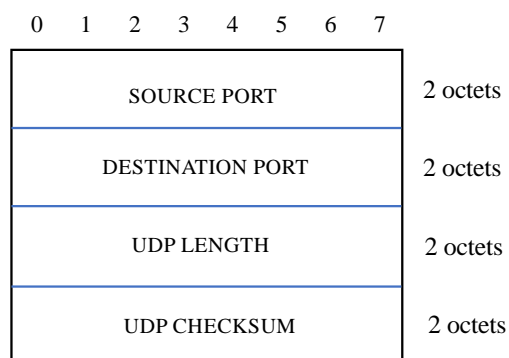


Figura 2.3 Cabecera UDP

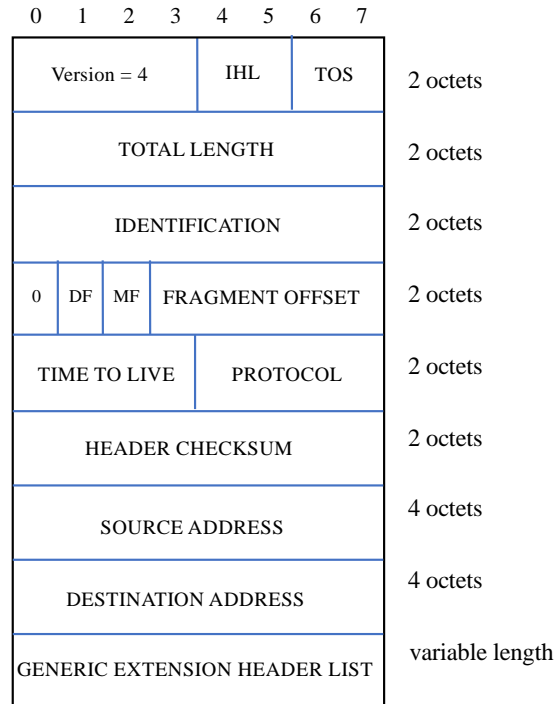


Figura 2.4 Cabecera IPv4

2.2 Estándares de compresión: principios de funcionamiento

La idea de comprimir las cabeceras de los paquetes fue introducida por primera vez por Van Jacobson en 1990 [3]. Desde entonces se han introducido varios mecanismos de compresión de cabeceras. La principal motivación de un protocolo de compresión es reducir la cantidad de datos que se transmiten a través de un canal. Esto es posible porque parte de la información incluida en la cabecera no cambia con el tiempo para una comunicación determinada. Los mecanismos de compresión más simples basaron su funcionamiento en la creación de una copia de los campos originales en el receptor. Después, la fuente sustituye los campos originales por los campos más cortos. Sin embargo, esto no es posible para todos los campos que componen la cabecera. Por lo tanto, el diseño de un mecanismo de compresión debe tener en cuenta las características de los distintos campos. La Figura 2.5 muestra la manera en que el protocolo ROHC define los diferentes campos de las cabeceras UDP en dos partes principales.

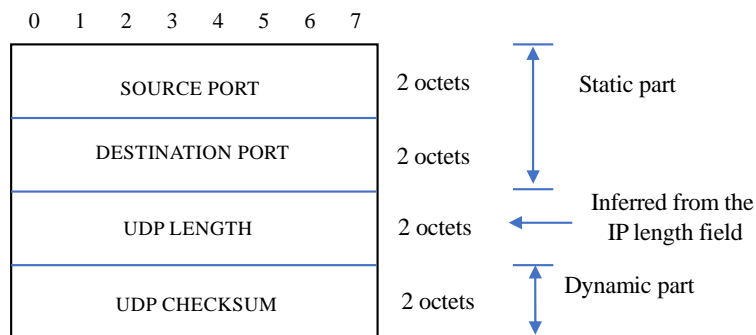


Figura 2.5 Partes estáticas y dinámicas de la cabecera UDP [2]

- Parte estática. Los campos que permanecen constantes durante la entrega de un flujo de paquetes tienen como origen y destino el mismo número de puerto.
- Parte dinámica. Los campos que cambian de un segmento a otro. En el caso del protocolo UDP, el CRC debe calcularse a partir de la información contenida en el encabezado.

En cuanto al campo de longitud, el protocolo ROHC puede inferir su valor a partir de la longitud del paquete IP, es decir, debe coincidir con el campo de longitud IP. Como se verá más adelante, esta clasificación ha tenido un impacto importante a la hora de diseñar el protocolo ROHC.

2.3 Estándares de compresión de cabeceras: evolución histórica

Desde la introducción del primer algoritmo de compresión de cabeceras por Van Jacobson, se han introducido numerosos estudios en la literatura y definido varios estándares con el objetivo de desarrollar algoritmos de compresión de cabeceras robustos. En esta sección se describen algunas de las obras más relevantes. En el siguiente capítulo se estudiará en detalle la estructura y el funcionamiento del esquema de compresión de cabeceras robusto (ROHC).

2.3.1 Compressed Transport Control Protocol (CTCP)

Van Jacobson introdujo el primer esquema de compresión de cabeceras IP denominado Compressed Transport Control Protocol (CTCP o Van Jacobson Header Compression) [3]. Este mecanismo fue diseñado específicamente para la pila de protocolos TCP/IP. CTCP procesa las cabeceras TCP e IP y comprime los 40 bytes de las cabeceras TCP e IP a 4 bytes. Su simplicidad es una de las principales características del CTCP. Sin embargo, el mayor inconveniente es que es muy sensible a los errores: un error en un paquete con cabecera comprimida produce la propagación del error e inclusive la pérdida de los paquetes cuya codificación depende del paquete afectado.

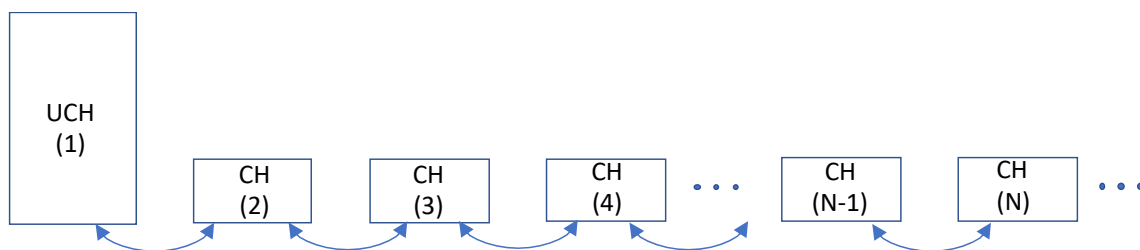


Figura 2.6 Compresión de cabeceras - codificación delta

Como se observa en la Figura 2.6, CTCP se basa en la codificación delta. El primer paquete se envía sin compresión; la cabecera se envía sin comprimir (UCH) mientras que los siguientes paquetes se envían con las cabeceras comprimidas, denotados por CH en la figura. CTCP determina la diferencia entre dos cabeceras de paquetes. CTCP solo tiene que enviar en las cabeceras comprimidas la diferencia entre el valor del campo correspondiente en cada una de dos paquetes consecutivos. Esta diferencia se conoce como delta.

El mayor inconveniente de la codificación delta es que es muy vulnerable a errores y pérdidas de paquetes. Si una de las cabeceras haciendo uso de la codificación delta se pierde, todos los paquetes que siguen se verán afectados. Es por esta razón que el mecanismo CTCP requiere la ayuda de los mecanismos de control de error de las capas superiores. En caso de error, CTCP renvía todos los paquetes perdidos sin comprimir. De esta manera, se evita la necesidad de tener que implementar un protocolo de señalización entre el compresor y el descompresor. Sin embargo, no se recomienda su uso en canales inalámbricos, donde los errores y pérdidas de paquetes pueden ser frecuentes.

2.3.2 Mejoras al mecanismo CTCP

En [4], Perkins y Mutka introdujeron algunas mejoras al esquema CTCP. Sus esfuerzos se centraron en mejorar su robustez sacrificando para ello la tasa de compresión. Como se muestra en la Figura 2.7, los autores mejoraron la codificación delta de los paquetes adyacentes

introduciendo un marco de referencia. Un marco se compone de varios paquetes consecutivos. El primer paquete dentro de un marco de referencia se envía sin comprimir, mientras que los siguientes paquetes se envían utilizando una codificación delta utilizando como referencia el primer paquete sin comprimir, es decir, el primer paquete dentro del marco correspondiente. La ganancia de compresión es inferior a la obtenida utilizando el esquema CTCP. En el lado positivo, un paquete comprimido corrompido no resulta en la pérdida de sincronización. Esta es la mayor ventaja de este esquema sobre el esquema CTCP.

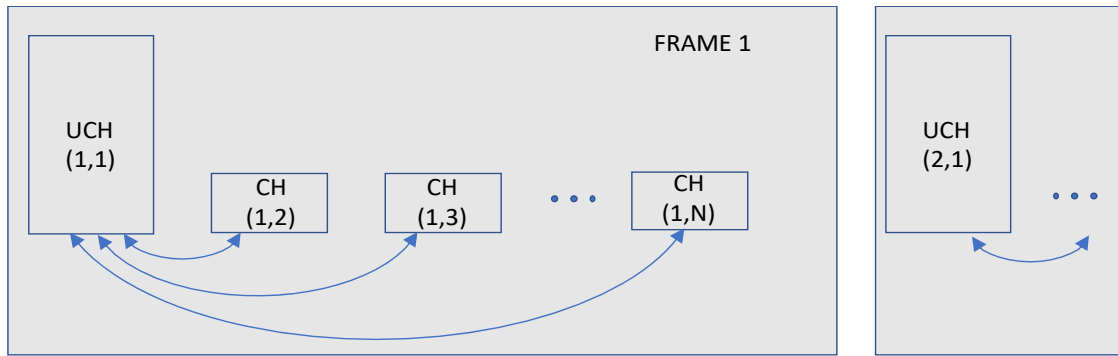


Figura 2.7 Esquema de compresión con marco de referencia [4]

En[5], Calveras et al. introdujeron una mejora a la propuesta de Perkins. El mecanismo hace uso de un marco de referencia dinámico definido teniendo en cuenta el comportamiento del canal para poder mantener la sincronización adecuada de los dos extremos, es decir, el compresor y el descompresor. En otras palabras, todos los esquemas actualizan a intervalos determinados la información compartida por el compresor y el descompresor mediante cabeceras comprimidas y no comprimidas. Sin embargo, cuando una actualización de la referencia o base se corrompe o se pierde, puede producirse una pérdida de sincronización que imposibilite la recuperación de la información en el extremo del descompresor.

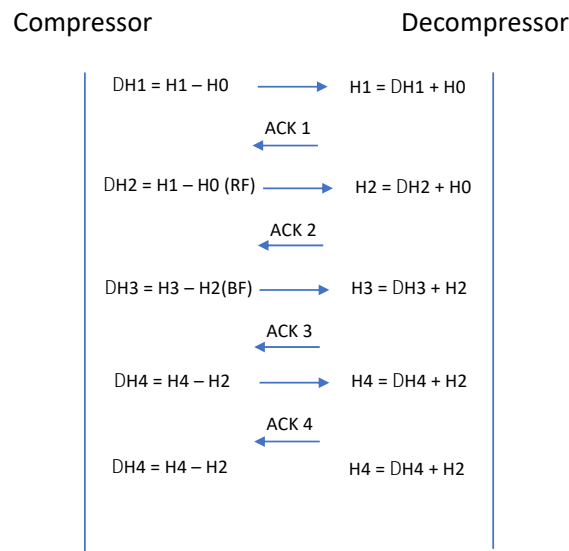


Figura 2.8 Mecanismo propuesto por Rossi et al. [6]

En [6], Rossi et al. introdujeron un nuevo esquema de modificación de actualización de base. Su propuesta no se incluyó en las normas. Sin embargo, ilustra bien los esfuerzos realizados por otros

grupos de investigación en el desarrollo de esquemas de compresión de encabezados IP. Los principios de funcionamiento de su propuesta se describen en la Figura 2.8

De acuerdo con el principio de funcionamiento de su propuesta, el compresor inicia la actualización de la base de cabecera enviando al descompresor un indicador de solicitud (RF). El procedimiento es el siguiente:

- el compresor envía una nueva base utilizando una cabecera comprimida
- el descompresor descomprime la nueva base
- el descompresor espera una indicación explícita del compresor antes de utilizar la nueva base
- el descompresor envía el número de secuencia del paquete usando un TCP SACK
- el compresor recibe el TCP SACK y espera un segundo TCP SACK. A partir de ese momento el compresor comenzará a utilizar la nueva base. El desempeño de este esquema ha sido comparado con otros esquemas por Rossi et al. en [6].

2.3.3 IP Header Compression (IPHC)

La norma RFC 2507 introduce la compresión de cabeceras IP (IPHC) que incorpora nuevas funcionalidades al mecanismo de VJHC [7]. El IPHC fue desarrollado en 1999 por un grupo de investigación de la Universidad Tecnológica de Lulea (Suecia). Entre las mejoras más relevantes, IPHC proporciona soporte a UDP e IPv6 junto con nuevas funcionalidades para comunicaciones basadas en TCP. IPHC también proporciona soporte al servicio de multidifusión basado en UDP.

Desafortunadamente, el IPHC no incluye mecanismos para el protocolo RTP, lo que hace que su uso sea inviable para muchas aplicaciones multimedia. Del mismo modo que el esquema introducido por Van Jacobson, IPHC se basa en los cambios de los valores de algunos de los campos de cabecera y también hace uso de la derivación de campos introducidos por otras capas de protocolo. El esquema de codificación de IPHC también utiliza un esquema de codificación delta, es decir, las cabeceras de paquetes comprimidos contienen los cambios en los campos de cabecera con respecto a la cabecera del paquete anterior.

Los esquemas de corrección de errores de IPHC se basan en los mecanismos de control de errores TCP. En el caso de que el protocolo TCP no se utilice en la capa de transporte, el IPHC no utiliza la codificación delta. En este caso, la relación de compresión alcanzada para UDP es peor que la obtenida para TCP.

Según las especificaciones de IPHC, los paquetes de control se intercambian periódicamente entre el compresor y el descompresor. IPHC hace uso de dos mecanismos:

- TWICE [8]. Consiste en calcular la suma de comprobación. Si el cálculo de la suma de comprobación falla, el paquete comprimido se pierde. El descompresor calcula la suma de control del siguiente paquete. Si el cálculo de la suma de comprobación del nuevo paquete es correcto, el descompresor determina el valor de la base incrementando el conteo en dos deltas.
- Solicitud de cabecera. Cuando el descompresor no puede recuperar el contexto, emite una petición al compresor para una descompresión. Esto sólo es posible cuando se trabaja en modo bidireccional.

Numerosos estudios se han llevado a cabo a fin de evaluar el rendimiento de la IPHC en aplicaciones de voz a través de enlaces alámbricos [9]. Otros estudios se han centrado en el estudio de mecanismos de compresión de cabeceras especialmente diseñados para aplicaciones móviles IPv6[10]. En el caso del esquema de compresión de encabezados IPv6, éste utiliza un mecanismo de reasignación (traducción) de direcciones. La reasignación de dirección de ICPH asigna la dirección IPv6 a identificadores más cortos utilizando una codificación Lempel-Ziv-Welch (LZW) para el primer encabezado, y se utiliza una codificación incremental para los siguientes paquetes.

2.3.4 Compressed Real Time Protocol (CRTP)

La norma RFC 2508 Compressed Real Time Protocol (CRTP) fue introducida en 1999. Su desarrollo respondió a la necesidad de la implantación de servicios audiovisuales a través de enlaces seriales de baja velocidad. La relación de compresión y el funcionamiento máximos alcanzables de CRTP dependen en gran medida de los protocolos y opciones implementados.

En cuanto a la relación de compresión, cuando la opción de suma de comprobación UDP está activada, CRTP consigue una ganancia de compresión del 90% resultante de la compresión de la cabecera IP de 40 bytes en cuatro bytes.

CRTP es particularmente útil para comprimir conjuntamente las cabeceras RTP/UDP/IP. Esto se debe principalmente a las características de los campos del protocolo RTP: los cambios de los campos de cabecera RTP que pertenecen a un flujo dado pueden predecirse utilizando los valores anteriores. Hay muchos campos que cambian usando un patrón bien determinado, por ejemplo, el número de secuencia de los paquetes puede predecirse fácilmente. Siguiendo la terminología utilizada en los estándares de compresión, esto se conoce como compresión de primer orden [27]. Este tipo de parámetros se mantienen en la estructura de datos que define el contexto general. Los campos que deben comprimirse se conocen como campos de segundo orden, por ejemplo, los saltos de fotogramas audiovisuales. Supongamos, por ejemplo, el caso de un servicio de vídeo en el que los fotogramas deben enviarse regularmente. A veces las tramas pueden no ser entregadas debido a la pérdida de sincronización. El proceso de compresión debe ajustarse al estado de primer orden.

Dado que CRTP ha sido diseñado para soportar servicios de voz y vídeo utilizando protocolos UDP/RTP, su diseño incluye protocolos de sincronización de señalización. Las especificaciones del CRTP definen 15 mensajes de control de señalización para mantener la sincronización compresor/descompresor. Varios estudios han demostrado que cuando se opera en enlaces propensos a errores, la pérdida de un paquete puede producir la pérdida de varios paquetes. El número de paquetes perdidos dependerá del retardo de ida y vuelta. Estas son las principales razones por las que no se recomienda este mecanismo para los canales inalámbricos [11].

Varias mejoras al esquema CRTP han sido introducidas en la literatura. En [12], los autores introdujeron el esquema RObust Checksum-based COmpression (ROCCO). ROCCO utiliza la misma suma de comprobación para las cabeceras comprimidas y no comprimidas. El descompresor utiliza la suma de comprobación para mantener la sincronización. Además, ROCCO define varios tipos de compresión diseñados específicamente para soportar muchas aplicaciones [13, 12]. Finalmente, el Enhanced Compressed RTP (ECRTP) [14] mejora las funcionalidades de CRTP utilizando retransmisiones locales.

2.4 Resumen de los principales estándares de compresión de cabeceras IETF

La Tabla 2.2 resume la evolución y las principales características de las normas de compresión de cabeceras IETF. Como se puede ver en la tabla, desde el primer estándar de compresión de cabeceras IETF introducido por Van Jacobson en 1990, se han desarrollado numerosos estándares. La última entrada de la tabla, el RFC 3905, corresponde a la definición del protocolo de Compresión de cabeceras Robusto (ROHC), el cual es el objeto de estudio y evaluación de este proyecto.

Como ya se ha mencionado, debido al incremento en la demanda de aplicaciones móviles haciendo uso de protocolos IP, tales como aplicaciones multimedia o servicios de voz, los protocolos de compresión de cabecera se han ampliado para cubrir la compresión de múltiples protocolos: IPv6, TCP, UDP, RTP, entre otros.

En los últimos años, el diseño y desarrollo de una compresión robusta de cabeceras de paquetes ha despertado mucho interés. Este interés se ha visto motivado por el desarrollo y el amplio despliegue de diversas tecnologías de redes de sensores inalámbricos, como las tecnologías de

comunicaciones de bajo consumo y largo alcance (LoRA, SigFox, NBIoT) utilizadas en el despliegue de un gran número de aplicaciones IoT.

Norma IETF	Cabeceras	Min. tamaño comprimido	Tipo de enlace	Codificación
			(BER, RTT)	Control de error (Feedback)
RFC 1144 (VJ,CTCP)	IPv4/TCP	2 Bytes	Dial-up (low, short)	Delta TCP based (No)
RFC 2507 (IPHC)	IPv4 (including options and fragments), IPv6 (including extension headers), AH, minimal encapsulation header, tunneled IP headers, TCP (including options), UDP, ESP	2 Bytes	Dial-up and wireless (low to medium, short to medium)	Delta TWICE (Yes)
RFC 2508 (CRTP)	IPv4, IPv6 (including extension headers), AH, Minimal Encapsulation header, Tunneled IP headers, UDP, RTP	2 Bytes	Dial-up and wireless (low to medium, short to medium)	Delta TWICE (Yes)
RFC 3095 (ROHC)	IPv4, IPv6 (including extension headers), AH, Minimal Encapsulation header, Tunneled IP headers, UDP, RTP	1 Byte	Wireless (high, long)	window-based least significant bit local repair (Yes)

Tabla 2.2 Principales estándares de compresión de cabeceras de la IETF [1]

Capítulo 3. COMPRESIÓN DE CABECERAS ROBUSTA (ROHC)

Este capítulo proporciona una descripción detallada de la Compresión Robusta del Encabezado que se proporciona a continuación [3]. También proporciona información sobre las directrices de implementación del protocolo y los mecanismos de señalización necesarios para el correcto funcionamiento del mecanismo de compresión a través de enlaces inalámbricos.

3.1 Introducción

ROHC es la norma de compresión de cabeceras propuesta por la IETF [3]. Se introdujo por primera vez en 2001 para responder a la creciente demanda de los servicios de telefonía móvil y las aplicaciones de voz sobre IP (VoIP) [1]. Uno de los principales parámetros de diseño de ROHC es el asegurar su correcto funcionamiento (robustez) en enlaces donde son comunes los errores y pérdidas de información. Para ello los diseñadores han incorporado los mecanismos necesarios para adaptar su funcionamiento ante la presencia de altas tasas de errores, en el rango de 10^{-4} - 10^{-2} , y latencias de hasta 200 ms.

El diseño de ROHC se ha visto beneficiado de los resultados obtenidos por los mecanismos de compresión descritos anteriormente. Dichos resultados han permitido a los desarrolladores y operadores de identificar tres indicadores claves de desempeño (KPIs por sus siglas en inglés). La definición de los dichos indicadores se ha tenido en cuenta considerando que los mecanismos de compresión tendrán que desplegarse en canales inalámbricos donde la tasa de errores es elevada.

- Eficiencia de compresión - Esta métrica representa la ganancia de compresión media del tamaño de la cabecera obtenido por mecanismo de compresión.
- Robustez- La robustez se refiere a la capacidad que ha de tener el mecanismo para restablecer su funcionamiento ante la presencia de pérdidas de paquetes. En otras palabras, el protocolo no debe producir pérdidas adicionales de paquetes debido a la pérdida de un paquete.
- Transparencia de la compresión- El mecanismo no debe modificar el significado de los campos de la cabecera. Un mecanismo es transparente si todos los campos de las cabeceras descomprimidas coinciden exactamente con los campos de cabecera originales.

El principal objetivo de diseño de un protocolo de compresión de cabeceras robusto debe tener en cuenta las características mencionadas anteriormente. Su diseño debe hacerse manteniendo un equilibrio entre el nivel de la robustez y la eficiencia de la compresión deseada. Además, el diseño debe tener en cuenta la gran diversidad de las los diferentes aplicaciones y arquitectura de protocolos existentes, así como la evolución futura y prevista en el ámbito de las telecomunicaciones.

ROHC incorpora algunos de los principios de funcionamiento introducidos por el mecanismo ROCCO [15]. El estándar RFC 3095 define los principios de funcionamiento de ROHC donde se han incluido tres familias de protocolo: RTP/UDP/IP, UDP/IP y ESP/IP. Dentro de la terminología ROHC, los mecanismos de compresión correspondientes a cada familia de protocolos se denomina perfil.

La norma ROHC incluye además ambas versiones del protocolo IP, es decir, IPv4 e IPv6. Un cuarto perfil *sin comprimir* ha sido incluido a fin de permitir la interoperabilidad de las implementaciones habilitadas para ROHC. De esta manera, cuando ROHC detecta un paquete IP que incluye campos que pertenecen a otros protocolos no incluidos en los perfiles anteriores, no realiza ninguna compresión. En la actualidad, se vienen realizando esfuerzos para incluir más protocolos, ver referencias [16, 17, 18, 19, 20, 21, 22].

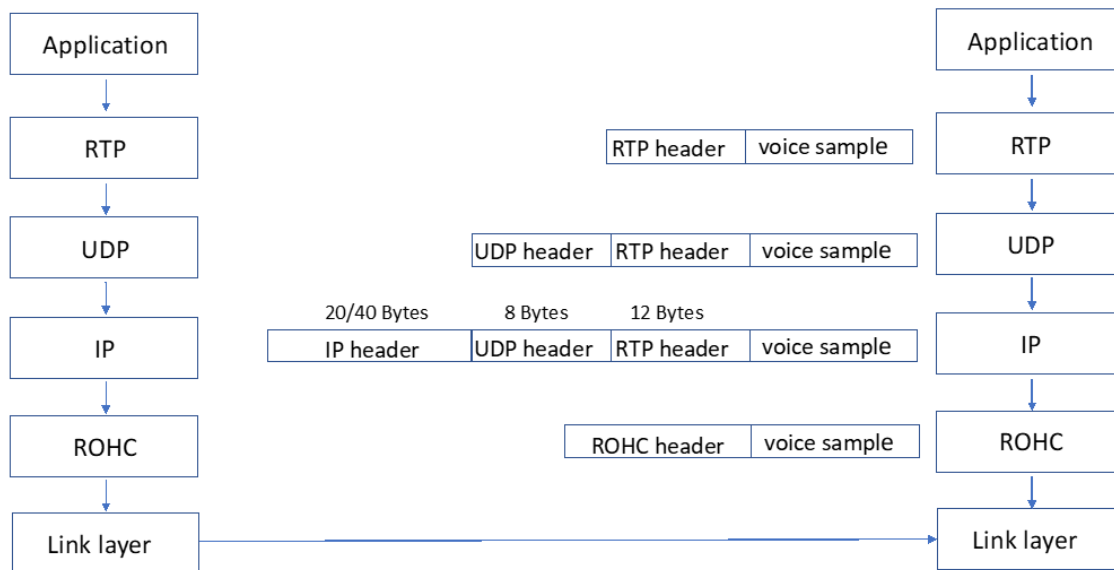


Figura 3.1 ROHC dentro de la pila de protocolos TCP/IP

Tal y como se muestra en la Figura 3.1, ROHC se sitúa entre la capa de red y la capa de enlace, es decir la compresión ROHC se lleva a cabo justo por encima del nivel de enlace por lo que el funcionamiento de las capas por encima de la capa de enlace permanece inalterado. El compresor ROHC se implementa en el emisor inalámbrico y el descompresor en el receptor. Las principales funciones de ROHC son:

1. Una reducción sustancial del tamaño de las cabeceras de uno o más de los protocolos por encima de la capa de enlace, remplazándolas por una cabecera ROHC de menor tamaño.
2. El uso del mecanismo de compresión ROHC no debe comprometer la calidad de servicio ofrecida a las aplicaciones de usuario. Esta característica es de particular importancia para aplicaciones multimedia (voz/vídeo).

El compresor ROHC debe poder configurarse de tal modo que en ocasiones puede limitar la tasa de compresión con el fin de asegurar una mayor robustez en presencia de errores. Para ello, ROHC proporciona diferentes niveles de compresión que pueden activarse dinámicamente dependiendo de las tasas de error del canal. El compresor y el descompresor pueden alternar entre diferentes niveles (estados) de compresión.

3.2 Contexto y Estados

La compresión de cabeceras del paquete se realiza asociando flujos de paquetes de datos pertenecientes a una comunicación dada a través de un enlace inalámbrico. Esto se puede realizar gracias a que los paquetes pertenecientes aun flujo comparten ciertos parámetros, como por ejemplo las direcciones IP o puertos de la capa de transporte. En otras palabras, los paquetes que transitan por un enlace inalámbrico remitente/receptor pueden agruparse.

Un ejemplo típico de tal escenario es la secuencia de paquetes transportando el audio asociada a otra secuencia de paquetes que transmiten el vídeo. asociado En tal escenario, ROHC puede comprimir las cabeceras de los paquetes explotando las redundancias encontradas en los diversos campos de cabecera.

De acuerdo con el funcionamiento de ROHC, cada flujo se identifica mediante un identificador de contexto (CID por sus siglas en inglés) a que debe ser el mismo a ambos lados de la

comunicación, véase la Figura 3.2. Un contexto es un conjunto de datos utilizados para determinar el estado de la comunicación entre el compresor y el descompresor.

ROHC define tres conjuntos de datos:

- parámetros por canal.
- parámetros por contexto y perfiles.
- contextos e identificadores de contexto.

La compresión se lleva a cabo mediante un conjunto de paquetes ROHC cuyo formato y funciones han sido definidos por la norma del protocolo ROHC [3].

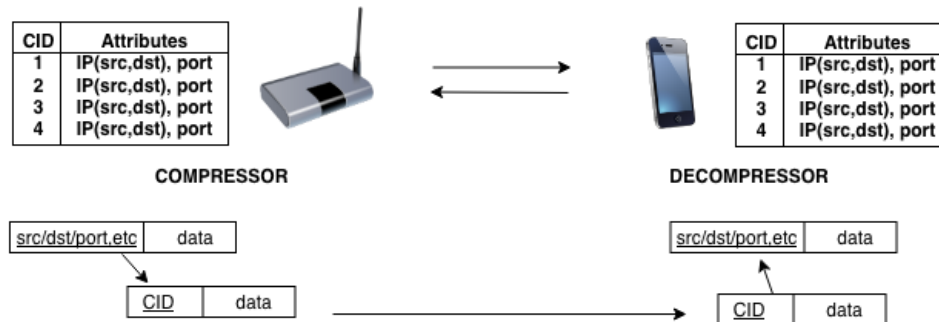


Figura 3.2. Contexto ROHC

3.2.1 Parámetros de canal

La norma ROHC especifica los parámetros globales a ser utilizados por todos los flujos de paquetes ROHC transmitidos a través de un canal determinado. Estos parámetros se establecen en el momento de la inicialización del canal. Para asegurar el correcto funcionamiento del esquema de compresión, el compresor y el descompresor que operan sobre el canal deben utilizar el mismo conjunto de parámetros. La Tabla 3.1 muestra los principales parámetros del canal.

Parámetro	Descripción
MAX_CID	Este parámetro define los números de identificación de contexto máximos. Se especifica como un número entero no negativo cuyo valor debe ser menor o igual al valor máximo permitido por el parámetro LARGE_CID.
LARGE_CIDS	Booleano. Si falso indica un CID corto, valores entre 0 y 15, en caso contrario es un CID grande, valores entre 0 y 16383.
PROFILES	Un conjunto de números enteros no negativos, correspondientes a diferentes perfiles. Cada perfil define las cabeceras del protocolo a comprimir.
FEEDBACK_FOR	El compresor y el descompresor utilizan el canal de realimentación para controlar los flujos de paquetes comprimidos que se ajustan sobre el canal.

Tabla 3.1 Parámetros de canal [2]

3.2.2 Perfiles

La norma RFC 3095 describe los cuatro perfiles siguientes, véase Tabla 3.2. En el momento de la inicialización de una comunicación ROHC, el compresor debe indicar al descompresor el perfil a ser utilizado durante la comunicación.

Perfil	Descripción
0x0000	uncompressed IP packets.
0x0001	RTP/UDP/IP compression.
0x0002	UDP/IP compression.
0x0003	is for ESP/IP compression.

Tabla 3.2 Perfiles

3.2.3 Contextos e identificadores de contexto

Los contextos se identifican mediante un identificador de contexto, CID, que se envía en las cabeceras de todos los paquetes ROHC a fin de identificar el contexto asociado.

El número CID utilizado en un canal es independiente del usado en otro canal. Por ejemplo, el CID 2 sobre el canal B y CID 2 sobre el canal D no se refieren al mismo contexto, incluso en el caso en que los sistemas (nodos) finales de los canales B y D sean los mismos. Este escenario puede surgir cuando los dispositivos inalámbricos utilizan diferentes tecnologías de transmisión o utilizan diferentes bandas de frecuencia, por ejemplo, las bandas WiFi de 2,4 GHz y 5 GHz.

La información de un contexto dado se almacena en una estructura de datos. La estructura de datos se indexa mediante el CID, que debe incluirse siempre en las cabeceras comprimidas. El tamaño del espacio CID puede ser pequeño o grande como se describió anteriormente.

El CID con valor 0 está representado por la ausencia del campo. El pequeño CID con valores que van de 1 a 15 ocupa cuatro bits. La codificación de un CID grande es más compleja y se describe en la norma RFC 3095 [2, sección 4.5.6].

3.3 Diagrama de estados del compresor

La Figura 3.3 muestra el diagrama de estados del compresor ROHC:

- Inicialización y estado de actualización (IR): El compresor inicia su funcionamiento en estado IR. El compresor y el descompresor deben pasar a un estado superior lo antes posible para activar el mecanismo de compresión. La transición del estado IR al estado FO tiene lugar tan pronto como el compresor establece con éxito un contexto de compresión con el descompresor. En caso de fallas repetidas, el compresor puede pasar del estado de FO al estado IR.
- Estado de primer orden (FO): El estado se alcanza desde el estado IR como resultado de establecer el contexto. Si se produce un error en el proceso de decodificación de un paquete, el compresor vuelve al estado de FO. En caso de error el compresor pasa del estado SO al estado FO.
- Estado de segundo orden (SO): En el estado SO, el compresor sólo es necesario para transmitir el número de secuencia de paquetes. El descompresor puede derivar los valores de todos los demás campos de cabecera utilizando la información disponible en el contexto preestablecido.

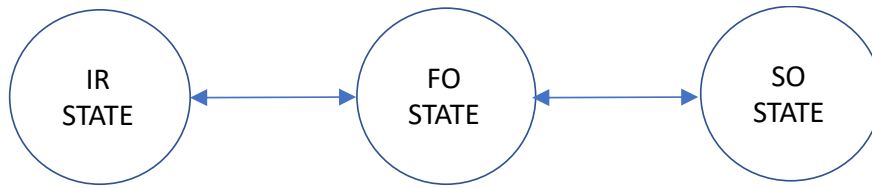


Figura 3.3 Diagrama de estados del Compresor [2]

3.4 Diagrama de estados del descompresor

La Figura 3.4 muestra el diagrama de estado del descompresor. Los nombres de los estados se refieren estado del contexto que mantiene el descompresor. El funcionamiento del descompresor en cada uno de los tres estados puede describirse de la siguiente manera:

- El estado Sin contexto (NC) denota el hecho de que el descompresor no cuenta con la información necesaria para crear el contexto. Este estado representa la condición inicial del descompresor. Este estado también puede surgir ante la pérdida de la información por parte del descompresor.
- El estado estático-contexto (estado SC) indica que el descompresor carece de todos o parte de los elementos dinámicos. Para pasar del estado SC al estado de contexto dinámico, el descompresor debe recibir de los paquetes de compresor que contienen los campos de cabecera dinámicos. Por el contrario, la transición del estado Contexto Dinámico al estado SC puede ocurrir por causa de errores de transmisión de paquetes o ante la imposibilidad de mantener la sincronización entre el compresor y el descompresor.
- El estado de contexto dinámico (estado DC) indica que el descompresor cuenta con todo el contexto.

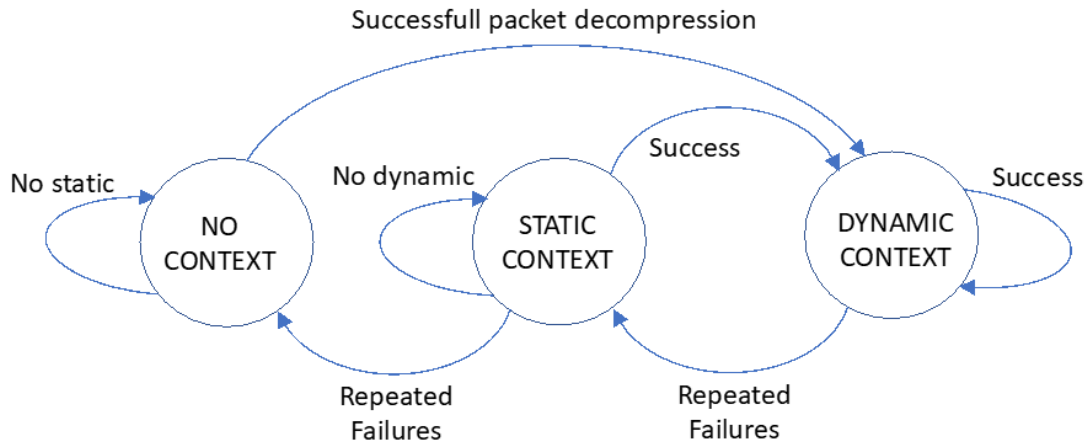


Figura 3.4 Diagrama de estados del descompresor

3.5 Modos de Transmisión

La norma ROHC define tres modos: Modo Unidireccional, Bidireccional Optimista y Bidireccional Confiable.

ROHC siempre inicia su funcionamiento en el modo unidireccional y dependiendo de las características del enlace, puede cambiar su funcionamiento a uno de los dos modos direccionales, véase la Figura 3.5. Los modos se han definido teniendo en cuenta las características del compresor y el descompresor. Por ejemplo, en el modo Unidireccional, solo el compresor puede hacer el envío de paquetes ROHC. En el caso del modo bidireccional fiable, el descompresor y el compresor intercambiarán paquetes ROHC. El descompresor puede en este caso informar al compresor el estado en que se encuentra.

- Unidirectional Mode (U-Mode).** Este modo se ha definido a fin de implementar ROHC en enlaces sin canal de retroalimentación. En este caso, el compresor es incapaz de identificar si el descompresor ha podido establecer o mantener correctamente el contexto. El compresor sólo puede asumir de forma optimista que el descompresor cuenta con todo lo necesario para llevar a cabo la descompresión.

Con el fin de garantizar la correcta operación (robustez) del proceso de compresión, el compresor debe actualizar y corregir periódicamente el contexto. En este caso, el compresor regresa al estado IR, como se muestra en la Figura 3.6. El descompresor utiliza los paquetes IR recibidos del compresor para verificar y corregir el contexto de compresión. Este procedimiento es una de las mejoras de robustez frente a errores introducida por el mecanismo ROHC. El U-mode es sin embargo el menos eficiente de los tres modos ROHC.

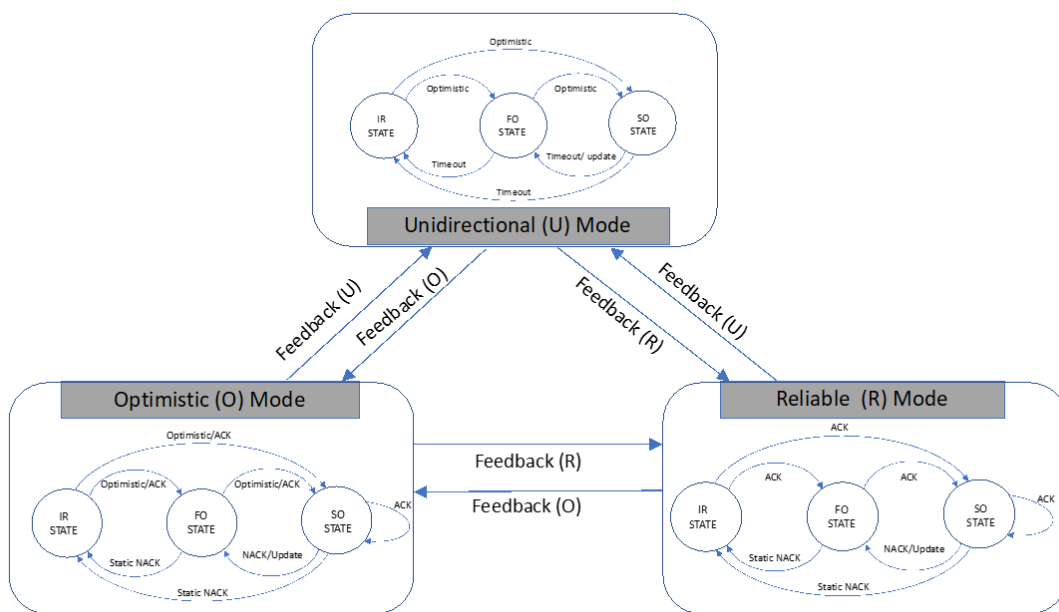


Figura 3.5 Transiciones entre Modos

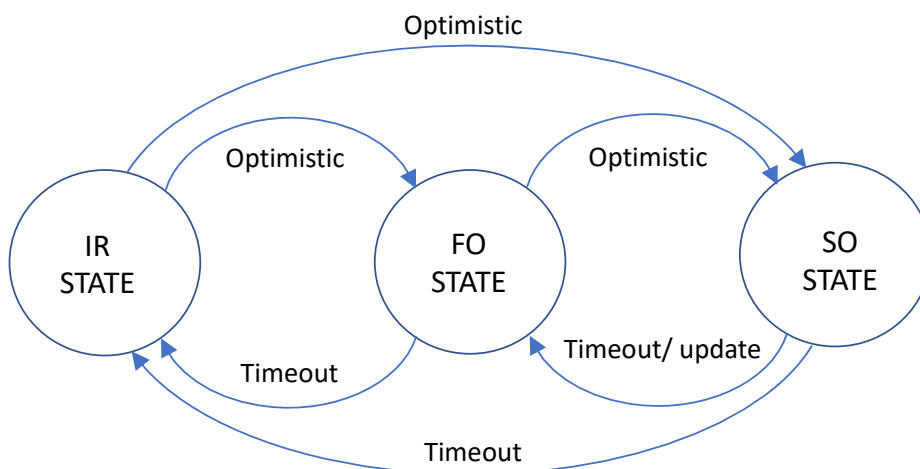


Figura 3.6 Transiciones del compresor - Unidireccional mode [2].

- Bidirectional Optimistic Mode (O-mode).** La Figura 3.7 muestra el diagrama de transiciones entre estados del compresor cuando éste funciona en el modo optimista bidireccional. Cuando la compresión funciona en modo O-Mode, el descompresor puede utilizar paquetes ROHC de retroalimentación al compresor a fin de informarle su estado. En el O-mode, el descompresor puede emitir paquetes ACK para notificar al compresor de la correcta recepción de los paquetes ROHC. De acuerdo con las especificaciones de ROHC, el uso de ACKs emitidos por el descompresor son opcionales. Por lo que es posible que el compresor necesite implementar las mismas transiciones que en el modo de operación U-Mode.

Dado que este modo de operación ha sido concebido para ser utilizado en enlaces con bajas probabilidades de error, los datos de actualización de contexto se envían utilizando un código de redundancia cíclico de 3 bits. Sin embargo, en el caso de aplicaciones sensibles a errores, el uso del modo bidireccional fiable es una mejor opción.

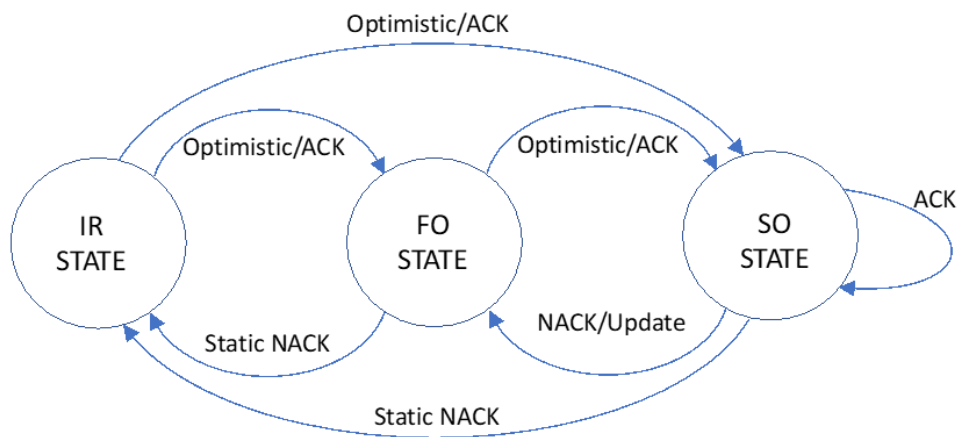


Figura 3.7 Transiciones del compresor - Bidirectional Optimistic mode [2]

- Bidirectional Reliable Mode (R-mode).** Este modo es más robusto. En este caso, el compresor pasa a los estados FO u SO sólo después de recibir un ACK del descompresor, como se muestra en la Figura 3.8. El compresor vuelve a la inicial al recibir un NACK. Bajo este modo, el compresor y el descompresor colaboran estrechamente para mantener una correcta sincronización.

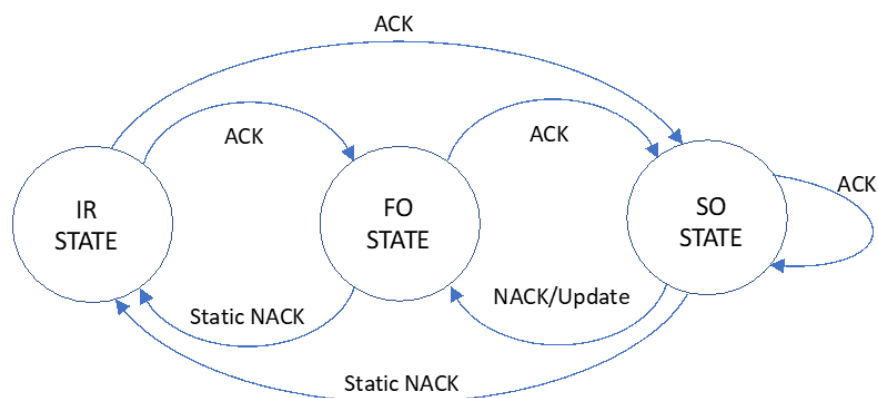


Figura 3.8 Transiciones del compresor - Bidirectional Reliable mode [2]

3.6 Métodos de Codificación

El proceso de compresión ROHC de la parte estática de los encabezados utiliza un mecanismo similar al que utilizan otros mecanismos de compresión de cabeceras. Los campos de cabecera que no se modifican sólo se transmiten en el momento del establecimiento del contexto y permanecen constantes a partir de ese momento. ROHC puede determinar los valores de cabecera dinámicos utilizando una función lineal a partir del número de secuencia de cada paquete. Si los campos dinámicos del encabezado cambian aleatoriamente, el compresor debe comunicarlo al descompresor.

ROHC utiliza principalmente los siguientes dos métodos:

- Codificación Bits Menos Significativos (LSB). Este algoritmo se utiliza para reducir el número de bits necesarios para representar un valor absoluto. Se utiliza para campos de cabecera cuyos valores suelen estar sujetos a pequeñas modificaciones. En lugar de transmitir el valor del campo original, sólo se transmiten los bits menos significativos del valor k , donde k es un número entero positivo. Al recibir los k bits, el descompresor calcula el valor absoluto del campo utilizando como referencia un valor recibido previamente, denotado por v_{ref} . A fin de garantizar el correcto funcionamiento de la codificación LSB, ambos extremos deben utilizar: 1) el mismo valor original y 2) el mismo número de k bits menos significativos correspondientes a los k bits, véase la Figura 3.9

El intervalo de referencia se defina de la siguiente manera:

$$f(v_{ref}, k) = [v_{ref} - p, v_{ref} + (2^k - 1) - p] \quad (2.1)$$

donde p es un entero.

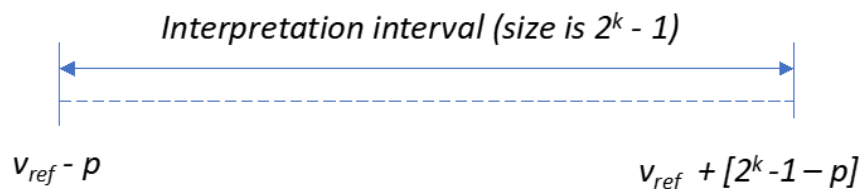


Figura 3.9 Intervalo utilizado por la codificación LSB [2]

La función $f(v_{ref}, k)$ debe satisfacer la siguiente propiedad: para cualquier k , los k bits k menos significativos identificarán unívocamente un valor en $f(v_{ref}, k)$. El parámetro p se utiliza para desplazar el intervalo de interpretación con respecto a v_{ref} en función de los diferentes campos de cabecera de los protocolos, como el campo ID del protocolo IP o el campo SN del protocolo RTP.

- Codificación Bits Menos Significativos con ventanas (W-LSB). Los valores que cambian dinámicamente suelen tener un comportamiento dinámico determinista, por ejemplo, siempre aumentando en uno para los números de secuencia. Una vez identificado el patrón dinámico, se construye una ventana alrededor de un valor de referencia. Dependiendo de la distancia entre el nuevo valor y el valor de referencia y de la posición relativa de la ventana respecto al valor de referencia, se determina el número de bits para transmitir el nuevo valor comprimido. Los bits indican el desplazamiento a partir de un valor de referencia: un número más pequeño utilizado para transmitir los valores del campo de encabezado que no cambian aleatoriamente, sino que aumentan o disminuyen

continuamente siguiendo un patrón regular. Este algoritmo se beneficia del patrón regular que describe el comportamiento de varios campos de encabezado comúnmente utilizados en los protocolos de comunicación. El uso de una ventana resulta en una reducción en el número medio de bits requeridos para describir la secuencia de valores de ciertos campos de la cabecera. El algoritmo de compresión W-LSB en combinación con un elaborado esquema de protección para datos sensibles en cabeceras comprimidas con ROHC contribuye a la robustez de ROHC.

3.7 Compresión RTP

El diseño de la norma ROHC aplicada al protocolo RTP ha recibido especial atención. El protocolo RTP es ampliamente utilizado en la pila de protocolos de los servicios de voz por lo que la compresión del RTP puede traer grandes beneficios: las muestras de voz se caracterizan por su tamaño corto.

La marca de tiempo es uno de los principales parámetros del protocolo RTP. Esta marca de tiempo RTP se determina en función de las características del canal. Por ejemplo, en el caso que una comunicación de voz genera muestras de voz a intervalos regulares utilizando el protocolo RTP. En este caso, los paquetes RTP se pasan en tiempo real al compresor ROHC. Por consiguiente, la diferencia de tiempo entre dos paquetes RTP consecutivos es la diferencia de los valores de la marca de tiempo en los dos paquetes. Asumiendo que la transmisión de los paquetes de voz muestra una baja fluctuación de retardo, el descompresor puede usar la diferencia de tiempo entre los tiempos de llegada de los dos paquetes comprimidos para estimar el nuevo valor de la marca de tiempo RTP basado en los valores anteriores. Bajo estas condiciones, el compresor puede calcular el número de bits necesarios para estimar la marca de tiempo RTP. De hecho, el compresor sólo necesita enviar un número reducido de bits para mejorar la estimación del descompresor [9].

3.8 Paquetes ROHC

Esta sección describe los distintos tipos de paquetes definidos por la norma ROHC. De acuerdo con la norma, el número de tamaños de paquetes se ha reducido al mínimo. Los diferentes tipos de paquetes definidos por el estándar pueden agruparse en dos grupos:

- Paquetes tipo ROHC. Estos paquetes proporcionan los medios para que el compresor transmita al descompresor la información necesaria para la creación y mantenimiento de los contextos. También se utilizan para transportar los datos con las cabeceras comprimidas.
- Paquetes de retroalimentación de ROHC. El descompresor debe ser capaz de proporcionar información de retroalimentación al compresor utilizando diferentes tipos de paquetes. El compresor mantiene o cambia su estado y/o modo de transmisión basándose en la retroalimentación proporcionada por el descompresor.

3.8.1 Tipos de paquetes ROHC

La figura 3.10 muestra el formato general de los paquetes ROHC. El primer octeto, llamado ADD-CID, se divide en dos partes. Los primeros cuatro bits llevan el código 0xE. Si los valores de los últimos cuatro bits corresponden a un valor entre 0x1y 0xF, indican los CIDs 1 a 15. Este valor es usado cuando se usa la opción CID pequeña. En el caso de que se prefiera la opción de CID grande, estos últimos cuatro bits se usa el código 0x0 y el número de CID se especifica en uno o dos octetos, véase la Figura 3.10.

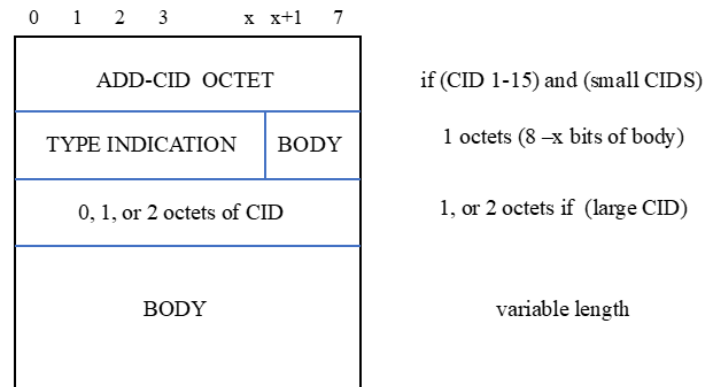


Figura 3.10 Formato de paquete ROHC type

Profile Identifier	Description
IR packet	The decompressor creates the context when receiving this kind of packets. IR packets always convey the static part the context while the dynamic part may or may not be present. Used in FO state
IR-DYN packet	These packets convey the dynamic part updates of the context. Used in FO state.
Type 0 packets (R-0, R-0-CRC and UO-0)	The information contained in these packets are used by the decompressor to update RTP-SN. They consist on one or two octets.
Type 1 packets – Reliable mode (R-1, R-1-TS and R-1-ID)	These packets are used when the compression process is operating in the reliable mode. They do not update the context.
Type 1 packets – U/O-Mode (UO-1, UO-1-ID and UO-1-TS)	These packets are used when the compression process is operating in unidirectional or bidirectional optimistic mode. UO-1 and UO-1-TS packets used by RTP and UO-1-ID use for IP-ID.
Type 2 packets (UOR-2, UOR-2-ID, UOR-2-TS)	These packets are used to encode the RTP-SN. Headers of packets transferred using packet Type 2 can be used as references for subsequent decompression.

Tabla 3.3 Tipos de paquetes ROHC [2]

Además de la información contenida en los paquetes de Tipo 1 y Tipo 2, pueden añadirse campos de extensión. Los paquetes tipo mostrado en la tabla se utilizan cuando el mecanismo de compresión funciona con la tasa de compresión más alta. Se pueden encontrar más detalles para cada perfil en la norma [2].

3.8.2 Paquetes ROHC feedback

Estos paquetes son utilizados por el descompresor en los modos de operación bidireccional. En la Figura 3.11 se muestra el formato general de los paquetes de retroalimentación de ROHC. Los dos tipos siguientes de paquetes de retroalimentación definidos por el estándar son los siguientes:

- Feedback Type 1 packets. Se utilizan para enviar ACK al compresor a fin de confirmar la descompresión exitosa de un paquete, haciendo saber al compresor que el contexto está actualizado.

- Feedback Type 2 packets. Se utilizan para enviar NACK o STATIC-NACK junto con más información. Un NACK indica que el contexto dinámico del descompresor no está sincronizado. El NACK se emite tras la descompresión infructuosa de varios paquetes comprimidos sucesivamente. Un STATIC-NACK indica que el contexto estático del descompresor no es válido o no ha sido establecido. La norma define nueve opciones de retroalimentación diferentes que permiten la transmisión de información de retroalimentación adicional al compresor.

Puede encontrar más información sobre los diferentes tipos de paquetes en el documento RPC3095[3]. La siguiente sección describe el uso de los diferentes paquetes en las diferentes fases de funcionamiento del esquema ROHC.

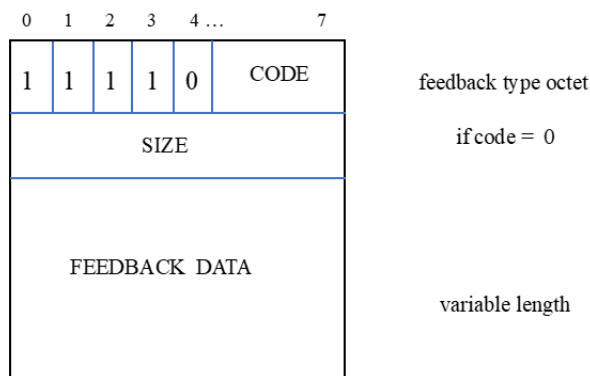


Figura 3.11 Formato del paquete ROHC feedback Type 2

3.9 El protocolo ROHC

Esta sección describe la operación del protocolo ROHC. La descripción incluye el intercambio de los diversos paquetes ROHC utilizados por el compresor y descompresor ROHC. La descripción está organizada en subsecciones correspondientes a los tres estados de funcionamiento del compresor (estado de protocolo de inicialización y actualización, estado de compresión de primer orden y estado de compresión de segundo orden), mecanismo de realimentación y transiciones de modo.

3.9.1 Estado de inicialización y reinicialización

La Figura 3.12 muestra el procedimiento de configuración del contexto e intercambio de paquetes comprimidos entre dos nodos. El establecimiento del canal se lleva a cabo en el paso 0, es decir antes de cualquier intercambio de paquetes ROHC. El compresor primeramente configura localmente el contexto. A partir de este momento el compresor inicia la comunicación con el descompresor.

En el paso 1, el compresor envía un paquete IR utilizando el U-mode donde incluye el CID y las cabeceras IP no comprimidas pertenecientes al flujo de paquetes.

En el paso 2, el descompresor registra e inicializa el contexto usando para ello el nuevo CID. En este momento, el descompresor cambia su estado de sin contexto (NC) en modo U al estado de contexto completo (FC) en modo U. En este modo de operación, el descompresor no puede proporcionar información de retroalimentación al compresor.

En el paso 3, el compresor cambia su modo de operación al modo de operación unidireccional ROHC y sigue enviando paquetes ROHC. El compresor puede decidir enviar a intervalos irregulares paquetes IR o IR-DYN. De esta manera, el descompresor se asegura de que el compresor mantenga correctamente el contexto.

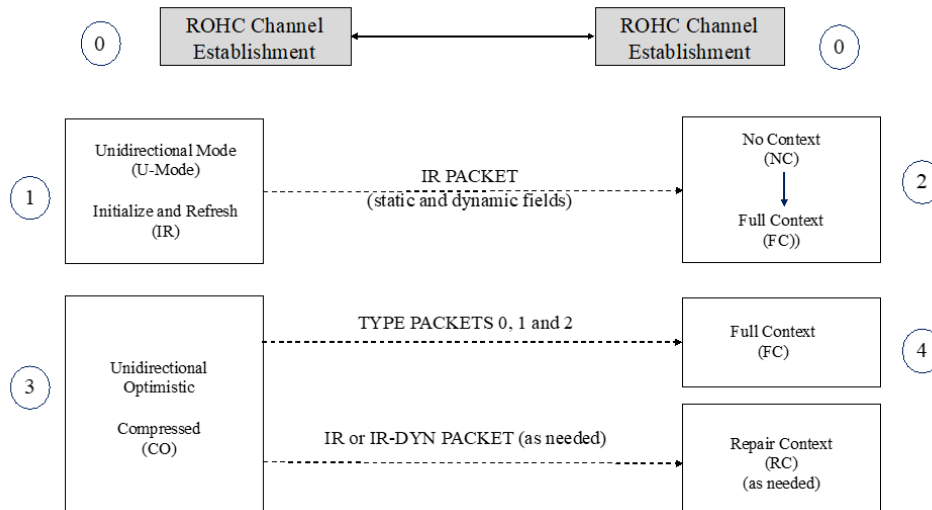


Figura 3.12 Protocolo ROHC en modo unidireccional

En el caso de que la capa física permita un canal de retroalimentación, el receptor (descompresor) puede enviar paquetes de retroalimentación al compresor. La Figura 3.13 muestra el caso en que el descompresor envía un ACK en respuesta al IR enviado por el compresor.

En el paso 3, el compresor puede enviar paquetes de ROHC asumiendo que el descompresor mantiene correctamente el contexto.

En el paso 4, dado que el descompresor mantiene correctamente el contexto es capaz de recibir y descomprimir correctamente los paquetes ROHC enviados por el compresor utilizando para ello la información almacenada en el contexto.

En el caso que el descompresor sea incapaz de descomprimir correctamente, éste emite un paquete NACK (Tipo 2) solicitando al compresor la retransmisión de un paquete IR. Al recibir el paquete de realimentación NACK, el compresor responde con un paquete IR o IR DYN ROHC. De esta manera, el descompresor puede restablecer el contexto.

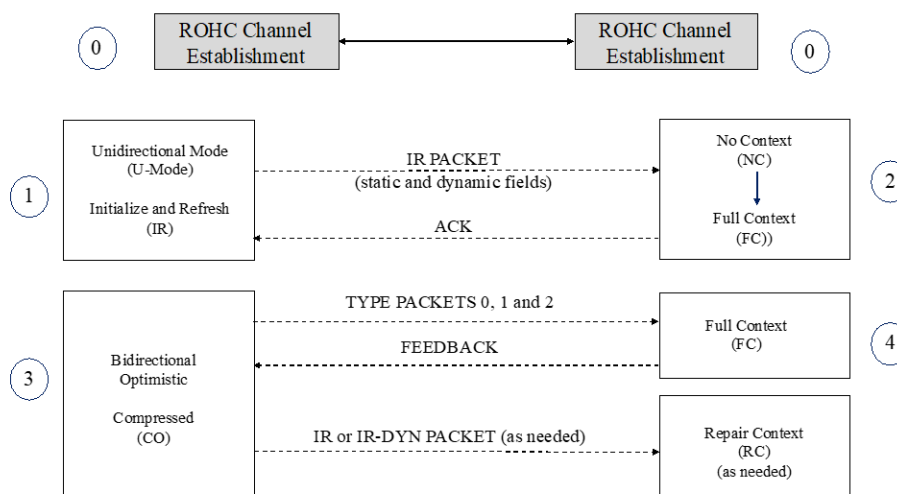


Figura 3.13 ROHC state transition in bidirectional mode

La Figura 3.14 y la Figura 3.15 muestran los formatos de los paquetes IR e IR-DYN ROHC. Como muestra la figura, los formatos de ambos paquetes son muy similares. Aparte de los octetos de tipo paquete, el contenido de estos dos tipos de paquetes difiere en la información que debe incluirse en la información específica del perfil. En el caso del paquete IR ROHC, los campos de información específicos del perfil contienen la parte estática del contexto. El paquete IR RHOC se utiliza en el momento de establecer el contexto o cuando se requiere una reparación importante. Un paquete IR RHOC puede opcionalmente llevar la parte dinámica del contexto.

En cuanto al paquete IR-DYN ROHC, se utilizan para transmitir la información necesaria para mantener actualizado el contexto. Como por ejemplo, la Figura 3.16 muestra las partes estáticas y dinámicas de una cabecera IPv4. Como se puede ver en la figura, la versión, el protocolo y las direcciones de origen y destino forman parte de la parte estática. Los campos restantes se modifican con el tiempo. La norma ROHC especifica las partes estáticas y dinámicas de las cabeceras de los diferentes protocolos soportados, es decir, todas aquellas que forman parte de uno o más perfiles ROHC.

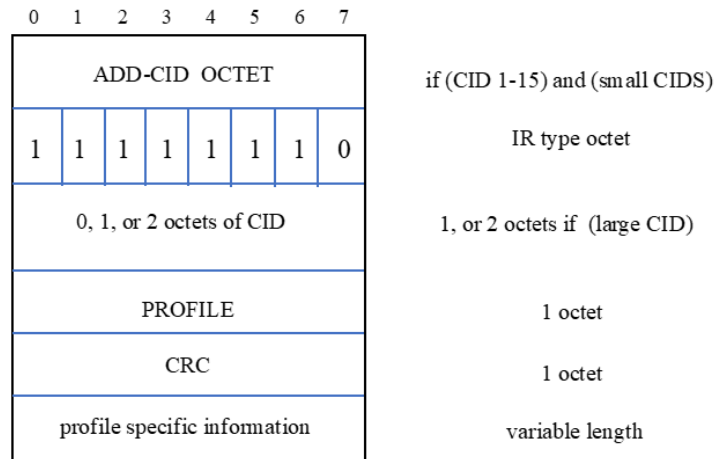


Figura 3.14 Formato del paquete IR

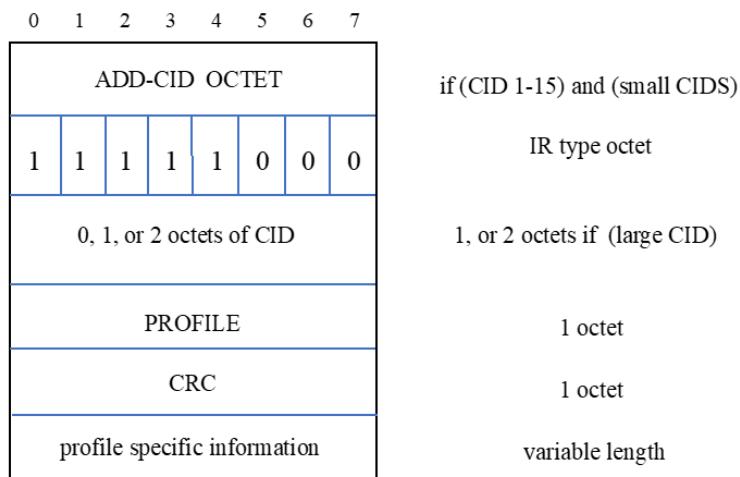


Figura 3.15 Formato del paquete IR DYN

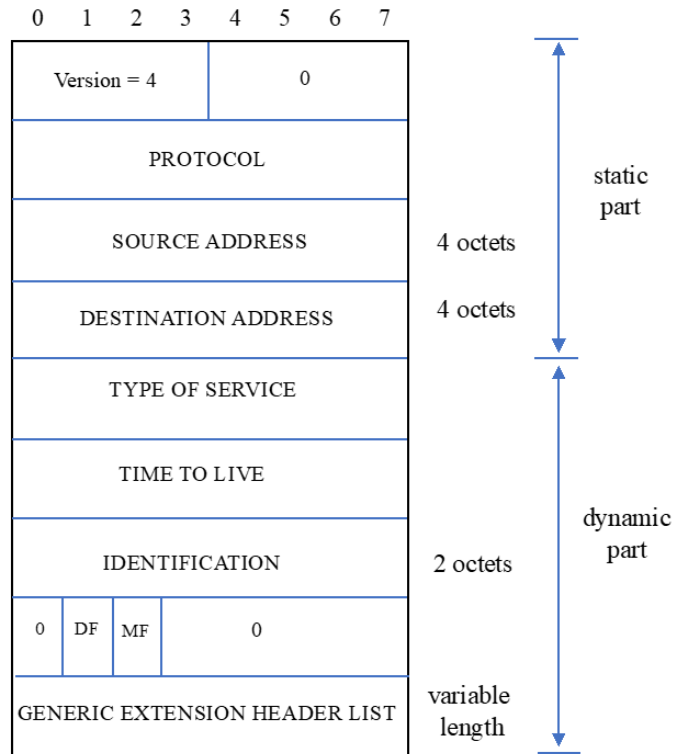


Figura 3.16 Campos estáticos y dinámico de la cabecera IPv4

3.9.2 Estado de compresión de primer orden

Una vez establecido el contexto, el compresor pasa al estado de Primer Orden (FO). El compresor asume que las partes estáticas ya han sido guardadas en el contexto del descompresor, lo que permite al compresor eliminar la parte estática de la cabecera del mensaje. El nombre de compresión de primer orden proviene del hecho de que es la etapa en la que la parte estática no requiere ser incluida en la cabecera del paquete. Si el descompresor descomprime con éxito el contenido del paquete RHOC IR DYN, se dice que el descompresor se encuentra en el estado de contexto completo (FC).

La parte dinámica contiene los campos dinámicos de las cabeceras específicas del perfil. Los campos dinámicos cambian con frecuencia en la cabecera original y pueden cambiar de forma imprevista en todo el contexto de comunicación. La cadena dinámica puede actualizarse en cualquier momento, lo que ha llevado al marco de trabajo de ROHC a permitir que los paquetes de segundo orden actualicen las variables dinámicas para maximizar el índice de compresión, por ejemplo, los paquetes ROHC Tipo 2.

3.9.3 Estado de compresión de Segundo Orden

En este estado, el compresor asume que el contexto del descompresor tiene toda la información necesaria para descomprimir con éxito las cabeceras de los paquetes y recuperar los datos.

La figura 3.17 muestra a modo de ejemplo el formato general de los paquetes R0-CRC y R0 utilizados por el compresor. El primer paquete se utiliza para actualizar el número de secuencia RTP (SN) de una secuencia de paquetes. El segundo no debe ser utilizado por el descompresor para actualizar el contexto. El objetivo de este paquete es solo el de informar al descompresor del número de secuencia actual de la secuencia de paquetes. Como se puede ver en la figura, el primer paquete contiene un campo CRC. De esta manera, al recibir un R0-CRC, el descompresor puede

verificar su exactitud. En este caso, si el descompresor detecta un error en el paquete no actualiza el contexto. En caso contrario, procede a actualizar el SN almacenado en el contexto.

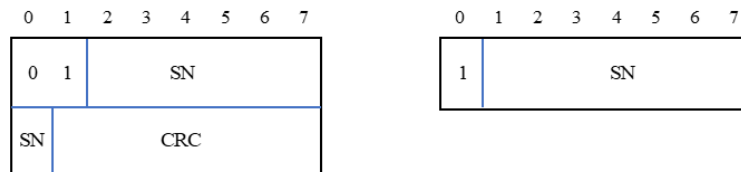


Figura 3.17 Paquetes R0-CRC y RO

3.9.4 Transición entre modos

Un compresor ROHC siempre inicia su funcionamiento en U-mode. Las reglas que definen las transiciones del modo de operación dependen del modo inicial en el que se encuentre el compresor.

Así en el caso de la transición del U-mode al O-mode, el compresor sólo necesita descomprimir con éxito un único segmento de realimentación en O-mode. Todas las demás transiciones conllevan la ejecución del intercambio de paquetes de control entre el compresor y el descompresor. La Figura 3.18 muestra el procedimiento seguido en la operación de transición del modo fiable al modo optimista.

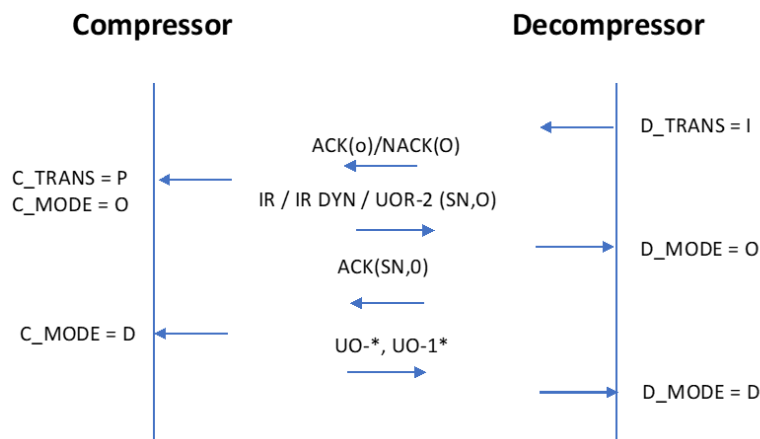


Figura 3.18 Cambio de modo

El procedimiento se inicia con un paquete de retroalimentación emitido por el descompresor. El compresor sigue enviando paquetes IR/IR_DYN/DYN o UO-*, UO-1* hasta que el descompresor logra descomprimir con éxito un paquete. El descompresor responde con un ACK que incluye el número de secuencia del paquete descomprimido. El compresor puede pasar al nuevo modo. Si el descompresor vuelve a descomprimir otro paquete con éxito, tanto el compresor como el descompresor pueden funcionar bajo el mismo modo negociado.

3.10 Estándares ROHCv2 y ROHC TCP

Desde sus orígenes la norma RFC 3095 fue concebida como un marco de referencia, permitiendo la incorporación de extensiones. Además de los perfiles ya especificados para comprimir las cabeceras IP/UDP/RTP, IP/UDP e IP/ESP, el IETF ha llevado a cabo dos importantes trabajos. La introducción de ROHCv2 cuyo objetivo ha sido mejorar la robustez de los mecanismos de compresión y la definición del perfil TCP [23][24].

3.10.1 ROHCv2

En el año 2008, el IETF publicó una segunda versión mejorada de ROHC. La nueva versión, conocida como ROHCv2, introduce una serie de simplificaciones en las reglas y algoritmos de compresión [23].

ROHCv2 incluye el perfil sólo IP destinado a ser utilizado con diferentes protocolos encapsulados utilizando las cabeceras IP. La compresión de la cabecera IP puede reducir el tamaño de las cabeceras hasta un cierto nivel. Como modificación a los perfiles ya existentes para la compresión de las cabeceras IP/UDP e IP/UDP/RTP, se proponen dos nuevos perfiles que especifican la compresión para las cabeceras IP/UDP-lite e IP/UDP-lite/RTP.

Un ejemplo de las mejoras introducidas es el caso del protocolo UDP. En el UDP estándar, la carga útil UDP puede hacer uso o no de un código de verificación de error CRC. UDP-lite ofrece flexibilidad en la cantidad de datos de carga útil protegidos por el CRC. Esta característica puede ser de gran beneficio para los servicios de audio y vídeo que implementan la ocultación de errores en el extremo receptor, es decir, a nivel de decodificador. Este último enfoque se utiliza frecuentemente en las implementaciones de aplicaciones streaming en tiempo real. El perfil solo IP puede ser usado para cualquiera de estos protocolos.

3.10.2 ROHC-TCP

El perfil ROHC TCP se introdujo originalmente en el RFC 4996. En lugar de establecer un nuevo perfil, se actualizó este perfil tomando como referencia el marco de compresión ya existente. Esto fue particularmente útil en la implementación de transferencias TCP de corta duración, como las transferencias HTTP utilizadas por los navegadores de la World Wide Web (WWW).

Sin embargo, el IETF consideró necesario introducir cambios importantes en el TCP original de ROHC. En el año 2013, se introdujo un nuevo perfil ROHC TCP definido en el estándar RFC 6846. El nuevo perfil ROHC soluciona un problema técnico con la compresión SACK y aclara otros métodos de compresión utilizados.

La Tabla 3.4 muestra la ganancia de compresión de cabecera que puede obtenerse al hacer uso de la nueva norma. Como se puede ver en la tabla, las ratios de compresión esperados dependen de la configuración TCP/IP.

TCP /IP configuration	Uncompressed (Bytes)	DATA Packet (Bytes)	ACK Packet (Bytes)
IPv4+TCP+TS	52	8	8
Ipv4+TCP+TS	52	8	7
Ipv6+TCP+TS*	72	8	7
Ipv6+TCP no-opt	60	6	5
Ipv6+TCP+SACK**	80	-	15
Ipv6+TCP+SACK***	80	-	9

*Payload size constant/**SACK present in previous packet/ ***SACK not present in previous packet

Tabla 3.4 Ganancia de compresión esperadas – norma ROHC-TCP [24]

RFC 6846 proporciona una compresión eficiente y robusta de las cabeceras TCP. Tiene en cuenta las diversas opciones TCP utilizadas, incluidos los SACKs. El perfil ROHC-TCP utiliza tres tipos de paquetes diferentes: el tipo de paquete Inicialización y Actualización (IR), el tipo de paquete Context Replication (IR-CR) y el tipo de paquete Compressed (CO).

Los principales cambios introducidos por el RFC 6846 se refieren a los formatos de cabecera comprimidos. Esto se ha considerado necesario debido a las muchas maneras en que TCP maneja los números de secuencia. El RFC 6846 define los siguientes dos conjuntos separados de cabeceras comprimidas:

- El conjunto secuencial para los paquetes que contienen un IP-ID secuencial
- El conjunto aleatorio para los paquetes sin IP-ID secuencial.

Cada conjunto de formatos de contiene a su vez ocho encabezados de base comprimidos diferentes. La razón de tener tantos formatos de cabecera es que el número de secuencia TCP, el número de confirmación TCP y la ventana TCP cambian con frecuencia en un patrón no lineal (ver Tabla 3.5). Dado el uso creciente del protocolo TCP, el perfil ROHC TCP ha recibido mucha atención en los últimos años.

Header-packet Format	Usage
Format 1	Header format used to carry changes to the TCP Sequence Number and is expected to be used on the downstream of a data transfer
Format 2	Header format is used to convey the TCP Sequence Number in scaled form and is expected to be useful for the downstream of a data transfer where the payload size is constant for multiple packets.
Format 3	Header format used to transport changes in the TCP Acknowledgment Number and is expected to be useful for the acknowledgment direction of a data transfer.
Format 4	This header format is similar to format 3, but carries a scaled TCP Acknowledgment Number.
Format 5	This header format conveys the TCP Sequence Number and the TCP Acknowledgment Number and is expected to be useful for flows that send data in both directions.
Format 6	Header format similar to format 5, but its use is to carry the TCP Sequence Number in scaled form, when the payload size is static for certain intervals in a data flow.
Format 7	This header format carries changes to both the TCP Acknowledgment Number and the TCP Window and is expected to be useful for the acknowledgment flows of data connections.
Format 8	This header format is used to convey changes to some of the more seldom changing fields in the TCP flow, such as ECN behavior, RST/SYN/FIN flags, the TTL/Hop Limit, and the TCP options list. This format carries a 7-bit CRC, since it can change the structure of the contents of the irregular chain for subsequent packets. Note that this can be seen as a reduced form of the common packet format.
Common header format	This header format can be used for all kinds of IP-ID behavior and should be useful when some of the more rarely changing fields in the IP or TCP header change. Since this header format can update control fields that decide how the decompressor interprets packets, it carries a 7-bit CRC to reduce the probability of context corruption. This header can basically convey changes to any of the dynamic fields in the IP and TCP headers, and it uses a large set of flags to provide information about which fields are present in the header format.

Tabla 3.5 Tipos de paquetes del perfil ROHC TCP [24].

La Tabla 3.6 resume los perfiles de compresión ROHC más relevantes. De la descripción anterior, desde la introducción de la ROHCv1 en 2001, los dos cambios más relevantes han sido:

- ROHCv2. Esta segunda versión de ROHC introduce una serie de simplificaciones en las reglas y algoritmos. También define mecanismos de robustez que pueden ser utilizados por la implementación de un compresor para aumentar la probabilidad de éxito de la descompresión cuando los paquetes pueden perderse y/o reordenarse en el canal ROHC.
- Un nuevo perfil de ROHC TCP. El nuevo perfil ROHC soluciona un problema técnico con la compresión SACK y aclara otros métodos de compresión utilizados. Se han incluido dos nuevos conjuntos de paquetes de cabecera comprimidos para abordar el comportamiento aleatorio de los números de secuencia TCP.

Perfil	Protocolos	v1 RFC (date)	v2 RFC (date)
0x0000	uncompressed	3095 (2001)	
0x0001 or 0x0101	RTP/UDP/IP	3095 (2001)	5525 (2008)
0x0002 or 0x0102	UDP/IP	3095 (2001)	5525 (2008)
0x0003 or 0x0103	ESP/IP	3095 (2001)	5525 (2008)
0x0004 or 0x0104	IP	3843 (2004)	5525 (2008)
0x0007 or 0x0107	RTP/UDP-Lite/IP	4019 (2005)	5525 (2008)
0x0008 or 0x0108	UDP-Lite/IP	4019 (2005)	5525 (2008)
0x0006	TCP/IP	6846 (2013)	

Tabla 3.6 Principales perfiles definidos por las diferentes normas ROHC

Capítulo 4. ROHC Y HERRAMIENTAS SOFTWARE

Este capítulo describe la librería ROHC y las herramientas asociadas utilizadas en este proyecto. También se incluye una breve descripción de la herramienta Wireshark y R.

4.1 LIBRERÍA ROHC

En este proyecto, se ha utilizado la biblioteca ROHC de código abierto desarrollada por Viveris, una empresa francesa especializada en tecnologías de la información [25]. Las plataformas soportadas incluyen Linux, BSD, Windows y Android. La biblioteca ha sido publicada como Open Source bajo la licencia GNU LGPL2.1+ desde la versión 1.7.0. La versión 2.2 es la versión estable más actualizada y la utilizada en este proyecto.

La biblioteca de ROHC implementa muchas de las funcionalidades especificadas por los RFCs de ROHC del IETF. Sin embargo, debido a la difícil tarea de implementar todas las funcionalidades especificadas por los RFCs, algunas características no se implementan. La Tabla 4.1 resume los perfiles cuyas funcionalidades han sido implementadas en su mayor parte por la biblioteca de ROHC. Una descripción detallada de las características soportadas para cada perfil se puede encontrar en el sitio web de la biblioteca de ROHC[25].

Profile Identifier	ROHC version /RFC
ROHC_PROFILE_UNCOMPRESSED	The ROHCv1 Uncompressed profile (RFC 3095, section 5.10)
ROHC_PROFILE_RTP	The ROHCv1 RTP profile (RFC 3095, section 8)
ROHC_PROFILE_UDP	The ROHCv1 UDP profile (RFC 3095, section 5.11)
ROHC_PROFILE_ESP	The ROHCv1 ESP profile (RFC 3095, section 5.12)
ROHC_PROFILE_IP	The ROHCv1 IP-only profile (RFC 3843, section 5)
ROHC_PROFILE_RTP_LLA	The ROHCv1 IP/UDP/RTP Link-Layer Assisted Profile (LLA) profile (RFC 4362, section 6)
ROHC_PROFILE_TCP	The ROHCv1 TCP profile (RFC 4996)
ROHC_PROFILE_UDPLITE_RTP	The ROHCv1 UDP-Lite/RTP profile (RFC 4019, section 7)
ROHCv2_PROFILE_IP_UDP_RTP	The ROHCv2 RTP/UDP/IP profile
ROHCv2_PROFILE_IP_UDP	The ROHCv2 UDP/IP profile
ROHCv2_PROFILE_IP_ESP	The ROHCv2 ESP/IP profile
ROHCv2_PROFILE_IP_UDPLITE_RTP	The ROHCv2 IP/UDP-Lite/RTP profile
ROHCv2_PROFILE_IP_UDPLITE	The ROHCv2 IP/UDP-Lite profile

Tabla 4.1 ROHC profiles supported by ROHC library release 2.2 [25].

La librería ROHC ha sido desarrollada y puesta a disposición de desarrolladores interesados en implementar aplicaciones que puedan beneficiarse de la reducción de los requisitos de ancho de banda requerida por las diversas aplicaciones móviles. La biblioteca ROHC proporciona un conjunto completo de APIs de compresión/descompresión, incluyendo la gestión de las estructuras de datos que permiten crear y gestionar el contexto de las comunicaciones. La Figura 4.1 proporciona la descripción de una de las varias APIs de ROHC para crear un compresor ROHC. Como se ve en el código, la API sólo requiere indicar el tamaño del CID. En cuanto al

perfil, el compresor seleccionará el que mejor responda al flujo de paquetes. Otras APIs proporcionan el medio para habilitar o deshabilitar algunas funcionalidades. Por ejemplo, el usuario puede elegir hacer uso de un perfil específico. El lector interesado puede consultar el sitio web de la biblioteca de ROHC para más detalles [25] que incluye un tutorial sobre cómo construir un compresor y un descompresor.

```
compressor = rohc_comp_new2(ROHC_SMALL_CID, ROHC_SMALL_CID_MAX,
                            gen_random_num, NULL);
if(compressor == NULL)
{
    fprintf(stderr, "failed create the ROHC compressor\n");
    goto error;
}
```

Figura 4.1 ROHC library API to create a compressor

4.2 ROHC STATS

La aplicación de estadísticas de ROHC facilita la recopilación de algunas estadísticas de (des)compresión del tráfico transitando sobre una interfaz de red o analizar trazas de tráfico (en formato PCAP). ROHC reporta las siguientes estadísticas por cada paquete capturado, ver Figura 4.2:

- *context mode*: modo de operación del protocolo ROHC: U-mode, O-mode or R-mode. Expresado en formato numérico 1, 2 y 3 correspondiendo a los modos U-mode, O-mode, R-mode, respectivamente.
- *context state*: IR, FO or SO. Expresado en formato numérico 1, 2 y 3 correspondiendo a IR, FO and SO, respectivamente.
- *packet type*: Diferentes tipos de paquetes. El Packet type 0 se refiere al paquete IR.
- *uncompressed packet size* (Bytes): tamaño original del paquete IP
- *uncompressed header size* (Bytes): tamaño de la cabecera IP antes de la compresión.
- *compressed packet size* (Bytes): tamaño del paquete IP comprimido incluyendo la cabecera
- *compressed header size* (Bytes): tamaño de la cabecera comprimida.

	A	B	C	D	E	F	G	H	I	J	K	L
1	STAT	packet number	context mode	context mode (string)	context state	context state	packet type	packet type (string)	uncompressed packet size	uncompressed header size (l	compressed packet size	compressed header size (bytes)
2	STAT	1	1	U-mode	1	IR	0	IR	974	72	974	72
3	STAT	2	1	U-mode	1	IR	0	IR	72	72	73	73
4	STAT	3	1	U-mode	1	IR	0	IR	187	72	188	73
5	STAT	4	1	U-mode	1	IR	0	IR	72	72	73	73
6	STAT	5	1	U-mode	1	IR	0	IR	631	72	631	72
7	STAT	6	1	U-mode	1	IR	0	IR	72	72	73	73
8	STAT	7	1	U-mode	1	IR	0	IR	184	72	185	73
9	STAT	8	1	U-mode	1	IR	0	IR	72	72	73	73
10	STAT	9	1	U-mode	1	IR	0	IR	609	72	609	72
11	STAT	10	1	U-mode	1	IR	0	IR	72	72	73	73
12	STAT	11	1	U-mode	3	SO	15	co_common	482	72	430	20
13	STAT	12	1	U-mode	3	SO	23	ROHCv1/TCP/rnd_8	72	72	22	22
14	STAT	13	1	U-mode	1	IR	0	IR	188	72	189	73
15	STAT	14	1	U-mode	1	IR	0	IR	72	72	73	73
16	STAT	15	1	U-mode	3	SO	15	co_common	510	72	458	20
17	STAT	16	1	U-mode	3	SO	23	ROHCv1/TCP/rnd_8	72	72	22	22
18	STAT	17	1	U-mode	3	SO	15	co_common	184	72	133	21
19	STAT	18	1	U-mode	3	SO	23	ROHCv1/TCP/rnd_8	72	72	22	22
20	STAT	19	1	U-mode	3	SO	15	co_common	993	72	941	20

Figura 4.2 Resultados de la herramienta ROHC STATS

4.3 ROHC SNIFFER

El Sniffer ROHC es una herramienta desarrollada para probar el funcionamiento y la robustez de la librería ROHC. El Sniffer ROHC monitoriza la actividad de una interfaz de red. Luego comprime y descomprime el flujo de paquetes. Luego compara el flujo de paquetes comprimido resultante con el tráfico original. El Sniffer detiene su operación y reporta los problemas encontrados durante el proceso en caso de error. La Figura 4.3 muestra una captura de pantalla de los resultados reportados por el Sniffer ROHC.

```
Archivo Editar Pestañas Ayuda
all ROHC sniffer statistics dumped
=====
packet #5000
=====
dump ROHC sniffer statistics...
general:
  total packets: 5000 packets
  bad packets: 0 packets (0%)
  loss (estim.):
    0 packets among 0 bursts (0%)
    packets per burst: max 0, avg 0, min 0
  mis-ordered packets (estim.): 0 packets (0%)
  duplicated packets (estim.): 0 packets (0%)
compression gain:
  pre-compress: 925 KB (incl. 249 KB of headers)
  post-compress: 699 KB (incl. 24 KB of headers)
  compress ratio: 75% of total packets, ie. 25% of gain on full packets
  compress ratio: 9% of total headers, ie. 91% of gain on headers alone
  used and re-used contexts: 29
packets per profile:
  ROHCv1 Uncompressed profile: 0 packets (0%)
  ROHCv1 IP/UDP/RTP profile: 0 packets (0%)
  ROHCv1 IP/UDP profile: 4561 packets (91%)
  ROHCv1 IP-only profile: 6 packets (0%)
  ROHCv1 IP/TCP profile: 428 packets (8%)
  ROHCv1 IP/UDP-Lite profile: 0 packets (0%)
  ROHCv2 IP/UDP profile: 0 packets (0%)
  ROHCv2 IP/ESP profile: 0 packets (0%)
  ROHCv2 IP-only profile: 0 packets (0%)
packets per mode:
  U-mode: 4995 packets (100%)
  O-mode: 0 packets (0%)
  R-mode: 0 packets (0%)
packets per state:
  IR state: 70 packets (1%)
  FO state: 27 packets (0%)
  SO state: 4898 packets (98%)
packets per packet type:
  packet type IR: 70 packets (1%)
  packet type IR-DVN: 8 packets (0%)
  packet type ROHCv1/U0-0: 4521 packets (90%)
  packet type ROHCv1/non-RTP/U0-1: 0 packets (0%)
  packet type ROHCv1/RTP/U0-1-ID: 0 packets (0%)
  packet type ROHCv1/RTP/U0-1-TS: 0 packets (0%)
  packet type ROHCv1/RTP/U0-1: 0 packets (0%)
  packet type ROHCv1/non-RTP/U0R-2: 21 packets (0%)
  packet type ROHCv1/RTP/U0R-2: 0 packets (0%)
  packet type ROHCv1/RTP/U0R-2-ID: 0 packets (0%)
  packet type ROHCv1/RTP/U0R-2-TS: 0 packets (0%)
  packet type ROHCv1/Uncomp/Normal: 0 packets (0%)
  packet type co_common: 134 packets (2%)
  packet type ROHCv1/TCP/rnd_1: 115 packets (2%)
  packet type ROHCv1/TCP/rnd_2: 0 packets (0%)
  packet type ROHCv1/TCP/rnd_3: 44 packets (0%)
  packet type ROHCv1/TCP/rnd_4: 7 packets (0%)
  packet type ROHCv1/TCP/rnd_5: 53 packets (1%)
  packet type ROHCv1/TCP/rnd_6: 0 packets (0%)
  packet type ROHCv1/TCP/rnd_7: 20 packets (0%)
  packet type ROHCv1/TCP/rnd_8: 2 packets (0%)
  packet type ROHCv1/TCP/seq_1: 0 packets (0%)
  packet type ROHCv1/TCP/seq_2: 0 packets (0%)
  packet type ROHCv1/TCP/seq_3: 0 packets (0%)
  packet type ROHCv1/TCP/seq_4: 0 packets (0%)
  packet type ROHCv1/TCP/seq_5: 0 packets (0%)
  packet type ROHCv1/TCP/seq_6: 0 packets (0%)
  packet type ROHCv1/TCP/seq_7: 0 packets (0%)
  packet type ROHCv1/TCP/seq_8: 0 packets (0%)
  packet type ROHCv1/IR-CR: 0 packets (0%)
  packet type ROHCv2/co_repair: 0 packets (0%)
  packet type ROHCv2/pt_0_crc3: 0 packets (0%)
  packet type ROHCv2/non-RTP/pt_0_crc7: 0 packets (0%)
```

Figura 4.3 Resultados de la herramienta ROHC Sniffer

Como se observa en la figura, las estadísticas reportadas están organizadas en cinco bloques principales:

- *General*: este bloque incluye el número total de paquetes capturados y procesados, paquetes por ráfaga, paquetes desordenados y paquetes duplicados
- *Compression gain*: número total kiloBytes sin comprimir, número total de kiloBytes, comprimidos
- *Compression ratio*: ratio de compresión por paquete y por cabecera
- *Compression gain*: ganancia de compresión por paquete y por cabecera
- *Packets per profile*: número de paquetes no comprimidos procesados por perfil y versión de ROHC.
- *Packets per mode*: número de paquetes no comprimidos procesados en los diferentes modos ROHC: U-mode, O-mode and R-mode.
- *Packets per state*: número de paquetes procesados bajo cada estado: IR, FO y SO.
- *Packets per type*: número de paquetes por tipo.

Esta herramienta es especialmente útil para verificar el funcionamiento del compresor y del descompresor. Su uso nos ha permitido verificar las prestaciones de los mecanismos de compresión e identificar las limitaciones de la librería ROHC.

4.4 WIRESHARK

Wireshark es un software de código abierto utilizado para el análisis del tráfico de red [26]. Se utiliza para la resolución de problemas de red, análisis, desarrollo de software y protocolos de comunicaciones, y educación.

Wireshark tiene muchas funcionalidades [26]. En el marco de este proyecto, las funcionalidades más utilizadas fueron:

- Análisis de trazas de protocolos RTP, IP, UDP y TCP.
- Captura en tiempo real y análisis off-line.
- Versiones disponibles para diversos sistemas operativos: Windows, Linux y macOS.
- Interfaz gráfica.
- Formato de fichero tcpdump (libpcap) y PCAP.
- Tráfico Ethernet, IEEE 802.11, PPP/HDLC, ATM, Bluetooth, USB, Token Ring, Frame Relay, FDDI entre otros. En este Proyecto se hizo uso de una interfaz de red IEEE 802.11.

En este proyecto, Wireshark fue utilizado para verificar los protocolos utilizados por las distintas aplicaciones: Youtube, Spotify, etc. La información proporcionada por Wireshark se ha utilizado en la evaluación para identificar el tipo de paquetes, por ejemplo, paquetes de datos y ACK, que están comprimidos, ya que esta información no puede ser obtenida con las herramientas ROHC.

4.5 Herramientas de procesamiento y visualización

Para el procesamiento y visualización de los datos se han utilizado R y MS Excel [27].

R es un software de procesamiento de datos para computación estadística y gráficos. Es un proyecto de código abierto desarrollado en Bell Labs por John Chambers y su grupo.

En este proyecto, R ha sido utilizado para procesar los datos capturados por las herramientas de la biblioteca de ROHC. Las principales métricas obtenidas por R han sido:

- Ganancia de compresión de paquetes (absoluta y porcentual).
- Relación de compresión de paquetes (absoluta y porcentual).



MS Excel ha sido utilizado para producir los gráficos y gráficos de los siguientes resultados:

- Trazas de encabezado comprimidas y no comprimidas.
- Histogramas de las ganancias de compresión por tipo de paquete ROHC.
- Porcentaje de tipos de paquetes ROHC.

Todos los resultados anteriores se reportan para las varias aplicaciones de interés.

Capítulo 5. PLATAFORMA EXPERIMENTAL Y EVALUACIÓN

Este capítulo consta de cuatro partes principales. La primera sección describe la instalación de las herramientas y las trazas de datos obtenidas de los experimentos. La segunda y tercera sección presentan los resultados y el análisis de la primera y segunda serie de experimentos. Por último, un breve resumen revisará las principales conclusiones.

5.1 Descripción de la plataforma

En esta sección se describe en primer lugar el montaje experimental. Esta descripción proporciona detalles sobre la configuración del hardware y el software. A continuación, se proporcionan los detalles de las trazas.

5.1.1 Hardware y software

En la Tabla 5.1 se resumen las características del hardware y software utilizadas para llevar a cabo la evaluación de la compresión/descompresión de ROHC. La plataforma de experimentación consistía en un Raspberry PI 3 que incluía interfaces WiFi, Bluetooth y Ethernet. Los datos capturados durante los experimentos se obtuvieron mediante la monitorización de la interfaz WiFi.

Componente	Características
Raspberry	Modelo RPi3Plus
Procesador	1.4 GHz 64-bit 4 core ARM cortex-a53 CPU
Memoria	1GB LPDDR2 SDRAM
SD card	SanDisk Class 10 - 32 GB
Interfaz WiFi	2,4 GHz y 5 GHz IEEE 802.11.b / g / n / AC
Bluetooth	4.2 BLE
Sistema Operativo	Raspbian
ROHC	ROHC V2.2
Herramientas ROHC	Sniffer Stat
Wireshark	v.22

Tabla 5.1 Plataforma experimental

La figura 5.1 muestra las características del hardware empleado en la plataforma experimental. Como se muestra en la figura, el montaje consiste en una Raspberry Pi conectada vía WiFi a Internet. La Figura 5.2 muestra la arquitectura del protocolo de red y los módulos ROHC instalados en el modelo Raspberry Pi. Las principales herramientas de ROHC utilizados durante los experimentos consistieron en el ROHC STAT y el ROHC SNIFFER.

En la primera fase, la herramienta STAT se utilizó para comprimir los flujos de paquetes de cinco aplicaciones de usuario diferentes. Las trazas de paquetes para las cuatro aplicaciones siguientes se obtuvieron mediante la monitorización de la interfaz inalámbrica de la Raspberry:

- YouTube video streaming.
- YouTube video almacenado en la nube
- Spotify
- Google Hangouts

Una quinta traza de VoIP descargada de Internet también fue comprimida y sus estadísticas fueron obtenidas usando la herramienta ROHC STAT.

En una segunda fase, se utilizó la herramienta Sniffer de la biblioteca ROHC. Los principales objetivos de este segundo experimento fueron:

- Verificar el funcionamiento de la librería ROHC con v1 y v2.
- Evaluar los beneficios del mecanismo de compresión ROHC para diferentes aplicaciones

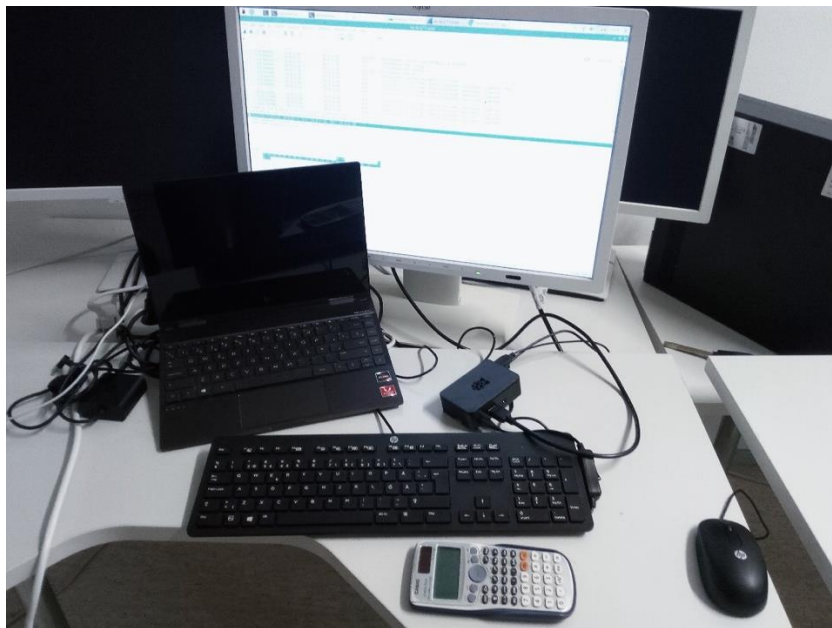


Figura 5.1 Plataforma experimental - hardware

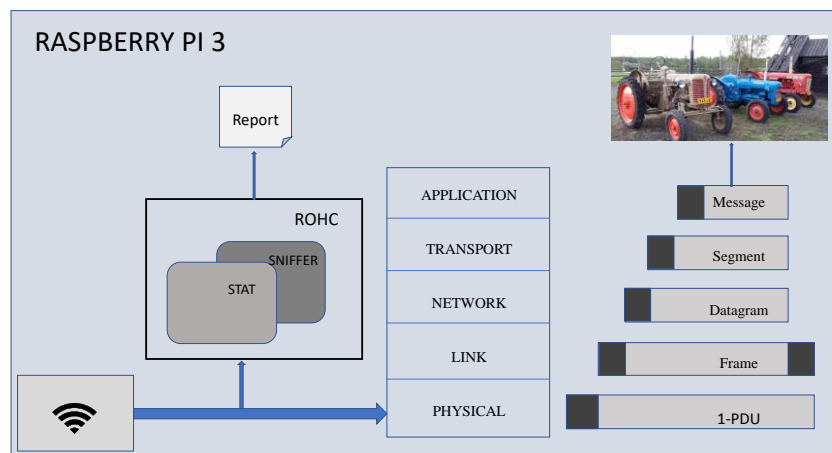


Figura 5.2 Plataforma experimental - software

5.1.2 Métricas

Para evaluar el rendimiento del esquema de compresión ROHC, las principales métricas de interés que se están considerando son:

- Ratio de compresión de la cabecera. Definido como la relación entre el tamaño de la cabecera comprimida y el tamaño de la cabecera sin comprimir.
- Ratio de compresión del paquete. Definido as the ratio between compressed packet size and the uncompressed packet size.
- Ganancia de compresión de cabecera por tipo de paquete ROHC.
- Ganancia de compresión por paquete ROHC por aplicación.

5.2 Evaluación de prestaciones – ROHC STAT

Esta primera fase de evaluación utiliza la herramienta STAT de la librería ROHC. La evaluación se realiza considerando cinco aplicaciones diferentes para el usuario final.

5.2.1 Datos experimentales

En la Tabla 5.2 se resumen las principales características de las trazas de datos utilizado en la primera fase de los experimentos.

Trazas	Protocolos	Aplicación
Video/Audio streaming	TCP/IPv6	Youtube vídeo streaming
Reproducción de vídeo	TCP/IPv6	Youtube reproducción vídeo
Reproducción de música	TCP/IP	Spotify
Video conferencia	UDP/IP	Google Hangouts
VoIP G.711	RTP/UDP/IP	VoIP

Tabla 5.2 Trazas experimentales

5.2.2 Video streaming - Youtube

La Figura 5.3 muestra el pantallazo de Wireshark de la aplicación de streaming de vídeo y audio. La figura muestra que la pila de protocolos está formada por protocolos TCP/IP. El tamaño de la cabecera TCP es de 20 bytes y el tamaño de la cabecera IPv6 es de 40 bytes

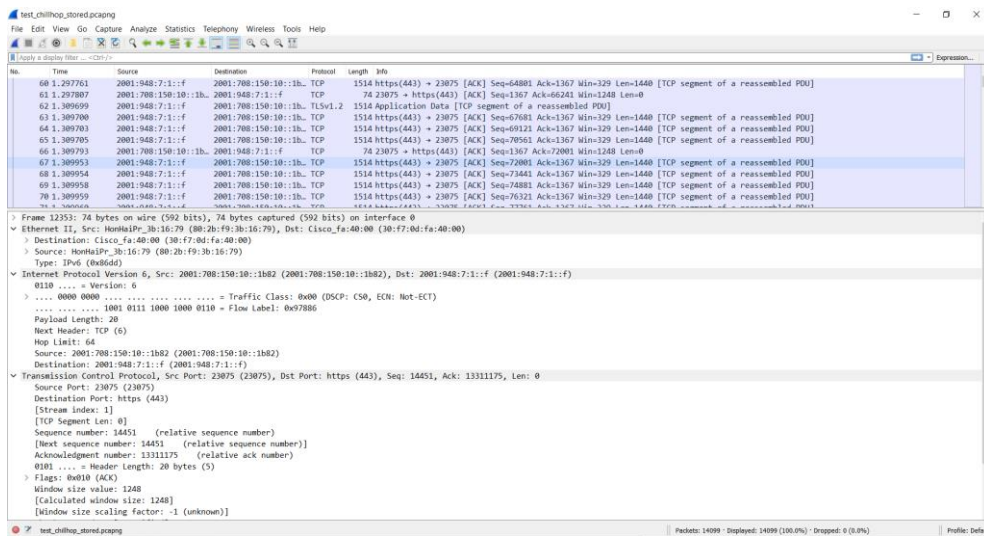


Figura 5.3 Video y audio streaming – pantalla Wireshark

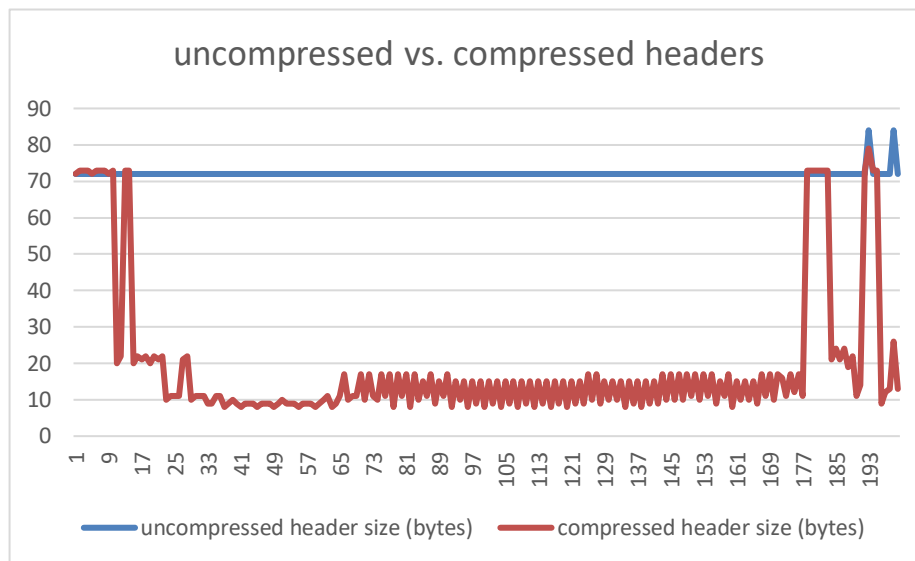


Figura 5.4 Vídeo streaming YouTube – cabeceras

La Figura 5.4 muestra las trazas de las cabeceras sin comprimir y comprimidas utilizando la librería ROHC. En este experimento se utilizó una traza de 200 paquetes de una sesión de streaming YouTube. El compresor se inició a los 30 segundos de haber iniciado la sesión streaming dado que nuestro objetivo es estudiar la compresión durante el envío del vídeo.

Al principio de la traza capturada, el tamaño de las cabeceras comprimidas es mayor que el de las cabeceras originales. Esto se debe a que los paquetes IR contienen información adicional necesaria para crear el contexto de la comunicación en el extremo receptor. El compresor se mueve al estado SO después del paquete número 15, lo cual significa que el proceso de compresión comienza a funcionar con la tasa de compresión más alta.

Mientras que el compresor se mantiene en el estado SO, se pueden distinguir claramente dos intervalos de tiempo:

- Del paquete 29 al paquete 64. Los paquetes intercambiados entre ambos extremos consisten principalmente en paquetes de tipo ROHCv1/TCP/rnd_1 y ROHCv1/TCP/rnd_3. El primer tipo se utiliza para enviar datos de vídeo, mientras que el segundo (rnd_3) se corresponde a los paquetes TCP ACK.
- Del paquete 65 al paquete 175. Los paquetes consisten principalmente en paquetes ROHCv1/TCP/rnd_1 y co_common. Los paquetes ROHCv1/TCP/rnd_1 llevan cambios en el SN de TCP en la dirección de bajada y los paquetes ROHCv1/TCP/rnd_3 corresponden a los ACKs. Los paquetes co_common, utilizados solo en el perfil ROHC/TCP, realizan la misma función que ROHCv1/TCP/rnd_3 pero llevan información adicional. Esta es la razón por la que la ratio de compresión obtenida es más baja que la conseguida durante el período anterior.

Hacia el final de la traza obtenida, los paquetes IR aparecen de nuevo. Dado que el compresor funciona en U mode (sin feedback). Por esta razón, el compresor debe enviar de vez en cuando paquetes IR. De esta manera, el compresor se asegura de que toda la información del contexto guardada por el descompresor se actualiza correctamente.

La Figura 5.5 muestra las estadísticas de los paquetes ROHC. Como se puede ver en la figura, la ganancia total de compresión de cabecera es de 16,7%, lo que puede considerarse un buen resultado. La Figura 5.6 muestra el porcentaje de paquetes comprimidos generados por el proceso de compresión ROHC. Como se puede ver en la figura, a pesar de que los paquetes ROHCv1/TCP/rnd_3, ROHCv1/TCP/rnd_5, ROHCv1/TCP/rnd_7 y ROHCv1/TCP/rnd_8 tienen una tasa de compresión muy alta, estos paquetes no tienen un gran impacto en relación con la compresión global. Esto se debe al hecho de que los paquetes ROHCv1/TCP/rnd_1 y co_common representan más del 80% del total de paquetes comprimidos.

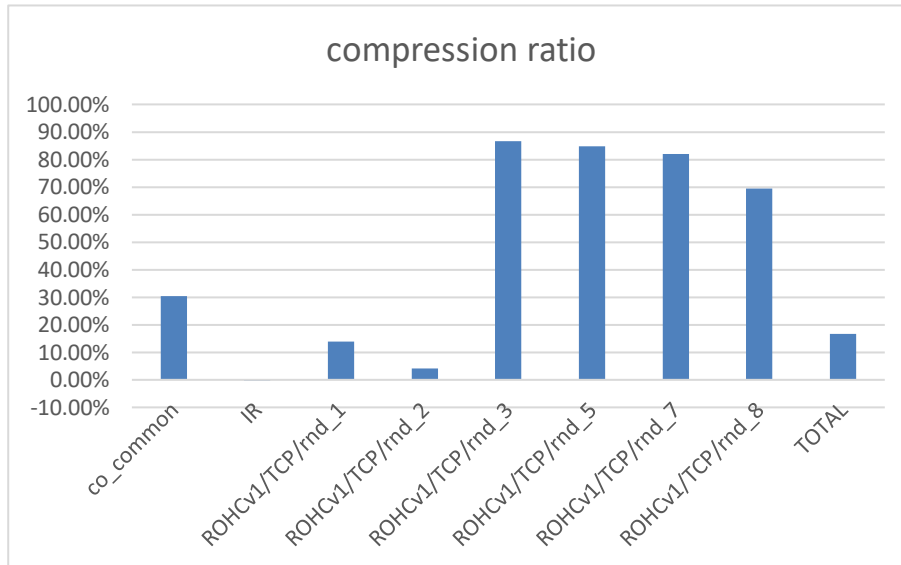


Figura 5.5 Vídeo streaming Youtube – ganancia de compresión por tipo de paquete

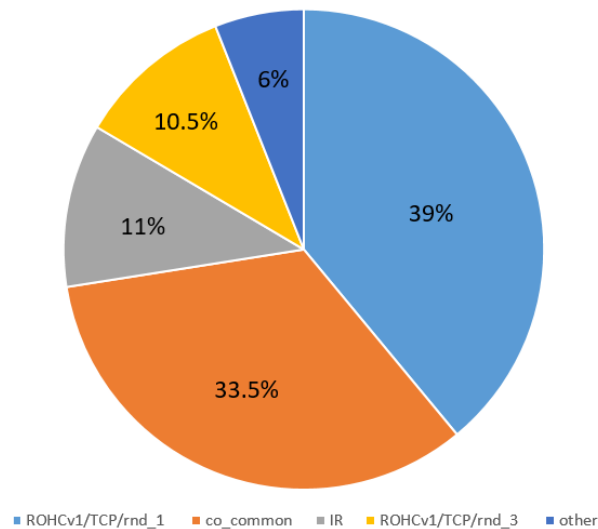


Figura 5.6 Vídeo streaming Youtube – porcentajes por tipo de paquetes ROHC

5.2.3 Video retrieval - YouTube

La pila de protocolos utilizada por esta aplicación consiste en los protocolos TCP/IP. A diferencia de la aplicación anterior, en la que por cada paquete enviado se devuelve un ACK, en este caso se recibe un ACK asociado a un grupo de paquetes recibidos. Aproximadamente, los paquetes que transportan 20 segundos de vídeo son reconocidos por un solo ACK.

Figura 5.7 Reproducción de vídeo YouTube – trazasLa figura 5.7 muestra el tamaño de las cabeceras de los paquetes comprimidos y los originales para los 200 paquetes obtenidos durante la reproducción del vídeo de YouTube. Desde el paquete 0 hasta el paquete 50, la traza muestra dos picos, que corresponden a los paquetes IR. Los tres picos mostrados en el período entre el número de paquete 70 y 90 corresponden a paquetes co_comunes, que a su vez corresponden a ACKs emitidos por el receptor.

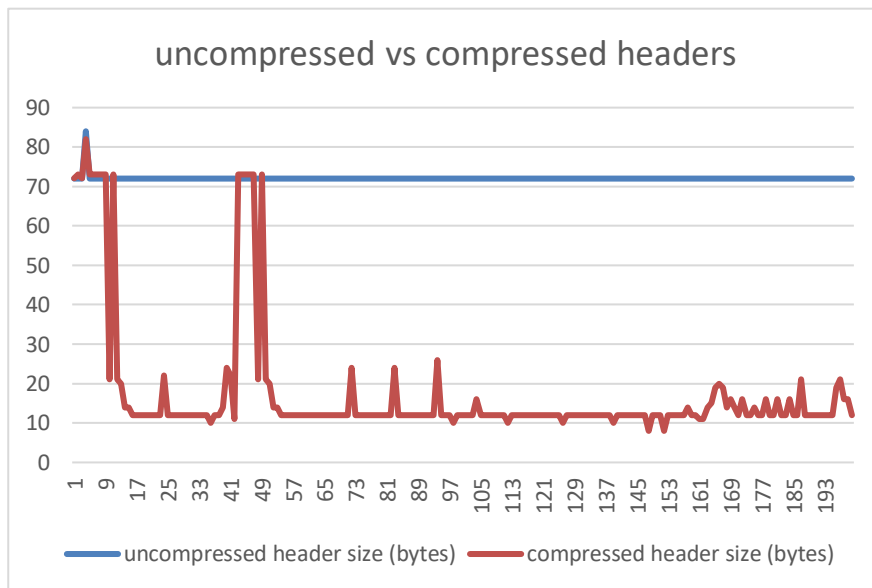


Figura 5.7 Reproducción de vídeo YouTube – trazas de cabeceras

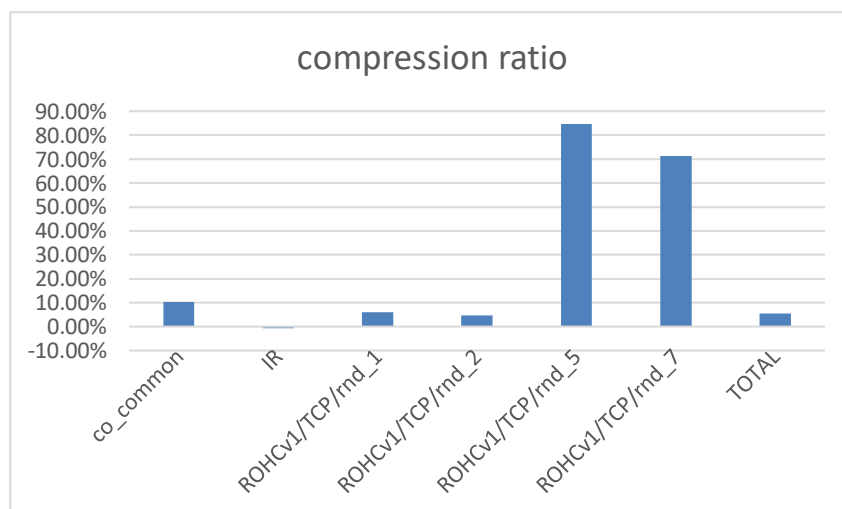


Figura 5.8 Reproducción de vídeo YouTube – ganancia de compresión

La Figura 5.8 muestra la contribución de la ganancia de compresión de los diferentes formatos de paquetes utilizados por el perfil TCP de ROHC. Para esta aplicación, la compresión en cabecera resulta muy eficiente, pero la compresión no tiene un impacto relevante en relación con el tamaño total del paquete. La compresión del paquete es aproximadamente del 5% para esta aplicación.

La Figura 5.9 muestra el peso de cada tipo de paquete en la traza obtenida: Como se observa en la figura, el porcentaje de paquetes comprimidos ROHCv1/TCP/rnd_2 (70,5%), co_common (9%), IR (8%), suman el 87.5% de la traza total. La ganancia de compresión resultante es baja para este tipo de paquetes, ver Figura 5.8. este resultado revela que el uso de la compresión de cabeceras debe evaluarse teniendo en cuenta las características de la pila de protocolos que utilizan cada tipo de aplicación.

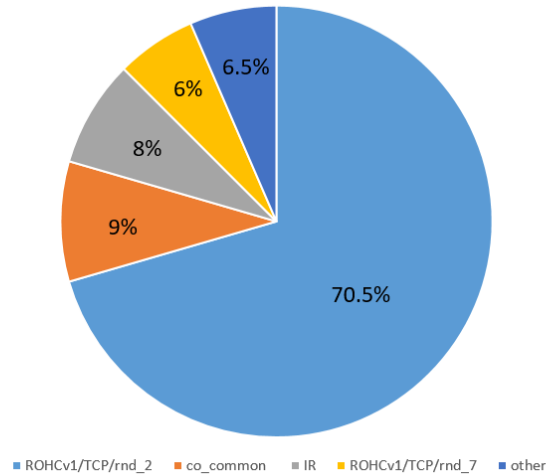


Figura 5.9 Reproducción de vídeo YouTube – porcentaje por tipo de paquetes ROHC

5.2.4 Reproducción de música - Spotify

La Figura 5.10 muestra el comportamiento del compresor ROHC al ser utilizado en la aplicación Spotify para escuchar música. La pila de protocolos sigue estando formada por protocolos TCP/IP, pero muestra un comportamiento muy diferente. El tráfico se envía en ráfagas de manera similar a la reproducción del vídeo almacenado (YouTube), y ACK (co_common) se emiten en el sentido opuesto, es decir desde el cliente al servidor.

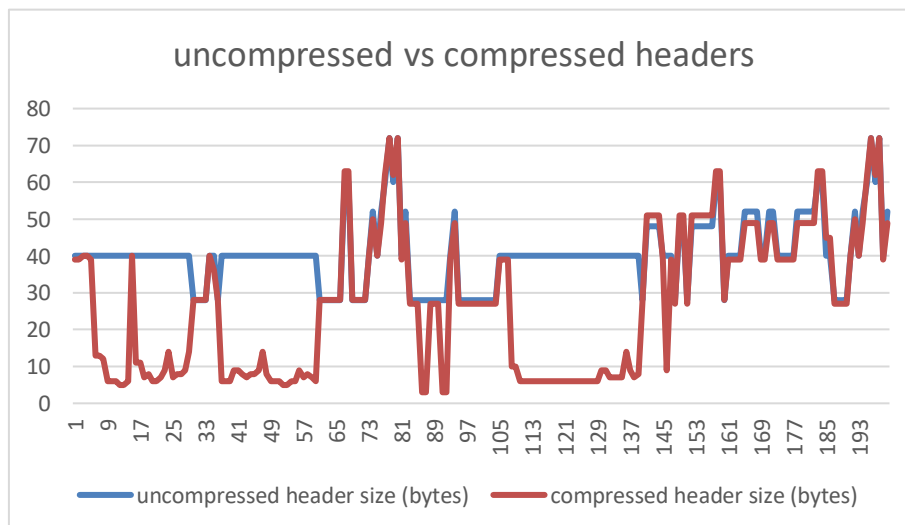


Figura 5.10 Spotify – trazas de cabeceras

La Figura 5.10 muestra que el compresor transmite un gran número de paquetes IR, todos los puntos donde las líneas roja y azul se tocan. Sólo hay unas pocas áreas en las que el compresor es eficaz: desde los paquetes 5 a 29; desde los 37 a los 60; y desde los 108 a los 139. Lo que significa que el 58% de los paquetes que pasan por el compresor son iguales o mayores que los paquetes originales.

La Figura 5.11 muestra la compresión por paquetes. Este gráfico se ve afectado por los resultados de compresión de cabecera y esto se refleja en los resultados, ya que el porcentaje de paquetes IR es muy elevado, seguidos por ROHCv1/TCP/seq_1 (20%), véase la Figura 5.12. Esto significa que la mayoría de los paquetes ROHC serán más grandes o iguales que los paquetes IP originales (no comprimidos), lo que obviamente penaliza significativamente la tasa de compresión total, 6,8%.

Los paquetes de tipo ROHCv1/TCP/seq_5 tienen una ganancia de compresión de paquete elevada, sin embargo, a pesar de ser los terceros más abundantes, su peso sobre el tráfico total es reducido (7% aproximadamente). Por esta razón se puede decir que el compresor ROHC v1 no es capaz de establecer un contexto de manera correcta para esta aplicación. Unos de los motivos por los cuales se da este caso es por la incompatibilidad de este protocolo con ciertas versiones del protocolo TCP.

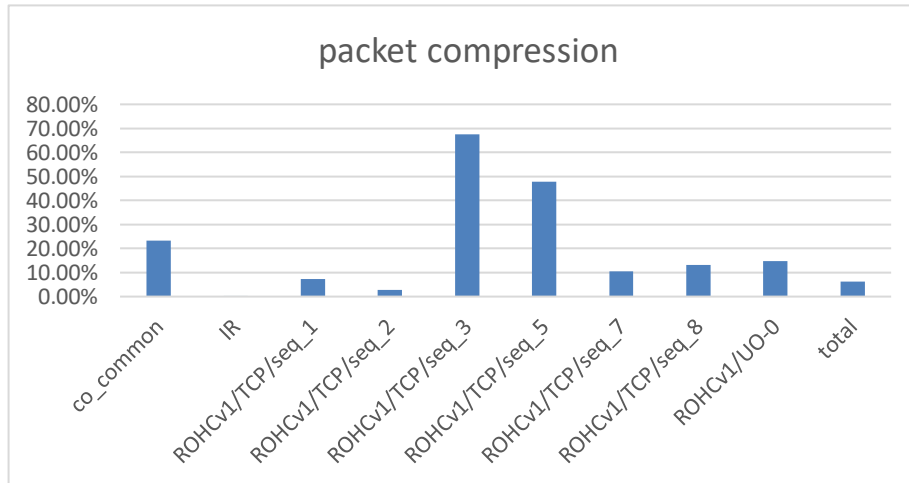


Figura 5.11 Spotify – ganancia de compresión

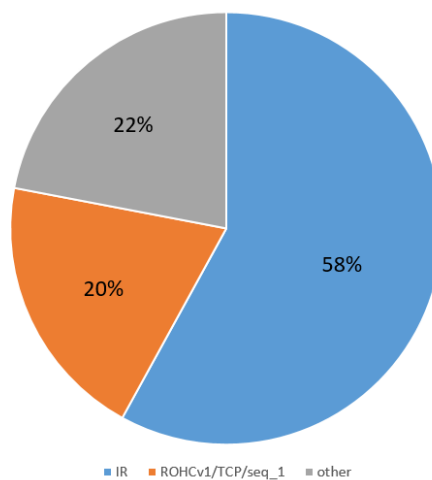


Figura 5.12 Spotify – porcentaje por tipo de paquetes ROHC

5.2.5 Video conferencia – Hangouts

La figura 5.13 muestra una captura de Wireshark correspondiente a una videoconferencia con la aplicación de Google Hangouts. La figura muestra que la pila de protocolos está formada por protocolos UDP/IP. La cabecera UDP tiene un tamaño de 8 bytes y la cabecera IP, 40 bytes. Existen algunos paquetes TCP, que pertenecen a la Seguridad de la Capa de Transporte (TLS), pero que no representan una parte significativa de los datos obtenidos.

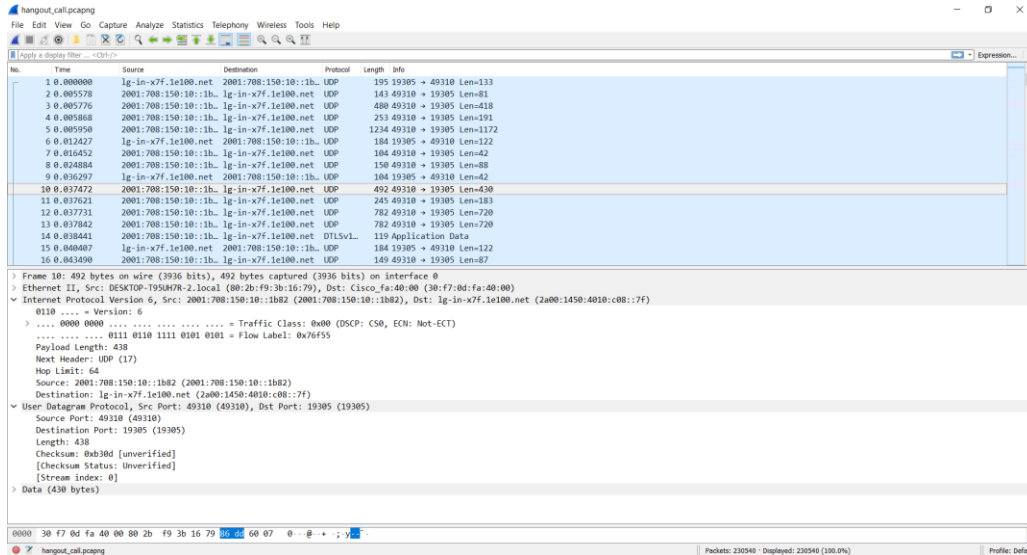


Figura 5.13 Video conferencing Hangouts – pantalla Wireshark

La Figura 5.14 muestra el tamaño de las cabeceras obtenidas y sus correspondientes cabeceras ROHC comprimidas para esta aplicación. En este caso, el compresor ROHC trabaja muy eficientemente, comprimiendo cabeceras de 48 a 3 bytes en el mejor de los casos. En cuanto a los paquetes IR, los resultados muestran cómo se envían estos paquetes en seis ocasiones diferentes durante el periodo de observación. Esto puede deberse al protocolo de capa de control, UDP, este protocolo no tiene control sobre los paquetes que envía, no ACKs, esta puede ser la razón por la que se necesita enviar más IRs por parte del compresor, para asegurarse de que el receptor siempre tenga la información correcta sobre el contexto.

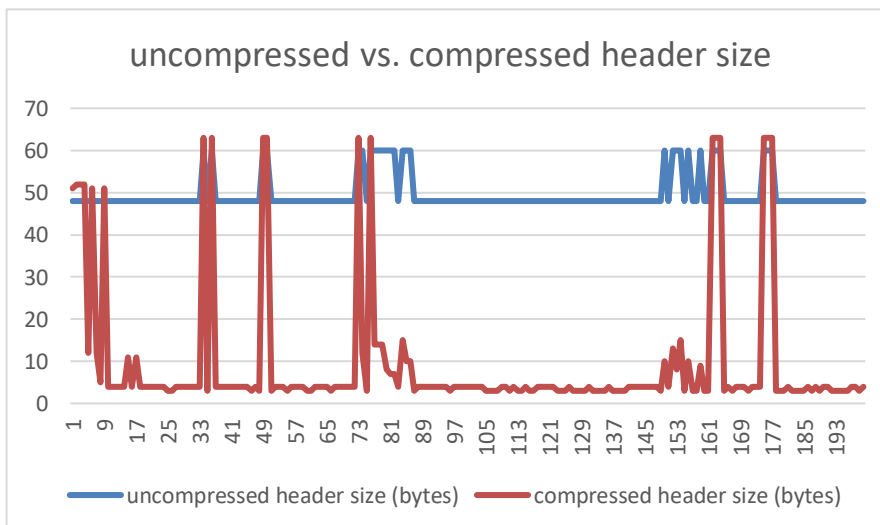


Figura 5.14 Video conferencia Hangouts – trazas de cabeceras

Como se puede ver en la Figura 5.14, se puede esperar una buena ganancia de compresión ya que hay largos períodos de tiempo en los que la compresión alcanza una relación de compresión de más del 90% a nivel de cabecera. Sin embargo, la Figura 5.15 muestra que los paquetes más comunes ROHCv1/UO-0 tienen una gran carga útil por lo que contribuyen poco a la compresión. Por lo tanto, los grandes esfuerzos del compresor perderán relevancia, obteniendo una ganancia total de compresión de paquetes del 6,42%.

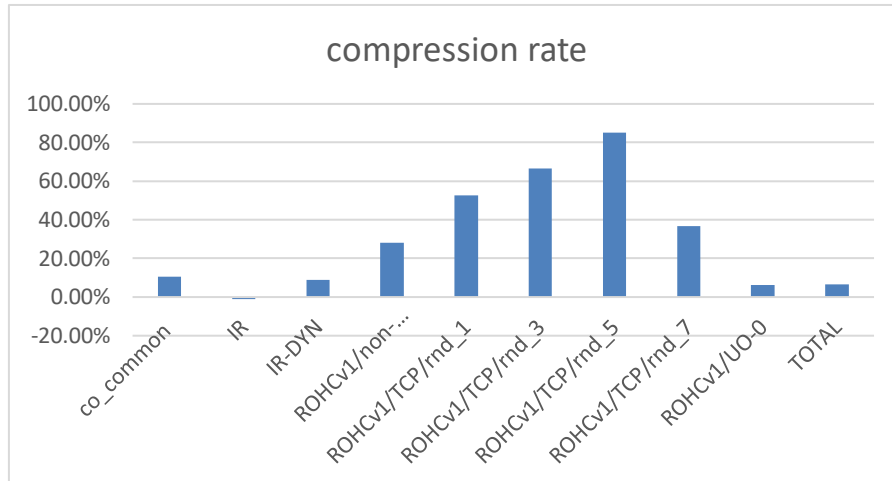


Figura 5.15 Vídeo conferencia Hangouts – ganancia por tipos de paquete ROHC

La figura 5.16 muestra el porcentaje de tipos de paquetes ROHC comprimidos. Como se puede ver en la figura, el peso de cada paquete es: ROHCv1/UO-0 80%, IR 9%, IR-DYN 2% y paquetes TCP (8%). A diferencia de las trazas anteriores, esta traza comprende paquetes IR-DYN. Este último tipo de paquetes se utiliza para actualizar el contexto del receptor sin tener que enviar los campos estáticos.

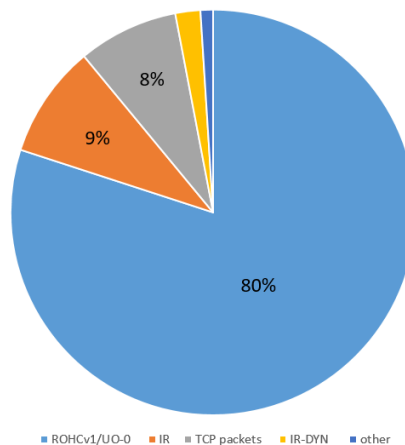


Figura 5.16 Vídeo conferencia Hangouts – porcentaje por tipo de paquete ROHC

5.2.6 VoIP G.711

A fin de evaluar las prestaciones de una aplicación VoIP utilizando el protocolo RTP, se ha utilizado una traza en formato PCAP. La Figura 5.17 muestra los resultados obtenidos. Como se puede ver de la figura, la pila de protocolos consiste en protocolos RTP/UDP/IP. La cabecera UDP consta de 8 bytes y el tamaño de la cabecera IPv4 es de 20 bytes. Además, la figura muestra que el proceso de compresión sólo se aplica a las cabeceras UDP/IP, pero no a las cabeceras RTP. Esta es una de las series limitaciones de la librería ROHC.

STAT	packet number	context mode	context mode (string)	context state	packet type	packet type (string)	uncompressed packet size	uncompressed header size	compressed packet size	compressed header size (bytes)
1	1	1	IR	1	IR	IR	486	28	485	27
2	2	1	U-mode	1	IR	IR	314	28	314	28
3	3	1	U-mode	1	IR	IR	33	28	33	28
4	4	1	U-mode	1	IR	IR	1089	28	1089	28
5	5	1	U-mode	1	IR	IR	340	28	339	27
6	6	1	U-mode	1	IR	IR	200	28	200	28
7	7	1	U-mode	1	IR	IR	200	28	200	28
8	8	1	U-mode	1	IR	IR	200	28	200	28
9	9	1	U-mode	1	IR	IR	200	28	200	28
10	10	1	U-mode	3	SO	ROHCv1/non-RTP/UO-1	200	28	177	5
11	11	1	U-mode	3	SO	ROHCv1/non-RTP/UO-1	200	28	177	5
12	12	1	U-mode	3	SO	ROHCv1/non-RTP/UO-1	200	28	177	5
13	13	1	U-mode	3	SO	ROHCv1/non-RTP/UO-1	200	28	177	5
14	14	1	U-mode	3	SO	ROHCv1/non-RTP/UO-1	200	28	177	5
15	15	1	U-mode	3	SO	ROHCv1/non-RTP/UO-1	200	28	177	5
16	16	1	U-mode	3	SO	ROHCv1/non-RTP/UO-1	200	28	177	5
17	17	1	U-mode	3	SO	ROHCv1/non-RTP/UO-1	200	28	177	5
18	18	1	U-mode	3	SO	ROHCv1/non-RTP/UO-1	200	28	177	5
19	19	1	U-mode	3	SO	ROHCv1/non-RTP/UO-1	200	28	177	5
20	20	1	U-mode	3	SO	ROHCv1/non-RTP/UO-1	200	28	177	5
21	21	1	U-mode	3	SO	ROHCv1/non-RTP/UO-1	200	28	177	5
22	22	1	U-mode	3	SO	ROHCv1/non-RTP/UO-1	200	28	177	5
23	23	1	U-mode	3	SO	ROHCv1/non-RTP/UO-1	200	28	177	5
24	24	1	U-mode	3	SO	ROHCv1/non-RTP/UO-1	200	28	177	5
25	25	1	U-mode	3	SO	ROHCv1/non-RTP/UO-1	200	28	177	5
26	26	1	U-mode	3	SO	ROHCv1/non-RTP/UO-1	200	28	177	5
27	27	1	U-mode	3	SO	ROHCv1/non-RTP/UO-1	200	28	177	5
28	28	1	U-mode	3	SO	ROHCv1/non-RTP/UO-1	200	28	177	5
29	29	1	U-mode	3	SO	ROHCv1/non-RTP/UO-1	200	28	177	5
30	30	1	U-mode	3	SO	ROHCv1/non-RTP/UO-1	200	28	177	5

Figura 5.17 VoIP G.711 – pantalla Excel

La Figura 5.18 muestra la traza de las cabeceras originales y de las comprimidas. Para este caso, el compresor ROHC funciona muy bien. Sólo se envían ocho paquetes IR al principio de la traza y no se envían de nuevo. Esto puede deberse al hecho de que el compresor no ha estado expuesto a un tráfico real (impredecible). Por esta razón, el proceso de compresión funciona a la tasa máxima de compresión, es decir, en estado SO como puede observarse en la Figura 5.17. La mayoría de los paquetes comprimidos son ROHCv1/no-RTP/UO-1. Dando una ganancia resultante de compresión de cabecera del 82%.

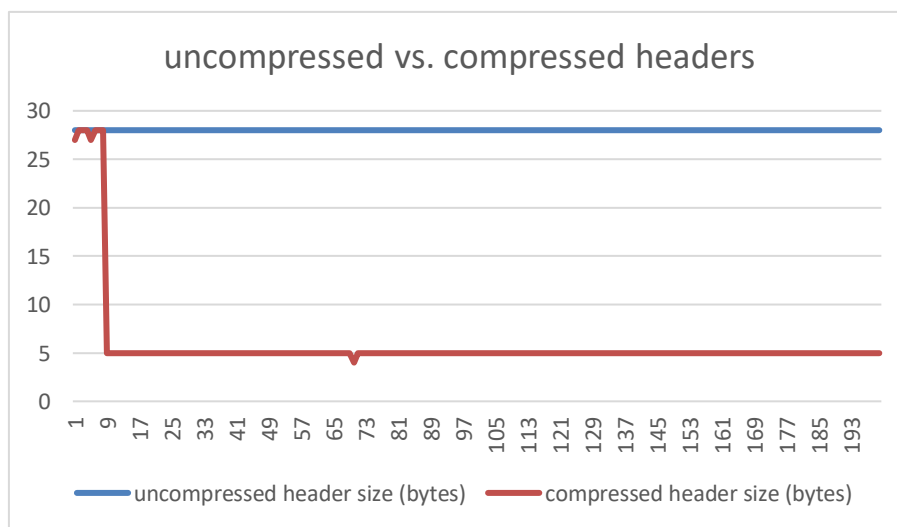


Figura 5.18 VoIP G.711 – trazas de cabeceras

La Figura 5.19 muestra los diferentes tipos de paquetes y el impacto que tiene el compresor sobre el tamaño de los paquetes. La figura 5.20 muestra la distribución de los tipos de paquetes: 95,5% de ROHCv1/no-RTP/UO-1, 4% de IR y 0,5% de ROHCv1/UO-0. En este caso, teniendo en cuenta que los paquetes llevan una carga útil pequeña, la compresión de cabecera resulta muy efectiva: la ganancia total de compresión de los paquetes es del 10,7%. Este resultado muestra que el uso del compresor ROHC debe ser considerado como una buena opción para comunicaciones de voz.

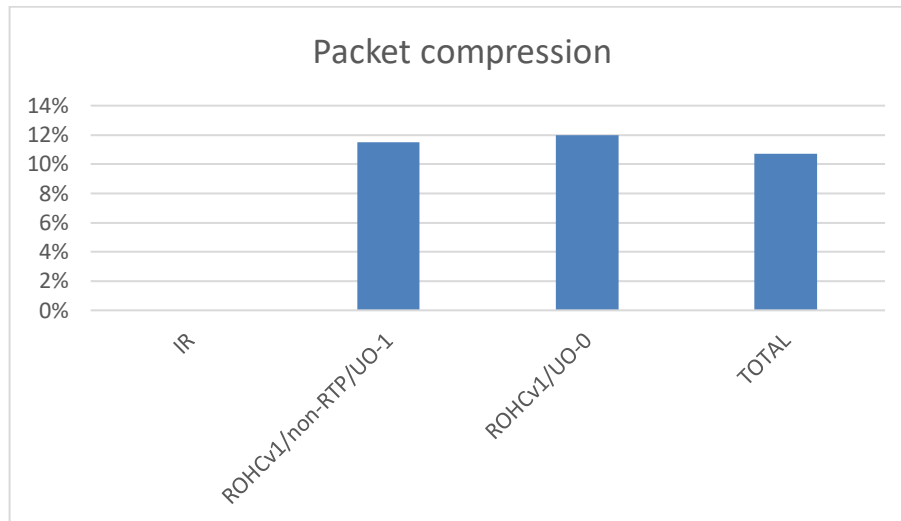


Figura 5.19 VoIP G7.11 – ganancia de compresión por tipo de paquete ROHC

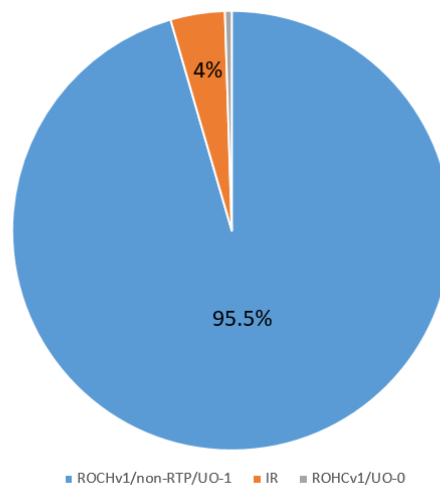


Figura 5.20 VoIP G.711 – porcentaje por tipo de paquetes ROHC

5.3 Evaluación de prestaciones – ROHC Sniffer

Los resultados obtenidos durante la primera fase muestran una alta variabilidad en la relación de compresión obtenida por las diferentes aplicaciones. Mientras que la relación de compresión de los paquetes fue del 5% para la reproducción de un video online (YouTube) para las aplicaciones de voz fue del 10,7%.

5.3.1 Escenario de experimentación

El segundo estudio de prestaciones evalúa y compara las relaciones de compresión de cabecera y de paquetes para dos configuraciones diferentes: Para ello, se utilizó la herramienta ROHC Sniffer. Se estudiaron las dos siguientes aplicaciones:

- Conferencia de sólo voz: Consistía en una conferencia de voz utilizando Google Hangouts. Durante la conferencia, el video fue desactivado. Dado que la pila de protocolos consistía en UDP sobre IP, se evaluaron las dos versiones ROHC, ROHCv1 y ROHCv2 de los perfiles UDP/IP de ROHC.
- Videoconferencia. Al igual que en el caso anterior, se usó Google Hangouts como la fuente de tráfico. Sin embargo, en este caso, tanto el vídeo como la voz fueron activados. La evaluación se realizó utilizando los perfiles ROHCv1 y ROHCv2 y el perfil UDP/IP.

5.3.2 Audio conferencia – Google Hangouts

Las Figura 5.21 y la Figura 5.22 muestran los resultados obtenidos para las aplicaciones de audioconferencia usando la aplicación Google Hangouts.

En este caso se realizaron pruebas con la herramienta de ROHC Sniffer utilizando las dos versiones ROHC, es decir ROHCv1 y ROHCv2. Como se puede observar en las dos figuras, 5000 paquetes UDP/IP han sido capturados y procesados (comprimidos/no comprimidos) por la herramienta ROHC Sniffer. Debido a que los resultados que se obtienen ejecutando dos pruebas diferentes en dos sesiones independientes, los resultados varían ligeramente entre sí. Por ejemplo, el número de paquetes de tipo IR ROHC en el caso de la ROHCv1 es 70, mientras que en el caso de la ROHCv2 es 65. Se puede afirmar que las pruebas se realizaron bajo condiciones de operación similares.

La Tabla 5.3 muestra las ganancias de compresión obtenidas para la aplicación utilizando ambas versiones de ROHC. Como se puede ver en la tabla, las ganancias de compresión tanto para la cabecera del paquete como para el paquete son muy buenas. Dado que la longitud del paquete de audio es corta, un total de 925 KBytes para 5000 paquetes en el caso del ROHCv1, el 91% de la ganancia de compresión de cabecera se traduce en un 25% en la longitud del paquete.

ROHC versión	Ganancia de compresión paquete	Ganancia de compresion cabecera
ROCHv1	25%	91%
ROHCv2	25%	91%

Tabla 5.3 Resultados audio conferencia Hangouts – ROHC Sniffer

Ambas figuras, Figura 5.21 y Figura 5.22, también muestran las siguientes métricas de interés:

- En ambos casos, el modo de funcionamiento a lo largo de toda la comunicación es U-mode y el perfil UDP/IP
- El compresor genera 70 y 65 paquetes IR cuyo porcentaje sobre el total de los paquetes ROHC representa el 1% del total de paquetes emitidos.
- El 98% de los paquetes enviados por el compresor corresponden a paquetes con alto índice de compresión. En otras palabras el compresor funciona durante la mayor parte de la comunicación en el estado de alta compresión SO.
- En el caso de la versión ROHCv1, la gran mayoría de paquetes ROCH corresponden a paquetes UO-0 cuyo propósito es actualizar el numero se secuencia (SN) incluido en la cabecera RTP e incluye un campo CRC de 3 bits para proteger el SN.
- En el caso de la versión ROHCv2, las gran mayoría de paquetes ROHC corresponden al tipo pt_crc3, los cuales incluyen en la cabecera un campo CRC de 3 bits protegiendo varios campos de la cabecera. Esto ofrece mayor robustez que el paquete UO-0.


```
Archivo Editar Pestañas Ayuda
all ROHC sniffer statistics dumped
=====
packet #5000
=====
dump ROHC sniffer statistics...
general:
total packets: 5000 packets
bad packets: 0 packets (0%)
loss (estim.):
  0 packets among 0 bursts (0%)
  packets per burst: max 0, avg 0, min 0
mis-ordered packets (estim.): 0 packets (0%)
duplicated packets (estim.): 0 packets (0%)
compression gain:
pre-compress: 925 KB (incl. 249 KB of headers)
post-compress: 699 KB (incl. 24 KB of headers)
compress ratio: 75% of total packets, ie. 25% of gain on full packets
compress ratio: 9% of total headers, ie. 91% of gain on headers alone
used and re-used contexts: 29
packets per profile:
ROHCv1 Uncompressed profile: 0 packets (0%)
ROHCv1 IP/UDP/RTP profile: 0 packets (0%)
ROHCv1 IP/UDP profile: 4561 packets (91%)
ROHCv1 IP-only profile: 6 packets (0%)
ROHCv1 IP/TCP profile: 428 packets (8%)
ROHCv1 IP/UDP-Lite profile: 0 packets (0%)
ROHCv2 IP/UDP profile: 0 packets (0%)
ROHCv2 IP/ESP profile: 0 packets (0%)
ROHCv2 IP-only profile: 0 packets (0%)
packets per mode:
U-mode: 4995 packets (100%)
O-mode: 0 packets (0%)
R-mode: 0 packets (0%)
packets per state:
IR state: 70 packets (1%)
FO state: 27 packets (0%)
SO state: 4898 packets (98%)
packets per packet type:
packet type IR: 70 packets (1%)
packet type IR-DYN: 8 packets (0%)
packet type ROHCv1/UO-0: 4521 packets (90%)
packet type ROHCv1/non-RTP/UO-1: 0 packets (0%)
packet type ROHCv1/RTP/UO-1-ID: 0 packets (0%)
packet type ROHCv1/RTP/UO-1-TS: 0 packets (0%)
packet type ROHCv1/RTP/UO-1: 0 packets (0%)
packet type ROHCv1/non-RTP/UOR-2: 21 packets (0%)
packet type ROHCv1/RTP/UOR-2: 0 packets (0%)
packet type ROHCv1/RTP/UOR-2-ID: 0 packets (0%)
packet type ROHCv1/RTP/UOR-2-TS: 0 packets (0%)
packet type ROHCv1/Uncomp/Normal: 0 packets (0%)
packet type co_common: 134 packets (2%)
packet type ROHCv1/TCP/rnd_1: 115 packets (2%)
packet type ROHCv1/TCP/rnd_2: 0 packets (0%)
packet type ROHCv1/TCP/rnd_3: 44 packets (0%)
packet type ROHCv1/TCP/rnd_4: 7 packets (0%)
packet type ROHCv1/TCP/rnd_5: 53 packets (1%)
packet type ROHCv1/TCP/rnd_6: 0 packets (0%)
packet type ROHCv1/TCP/rnd_7: 20 packets (0%)
packet type ROHCv1/TCP/rnd_8: 2 packets (0%)
packet type ROHCv1/TCP/seq_1: 0 packets (0%)
packet type ROHCv1/TCP/seq_2: 0 packets (0%)
packet type ROHCv1/TCP/seq_3: 0 packets (0%)
packet type ROHCv1/TCP/seq_4: 0 packets (0%)
packet type ROHCv1/TCP/seq_5: 0 packets (0%)
packet type ROHCv1/TCP/seq_6: 0 packets (0%)
packet type ROHCv1/TCP/seq_7: 0 packets (0%)
packet type ROHCv1/TCP/seq_8: 0 packets (0%)
packet type ROHCv1/IR-CR: 0 packets (0%)
packet type ROHCv2/co_repair: 0 packets (0%)
packet type ROHCv2/pt_0_crc3: 0 packets (0%)
packet type ROHCv2/non-RTP/pt_0_crc7: 0 packets (0%)
```

Figura 5.21 Audio conferencia Hangouts – resultados ROHC Sniffer - ROHCv1


```
=====  
packet #5000  
=====  
dump ROHC sniffer statistics...  
general:  
  total packets: 5000 packets  
  bad packets: 0 packets (0%)  
  loss (estim.):  
    0 packets among 0 bursts (0%)  
  packets per burst: max 0, avg 0, min 0  
  mis-ordered packets (estim.): 0 packets (0%)  
  duplicated packets (estim.): 0 packets (0%)  
compression gain:  
  pre-compress: 931 KB (incl. 251 KB of headers)  
  post-compress: 704 KB (incl. 24 KB of headers)  
  compress ratio: 75% of total packets, ie. 25% of gain on full packets  
  compress ratio: 9% of total headers, ie. 91% of gain on headers alone  
  used and re-used contexts: 27  
packets per profile:  
  ROHCv1 Uncompressed profile: 0 packets (0%)  
  ROHCv1 IP/UDP/RTP profile: 0 packets (0%)  
  ROHCv1 IP/UDP profile: 0 packets (0%)  
  ROHCv1 IP-only profile: 0 packets (0%)  
  ROHCv1 IP/TCP profile: 474 packets (9%)  
  ROHCv1 IP/UDP-Lite profile: 0 packets (0%)  
  ROHCv2 IP/UDP profile: 4523 packets (90%)  
  ROHCv2 IP/ESP profile: 0 packets (0%)  
  ROHCv2 IP-only profile: 2 packets (0%)  
packets per mode:  
  U-mode: 4999 packets (100%)  
  O-mode: 0 packets (0%)  
  R-mode: 0 packets (0%)  
packets per state:  
  IR state: 65 packets (1%)  
  FO state: 18 packets (0%)  
  SO state: 4916 packets (98%)  
packets per packet type:  
  packet type IR: 65 packets (1%)  
  packet type IR-DYN: 0 packets (0%)  
  packet type ROHCv1/UO-0: 0 packets (0%)  
  packet type ROHCv1/non-RTP/UO-1: 0 packets (0%)  
  packet type ROHCv1/RTP/UO-1-ID: 0 packets (0%)  
  packet type ROHCv1/RTP/UO-1-TS: 0 packets (0%)  
  packet type ROHCv1/RTP/UO-1: 0 packets (0%)  
  packet type ROHCv1/non-RTP/UOR-2: 0 packets (0%)  
  packet type ROHCv1/RTP/UOR-2: 0 packets (0%)  
  packet type ROHCv1/RTP/UOR-2-ID: 0 packets (0%)  
  packet type ROHCv1/RTP/UOR-2-TS: 0 packets (0%)  
  packet type ROHCv1/Uncomp/Normal: 0 packets (0%)  
  packet type co_common: 167 packets (3%)  
  packet type ROHCv1/TCP/rnd_1: 123 packets (2%)  
  packet type ROHCv1/TCP/rnd_2: 2 packets (0%)  
  packet type ROHCv1/TCP/rnd_3: 45 packets (0%)  
  packet type ROHCv1/TCP/rnd_4: 8 packets (0%)  
  packet type ROHCv1/TCP/rnd_5: 50 packets (1%)  
  packet type ROHCv1/TCP/rnd_6: 0 packets (0%)  
  packet type ROHCv1/TCP/rnd_7: 20 packets (0%)  
  packet type ROHCv1/TCP/rnd_8: 8 packets (0%)  
  packet type ROHCv1/TCP/seq_1: 0 packets (0%)  
  packet type ROHCv1/TCP/seq_2: 0 packets (0%)  
  packet type ROHCv1/TCP/seq_3: 0 packets (0%)  
  packet type ROHCv1/TCP/seq_4: 0 packets (0%)  
  packet type ROHCv1/TCP/seq_5: 0 packets (0%)  
  packet type ROHCv1/TCP/seq_6: 0 packets (0%)  
  packet type ROHCv1/TCP/seq_7: 0 packets (0%)  
  packet type ROHCv1/TCP/seq_8: 0 packets (0%)  
  packet type ROHCv1/IR-CR: 0 packets (0%)  
  packet type ROHCv2/co_repair: 0 packets (0%)  
  packet type ROHCv2/pt_0_crc3: 4493 packets (89%)  
  packet type ROHCv2/non-RTP/pt_0_crc7: 18 packets (0%)  
  packet type ROHCv2/non-RTP/pt_1_seq_id: 0 packets (0%)
```

Figura 5.22 Audio conferencia Hangouts – resultados ROHC Sniffer– ROHCv2

5.3.3 Video conferencia – Hangouts

La Figura 5.23 y Figura 5.24 muestran los resultados obtenidos para las aplicaciones de videoconferencia obtenidas utilizando la herramienta Sniffer ROHCv1 y ROHCv2, respectivamente. Como se puede ver en la figura, 5000 paquetes UDP/IP han sido capturados y procesados (comprimidos/no comprimidos) por la herramienta ROHC Sniffer. Al igual que en la aplicación de audio-conferencia, los resultados varían ligeramente entre sí, ya que se obtuvieron utilizando dos pruebas diferentes

La Tabla 5.4 muestra la ganancia de compresión obtenida para la aplicación de videoconferencia. Como se puede ver en la figura, durante el período de observación se han recolectado 5000 paquetes UDP/IP utilizando la herramienta de rastreo ROHC. Los resultados de la videoconferencia son similares a los obtenidos en el caso anterior. Para ambas versiones de ROHC, la ganancia de compresión de la cabecera de paquete es del 91 al 92%. Sin embargo, la compresión a nivel de paquete no es tan efectiva como para el caso de audio. Esto se debe a que la carga útil del paquete de vídeo es mayor que la del paquete de audio. Es decir, en el caso de la videoconferencia, como se transporta tanto información de video como de audio, la ganancia global de compresión se reduce significativamente, un 6%.

ROHC versión	Ganancia de compresión paquete	Ganancia de compresión cabecera
ROHCv1	6%	91%
ROHCv2	6%	92%

Tabla 5.4 Video conferencia– ROHC Sniffer

Al igual que para el caso de la audioconferencia Google Hangouts, los resultados mostrados en la Figura 5.23 y la Figura 5.24 proporcionan detalles del funcionamiento del mecanismo de compresión para ambas versiones.

Los principales detalles del funcionamiento pueden resumirse en:

- En ambos casos, el modo de funcionamiento a lo largo de toda la comunicación es U-mode y el perfil UDP/IP
- El número de paquetes IR representa el 1% del total de paquetes emitidos.
- El 98% de los paquetes enviados por el compresor corresponden a paquetes con alto índice de compresión. El compresor funciona durante la mayor parte de la comunicación en el estado de alta compresión SO.
- En el caso de la versión ROHCv1, la gran mayoría de paquetes ROHC corresponden a paquetes UO-0 cuyo propósito es actualizar el número de secuencia (SN) incluido en la cabecera RTP e incluye un campo CRC de 3 bits para proteger el SN.
- En el caso de la versión ROHCv2, la gran mayoría de paquetes ROHC corresponden al tipo pt_crc3, los cuales incluyen en la cabecera un campo CRC de 3 bits protegiendo varios campos de la cabecera. Esto ofrece mayor robustez que los paquetes UO-0.

```
pi@raspberrypi: ~
pi@raspberrypi: ~
pi

Archivo  Editar  Pestañas  Ayuda

=====
packet #5000
=====
dump ROHC sniffer statistics...
general:
  total packets: 5000 packets
  bad packets: 0 packets (0%)
  loss (estim.):
    0 packets among 0 bursts (0%)
    packets per burst: max 0, avg 0, min 0
  mis-ordered packets (estim.): 0 packets (0%)
  duplicated packets (estim.): 0 packets (0%)
compression gain:
  pre-compress: 3947 KB (incl. 242 KB of headers)
  post-compress: 3727 KB (incl. 22 KB of headers)
  compress ratio: 94% of total packets, ie. 6% of gain on full packets
  compress ratio: 9% of total headers, ie. 91% of gain on headers alone
  used and re-used contexts: 9
packets per profile:
  ROHCv1 Uncompressed profile: 0 packets (0%)
  ROHCv1 IP/UDP/RTP profile: 0 packets (0%)
  ROHCv1 IP/UDP profile: 4891 packets (97%)
  ROHCv1 IP-only profile: 0 packets (0%)
  ROHCv1 IP/TCP profile: 108 packets (2%)
  ROHCv1 IP/UDP-Lite profile: 0 packets (0%)
  ROHCv2 IP/UDP profile: 0 packets (0%)
  ROHCv2 IP/ESP profile: 0 packets (0%)
  ROHCv2 IP-only profile: 0 packets (0%)
packets per mode:
  U-mode: 4999 packets (100%)
  O-mode: 0 packets (0%)
  R-mode: 0 packets (0%)
packets per state:
  IR state: 29 packets (0%)
  FO state: 25 packets (0%)
  SO state: 4945 packets (98%)
packets per packet type:
  packet type IR: 29 packets (0%)
  packet type IR-DYN: 4 packets (0%)
  packet type ROHCv1/UO-0: 4851 packets (97%)
  packet type ROHCv1/non-RTP/UO-1: 0 packets (0%)
  packet type ROHCv1/RTP/UO-1-ID: 0 packets (0%)
  packet type ROHCv1/RTP/UO-1-TS: 0 packets (0%)
  packet type ROHCv1/RTP/UO-1: 0 packets (0%)
  packet type ROHCv1/non-RTP/UOR-2: 21 packets (0%)
  packet type ROHCv1/RTP/UOR-2: 0 packets (0%)
  packet type ROHCv1/RTP/UOR-2-ID: 0 packets (0%)
  packet type ROHCv1/RTP/UOR-2-TS: 0 packets (0%)
  packet type ROHCv1/Uncomp/Normal: 0 packets (0%)
  packet type co_common: 19 packets (0%)
  packet type ROHCv1/TCP/rnd_1: 28 packets (0%)
  packet type ROHCv1/TCP/rnd_2: 0 packets (0%)
  packet type ROHCv1/TCP/rnd_3: 27 packets (0%)
  packet type ROHCv1/TCP/rnd_4: 0 packets (0%)
  packet type ROHCv1/TCP/rnd_5: 15 packets (0%)
  packet type ROHCv1/TCP/rnd_6: 0 packets (0%)
  packet type ROHCv1/TCP/rnd_7: 5 packets (0%)
  packet type ROHCv1/TCP/rnd_8: 0 packets (0%)
  packet type ROHCv1/TCP/seq_1: 0 packets (0%)
  packet type ROHCv1/TCP/seq_2: 0 packets (0%)
  packet type ROHCv1/TCP/seq_3: 0 packets (0%)
  packet type ROHCv1/TCP/seq_4: 0 packets (0%)
  packet type ROHCv1/TCP/seq_5: 0 packets (0%)
  packet type ROHCv1/TCP/seq_6: 0 packets (0%)
  packet type ROHCv1/TCP/seq_7: 0 packets (0%)
  packet type ROHCv1/TCP/seq_8: 0 packets (0%)
  packet type ROHCv1/IR-CR: 0 packets (0%)
  packet type ROHCv2/co_repair: 0 packets (0%)
  packet type ROHCv2/pt_0_crc3: 0 packets (0%)
  packet type ROHCv2/non-RTP/pt_0_crc7: 0 packets (0%)
  packet type ROHCv2/non-RTP/pt_1_seq_id: 0 packets (0%)
```

Figura 5.23 Video conferencia – resultados ROHC Sniffer - ROHCv1


```
Archivo Editar Pestañas Ayuda
packet type ROHCv2/RTP/pt_0_crc7: 0 packets (0%)
packet type ROHCv2/RTP/pt_1_rnd: 0 packets (0%)
packet type ROHCv2/RTP/pt_1_seq_id: 0 packets (0%)
packet type ROHCv2/RTP/pt_1_seq_ts: 0 packets (0%)
packet type ROHCv2/RTP/pt_2_rnd: 0 packets (0%)
packet type ROHCv2/RTP/pt_2_seq_id: 0 packets (0%)
packet type ROHCv2/RTP/pt_2_seq_ts: 0 packets (0%)
packet type ROHCv2/RTP/pt_2_seq_both: 0 packets (0%)
all ROHC sniffer statistics dumped
=====
packet #5000
=====
dump ROHC sniffer statistics...
general:
total packets: 5000 packets
bad packets: 0 packets (0%)
loss (estim.):
0 packets among 0 bursts (0%)
  packets per burst: max 0, avg 0, min 0
mis-ordered packets (estim.): 0 packets (0%)
duplicated packets (estim.): 0 packets (0%)
compression gain:
pre-compress: 4036 KB (incl. 242 KB of headers)
post-compress: 3815 KB (incl. 20 KB of headers)
compress ratio: 94% of total packets, ie. 6% of gain on full packets
compress ratio: 8% of total headers, ie. 92% of gain on headers alone
used and re-used contexts: 32
packets per profile:
ROHCv1 Uncompressed profile: 0 packets (0%)
ROHCv1 IP/UDP/RTP profile: 0 packets (0%)
ROHCv1 IP/UDP profile: 0 packets (0%)
ROHCv1 IP-only profile: 0 packets (0%)
ROHCv1 IP/TCP profile: 150 packets (3%)
ROHCv1 IP/UDP-Lite profile: 0 packets (0%)
ROHCv2 IP/UDP profile: 4844 packets (96%)
ROHCv2 IP/ESP profile: 0 packets (0%)
ROHCv2 IP-only profile: 1 packets (0%)
packets per mode:
U-mode: 4995 packets (100%)
O-mode: 0 packets (0%)
R-mode: 0 packets (0%)
packets per state:
IR state: 80 packets (1%)
F0 state: 18 packets (0%)
S0 state: 4897 packets (98%)
packets per packet type:
packet type IR: 80 packets (1%)
packet type IR-DYN: 6 packets (0%)
packet type ROHCv1/U0-0: 0 packets (0%)
packet type ROHCv1/non-RTP/U0-1: 0 packets (0%)
packet type ROHCv1/RTP/U0-1-ID: 0 packets (0%)
packet type ROHCv1/RTP/U0-1-TS: 0 packets (0%)
packet type ROHCv1/RTP/U0-1: 0 packets (0%)
packet type ROHCv1/non-RTP/UOR-2: 0 packets (0%)
packet type ROHCv1/RTP/UOR-2: 0 packets (0%)
packet type ROHCv1/RTP/UOR-2-ID: 0 packets (0%)
packet type ROHCv1/RTP/UOR-2-TS: 0 packets (0%)
packet type ROHCv1/Uncomp/Normal: 0 packets (0%)
packet type co_common: 89 packets (1%)
packet type ROHCv1/TCP/rnd_1: 9 packets (0%)
packet type ROHCv1/TCP/rnd_2: 0 packets (0%)
packet type ROHCv1/TCP/rnd_3: 0 packets (0%)
packet type ROHCv1/TCP/rnd_4: 0 packets (0%)
packet type ROHCv1/TCP/rnd_5: 7 packets (0%)
packet type ROHCv1/TCP/rnd_6: 0 packets (0%)
packet type ROHCv1/TCP/rnd_7: 27 packets (0%)
packet type ROHCv1/TCP/rnd_8: 0 packets (0%)
packet type ROHCv1/TCP/seq_1: 0 packets (0%)
packet type ROHCv1/TCP/seq_2: 0 packets (0%)
packet type ROHCv1/TCP/seq_3: 0 packets (0%)
packet type ROHCv1/TCP/seq_4: 0 packets (0%)
```

Figura 5.24 Video conferencia Hangouts – resultados ROHC Sniffer - ROHCv2

5.4 Resumen de resultados

En este capítulo se ha descrito la plataforma experimental y se han analizado los resultados de las prestaciones de los mecanismos de compresión. Antes de resumir los principales resultados experimentales es importante señalar las características y limitaciones de la plataforma experimental.

La librería ROHC utilizada en la fase experimental se ha venido desarrollando desde hace más de 15 años. Durante este periodo la librería ha sido modificada debido a:

- La definición de una nueva versión de los mecanismos ROHC. En 2008, la IETF publica una segunda versión de la norma ROHC denominada: ROHCv2. Dicha norma introduce nuevas funcionalidades (paquetes) con el fin de mejorar la robustez del mecanismo.
- La definición de una nueva norma del perfil ROHC TCP. En 2013, la IETF publica una nueva norma del perfil ROHC TCP eliminando a su vez la norma anterior. La definición de esta nueva norma se debió al creciente número de mecanismos de control de flujo y error de TCP, así como el uso creciente de TCP en las redes móviles. Este cambio introdujo un gran número de paquetes ROHC descritos en la Tabla 3.5 en el Capítulo 3.

La librería ROHC implementa solo un subconjunto de las funcionalidades de ROHC, la relación de los funcionalidades y planes de desarrollo pueden consultarse en [25].

Teniendo en cuenta lo anterior, los resultados nos permiten hacer las siguientes observaciones.

- Entre las cinco aplicaciones diferentes consideradas en este proyecto, el uso de ROHC es de gran ayuda para la aplicación de streaming YouTube, para la cual se obtiene la mejor ganancia de compresión, ver Tabla 5.5. Para esta, hay que tener en cuenta que se emplea el protocolo DASH (Dynamic Adaptive Streaming over HTTP), el cual hace uso de la segmentación, transmitiendo así paquetes audio y video de tamaño pequeño. Esto tiene una relación directa con el tamaño de la carga útil lo que confirma que el uso de ROHC es de gran utilidad en el caso de aplicaciones que hacen uso de paquetes cortos.
- En cuanto a los resultados audio y vídeo streaming Google Hangouts, estos confirman una vez los beneficios de utilizar ROHC para las comunicaciones de voz.
- Las implementaciones de ROHCv1 y ROHCv2 utilizad en la segunda parte de las pruebas experimentales s han confirmado que presentan similares resultados en términos de la ganancia y la ratio de compresión. En cuanto a la robustez, ROHCv2 debe ofrecer mejores prestaciones que ROHCv1. La versión 2 hace uso de paquetes pt_0-crc3 que incluyen un CRC de 3 bits protegiendo varios campos de la cabecera. Por el contrario, ROHCv1 utiliza paquetes UO-0 que también incluyen un CRC de 3 bits, pero solo para proteger el número de secuencia. En general, ROHCv2 ofrece mejores garantías que ROHCv1, pero su complejidad computacional es mayor [28].
- La aplicación Spotify no se ha beneficiado del uso de mecanismo ROHC. Las causas detrás de las malas prestaciones en este caso no han podido ser identificadas.

Los resultados de este trabajo pueden guiar a los desarrolladores en identificar errores de implementación, así como en la evaluación de los mecanismos de la norma ROHC.

Aplicación	IP Protocols	Ganancia (paquete)
Youtube Video streaming	TCP/IPv6	16%
Youtube Video retrieval	TCP/IP	5%
Spotify	TCP/IP	6.8%
Hangouts	UDP/IP	6.4%
VoIP G.711	RTP/UDP/IP	10.7%

Tabla 5.5 Ganancias de compresión obtenidas por aplicación

Capítulo 6. CONCLUSIONES Y TRABAJOS FUTUROS

Teniendo en cuenta los objetivos del proyecto, las principales tareas desarrolladas pueden resumirse brevemente como sigue:

- Estudio del esquema de compresión de cabeceras de paquetes. He revisado la motivación, la evolución y los mecanismos de compresión de cabeceras de paquetes más relevantes. El principal resultado del aprendizaje ha sido familiarizarse con la terminología, los principales estándares de compresión de las cabeceras de paquetes y algunos trabajos de investigación.
- Herramientas de software: Me he familiarizado con una implementación de código abierto del mecanismo de ROHC. A lo largo del análisis, me he dado cuenta de que el diseño y desarrollo de la norma ROHC es una tarea muy difícil. Como se indica en el sitio web de la librería ROHC, el desarrollo de la biblioteca de ROHC se inició en el año 2003 en la Universidad de Lulea (Suecia). Desde entonces varias versiones han sido puestas a disposición de la comunidad.
- Desarrollo de una plataforma ROHC experimental. El diseño y desarrollo de una plataforma experimental sufrió varias modificaciones a lo largo del proyecto. Los planes iniciales consistían en el desarrollo de una plataforma interconectando dos nodos a fin de evaluar el mecanismo ROHC. Sin embargo, después de estudiar el código y debido a la complejidad del desarrollo de tal plataforma, el objetivo se redirigió a analizar las prestaciones de la librería ROHC utilizando para ello diversas aplicaciones móviles.
- Evaluación y análisis de resultados. Esta tarea se dividió en dos partes. Primeramente, se analizaron los resultados obtenidos utilizando las herramientas STAT y Sniffer de la librería ROHC. Las métricas de interés fueron la ratio y ganancia de compresión. Ambas métricas se evaluaron a nivel de la cabecera y a nivel de paquete por considerar este último parámetro de particular relevancia.

En una segunda parte se ha analizado el impacto y las dificultades que los mecanismos tienen en realizar la compresión de ciertos paquetes. En particular se ha analizado las trazas y prestaciones de las aplicaciones que mejores y peores resultados han obtenido. Este tipo de análisis es de gran interés para los desarrolladores de los mecanismos de compresión. Estos resultados permiten guiar los esfuerzos de mejora y desarrollo de los diferentes mecanismos de compresión especificados en los estándares.

En cuanto a los principales resultados obtenidos del análisis de los resultados experimentales:

- Entre las cinco aplicaciones estudiadas, se obtuvieron las mejores ganancias en la cabecera del paquete y en la compresión del paquete para las aplicaciones de voz y de vídeo streaming (Youtube). Esto se debe a que la carga útil que transportan los paquetes es más pequeña por la utilizada en las otras aplicaciones.
- ROHCv1 y ROHCv2 reportan las mismas ganancias en la cabecera del paquete y en la compresión del paquete. En [28], los autores llegaron a conclusiones similares. También llevaron a cabo un análisis de complejidad computacional de ambas versiones de ROHC. Encontraron que ROHCv2 tiene mayores demandas computacionales. Esta complejidad adicional puede deberse a la mayor robustez de los errores y pérdidas incluidos en la norma ROHCv2.

En cuanto al desarrollo y los estudios futuros, el ROHC puede ser útil para el desarrollo de aplicaciones de las IoT. Hoy en día se están desplegando un gran número de dispositivos IoT. La mayoría de los datos capturados y transmitidos por las aplicaciones de IoT constan de pocos bytes. El uso de mecanismos de compresión de cabeceras puede reducir el ancho de banda y las necesidades de energía de dichas aplicaciones [29].

En la Universidad de Aalto donde he realizado la mayor parte de mi trabajo se están estudiando los trabajos y proyectos de normas actuales para hacer posible la integración de las tecnologías



IoT en las infraestructuras de las redes móviles (C-IoT). Este tema es de particular relevancia para las operadoras de telecomunicaciones [30].

Entre las tareas de desarrollo de investigación en el desarrollo de ROHC se encuentran:

- Evaluación de ROHC en la presencia de errores. Este tema es de particular relevancia debido a que ROHC deberá implantarse en sistemas móviles y aplicaciones IoT. Las condiciones de operación y las tasas de errores pueden variar mucho en función del ámbito de aplicación. Por ejemplo, el uso y ámbitos de aplicación de los dispositivos IoT no para de crecer. Los mecanismos ROHC deberán satisfacer los tres KPIs mencionados en el Capítulo 3.
- Evaluación del consumo energético. El consumo energético es un parámetro de gran relevancia en los dispositivos móviles e IoT. Esto se debe al hecho que todos estos sistemas dispositivos son por lo general alimentados por pilas. El consumo energético es un parámetro importante de diseño en el desarrollo de los mecanismos ROHC a implantar en los dispositivos móviles e IoT.



BIBLIOGRAFÍA

- [1] EFFNET AB, An introduction to IP header compression, White Paper, 2004.
- [2] RFC 3095, ROBust Header Compression (ROHC): Framework and four profiles: RTP, UDP, ESP, and uncompressed.
- [3] V. Jacobson, "Compressing TCP/IP Headers, Internet Engineering Task Force (IETF), Request for Comments: 1144, Category: Standards Track, pp. 1-46.
- [4] S. J. Perkins and M. W. Mutka, "Dependency Removal for Transport Protocol Header Compression over Noisy Channels," in Proc. of IEEE International Conference on Communications (ICC), vol. 2, Montreal, Canada, June 1997, pp. 1025– 1029.
- [5] A. Calveras, M. Arnau, and J. Paradells, "A controlled Overhead for TCP/IP Header Compression Algorithm over Wireless Links," in Proc. of the 11th International Conference on Wireless Communications (Wireless'99), Calgary, Canada, 1999.
- [6] M. Rossi, A. Giovanardi, M. Zorzi, G. Mazzini, "TCP/IP header compression: proposal and performance investigation on a WCDMA air interface," Proceeding of the 12th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications. PIMRC 2001, vol. 1, Sept. 2001, pp. 78–82.
- [7] M. Dagermark, B. Nordegren, S. Pink, IP Header Compression, Internet Engineering Task Force (IETF), Request for Comments: 2507, Category: Standards Track, pp. 1-47.
- [8] M. Degermak, M. Engan, B. Nordgren, and S. Pink, "Low-loss TCP/IP header compression for wireless networks," in Proceedings of ACM MobiCom '96, vol. 3, New York, New York, Oct. 1997, pp. 375–387.
- [9] C. Perkins and J. Crowcroft, "Effects of interleaving on RTP header compression," in Proceedings of IEEE Infocom 2000, Tel Aviv, Israel, 2000, pp. 111–117.
- [10] J. Lim and H. Stern, "IPv6 header compression algorithm supporting mobility in wireless networks," in Proceedings of the Southeastcon 2000, 2000, pp. 535–540.
- [11] M. Degermark, H. Hannu, L. Jonsson, and K. Svanbro, "Evaluation of CRTP performance over cellular radio links," IEEE Personal Communications, vol. 7, no. 4, pp. 20–25, 2000.
- [12] K. Svanbro, H. Hannu, L.-E. Jonsson, and M. Degermark, "Wireless Real-time IP Services Enabled by Header Compression," in Proceedings of the IEEE Vehicular Technology Conference (VTC), vol. 2, Tokyo, Japan, 2000, pp. 1150–1154.
- [13] A. Cellatoglu, S. Fabri, S. Worrall, A. Sadka, and A. Kondo, "Robust header compression for real-time services in cellular networks," in Proceedings of the IEE 3G 2001, London, GB, Mar. 2001, pp. 124–128.
- [14] W.-T. Chen, D.-W. Chuang, and H.-C. Hsiao, "Enhancing CRTP by retransmission for wireless networks," in Proceedings of the Tenth International Conference on Computer Communications and Networks, 2001, pp. 426–431.
- [15] L. Khiem, C. Clanton, L. Zhigang, and Z. Haihong, "Efficient and robust header compression for real-time services," in Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC), vol. 2, Chicago, IL, 2000, pp. 924–928.
- [16] Z. Kostic, Q. Xiaoxin, and L. Chang, "Impact of TCP/IP header compression on the performance of a cellular system," in Proceedings of the Wireless Communications and Networking Conference, vol. 1, Chicago, IL, 2000, pp. 281–286.
- [17] H. Liao, Q. Zhang, W. Zhu, and Y.-Q. Zhang, "A robust TCP/IP header compression scheme for wireless networks," in Proceedings of the IEEE International Conference on 3G Wireless and Beyond, San Francisco, CA, June 2001.
- [18] G. Boggia, P. Camarda, and V.G. Squeo, "ROHC+: A New Header Compression Scheme for TCP Streams in 3G Wireless Systems," in Proceedings of the IEEE International Conference on Communications (ICC), vol. 5, 2002, pp. 3271–3278.



- [19] C. Jiao, L. Schwiebert, and G. Richard, “Adaptive header compression for wireless networks,” in Proceedings of the 26th Annual IEEE Conference on Local Computer Networks, Nov. 2001, pp. 377–378.
- [20] G. Pelletier, Q. Zhang, L.-E. Jonsson, H. Liao, and M. West, “RObust Header Compression(ROHC): TCP/IP Profile (ROHC-TCP),” Nov. 2002, RFC–draft, work in progress.
- [21] R. Price, R. Hancock, S. McCann, M. A. West, A. Surtees, P. Ollis, Q. Zhang, H. Liao, W. Zhu, and Y.-Q. Zhang, “TCP/IP compression for ROHC, proposed standard, draft-ietf-rohc-tcp-epic-02.txt,” Nov. 2001.
- [22] J. Lilley, J. Yang, H. Balakrishnan, and S. Seshan, “A unified header compression framework for low-bandwidth links,” in Proceedings of ACM MobiCom, 2000, pp. 131–142.
- [23] G. Pelletier and K. Sandlund, "RObust Header Compression Version 2 (ROHCv2): Profiles for RTP, UDP, IP, ESP and UDP-Lite", RFC 5225, April 2008.
- [24] Pelletier, G., K. Sandlund and L-E. Jonsson, “RObust Header Compression (ROHC): A Profile for TCP/IP (ROHC-TCP), RFC 6846, January 2013.
- [25] <https://rohc-lib.org/>
- [26] <https://www.wireshark.org/>
- [27] <https://www.rstudio.com/>
- [28] M. Tömösközi, P. Seeling, P. Ekler, and H.P Fitzek Frank, “Performance evaluation and implementation of IP and robust header compression schemes for TCP and UDP traffic in static and dynamic wireless contexts,” Computer Science and Information Systems 2017 Vol. 14, Issue 2, pp: 283-308, 2017.
- [29] P. Koskela, M. Majanen and M. Valta, “Packet Header Compression for the Internet of Things”, Sensors & Transducers, Vol. 196, Issue 1, January 2016, pp. 43-51.
- [30] http://www.effnet.com/solutions/cellular_iot/