EFFNET AB

WHITE PAPER

The concept of robust header compression, ROHC

FEBRUARY 2004

WWW.EFFNET.COM

The concept of robust header compression, ROHC

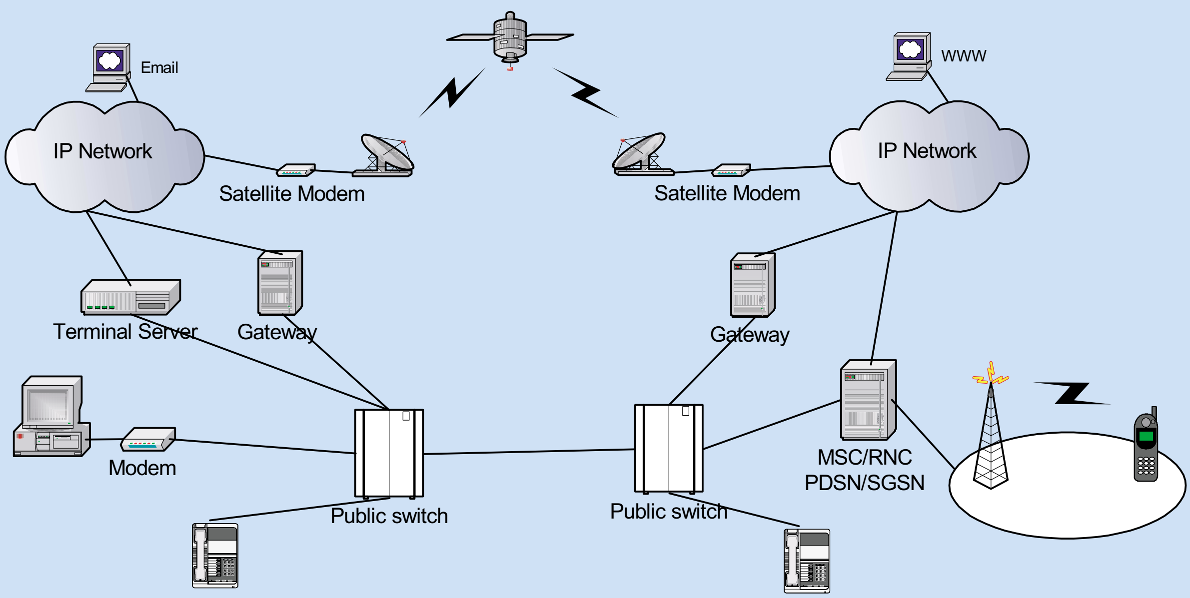
CONTENTS

The concept of IP header compression

The Internet Protocol (IP) is the choice of transport protocol on both wired and wireless networks and this choice is leading to the convergence of telecommunication and data networks. These converged networks will be the building blocks of the All-IP vision.

As the networks evolve to provide more bandwidth, the applications, services and the consumers of those applications and services, all compete for that bandwidth. For network operators it is important to offer a high quality of service (QoS) in order to attract more customers and encourage them to use their network as much as possible, thus achieving higher average revenue per user (ARPU).

The chain is as strong as its weakest link



In many services and applications e.g., Voice over IP, interactive games, messaging etc, the payload of the IP packets is almost of the same size or even smaller than the header. Over the end-to-end connection, comprised of multiple hops, these protocol headers are extremely important but over just one link (hop-to-hop) these headers can be compressed (and must be uncompressed at the other end of the link). It is possible to compress those headers, providing in many cases more than 90% savings, and thus save the bandwidth and use the expensive resource efficiently. IP header compression also provides other important benefits, such as reduction in packet loss and improved interactive response time.

In short, IP header compression is the process of compressing excess protocol headers before transmitting them on a link and uncompressing them to their original state on reception at the other end of the link. It is possible to compress the protocol headers due to the redundancy in header fields of the same packet as well as consecutive packets of the same packet stream.

To learn more about IP header compression, please see the Effnet white paper “An introduction to IP header compression” at www.effnet.com

Why one more IP header compression scheme?

The original Van Jacobson compression scheme was developed to increase the performance of IP/TCP flows over low bandwidth links such as PSTN. It does not even support compression of IP/UDP flows since at that time UDP traffic was very low. This scheme uses delta compression, sending only the difference in the value of the changing fields, to minimize the number of bits sent. It achieves compression from 40 bytes to on an average 4 bytes. It relies on the TCP recovery mechanism to recover from errors in the context due to bit errors and residual errors due to packet loss on the link. This scheme is obviously unsuitable for wireless links and multimedia applications.

The IPHC and the CRTP schemes made it possible to compress UDP as well as RTP traffic. They essentially use a similar mechanism of delta compression as the Van Jacobson compression scheme. Rather than depending on the TCP recovery mechanisms they add their own feedback mechanisms to recover from error conditions. They achieve compression up to 2 bytes. These schemes are suitable for wireless links with strong link layer checksum but are not robust enough to handle high bit error rates, high losses and long round trip times.

Since high BER and long RTT are common on 2.5G and 3G links, an efficient and robust compression scheme was needed. The ROHC scheme was developed to fulfill these criteria. It is an extendible framework of packet stream profiles e.g. IP/UDP/RTP, IP/ESP, IP/UDP and Uncompressed. As new profiles are defined e.g. IP/TCP, they can be easily added to the framework.

The ROHC scheme uses window based least significant bits encoding for the compression of dynamic fields in the protocol headers. Due to its feedback mechanism, ROHC is robust on wireless links with high BER and long RTT. It can achieve compression up to 1 byte and thus it is more efficient than other compression schemes. Even though it is complex compared to earlier schemes, it is suitable for wireless networks, which use the very expensive radio spectrum resource. To quote from RFC3095, “Bandwidth is the most costly resource in cellular links. Processing power is very cheap in comparison. Implementation or computational simplicity of a header compression scheme is therefore of less importance than its compression ratio and robustness.”

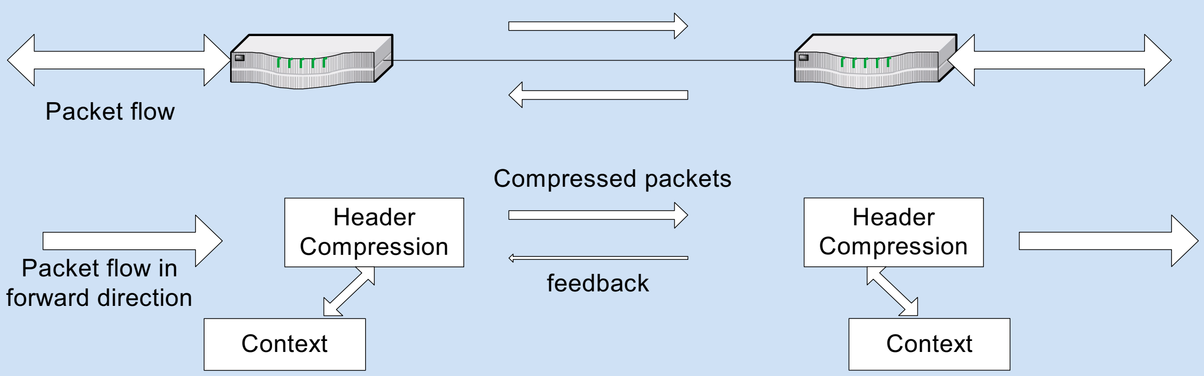
Robust Header Compression explained

The Robust Header Compression is an extensible framework consisting of the following profiles:

- **ROHC Uncompressed (Profile 0)**
  Compresses packets, which cannot be compressed by any of the following profiles.
- **ROHC RTP (Profile 1)**
  Compresses packets with IP/UDP/RTP protocol headers.
- **ROHC UDP (Profile 2)**
  Compresses packets with IP/UDP protocol headers.
- **ROHC ESP (Profile 3)**
  Compresses packets with IP/ESP protocol headers.

As described earlier, it is possible to compress the protocol headers due to the redundancy in the header fields of the same packet as well as consecutive packets of the same packet stream. A packet classifier can identify different packet streams by the combination of parameters like protocol headers being carried in the packet, the source and destination addresses and the source and destination ports etc. Initially, a few packets are sent uncompressed and are used to establish the context on both sides of the link. The context comprises information about static fields, dynamic fields and their change pattern in protocol headers. This information is used by the compressor to compress the packet as efficiently as possible and then by the decompressor to decompress the packet to its original state.

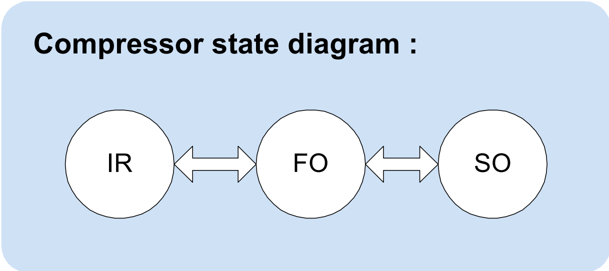
The concept of flow context in header compression:



The state of operation of the compressor is influenced by the characteristics of the packet flow like reordering or loss before compression, change pattern of fields in the headers either due to application behavior or the operating system. Similarly the state of operation of the decompressor is influenced by the link conditions like BER, RTT etc. In case of errors and the presence of the return channel, the decompressor sends feedback packets to the compressor and thus influences its state of operation. For each of the compressor and the decompressor three different states have been defined.

Compressor states

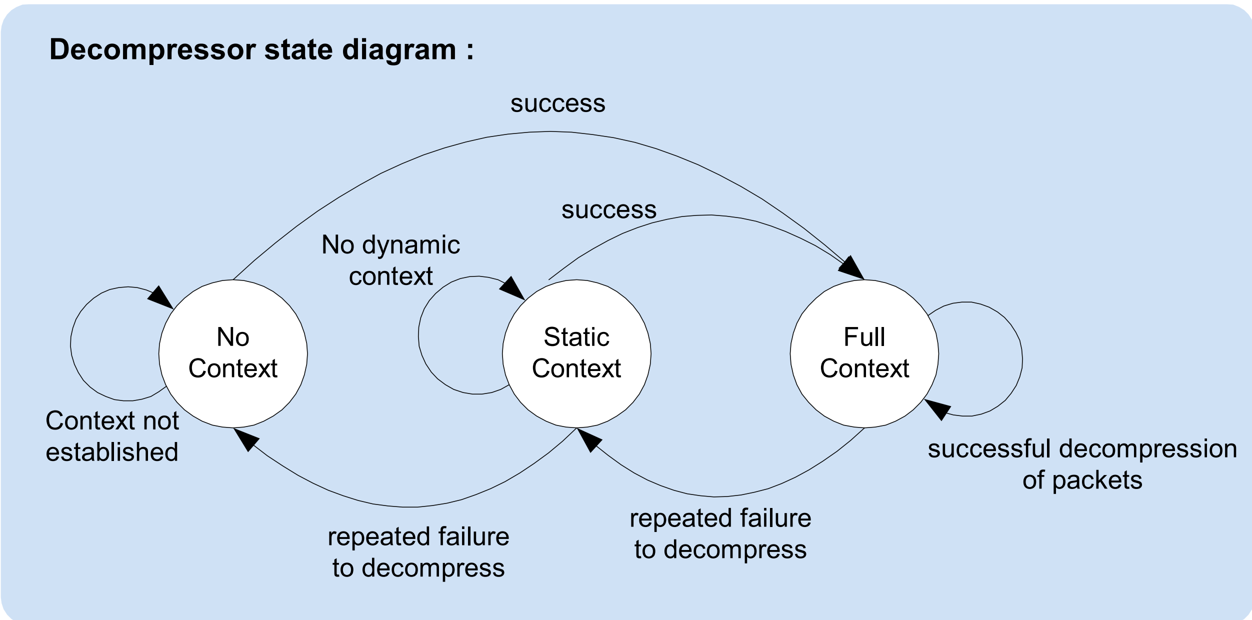
The ROHC compressor operates in 3 states: Initialization and Refresh (IR), First Order (FO) and Second Order (SO). The states describe the increasing level of confidence about the correctness of the context at the decompressor side. This confidence is reflected in the increasing compression of packet headers. In case of error conditions, as indicated by the decompressor using feedback packets, the compressor can move to a lower state to send packets that carry enough information to fix the error in the context of the decompressor. In some cases, the compressor periodically moves to a lower state of operation to ensure the context validity at the decompressor.



The compressor always starts in the IR state. In this state, it sends uncompressed packets to establish the context at the decompressor side. Once it gains the confidence that the decompressor has the context information, it moves to higher states of operation, either via FO state to SO state or directly to SO state. It dynamically changes its states to react to link conditions and error conditions as observed and reported by the decompressor.

Decompressor states

The ROHC decompressor operates in 3 states: No Context, Static Context and Full Context. The decompressor starts in the No Context state, as it has no context information available in the beginning of the packet flow. The successful decompression of an Initialization and Refresh packet (containing both static and dynamic information) from the compressor will create the context information at the decompressor side. At this point, the decompressor can move to the Full Context state as it has received both static and dynamic information. Once in the Full Context state, the decompressor moves to lower states only in error conditions. When moving to a lower state, it moves to the Static Context state and then hopefully can move back to the Full Context state by restoring the context by successfully decompressing FO state packets. If it still fails to decompress, it moves to the No Context state. In this case, the compressor needs to send IR packets to restore the context at the decompressor.

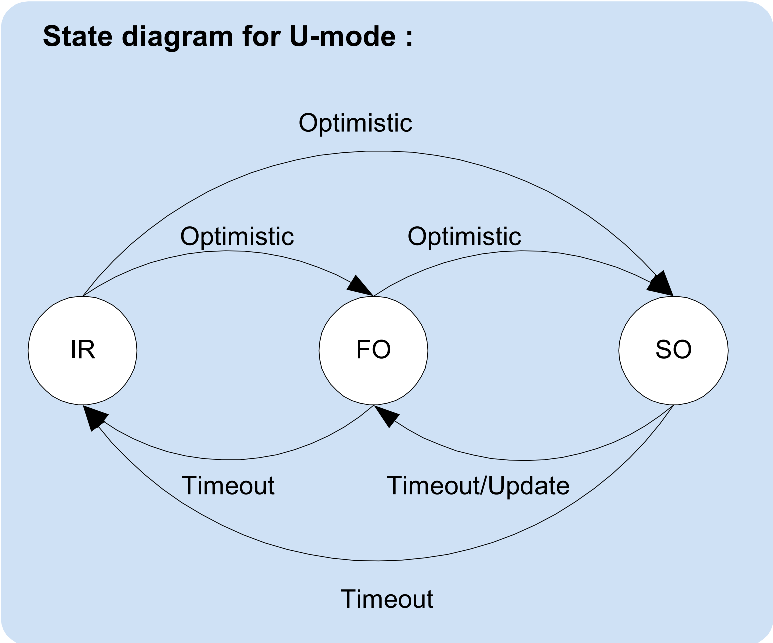


The ROHC framework defines 3 modes of operations. The selection of mode is based on parameters such as availability of feedback channel, error probabilities and header size variations. The operators of the network can choose not to use a channel to carry feedback packets for various reasons and thus limit the mode of operation of ROHC.

The following three modes have been defined:

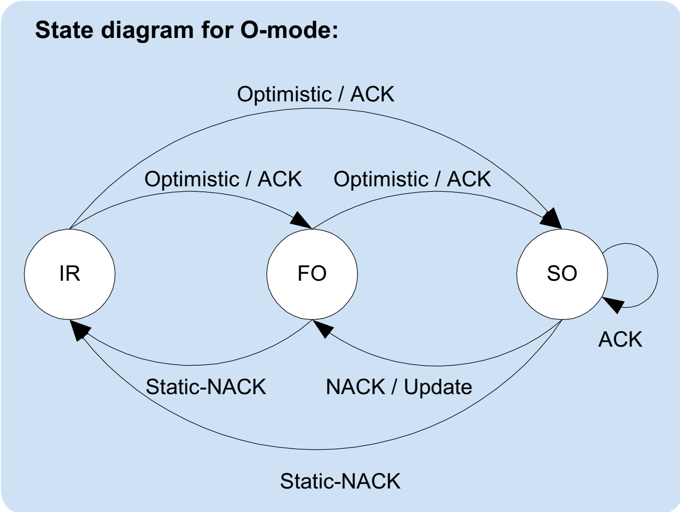
- **Unidirectional mode (U-mode)**
  In this mode, packets are sent in one direction, from the compressor to the decompressor. In cases where the return path or the reverse channels are not available, ROHC can still be used. If return path or feedback channel is available, it may be used by the decompressor to acknowledge successful decompression.

  In U-mode, the compressor starts in the IR state (as described earlier) and in an optimistic approach it sends a number of IR packets to establish the context at the decompressor side. The number of IR packets is usually dependent on the link characteristics such as RTT. The compressor uses the optimistic approach, sending enough information to reconstruct the context, to maintain the integrity of the decompressor context. The context updating information is sent periodically to ensure context synchronization. A time-out mechanism (periodic) is used to transit to lower states and send FO and IR state packets. Due to this inefficient method of error recovery and context synchronization, this mode of operation results in lower compression gain compared to other modes of operations.



  The compressor always starts in U-mode. It can move to other modes of operation if it receives a feedback (request) packet from the decompressor. As the decompressor can estimate link conditions, and knows the availability of the feedback channel, it can choose to move to other modes of operation than U-mode.

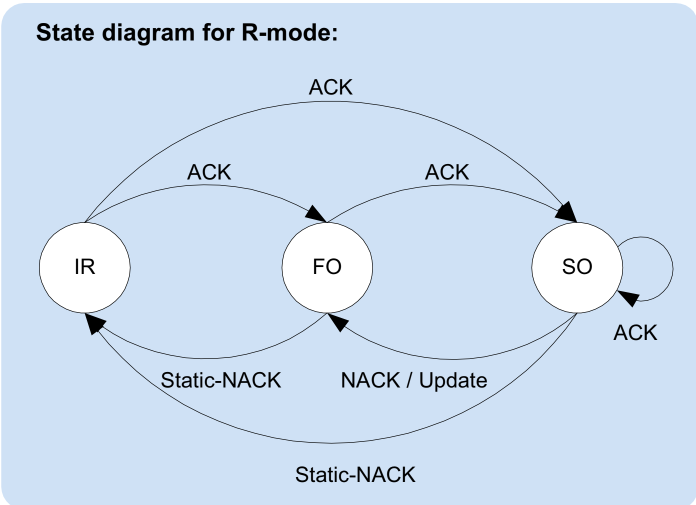
- **Bi-directional Optimistic mode (O-mode)**
  In this mode, the decompressor can send feedback in the form of requests for error recovery (negative acknowledgements, NACKs) and indication of successful context update (acknowledgements, ACKs). As shown in the following diagram, the compressor relies on the optimistic approach or ACKs from the decompressor to move to higher states. The decompressor sends ACKs for IR packets. For other context updating packets, it is optional to send ACKs. To recover from error conditions, it sends NACKs or static NACKs (depending on its state). The following diagram shows state changes of the compressor in the optimistic mode upon receiving feedback from the decompressor and using the optimistic approach.



This mode yields higher compression gains while making sparse use of the feedback channel and it reduces the packet loss due to the context invalidation.

- **Bi-directional Reliable mode (R-mode)**
  In this mode, the feedback channel is used quite often to avoid packet loss due to context invalidation. R-mode uses the secure reference principle rather than the optimistic approach as in other modes. In secure reference principle, confidence of the compressor depends on acknowledgements from the decompressor for every context updating packet. However, not every packet in R-mode updates the context.

  As per the secure reference principle, the compressor should send the context updating packets periodically and repeat until acknowledgement is received from the decompressor.



The periodic context updating packets reduce compression efficiency compared to O-mode. R-mode tries to improve the robustness in case of packet loss and bit errors. It has very low probability of the context invalidation but if it occurs, it can result in large number of damaged packets being delivered to upper layers.

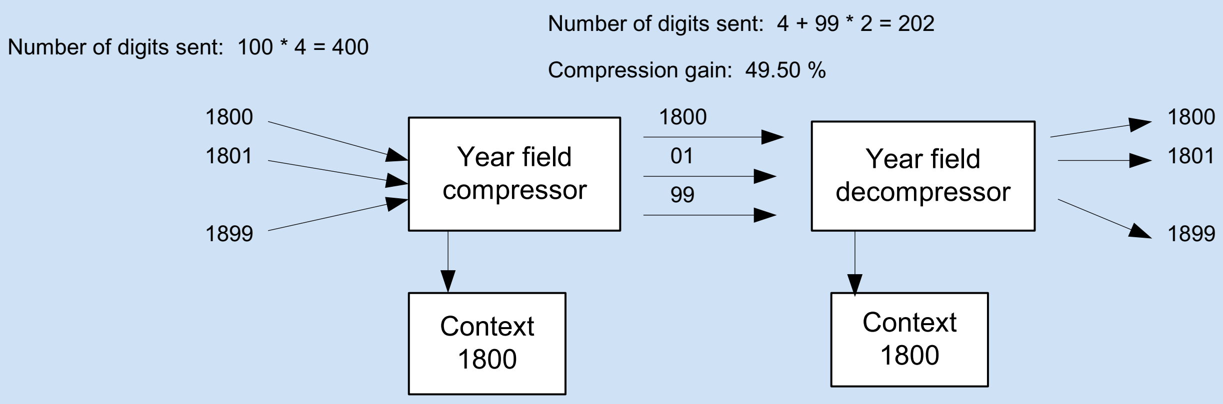
The ROHC compressor always starts in the U-mode and the IR state. Depending on the operational environment like the availability of the feedback channel, bit error conditions on the link and change patterns in packet header fields, the decompressor can request for mode change. All kinds of mode changes are possible as from U-mode to O-mode/R-mode, from O-mode to U-mode/R-mode and from R-mode to U-mode/O-mode. The operator can set the preferred mode of operation and he can change it at any time.

The encoding mechanisms

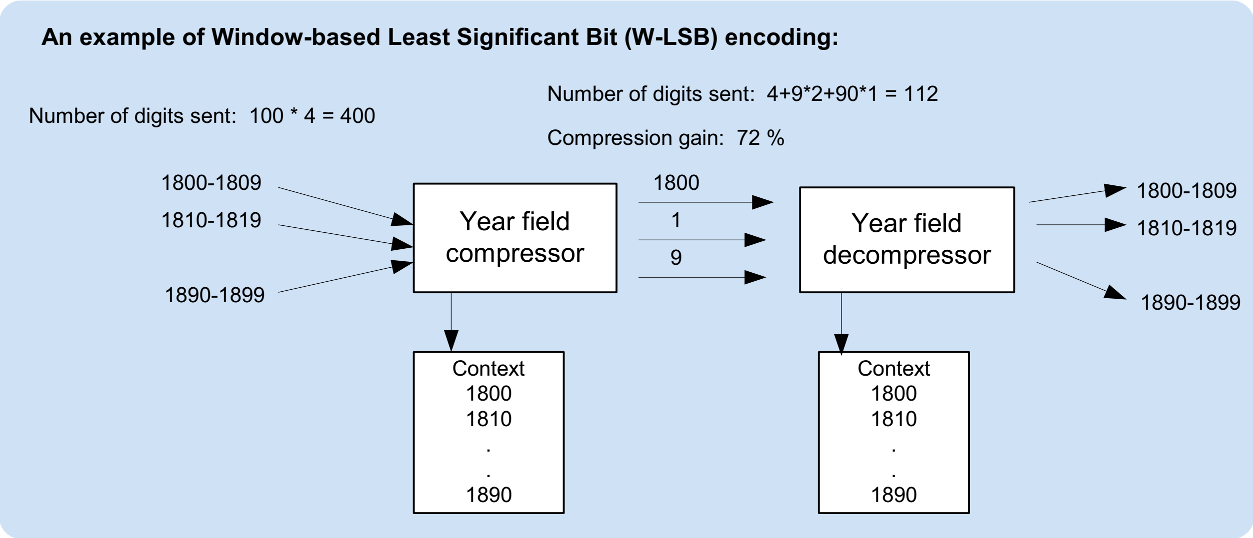
The earlier header compression schemes like Van Jacobson compression (RFC 1144), IPHC (RFC 2507) use the delta-encoding method for the compression of header fields. Some of the fields in a packet flow remain constant like source and destination addresses in IP header, source and destination ports in UDP/TCP header etc. Some fields change in a specific pattern like Identifier field in the IP header which can increase by 1 or by any other fixed number or can remain constant, usually zero, in some operating systems, while some fields change randomly from packet to packet like the checksum in UDP/TCP headers. The fields which remain constant need not to be sent at all once the context is established on the compressor and the decompressor sides whereas the fields which change randomly must be sent as they are. One can use various algorithms to compress the fields, which change in specific patterns. The delta encoding is one such algorithm. It simply calculates the difference between the fields in the consecutive packets. Using a method of self-describing variable length values it represents the difference in fewer bits. Even though the method is simple, it is not efficient in case of packet losses before and after compression. It is not robust enough for bit errors on the link and it can lead to packet losses at the decompressor.

The ROHC framework introduces a more complex algorithm to find patterns and to compress the field values. As it has been designed to work on cellular links, it is robust against bit errors, packet loss and long round trip times. It uses the Window-based Least Significant Bit (W-LSB) encoding algorithm. A simple example of this is the way we talk about the year field in the date. The following diagram shows an example of LSB in decimal form.

An example of Least Significant Bit (LSB) encoding:



Say, we are discussing about the 19th century i.e. the year field 18xx then we can represent the year field with only the last two numbers. The 77th year will be understood as 1877 and the 83rd year will be understood as 1883. In other words we are representing the year field with the 2 least significant numbers with reference to the 19th century as the window. As we shift the window back to the 18th or forward to the 20th century, the year field will be interpreted as 1777 or 1977.



If we can be even more specific about the description of the window e.g. the 7th decade in the 19th century (187x) then we can represent each year with just one number. The 7th year will be interpreted as year the 1877.

We can take another example using bits instead of numbers. The IP-ID field in the IP header is represented using 16 bits without any compression. Let us say that IP-ID is starting from 0 and increasing by 1 each time a packet is sent. Let us assume that the window size is 8 packets. We can observe that we need only 3 bits instead of 16 bits to represent the IP-ID. As IP-ID starts to increase beyond number 7, we simply shift the window and calculate the IP-ID. We can observe that even if the packets are reordered (before being compressed) i.e. packet 5 comes before packet 3 we still need only 3 bits for compression. We can also see that if we loose a small number of packets, we can successfully decompress the packets. The compression is not completely dependent on consecutive packets. The window size plays an important role in this algorithm. It can be adjusted dynamically as link conditions are observed. Of course, packet reordering and losses beyond the current window size limits cannot be handled. These situations can be effectively handled by feedback mechanisms supported by ROHC. To gain even more compression of bits, ROHC uses self-describing variable length values. Thus the W-LSB algorithm together with the feedback mechanism makes ROHC very robust against bit errors on the link and in turn suffers less due to long round trip times. It also results in high compression.

In the ROHC scheme, two special algorithms have been developed for compressing the RTP Timestamp (TS) field, which takes 32 bits in uncompressed form, to attain higher compression. These algorithms depend on factors such as sampling rate, frame size of audio/video RTP packets and the fact that these packets are generated at a fixed interval. The relation between these factors and the RTP Timestamp change is established and used to gain higher compression. These algorithms are called Scaled RTP Timestamp Compression and Timer-based RTP Timestamp compression.

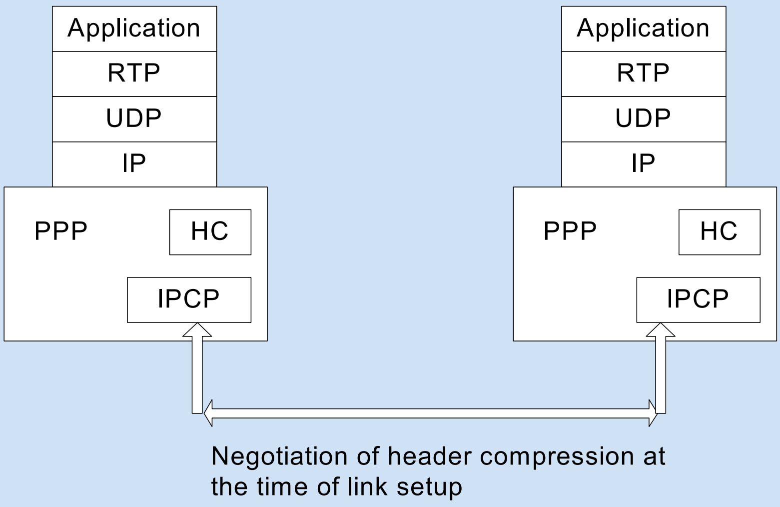
The ROHC protocol and its features

The ROHC protocol

The IP header compression scheme must be used on a hop-to-hop link as it compresses the IP header which is essential for the functions such as routing, QoS etc at a hop. The ROHC scheme uses concept of channel to represent a directional packet flow on the link. A link can have multiple channels in both forward and reverse directions.

The ROHC scheme does not depend on the link to provide the packet type indication. It is included in the ROHC packets. ROHC expects that packets are not reordered on the link between the compressor and the decompressor. Packet reordering before compression can be handled by ROHC. ROHC assumes that there will not be any duplication of packets on the link. The link should be able to provide packet length and supports framing. The ROHC scheme also recommends that the lower layers should use some level of error detection/protection mechanism and do not deliver ROHC headers with high residual errors.

An example of application of header compression in a protocol stack:



The header compression functionality must be negotiated with a set of parameters on the link. Such a negotiation protocol for ROHC over PPP has been defined in the IETF standard RFC 3241. The negotiation of the ROHC parameters like maximum context identifier, profiles supported etc takes place during link setup. The above diagram shows the location of the header compression functionality and the negotiation over a PPP link.

ROHC features

The ROHC scheme uses quite a few numbers of compressed packet types (more than 10) and feedback packet types (2). Some of the link layers require that the packets should be of fixed sizes (from a pre-determined set of sizes). For some packets, padding is required whereas some packets need to be segmented. The ROHC scheme supports the segmentation of packets. It is recommended to use ROHC Segmentation only if link layer segmentation is not available.

The ROHC scheme operates on a fixed packet header structures as defined in each profile. For example, the ROHC RTP profile can handle packet headers of different types like IPv4/UDP/RTP, IPv6/UDP/RTP and IP/IP/UDP/RTP etc. The profile identifies field characteristics and relations and suggests the compression mechanism to use. But when headers like RTP carry a list of contributing source identifiers (CSRC), it must be handled in a special way. Similarly when IPv6 extension headers are carried, they must be compressed in a special way. The compression mechanism for such fields and headers is called list compression. The ROHC scheme compresses the following fields/headers using list compression: CSRC list, extension header chains in IP headers (both IPv4 and IPv6). The following extension headers can be compressed using the list compression feature: AH header, null ESP header, minimal encapsulation header, GRE header and IPv6 extension headers. The IPv4 and IPv6 headers cannot be part of the extension header lists.

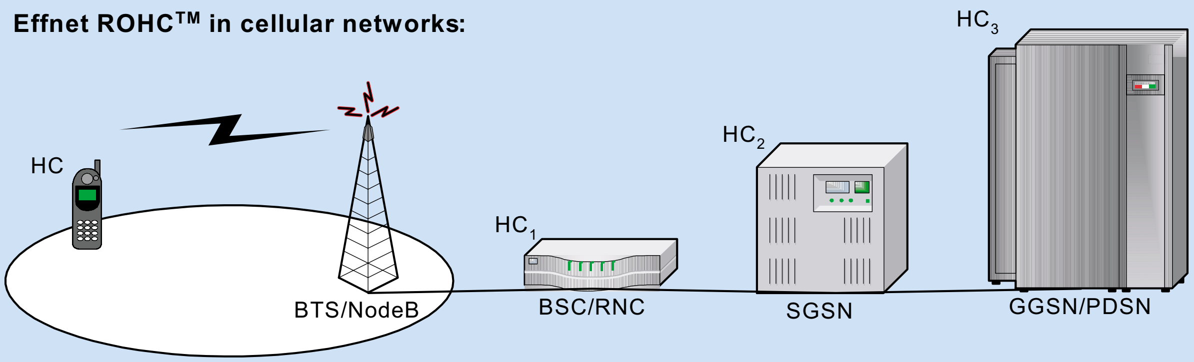
The ROHC standard specifies an optional feature of reverse decompression. This feature helps to reduce the number of packets discarded due to invalid context. When a context becomes invalid, the decompressor fails to decompress the packets. Those packets are discarded (not delivered to the application or not forwarded). To recover from the context invalidation, the decompressor can send a feedback to the compressor if the feedback channel is available or otherwise simply waits to receive a context-updating packet from the compressor. Until it receives such a packet, the decompressor has to discard all the packets it has received in the mean time. The number of packets will depend on the packet flow rate and round trip time of the link. In such cases, the decompressor can choose not to discard the decompression-failed packets but store them for future decompression attempts. It will try to decompress those packets again, in reverse order of reception, i.e. last received is decompressed first. This feature needs to buffer packets and cannot be used for delay sensitive applications.

The ROHC standard supports compression of the RTP packet flows. The real time control protocol (RTCP) is normally used by all applications, which use RTP. The ROHC scheme doesn’t have any specific support for RTCP packet compression but as IP/UDP packets carry RTCP, the ROHC UDP profile can be used to compress RTCP packet flows efficiently.

ROHC standards and application areas

Cellular standardization bodies like 3GPP and 3GPP2 have adopted the ROHC standard. The ROHC scheme is recommended in Release 4 of the 3GPP standard and is considered to be a critical component of Release 5 and onwards. The IP-based Multimedia Subsystem (IMS) to be introduced in Release 5 is based on IPv6 only. The IPv6 header itself being 40 bytes and together with UDP/RTP totalling 60 bytes becomes a considerable overhead for voice and video applications. It is essential to save the bandwidth, reduce the packet loss due to bit errors and reduce delays due to these large overheads. The ROHC header compression becomes indispensable to improve user experience of services and save the expensive radio resource.

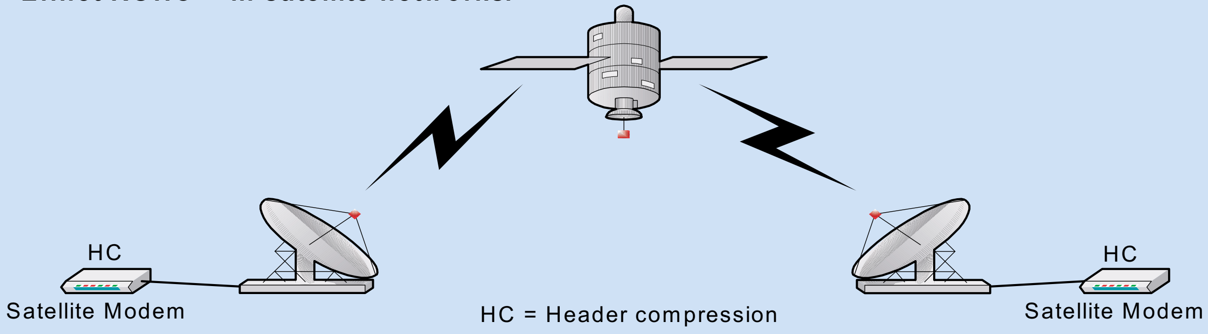
Effnet ROHC™ in cellular networks:



HC = Header compression
HC is always used in the terminal according to all the standards together with:
1. RNC as per UMTS standard, or
2. SGSN as per GPRS standard, or
3. PDSN as per CDMA2000 standard.

According to the 3GPP standards, the ROHC header compression function is a part of the mobile device on one end and the radio network controller (RNC) on the other end. As per the 3GPP2 standards, the ROHC header compression function is a part of the mobile device and the packet data switching node (PDSN).

Effnet ROHC™ in satellite networks:



The satellite communication market is evolving to provide broadband data services over their broadcast networks. It represents a low cost way to reach people in remote areas and enable them with broadband network access and all the services. The Digital Video Broadcasting standard enables satellite operators to incorporate data streams in their video and audio streams efficiently. The satellite links have high bit errors and long round trip times. Using header compression on these links minimizes the effects of both these factors and provides better quality of service for the users and increased link efficiency for the operators.

The wireless local area networks (WLAN) are supporting applications like Voice over IP, promising low cost telephony and data infrastructure for homes and offices. These wireless networks, even though high bandwidth, suffer from high bit error rates (most of these networks use Industrial, Scientific and Medical radio band of 2.4GHz which is used by many devices and a very common device, microwave ovens introduces radio emissions in the same radio band) and resulting delays in packet deliveries. With header compression, it is possible to save bandwidth reduce packet loss due to bit errors by inherently sending smaller packets and reduce delays introduced by packet losses on the radio link. Work is in progress to adapt the link layer to support negotiation of header compression in WLAN and to suggest a model of usage describing the locations of header compression entities in the network.

The ROHC standard is now evolving to include more protocol headers like TCP, UDP-Lite and more profiles like IP-Only profile to efficiently handle flows that are handled by the ROHC Uncompressed profile.

Effnet ROHC™

Effnet offers a variety of header compression products. They are used in many types of IP networks such as 2.5G and 3G cellular networks, satellite networks, dial-up modem links, wide area networks etc. All Effnet's header compression products are designed to be easily adapted to a variety of operating systems and hardware platforms. The implementations are developer-friendly and available both in user space for debugging and testing (with Effnet HC-Sim™). They have been successfully integrated in link layers such as the PPP according to the standards.

Effnet ROHC™ has undergone extensive testing. Effnet HC-Sim™ (Effnet Header Compression Simulator), another product from the Effnet Header Compression product family, is used to simulate traffic and link conditions to test the functionality of header compression modules. Effnet HC-Sim™ features a wide range of test cases with comprehensive logging and statistics generation capability. This ensures detailed testing of all features and functionality of Effnet's header compression products. For more information about Effnet HC-Sim™, see the related data sheet at www.effnet.com

Features of Effnet ROHC™:

- Software fully compliant with the IETF standard RFC 3095
- Lightweight implementation including all features suitable for low-end devices
- Highly portable product with ANSI-C implementation
- Platform, endianness and byte-order independent
- Highly configurable with compile- and run-time options
- Highly modular with external memory management
- Multi-threading support with re-entrant code
- Extensively tested, in-house as well as during interoperability and field tests

Functions of Effnet ROHC™:

- All profiles: 0 (Uncompressed), 1 (RTP), 2 (UDP), and 3 (ESP)
- Compression of both IPv4 and IPv6 headers
- All states and modes including mode transitions
- All ROHC packet types, including all extensions: 0, 1, 2, and 3
- Local repair mechanisms
- Timer-based compression of RTP Timestamp (profile 1)
- List compression
- ROHC Segmentation and Reassembly
- Packet Size Limitation Enforcements
- Reverse Decompression

Glossary of acronyms

2.5G	2.5 Generation wireless networks
3G	3rd Generation wireless networks
3GPP	3rd Generation Partnership Project
3GPP2	3rd Generation Partnetship Project 2
ARPU	Average Revenue Per User
BER	Bit Error Rate
BSC	Base Station Controller
BTS	Base Transceiver Station
CDMA2000	Code Division Multiple Access 2000 network
CRTP	Compressing IP/UDP/RTP Headers for Low Speed Serial Links (RFC2508)
CTCP	Compressing TCP/IP headers for low-speed serial links (RFC1144)
DVB	Digital Video Broadcasting
ECRTP	Enhanced Compressed RTP (CRTP) for Links with High Delay, Packet Loss and Reordering (RFC3545)
EDGE	Enhanced Data rates for GSM Evolution
FTP	File Transfer Protocol
GGSN	Gateway GPRS Support Node
GPRS	General Packet Radio Service
GSM	Global System for Mobile telecommunication
HC	Header Compression
IETF	Internet Engineering Task Force
IMS	IP based Multimedia Subsystem
IP	Internet Protocol
IPCP	The PPP Internet Protocol Control Protocol (RFC1332)
IPHC	Internet Protocol Header Compression (RFC 2507)
OSI	Open Systems Interconnections
PDSN	Packet Data Serving Node
PPP	Point-to-Point Protocol
PSTN	Public Switched Telephone Network
QoS	Quality of Service
RFC	Request For Comments
RNC	Radio Network Controller
ROHC	Robust Header Compression (RFC3095)
RTP	Real-Time Transport Protocol
RTT	Round Trip Time
SIP	Session Initiation Protocol
SGSN	Serving GPRS Support Node
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
VoIP	Voice over Internet Protocol
WCDMA	Wideband Code Division Multiple Access
WLAN	Wireless Local Area Network
xDSL	Digital Subscriber Line including Asymmetric, Very high rate etc.

For more information about header compression and the Effnet header compression products, please see our library of white papers and data sheets at www.effnet.com, or contact the Effnet sales office.

About Effnet AB

Since its beginnings in 1997, Effnet has been involved in research and development of technologies that improve the performance and efficiency of IP based networks. The Effnet Header Compression product family saves bandwidth and improves quality of service. Effnet is the leading independent provider of header compression products and is committed to continue to provide leading edge IP technology.

Effnet AB

040215

Visiting Address:
Gustavslundsvägen 151G
Bromma
Sweden

Postal Address:
Box 15040
SE-167 15 Bromma
Sweden

Phone: +46 (0)8 564 605 50
Fax: +46 (0)8 564 605 60

E-mail: info@effnet.com