EFFNET AB

WHITE PAPER

An introduction to IP header compression

FEBRUARY 2004

WWW.EFFNET.COM

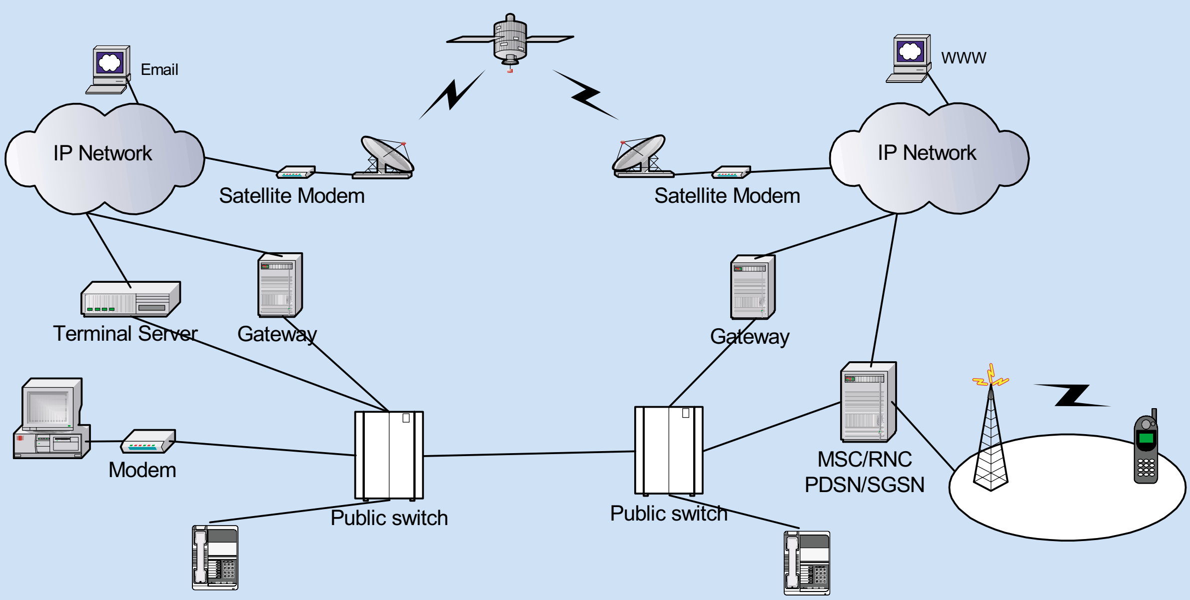
The need for IP header compression

The Internet Protocol (IP) is the choice of transport protocol both on 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 all compete for that bandwidth. For the 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 providing higher average revenue per user (ARPU).

As for wireless networks with their high bit error rates (highly prone to interference) and high latency (long round trip times), it is difficult to attain those high bandwidths required. When all these factors are taken into account it means that the available resources must be used as efficiently as possible.

The chain is as strong as its weakest link

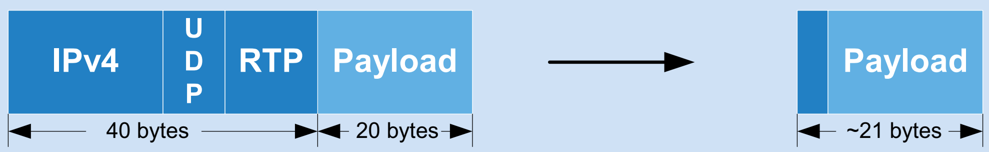


In many services and applications e.g., Voice over IP, interactive games, messaging etc, the payload of the IP packet is almost of the same size or even smaller than the header. Over the end-to-end connection, comprised of multiple hops, these protocol headers are extremely important but over just one link (hop-to-hop) these headers serve no useful purpose. It is possible to compress those headers, providing in many cases more than 90% savings, and thus save the bandwidth and use the expensive resources 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.

Header compression is about link efficiency

Let us look at some examples of how much compression (or bandwidth savings) we can achieve using header compression. The IP version 4 header is 20 bytes and when carrying UDP (8 bytes) and RTP (12 bytes, at least), the packet header becomes 40 bytes. A header compression scheme usually compresses such headers to 2 – 4 bytes. On an average, considering a few uncompressed packets and a few relatively large packets, more than 80% savings can be observed. When compared with the payload being carried, in such cases as voice where payload size is usually static and in the range of 20 – 60 bytes, the header size represents a huge overhead. Using header compression in such cases results in major bandwidth savings. The IP version 6 with a header size of 40 bytes is gaining wide acceptance and has been included in Release 5 and onwards versions of 3G wireless networks. In this case, header compression will result in even more savings.

The basic technology of header compression:



On low bandwidth networks, using header compression results in better response times due to smaller packet sizes. A small packet also reduces the probability of packet loss due to bit errors on wireless links resulting in better utilization of the radio spectrum. It has been observed that in applications such as video transmission on wireless links, when using header compression the quality does not change in spite of lower bandwidth usage. For voice transmission, the quality increases while utilizing lower bandwidth. In short header compression improves network transmission efficiency, quality and speed with:

- *Decrease in packet header overhead (bandwidth savings)*
- *Reduction in packet loss.*
- *Better interactive response time.*
- *Decrease in infrastructure cost, more users per channel bandwidth means less infrastructure deployment costs.*

The header compression gains:

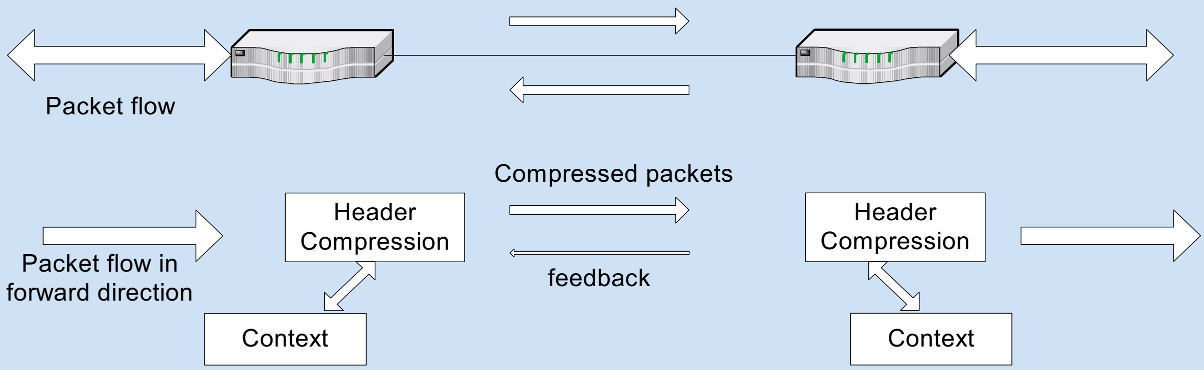
Protocol headers	Total header size (bytes)	Min. compressed header size (bytes)	Compression gain (%)
IP4/TCP	40	4	90
IP4/UDP	28	1	96.4
IP4/UDP/RTP	40	1	97.5
IP6/TCP	60	4	93.3
IP6/UDP	48	3	93.75
IP6/UDP/RTP	60	3	95

These benefits lead to improved QoS in the network and the possibility for operators to improve their ARPU. The operators will be able to retain and attract customers with better QoS on the network and more services and content on the links.

Header compression explained

The IP protocol together with transport protocols like TCP or UDP and optional application protocols like RTP are described as a packet header. The information carried in the header helps the applications to communicate over large distances connected by multiple links or hops in the network. The information comprises of source and destination addresses, ports, protocol identifiers, sequence numbers, error checks etc. As long as the applications are communicating most of this information carried in packet headers remains the same or changes in specific patterns. By observing the fields that remain constant or change in specific patterns it is possible either not to send them in each packet or to represent them in a smaller number of bits than would have been required originally. This process is described as compression.

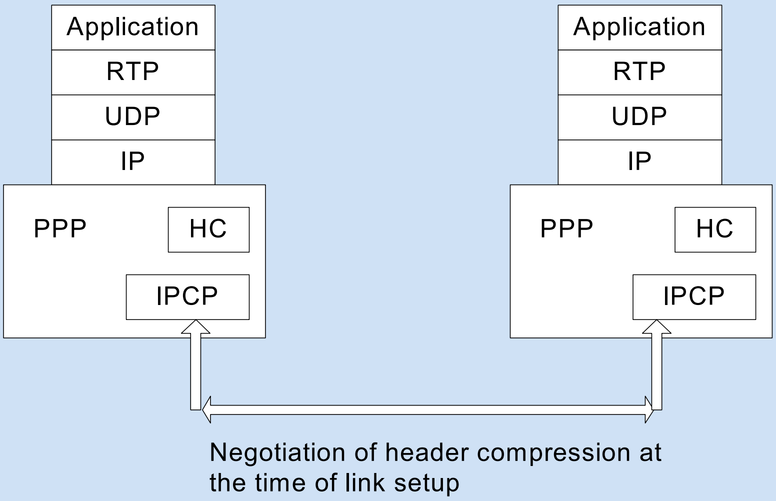
The concept of flow context in header compression:



The process of header compression uses the concept of flow context, which is a collection of information about field values and change patterns of field values in the packet header. This context is formed on the compressor and the decompressor side for each packet flow. The first few packets of a newly identified flow are used to build the context on both sides. These packets are sent without compression. The number of these first few packets, which are initially sent uncompressed, is closely related to link characteristics like bit error rate (BER) and round trip time (RTT). Once the context is established on both sides, the compressor compresses the packets as much as possible. By taking into account the link conditions and feedback from the decompressor, the compressed packet sizes vary. At certain intervals and in the case of error recovery, uncompressed packets are sent to reconstruct the context and revert back to normal operational mode, which is sending compressed packets.

The header compression module is a part of the protocol stack on the devices. It is a feature, which must be negotiated before it can be used on a link. Both end points must agree if they support header compression and on the related parameters to be negotiated.

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



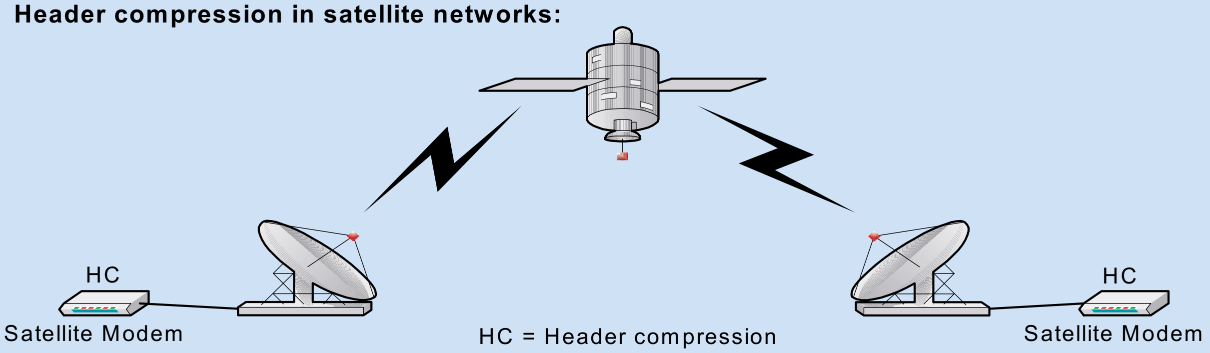
The above diagram shows the location of the header compression module in a protocol stack. The link layer for example PPP, uses the IPCP protocol to negotiate the use of header compression and related parameters at the time of the link set-up.

Header compression application areas

In keeping with the principle of end-to-end connectivity over IP, the header compression does not introduce any changes in the fields when it compresses and decompresses the header (reconstructing the header as it was before compression). The header compression is a hop-to-hop process and not applied end-to-end. At each hop in the IP network, it becomes necessary to decompress the packet to be able to perform the operations like routing, QoS etc. Header compression is best suited for specific links in the network characterized by relatively low bandwidth, high bit error rates and long round trip times.

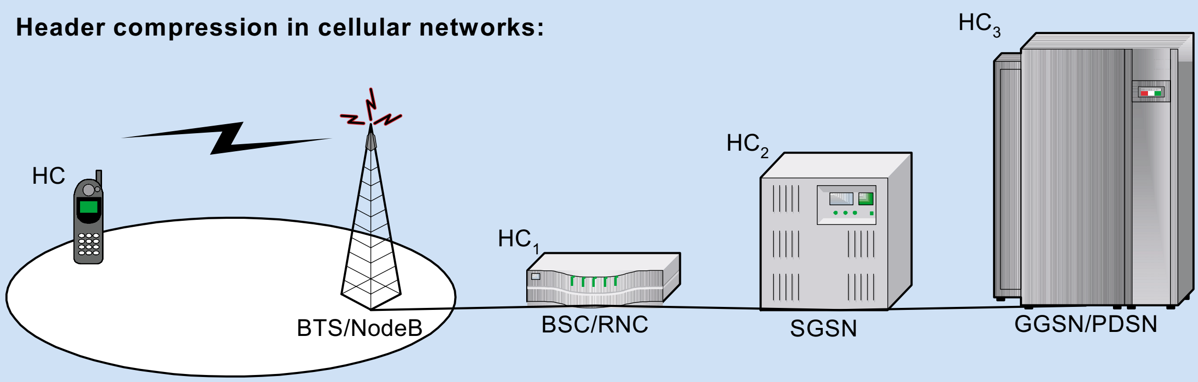
Realizing that the chain is as strong as its weakest link, header compression is the solution to improve the efficiency (strength) of this link and provide a better utilization of the network and improve user experience. Below are a few examples of such links and networks where header compression can be applied.

Header compression in satellite networks:



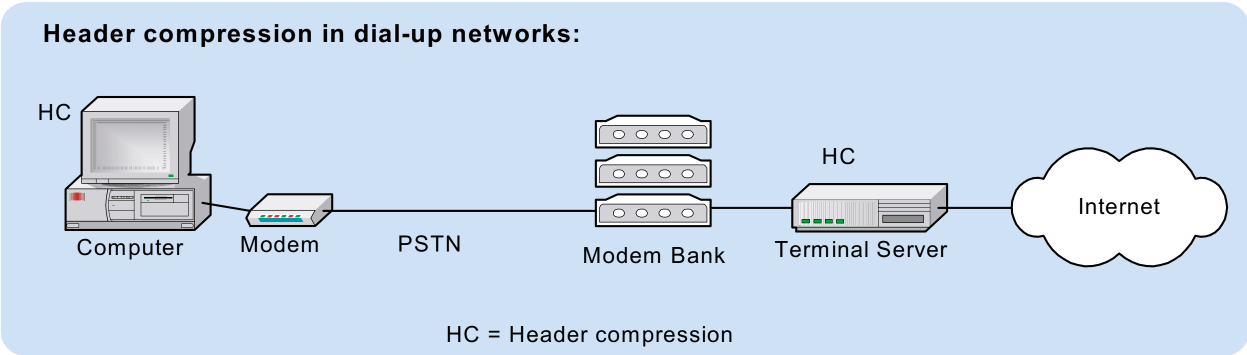
Satellite links have high bit error rates and high delays (delay varies from more than 500 milliseconds (ms) for geo-synchronous to a few ms for lower earth orbit satellites). The header compression module is part of satellite modem as shown above.

Header compression in cellular networks:

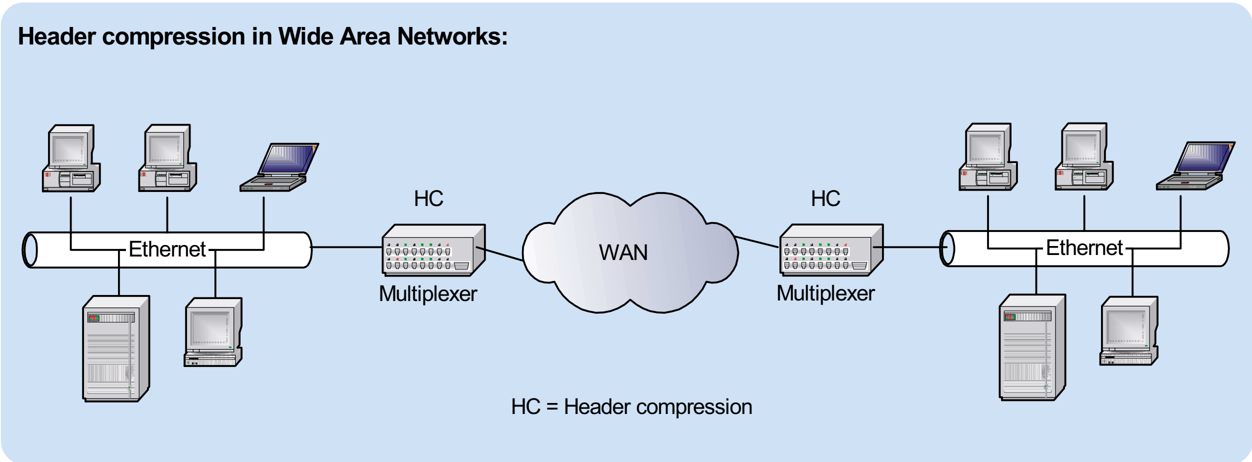


HC = Header compression
HC is always used in the terminal according to all the standards together with:
1. RNC as per UMTS standard, or
2. SGSN as per GPRS standard, or
3. PDSN as per CDMA2000 standard.

In 2.5G (GPRS) or 3G (WCDMA/CDMA2000) networks, the radio link has high bit errors. The standards specifications include the use of header compression for better utilization of the radio resource. In some applications, like IMS – IP Multimedia Subsystem, it is a critical component for successful operation. The header compression module is used in RNC as per the UMTS standard or SGSN as per the GPRS standard or PDSN as per the CDMA2000 standard together with the mobile terminal as specified in all the standards.



Internet access often uses modems and PSTN links, which have low bandwidth. To improve performance for interactive applications like Telnet, web services as well as Voice over IP, usage of header compression results in bandwidth savings. The header compression module is a part of the end user's computer operating system and the Terminal Server (PPP link end points) as shown in the diagram above.



Many multi-location offices are connected with WAN links that have comparatively high bandwidth (compared to PSTN links) and usually carry high data traffic. With the cost benefit of convergence of telephone and data networks, applications like voice and video over IP are competing for these WAN links. One of the best ways to save bandwidth is header compression (in this case with tunnelling and multiplexing features), which will remove unnecessary protocol overheads. The header compression module is a part of the routers/multiplexers connected to WAN.

Header compression standards

The Internet Engineering Task Force (IETF) was formed in 1986 to foster collaboration on the development and evolution of the Internet and related networking technologies. The IETF develops and standardizes header compression schemes.

The header compression standards are evolving and the following standards represent the steps in that evolution process:

Comparison of the IETF header compression standards:

IETF standard	RFC 1144 (VJ, CTCP)	RFC 2507 (IPHC)	RFC 2508 (CRTP)	RFC 3095 (ROHC)
Headers	IPv4/TCP	IPv4 (including options and fragments), IPv6 (including extension headers), AH, Minimal Encapsulation header, Tunnelled IP headers, TCP (including options), UDP, ESP	IPv4, IPv6 (including extension headers), AH, Minimal Encapsulation header, Tunnelled IP headers, UDP, RTP	IPv4 (including options and fragments), IPv6 (including extension headers), AH, Minimal Encapsulation headers, GRE, Tunnelled IP headers, UDP, RTP, ESP
Header compressed to minimum	2 bytes	2 bytes	2 bytes	1 byte
Link Type (BER, RTT)	Dial-up (Low, Short)	Dial-up and wireless (Low to medium, Short to medium)	Dial-up and wireless (Low to medium, Short to medium)	Wireless (High, Long)
Encoding	Differential	Differential	Differential	Window-based Least Significant Bit
Error recovery (Feedback)	TCP based (No)	TWICE (Yes)	TWICE (Yes)	Local repair (Yes)
Recommended in (standards)	-	UMTS Release 99 onwards CDMA2000 Release B onwards	-	UMTS Release 4 onwards CDMA2000 Release B onwards

- *The RFC 1144 (CTCP) header compression standard was developed by V. Jacobson in 1990. It is commonly known as VJ compression. It describes a basic method for compressing the headers of IPv4/TCP packets to improve performance over low speed serial links. VJ compression is the most commonly used header compression scheme in IP protocol stacks today. However, the evolution towards all IP networks has created new demands on header compression. Consequently, newer standards have developed with superior error recovery mechanisms, which work well on links that exhibit both non-trivial round-trip times and significant loss.*

- *The RFC 2507 (IPHC) was developed in 1999 by scientists, closely related to Effnet, of the Luleå University of Technology. This technique compresses, on a hop-by-hop basis, multiple IP headers including IPv4 and IPv6, TCP, UDP, ESP headers. The compression algorithms are specifically designed to work well over links with non-trivial packet-loss rates.*

- *The RFC 2508 (CRTP) standard, developed in 1999, was justified primarily by the specific problem of sending audio and video over low speed serial links. CRTP compresses the headers of IP/UDP/RTP packets used for audio and video, reducing overhead on a hop-by-hop basis. CRTP performs best on local links with low round-trip times.*

- *The RFC 3095 (ROHC) was developed in 2001. This standard can compress IP/UDP/RTP headers to just over one byte, even in the presence of severe channel impairments. This compression scheme can also compress IP/UDP and IP/ESP packet flows. ROHC is intended for use in wireless radio network equipment and mobile terminals to decrease header overhead, reduce packet loss, improve interactive response, and increase security over low-speed, noisy wireless links. ROHC has been adapted to work with link layer characteristics like those of GSM and CDMA and is known as Link Layer Assisted-ROHC (ROHC-LLA).*

These header compression schemes are widely adopted by various standardization bodies including the 3rd Generation Partnership Project (3GPP) and 3GPP2. The 3GPP and 3GPP2 standardize the specifications for 2.5G and 3G wireless networks. The header compression schemes such as IPHC and ROHC are already part of the Release 4 specifications of the 3GPP. These schemes are an essential ingredient for the success of the Release 5 and 6 specifications, which introduce IPv6 and IP Multimedia Subsystem. The header compression technology has also been adopted by satellite communication networks, low bandwidth wired networks and some special links like Frame Relay etc as well as unique networks like the Terrestrial Trunked Radio (TETRA).