University of Würzburg
Institute of Computer Science
Research Report Series

Performance Tradeoffs for Header Compression in MPLS Networks

Michael Menth, Oliver Rose

Report No. 291 November 2001

Institute of Computer Science, University of Würzburg
Am Hubland, D-97074 Würzburg, Germany
E-Mail:{menth|rose}@informatik.uni-wuerzburg.de

Performance Tradeoffs for Header Compression in MPLS Networks

Michael Menth, Oliver Rose
Institute of Computer Science, University of Würzburg
Am Hubland, D-97074 Würzburg, Germany
E-Mail:{menth|rose}@informatik.uni-wuerzburg.de

Abstract

In this paper, we propose the use of compression techniques for RTP/UDP/IP/MPLS headers in MPLS networks to enable header compression over several IP hops. We consider the transmission of low-bitrate real-time traffic and analytical results illustrate performance tradeoffs regarding network utilization by user data. Header compression reduces the gross rate of low-bitrate streams and increases the transmission capacity of the network for voice traffic by 150%. For circuit switched data services it is important to choose a suited packet size to maximize the performance. This explains why the reduced burstiness by header compression leads also to a more than intuitively expected performance gain on low-bandwidth access links.

Keywords: header compression, admission control, QoS, performance evaluation

1 Introduction

A large part of today's toll-quality real-time data consists of low-bitrate traffic such as voice, video and circuit switched data. They originate e.g. in the terrestrial radio access networks of wireless communication systems like GSM and UMTS and are carried over low-bandwidth links. When low-bitrate real-time data is transported over IP networks, the RTP/UDP/IP header suite results in a large overhead that decreases the bandwidth utilization by user data drastically. Header compression is a means to overcome this inefficiency due to increased processing complexity in the participating routers. It is conceived for point-to-point links and can in general not be used over several IP hops. MPLS is a newly emerging technology for traffic engineering in IP networks and it is likely to be deployed in future communication systems. It introduces a connection concept that defines logical links over several IP hops. This allows to apply compression schemes for low-bitrate real-time data over several IP hops which makes the use of compression schemes more flexible and less CPU time consuming in the intermediate routers. Thus, the use of header compression becomes more attractive in combination with MPLS and the benefits by header compression encourage the deployment of MPLS.

This work is structured as follows. In Section 2 we consider the transport of low-bitrate real-time data. We suggest an admission control mechanism to meet the QoS constraints. The transport over ATM and IP networks is inefficient and header compression techniques try to overcome this weakness. In Section 3, we shortly present the basics of MPLS and propose to adapt header compression to MPLS. The numerical results in Section 4 explain performance tradeoffs regarding low-bitrate real-time traffic and show the influence of header compression for RTP/UDP/IP/MPLS headers. Section 5 draws the conclusion from this work.

2 Transport of Low-Bitrate Real-Time Traffic

In this section we describe how admission control can be done for real-time traffic to meet its quality of service constraints. The low-bitrate and real-time properties lead to transport inefficiencies with current network protocols that can be overcome by various header compression and tunneling mechanisms.

2.1 Admission Control for Real-Time Traffic

In contrast to web traffic, real-time data yield higher revenues but they must be forwarded with low loss and delay even to mobile customers. In the terrestrial radio access network of GSM or UMTS, leased lines are used to interconnect the users with the core network. This is a costly solution and, therefore, it is desirable to make best use of the rented capacities to avoid unnecessary expenses. A high network utilization is required but the real-time contraints of the data must not be violated. To solve this conflict, the dimensioning of the network capacity is rather tight and admission control (AC) of new flows prevents congestion on the links. An efficient and still conservative AC must take advantage of the flow characteristics.

The major traffic volume of today's real-time data is due to telephony and video or it results from time critical applications that require a circuit switched data (CSD) emulation from source to destination. The low end-to-end delay requirement includes the traffic generation so that the time to assemble a data packet by the application is kept short which often results in small payload sizes. As a consequence, real-time traffic is often characterized by the the periodic production of small samples.

We multiplex several real-time flows with a fixed period packet inter-arrival time t_{IAT} on a common link. We assume only homogeneous traffic, so all the packets have the same size and the same service time t_{ST}. Since the inter-arrival times are periodic and the service times are deterministic, the whole queuing process is periodic with period t_{IAT}. It is obvious that the buffer queue must run empty at least once in a period if the load of the system is smaller than 1. Furthermore, the queuing behavior is fully determined of the inter-arrival pattern of the joint packet arrival process within a single interval of time t_{IAT}. To obtain the waiting time distribution, this system can be modeled by an $N * D/D/1$ queue which denotes the multiplexing of N identical flows with constant packet inter-arrival and service time. An analysis of that is found in [1]. It essentially randomizes all possible arrival patterns to obtain a waiting time distribution for arriving packet in this system. We just apply these mathematical formulae for the derivation of the numerical results in Section 4.

The waiting time is a suited criterion to define a QoS constraint for real-time traffic. We assume that a packet may wait for a given delay budget DB per hop because of queuing in the buffer. This is a very conservative approach since it is basically possible that all packets of the N sources arrive in one shot which leads to a maximum waiting time of $N \cdot t_{ST}$. Thus, it is better to soften this requirement to a probabilistic approach. The probability for the waiting time to exceed the delay budget DB must be smaller than p or in other words: The $1 - p$ quantile of the waiting time distribution must be smaller than DB. For interactive real-time traffic we use the parameter set $p = 10^{-4}$ and $DB = 5$ msec. Figure 1 illustrates this concept. Another aspect is the packet loss probability.

However, as mentioned before, the buffer queue runs empty at least once per t_{period} time and, therefore, it can be dimensioned large enough to prevent packet losses due to buffer overflow at all.

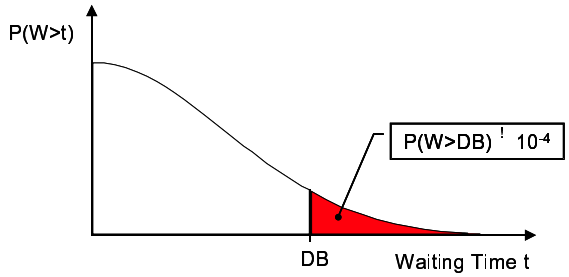

Figure 1: The quantile of the waiting time is a measure for QoS.

Using this analysis the maximum N_{max} can be found for which the QoS constraint is still met. Having a number of flows, we can compute the offered load of the system and we get the *critical load* based on N_{max}. For the computation of the offered load, we have the option to compute either the gross load including all protocol overhead or the net load which is based on the mere user payload. In our investigation, we always refer to net load since we want to compare the network utilization by user data.

2.2 Header Compression for Low-Bitrate Real-Time Traffic

We consider now transport alternatives for low-bitrate real-time traffic in the Internet. First, we present the conventional RTP tunneling approach which is inefficient regarding the network utilization by user data. Header compression can be done on a link-by-link basis to overcome this weakness. Finally, packets with compressed headers may be multiplexed into a single RTP/UDP/IP packet and tunneled over several hops to a common destination.

RTP Tunneling. For real-time transport in the Internet, the Real-Time Transport Protocol (RTP) [2] is used. The RTP header comprises 12 bytes. It carries a synchronization source identifier (SSRC), a timestamp, a sequence number, and some flags. It provides information to resynchronize different streams within an application. The port numbers of the communicating applications at the sender and the receiver machine are qualified in the User Datagram Protocol (UDP) [3]. Its header is 8 bytes large. Moreover, it records the length of the UDP packet and protects it with a checksum against errors. The IP protocol header carries the addresses of the source and the destination machine. In the old IP version 4 (IPv4) [4], the header comprises 20 bytes, thereof 4 octets for each IP address. The address space of IPv4 is likely to run out in the future, especially as soon as an all IP architecture requires lots of end devices. Therefore, the new IP version 6 (IPv6) [5] spends 16 octets per IP address and has a header size of 40 bytes. We use this alternative in our investigation. The RTP/UDP/IP protocol suite amounts to 60 bytes while the average voice packet size is not even 30 bytes large. This results into a protocol overhead of more than 200%.

RTP/UDP/IP Header Compression. When real-time data are exchanged, most of the protocol fields do not change during the session lifetime and the timestamp and the sequence number change steadily, e.g. by an increment of 1. This is a prerequisite for header compression over a point-to-point link because it relies exactly on this observation [6, 7, 8]. The constant and steadily changing data compose the session state that is mapped to a connection identifier (CID) such that the complete header can be reconstructed from the CID. To establish the context in the compressor and the decompressor at both sides of a point-to-point link, a full header is exchanged together with the CID. To make the system more robust, full headers are transmitted with regular distance but under good conditions, this is done only once for 256 frames. RTP/UDP/IP header compression works only on a link-by-link basis because of the compressed header. Without an IP header, packets can not be routed through an arbitrary transit IP network.

RTP Multiplexing. A means to overcome this handicap is to apply header compression between two arbitrarily distant peers and to transport several of the compressed packets in a single IP packet to the destination. This is called multiplexing. Figure 2 shows the advantage of multiplexing techniques in an IP network over pure header compression. The resulting packet has an ordinary IP header and can be carried transparently through the Internet.

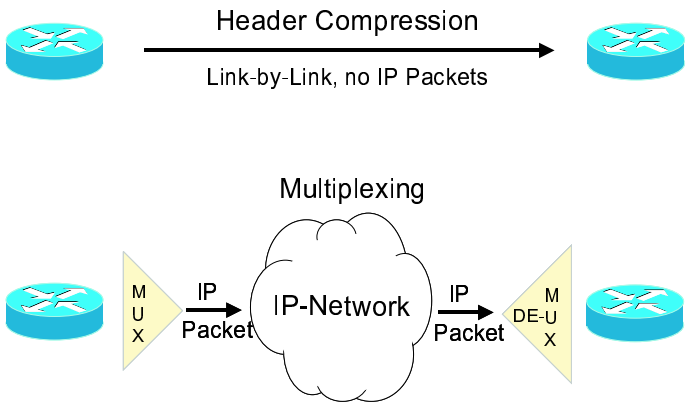

Figure 2: Header compression versus multiplexing.

The present Internet draft for multiplexing compressed RTP packets [9] is based on an enhanced RTP/UDP/IP header compression algorithm [7, 10] and a layer independent multiplexing protocol [11, 12]. A further compression step reduces the size of the multiplexing layer [13]. This kind of multiplexing requires to collect several samples with a compressed header for the resulting IP packet. The time for that must be limited to avoid additional delays for the transported data. This yields some interesting performance behavior that have been studied in [14, 15, 16]. The ATM Adaptation Layer Type 2 (AAL2) [17, 18] is basically the same approach for ATM systems and exhibits similar tradeoffs [19, 20].

3 Header Compression in MPLS Networks

In the recent years, research has concentrated on the transport of real-time data in packet switched networks. In IP networks, the IntServ [21] and DiffServ [22] approach seem to be promising. However, they lack powerful traffic engineering mechanism to perform route pinning, load sharing, fast rerouting, and others. To facilitate this, Multiprotocol Label Switching (MPLS) [23] is introduced. MPLS establishes virtual tunnels using only a small label for packet forwarding. Therefore, it offers itself as a tunneling technology with little header overhead. It may be used in conjunction with IP networks and is suited for carrying real-time data with compressed header information.

3.1 Some Basics about MPLS

MPLS is a mechanism to allow packet switching instead of routing over any network layer protocol [23]. A label switched path in MPLS represents a connection. The first label switching router (LSR) equips the IP packet with a label of 4 bytes and sends it to the next LSR. The LSRs classify a packet according to its incoming interface and label. Based on this information, label swapping is performed and the packet is forwarded to the particular outgoing interface. The last LSR only removes the label from the IP packet header. In practice, routers are capable to both IP routing and MPLS label switching.

MPLS is often viewed as modified version of the Asynchronous Transfer Mode (ATM) with variable cell size. But there is a profound difference: ATM enables with its virtual connection and virtual path concept a two-fold aggregation while MPLS allows for many-fold aggregation using multiple label stacking, i.e. an LSP may be transported over other LSPs. This feature may be exploited for scalable network structures [24].

3.2 The Use of Header Compression with MPLS

As we have seen in the previous section, the transmission of low-bitrate real-time data is inefficient if the size of the header leads to a small network utilization by user data. Header compression mitigates the problem at the expense of losing the IP header such that the packet can only be transported on a point-to-point basis using another special layer 2 transport protocol like PPP.

We propose to use MPLS for that purpose instead because the labels are small and MPLS is recommended for traffic engineering purposes anyway. We establish the compressor and the decompressor at the ingress and egress of an LSP. The LSP is a virtual tunnels that is able to carry packets with compressed headers transparently over several IP hops. This idea has been worked out in [25, 26]. Not only the RTP/UDP/IP header suite can be compressed but also a part of the label stack if multiple label stacking is used. According to the draft, these headers can be reduced to 2 or 4 bytes including the CID.

On the one hand it is evident that this would increase the network utilization by user data but on the other hand the use of header compression is costly. To operate at high speed, special purpose hardware must be designed and compressors and decompressors must work reliably even in case of packet losses and other network failures. Therefore, it is important to estimate the performance gain attained by header compression in order to balance the expenses against the expected benefits.

4 Numerical Results for Performance Tradeoffs

In this section, we compare the admissible load for voice traffic in terms of user data on a link with and without header compression. We argue that the performance gain is more than intuitively expected and illustrate the reason for that on the basis of a tradeoff observed with a 64 kbps circuit switched data service.

4.1 Performance Gain by Header Compression

We consider voice streams coded with 12 kbps such that every 20 msec a frame of 30 bytes is transmitted. The conventional RTP/UDP/IP/MPLS protocol suite yields a header size of 64 bytes (=94 bytes burst size) while header compression allows to work with compressed header of 4 bytes. Theses are tunneled through an LSP equipped with a label of 4 bytes such that the resulting burst are 38 bytes long.

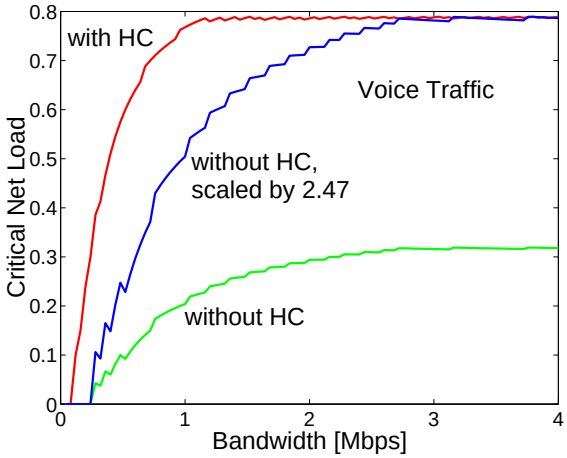

Figure 3: The performance gain by header compression (HC) is more than the reduced mean rate.

Voice transmission without header compression is clearly inferior to the header compression alternative. It is obvious that the gross rate of a stream is reduced from 37.6 kbps to 15.2 kbps, hence, we expect an increase of network utilization by user data of 147%. However, as we can see in Figure 3 the actual growth is even larger, especially for low-bandwidth links. This fact is due to the reduced packet size.

4.2 Influence of Burstiness and Protocol Overhead

The above observed phenomenon can be illustrated even more clearly by a 64 kbps circuit switched data service (CSD) emulation over IP. One packet is assembled per transmission time interval (TTI) and sent over the network. For a TTI=20 msec, the resulting user payload size is 160 bytes large. We investigate the performance of the system depending on the duration of TTI which is proportional to the resulting burst size. Again, the normal header size is 64 bytes, the size for an compressed header tunneled by an LSP is 8 bytes large.

Figure 4 shows that an optimum TTI exists for the transmission of CSD. This can be explained as follows. For decreasing TTIs, the ratio between user payload size and header size decreases, too, which has direct impact on the critical net load in the system. For increasing TTIs, the assembled packet size rises and the increased burstiness reduces the critical gross load on the link. Hence, there must be a TTI that maximizes the critical net load. The graph contrasts the critical load for links with a bandwidth of 1 and 4 Mbps. The conclusion is that the optimum TTI depends on the bandwidth of the carrier network. A similar behavior is observed for a 4 Mbps link with and without header compression. The critical net load is clearly optimized (see Figure 5) but the header compression does not displace the optimum value for TTI.

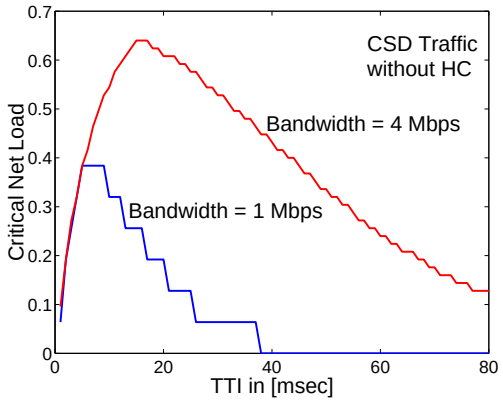

Figure 4: The impact of the link bandwidth on the optimum TTI value.

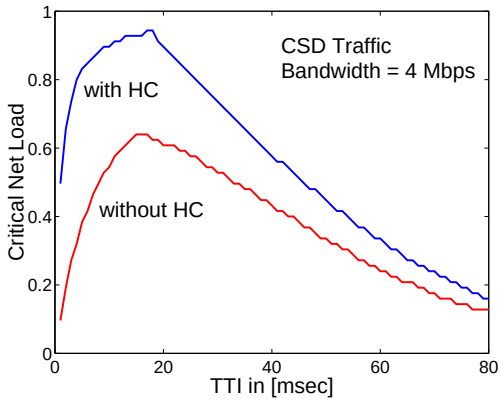

Figure 5: The impact of header compression (HC) on the optimum TTI value.

The experiments with CSD traffic prove that solely increased burstiness (even with slightly decreased mean rate) have an adverse impact on the critical gross load and its effect depends on the bandwidth. Hence, the benefit of header compression is not only the smaller mean rate but also the reduced burstiness of the streams. The first one contributes to a higher user proportion in the gross data and the second one allows a larger critical gross load in the system. That's why the profit of header compression is more than one would intuitively expect.

5 Conclusion

In this work we have proposed to compress RTP/UDP/IP/MPLS headers for low-bitrate real-time traffic in MPLS networks. An adaptation of existing header compression approaches is required and their application to LSPs instead of point-to-point links allows to take advantage of its benefits over several hops. Header compression reduces the gross rate of low-bitrate streams such that the transmission capacity of networks increases for voice traffic by almost 150%. For the transport of circuit switched data services or other multimedia applications it is important to choose a suited packet size to increase the network performance. This also motivates the positive impact of reduced data burstiness in

the presence of header compression which leads to a more than intuitively expected performance gain on low-bandwidth access links. Apart from traffic engineering capabilities, this finding is another reason to use MPLS in future networks.

References

[1] J. Roberts, U. Mocci, and J. Virtamo, *Broadband Network Teletraffic - Final Report of Action COST 242*. Berlin, Heidelberg: Springer, 1996.

[2] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RFC1889: RTP - A Transport Protocol for Real-Time Applications." ftp://ftp.isi.edu/in-notes/rfc1889.txt, Jan. 1996.

[3] J. Postel, "RFC768: User datagram protocol." http://www.ietf.org/rfc/rfc0768.txt, Sep. 1980.

[4] J. Postel, "RFC791: Internet Protocol." http://www.ietf.org/rfc/rfc0791.txt, Aug. 1981.

[5] S. Deering and R. Hinden, "RFC2460: Internet Protocol Version 6 (IPv6) Specification." ftp://ftp.isi.edu/in-notes/rfc2460.txt, Dec. 1998.

[6] M. Degermark, B. Norgren, and S. Pink, "RFC2507: IP Header Compression." ftp://ftp.isi.edu/in-notes/rfc2507.txt, Feb. 1999.

[7] S. L. Casner and V. Jacobson, "RFC2508: Compressing IP/UDP/RTP Headers for Low-Speed Serial Links." ftp://ftp.isi.edu/in-notes/rfc2508.txt, Feb. 1999.

[8] M. Engan, S. L. Casner, and C. Bormann, "RFC2509: IP Header Compression over PPP." ftp://ftp.isi.edu/in-notes/rfc2509.txt, Feb. 1999.

[9] B. Thompson, T. Koren, and D. Wing, "Tunneling Multiplexed Compressed RTP (TCRTP)." http://www.ietf.org/internet-drafts/draft-ietf-avt-tcrtp-05.txt, Nov. 2001.

[10] S. Casner, V. Jacobson, T. Koren, B. Thompson, D. Wing, P. Ruddy, A. Tweedly, and J. Geevarghese, "Compressing IP/UDP/RTP Headers on Links with High Delay, Packet Loss and Reordering." http://www.ietf.org/internet-drafts/draft-ietf-avt-crtp-enhance-03.txt, Nov. 2001.

[11] G. S. Pall, B. Palter, A. Rubens, W. M. Townsley, A. J. Valencia, and G. Zorn, "RFC2661: Layer Two Tunneling Protocol L2TP." ftp://ftp.isi.edu/in-notes/rfc2661.txt, Aug. 1999.

[12] R. Pazhyannur, I. Ali, and C. Fox, "PPP Multiplexing." http://www.ietf.org/internet-drafts/draft-ietf-pppext-pppmux-01.txt, Aug. 2001.

[13] A. J. Valencia, "L2TP Header Compression (L2TPHC)." http://www.ietf.org/internet-drafts/draft-ietf-l2tpext-l2tphc-04.txt, Oct. 2001.

[14] M. Menth, "Carrying Wireless Traffic in UMTS over IP Using Realtime Transfer Protocol Multiplexing," in *12th ITC Specialist Seminar*, (Lillehammer, Norway), pp. 13 – 25, March 2000.

[15] M. Menth, "The Performance of Multiplexing Voice and Circuit Switched Data in UMTS over IP Networks," in *Protocols for Multimedia Systems (PROMS2000)*, (Cracow, Poland), pp. 312 – 321, Oct. 2000.

[16] M. Menth, "Analytical Performance Evaluation of Low-Bitrate Real-Time Traffic Multiplexing in UMTS over IP Networks," *Journal of Interconnection Networks*, vol. 2, no. 1, pp. 147–174, 2001.

[17] ITU-T, "I.366.1 Segmentation and Reassembly: Service Specific Convergence Sublayer for the AAL Type 2," June 1998.

[18] ITU-T, "I.366.2 Draft Recommendation: AAL Type 2 Service Specific Convergence Sublayer for Narrow-Band Services," June 2000.

[19] N. Gerlich and M. Menth, "The Performance of AAL-2 Carrying CDMA Voice Traffic," in *11th ITC Specialist Seminar*, (Yokohama, Japan), Oct. 1998.

[20] M. Menth and N. Gerlich, "A Numerical Framework for Solving Discrete Finite Markov Models Applied to the AAL-2 Protocol," in *MMB '99, 10th GI/ITG Special Interest Conference*, (Trier), pp. 0163–0172, Sep. 1999.

[21] B. Braden, D. Clark, and S. Shenker, "RFC1633: Integrated Services in the Internet Architecture: an Overview." http://www.ietf.org/rfc/rfc1633.txt, June 1994.

[22] S. Blake *et al.*, "RFC2475: An Architecture for Differentiated Services." ftp://ftp.isi.edu/in-notes/rfc2475.txt, Dec. 1998.

[23] E. C. Rosen, A. Viswanathan, and R. Callon, "Multiprotocol Label Switching Architecture." http://www.ietf.org/rfc/rfc3031.txt, Jan. 2001.

[24] M. Menth and N. Hauck, "Scalable QoS Transport in IP Using LSP Hierarchies," Technical Report, No. 287, University of Würzburg, Institute of Computer Science, Nov. 2001.

[25] L. Berger and J. Jeffords, "MPLS/IP Header Compression." http://www.ietf.org/internet-drafts/draft-berger-mpls-hdr-comp-00.txt, Jan. 2000.

[26] L. Berger and J. Jeffords, "MPLS/IP Header Compression over PPP." http://www.ietf.org/internet-drafts/draft-berger-mpls-hdr-comp-over-ppp-00.txt, March 2000.