PREDICTING PERFORMANCE OF ROHC OVER THE ULTRA LIGHTWEIGHT ENCAPSULATION PROTOCOL

Mahesh Sooriyabandara, Michael Forrest, Gorry Fairhurst

Department of Engineering, Fraser Noble building, University of Aberdeen, Aberdeen AB24 3UE, UK. {mahesh, michael, gorry}@erg.abdn.ac.uk

Abstract:

This paper investigates the performance of UDP-based protocols operating over a DVB satellite link. The first part of the paper describes the Ultra-Lightweight Encapsulation (ULE), and compares this with the Multi Protocol Encapsulation (MPE), focusing on UDP traffic. The second examines the potential use of IP protocol header compression to improve the transmission efficiency of links. Performance is estimated, and issues identified in supporting Robust Header Compression (ROHC) over ULE links.

Key words:

MPEG-2 Transmission; IPv6; DVB-S, Header Compression, Satellite

1. INTRODUCTION

The Digital Video Broadcasting (DVB) standards[1] provide a link technology that may be used to build IP networks. Standard system components offer advantages of low cost and improved interoperability. Uses include data-broadcast, hybrid satellite/terrestrial networks, and two-way networks (e.g. DVB-RCS). Satellite links are particularly suited to delivery of internet multimedia, however, unlike wired connections, the cost of bandwidth is an appreciable component of the overall service cost, driving a commercial interest in achieving efficient usage of satellite capacity.

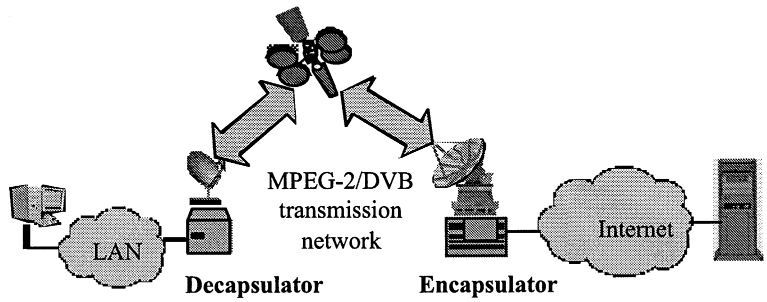

Figure 1. IP/DVB satellite scenario

Protocol Data Units (PDUs)[2] (i.e. IPv4, or IPv6 packets, or bridged Ethernet Frames) are sent over DVB networks using an encapsulation protocol[3]. The PDUs are processed by an IP Encapsulator to form Subnetwork Data Units (SNDUs). The most widely used standard is Multi-Protocol Encapsulation (MPE)[3]. This transmits each SNDU as a DSM-CC table section[3-4], that is fragmented into a series of MPEG-2 Transport Stream (TS) Packets. These are transmitted via satellite to a Receiver.

The MPE SNDU header retains compatibility with other MPEG-2 control frames. However, it is neither lean nor particularly efficient in terms of processing[4]. It also requires a destination MAC address, lacks an optional source MAC address, and has no next header protocol type field in the base header. The maximum size of PDU is slightly less than 4KB, although equipments may implement a smaller value.

2. ULTRALIGHT ENCAPSULATION (ULE)

An alternative to MPE, called the Ultra Light Encapsulation (ULE)[2], is being defined by the Internet Engineering Task Force, IETF, IP over DVB (ipdvb) working group[5]. This allows simple and efficient encapsulation of PDUs up to 32 KB and uses a mandatory CRC-32.

The SNDU header comprises 4 bytes of base header. The first two bytes carry a flag called the Destination Address Present Field (D), followed by the Length Field. By default, this flag is 0, indicating that a six-byte link destination address follows the base header. In certain network topologies (e.g. where the link destination is fixed) and some classes of packets (e.g. multicast packets) the sender may suppress this destination address (i.e. D=1). A two-byte Type field follows the Length. This uses IEEE Ethernet assigned values supporting IPv4, arp, IPv6, 203.1p and MPLS frames. The Type field may also be used to support extensions (e.g. MAC bridging)[2].

The ULE protocol[2] is intended to enhance performance and interoperability, while other work items will improve support for automated configuration for IP networks using DVB. A key goal of the activity leading to the definition of ULE was a desire to precisely define the protocol – such that any ULE Receiver may receive data from any ULE sender.

Table 1. Summary of SNDU Overhead

Overhead(Bytes)	Encapsulation	SNDU Header Fields and Function
16	IPv4/MPE	MPE, No LLC/SNAP Header No Ether type - Assume IPv4 (0x0800)
16+8 = 24	IPv4/MPE IPv6/MPE	MPE, with LLC/SNAP Header Ether type allows use of other protocols – IPv6 etc
16+24 = 40	Enet/MPE	MPE, LLC/SNAP Header containing Ethernet bridging header, but excluding Ethernet FCS.
16+28 = 44	Enet/MPE	MPE, LLC/SNAP Header containing Ethernet bridging header with FCS.
8	IPv4/ULE(D=1) IPv6/ULE(D=1)	ULE omitting destination receiver address
8+6 = 14	IPv4/ULE(D=0) IPv6/ULE(D=0)	ULE, including destination receiver address facilitating routing and L2 filtering
8+14 = 22	Enet/ULE(D=1)	ULE, containing Ethernet bridging header, but excluding Ethernet FCS.
8+6+14 = 28	Enet/ULE(D=0)	ULE, including destination receiver address and Ethernet bridging header, but excluding Ethernet FCS.

Table 1 compares the header overhead of MPE and ULE. MPE provides a range of encapsulation options leading to 4 basic header formats, ranging in size from 16 – 44 B. ULE may reduce the header overhead by 4% - 10%. For the general case, the size of the SNDU header in ULE and MPE does not differ significantly– although in ULE when the destination address is omitted, or bridging is used, a considerable saving may be achieved.

MPE & ULE transmission efficiency is a function not only of the PDU length, but also depends on the encapsulation method. Two methods are supported: Padding and Packing. With Padding, each new SNDU starts in a new TS-Packet (as for PES Audio/Video), and therefore Padding bytes fill any remaining payload within the final TS-Packet.

Padding negatively impacts performance for small PDUs. An Encapsulator that supports Packing can utilise the bytes following a previous SNDU to start a new SNDU. This removes the need for Padding, provided there is a continuous stream of PDUs arriving at the input of the Encapsulator. The figure below compares the efficiency MPE with Padding and Packing for a range of UDP payload sizes (which is commonly used for Internet multimedia). UDP introduces an additional 8 bytes per IP packet (RFC 768).

Both Padding and Packing offer good efficiency for large UDP payloads (comparable to Ethernet), however Ethernet is less efficient for small packets (although it was not conceived as a wireless technology). Packing can provide a significant benefit for IP traffic (reducing overhead from 8-80% to 1-30% - especially for small packets). The smaller ULE header may further improve transmission efficiency, increasing performance (by 1-2%). The gain depends on the amount of additional overhead added by MPE and is greatest when LLC/SNAP[6] headers would be required (e.g. for IPv6).

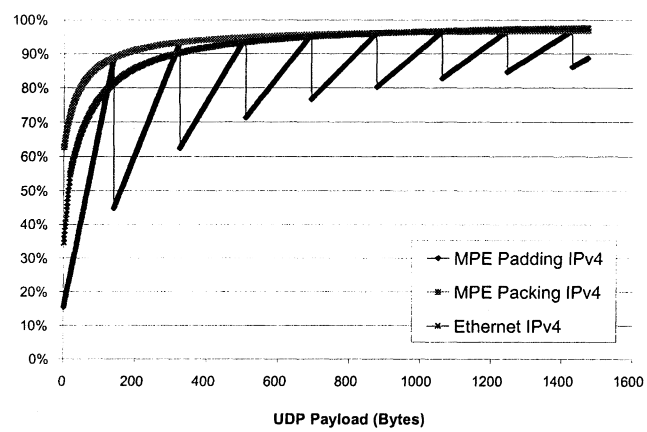

Figure 2. Calculation of Transmission Efficiency as a function of packet size

MPE allows an Encapsulator to use either/both mechanisms (if the corresponding Receiver supports them). The default in MPE is Padding, whereas ULE mandates that a Receiver supports both Packing and Padding and defines a fully-specified Packing algorithm. To control Packing, ULE introduces the concept of a Packing Threshold[2], which is a period of time that an Encapsulator may defer transmission of a partially filled TS-Packet to accumulate more SNDUs, rather than use Padding. After the Threshold expires, the Encapsulator uses Padding to send the partially filled TS-Packet.

2.1 Effect of packet size distribution

The distribution of PDU sizes, and the proportion of small packets, becomes important when analysing protocol efficiency. Figure 3 shows a set of IP traffic profiles expressed as a Cumulative Probability Distribution (CPD). This was measured for a range of TCP/IP applications by simulating a DVB-RCS link using dummynet[7]. The probability (P) of observing a packet greater than L, is determined by drawing a vertical line from the X-axis value of L to the curve and horizontally back to the y-axis value, P.

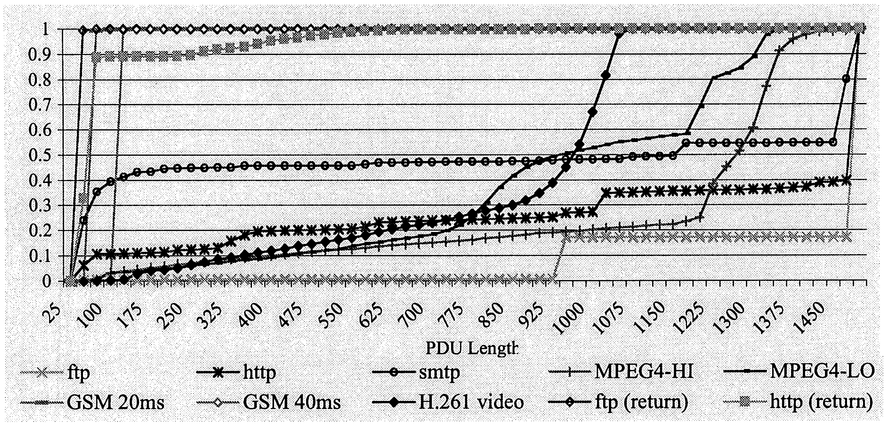

Figure 3. Summary of CPD of IP Packet size for traffic by application protocol

A number of trends may be observed: TCP-based applications transmit a high proportion of large IP packets, the traffic returned by the recipient towards the source consists mainly of a mixture of small TCP control packets and application messages (e.g. HTTP GET Requests). The remainder of the paper focuses only on UDP traffic.

UDP multimedia streaming is an important application for satellite. Video codecs typically generate a range of packet sizes although voice codecs (GSM, RAT) send fixed sized packets at a pre-defined rate. The impact of this type of traffic on transmission efficiency reduces when these flows are multiplexed with each other, or with other traffic.

2.2 MPE and ULE Performance

Tables 2 and 3 compare performance for each individual protocol in each captured data set. The same Packing algorithm[2] is used for both MPE and ULE. Columns are provided for MPE with Padding, and MPE without and with LLC/SNAP. The final two columns show performance of ULE (D=0, and D=1*)*. Table 2 shows perfect Packing (i.e. infinite Packing Threshold). Table 3 uses a practical Packing Threshold of 20ms.

The video-based applications and MP3 streaming content utilised a range of (larger) packet sizes, and showed similar transmission efficiencies. Packing improved transmission efficiency (0-36% for MPE; 4-40% for ULE) dependent on the application. The real-time audio codecs (PCM and GSM) all emitted small packets with RTP[8] media payloads, and showed considerable improvement in performance when using Packing. The fixed sized GSM packets were less than a TS-Packet payload (i.e. when Padding, 100% of TS-Packets were Padded). In many practical scenarios traffic will be aggregated and the expected performance (even with a Packing Threshold of 20ms) will be closer to that of the first table.

Table 2. Transmission Efficiency (Perfect Packing)

Transmission Efficiency (%)	MPE Padding		MPE Packed	MPE LLC	ULE (D=0)	ULE (D=1)
	TSPac Padded	Efficiency				
MPEG-4 HQ	14.9	88.8	91.8	88.8	92.6	95.9
MPEG-4 LQ	17.6	87.5	98.2	97.4	98.5	99.1
MP3	13.2	83.9	98.5	97.8	98.7	99.2
H.261	19.6	87.0	98.1	97.1	98.3	99.0
PCM	50.2	53.3	92.5	89.2	93.4	99.1
GSM audio (20ms)	100	39.7	81.2	75.2	83.9	90.1
GSM audio (40ms)	100	57.5	86.8	81.4	88.2	92.9

Table 3. Transmission Efficiency (20 ms Packing)

Transmission Efficiency (%)	MPE Padding		MPE Packed	MPE LLC	ULE (D=0)	ULE (D=1)
	TSPac Padded	Efficiency				
MPEG-4 HQ	14.9	88.8	91.8	88.8	92.6	95.9
MPEG-4 LQ	17.6	87.5	94.2	93.6	94.4	95.0
MP3	13.2	83.9	85.8	84.6	98.5	98.5
H.261	19.6	87.0	93.0	92.3	93.1	93.8
PCM	50.2	53.3	71.8	71.8	71.8	71.8
GSM audio (20ms)	100	39.7	75.6	41.3	75.6	75.6
GSM audio (40ms)	100	57.5	57.6	57.6	57.6	57.6

3. HEADER COMPRESSION

Transmission efficiency could further be improved by the removal or compression of redundant protocol header fields from the encapsulated packets. This trades computational processing costs at the Encapsulator and Receiver to gain improved transmission efficiency. The issues and potential gain of such a scheme are discussed in the remainder of this section.

3.1 The Header Compression Protocols

Existing IETF header compression schemes (e.g. RFC 1144, RFC 2508) do not perform well over wireless/satellite links, due to appreciable packet loss rates and long link round trip times. The IETF Robust Header Compression (ROHC)[9] WG has developed a series of Internet standards and related informational RFCs documents. The work is motivated by the Third Generation Partnership Project (3GPP)[11], where (like satellite) it is expected there will be considerable benefit from header compression. ROHC currently specifies compression of IP, UDP, RTP[12], and the work is on-going with several key issues to be addressed.

ROHC compresses Internet packet headers by using prediction logic at the transmitting link. At the link receiver, the same logic is used to regenerate headers that are semantically identical to those compressed. The logic requires state to be maintained for each compressed packet flow, called the compression Context. The ability to effectively compress depends upon the dynamics of the protocol header fields (i.e., how the headers change from packet to packet). Performance depends on link rate, link round trip time, and traffic. Work initially focused on packet voice traffic[16] and its associated signalling protocol. Low-rate packet video has also been considered, although the protocol field behaviour differs from that of audio and the current RTP profile may not provide the same level of compression.

The ROHC framework has specified compression only for unicast flows and was designed for bi-directional links. Support for uni-directional links is also being examined, and could in future evolve to support multicast – although this is not currently seen as a key 3GPP requirement. Currently there are no standards defining ROHC with multicast and broadcast – although this raises similar issues to the uni-directional case.

3.2 ROHC over DVB

It is expected that in future ROHC will be developed for other link technologies including those with high loss and long round trip times. MPE would not directly support such header compression, because it is not able to provide a Type field to identify which packets have been compressed. In contrast, ULE would allow this, although no type field codepoint(s) have been yet assigned to support ROHC.

For ULE to support ROHC, parameter selection procedures must also be defined and ROHC adapted to a broadcast channel properties. Addressing must also be considered: A Receiver that receives a ULE compressed PDU header with D=1, must decompress all PDUs to determine if the packets has an IP is to be forwarded. For point-to-point cases, this may be acceptable, it is desirable that shared links enable the L2 destination address (i.e. D=0), allowing Receivers to discard unwanted PDUs without decompressing.

4. PREDICTED PERFORMANCE OF ROHC

This section estimates the performance of using ROHC with Internet Multimedia applications over ULE. To simplify analysis, implementation of ROHC compressors and decompressors was avoided, specifically the ROHC context initialisation code (a current research problem for uni-directional links) was not implemeneted. Instead, performance was evaluated by writing a tool that modelled the ROHC behaviour in java. The tool analysed datasets captured using tcpdump, allowing repeatability when comparing different compression and Packing methods.

A range of RTP-based voice and video codecs were analysed to assess encapsulation efficiency. Applications were considered one at a time. Each application had specific characteristics, and exhibited a typical distribution of packet sizes. The following data sets were used:

- 3 ISMA video streams (High and Low quality MPEG-4 Video streamed by a Darwin Streaming Server; and a H.261 conference codec).
- 2 Audio streams (PCM/32kbps; GSM with 20ms RTP sample period).

Table 4. Transmission Efficiency for various UDP protocols using ROHC

	Max ULE	ULE (20ms)	Max ROHC /ULE	ROHC/ULE (20 ms)
1 Mbps, MPEG-4 (D=0)	96.8	96.8	100.2-102.1	98.1-100.1
0.3Mbps, MPEG-4 (D=0)	98.5	94.4	100.3-102.5	95.7-98.1
0.128Mbps, H.261 (D=0)	98.3	93.1	100.3-102.8	94.8-96.9
0.064 Mbps, PCM (D=0)	93.4	71.8	112.9	108.6
0.032 Mbps, GSM (D=0)	83.9	75.6	145.5	78.8
0.032 Mbps, GSM (D=1)	90.1	75.6	165.3	79.7

Column one of Table 4 shows the performance for Packing with an unlimited Packing Threshold. In practice, such a large value would be deleterious to real-time traffic, however the same efficiency results when a link carries multiplexed traffic (i.e. a high PDU arrival rate ensures Packing). The next column shows a Packing Threshold of 20 ms, the worst-case, for a single flow. The final pair of columns show corresponding results when emulating ROHC compression.

The video codecs generated a range of packet sizes with a bias to large packets. The benefits of compression are therefore small, e.g. 2-10%, and depend on codec configuration and rate. Most audio codecs emit packets at a fixed rate and suppress silence, resulting in bursts of packets at two packet inter-arrival times. Actual performance depends on the tools and codecs employed, configuration and RTP clock. For example, 64 kbps PCM with an RTP sampling interval of 20ms, generates evenly paced packets (at 20ms) of approximately 150 B. ROHC provides a gain of approximately 20%, for GSM speech, the gain is 60% with ULE (D=0) and 75% with ULE (D=1).

The impact of Packing Threshold on single periodic streams (e.g. voice codecs) can be unusual; resulting in some TS-Packets being Packing while others are Padded. An increase in Packing Threshold from 20ms to 40ms therefore may be expected to have a substantial impact on overall performance in such cases. For the GSM traffic, the overall benefit from ROHC results in increase of transmission efficiency from 79.9% to 141%.

4.1 Discussion

No data has been presented for TCP-based traffic, although clearly the same methodology could be applied. Analysis of traffic aggregates combining both UDP and TCP packet flows is likely to be much more complex, and would require development of new analysis methods.

The results in section 4 assume all packets are compressible, and therefore the achievable performance may be less than that estimated, especially for short flows where the time to establish a compression context is significant compared to the duration of a flow. This assumption is valid for multimedia flows: such as voice-over-IP or video-over-IP are actually long-lived, but is not necessarily true for TCP traffic flows. The benefits of ROHC increase significantly for IPv6, where the larger (40B) network header is more easily compressed (to a size comparable to a compressed IPv4 packet). In ULE this may require an additional extension code-point.

Most current use of ROHC anticipates bi-directional communication (i.e. that the links support full-duplex packet communication). Use over uni-directional links (e.g. DVB-S broadcast links) may be desirable, especially for UDP-based applications and hybrid networks that have asymmetric routing. Such schemes have been suggested for the MBMS (Multimedia Broadcast / Multicast Service) 3G service and would require a robust method for context initialization. This is an item of current research.

To determine actual performance, requires an implementation of the ROHC protocol, allowing evaluation of trade-offs between increased transmission efficiency and robustness of the compression protocols. This implementation is being currently studied at the University of Aberdeen.

5. CONCLUSIONS

The performance of the Ultra Lightweight Encapsulation (ULE) has been examined. In all cases, performance is at least comparable to the Multi-Protocol Encapsulation, and in some cases it is significantly better. Key features of ULE include the support of a Type field (allowing support of IPv4, arp, IPv6, 802.1Q, MPLS), and an increased maximum PDU size. In addition, the ULE protocol is fully-specified, and lightweight to implement.

The paper analyses the potential benefit of using ULE in combination with a Robust Header Compression (ROHC) protocol. Analytical approaches have highlighted the improvement in transmission efficiency for several types of UDP/IP traffic. The analysis showed that ROHC offers considerable performance gain when the traffic is characterised by small packets.

6. ACKNOWLEDGEMENTS

The authors acknowledge the support and funding from the European Space Agency (ESTEC Contract No.: 17471/03/NL/ND).

7. REFERENCES

[1] "Digital Video Broadcast", http://www.dvb.org.

[2] G. Fairhurst and B. Collini-Nocker, "Ultra-Lightweight Encapsulation (ULE)", IETF Work in Progress, 2004.

[3] "DVB Specification for Data Broadcasting," (ETSI EN 301 192, 2003.

[4] H. Clausen, H. Linder, and B. Collini-Nocker, "Internet over Direct Broadcast Satellites," *IEEE Communications Magazine*, vol. 37, pp. 146-151, 1999.

[5] IETF IPDVB WG http://www.ietf.org/html.charters/ipdvb-charter.html, 2004.

[6] IEEE, Part 2: Logical Link Control, IEEE, Std 802.2 ISO/IEC 8802-2:1998, 1998.

[7] S. Parker and C. Schmechel, "Some Testing Tools for TCP Implementors," IETF, RFC2398, 1998.

[8] H. Schulzrinne, et al, "RTP: A Transport Protocol for Real-Time Applications," IETF RFC1889, 1996.

[9] IETF RObust Header Compression WG, http://www.ietf.org/html.charters/rohc-charter.html.

[10] C. Bormann, et al "(ROHC): Framework", IETF, RFC 3095, 2001.

[11] The 3rd Generation Partnership Project (3GPP) http://www.3gpp.org/, 2002.

[12] L.-E. Jonsson and G. Pelletier, "ROHC: A Link-Layer Assisted Profile for IP/UDP/RTP", IETF, RFC3242, 2002.