Cross layer techniques for flexible transport protocol using UDP-Lite over a satellite network

William Stanislaus
william@erg.abdn.ac.uk

G.Fairhurst
gorry@erg.abdn.ac.uk

University of Aberdeen, Department of Engineering
Aberdeen, United Kingdom

Jose Radzik
jose.radzik@supaero.fr
Ecole Nationale Supérieure de l'Aéronautique et de l'Espace

Abstract— **Traditional real-time multimedia and streaming services have utilised UDP over RTP. Wireless transmission, by its nature, may introduce a variable, sometimes high bit error ratio. Current transport layer protocols drop all corrupted packets, in contrast, protocols such as UDP-Lite allow error-resilient applications to be supported in the networking stack. This paper presents experimental quantitative performance metrics using H.264 and UDP Lite for the next generation transport of IP multimedia, and discusses the architectural implications for enhancing performance of a wireless and/or satellite environment.**

I. Introduction

Recent years, have seen significant growth in the use of IP-based multimedia and streaming applications. UDP is an unreliable protocol suitable for delay sensitive applications which is suited to real-time applications that are sensitive to network delays and do not benefit from retransmission in the case of packet loss and/or error. UDP has a low protocol overhead of 8 bytes. Adding more functionality for in-order delivery or error recovery would likely increase the complexity and could significantly increase the delay and/or delay variation.

Example applications are VoIP, TVoIP, networked multiplayer games, streaming applications, etc. These applications currently use UDP as their transport protocol. RTP, layered above UDP, adds support for end-to-end delivery by providing information to the applications (e.g delivery monitoring, payload type and time stamps).

UDP has a strict checksum (recommended for IPv4 and required for IPv6) where corrupted packets will be discarded if these contain any transmission errors. This is reflected in the design of most current link protocols. In contrast, in a wireless network (e.g. Satellite, Wi-Fi, cellular mobile), links are characterised by low (or shared) capacity, appreciable delay (e.g. 100 s milliseconds) and variable (sometimes high BER).

Real-time streaming may use audio and video codecs that are error resilient. These give flexibility for applications to receive data with (some) bit errors within the packet payload. This suggests the packet may be divided into sensitive and insensitive parts. Errors in the sensitive part cause a packet to be discarded whereas errors in the insensitive part, are delivered, leaving the decisions to application codec. UDP-Lite [1,3] support this, by building on the features of UDP.

The design of UDP-Lite is discussed in section II *A*, followed by a brief discussion of partial checksum coverage, compared to a disabled UDP checksum. This also considers IPv6. Section II *B* provides an overview of the H.264 video codec. Section II *C* discusses checksum coverage and section II *D* discusses the link. Section III shows implementation and testing of UDP-Lite with H.264, followed by the conclusion and future work in Section IV.

II. Background

A. Transport Layer: UDP-Lite

The UDP checksum is a part of the UDP header and is calculated based on the UDP-Length (payload + UDP header) and the IP pseudo-header, which protects against routing errors, shown grey in figure 1. The strict checksum in UDP, requires that even if only a single bit in the received packet is in error the entire packet is dropped. This may be acceptable in wired IP network.

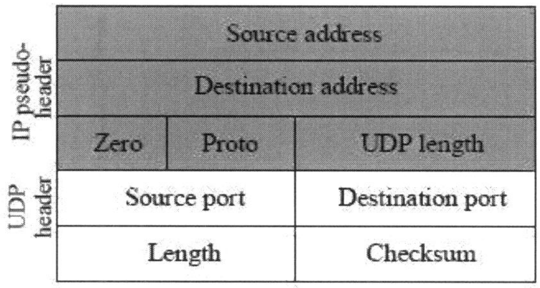

Figure 1. UDP Header

In IPv4, the UDP checksum is optional and a sender could disable the checksum. This may allow errored packets to be transmitted to applications. But this has the possibility of delivering packets with errors in the packet headers (e.g. UDP, RTP). A bit error in the sensitive part (corruption of the port fields could lead to delivery to the wrong application) when an error occurres in the network or transport header, guarding against this kind of error is an important function of the transport protocol.

The IPv6 header does not include a checksum field (although a similar function could be performed by an end-to-end AH header which provides cryptographically strong authentication of the whole packet). To verify the integrity of IPv6 datagrams therefore requires UDP to include a compulsory checksum when using IPv6 (Figure 2) RFC2460.

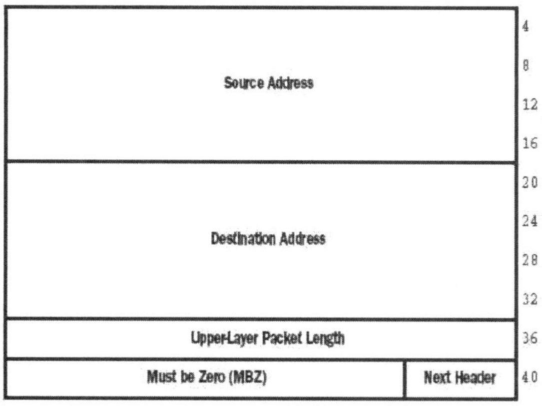

Figure 2. IPv6 Pseudo-header

The major difference between UDP and UDP-Lite divides each packet into two parts: sensitive and insensitive. The length field in UDP is replaced by the Checksum Coverage length, with a minimum of the UDP-Lite header length i.e. 8 bytes. A coverage of zero values indicates a UDP packet with full checksum coverage, Figure 3 shows the UDP-Lite header with the IP pseudo header.

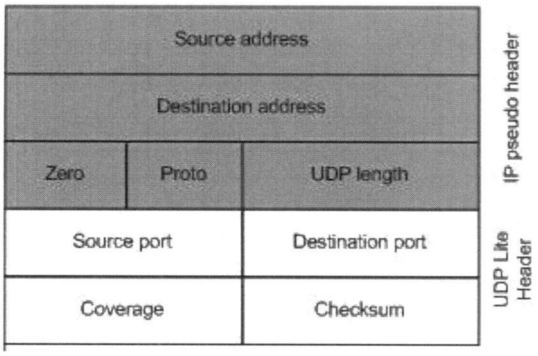

Figure 3. UDP-Lite header

B. *Application Layer: H.264*

The H.264[16] standard is a new ITU-T standard for video compression. The compression scheme in H.264 follows ISO MPEG and ITU-T H.26x, with new features to achieve higher compression efficiency [13].

H.264 is up to twice as efficient as MPEG-4 Part 2 (natural video) encoding (Table I), and has been welcomed into the MPEG-4 standard as Part 10 – Advanced Video Coding. Many established encoder and decoder vendors are moving directly to H.264 and skipping the intermediate step of MPEG-4 Part 2.

TABLE I. AVERAGE RATE SAVING COMPARED TO OTHER STANDARDS.

Codec	**MPEG4**	**H.263**	**MPEG1**
H.264	39%	49%	64%
MPEG4	-	17%	43%
H.263	-	-	31%

Bit-error resilient codecs (e.g. H.264) can deliver high quality video and voice even when supplied with corrupted data. A packet with an error in the application payload can cause a glitch in the audio/video quality, while an undelivered packet can cause a noticeable break in the audio/video stream. When motion-compensated prediction is utilized, the loss of information in one frame has a considerable impact on the quality of the following frames.

C. *Checksum Coverage Issues*

This section examines the need to perform integrity checks at various OSI layers:

IP Checksum: The IP checksum is a fast incremental checksum computed only on the IP header fields. The checksum is mandatory for IPv4 header and verifies that the header was not damaged in transit, and that packets were delivered to the correct end-host. In contrast, the IPv6 header relies on the link frame CRC in combination with the transport checksum (which includes the IP pseudo header). When UDP-Lite, is used with IPv6, a strong link layer checksum is required to ensure network header integrity.

Transport Checksum: Most transport protocols include the IP pseudo header as a part of the transport checksum calculation i.e. IP pseudo header + transport header + payload data. The transport checksum therefore provides on end-to-end check that the datagram was delivered to the correct application on the correct host.

RTP header Checksum coverage: RTP header coverage in the transport checksum is important, since this carries the information required for any real-time applications including delivery monitoring, payload type and time stamping.

Partial Checksum coverage for H.264 transmission: Recent video standards include techniques to enhance the robustness of the compressed data streams, for instance Reversible Variable Length Codecs (RVLC) and insertion of resynchronization markers. Other techniques, such as data partitioning and layered coding, provide classification and separation of bits according to their sensitivity to errors and losses.

H.264 [19] defines a method that may pack compressed video data in a partitioned fashion. The set of bits in a video frame may be subdivided by the encoder in an order that reflects their sensitivity to errors using three partitions or classes. Class A carries the most important headers, including macro block headers and motion vector information. Class B and C contain texture information of the various types of macro blocks. Class A should be covered in the data protected by the transport checksum (coverage) to guarantee error free transmission for the receiving application. Similar methods may be used by the AMR Voice over IP (VoIP) audio codec [11].

Link Layer CRC: Traditional Link frame CRCs are calculated over the entire frame, irrespective of the payload type. UDP Lite could be supported by a link implementing a partial checksum to ignore the link CRC at the receiver. However, the UDP Lite document mandates the use of partial header checksum by the link. Unequal Error Protection (UEP) schemes may take advantage of the different error sensitivity of various classes or layers.

D. *Cross-Layer optimization*

Network protocol design has traditionally adopted a layered approach. Each layer handles specific functionality, providing a service for higher layers. Coding and Modulation exist at the lowest (physical) layer. A typical link design is optimised by selecting appropriate fixed-rate FEC coding over a pre-selected modulation scheme. The network equipment connected to the link transmits and receives packets of data (encapsulated in frames) without providing indication of the actual requirements of the needs of specific packet flows (all packets are equal) [18]. Any corrupted packet is discarded and are not delivered end-to-end. The overall performance of the transport layer service may be suboptimal.

This new area of work is sometimes known as "cross-layer" support, and is consistent with the End-to-End argument [17], providing that system-level implications are understood, including the possible interactions with higher-layer mechanisms.

A "cross layer technique" is proposed that features partial checksum coverage for the packet header allowing the application to implicitly signal the link CRC coverage. The sending end-host implicitly signals (i.e. without explicit control messages) using a modified transport header, such as UDP-Lite [RFC3819].

A network router that supports UDP-Lite-aware link drivers can use this information to indicate that the packets being sent may exploit partial reliability features of a link [RFC3819], and configure appropriate checksum coverage. This does not indicate the actual requirements in terms of tolerable BER, loss ratio, etc. This requires explicit control messages to signal requirements to the physical layer from the higher protocol layers.

E. *Channel Link model*

Since performance of applications using these schemes is highly dependent on the actual experienced channel conditions at the link/physical layers, a detailed channel model is requires to assess the impact on the performance.

The initial analysis used a uniform error distribution (although uniform distributed error burst may result from physical layer FEC such as a Viterbi decoder) bursty error distribution is not expected to significantly change the results. However, real channel conditions often result in much more complex error distributions.

Layer design (Layer 1) allow adaptive coding and modulation (e.g. DVB-S2), which provide significant increases in physical layer flexibility allowing for improvements in efficiency/capacity of the link.

High capacity satellite systems or proposals for multimedia services use high frequency bands, mainly Ka band, in order to benefit from large available resources. However, deep fades can be observed above 20 GHz. Figure 4 shows attenuation for a typical convective rain event at 30 Ghz. The system can be no more conceived with a static margin and Fade Mitigation Techniques (FMT) are needed: ACM (Adaptive Coding and Modulation) and data rate adaptation.

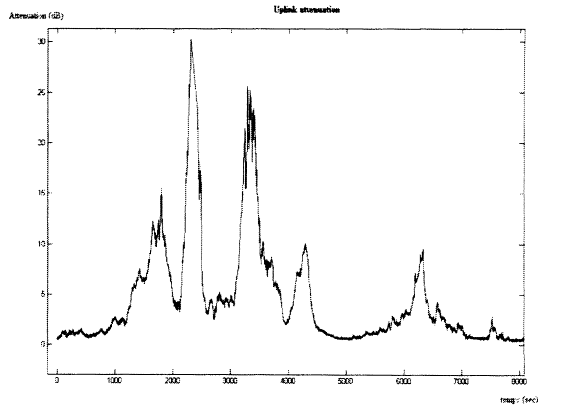

Figure 4. Attenuation for typical convective rain event (uplink, 30 GHz)

The DVB-S2 standard includes ACM with an adaptation of modulation (QPSK to 32 APSK) and coding rate on a frame per frame basis. The expected performances are close to an « error-free » channel (PER=10^{-7}). The switching criteria suppose a measurement of uplink and downlink propagation impairments, including a measure report form terminals on a return link [19]. Frames have a fixed size after coding, the actual data rate is thus variable (considering data rate=1 for 32APSK-9/10, data rate=0.07 for QPSK-1/4). The total data rate on one carrier is expected to be almost constant thanks to a statistical share between modes. The defined modes ensure an 18 dB dynamic margin on Es/No.

The DVB-RCS standard does not specify any FMT technique, however the flexibility of the MF/TDMA access technique allows for waveform adaptation to propagation conditions (carriers can be defined with different parameters). Terminal capacity is a limiting factor in the system design, since in most cases, only one or two modulations are useable (QPSK is the baseline). The margin is thus mainly obtained by the coding rates and generaly does not exceed 10 dB. In order to increase this margin (and the corresponding system availability), a data rate reduction is needed.

In both cases, data rate evolves with the measured propagation conditions. The observed effect on a single data flow is limited to jitter variation on the DVB-S2 link, depending on the service policy adopted for simultaneous QoS and ACM management. On the DVB-RCS return link, losses can occur if the coding rate is not fitted to propagation. The resource management with ACM is also a challenging task and resource shortages can happen even if the system as a whole is not saturated.

III. IMPLEMENTATION AND TESTING

A. *UDP-Lite Linux Kernel Implementation*

Modification is required to UDP in Linux Kernel 2.6.11 to support a new transport protocol such as UDP-Lite (0x88 protocol type). A new socket type for UDP-Lite was defined as SOCK_LDGRAM. Checksum coverage length from the application can be set to UDP-Lite socket via different socket options

- UDPLITE_SEND_CHCKLEN
- UDPLITE_RECV_CHCKLEN

The above options are for sender and receiver sockets respectively.

B. *VLC Modification for UDP-Lite*

VideoLAN [15] is an open source video streaming solution that targets multimedia streaming and live video on an IP network in unicast or multicast.

In this work, the transport layer protocol for VLC was modified to implement UDP-Lite and a socket-option added to indicate the checksum coverage required (e.g. to cover the IP/UDP-Lite/RTP header) at both a sender and a receiver. This supports both unicast and multicast transmission of IP packets.

C. *Test Topology*

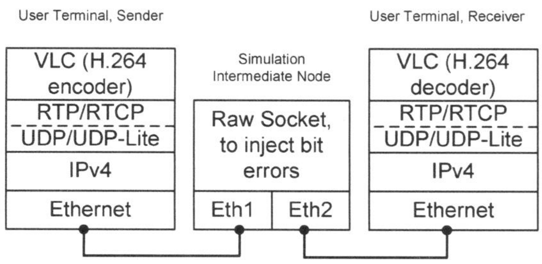

Figure 5. Evaluation Testbed

Simulations were conducted to identify the error resilience of H.264 video and to understand UDP-Lite performance with different checksum coverage for real-time multimedia streaming. The simulations used the testbed in shown Figure 5 to evaluate the performance of H.264 using the UDP and UDP-Lite transport protocols. The test environment includes: Two PCs with Intel 1.33 GHz processor, Linux kernel 2.6.11 with the modified UDP-Lite and VLC.

An intermediate node is used to inject bit errors in the packets transmitted from the sender to the receiver and to introduce delay in the packet flow. The bit errors considered only IP packets as it uses a RAW IP Socket call to pick the IP packets from Eth1 and inject error and transmit in Eth2, as shown in Figure 5 above.

D. *H.264 packet transmission*

The experimental results are tabulated in Table II for UDP and UDP-Lite. At a higher bit rate more the losses in general due to network congestion and packet processing overhead, here the transmitted and received packets are error free. In UDP-Lite the checksum coverage was considered minimal i.e. IP/UDP-Lite/RTP headers alone.

TABLE II. VLC PACKET TRANSMISSION USING UDP AND UDP-LITE WITH SAMPLE 50-CENT MUSIC AT RESOLUTION 320*240

Video Compress -ion Rate (Kbps)	Audio Compress-ion Rate (Kbps)	Packet Loss UDP(%)	Packet Loss UDP-Lite(%)	Play Time (Mins.)
2048	128	2.75%	2.5%	4.45
1024	128	1.46%	1.23%	4.45
768	128	0.71%	0.6%	4.45
512	128	0.21%	0.0%	4.45

E. *Measurements*

The above experimental results show that using UDP, if either the header or payload is corrupted then the whole packet will be discarded. If the header of packet is H bytes (UDP+RTP) and payload is D bytes, the UDP packet loss probability is

$$P_{PLP} = 1 - (1 - P_{BER})^{L*8}$$

Where, P_{PLP} = Packet Loss Probability,

L = Checksum coverage Length in Bytes (H + D)

Using UDP Lite, the packet will be discarded only when the header is corrupted, by taking advantage of the checksum coverage of UDP-Lite header, then the UDP-Lite packet loss probability will be calculated with L as H (UDP-Lite + RTP) in bytes.

The result is much lower packet loss at most BER values, in other words UDP-Lite is expected to show higher performance than UDP. In UDP, larger packet sizes increase packet loss rates. On the other hand, UDP-Lite shows flat loss rates since the error probability is only dependent on the header of the packet (Figure 6).

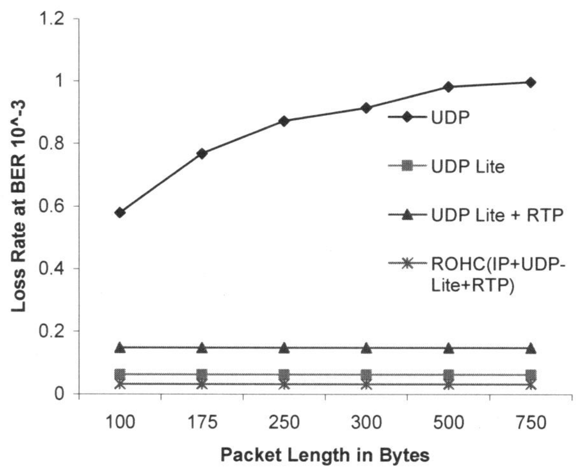

Figure 6. Payload length vs loss rate

Using RObust Header compression (ROHC): The ROHC compressor replaces IP/UDP-Lite/RTP overhead by its own, much smaller header [14]. On the receiver, it decompresses and transforms the ROHC header into the original protocol layer headers.

Assuming the error distribution is uniform (BER 10^-3), and then the probability of loss becomes the probability of error within the ROHC compressed header. If the ROHC compressed header size is about 4 bytes, the probability of packet loss due to header corruption is even reduced compared to flat header using UDP-Lite. Hence ROHC would make a significant difference to the subjective performance and the analytical results as shown in Figure 6.

IV. Future Work

In future cross-layer designs, messages provided from the lower layers can also be used to inform the transport protocol of prevailing link conditions and thereby interact with the end-to-end algorithms (such as H.264) for congestion and flow control.

The challenge is to design mechanisms at the various protocol layers that can optimize the overall end-to-end application performance, while minimizing the utilized radio resource.

V. Conclusion

The behavior of H.264-coded video transmission over UDP-Lite has been analyzed. The influence of partial checksum coverage by UDP-Lite in the real-time packet transmission has been studied by means of network simulations. Various Bit Error Ratio (BER) were considered for various video transmission rates using the simulation intermediate node. In particular, the change in the packet size with respect to the checksum coverage (IP/UDP-Lite/RTP headers only) and bit errors in payload data has yield markedly less packet drop and improved quality in video compared to performance of UDP on same circumstances.

Acknowledgements

This work was preformed with the support of funding provided by the University of Aberdeen and SUPAERO. The collaboration between the two research centres was funded as a part of the JA2430 work package of the EC IST Satellite Network for Excellence (SatNEx).

References

[1] L.-A. Larzon, M. Degermark, S. Pink, L.-E. Jonsson, and G. Fairhurst, "*The Lightweight User Datagram Protocol (UDP-Lite)*", IETF RFC 3828, 2004.

[2] A. Konrad, A. Singh, A. Joseph, and R. Ludwig, "*UDP Lite for Real-Time Wireless Video,*" 11th Int. workshop on Network and Operating Systems support for digital audio and video, Port Jefferson, NY, USA, 2001.

[3] L.-Å. Larzon, "*A Lighter UDP*," in Systemteknik/Datorkommunikation: Lulea, Sweden, 1999.

[4] A. D. Joseph, R. H. Katz, R. E. Ludwig, and S. Baucke, "*ICEBERG Update*," UCB June 19 2000.

[5] L.-Å. Larzon, M. Degermark, and S. Pink, "*Efficient use of Wireless Bandwidth for Multimedia Applications,*" Proc. 6th IEEE International Workshop on Mobile Multimedia Communications (MOMUC), 1999.

[6] M. Degermark and S. Pink, "*Issues in the Design of a New Network Protocol*," Proceedings Third COST 237 Workshop on Multimedia, Barcelona, Spain, 1999.

[7] C. Partridge and S. Pink, "*A Faster UDP*," IEEE/ACM Trans. on Networking, vol. 1, 1993.

[8] H. Zheng and J. Boyce, "*An improved UDP Protocol for Video Transmission over Internet-to-Wireless Links*," IEEE Transactions on Multimedia, vol. 3, 2001.

[9] L.-A. Larzon, M. Degermark, and S. Pink, "*UDP Lite for Real Time Multimedia Applications*," HPL-IRI-1999-001, 990421, April 1999.

[10] M. Sooriyabandara and G. Fairhurst, "*TCP and UDP over Satellite: Requirements for IP applications*," University of Aberdeen, Aberdeen, UK, Technical Report dom2-UoA-Apr-2001-r1-1, 2001.

[11] Q. Xie and S. Gupta, "*Error Tolerant RTP Payload Format for AMR*," IETF Work in Progress, <draft-xie-avt-et-rtp-amr-xx.txt>

[12] S. Wenger, M.M. Hannuksela, T. Stockhammer, M. Westerlund, D. Singer "*RTP Payload Format for H.264 Video*," IETF AVTWG RFC3984, Feb 2005.

[13] *Video Coding for Very Low Bit Rate Communication,* ITU-T Recommendation H.264, May 2003.

[14] G. Pelletier, "*RObust Header Compression (ROHC): Profiles for UDPLite*," Expired ID: <draft-pelletier-rohc-udplite-01.txt>, 2003.

[15] VLC http://www.videolan.org/

[16] ITU-T Rec. H.264 & ISO/IEC 14496-10 AVC, "Advanced video coding for generic audiovisual services", ITU-T, May 2003.

[17] Saltzer, J., Reed D. and D. Clark, "*End-to-End Arguments in System Design*", ACM Transactions in Computer Systems 2, 4, November, 1984, pages 277-288.

[18] Fairhurst, G., Wood, L., *Advice to link designers on link Automatic Repeat Request*. RFC 3366, August 2002.

[19] Bolea A , Bousquet M., Castanet L., *FMT control loop performance assessment on a point-to-point oriented satellite broadband system for multimedia applications.* Proceedings 10th Ka and Broadband Communications Conference, Vicenza, Italy, October 2004.