UDP-Lite Enhancement Through Checksum Protection

Suherman*, H Abdurrahman, O S Sitompul and N Mubarakah

Electrical Engineering Department, University of Sumatera Utara

*suherman@usu.ac.id

Abstract. UDP drops packet when checksum bits in header does not match the checksum bits calculated from the receive data. UDP-Lite improves UDP by using partial checksum coverage. However, checksum error in both UDP and UDP-Lite headers is potentially leading to wrong decision in dropping part(s) of packet. This article proposes protocol to avoid the problem by using Forward Error Correction (FEC) code to protect checksum bits. The evaluation results show that the number of the received bytes increases up to 0.3% compared to UDP-Lite, which leads to 3 dB increment of the received image quality.

1. Introduction

Checksum is a method to detect bit error within a bit sequence. Basically, checksum adds the bit sequence and folds the results up to 16 digits. User Datagram Protocol (UDP) [1] employs checksum to check error within its packets. If error exists, packet is discharged. On the other hand, many multimedia codecs allow information with error(s) to be decoded. Dropping packets caused by checksum error is not suitable for multimedia applications. Therefore, checksum is often ignored and left empty by using zero padding. However, Internet Protocol version 6 (IPv6) does not allow zero padding on checksum [2], so that UDP should activate the checksum field.

The UDP-Lite [3] tackles UDP problem by reducing checksum coverage only for sensitive data which potentially degrades encoding process if error occurred. The UDP-Lite replaces length field on UDP header to be coverage field. Length field in UDP means the number of octets of all UDP data, while coverage field in UDP-Lite means the octet position of data which is covered by the checksum.

There is a research to improve UDP-Lite performance by adding the same functionality in medium access layer (MAC) which is referred as to mac_lite [4]. The mac_lite treats voice as a priority which removes the MAC layer checksum on voice packets but applies checksum to other types of traffics. Even though it improves UDP-Lite, the mac_lite works only in a single link, depending on the type of mac. If the mac_lite is implemented on all links, the mac_lite potentially degrades the overall performances as each link is busy with error corrections. Furthermore, the mac_lite works in MAC layer, UDP-Lite is on transport layer.

Alfredsson [5] implements UDP-Lite idea on TCP. TCP-lite intercepts the received data to determine acknowledgement packet be sent or not. However, TCP_lite is a reliable protocol for connection oriented application, while UDP-Lite is an unreliable protocol for connectionless application. This paper focuses on UDP-Lite improvement by avoiding error in checksum bits. Schmidt et al [6] proofs that there is a significant error percentage within UDP header. An error on checksum bits may lead wrong decision on dropping parts of packet. This weakness exists on both UDP and UDP-Lite. On the other hand, length field on UDP header is not fully used by the coverage bits of UDP-Lite [7], this empty space can be used for UDP-Lite improvement purpose. As far as the study has been done, there is no work improving...
UDP and UDP-Lite by protecting checksum header using Forward Error Correction (FEC) code. This article is expected to fill the gap.

2. Proposed Method
In order to avoid performance degradation from the wrong decision of dropping part(s) of UDP-Lite packet that is covered by the checksum bits; this paper proposes the use of FEC code for protecting checksum bits. For evaluation purpose, this article employs Hamming code.

The addition of FEC code within UDP-Lite generates redundant or parity bits. However, as aforementioned in Section II, the coverage field on UDP-Lite does not use the whole space of length field; this article allocates the space for FEC redundant bits.

3. Research Method
In order to evaluate the proposed method impact to the streamed data, this article employs NS-2 simulator with network configuration as depicted in Figure 1. The evaluated traffic is taken from akiyo_cif.yuv, encoded using MPEG4 with bitrates 540 Kbps. The 802.11 media is used to introduce error about 5%, 10% and 15%. Constant bit rate (CBR) traffic is employed as background traffic.

![Simulated Network](image)

**Figure 1.** Simulated Network

The performance metrics are the average of delay, packet loss, throughput and PSNR (Peak Signal to Noise Ratio). The proposed protocol is compared to UDP and UDP-Lite. All protocols employ checksum on their payload.

![Delay Characteristics](image)

**Figure 2.** Delay Characteristics
4. Results and Analysis

4.1. Delay and Jitter
Delay characteristics are shown in Figure 2 for each error probability. Protocols delays vary around 45 ms. All protocols have almost similar delay. UDP-Lite and the proposed improvement have more stable jitter than UDP (Figure 3). UDP jitter is more than 8 ms, while UDP-Lite and the proposed one are less than 8 ms.

4.2. Packet Loss
UDP drops packet once it detects checksum error. Packet will be discharged even there is only 1 bit error. It causes UDP experience tremendous packet loss. The UDP packet loss achieves 4.8% for link error rate 5%. The loss increases linearly to link error, which causes UDP suffer 9.25% and 14.64% for link error 10% and 15% (Figure 4).

On the other hand, UDP-Lite is successfully reducing packet loss up to 0.09%, 0.71% and 0.09% subsequently for link error 5%, 10% and 15%. Since UDP-Lite has the capability of avoiding drop packets, checksum error only causes byte reduction in packet.
The proposed protocol is able to improve UDP-Lite further as the proposed protocol avoids misinterpretation of the checksum error. Packet loss of the proposed protocol is stable around 0.09% for all link error values.

![Figure 5. Throughput](image)

4.3. **Throughput and PSNR**

UDP-Lite drops part(s) of the received packet when checksum detects error. The dropped byte causes the number of received bytes reduced. UDP-Lite has no capability to determine whether error occurs in data or in header. The proposed protocol improves this weakness which avoids byte reduction within the received packet when error occurs in header, so that the overall throughput improved (Figure 5).

The proposed protocol is able to increase throughput from 0.13% to 0.4% over UDP-Lite and 3.8% to 13.2% over UDP (Figure 5). In average, checksum protection using FEC within the proposed method is able to increase throughput up to 0.3% over the basic UDP-Lite. Although this improvement is relatively small, the dropped bytes reduction within video frame avoids the frame quality degradation.
When the received packet is reconstructed and decoded to video image, and then compared to the original image, the result is PSNR improvement on the received image where FEC is applied. Figure 6 shows the average PSNR values for the three protocols. PSNR of the proposed protocol increases 1.12% to 6.46% over UDP-Lite and 41.8% to 75% over UDP.

5. Conclusion
To conclude, the FEC protection over the checksum bytes is able to avoid the wrong decision in dropping part(s) of UDP-Lite packet. It is proven by the increase of throughput up to 0.4% over the base UDP-Lite. Even though this increment is relatively small, it is able to improve image quality up to 3 dB in average.

On the other hand, delay and jitter are not affected by the FEC insertion. Delay and jitter are relatively similar. This article uses a simple Hamming code as FEC which results small improvement and stable delay. Future works may apply more complicated FEC codes that may increase the performance even more and its impact on delay can be explored deeper.

References
[1] Postel J 1980 User datagram protocol 1-3
[2] Deering S E 1998 Internet protocol, version 6 (IPv6) specifications
[3] Larzon L A 2002 The UDP lite protocol (Internet Engineering Task Force-Internet Draft).
[4] Yang B and Zhang Z 2011 An improved UDP-lite protocol for 3D model transmission over wireless network Proceedings of the 2011, International Conference on Informatics, Cybernetics, and Computer Engineering (ICCE2011) November 19–20 2011 Melbourne Australia 351-357 Springer Berlin Heidelberg.
[5] Alfredsson S 2001 TCP lite-a bit error transparent modification of TCP Master’s thesis June
[6] Schmidt F Orlea D and Wehrle K 2013 A heuristic header error recovery scheme for RTP In Wireless On-demand Network Systems and Services (WONS), 2013 10th Annual Conference on IEEE 186-190
[7] Fairhurst G and Westerlund M 2013 Applicability statement for the use of IPv6 UDP datagrams with zero checksums (RFC 6936)