Reducing link quality impact to encrypted packets by compressing the transmitted data

Rudi A Candra[1], Dirja Nur Ilham[1], Hardisal[1], E Sipahutar[1], Fardiansyah[1], Suherman[2,*]

[1]Politeknik Aceh Selatan, Aceh-Indonesia
[2]Electrical Engineering Department, Universitas Sumatera Utara, Medan-Indonesia

[*]suherman@usu.ac.id

Abstract. The encrypted traffics are sensitive to losses, while wireless link in prone to error. Although solutions have been proposed in each TCP/IP stack layer, error or losses is always exist. In order to reduce losses of part(s) of encrypted data, bytes or packets should be as many as possible received by the receiver. This paper examines data compression impact in reducing losses of encrypted data. NS-2 simulator is employed to evaluate the study by using the 802.11 radio link. Simulations show that the 3DES decryption rate increases when compression applied. Packet losses can be reduced about 0.32% and 0.54% for TCP and UDP subsequently. The successful encrypted data rises in average 0.3% and 1.3% for both TCP and UDP.

1. Introduction

TCP/IP network stacks allow solutions for link quality limitation performed in many layers. In application layer, concealment is an example of the solution of incomplete data as result network quality degradation. In transport layer, transmission control protocol (TCP) guarantees data transmission by employing acknowledgement services [1]. Various techniques in transport layer have also been proposed [2-5]. Medium access and physical layers provide various choices [6].

The application data is the main objective for these solutions. Some applications are very sensitive to losses such as encrypted data. Packet losses often result to failure to decrypt the cipher text. Many applications apply encryption such as email, e-commerce data and social media packets. In term reliability, those applications rely on TCP for the end to end transmission. When channel quality worsens, TCP often experiences loss packets. Meanwhile, on streamed application, UDP is often employed. UDP provides fast transmission but ignores loss packets.

In order to reduce loss packets, this paper proposes compression performed in radio link layer. Some researchers have applied compression, but mainly for data header [7]. The paper is organized as follows. The assessment method for the encrypted data transmission either compressed or not is outline in research method. Results of of assessments are reported in section results and analysis, followed by the conclusion.

2. Methodology

In order to examine the performance of compressed and uncompressed encrypted message transmission against link quality (packet losses) which increase to number of involved nodes, network simulator (NS-2) [8] is employed. In order to trace the network performance parameters, Evalvid framework [9] is inserted to NS-2 code. The modelled 802.11 radio consists of 2 to 20 mobile nodes as illustrated in Figure 1.

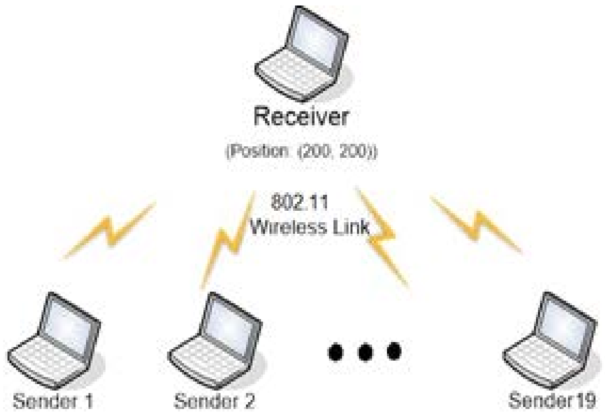

Figure 1. Simulated network configuration

Huffman compression [10] and 3DES encryption [11] techniques are employed. The cypher texts are transported in160 bits packet. Each cypher text slice is packed in TCP segments or UDP datagrams with length of 1024 bytes. Each node transmits data at 200kbps rate. After simulation, the lost packet id is identified and mapped by using the Evalvid software. Afterwards, the received messages were decompressed and decrypted so that successful received packets were obtained.

3. Results and discussion

Figure 2 shows the comparisons of average delay and losses of TCP and UDP in transporting the encrypted data without compression. TCP delay is higher than of UDP caused by the acknowledgment and retransmission services. TCP produces average delay of 82.4 ms while UDP is 41.7 ms. On the other hand, TCP is able to maintain lower loss packet rate than UDP. UDP experiences higher packet loss than TCP because it is fully depending on network quality. UDP packet loss rises from 1% to 5.7% for number of nodes 2 to 20. This low packet loss rates in a crowded network with up to 20 nodes because the rates of the transmitted data is low, about 1024 bytes every 4 ms or 200 kbps. Even dough, the figure shows that in average, UDP has 3.6% losses while TCP is only 0.7% losses.

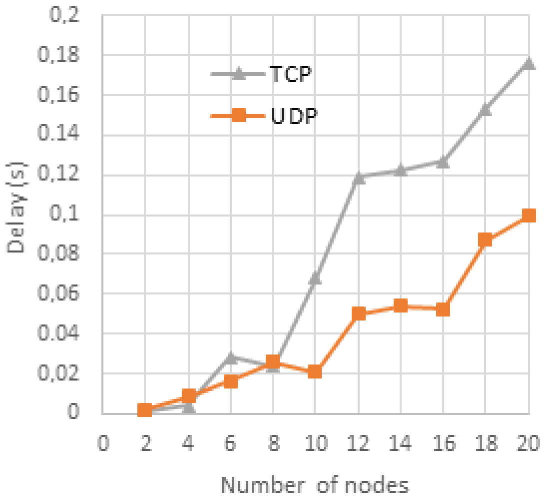

Figure 2. TCP and UDP delays

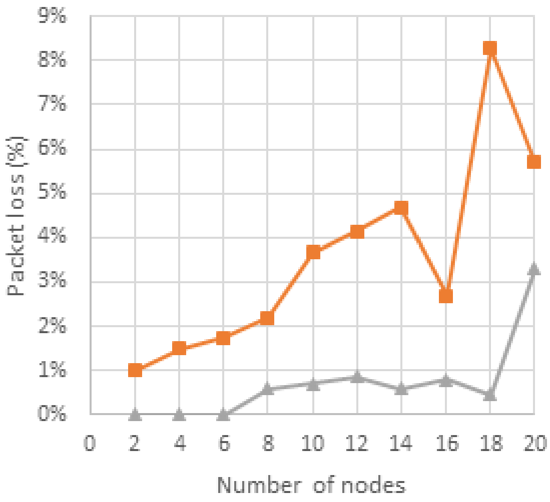

Figure 3. TCP and UDP losses

The successful uncompressed encrypted received packets are depicted in Figure 4. TCP successfully transmitted 99.27% of the encrypted data, while UDP did 96.44%. In decryption process, 3DES algorithm successfully decrypted TCP packets 99.44% of the original messages. TCP successfully delivers data much higher than UDP which generates only 64.5% successful decrypted packets.

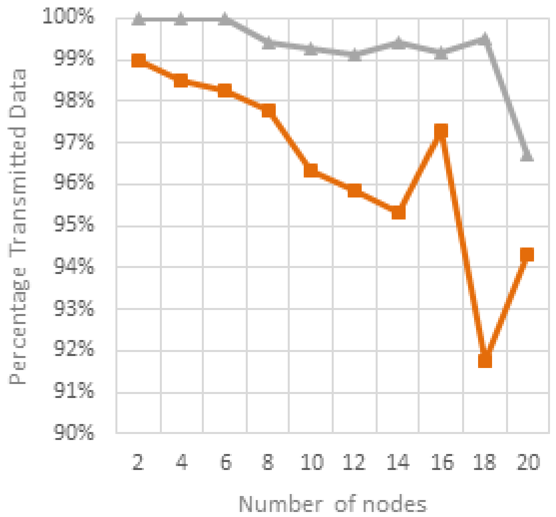

Figure 4. TCP and UDP transmitted packets

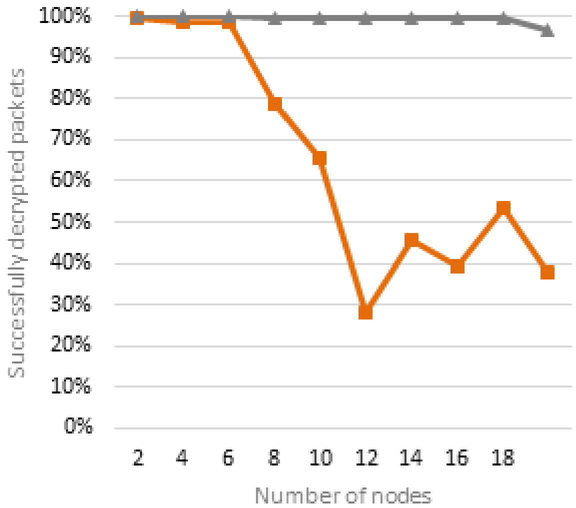

Figure 5. Decrypted packets

By applying data compression with Huffman compression, which by the time evaluation produces 80% compression ratio, the number of successfully transmitted packets increase. Figure 6 and 7 shows the received packet rate for TCP and UDP subsequently. TCP received data increases consistently for all number of nodes. The average increment is 0.32%. In average, UDP experiences 0.54%. Both increments occurred as compression causes less data sent by the sender.

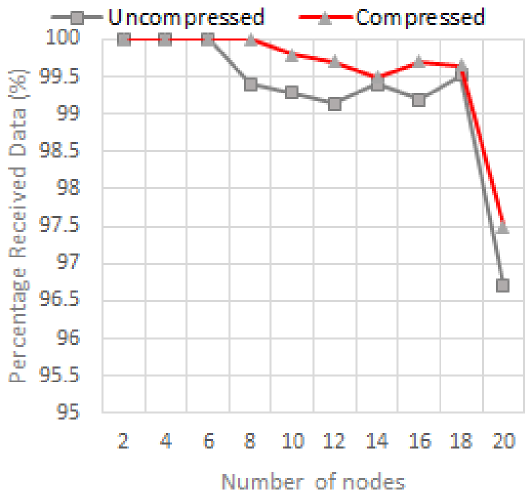

Figure 6. Uncompressed versus compress TCP received packets

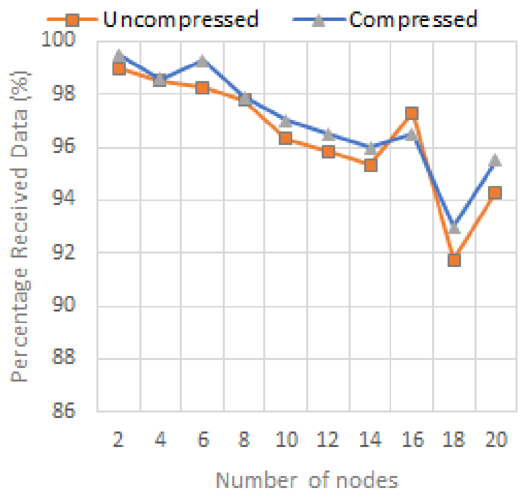

Figure 7. Uncompressed versus compress UDP received packets

After reconstructed the received data, TCP with compressed data successfully decrypted messages up to 99.35% in average. This is 0.3% higher than uncompressed data. Likewise, UDP successfully decrypted message increase about 1.3% from 64.42% to 65.76% after applying compression techniques. Both results are plotted in Figure 8 and 9.

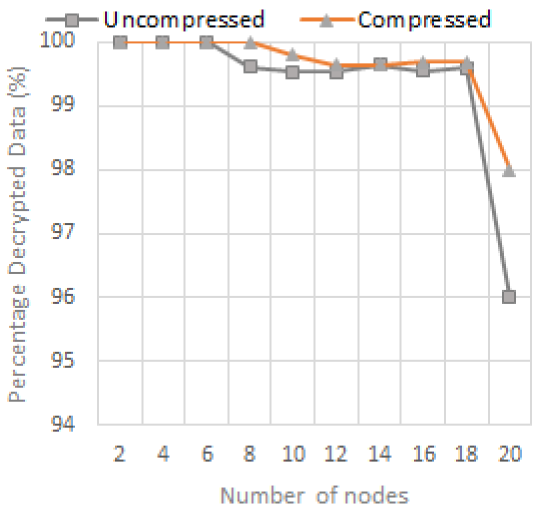

Figure 8. Uncompressed versus compress TCP decrypted packets

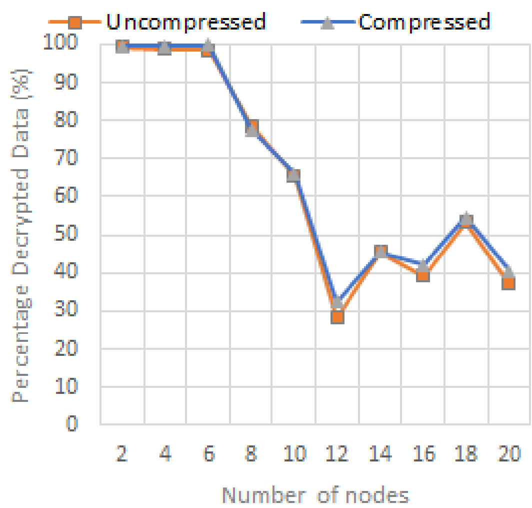

Figure 9. Uncompressed versus compress UDP decrypted packets

4. Conclusion

To conclude, TCP is able to deliver encrypted data better than UDP, so that the decrypted data is much higher than UDP. TCP produces x% decrypted data while UDP x %. These figures show that it is important to supress packet loss caused by link quality as low as possible for encrypted data.

The proposed compression process for wireless link is able to increase the decryption ratio as number of packet losses decrease. The compressed 3DES cipher text by using Huffman code transported by TCP experience decryption rate increment about 0.3% as loss packet decreases 0.32%. UDP experiences higher decryption rate increment up to 1.3% as packet loss decreases 0.54%.

5. Acknowledgement

This research has been funded by the Directorate of Research and Public Services (DRPM) Indonesia under the applied research schema no.D/79.K/POLTAS/IV/2019.

References

[1] Al-Akaidi M., “A review on transport layer protocol performance for delivering video on an adhoc network,” in IOP Conference Series: Materials Science and Engineering, 2017, p. Vol. 237, No. 1, 012018.

[2] Huh K, Seo J. H., “RUDP design and implementation using OPNET simulation.,” in Computer science and its applications, 2015, pp. 913–919.

[3] Suherman, S., Al-Akaidi M., “Increasing uplink broadband video streaming protocol performance in WiMAX network.,” Int. J. Internet Protoc. Technol., vol. 7, no. 3, pp. 176–185, 2013.

[4] Fairhurst T., Jones G, “Transport Features of the User Datagram Protocol (UDP) and Lightweight UDP (UDP-Lite),” 2018.

[5] Abdurrahman, H., Sitompul, O. S., Mubarakah N, “UDP-Lite Enhancement Through Checksum Protection.,” IOP Conf. Ser. Mater. Sci. Eng., vol. 180, no. 1, p. 12146, 2017.

[6] Rudi A Candra, Dirja N Ilham, Mubarakah N., "Various layer techniques to improve video transmission on 20 GHz radio link in a rainy environment". IOP Conference Series: Materials Science and Engineering 2019 Apr (Vol. 506, No. 1, p. 012044). IOP Publishing.

[7] Lee J, Ho SY, Ulupinar F, "inventors; Qualcomm Inc, assignee. Robust header compression for relay nodes". United States patent US 9,674,311. 2017.

[8] E. Issariyakul, T. Hossain, "Introduction to network simulator NS2". Springer Science & Business Media., 2011.

[9] J. Klaue, B. Rathke, A. Wolisz, “Evalvid–A Framework for Video Transmission and Quality Evaluation,” in 13th International Conference, TOOLS 2003, Urbana, IL, USA, September 2-5, 2003. pp. 255–272.

[10] D. Furht, B., & Kirovski, Multimedia security handbook. CRC press., 2004.

[11] Host, S. "Adaptive Source Coding." (2019): 105-131.