УДК 621.391
DOI: 10.22213/2413-1172-2020-2-82-90

Developing AL-ARQ Module for Automatic Measurement of One-Way Data Transmission Delay [*]

M.A. Lamri, Post-graduate, Kalashnikov ISTU, Izhevsk, Russia
I.A. Kaisina, Post-graduate, Kalashnikov ISTU, Izhevsk, Russia
D.S. Vasiliev, PhD in Engineering, Kalashnikov ISTU, Izhevsk, Russia
A.V. Chunaev, PhD in Engineering, Izhevsk, Russia
A.V. Abilov, PhD in Engineering, Associate Professor, Kalashnikov ISTU, Izhevsk, Russia

In different video streaming services and applications in mobile wireless networks, the QoS may not satisfy the users' desire if the one-way transmission delay (end-to-end delay) and delay variations between hosts are large relative to some threshold values. In this paper, we develop a new module for measuring and investigating a one-way transmission delay, as well as delay jitter for the application layer automatic repeat-request (AL-ARQ) algorithm based on UDP by adapting the protocol header for delay metrics and updating receiver's playback buffer. We created a standalone network where a microcomputer (Raspberry Pi) sends video streaming to a destination node (HP laptop) using the 802.11g standard. We used the NTP server for time synchronization between end-devices to provide a frame of time reference for delay metrics as well as delay variation (Jitter).

The results showed the importance of the module proposed and its effectiveness in measuring AL-ARQ transmission delay and delay jitter at a high level of accuracy during the requesting and the retransmission of lost packets. It showed that the retransmission mechanism of AL-ARQ algorithm affects both maximum and average values of one-way transmission delay, which increases by increasing in mean packet loss rate and burst length. Moreover, it also showed that the jitter of lost packets is spread more widely than non-lost packs. Although AL-ARQ is still under the requirements of a good video streaming provider for small standalone networks. However, using the results of the module proposed, an adaptive jitter buffer could be implemented to refine the dynamics of the AL-ARQ buffering mechanism and enhance its QoS in more complicated scenarios.

Keywords: ad-hoc network, ARQ, delay, jitter, UDP, video streaming, QoS.

Introduction

The Open system interconnection model (OSI) is a conceptual standard used to describe the functions of a network system. OSI transport layer provides a mechanism for the exchange of data between end systems; it presents transmission control protocol (TCP) and User datagram protocol (UDP) as two main transport protocols, which provide connection-oriented and connectionless services respectively. TCP ensure ordered and reliable data delivery while also introducing processing overhead and bandwidth limitations due to flow and congestion control mechanisms [1]. The lightweight UDP defined as a non-reliable protocol that doesn't guarantee data delivery and doesn't suffer from processing overhead and bandwidth limitation and hence is used in time sensitive applications because dropping packets is preferable than waiting for delayed packets, which may not be an option in a real-time system like VoIP, streaming data and online gaming.

Real-time network applications such as video streaming depend on the ability to measure and monitor performance metrics for packet loss and one-way delay. A one-way delay (also known as end-to-end delay) higher than a given threshold may compromise the quality of video streaming [2]. Jitter (variation in the latency on a packet flow between two systems) is another key factor that impairs the quality of such applications by creating drop-outs and video artefacts which ends up as the same as if there was a packet loss, but there are no actual lost packets.

According to [3], delay and jitter must be lower than 200 ms and 50 ms respectively, to ensure a good quality of experience in video streaming. Meeting such requirements can be achieved using UDP as transmission protocol since it is based on unreliable packet delivery schema, which doesn't require much time to handle with acknowledgments and dropped packets. However, the high packet loss rate in some specific classes of wireless networks

* The reported study was funded by RFBR, project number 19-29-06076 (Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта №19-29-06076).

such as mobile ad-hoc networks makes UDP suffer to outcome the best QoS.

There are proposals in the literature for refinement and amelioration of UDP QoS for video streaming by injection of one of the methods for packet loss recovery on the application layer. One of the most effective algorithms is AL-ARQ [4] which showed a good performance in different scenarios [5] for handling burst losses since it's based on a lightweight automatic repeat request mechanism to feedback loss-packet events and packet retransmission. Another good feature for AL-ARQ is its ability to reorder the out-of-sequence arriving packets on destination node by integrating the buffering mechanism before presenting the data as an incoming stream for displaying.

However, those improvements on the application layer have a number of drawbacks, one of which is the impact on theQoS performances' metrics such as end-to-end delay and delay jitter. To solve this problem, a measurement and investigation of one-way transmission delay and delay jitter will be handled in this paper to check whether AL-ARQ algorithm meet the needs for a good real-time applications' provider in standalone networks.

Therefore, since AL-ARQ is based on UDP transmission protocol, an important scientific task is to develop a metric model for measurement of one-way transmission delay and delay jitter for AL-ARQ during the detection and retransmission of lost packets.

We consider a test bed with: source-node and destination-node. Both nodes have ad-hoc connection between each other and exchange video data using 802.11g transmission standard. Source-node A will provide HD video stream, while destination-node B will be considered as a base station. Mobile ad-hoc networks are characterized by high mobility and unreliable transmission channels. Such characteristics influence on the QoS parameters such as packet loss rate (PLR) and burst length distribution (BL). Because of this, an artificial packet loss schema with a variable burst length distribution was implemented to simulate the packet-loss event during the experimentations to better investigate the behaviour of one-way delay as a function over burst length and, Jitter as a function of variation in delay time. Moreover, since our scenario is considered as a standalone network, we used Network Time Protocol (NTP) for clock synchronization between end devices to provide a frame of time reference for measuring delay.

The rest of this paper is organized as follows. Section 2 contains a preview about delay and jitter in real-time applications. Section 3 contains description of AL-ARQ transmission algorithm and packet delay calculation. In section 4 we evaluate the performance of the mechanism proposed for calculating the one-way transmission delay and in section 5 we present our conclusions.

Delay and Jitter in Real-time applications

Real-time traffic requires strong bounds on packet delay and Jitter, the effect of those components appears at different stages of packet transmission. An analysis of those different components is essential tounderstand the overall cause of the irregular arriving interval of packets and determining a betterQoS performances [6, 7]. The different types of delays encountered in a packet-switched network are shown in (Fig. 1).

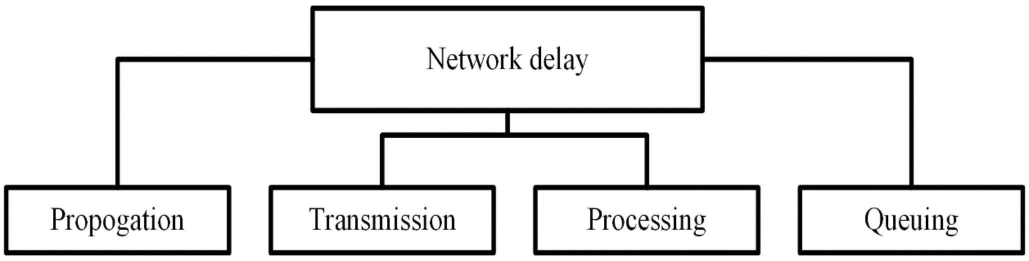

Fig. 1. Types of network Delay

• ***Propagation delay:*** defined as the flight time of packets over the transmission link and is limited by speed of light. The value of propagation delay represents the physical limits and cannot be reduced.

• ***Transmission delay:*** is the time taken to transmit the packet over the link, this delay depends on multiple factors including: the number of active sessions, transmission capacity of the link, medium access control (MAC) access delay and context switch in the OS. Reduction in this delay requires supporting the operating system with real-time scheduling algorithms, increasing transmission capacity of the link, choosing a suitable MAC protocol for accessing the link, enhancing the device drivers, and increasing the operating speed of the device [8].

• ***Processing delay:*** is the amount of delay faced at both the source and the destination. This delay presents the time taken to packetize data through different layers of protocols; it may include also the time to convert analogue data to digital form. This delay depends on the characteristics of multimedia application and the OS, and it is independent of the network model. Any reduction in this delay will require software enhancements and improvements on the application level or to the OS Kernel [9].

• ***Queuing delay:*** The queuing delay depends on several system specifications like, the arrival process, the service time distribution, the available number of links, and the queue's maximum allowed length. It is random because when the packet arrives at a queue, it has to wait for a random time before it gets processed and hence, it is considered as the major contributor to jitter in the traffic stream [10]. In sensitive traffic, when queuing delay reaches a defined threshold, the sender (or receiver) application times out and resends (or ask for) the packet again which can lead to a flood effect that increases the delay [11].

For our case, AL-ARQ which is considered as a lightweight reliable transmission protocol UDP based algorithm, implements a selective repeat approach of automatic repeat request (ARQ) algorithm for flow control on the application layer [12, 13]. The destination detects loss events by the sequence number (SN) which is included in the header of each sent packet, and forms a negative acknowledgment (NACK) that includes SNs of lost packets, the destination node will keep asking for the packet every period of time called retransmission time out (RTO), while the packet is still relevant to the application. The relevance of the packet is determined by the buffer size and the delay; this buffer is used to maintain the packets' order in a given amount of time, after this, the packet's payload will be processed and sent to be displayed. Due to this mechanism, if a packet is lost in the network, it would suffer additional delay and this will affect the delay jitter [14]. To evaluate how much additional packet delay is caused by in-order delivery in AL-ARQ algorithm, a full description scheme of AL-ARQ will be presented in the following section.

AL-ARQ: transmission algorithm and packet delay calculation

Transmission algorithm

As mentioned in the previous section, AL-ARQ is based on selective repeat ARQ approach. The pseudo algorithm in (Fig. 2) shows two important parts of AL-ARQ source-node, where the application distinguishes when the data coming through the UDP socket is data coming from a video provider (video player, video camera) or a Negative acknowledgment (NACK) from the destination node. The application gets the first byte of received data as an identifier (ID), thus, if this ID is equal to the ID of NACK (ID_ARQ_NACK_AL), the algorithm will check if the requested packet (using the sequence number SN fed back in the NACK) still in the playback buffer to retransmit it again. If no, the source will send a cancellation message to inform the destination node to delete requested packet from the waiting list.

The second part of the algorithm's pseudo code shown in (Fig. 2) handles the case where the application listens to data coming from the video provider. In this case a new packet will be generated as shown in (Fig. 3). The first 17 bytes of data will be reserved for the header (AL-ARQ uses a 13-byte header size and 4 additional bytes for timestamp value). The header contains different information like Global Sequence Number (SNg) which is used to detect loss-event in tree and mesh topologies, and Video Quality Priority (VQP) which defines the priority of the cadre according to the video quality [15].

The additional field in AL-ARQ header is the timestamp (T_S). This option consumes 4 bytes and it contains the time in microsecond when the packet was generated. The T_S uses the format of network time protocol (NTP) [16], but it takes only 1 byte for seconds and 3 bytes for the fractional second. After generating the header, the application adds the data payload to the packet and saves the packet in the playback buffer then sends it to the destination node.

The (Fig. 4) shows the AL-ARQ behaviour on the destination node when the data received on UDP socket is a video packet, which will be distinguished by the value of the first data byte marked as the packet's ID.

The video packet received could be a lost requested packet during the in-order buffering. The application gets the timestamp value from the header and subtracts it from the current clock time value (in microseconds). Subsequently, it will check if this packet exists in the display buffer since the NACK combines few burst losses together. If so, the packet will be dropped, else, it will be set in order in the display buffer.

```
Begin
   Initialize buffer_in [], buffer_out [];
   buffer_in [ ] = receive_data (UDP_socket);
   bytes_in = buffer_in[ ].length;
  While (bytes_in > 0) {
     Id = buffer_in [0];
     If (Id = ID_ARQ_NACK_AL) {
     If (get_packet_from_playbackBuffer ==exit_succes)
        { send_packet () ; }
     Else send_cancellation_msg() ;
     }//end If (Id = ID_ARQ_NACK_AL)
     If (bytes_in >=VIDEO_CHUNCK_SIZE){
        Add_ARQ_header (ID, NR ,SN, SNg, Vqp );
        Timestamp = get_current_time ();
        Add_ARQ_header (Timestamp);
        Add ARQ_header to buffer_out;
        Add data from buffer_in to buffer_out;
        New_packet (buffer_out);
        Copy packet in playbackBuffer ();
        send_packet ();
     }// end If (bytes_in >=VIDEO_CHUNCK_SIZE)
    bytes_in = 0;
  }
End
```

Fig. 2. Pseudo algorithm for Sending data packet or retransmit lost packet

ID	RN	TN	SNg	SN	VQP	T_S	Payload	
1	1	1	4	4	2	4	*N*	Bytes

Fig. 3. AL-ARQ packet format

```
Begin
 Initialize buffer_in [], buffer_out [];
  buffer_in [ ] = receive_data (UDP_socket);
  bytes_in = buffer_in[ ].length;
  While (bytes_in > 0) {
     Id = buffer_in [0];
     If   (Id= ID_ARQ_VIDEO || Id= ID_ARQ_VIDEO_R){
          Get_current_time();
          Get timestamp from packet header;
          Pckt_delay = current_time - timestamp;
          If (SN_packet exist in buffer) {
             delete packet ; }
          Else  Copy packet to display buffer;
                Write packet in log-file for statistics;
      }// end if
 }// end while
End
```

Fig. 4. Pseudo algorithm for receiving data packet

Delay measurement

As noted in the previous section, measurement of one-way delay requires clock synchronization between the involved nodes; the measurement accuracy will be limited by the quality of the synchronization. The NTP synchronization format used in our scheme has the advantages of wide use and long deployment in the Internet [17]. However, for an accurate measurement of one-way delay in standalone networks, our approach implements the following features.

• In some cases, one-way delay can be as low as $100 \mu sec$ [18] which is not the case in some NTP synchronization periods between end systems because of NTP offset. Testing the NTP synchronization before the transmission will give us the exact value of offset which is used to correct clock synchronization on source node by subtracting it from the timestamp value:

$$TimeStamp = TimeStamp - offset, \qquad (1)$$

where *TimeStamp* – current time when the packet was formed; *offset* – difference in time between nodes using NTP server.

• The burst length causes duplicate retransmission of lost packets. A multiple copies arrive at the destination and the packet is counted as received, however, only the first copy arrived will determine the packet's one-way delay.

• The approach is intended to investigate the one-way transmission delay and delay jitter caused by packet loss. However, the global application delay is measured as the sum of all delay components contributed in different stages:

$$App_{del} = \text{Propagation}_{del} + \text{Processing}_{del} + \\ + \text{Queuing}_{del} + \text{Transmission}_{del}, \qquad (2)$$

where Propagation$_{del}$ – propagation delay; Processing$_{del}$ – processing delay; Queuing$_{del}$ – queuing delay; Transmission$_{del}$ – transmission delay.

Delay Jittermeasurement

Jitter is considered as one of the key statistics for characterizing the temporal performance on a network, it is defined as the absolute value of the difference between the forwarding delays of two consecutive packets [19]. Excessive jitter could cause the end node buffer to overflow or underflow. However, AL-ARQ used a statistic destination node buffer to manage the data received, this buffer introduces the queuing delay and it doesn't depend on delay deviation since it is implemented independently.

In more complicated scenarios using AL-ARQ, the static buffers introduce decreasing in QoS delivering; an injection of an accurate measurement of Jitter helps to develop a dynamic buffer mechanism to manage the received data and reduces application delay (Fig. 5) presents true real-time jitter measurement flowchart which takes into account lost or corrupt packets [20].

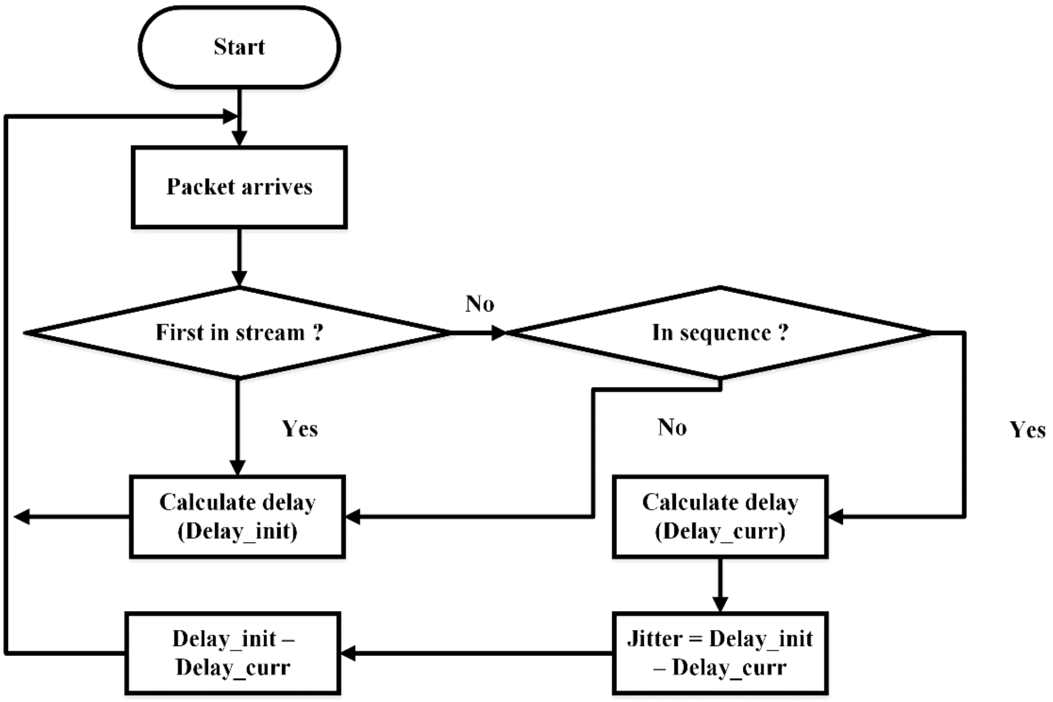
Start
Packet arrives
First in stream ?
No
In sequence ?
Yes
No
Yes
Calculate delay (Delay_init)
Calculate delay (Delay_curr)
Delay_init – Delay_curr
Jitter = Delay_init – Delay_curr

Fig. 5. Jitter calculation flowchart

The measurement is defined by two types of delay calculation, if the packet received is the first packet in the stream then packet transmission delay (Delay_init) is calculated and stored, if not, a check will be performed to make sure that the packet is in sequence. If the packet is out of order then the jitter calculation will be discarded and the current packet will be marked as the first packet in stream, else, the jitter will be calculated by taking the difference of two consecutive measured delays:

$$\text{Delay} = \text{Delay}_{init} - \text{Delay}_{curr}, \qquad (3)$$

where Delay_{init} – Packet delay for first or not-in-order packet; Delay_{curr} – Packet delay for in-order packets.

Experimental Scenario

Test bed

We conduct experiments to investigate the QoS of AL-ARQ using a raspberry Pi running on Ubuntu Mate 16.04 operating system as a data source-node and an HP laptop running on Ubuntu Mate 16.04 as a destination-node. Both are connecting using 802.11g. We created a stand-alone network topology where the two devices are connected on an ad-hoc regime. The results of the experiments are represented in figures 6 to 8.

We used the artificial network packet loss for measuring and investigatingthe delay and the delay jitter. The author in explained the relationship between PLR metrics and distance in real experiments and showed that the PLR = 0.075 indicate the maximum value in which the quality of the video transmitted still considered satisfactory, we based on these measurements of PLR to set the artificial network packet loss.

Experimental parameters

Parameter Name	Value
Operating system	Ubuntu mate 16.04
Application layer	AL-ARQ
Video Coding	H.264
Transport layer	UDP
Wireless standard	802.11
Video packet size	1.5 Mb
NTP Offset (*ms*)	[0.1 ; 0.8]

Discussion

Figure 6 shows the difference between variations of one-way transmission delay with packet loss and without packet loss for 1000 consecutive packets. It shows that additional packet transmission delay was occurred because of retransmission of lost packets. The results show maximum delay of 93 ms in packet loss scenario and 15 ms in non-packet loss. The average of one-way transmission delay without packet loss was around ±2 ms and the average for QoS metrics (PLR = 0.07 and BL = 10) was ±6 ms which doesn't rise excessively. This shows that the retransmission mechanism affects both maximum and average values of one-way transmission delay.

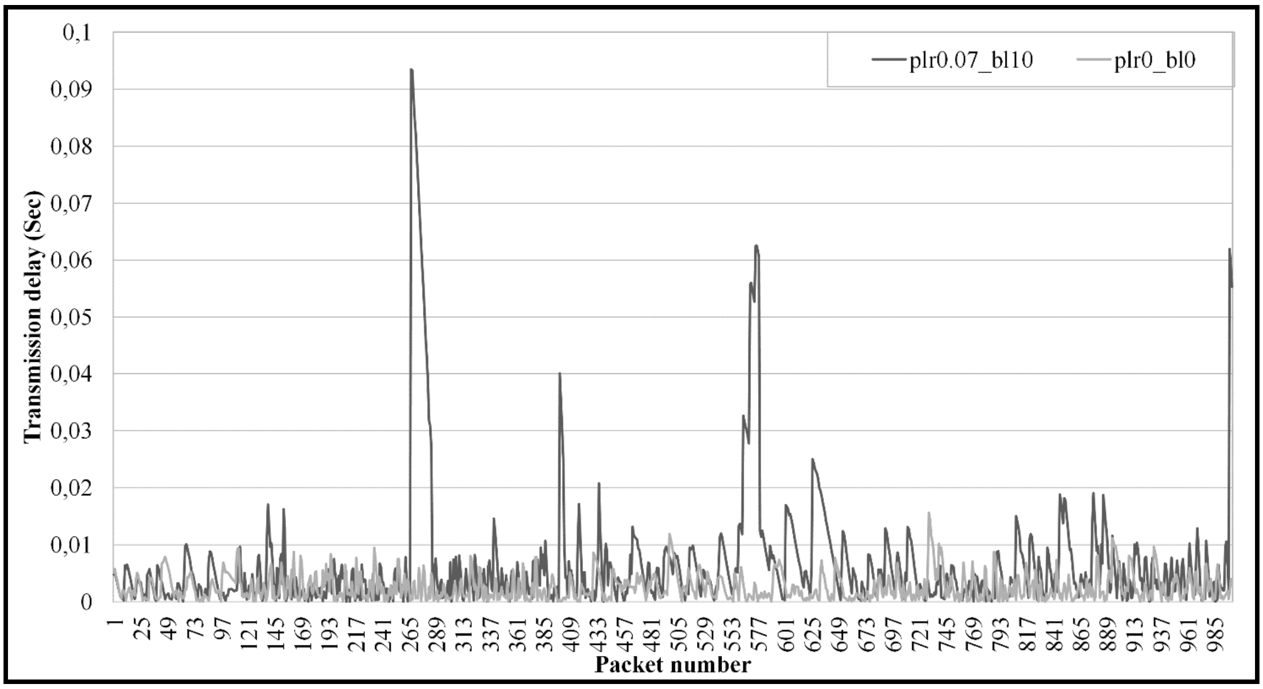

Fig. 6. One-way packet delay values during experimental runs of AL-ARQ algorithm with packet loss (PLR = 0.07, burst length is 10 packets) and without it

Figure 7 represents the measurement of the average one-way packet transmission delay for 1000 consecutive packets as a function of burst length for different packet loss rates. The figure shows the increasing in average delay caused by the increasing in mean packet loss rate and burst length. This is introduced by the retransmission of a lost packet because the sender needs extra time to detect and resend it. AL-ARQ recorded maximum one-way transmission delay for PLR = 0.01 and BL = 20 was equal to 27 ms. In the worst case of packet loss events (PLR = 0.07 and BL = 20), the maximum average transmission delay was equal to 43 ms. These values are very acceptable since AL-ARQ one-way transmission delay has to go through the application layer which requires more processing delay.We denote that the global application delay is determined by the four delay components presented in section II, in which, the queuing delay is considered as the main component since the application layer buffer has fixed length.

Figure 8 shows the variation of Jitter and transmission delay for 1000 consecutive packets for QoS metrics of PLR = 0.07 and BL = 10. The results are showing that jitter of lost packets is spread a little bit more widely than that of non-lost packets. AL-ARQ recorded an interval of jitter between [–0.006; 0.009] (values are in seconds). These results are providing a simple overview on the jitter values since the emulated scenario is a simple P2P network, which doesn't implement any relay nodes or central servers; however, these measurements will provide AL-ARQ in future work with accurate jitter metrics to implement dynamic jitter buffer on the destination node to reduce the application delay [21].

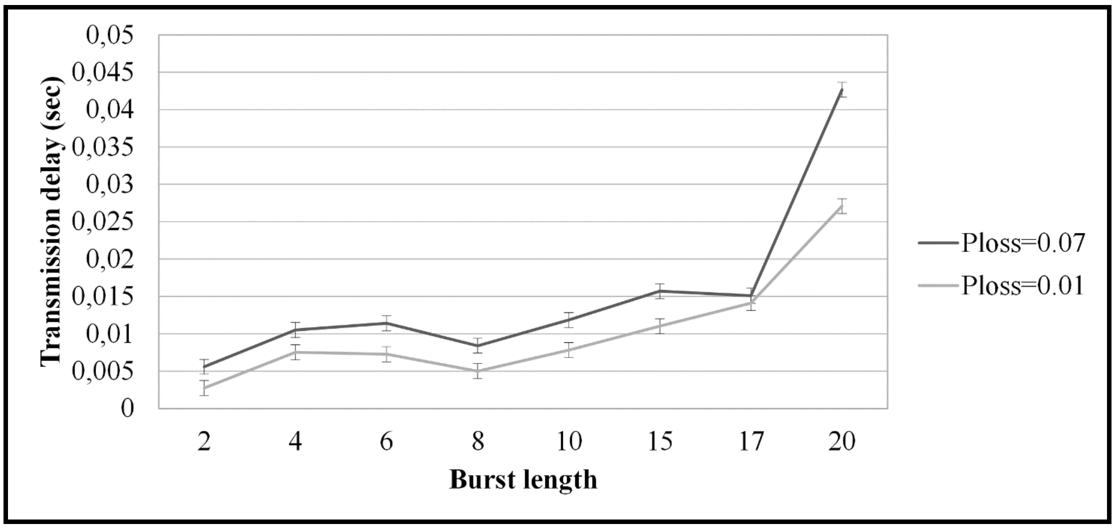

Fig. 7. Average one-way transmission delay during the experimental run of AL-ARQ algorithm with packet loss, PLR = {0.01,0.07}

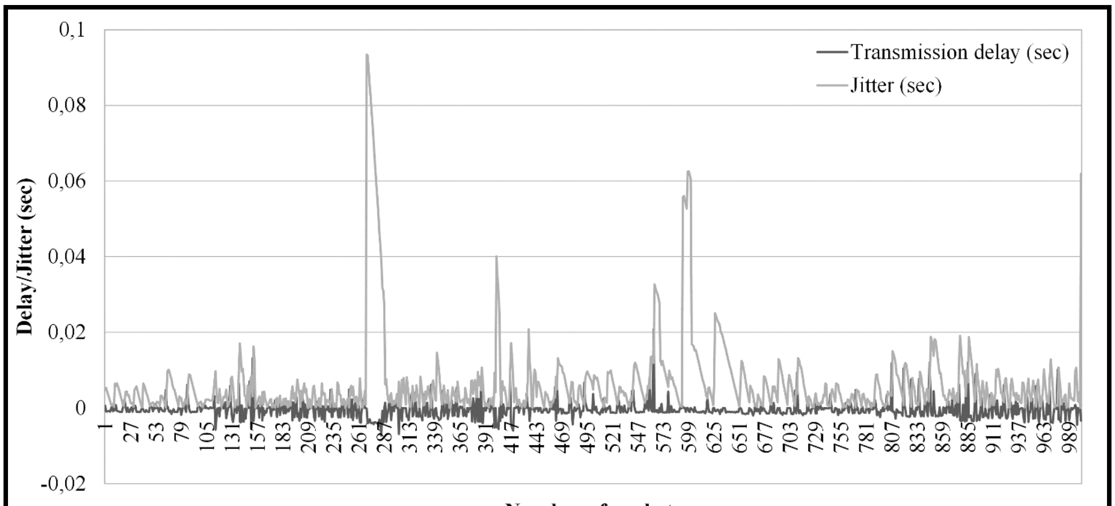

Fig. 8. One-way delay and jitter values during the experimental run of AL-ARQ algorithm with packet loss (PLR = 0.07, burst length is 10 packets)

Conclusion

We have analysed the performance of a light-weight reliable protocol AL-ARQ algorithm on the application layer by investigating the one-way transmission delay and delay jitter. The results showed the importance of the model proposed and its effectiveness to measure AL-ARQ transmission delay and delay jitter at a high level of accuracy during the requesting andthe retransmission of lost packets. These results are also considered as key factors to improve the QoS of AL-ARQ algorithm in further complicated scenarios where relay nodes should be implemented. Moreover, the results showed that AL-ARQ still requires a good real-time applications' provider, although an adaptive jitter buffer should be implemented which will be considered in a future work, to refine the dynamics of AL-ARQ and enhance its QoS.

References

1. Yuan-Cheng Lai, Ahsan Ali, Md. Shohrab Hossain, Ying-Dar Lin. Performance modeling and analysis of TCP and UDP flows over software defined networks. *Journal of Network and Computer Applications*, 2019, vol. 130, pp. 76-88. ISSN: 1084-8045.

2. Yusoff N. M. Performance comparison of video streaming application between Mobile WiMAX and UMTS. 4th Control and System Graduate Research Colloquium. IEEE, 2013, pp. 86-92.

3. Lee M. G., Lee S. Delay analysis for statistical real-time channels in mobile ad hoc networks. 10th International Workshop on Object-Oriented Real-Time Dependable Systems. Sedona, Arizona, USA – IEEE, 2005, pp. 363-370.

4. Vasiliev D. Application Layer ARQ and Network Coding for QoS Improving in UAV-assisted networks. 25th Conference of Open Innovations Association (FRUCT). IEEE, Helsinki, Finland, 2019, pp. 353-360.

5. Vasiliev D.S., Meitis D.S., Abilov A. Simulation-based comparison of AODV, OLSR and HWMP protocols for flying Ad Hoc networks. International Conference on Next Generation Wired/Wireless Networking. St. Petersburg, Springer, 2014, pp. 245-252.

6. Bertsekas D.P., Gallager R.G., Humblet P. Data networks. New Jersey, Prentice-Hall International, 1992, vol. 2.

7. Martinez J., Sañudo I., Bertogna M. Analytical characterization of end-to-end communication delays with logical execution time. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2018, vol. 37, no. 11, pp. 2244-2254.

8. Khanvilkar S., Bashir F., Schonfeld D., Khokhar A. Chapter 7 - Multimedia Networks and Communication. The Electrical Engineering Handbook, Academic Press, 2005, pp. 401-425. ISBN: 9780121709600.

9. Ramaswamy R., Weng N., Wolf T. Characterizing network processing delay. IEEE Global Telecommunications Conference GLOBECOM'04. IEEE, Dallas, TX, USA, 2004, vol. 3, pp. 1629-1634.

10. Ana I., Pérez-Neira, Marc R.C. Chapter 7 - Different views of delay in resource allocation for wireless systems, Cross-Layer Resource Allocation in Wireless Communications, Academic Press, 2009, pp. 125-149. ISBN: 9780123741417.

11. Shashank Khanvilkar, Faisal Bashir, Dan Schonfeld, AshfaqKhokhar, 7 - Multimedia Networks and Communication, The Electrical Engineering Handbook, Academic Press, 2005, pp. 401-425. ISBN: 9780121709600.

12. Chunaev A.V., Abilov A.V., Pavlova M.M. [AL-ARQ Algorithm for Streaming Video Delivery on a Wireless LAN]. *Infokommunikatsionnye tekhnologii*, 2015, vol. 13, no. 1, pp. 68-73 (in Russ.).

13. Vasil'ev D.S., Chunaev A.V., Abilov A.V. [An experimental study of the quality of video transmission in a P2P tree with an application level ARQ algorithm]. *T-Comm: Telekommunikatsii i Transport*, 2014, vol. 8, no. 1 (in Russ.).

14. Fallah Y.P., Sengupta R. Protocol Design for Real-Time Estimation Using Wireless Sensors. The Art of Wireless Sensor Networks. Springer, Berlin, Heidelberg, 2014, pp. 201-234.

15. Jie X., Liang X., Lian D. and Delin Z. [Research on network timing system based on NTP]. 2017 IEEE 13th International Conference on Electronic Measurement & Instruments (ICEMI), Yangzhou, China, 2017, pp. 356-360.

16. Capuni I., Zhuri N., Dardha R. TimeStream: Exploiting video streams for clock synchronization. Ad Hoc Networks, 2019, vol. 91, pp. 101878.

17. Almes G. A one-way delay metric for IP performance metrics (IPPM). RFC 7679 IETF, 2016.

18. Poretsky S. Terminology for benchmarking network-layer traffic control mechanisms. IETF RFC 4689, 2006.

19. Cocker E. Measurement of buffer requirement trends for real time traffic over TCP. 2014 IEEE 15th international conference on high performance switching and routing (HPSR). IEEE, Vancouver, BC, 2014, pp. 120-124.

20. Batra S., Chu Y., Bai Y. Packet buffer management for a high-speed network interface card. 2007 IEEE 16th International Conference on Computer Communications and Networks. IEEE, Honolulu, HI, 2007, pp. 191-196.

21. Li Z., Wu W., Ren H. Research on Jitter Buffering Algorithm Based on E-MODEL Optimized Speech Quality. 2018 IEEE 3rd International Conference on Cloud Computing and Internet of Things (CCIOT). IEEE, Dalian, China, 2018, pp. 157-161.

Разработка программного модуля AL-ARQ для автоматического измерения задержки односторонней передачи данных

М. А. Ламри, аспирант, ИжГТУ имени М. Т. Калашникова, Ижевск, Россия
И. А. Кайсина, аспирант, ИжГТУ имени М. Т. Калашникова, Ижевск, Россия
Д. С. Васильев, кандидат технических наук, ИжГТУ имени М. Т. Калашникова, Ижевск, Россия
А.В. Чунаев, кандидат технических наук, Ижевск, Россия
А. В. Абилов, кандидат технических наук, доцент, ИжГТУ имени М. Т. Калашникова, Ижевск, Россия

В мобильных самоорганизующихся сетях часто наблюдаются задержки при передаче потоковых данных и снижение QoS. В статье представлен разработанный модуль для измерения задержки и джиттера. Модуль встроен в программу автоматического запроса повторной передачи прикладного уровня (AL-ARQ). Для оценки эффективности работы модуля была собрана сеть, состоящая из узла-источника (микрокомпьютер RaspberryPi) и узла-получателя (ноутбук HP). С узла-источника отправлялось потоковое видео к узлу-получателю с использованием беспроводного стандарта 802.11g. Для синхронизации времени между устройствами применялся NTP-сервер.

Результаты эксперимента показали эффективность предложенного модуля для измерения задержки и джиттера при передаче потоковых данных. При исследовании работы модуля было выявлено, что механизмы повторной передачи AL-ARQ влияют как на максимальное значение задержек при передаче, так и на среднее значение за время всей передачи. Был сделан вывод, что снижению уровня задержек и общему улучшению QoS может способствовать адаптивный буфер.

Ключевые слова: самоорганизующаяся сеть, AL-ARQ, задержка, джиттер, UDP, потоковая передача данных, QoS.

Получено 27.04.2020

For Citation

Lamri M.A., Kaisina I.A., Vasiliev D.S., Chunaev A.V., Abilov A.V. [Developing AL-ARQ Module for Automatic Measurement of One-Way Data Transmission Delay]. *Vestnik IzhGTU imeni M.T. Kalashnikova*, 2020, vol. 23, no. 2, pp. 82-90. DOI: 10.22213/2413-1172-2020-2-82-90.