

Network Technology for Transmission of Visual Information

Stanislav Bieliyevtsov¹, Igor Ruban¹, Kyrylo Smelyakov¹, Dmytro Sumtsov¹

¹Kharkiv National University of Radio Electronics, Kharkiv, Ukraine
stanislav.bieliyevtsov@nure.ua, ruban_i@ukr.net,
kirillsmelyakov@gmail.com, dmytro.sumtsov@nure.ua

Abstract. At present, the information and communication systems and technologies are developing at a very high rate and are becoming widely available. At this, with every year there are more and more programs appear on the market which allows communication between people at a great distance with the use of network technologies and the Internet. Many already known programs provide intercourse for their users with the use of reliable communication for free or almost free of charge; for example: Skype, Viber, NetMeeting, Net Speakerphone, Team Speak, and Discord. All these programs and technologies are called IP-telephony; it includes a variety of technologies that provide the transfer of multimedia data (voice, video and various multimedia) over computer networks and the Internet. They are based on various real-time streaming protocols. Many companies and users around the world use IP-telephony services to support remote communication and video communication between people in real time. In this regard, the work considers the features of the functioning of various protocols that are used to transmit visual information between network users and users of various programs and subnets. Most of them use the same network protocols, but have different ways of dealing with network problems, such as insufficient carrying capacity or network losses, which in one way or another affect the quality of real time communication, and, as a result, the quality of the displayed visual information. In particular, the features of several most widely used multimedia network real time transfer protocols (video data with sound - video conferencing) are studied under certain limitations and losses. As well, the results of experiments, the obtained estimates of protocols effectiveness, and the peculiarities in operation of various protocols under the same test conditions are given.

Keywords: Protocol, Carrying Capacity, Efficiency, Packet Loss, Frame Frequency, visual information

1 Analysis of Efficiency of the Key Technologies

One of the most important trends in the evolution of modern telecommunications is the development of IP telephony – a multitude of new technologies that provide the transmission of multimedia messages (voice, data, video) over computer networks on the basis of the IP protocol, including local, corporate, global computer networks and the Internet. The concept of IP-telephony includes Internet telephony, allowing organizing telephone communication between subscribers of the Internet, between sub-

scribers of Public Switched Telephone Network (PSTN) through the Internet, as well as telephone communication of subscribers of PSTN and Internet with each other.

IP-telephony has a number of undoubted advantages, ensuring its rapid development and expansion of the computer telephony market. It is beneficial to end users who are provided with a telephone connection at a rather low per-minute payment or for no charge at all. For companies with remote branches, IP telephony technology allows voice communications to be organized using existing corporate IP networks. Instead of several communication networks, one is used. The undoubted advantage of IP telephony over a regular phone is also the possibility of providing additional services through the use of a multimedia computer and various Internet applications. Thus, thanks to IP-telephony, enterprises and individuals can expand communication capabilities by incorporating modern video conferencing, application sharing, etc.

In networks that do not provide guaranteed quality of service, packets may be lost, the order of their arrival may change, data transmitted in packets may be distorted. Under these conditions, various procedures of the transport layer are used to ensure reliable delivery of the transmitted information. When transmitting digital data, the Transmission Control Protocol (TCP) is used for this purpose. This protocol provides reliable data delivery and restores the original order of the packets. If an error is detected in the packet, or the packet is lost, the TCP procedures send a request for re-transmission.

For audio and video conferencing applications, packet delays have a much greater effect on signal quality than an individual corruption of data. Differences in delays can lead to pauses, "lags". For such applications, a different transport-level protocol is needed, which ensures the restoration of the original sequence of packets, their delivery with minimal delay, real-time playback at exactly specified moments, recognition of traffic type, group or two-way communication. This protocol is the Real-Time Transport Protocol (RTP) which regulates the transmission of multimedia data in packets over a computer network at the transport level and is complemented by the Real-Time Control Protocol (RTCP). The RTCP protocol, in turn, provides control of multimedia data delivery, quality of service control, transfer of information about participants in the current communication session, management and identification, and is sometimes considered as part of the RTP protocol. As well, many other protocols are used in IP-telephony.

UDP (User Datagram Protocol) is one of the key elements of the TCP / IP stack. Using UDP, the computer applications can send messages (in this case called datagrams) to other nodes via the IP network without having to send a preliminary message to set up special transmission channels or data paths. UDP uses a simple transmission model, without any implicit "handshakes" to ensure reliability, ordering, or data integrity. Thus, UDP provides an unreliable service, and datagrams may come out of order, be duplicated, or not reach the addressee at all. UDP implies that error checking and correction are either not needed or must be performed in an application. And although time-sensitive applications often use UDP, since it is more preferable to drop some packets than to wait for delayed packets, in real-time systems this is not allowable. If it is necessary to correct errors at the network interface level, the application can use TCP or SCTP designed for this purpose.

MPEG-DASH (HLS) (Dynamic Adaptive Streaming over HTTP) is an adaptive streaming technology that provides the ability to deliver streaming multimedia content via an IP-based network by the HTTP protocol. It is the first decision on streaming data transmission with an adaptive bit rate, which received the status of an international standard. The technology involves splitting content into a sequence of short segments, each of which contains a small excerpt of the content. The content itself can be created in several bit-rate options, and alternative segments aligned on the same timeline become available to the DASH client. As the playback goes on, the client selects the next segment to download and play from the available alternatives automatically, according to the network operation conditions. The client selects the segment with the highest bit rate, which can be downloaded and played back on time, without lagging and buffering. As it plays, the client automatically selects the next segment to download and play from the available alternatives, based on the network conditions. The client selects the segment with the highest bit rate, which is possible to download and play on time, without lagging and buffering.

The specification provides a special format for describing the media stream, MPD (Media Presentation description) which contains information about the segments (timeline, URL, media characteristics such as video resolution and bit rate). Segments can contain any media data, but, in detail, the specification describes two types of containers: an ISO media file (for example, an MP4 file format) and an MPEG-2 Transport Stream.

The technology does not depend on the used audio- or video-codec. As a rule, one or several representations of multimedia files are available (for example, with different resolution or bit rate), and the choice can be made based on the state of the data network, device capabilities or user preferences, thus creating conditions for streaming with adaptive bit rate and best quality. DASH is also independent of the application layer protocols, so the technology can be used on top of any protocol, e.g. such as CCN.

The Real-Time Streaming Protocol (RTSP) – is the application protocol which is designed for use in systems which deal with multimedia data and allows you to remotely control the flow of data from the server, providing the ability to execute such commands as start , pause and stop of playback of multimedia content, as well as access to files located on the server by time.

The RTSP protocol does not perform compression, nor does it determine an encapsulation method for multimedia data. Streaming data transmission is not in itself a part of the RTSP protocol.

As a transport protocol is absent, by not relying on UDP or RTP for transporting, for providing the content as a stream of one-address data it is possible to use the Real-Time Protocol (RTSP). This is an application level protocol that was created specifically to control the transfer of real-time data, such as audio and video content. It is implemented according to the correction-oriented transport protocol. It supports player control actions such as stopping, pausing, and expedited forwarding in indexed Windows Media files. You can use the RTSP protocol to stream content to computers running Windows Media Player 9 or later or Windows Media Services 9 or later.

RTSP is a control protocol that works with the RTP data delivery protocol to provide content to clients.

Windows Media Services implements RTSP via the WMSRTSP server control protocol plug-in. With a standard installation of Windows Media Services, this plug-in is enabled and connected to TCP port 554 [1].

The protocols for streaming data have different implementation principles, strategies for ensuring quality of services and images, as well as methods for transmitting data over the network. It is precisely because of these differences that doubts arise as to which particular protocol is needed and advantageously used for transmitting streaming information.

This study considers a network for testing IP-telephony (video conferencing) with the aim to estimate the data packet losses, which uses various real-time streaming protocols, but under the same conditions. The obtained data are displayed graphically and recorded in the table of characteristics. On the analysis of the obtained data, the practical recommendations as to the use of the considered protocols are proposed, which is typical for studies of this type [2].

2 Design of Experiments

It is assumed that the network consists of the following components (Fig. 1):

- Multimedia server, which houses the video file that simulates video conferencing in real time;
- Ethernet Commutator with the function of setting bandwidth and transmission losses;
- a client, which uses a network analyzer to capture traffic and a video player that uses various real-time protocols to receive multimedia information from the server.

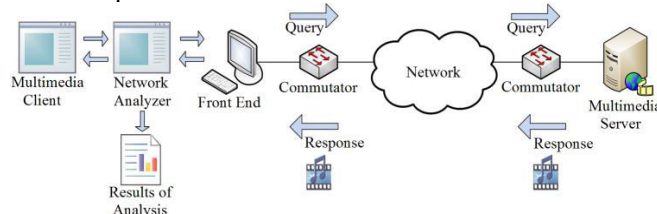


Fig. 1. Simplified presentation of the network [3].

At the multimedia server a video file is disposed which consists of 4 scenes with a resolution of 768×432 pixels, a refresh rate of 25 frames/s, a total stream rate of 1000 Kbps (1 Mbit/s) and duration of 40 s. On the client side, the video file is displayed in the player, while all traffic between the server and the client is captured into the network analyzer buffer. The frame rate in the player was also monitored, which varied depending on the protocol used and various network restrictions. Based on the data obtained, recommendations on the use of the protocols are given.

At first, for registration of changes in frames, a 8% network loss limit was used. Then, a speed limit of 900 Kbps was used. After that, depending on the change in

speed from 2 Mbit/s to 800 Kbit/s, the level of losses from 0% to 10% was used; the obtained results are displayed graphically below.

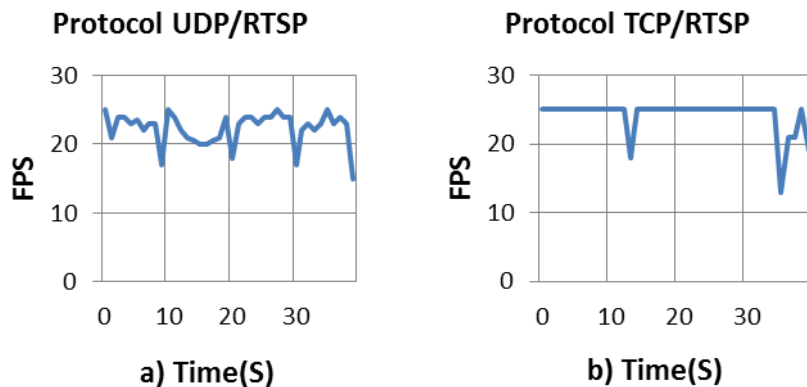
3 The Results of the Experiment

3.1 Network Performance Parameters under Unchanged Conditions

Monitoring of the frame rate during streaming well reflects the playback progress and identifies changes in stream quality that are perceived by a user. In a network in which a packet loss may take place, or which has an insufficient bandwidth, the player can begin to correct the playback speed, and even to reduce it. In such cases, the player can display all the frames included in the video, which increases playback time. The player can also discard image frames; this is perceived by the user as periodic breaks. The breaks may be barely noticeable, or they may be longer periods of freezing frames. They are often caused by a decoder strategy that cancels a damaged video frame and repeats the previous frame until the next valid decoded frame is available. All these player reactions cause changes in the frame rate.

Since the network conditions in the described experiment were simplified, the player basically responded to frame distortions using TCP-based streaming protocols. In most cases, the displayed frames were error free. A user observes the frame errors (reduction in the number of reproduced frames) as periodic breaks or jerks of the image. With a 8%packet loss ratio, the player reproduces video with each streaming protocol tested in the test environment (Fig. 2).

Fig. 2 shows examples of frame rate drops when packet loss is set to 8% and, thus, indicates differences in the interaction of protocols with the FFMPEG player. The first three diagrams (UDP/RTSP, TCP/RTSP, and HLS) well illustrate the distortions. With RTMP protocol, the playback stops once, and the video freezes for a few seconds. The rest of the sequence is played smoothly. However, in playback with UDP / RTSP, the displayed frames were also strongly distorted [2]. The bandwidth limit, which made visible changes to streaming playback, in the test bench was 900 Kbps for each tested protocol.



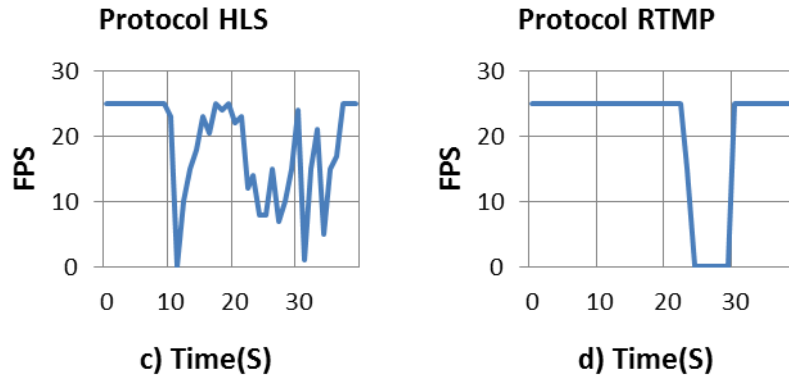
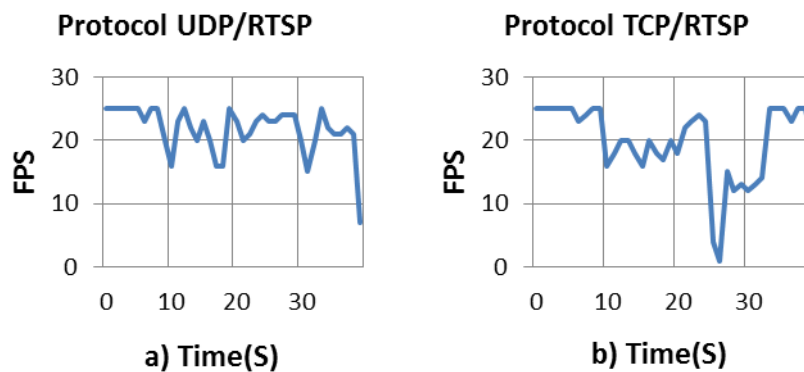


Fig. 2. Falling of frame rate at 8% loss of packets without limiting the rate for protocols UDP/RTSP (a), TCP/RTSP (b), HLS (c), RTMP (d).

Fig. 3 shows examples of frame rates with a 900 Kbps bandwidth. This shows how the RTMP protocol manages to maintain the maximal frame rate. When using the HLS protocol, the playback time increases.

This indicates that the player uses either very short periods of deferral or reduces the playback speed to cope with insufficient bandwidth. In these examples, the RTMP protocol displays most frames: 968 out of 1000. HLS protocol displays 953 frames, RTSP protocol - 873 frames over UDP and 785 frames - over TCP.

Unevenness in the reproduction often occurred near the change of scenes. In addition, the observed quality may vary significantly between scenes. Fig. 4 shows the differences in the scenes during playback. The data were collected from the same test measurements as in Fig. 3. The points associated with the dotted line show the time spent on each scene, compared to the actual duration (40 s), i.e. values greater than 100% indicate that the video was stretched due to a decrease in playback speed or correction of damaged frames due to packet loss.



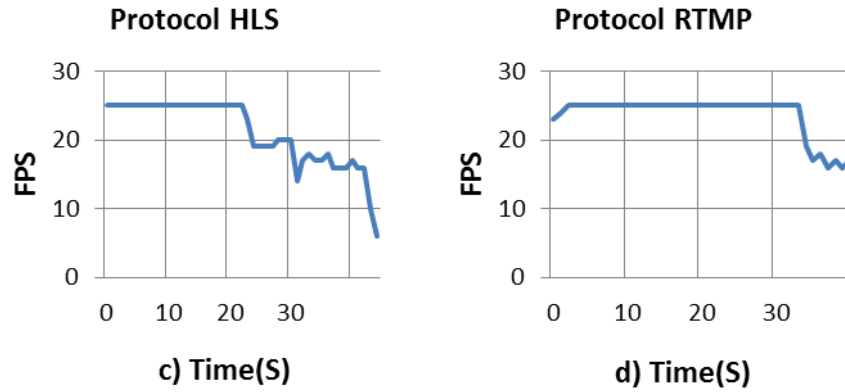


Fig. 3. Frame rate with a rate of 900 Kbps is lossless for protocols: UDP / RTSP (a), TCP / RTSP (b), HLS (c), RTMP (d).

The points associated with the solid line correspond to the displayed frames, i.e. a drop below 100% means that instead of 25 frames/s, fewer frames were displayed due to network losses.

If the streaming conditions were sufficient, both values would remain at 100%. In the RTSP / TCP stream, the third concatenated clip loses most frames. The streaming gets better in the final scene, allowing you to play almost all the frames, but the increased time spent on it indicates an uneven playback. Listening to the 40-second test sequence took 44 seconds with HLS. With RTMP, the first three scenes were played perfectly [3-5].

From the obtained results, it can be assumed that the number of displayed frames for each protocol directly depends on the introduced network restrictions presented in the following expression

$$F(t) = V \cdot L \cdot P, \quad (1),$$

where $F(t)$ – is the frame rate (fps); V – data transfer rate; L – is the share of lost packets (Packet Loss Rate, omitted in the formula when it is 0%); P – coefficient for the studied protocols (calculated for each protocol): P for UDP/RTSP = 0.00025; P for TCP/RTSP = 0.00027; P for HLS = 0.00022; P for RTMP = 0.00024.

Exception: at $L = 0\%$. L is omitted and the coefficient K is entered, which has its own value for each protocol: K for UDP/RTSP = 0.1; K for TCP/RTSP = 0.082; K for HLS = 0.107; K for RTMP = 0.11.

In this case the formula takes the form

$$F(t) = V \cdot K \cdot P. \quad (2)$$

Based on the analysis of the results obtained, the Tab. 1 is compiled. According to the data of this table, the final diagram was constructed (Fig. 5).

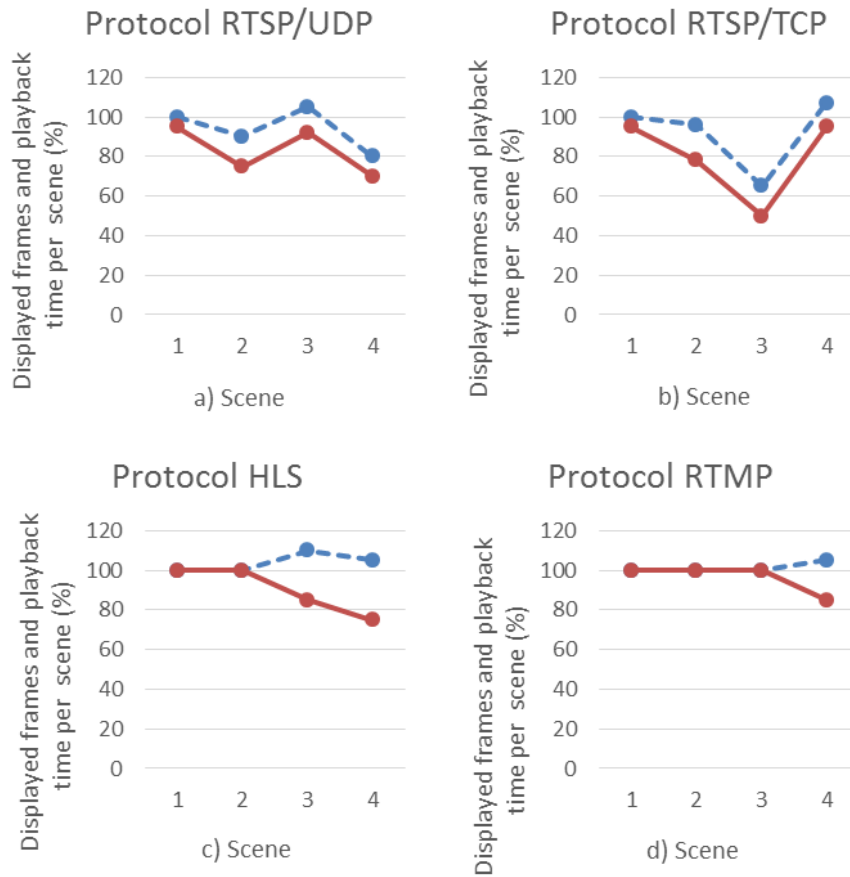


Fig. 4. Differences in scenes during playback at a rate of 900 Kbps without loss for protocols: UDP / RTSP (a), TCP / RTSP (b), HLS (c), RTMP (d)

Table 1. Results of a comparative evaluation of video transmission protocols.

Network restrictions	Protocol UDP/RTSP	Protocol TCP/RTSP	Protocol HLS	Protocol RTMP
V = const (1,1 Mbps) L = 8%	22	24	19	21
V = 900 kbps L = const(0%)	22	20	21	24

With a duration of observation interval of 40 s and a frame rate of 25 fps, the total number of frames is 1000. For a confidence probability of $\beta = 0.99$ and the number of experiments $n = 1000$, the half-width of the confidence interval makes $p = 0.005$. In other words, with a given number of experiments, the obtained value of the frequency

of reproduced frames deviates from the real by no more than ± 0.005 with a probability of 0.99 [3, 6]. To improve the quality of the results, it is planned to use specialized intelligent systems of frame preprocessing [7-11] and dynamic correction of the transfer process [12].

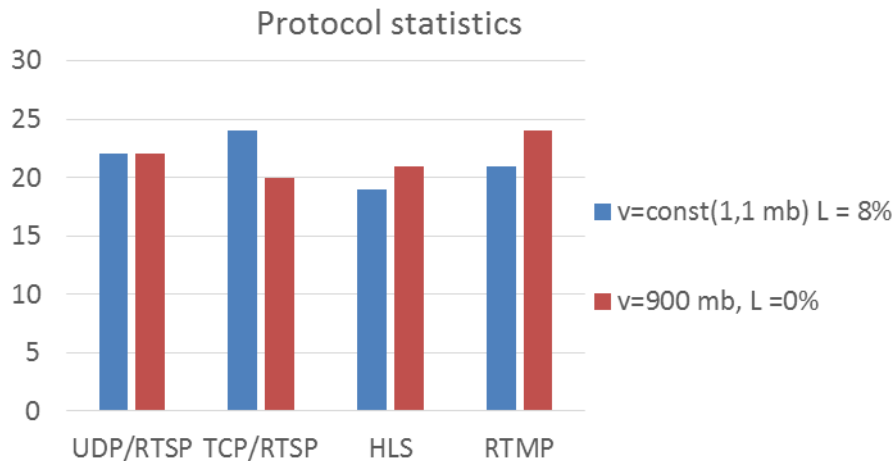


Fig. 5. The ratio of the number of reproduced frames to packet loss.

3.2 Losses at Changeable Network Operation Conditions

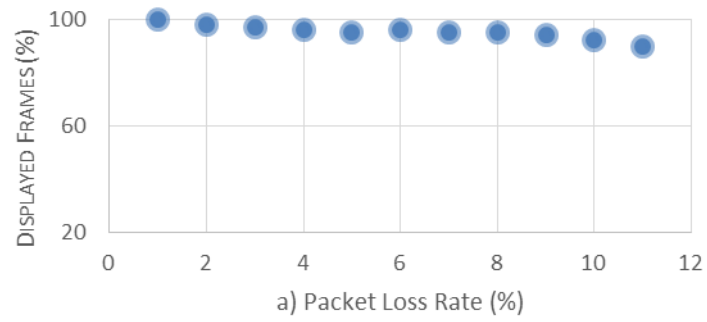
3.2.1 Losses at a Changeable Share (%) of Packet Losses

Fig. 6 shows the number of displayed frames when packet loss was increased from 1 to 10% at a bandwidth of 2 Mbps. In Figs. 6 (a) – (d) the volume of video stream of the player is not limited.

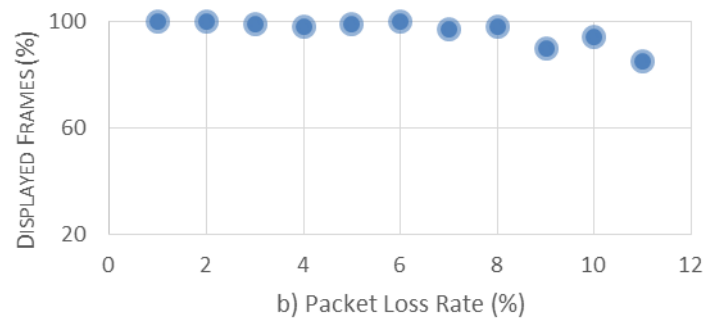
In the RTSP/TCP protocol, the first interference was observed when the loss was 4%. For HLS the limit was 8%, and for RTMP – 11%. For these settings, the RTMP was the only one for which the connection was not interrupted during the streaming, it was able to maintain flawless playback. According to Fig. 6 in the RTSP/UDP stream, the number of displayed image frames and packet losses are directly proportional. However, RTSP / UDP could not cope even with losses of 2% and displayed highly distorted video images. Obviously, this is caused by not providing a support for repeated sending of packets by the UDP protocol.

In Figs. 6 (e) – (h) the size of the video stream of the player is limited. Since the quality of the RTSP / UDP streaming was already poor, changing the queue size did not have a fundamental impact on the quality. HLS protocol cannot display all frames with 3% and RTSP / TCP with 4% ahead.

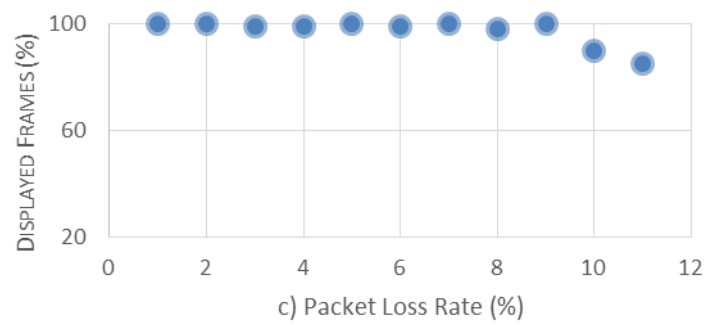
Protocol RTSP/UDP



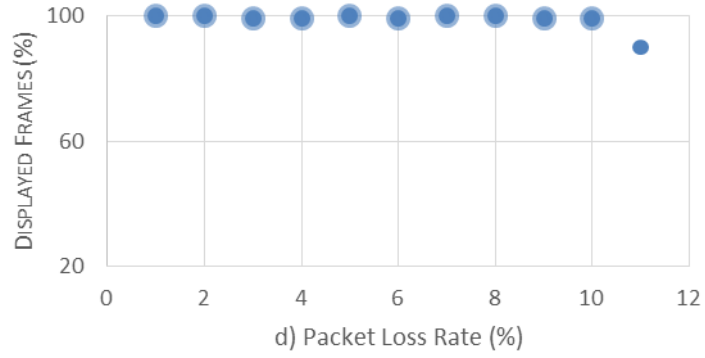
Protocol TCP/RTSP



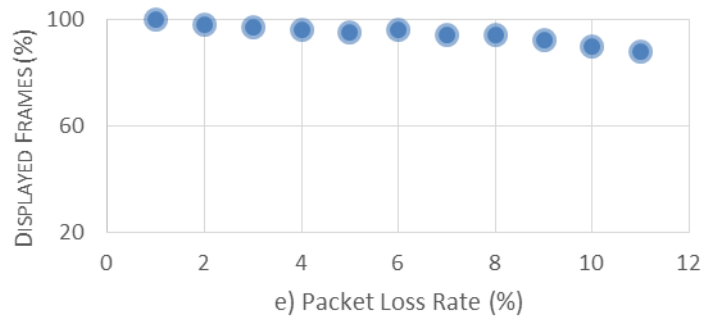
Protocol HLS



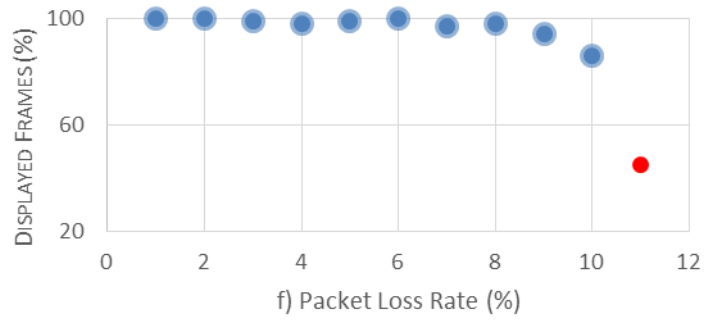
Protocol RTMP



Protocol RTSP/UDP



Protocol TCP/RTSP



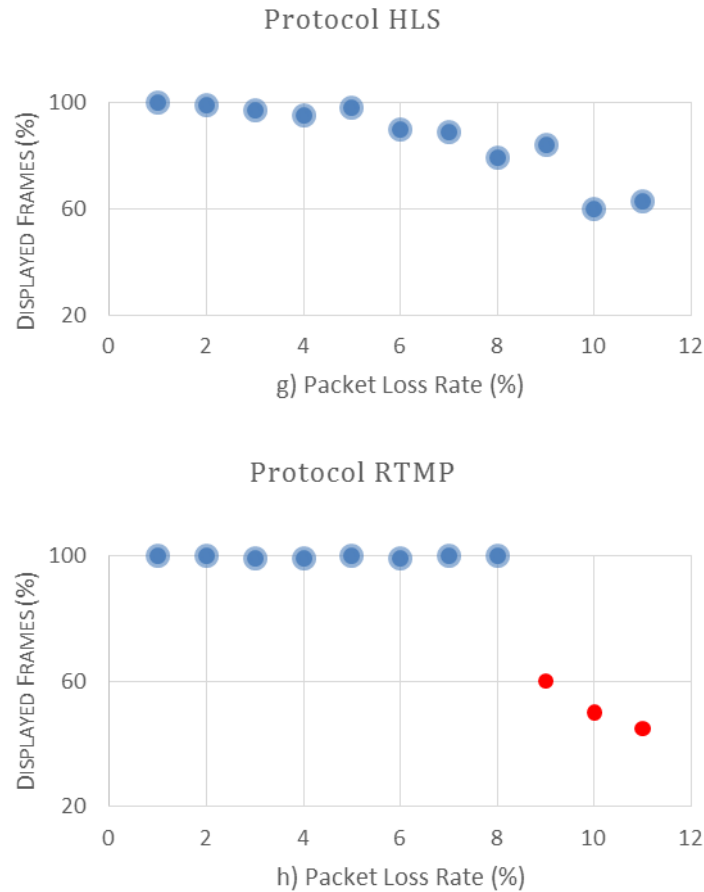


Fig. 6. The ratio of the number of reproduced frames to packet loss. On (a) - (d) the volume of video stream of the player is not restricted, whereas on (e) – (h) – it is limited.

RTMP again supported perfect quality up to 9%. HLS may lose more frames, but it manages to maintain the connection and reproduce the entire sequence with these settings. Since the type of packet lost has a great influence on the propagation of errors and maintaining the quality of streaming, there is also some variation in the displayed image frame metric.

3.2.2 Losses at a Changeable Rate Limit

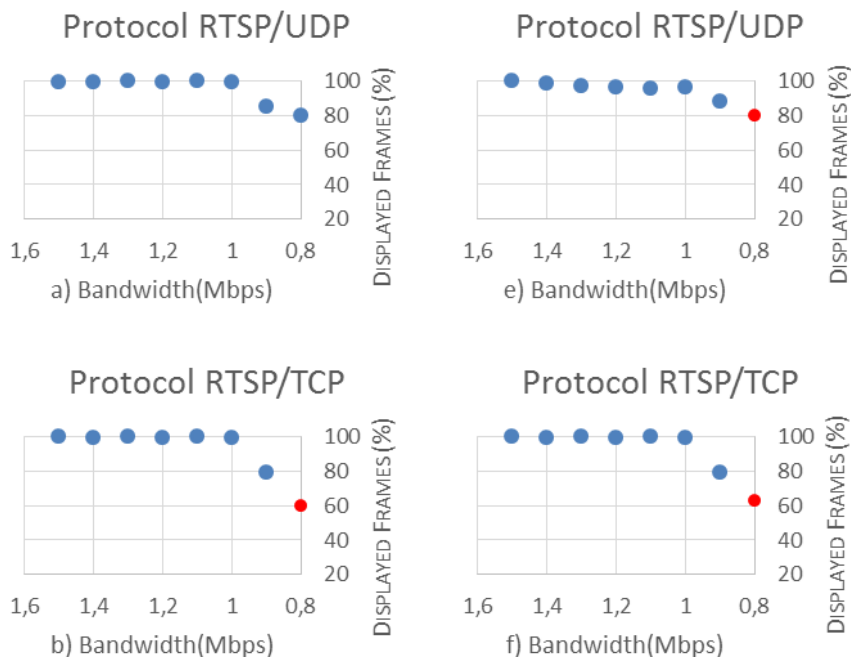
In order to investigate the response of the player under slow bandwidth conditions, a larger queue depth (of buffer) was used to minimize packet loss caused by the emulator. The bandwidth is limited by the interval of 1.5 to 0.7 Mbps, in descending order. Part of the displayed frames with both queue sizes of the video is shown in Fig. 7. In tests with limited bandwidth, the sequence was reproduced with the same parameters.

As in the case of packet loss, frame loss manifested itself as a jerk and short breaks. With RTSP / UDP, a video corruption occurred.

In fig. 7 (a) – (d) video queue volume is unlimited. The streaming was hopeless with all protocols up to 1 Mbit/s bandwidth, which was expected, since the average bit rate of the test sequence was almost the same. Below 1.1 Mbit/s, the HLS protocol began to reduce the number of frames, causing gaps (jerks) of the video. During the experiment, the RTMP protocol lost the connection to the server. The RTSP stream responded by reducing the frame rate more radically, and also lost the connection several times. When the RTP packets were transmitted over UDP, distorted images were also observed at a throughput of 900 Kbps.

In Fig. 7 (e) – (h) the video queue size was limited. With the RTSP / UDP protocol, the player showed highly distorted video images already at a rate of 1.2 Mbps. As the bandwidth decreased to around 1 Mbit / s, the video image was displaying perfectly, with minor jerks. Improving the quality of transients may indicate that at a higher bandwidth, the queue (buffer) of the video player is filled and discards the surplus packets.

When the bandwidth was limited to 800 Kbps, the videos pertaining to the last 10 seconds were distorted. With RTSP / TCPR, resizing the video queue did not significantly affect the performance of the player. In HLS streaming, video twitching (gaps) began with greater bandwidth than with an unlimited queue size (buffer). Similarly, with RTMP streaming, the connection was lost already at 900 Kbps bandwidth [2].



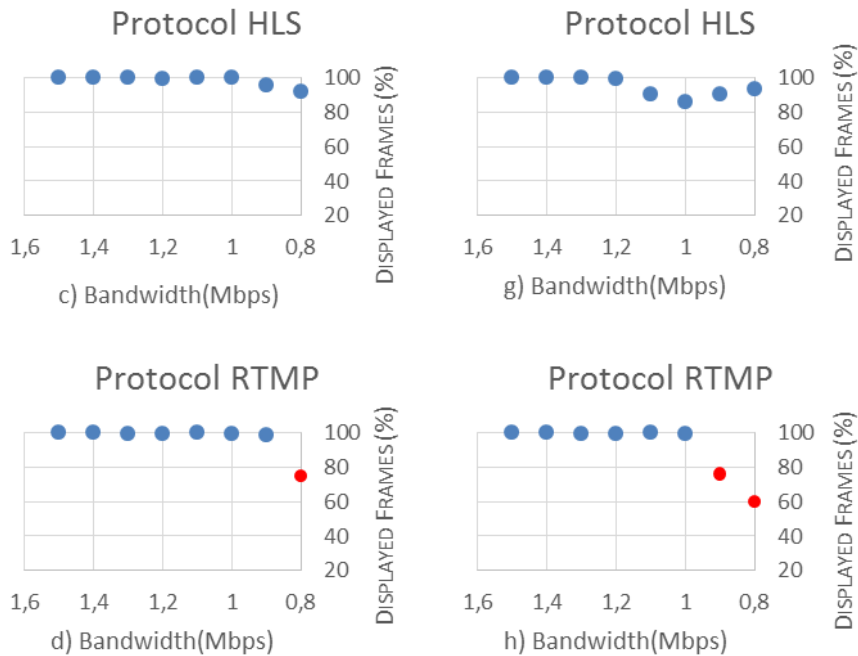


Fig. 7. The change in the proportion of reproduced frames at restricting the rate. On (a) – (d) the volume of the video queue is not limited, whereas on (e) – (h) it was limited.

4 Conclusions

This study compared various real-time data streaming protocols in a computer environment for which packet loss and bandwidth limiting were determined. For these parameters, the boundaries were searched, after which the quality of the streaming begins to deteriorate. According to the simulation results, there are clear differences in the quality of the TCP and UDP streaming protocols. First of all, this is caused by the lack of application layer retransmission facilities in UDP. This means that each lost packet causes gaps (jerks) in the video being played, which are visible to the user, and the number of lost packets can work as an indicator of the quality of the protocol operation. To improve the situation, a re-send mechanism is typically used; for this, for example, a retransmission of lost packets can be used.

In addition to the lack of support for resubmission in the UDP protocol, the player also defines the different sizes of the video data queue in the RTSP as compared to the HLS and RTMP as default values. This was taken into account when modeling. Accordingly, this demonstrates that players compatible with multiple streaming protocols may have different quality indicators.

For protocols based on TCP, the frame rate, the ratio of lost / displayed frames and the monitoring of the playback duration are reflected on the quality of streaming playback, including video slowdown (the appearance of gaps in frames or an increase

in the duration of the video due to loss). These indicators can be measured at the application level, and changes in them can be considered noticeable to the user. In particular, changes in frame rates reveal the characteristics of streaming protocols. The ratio of lost / displayed frames shows changes in the frame rate and / or playback duration. Thus, the number of lost / displayed image frames is a metric that generalizes information and is more suitable for assessing the performance of protocols.

In the tests performed, two protocols based on stretching were used. The HLS protocol was the only one that has not lost the connection when network conditions deteriorated. On the other hand, when the size of the video stream (buffer) of the player was reduced, streaming via the HLS protocol was the first to show changes in quality in terms of both loss and bandwidth limitation. This may indicate that this protocol consumes more bandwidth than other TCP-based protocols. The RTMP protocol was retaining the quality somewhat longer than the others. The HLS protocol was the most stable, without breaking the connection during the tests. The RTSP/TCP protocol can cause a greater change in quality in the presence of packet loss. On the other hand, when the bandwidth was limited, reducing the queue size of the player had no effect on the quality. The RTSP / UDP protocol turned out to be the worst because of the lack of mechanism of repeated sending in the UDP protocol.

References

1. MSDN Microsoft, <https://msdn.microsoft.com/en-us/library/cc239484.aspx>.
2. H.264 QoS and Application Performance with Different Streaming Protocols / Laine, S., & Hakala, I. // 2015 EAI Endorsed Transactions on Future Intelligent Educational Environments, 2015 (3), e3. doi:10.4108/icst.mobimedia.2015.259061
3. D. Sumtsov Development of a method for the experimental estimation of multimedia data flow rate in a computer network / Dmytro Sumtsov, Serhii Osievskyi, Valentyn Lebediev // Eastern-European Journal of Enterprise Technologies. – 2018. – Vol. 2, N 2 (92). – P. 56-64. – Way of Access: DOI: 10.15587/1729-4061.2018.128045
4. NbuV, http://nbuv.gov.ua/UJRN/suntz_2017_2_40.
5. S.P. Yevseiev, H.N. Rzayev, S.E. Ostapov, V.I. Nikolaenko Data Exchange Evaluation in global networks based on Integrated Quality Indicator of Service Network // Radio Electronics, Computer Science, Control. – 2017. – Vol. 1. – P. 115-128. doi: 10.15588/1607-3274-2017-1-14
6. H. Venttsel, L. Ovcharov Probability Theory and its Engineering Applications. Moscow: Vysshaya Shkola, 2000. – 480p.
7. I. Ruban, K. Smelyakov, V. Martovytskyi, D. Pribylnov and N. Bolohova Method of neural network recognition of ground-based air objects // IEEE 9th International Conference on Dependable Systems, Services and Technologies (DESSERT), 24-27 May 2018. – P. 589-592. DOI: 10.1109/DESSERT.2018.8409200
8. K. Smelyakov, D. Pribylnov, V. Martovytskyi, A. Chupryna Investigation of network infrastructure control parameters for effective intellectual analysis // IEEE 14th International Conference on Advanced Trends in Radioelectronics, Telecommunications and Computer Engineering (TCSET), 20-24 Feb. 2018. – P. 983-986. DOI: 10.1109/TCSET.2018.8336359

9. K. Smelyakov, A. Chupryna, D. Yeremenko, A. Sakhon, V. Polezhai Braille Character Recognition Based on Neural Networks // IEEE Second International Conference on Data Stream Mining & Processing (DSMP), 21-25 August 2018. – P. 509-513.
10. G. Churyumov, V. Tokarev, V. Tkachov and S. Partyka, "Scenario of Interaction of the Mobile Technical Objects in the Process of Transmission of Data Streams in Conditions of Impacting the Powerful Electromagnetic Field", 2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP), 2018. – DOI: 10.1109/DSMP.2018.8478539.
11. I.V. Ruban, G.I. Churyumov, V.V. Tokarev, V. M. Tkachov, "Provision of Survivability of Reconfigurable Mobile System on Exposure to High-Power Electromagnetic Radiation", Selected Papers of the XVII International Scientific and Practical Conference on Information Technologies and Security (ITS 2017), CEUR Workshop Processing, pp. 105-111, November 30, 2017.
12. S. Mashtalir, O. Mikhnova, M. Stolbovyi Sequence Matching for Content-Based Video Retrieval// IEEE Second International Conference on Data Stream Mining & Processing (DSMP), 21-25 August 2018. – P. 549-553.