International Journal of Computer Engineering and Data Science (ISSN: 2737-8543)
Volume 5–Issue 1, January -March 2026

TFRC-Based Selective Retransmission for Free-Viewpoint Video Streaming

Árpád Huszák[12]

[1]Budapest University of Technology and Economics, Department of Networked Systems and Services, Műegyetem rkp. 3, Budapest, 1111, Hungary.
[2]HUN-REN-BME, Cloud Applications Research Group, Piarista utca 4, Budapest, 1052, Hungary.
E-mail: huszak@hit.bme.hu

Article history

Received Feb 17, 2026
Revised Mar 07, 2026
Accepted Mar 30, 2026
Published Apr 14, 2026

ABSTRACT

Free-viewpoint video may become the next big step in media technology that allows users to change the displayed viewpoint and synthesize custom views of a dynamic scene from a user-controlled perspective. The user-specific views are generated from two or more color and depth camera stream pairs that must be successfully delivered to the clients according to their continuously changing perspectives. Besides the view synthesis distortions, the higher error rate and limited bandwidth can cause quality degradation. We proposed an adaptive retransmission-based error recovery scheme allowing retransmissions according to the network congestion state and the allowed latency. The selective retransmission (TbSR) scheme for free-viewpoint video is based on the TCP Friendly Rate Control utilized as a bandwidth estimation algorithm. The TbSR approach was evaluated using the NS-2 network simulator and VSRS reference view synthesis tool in order to prove the efficiency of the proposed method.

Keywords: *Free-Viewpoint Video, TFRC, Selective Retransmission, Streaming.*

I. Introduction

Free-viewpoint video (FVV) enables users to have interactive control over their viewpoint and viewing direction within a visual scene, similar to the experience in 3D computer graphics applications. It offers an immersive multimedia service where users can freely navigate and explore the scene from different angles, enhancing their engagement and sense of presence. Likewise, the clients of FVV interactive multimedia service are allowed to control the viewpoint and synthesize custom virtual views of a dynamic scene continuously. The main difference between model-based rendered 3D content and FVV is that it targets real-world scenes acquired by cameras. The uniquely synthesized views are generated from several color and depth camera pairs delivered as a stream to the viewpoint synthesis algorithm. By increasing the number of camera views used to capture the scene, the quality of the synthesized free-viewpoint video improves, while on the other hand, the delivery of more camera streams will increase the network traffic load, as well.

FVV streaming has significant differences compared to traditional video streaming. The increased bandwidth requirement caused by multiple camera streams per user is one of the most important dissimilarities. Forwarding only those camera views (at least two views) that are currently necessary for the custom viewpoint synthesis is an effective way to reduce the traffic in the network. Since the user perspective can change continuously, camera view switchovers will occur more frequently, which must be handled without interrupting the playout.

The increased network utilization can lead to congestion in the network, causing packet losses and high delivery latency. Conventional error concealment techniques typically rely on retransmissions to ensure reliability, but this comes at the cost of increased latency. For multimedia applications that are less sensitive to latency, such as on-demand and one-way real-time video services, retransmissions can be used. However, their success depends on the timely reception of the retransmitted packet before its intended playback. To minimize unnecessary retransmissions, a receiver-side playout buffer is commonly employed, which helps mitigate the effects of network variations. By prefetching data into the playout buffer, additional time is provided to ensure the successful completion of retransmissions while maintaining smooth playback.

Unfortunately, the substantial volume of traffic generated by multiple camera streams per user in a free-viewpoint video (FVV) service can overload network capacities and lead to congestion. In a congested network, both the round-trip time (RTT) and the experienced packet loss ratio significantly increase, thus increasing the possibility that the retransmitted packets will be lost once again. In such conditions, retransmission is not advisable. Therefore, it becomes important to assess which lost packets should be retransmitted in order to improve the quality of the video stream selectively. To effectively manage the retransmission process, a selective retransmission approach is required.

General video streaming applications rely on traditional transport protocols (UDP or TCP), but alternative protocols are also available, such as the Lightweight User Datagram Protocol (UDP-Lite) or the Datagram Congestion Control Protocol (DCCP). In our work, we used DCCP [1] because it offers the best features of TCP and UDP for multimedia transmission. Moreover, DCCP is the only one among the unreliable transport protocols that includes congestion control algorithm. Actually, DCCP can be considered as TCP minus bytestream semantics and reliability or as UDP plus congestion control, handshakes and acknowledgments. In this work, DCCP was applied as a transport protocol due to its advantages, such as sequence numbering, acknowledgments, and TFRC [2] congestion control algorithm support. To implement a selective retransmission approach, it is necessary to utilize sequence numbers and acknowledgments to identify the lost packets. Additionally, a congestion control algorithm is needed to estimate the available network capacity. These components play a crucial role in effectively identifying and retransmitting the necessary packets while considering the congestion level of the network.

This paper proposes a source-controlled TFRC-based selective retransmission scheme (TbSR) for FVV streaming services. Due to the features of the DCCP protocol [1] and the TFRC [2] bandwidth estimation, retransmission decisions can be performed on the server side, so the proposed TbSR approach does not need any additional administration messages. The solution aims to make the lost packet recovery possible without increasing the level of congestion. To the best of our knowledge, no retransmission-based concealment approach for FVV streaming was investigated and published earlier.

The rest of this paper is organized as follows. The overview of free-viewpoint video services and the review of related works in selective retransmission are presented in the next section. In Section 3, the proposed TFRC-based selective retransmission scheme for FVV streaming applications is introduced. The performance evaluation of our method and the obtained results follow in Section 4. Finally, we summarize and conclude the paper in the last section.

II. RELATED WORKS

Multimedia streaming necessitates high network capacity and low delay to provide adequate quality of streaming services. The distribution of ultra-high-resolution single-view video streams is challenging in a mobile environment, but it becomes even more complex for multi-view videos due to limited link capacity and packet losses. First, the characteristics of free-viewpoint video are overviewed, followed by the introduction of existing adaptive retransmission schemes.

A. *Overview of Free-Viewpoint Video*

Free-viewpoint video has been attracting increasing attention due to its promising application scenarios. Based on its advanced features, FVV is expected to be the next big step in 3D video technology. The intense research activity on FVV topic mainly focuses on encoding and viewpoint rendering, while network delivery and streaming techniques are investigated with lower intensity. Currently, the key technologies used in multi-view streaming methods are still in development and do not yet achieve optimal efficiency for providing services of satisfactory quality. This limitation primarily stems from the computationally intensive nature of viewpoint synthesis, making it challenging to strike the right balance between the quality of the synthesized video and the rendering time required by FVV algorithms.

In order to capture real-world scenery for FVV services, special acquisition systems with multiple cameras are required. Two types of approaches are used to generate the desired views: image-based view synthesis and depth image-based rendering (DIBR).

Image-based view synthesis in real-time has gained a lot of attention from the research community. The main concept of the synthesis method is to generate intermediate virtual views from color camera views by interpolation. The approach requires a dense camera set; otherwise, the interpolation and occlusion artifacts will distort the generated views. Therefore, a tremendous amount of image data needs to be processed to provide acceptable video quality. Different image-based rendering algorithms have been introduced by [3–5] that often show problems in terms of computation time and synthesized view quality.

In the case of DIBR method, the deployed camera systems consist of traditional color cameras with depth sensors along with the camera calibration information. Depth views are acquired simultaneously with a general color camera array and transformed to a monochromatic image taking values in the range of 0 and 255, as illustrated in Figure 1, with its corresponding color image. Fortunately, several manufacturers offer high-performance depth cameras, such as Azure Kinect DK, Intel RealSense or Luxonis OAK-D [6].

Figure 1: Text and features extraction process.

The main idea of the depth image-based rendering (DIBR) algorithms introduced in [7] is to perform 3D warping to the virtual viewpoint based on the texture and depth information of the reference cameras and then projected back onto an arbitrary virtual camera plane, creating a virtual image. According to [8], the process has two main steps: (*i.*) 3D image warping and (*ii.*) reconstruction and resampling. The image warping process involves utilizing depth information and corresponding camera parameters to project pixel samples from reference images onto appropriate 3D coordinates, which are then re-projected onto the newly synthesized image space. The computation of pixel values in the synthesized image occurs in the second phase. It is important to note that the Depth-Image-Based Rendering (DIBR) approach is highly sensitive to the quality of the captured color and depth camera streams. The accuracy of the depth sensor images significantly impacts the quality of the synthesized virtual view, as extensively investigated in [9, 10].

The 3D Video Coding Team of the ISO/IEC Moving Pictures Experts Group (MPEG) provides the Depth Estimation Reference Software (DERS) and the View Synthesis Reference Software (VSRS) tools, which are widely employed in research studies for implementing depth and view synthesis algorithms.

Efficient multi-view video compression is essential from FVV storage and network delivery point of view. The encoding scheme is known as Multiview Video Coding (MVC) [11, 12], utilizes the similarity of two or more camera streams showing the same scenery from different perspectives. MVC was introduced as an extension of the H.264/AVC that exploits both inter-view and temporal redundancies for efficient compression keeping the resolution of the views. In order to handle depth images besides general camera views, multi-view video-plus-depth (MVD) coding scheme is used introduced in [9].

The FVV service architecture is categorized according to the location of the custom viewpoint synthesis process within the network. Typically, three approaches can be distinguished: server-based, client-based, and distributed model.

In case of the server-based model, the camera views and depth map sequences are streamed to the central processing server, while the requested viewpoint coordinates are received from the customers. Based on the desired user perspectives, the server syntheses the unique virtual viewpoint stream for each user. The disadvantage of this centralized approach is that the computational capacity of the central server may limit the number of users. On the other hand, the network traffic between the server and the client is remaining similar to traditional video-on-demand streaming services because the color and depth camera streams are traveling just between the cameras and the server but not to the clients.

The client-based approach solves the scalability problems of computational capacity because the color camera views and the depth sequences are streamed to the clients to synthesize their own virtual views independently. Unfortunately, in this case, the generated network traffic can easily overload the network. Advanced streaming solutions, such as multicast delivery or selective camera view delivery can decrease the probability of network congestion. If not all the camera sequences and corresponding depth images are delivered to the users but only the currently required ones, the overall traffic load can be decreased. However, the requested camera streams are changing continuously depending on the current viewpoint, so adaptive camera view streaming methods are required. We proposed such solutions in our previous work [13], where the camera selection was handled using threshold areas to prefetch the camera view sequences that will probably be required for the view synthesis.

The third model is a distributed approach that uses proxy servers for view synthesis. Instead of direct connection to the media server, the client requests the generation of the stream with the desired viewpoint from the most appropriate proxy server. In case of the remote rendering approach, bandwidth, and computation problems can be solved with a simple but effective solution because all virtual views are synthesized remotely on a powerful server at the price of increased latency investigated in [13, 14]. Fortunately, edge cloud architecture can accelerate the process, e.g., [15] introduced a live FVV system architecture, including the acquisition and encoding of multi-view plus depth data in several capture servers and virtual view synthesis on an edge server.

B. *Frame Loss Concealment for Free-Viewpoint Video*

Packet loss during the transmission of camera and depth video streams will result in distortion of synthesized virtual views. The rendered video quality degradation due to frame losses may be worse compared to that of conventional video applications.

A number of concealment algorithms were introduced to recover the lost frames, prevent error propagation, and improve the quality of the reconstructed streamed 3D video. A method was introduced by [16] based on the inter-view similarity of the multi-camera views and intensity differences, while in [17], a frame loss concealment method was proposed for multi-view video plus depth (MVD) based on motion information sharing and backward warping. Most of the existing concealment uses spatial or temporal similarity to reproduce the missing frames; however, retransmission of lost packets can be even more effective, because the original content can be used in the decoding process. However, retransmission-based solutions are acceptable only for non-interactive services, where the latency requirements are not so strict.

A prevention technique was introduced for congested networks in [18] based on rate control. The authors proposed a rate adaptation method based on controlled packet dropping. The method extracts the more important packets of the MVD video stream to match the currently available bandwidth. By proactively dropping the less important 3D layers and using error concealment strategies for MVD streaming, higher QoE can be provided.

In a congested network, packet losses and increased delay can disturb the user experience. To decrease the traffic load or support more multi-view videos in IP networks, a simple approach is to reduce the bandwidth consumption by providing only the lowest number of camera views requested for the multi-view video streaming service, as it was investigated in [13, 14, 19].

C. Selective Retransmission Methods

Many papers propose retransmission for video transmission, but to the best of our knowledge, retransmission has not yet been investigated for MVD streaming. Similarly, different selective retransmission approaches were analyzed for conventional video streaming and categorized into content-based and network-characteristic-based methods.

Content-based retransmission methods differentiate the streamed data content based on its importance and retransmit only the prioritized data types of the stream. E.g., it makes the priority order of the video stream packets based on the MPEG frame type delivered in the packet. Correcting errors in keyframes due to earlier packet losses prevents error propagation to other frames in the Group of Pictures range.

Feamster and Balakrishnan [20] designed an RTP extension that is able to selectively retransmission important data in conjunction with receiver postprocessing. The approach also uses a TFRC-based congestion manager that provides a more amenable sending bitrate for video streaming services. The authors proved that the recovery of the most important data in the video stream significantly improves the performance without much additional penalty in terms of latency.

Attempts were made to implement a selective multi-path retransmission control model for point-to-point conversational video services introduced in [21, 22]. Both approaches use a combination of push-based and pull-based mechanisms for fast lost detection. Multiple available paths enable retransmission of the dropped packets through an alternative path with the lowest end-to-end delay, avoiding continuous packet loss on the original path.

L. Han et al. [23] proposed an adaptive retransmission scheme that adjusts the retransmission window to achieve a better loss recovery rate. In their approach, the client controls a retransmission window, which is equal to the number of allowed pending retransmission packets. According to TCP congestion control mechanism behavior, the retransmission window is set to half when congestion is detected.

The authors of [24] implemented a selective retransmission protocol with a decision algorithm. The control mechanism of this approach decides whether or not to request retransmission for a lost packet using a decision rule based on the Euclidean distance calculated by the loss and latency ratio.

III. TFRC-BASED SELECTIVE RETRANSMISSION SCHEME FOR FVV STREAMING

Packet losses and the increased latency in a congested network can significantly disturb the user experience, hence we aim to reduce the traffic load if necessary while maximizing the service quality. To support the delivery of more multi-view videos in the network, a general approach is to minimize the bandwidth consumption by transmitting only the lowest number of camera views required, as it was investigated in [13, 14].

Depending on the type of multimedia service, different transport protocols are used. For VoD and real-time one-way streaming services, the latency requirements are not so strict, therefore TCP can also be used, which provides reliable packet delivery based on retransmissions. Although frequent retransmissions may lead to severe network congestion. While in the case of real-time interactive services, UDP is the standard protocol because it transmits the packets as they are generated, providing as low delivery latency as possible without retransmission support.

In this paper, we propose a midway solution that partially retransmits the lost packet if the state of the connection makes it possible. In order to make the retransmission successful, two requirements must be fulfilled. First, the lost packets must be identified; second, the retransmitted packet must be received in time before the playback without being dropped again due to congestion. In a congested network, the RTT (round-trip time) increases so much that the retransmitted packet will likely not arrive before the playout. Therefore, retransmission will not improve service quality; moreover, it will increase network load and delay.

To ensure a successful retransmission, the transmitter must know the playout buffer level and the current network delivery delay. There are two approaches that can give information about the client-side buffer state. The simpler solution is to use administrative messages, periodically providing the buffer level to the transmitter. The other solution is more complex but can work without generating administrative messages. The main idea behind this approach is that at the beginning of the video transmission a certain amount of data is being heaped up in the receiver's playout buffer in a controlled way, as was published in our previous work [25]. The proposed approach initially builds up a certain buffer level at the receiver's end and henceforth tries to maintain this buffer level as precisely as possible by constant-rate transmission. The playout buffer level should be high enough to detect the packet losses and retransmit the lost packets, but it should be as short as possible, not to reduce the user experience. If the bitrates of all camera streams and depth views are constant, the buffer delay will not vary significantly. We assume that a proper playout buffer is set up, allowing the retransmitted packets to be delivered in time.

The transport protocol that fulfills all requirements for congestion-based selective retransmission is the Datagram Congestion Control Protocol (DCCP) [1]. It does not retransmit the lost packets by default but uses sequence numbers to identify the datagrams. Also, it continuously measures the available bandwidth and RTT for its built-in TCP-Friendly Rate Control (TFRC). This information is accessible by socket system calls.

The proposed TbSR approach utilizes the TFRC congestion avoidance algorithm to decide whether a lost packet should be retransmitted. To estimate the network congestion level, the proposed sending rate calculated by TFRC equation-based congestion control protocol is used. According to RFC 5348, the sending rate (available bandwidth, T_t) is calculated as a function of packet size (s), smoothed RTT (R), and packet loss ratio (p).

$$T_t = \frac{s}{R_t\sqrt{\frac{2bp}{3}} + t_{RTO}\left(3\sqrt{\frac{3bp}{8}}\right)p\left(1+32p^2\right)}, \tag{1}$$

where t_{RTO} is the retransmission timeout value in seconds and b is the maximum number of packets acknowledged by a single acknowledgment.

Instead of the current RTT value (R_t), the TFRC algorithm uses smoothing in order to avoid a high variance of the value according to the following equation

$$R_t = qR_{t-1} + (1-q)\,RTT, \tag{2}$$

where the q smoothing factor is 0.9 by default.

If the actual video bitrate (λ_t) is known, the proposed TbSR method can decide whether there is enough free network capacity to allow retransmissions or not.

Of course, if the video bitrate is significantly higher than the estimated available network bandwidth, the selective retransmission scheme can not improve the stream quality. However, if the network is close to a congested state, the proposed approach can conceal the effect of packet loss.

The retransmission is also limited by the playout buffer delay (β) installed at the receiver and the loss detection delay (τ). According to the DCCP protocol specification, the receiver sends DCCP-Ack packets at least once per RTT to acknowledge data packets, unless the sender sends fewer than one packet per RTT, as described in the TFRC specification [RFC 5348]. According to the specification, the loss detection delay is

$$\frac{1}{f_p} < \tau < RTT, \tag{3}$$

where f_p is the frequency of packet sending. Hence, the elapsed time between two packets is $1/f_p$. The available time for the first transmission, loss detection, and retransmission is equal to the playout buffer delay (T_{bd}).

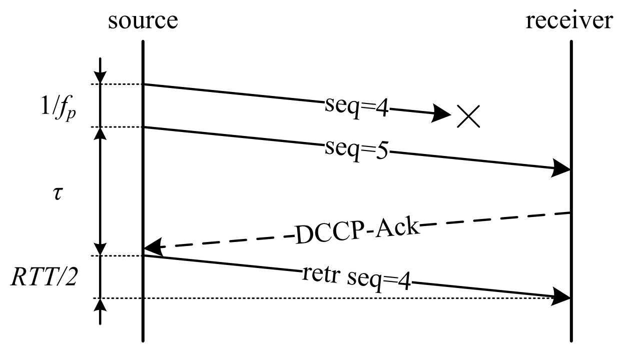

Figure 2: Data delivery sequence

Figure 2 illustrates the packet delivery sequence with loss detection and retransmission. According to the loss detection retransmission timing, the retransmission is enabled if

$$\frac{1}{f_p} + \tau + \frac{1}{2}RTT < T_{bd}. \tag{4}$$

By replacing the loss detection delay (τ) with its upper bound according to equation (3), the upper bound of the time (ρ) for the first transmission, loss detection, and retransmission is

$$\rho = \frac{1}{f_p} + \frac{3}{2}RTT \tag{5}$$

Successful retransmission can be performed if

$$\rho < T_{bd} \tag{6}$$

Overall, the decision is made based on the following rule:

Algorithm 1 Decision-making process

while not end of stream **do**
  calculate T_t
  calculate ρ
  if $\lambda < T_t$ **and** $\rho < T_{bd}$ **then**
    retransmission allowed
  else
    retransmission denied
  end if
end while

In the case of free-viewpoint video streaming, two camera views are required to synthesize the user-specified perspective. By minimizing the distortions due to packet losses of the delivered left and right camera streams, the quality of the synthesized view will also improve.

IV. SIMULATION RESULTS

In order to evaluate the effectiveness of the proposed TbSR algorithm, we simulated the DCCP-based multi-view video-plus-depth (MVD) streaming in a simplified network using the NS-2 network simulator tool. We chose NS-2 because it has DCCP implementation included. The topology of the analyzed MVD streaming service is presented in Figure 3, where all link capacities were set to 100 Mbps.

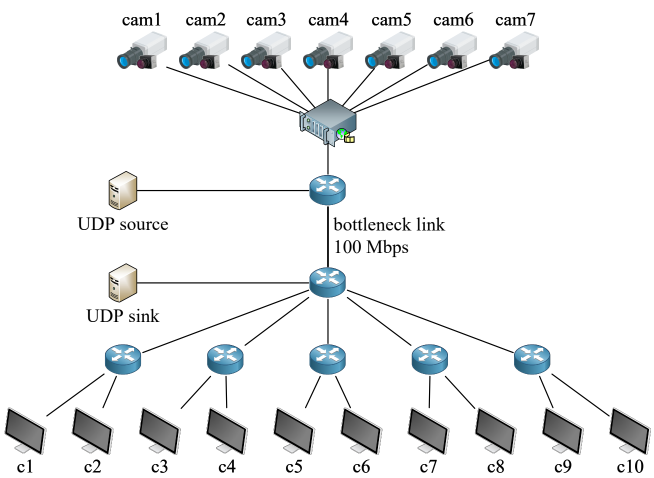

Figure 3: Simulated network topology

We used a bottleneck topology to control the packet loss rate, traffic load, and congestion level in the network. The background traffic on the bottleneck link was generated by a UDP source with adjustable bitrate. We assumed seven color cameras plus depth from which each client can choose the left and right camera sources for the view synthesis required for the desired view perspective. In the analyzed scenario, ten clients were deployed. The seven color camera and corresponding depth views were replaced by the *Balloons* sequence created for test purposes provided by Fujii Laboratory at Nagoya University nagoya. The 1024×768 resolution YUV sequences were encoded to MPEG-4 using the FFMPEG codec. The encoded color camera streams bitrate was 1.0-1.5 Mbps, while the black and white depth camera streams were 0.5-1.0 Mbps. The encoded streams were packetized and delivered to the clients using the DCCP protocol.

Each client received two color and two depth camera sequences, according to their desired viewpoint. The next step was to decode and synthesize the desired view from the received streams using the View Synthesis Reference Software (VSRS) developed by tanimoto2008reference. By adjusting the constant bitrate UDP background traffic, some packets can be lost due to congestion. Therefore, the distortion in the color or depth camera streams will propagate to the synthesized view. Our aim was to analyze how the packet losses decrease the synthesized view quality and how the proposed selective retransmission scheme can improve the QoS. The overview of the process is illustrated in Figure 4.

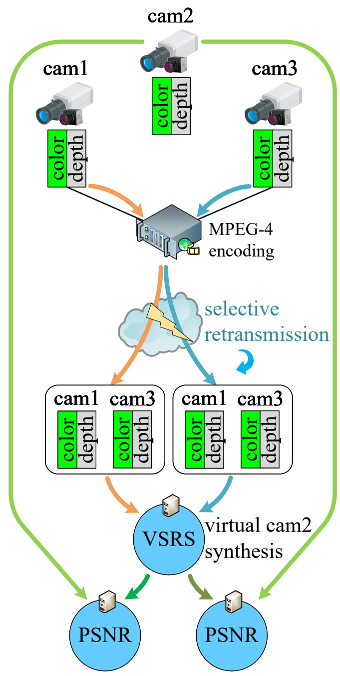

Figure 4: MVD streaming evaluation workflow

In the first scenario, the packet losses were generated by increasing the UDP background traffic on the bottleneck link and causing congestion. The sending rate of the constant bitrate UDP source was set between 10 Mbps and 50 Mbps. As each client received two color and two depth camera streams that all together required approximately 4.0-5.0 Mbps connection bandwidth, the requested network capacity for the ten clients was 40-50 Mbps. We analyzed the measured packet loss rate due to congestion and the ratio of retransmitted packets. The obtained results are introduced in Figure 5.

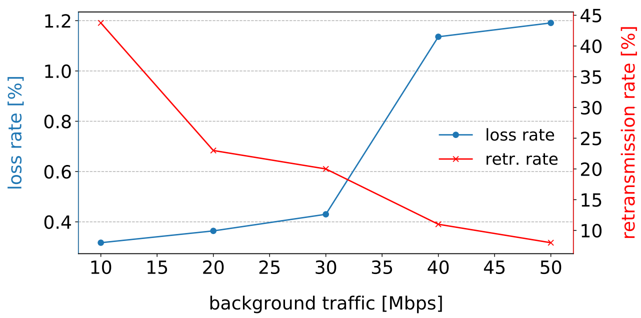

Figure 5: Packet loss ratio due to congestion

According to the obtained results, low background traffic (10-20 Mbps) caused low packet drop rates of around 0.3%, and almost half of these lost packets were retransmitted. On the other hand, setting up high background traffic (40-50 Mbps) and causing congestion, the packet drop rate increased above 1%. The reason why the packet loss rate in case of 40 Mbps and 50 Mbps is similar, is that the DCCP protocol includes congestion control mechanism that reduces the sending rates trying to avoid congestion.

The main goal of our TbSR approach is to increase the clients' satisfaction by improving the quality of the synthesized video stream. The desired perspective of the free-viewpoint video is synthesized by the client, based on the left and right cameras' color and depth views. These camera streams are affected by the packet losses and the distortions propagating to the synthesized views. Therefore, we measured the PSNR values of the received camera streams and the synthesized view quality. In order to make the comparison possible *#cam1* and *#cam3* was delivered and the synthesized perspective was the same as *#cam2* perspective. Therefore, we could compare the original *#cam2* color camera view with the synthesized one. The delivered streams measured PSNR quality is presented in Figure 6.

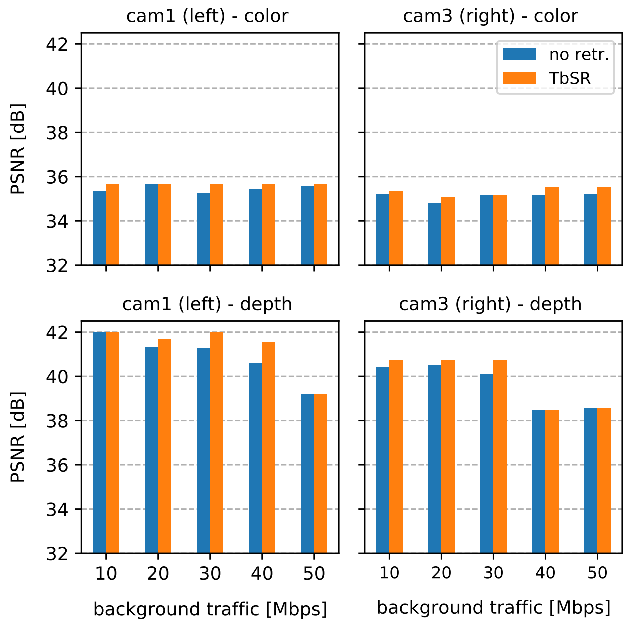

Figure 6: Color and depth camera streams quality of *#cam1* and *#cam3* with and without the selective retransmission scheme

Based on the results, we could improve the received video qualities used for the synthesis. The composed video quality showed similar behavior as shown in Figure 7.

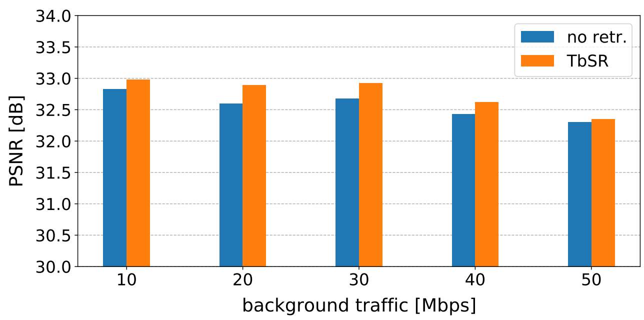

Figure 7: Synthesized view PSNR quality

Unfortunately, neither TCP nor TFRC congestion control algorithms can distinguish packet drops due to congestion and channel errors. All packet losses are assumed to be caused by congestion, hence the estimated free network capacity is always reduced if any package is lost. In a wireless environment, where channel errors are more frequent, the proposed TFRC-based selective retransmission scheme performance also decreases due to underestimated free network capacity.

In our experiments, we set different levels of packet loss rate (1%, 2%, 5%) without congestion in the network and measured the synthesized video quality using the PSNR (peak signal-to-noise ratio) metric. Figure 8 shows a frame-by-frame comparison of the calculated PSNR values with and without the proposed retransmission scheme.

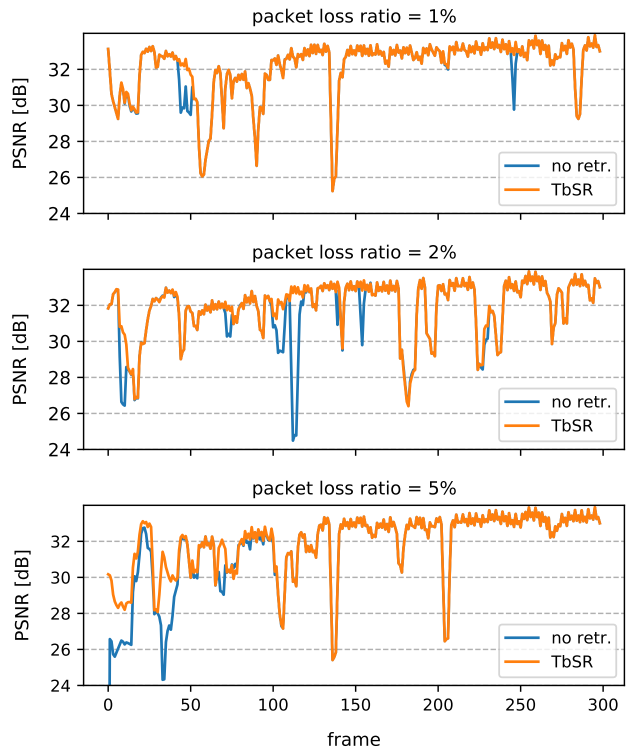

Figure 8: Video quality of synthesized views in case of 1%, 2%, and 5% packet loss rate

In the case of 1% and 2% packet loss rate, the selective retransmission method could improve the quality of any frames in the sequence, while in case of 5% packet loss rate, only the distorted frames at the beginning of the sequence were improved. The reason is that TFRC rate control estimated bandwidth value became lower than the video bitrate and the retransmission was disabled. In general, we could show that the ratio of retransmitted packets decreases with the packet loss ratio.

According to the obtained results, the proposed TFRC-based selective retransmission scheme was most effective when the packet loss ratio was lower than 1%. By increasing the packet loss rate, the TFRC bandwidth estimation algorithm assumes congestion in the network, therefore our method disables the retransmissions. Although, when the background traffic is high and the network is close to the congested state, the algorithm successfully controls the retransmissions and disables them if the retransmitted packets would be lost again due to congestion.

V. CONCLUSIONS

Free-viewpoint video is a promising application that offers freedom for users to change scene perspective. As multi-view video-plus-depth sources – required for the free-viewpoint video streaming service – provide two color and two depth video sequences, the bandwidth requirement is 3-4 times higher compared to regular streaming services. The delivery of these video streams can overload the network. However, retransmission is the simplest error concealment technique, it works effectively only in an uncongested network. The proposed TbSR scheme relies on the TCP-Friendly Rate Control (TFRC) which estimates the available capacity of the links. Although we used DCCP transport protocol in the transport layer, because it provides loss detection and includes the TFRC algorithm, these functionalities can also be implemented on top of UDP in the application layer.

To evaluate our selective retransmission approach, simulations were performed in NS-2 environment and the VSRS tool was used for view synthesis. The algorithm for deciding whether to retransmit a lost packet is able to adapt itself to alternate network conditions. The obtained results proved that the flexibility and performance of the proposed TFRC-based selective retransmission approach provide a potential quality improvement for free-viewpoint video streaming services.

REFERENCES

[1] E. Kohler, M. Handley, and S. Floyd, “Datagram congestion control protocol (DCCP),” RFC 4340, March 2006.

[2] S. Floyd, E. Kohler, and J. Padhye, “Profile for Datagram Congestion Control Protocol (DCCP) Congestion Control ID 3: TCP-Friendly Rate Control (TFRC),” RFC 4342, March 2006.

[3] M. Domański, M. Gotfryd, and K. Wegner, “View synthesis for multiview video transmission,” in *The 2009 International Conference on Image Processing, Computer Vision, and Pattern Recognition, Las Vegas, USA*, vol. 13, 2009, p. 16.

[4] S. Jo, D. Lee, Y. Kim, and C. D. Yoo, “Development of a simple free viewpoint video system,” in *2008 IEEE International Conference on Multimedia and Expo.* IEEE, 2008, pp. 1577–1580.

[5] S. Lu, T. Mu, and S. Zhang, “A survey on multiview video synthesis and editing,” *Tsinghua Science and Technology*, vol. 21, no. 6, pp. 678–695, 2016.

[6] C. Neupane, A. Koirala, Z. Wang, and K. B. Walsh, “Evaluation of depth cameras for use in fruit local-

ization and sizing: Finding a successor to kinect v2," *Agronomy*, vol. 11, no. 9, p. 1780, 2021.

[7] C. Fehn, "Depth-image-based rendering (DIBR), compression, and transmission for a new approach on 3D-TV," in *Stereoscopic displays and virtual reality systems XI*, vol. 5291. SPIE, 2004, pp. 93–104.

[8] Z. Ni, D. Tian, S. Bhagavathy, J. Llach, and B. S. Manjunath, "Improving the quality of depth image based rendering for 3D video systems," in *IEEE International Conference on Image Processing (ICIP)*. IEEE, 2009, pp. 513–516.

[9] P. Merkle, Y. Morvan, A. Smolic, D. Farin, K. Mueller, P. De With, and T. Wiegand, "The effects of multiview depth video compression on multiview rendering," *Signal Processing: Image Communication*, vol. 24, no. 1-2, pp. 73–88, 2009.

[10] E. Bosc, R. Pepion, P. Le Callet, M. Koppel, P. Ndjiki-Nya, M. Pressigout, and L. Morin, "Towards a new quality metric for 3-D synthesized view assessment," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 7, pp. 1332–1343, 2011.

[11] Y. Chen, M. M. Hannuksela, T. Suzuki, and S. Hattori, "Overview of the mvc+ d 3d video coding standard," *Journal of Visual Communication and Image Representation*, vol. 25, no. 4, pp. 679–688, 2014.

[12] M. M. Hannuksela, Y. Yan, X. Huang, and H. Li, "Overview of the multiview high efficiency video coding (mv-hevc) standard," in *2015 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2015, pp. 2154–2158.

[13] Á. Huszák, "Advanced free viewpoint video streaming techniques," *Multimedia Tools and Applications*, vol. 76, no. 1, pp. 373–396, 2017.

[14] T.-Y. Ho, Y.-N. Yeh, and D.-N. Yang, "Multi-view 3d video delivery for broadband ip networks," in *2015 IEEE International Conference on Communications (ICC)*. IEEE, 2015, pp. 5796–5802.

[15] P. Carballeira, C. Carmona, C. Díaz, D. Berjón, D. Corregidor, J. Cabrera, F. Morán, C. Doblado, S. Arnaldo, M. del Mar Martín *et al.*, "Fvv live: A real-time free-viewpoint video system with consumer electronics hardware," *IEEE Transactions on Multimedia*, vol. 24, pp. 2378–2391, 2021.

[16] T.-Y. Chung, S. Sull, and C.-S. Kim, "Frame loss concealment for stereoscopic video based on interview similarity of motion and intensity difference," in *2010 IEEE International Conference on Image Processing*. IEEE, 2010, pp. 441–444.

[17] X. Liu, Q. Peng, and X. Fan, "Frame loss concealment for multi-view video plus depth," in *2011 IEEE 15th international symposium on consumer electronics (ISCE)*. IEEE, 2011, pp. 208–211.

[18] B. Oztas, M. T. Pourazad, P. Nasiopoulos, and V. C. Leung, "Adaptive 3d-hevc video streaming over congested networks through layer prioritization," in *2017 24th International Conference on Telecommunications (ICT)*. IEEE, 2017, pp. 1–5.

[19] E. Kurutepe, A. Aksay, C. Bilen, G. Gurler, T. Sikora, G. B. Akar, and A. M. Tekalp, "A standards-based, flexible, end-to-end multi-view video streaming architecture," in *Packet Video 2007*. IEEE, 2007, pp. 302–307.

[20] N. Feamster and H. Balakrishnan, "Packet loss recovery for streaming video," in *12th International Packet Video Workshop*. PA: Pittsburgh, 2002, pp. 9–16.

[21] G. Li, S. Liu, H. Li, W. Lei, and W. Zhang, "An adaptive retransmission-based multipath transmission mechanism for conversational video," *International Journal of Communication Systems*, vol. 31, no. 15, p. e3778, 2018.

[22] S. Liu, W. Lei, W. Zhang, and Y. Guan, "Cmt-sr: A selective retransmission based concurrent multipath transmission mechanism for conversational video," *Computer Networks*, vol. 112, pp. 360–371, 2017.

[23] L. Han, S.-S. Kang, H. Kim, and H. P. In, "Adaptive retransmission scheme for video streaming over content-centric wireless networks," *IEEE Communications Letters*, vol. 17, no. 6, pp. 1292–1295, 2013.

[24] M. T. Piecuch, K. French, G. Oprica, and M. Claypool, "Selective retransmission protocol for multimedia on the internet," in *Multimedia Systems and Applications III*, vol. 4209. SPIE, 2001, pp. 79–90.

[25] Á. Huszák and S. Imre, "Source controlled semireliable multimedia streaming using selective retransmission in dccp/ip networks," *Computer communications*, vol. 31, no. 11, pp. 2676–2684, 2008.