

Live Streaming Architectures for Video Data: A Review

Ishwari Ginimav, Prasad G. R, Gowrishankar,
BMS College of Engineering, Bangalore, India.
Email: ishwari@bmsce.ac.in (I.G), prasad.cse@bmsce.ac.in (P.G.R),
gowrishankar.cse@bmsce.ac.in (G).

Abstract: Video communication has become a crucial part of our everyday lives due to its ease of use and applicability. Also, the demand for video streaming services has grown exponentially due to the recent Covid19 situation. Over the years, researchers have formulated different algorithms and architectures for effective video streaming. Due to the presence of a large variety of algorithms, system designers are often unable to decide the most optimum video streaming architectures for effective system design. In this work, some of the most recent and effective streaming architectures and algorithms are discussed along with their performance metrics. This will assist readers in deciding the most optimum algorithms and architectures for their application. Along with the algorithmic comparisons, this text also recommends certain improvements in the reviewed algorithms for further improving their performance.

Keywords: Video streaming; big data; machine learning computations.

1. Introduction

In order to achieve effective video streaming the system designers must gain expertise in multi-domain operations, which include but are not limited to [1],

- Video encoding and decoding processes
- Channel encoding and decoding processes
- Stream-data processing
- Channel and stream feedback-based improvement via run-time reconfiguration of the above processes

To design an effective video-streaming system, the system designers first need to analyse the different video types that will be supported by their platform. Once these video types are identified, then the encoding and decoding processes of these video types must be studied and implemented. Thorough testing of these encoders and decoders along with performance evaluation must be done to identify the performance gaps. These gaps can be in terms of delay, throughput, bandwidth, error rate, and any other application-specific parameters. Once these gaps are identified then the optimizations techniques are applied to fill in these gaps and make the system more efficient.

These encoding and decoding systems are then deployed on different devices separated by a communication medium. The output stream data from the transmitter section is passed through the communication channel, wherein noise and distortions are added to it. The received signal is decoded by the selected decoder, and the output signal to noise ratio (SNR) is observed. Due to the noise and distortions, the output SNR values are low. This is an indication of low visual quality at the receiver side. The output SNR is improved by adding channel models to the transmitter and receiver sections. The channel models analyse the distortions added by the channel, and then modify the transmitted and received signal processing equations. The models analyse the channels by sending and receiving dummy packets and then doing the said modifications. A good channel model will modify the transmitted signal such that there are minimum distortions during signal reception even under moderate to heavy noise conditions.

Usually audio and image streams are processed separately [2], because encoders and decoders for audio and video are designed separately. A sample video streaming system can be observed from figure 1, wherein the video encoder and audio encoder streams are given to a multiplexing device.

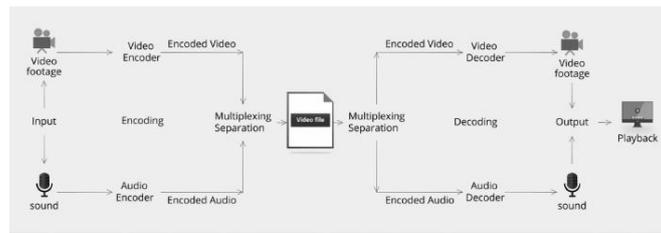


Fig. 1: A sample AV streaming system

This device performs audio and video stream synchronization and then performs decoding at the receiver side for effective video reproduction. Usually, a streaming server is needed to divide the input video into small chunks of video snippets. These snippets can be formed using protocols such as Moving Picture Experts Group Dynamic Adaptive Streaming over HTTP (MPEG DASH), Real-Time Messaging Protocol (RTMP), Smooth Protocol by Microsoft, HTTP Live Streaming (HLS), Real-Time Streaming Protocol (RSTP), and others [3]. The stream-data processing layer handles these operations, and finally streams the video on the receiver device.

2. Literature Review

2.1 Buffer Management System

Video streaming protocols are based on effective real-time processing and continuous trans-reception of audio-video chunks. These chunks of audio and video are usually modeled in a manner so that the trans-reception process has a high quality of service (QoS) and quality of experience (QoE). For instance, the work in [3] uses an adaptive video streaming service using buffer analysis, wherein instead of using the ping-packet arrival rate as an estimate of video transmission bandwidth, the system uses a buffer management system (BMS). The BMS checks for the delay needed to fill in the buffer at the receiver-side and then sends a ping packet to the transmitter. The delay between receiving these packets on the sender side evaluates the bandwidth of the trans-receiver combination. The main advantage of using this approach is the possibility of adaptive bandwidth calculations. Using this adaptive bandwidth, the transmission speed can be modified, which results in a minimum quality loss, and better end-user experience. It is observed that the proposed system can improve the video stream quality by 15% in terms of peak signal to noise ratio (PSNR) and structural similarity indices. The architecture for the buffer management system can be observed in figure 3, wherein message analysis and video parameter adaptation are performed based on the buffer fill packets on the server-side. While buffer underflow monitoring combined with regression analysis & feedback analysis is responsible for checking the buffer status and informing the server about the buffer-filling delays via a feedback message loop.

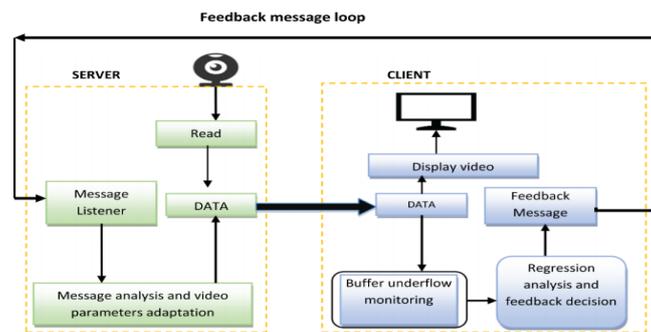


Fig. 2: Buffer over-flow analysis for video streaming [3]

2.2 Channel Rescheduling Mechanism

The channel rescheduling mechanism [6] is based on the HTTP algorithm. The proposed algorithm performs 2 tasks,

- Channel placements, wherein the video streaming channel is allotted to a given trans-receiver pair.
- Channel rescheduling, wherein the unused channels are rescheduled to support higher bit rates, and if any intermediate channels are showcasing high packet loss, then another channel is replaced to improve the QoE.

The block diagram of the channel re-scheduling architecture can be observed from figure 3, wherein the resource-aware video streaming model is defined. The model uses algorithms for checking the fluency of the video over HTTP which is controlled by evaluating the bandwidth via which the video is transmitted. These parameters are combined and given to a HAS Channel Placement (HCP) and HAS Channel Rescheduling (HCR) block, for better channel allocation during video streaming.

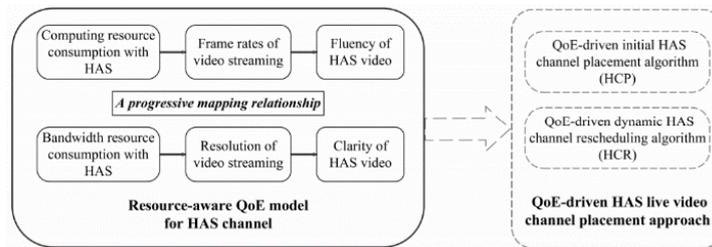


Fig. 3: Channel rescheduling architecture [6]

2.3 Satellite Backhauling Effect

The satellite backhauling mechanism [9] utilizes the concepts of initial stalls and delays on the HTTP DASH and MPEG DASH protocols over 5th Generation (5G) networks. This resulted in an increase of 20% in the throughput values while reducing the latency to less than 10%. All this is possible due to satellite backhauling, which can be observed in figure 4. Using this effect, the satellite terminals can increase the performance of video transmission. Backhauling connects the clients directly to the sending mobile edge node, thereby reducing the latency in transmission. But this system is very computationally complex to implement because the sender and receiver devices must be synchronized using data-link level protocols, rather than application-level protocols. Moreover, due to backhauling the security of the satellites might get compromised, thereby compromising the security of the entire video streaming architecture. This makes the system prone to attacks like side-channel attacks.

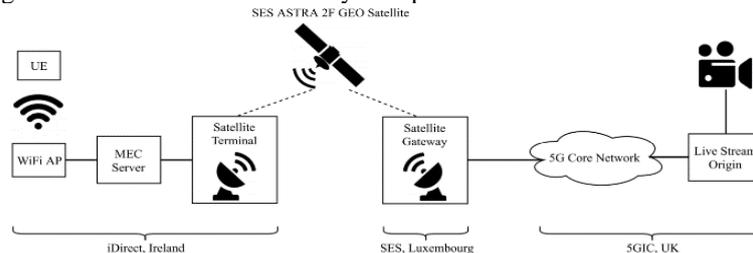


Fig. 4: Satellite Backhauling Effect [9]

2.4 Crowdsourced Network

Crowdsourcing [13] assists the streaming quality by storing parts of the video stream on different available mobile devices, each of these devices can communicate the video to nearby devices (when needed). This further shifts the computational device closer to the video consuming entity, thereby reducing the delay and increasing the throughput of video streaming. The work in [13] proposes such crowdsourced live streaming architecture or CLS. Moreover, due to the crowdsourcing architecture, the server load is drastically reduced by 40%, thereby increasing the streaming revenues by the same percentage. The system uses a 2-level network as shown in figure 5, wherein the first level is made up of crowdsources. These crowdsources try to locally fulfill the user's video streaming quality and bandwidth requirements. Sometimes these requirements are not fulfilled due to the following cases,

- Content not available with the local streaming devices.
- The available content is present but is fragmented across so many devices, that it is not effective to combine the streams from these devices for single video transmission.
- The distance between the devices is larger than the threshold for effective communication.
- Video content present in the devices is not complete or is getting dropped during transmissions.

Under any such case, the level 2 data is used. The level 2 data is an entire repository of the videos, from where videos can be effectively fetched. This kind of network can reduce the server network load, by keeping a respectable level of PSNR and bandwidth values.

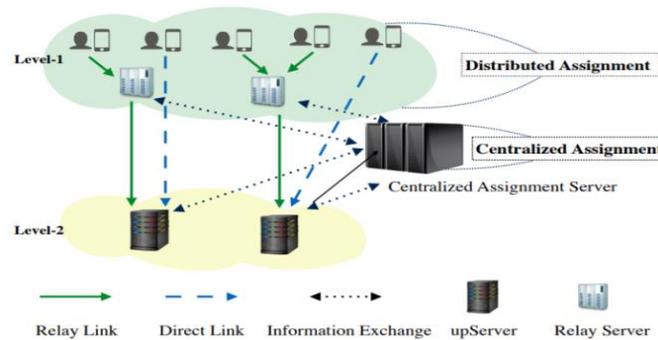


Fig. 5: The 2-Level Crowdsourced Network [13]

2.5 QoE Prediction Using Machine Learning

The work proposed in QoE prediction using machine learning [19] describes different algorithms that are applied to the MPEG DASH protocol. Algorithms like Artificial Neural Network (ANN), Support Vector Machine (SVM) and, K-Nearest Neighbours Algorithm (KNN) are applied to the MPEG DASH protocol for prediction of QoE. The architecture of prediction can be observed from figure 6, wherein the audio and video streams are separately encoded and decoded to produce the most effective transmission quality. The encoding and decoding process is governed using a subjective QoE database, which is backed by a machine learning layer. The machine learning layer can optimize the bit-rate with the help of pre-trained models stored in the QoE database. As a result of this, the NN-based system can improve the throughput of the system by more than 10% when compared to conventional DASH protocol.

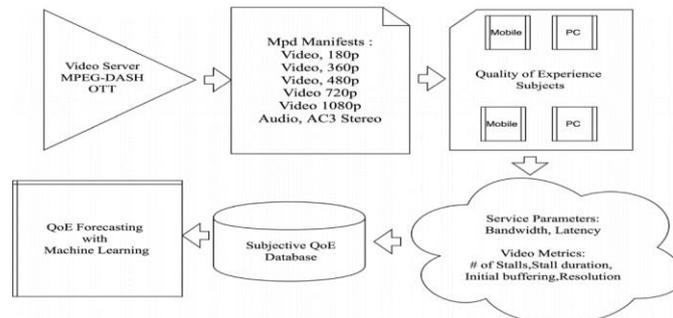


Fig. 6: QoE Prediction Using Machine Learning [19]

2.6 Drone Based Communication System

In the work proposed in drone-based communication network [20] the drones are used as mini-mobile edge centers for broadcasting video data. The drone-based solution categorizes the drones into two types,

- Video transmission drone
- Video multi-cast gateway drone

The video transmission drone collects data from the physical video streaming device and sends it to the multi-cast gateway drone. This drone selects the best possible drone-path for the data to be communicated. Using this path, the data is sent to the receiving entity. The architecture has a lot of issues, which include but are not limited to,

- The insufficient battery capacity of drones for long video playback.
- Security concerns for the drones.
- Physical damage to the drones might reduce visual quality.

The inter-drone data is communicated using multi-cast protocols, while the drone-to-device communication uses unicast, as shown in figure 8, wherein a two-hop communication network is shown in figure 7. The network has improved throughput and reduced packet loss but suffers from low network lifetime, and high-security concerns as already mentioned. The drone-based systems can be effectively applied to surveillance applications.

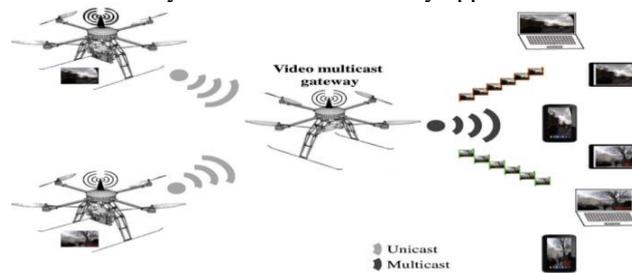


Fig. 7: Drone Based Communication Network [20]

2.7 The CaR-PLive Network

The work proposed in the CaR-PLive network [22] combines the advantages of P2P networks cloud computing to design an effective cloud-based P2P live video streaming algorithm. The algorithm cloud-assisted reinforcement learning with P2P networks for live video streaming (CaR-PLive) uses cloud storage services (CSSs) to store the P2P data and then broadcasts this data with the help of the content delivery network (CDN). Due to the combination of cloud and peer to peer delivery network, the overall network throughput is improved by 20%, while improving the speed by 10%. The CaR-PLive network architecture can be seen in figure 8, wherein components like overlay manager, buffer manager, chunk scheduler, and buffer to video interface are connected to achieve better video transmission quality.

Users are divided into root and non-root peers. Wherein, root nodes are responsible to collect the data from the users, while the non-root users are responsible for data communication. The root-users are also responsible for data syncing between the cloud and the P2P network. This synchronization has to be managed by the network peers, which requires a lot of computations to be performed on the peer nodes. This reduces the peer lifetime and most importantly reduces the peer speed.

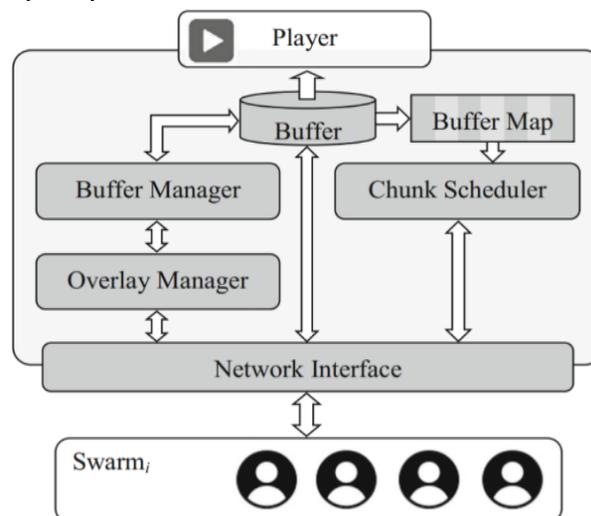


Fig. 9: The CaR-PLive Network [22]

2.8 AC Unit System

The work uses Autonomic Computing (AC) [23] to perform this task. The AC units can achieve monitoring, failure diagnosis, repair, and prediction features. These features are incorporated using context-sensitive machine learning techniques. Due to the proposed system, the delay can be reduced to 1/3rd, while maintaining a high level of prediction accuracy and good PSNR. An Autonomic computing element can be observed from figure 9, wherein processes like monitoring, analysis, planning, and execution are performed with the help of sensors and actuators.

This monitoring unit uses probing agents, data aggregation, and failure detector to check the status of the devices. This is followed by a prediction layer, which checks for server load analysis, with a prediction of failure and learning from these failure data using machine learning analysis. The diagnosis layer is attached to the prediction layer, which performs recovery self-injection and finally, the cloud stream is recovered from any inherent blockages. Due to such a complex process, the overall uptime of the server is improved to more than 99%. In the case of the absence of this layer, there is a lot of packet loss in the network.

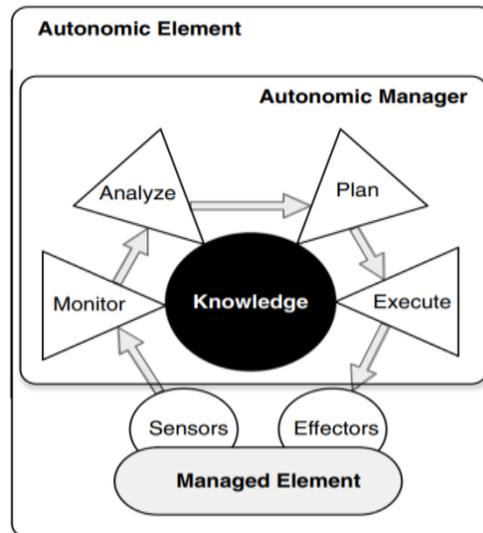


Fig. 9: The AC Unit for Live Video Prediction [23]

3. Comparison of Different Algorithms

To statistically analyse the algorithms, this section compares the algorithms in terms of average end to end delay for communication, average throughput (T), average packet loss (PL), average peak signal to noise ratio (PSNR), and average computational complexity (CC). Based on this analysis, researchers can select these algorithms for their specific application, and if needed combine the algorithms to form hybrid versions of the same. Also, some applications are suggested in the (Application) column of the table. Table 1 showcases the statistical analysis of these algorithms.

The ranges of low, medium, and high are decided based on the performance of the said algorithms when compared to standard DASH, HTTP DASH, and MPEG DASH algorithms. These algorithms are considered to be the base-line techniques, and if the performance metric of the given technique is higher than the one presented by these techniques, then the metric 'H' is used, if it is similar to these techniques, then 'M' is used, but if the metric is lower than 'L' is used. It is observed that the CaR-PLive method outperforms other methods for effective live video streaming. But it can be further improved by adding certain artificial intelligence algorithms for improving the QoE parameters.

Table 1: Comparison of Different Algorithms

Algorithm	Application	T	PL	PSNR	CC
Buffer Analysis [3]	Mobile streaming	H	M	M	M
SecP2 PVoD [4]	Security	M	L	H	H

MEC [5]	General live streaming	H	L	M	H
HCP & HCR [6]	Media Cloud	M	M	M	H
HTTP2 [7]	Live streaming	H	M	M	H
Prediction [8]	Live streaming	M	M	M	H
Backhaul [9]	5G	H	L	H	H
Co- operative Client Server [10,11]	Live video	M	L	M	H
Actor critic [12]	Live video	H	M	M	M
Crowd sourced [13]	Live video broadcast	H	M	M	H
GOP [14]	Live streaming	H	L	H	M
P2P Multi Source [16]	MANET	H	M	M	M
Prediction [17]	Live video transcoding	M	M	M	H
QoE Predict [19]	Live stream	M	L	H	H
CaR-PLive [20]	Live stream	H	L	H	L
Drones [21]	Live stream	M	H	M	M
AE [23]	Fault aware systems	M	L	M	H
Priority based Scheduling [26]	Video streaming	H	M	M	H
Cluster compute [27]	Video streaming	H	L	M	H
3D ROI [32]	3D Video streaming	M	H	M	H
Deeplive [34]	Live streaming	H	M	M	H
D-DASH [35]	Live streaming	H	H	M	H

4. Conclusion

From the discussion about the reviewed algorithms, it is observed that clustering and classification mechanisms increase the overall computational complexity of the algorithms, but also reduce the server load and assist in obtaining higher throughput and PSNR values. Scheduling is another important step to perform video streaming the methods discussed above to assist the video streaming server by scheduling the resources effectively. This improves the load balancing capability of the server, thereby improving the PSNR and reducing the delay of transmission. Algorithms that use crowdsourcing further optimize the overall streaming quality by offloading the server-side storage and computational requirements to the client-side. This has two advantages, first is, the server is relatively free to perform other quality-focused task and second are, the data is closer to the client device, thereby reducing the delay and increasing the throughput. The best work in video streaming can be observed in drone-based communication, wherein high values of throughput, high values of PSNR, low values of packet loss rate, and low values of computational complexity is observed. It is recommended that the CaR-PLive system be used for real-time deployment of video streaming networks, and performance on other video encoding and decoding protocols be studied based on the application area.

Acknowledgement

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of the paper.

References

- [1] Mu WANG, Changqiao XU, Shijie JIA, Gabriel-Miro MUNTEAN, Video streaming distribution over mobile Internet: a survey, *Higher Education Press and Springer-Verlag GmbH Germany, part of Springer Nature (2018)*.
- [2] Luis Rodriguez-Gil, Pablo Orduna, Javier Garcia-Zubia, Diego Lopez-de-Ipina, Interactive live-streaming technologies and approaches for web-based applications, *open access at Springerlink.com, DOI 10.1007/s11042-017-4556-6*.
- [3] L. Arun Raj, Dhananjay Kumar, H. Iswarya, S. Aparna and A. Srinivasan, Adaptive video streaming over HTTP through 4G wireless networks based on buffer analysis, *EURASIP Journal on Image and Video Processing (2017)*.
- [4] Mahender Kumar and Satish Chand, SecP2PVoD: a secure peer-to-peer video-on-demand system against pollution attack and untrusted service provider, *Springer Science+Business Media, LLC, part of Springer Nature (2019)*.
- [5] Yi-Hsuan Hung, Chih-Yu Wang and Ren-Hung Hwang, Optimizing Social Welfare of Live Video Streaming Services in Mobile Edge Computing, *IEEE Transactions on Mobile Computing, DOI 10.1109/TMC.2019.2901786*.
- [6] Junquan Liu, Weizhan Zhang, Shouqin Huang, Haipeng Du and Qinghua Zheng, QoE-driven HAS Live Video Channel Placement in the Media Cloud, *IEEE Transactions on Multimedia (2020)*.
- [7] Hung T. Le, Thoa Nguyen, Nam Pham Ngoc, HTTP/2 Push-based Low-delay Live Streaming over Mobile Networks with Stream Termination, *Ieee Transactions On Circuits And Systems For Video Technology, Doi 10.1109/Tcsvt.2018.2850740*.
- [8] Wenjuan Shi, Yanjing Sun, and Jinqiu Pan, Continuous Prediction for Quality of Experience in Wireless Video Streaming, *IEEE Transaction on Special Section On Recent Advances In Video Coding And Security, Doi:10.1109/Access.2019.2919610*.
- [9] Chang Ge, Ning Wang, et.al., QoE-Assured Live Streaming via Satellite Backhaul in 5G Networks, *IEEE Transactions on Broadcasting (2019)*.
- [10] Jiayi Gu, Jiliang Wang, Zhiwen Yu, and Kele Shen, Traffic-Based Side-Channel Attack in Video Streaming, *IEEE/ACM Transactions On Networking, Vol. 27, No. 3, June (2019)*.
- [11] Sangwook Han, Yunmin Go, Hyunmin Noh, Hwangjun Song, Cooperative Server-Client HTTP Adaptive Streaming System for Live Video Streaming, *IEEE Transactions, DOI 978-1-5386-8350-7/19*.
- [12] Zhicai Zhang, Ru Wang, F. Richard Yu, Fang Fu and Qiao Yan, QoS Aware Transcoding for Live Streaming in Edge-Clouds Aided HetNets: An Enhanced Actor-Critic Approach, *IEEE Transactions On Vehicular Technology, Vol. 68, No. 11, November (2019)*.
- [13] Haitian Pang, Zhi Wang, Chen Yan, Qinghua Ding, Kun Yi, Jiangchuan Liu and Lifeng Sun, Content Harvest Network: Optimizing First Mile for Crowdsourced Live Streaming, *IEEE Transactions on Circuits and Systems for Video Technology (2018)*.

- [14] Xiangbo Li, Mohsen Amini Salehi, Magdy Bayoumi, VLSC: Video Live Streaming Using Cloud Services, *2016 IEEE International Conferences on Big Data and Cloud Computing (BDCloud), Social Computing and Networking (SocialCom), Sustainable Computing and Communications (SustainCom)*.
- [15] Martin Fleury, Dimitris Kanellopoulos and Nadia N. Qadri, Video streaming over MANETs: An overview of techniques, *Springer Science+Business Media, LLC, part of Springer Nature (2019)*.
- [16] Muhammad Salman Raheel and Raad Raad, Energy efficient technique for P2P multisource video streaming in mobile ad hoc networks to improve QoE, *Springer Science+Business Media, LLC, part of Springer Nature (2019)*.
- [17] Pekka Paakkonen, Antti Heikkinen and Tommi Aihkisalo, Online architecture for predicting live video transcoding resources, *Journal of Cloud Computing: Advances, Systems and Applications*.
- [18] Arkadiusz Biernacki, Identification of adaptive video streams based on traffic correlation, *Springer Science+Business Media, LLC, part of Springer Nature (2019)*.
- [19] Utku Bulkan and Tasos Dagiuklas, Predicting quality of experience for online video service provisioning, *SuITE Research Group, Division of Computer Science, London South Bank University, London, UK, Multimedia Tools and Applications (2019)*.
- [20] Miguel Aguayo, Luis Bellido, et.al., Live multicast video streaming from drones: an experimental study, *Multimedia Tools and Applications, Springer Science+Business Media, LLC, part of Springer Nature (2019)*.
- [21] D. Kirubha, K. Ramar, MCCA scheduling for enhancing QoS based video streaming for video surveillance applications, *Springer Science+Business Media, LLC, part of Springer Nature (2018)*.
- [22] Majid Sina, Mehdi Dehghan and Amir Masoud Rahmani, CaR-PLive: Cloud - assisted reinforcement learning based P2P live video streaming: a hybrid approach, *Multimedia Tools and Applications, Springer Science+Business Media, LLC, part of Springer Nature (2019)*.
- [23] Carlos Cunha, Building Autonomic Elements from Video Streaming Servers, *Journal of Network and Systems Management (2019)*.
- [24] Anthony O. Adeyemi-Ejeye, Mohammed Alreshoodi, Laith Al-Jobouri, Martin Fleury, Impact of packet loss on 4K UHD video for portable devices, *Springer Science + Business Media, LLC, part of Springer Nature (2019)*.
- [25] P. Benel Prakash, C. Kezi Selva Vijila, Novel Scheduling Based Intelligent Video Streaming for Device-to-Device Communication in Wireless Networks, *Springer Science + Business Media, LLC, part of Springer Nature (2018)*.
- [26] Kunwar Pal, Mahesh Chandra Govil, Mushtaq Ahmed, Priority-based scheduling scheme for live video streaming in peer-to-peer network, *Springer Science + Business Media, LLC, part of Springer Nature (2018)*.
- [27] Janani Arthanari, R. Baskaran, Enhancement of video streaming analysis using cluster-computing framework, *Springer Science + Business Media, LLC, part of Springer Nature (2018)*.
- [28] A. O. Adeyemi-Ejeye, M. Alreshoodi, L. Al-Jobouri, M. Fleury, Prospects for live higher resolution video streaming to mobile devices: achievable quality across wireless links, *Springer-Verlag GmbH Germany, part of Springer Nature (2018)*.
- [29] Cagri Ozcinar, Erhan Ekmekcioglu, Gholamreza Anbarjafari, Ahmet Kondo, Adaptive multi-view video streaming using side information over peer-to-peer networks, *Springer Science + Business Media, LLC, part of Springer Nature (2018)*.
- [30] Qingyong Wang, Hong-Ning Dai, Di Wu and Hong Xiao, Data analysis on video streaming QoE over mobile networks, *EURASIP Journal on Wireless Communications and Networking (2018) 2018:173*.
- [31] Ali Ballout, Alia Ghaddar and Houssein Wehbi, MMVS/COE: mobile multi-view video streaming with constant order encoding, *Springer Science + Business Media, LLC, part of Springer Nature (2018)*.
- [32] Ehsan Rahimi and Chris Joslin, Reliable 3D video streaming considering region of interest, *EURASIP Journal on Image and Video Processing (2018)*.
- [33] Ran Dubin, Raffael Shalala, Amit Dvir, Ofir Pele and Ofer Hadar, A fair server adaptation algorithm for HTTP adaptive streaming using video complexity, *Springer Science+Business Media, LLC, part of Springer Nature (2018)*.
- [34] Zhao Tian, Laiping Zhao, Lihai Nie, Peiqi Chen and Shuyu Chen, Deeplive: QoE Optimization for Live Video Streaming through Deep Reinforcement Learning, *EasyChair (2019)*.
- [35] Matteo Gadaleta, Federico Chiariotti, Student, Michele Rossi and Andrea Zanella, D-DASH: A Deep Q-Learning Framework for DASH Video Streaming, *IEEE Transactions On Cognitive Communications And Networking, Vol. 3, No. 4, December 2017*.