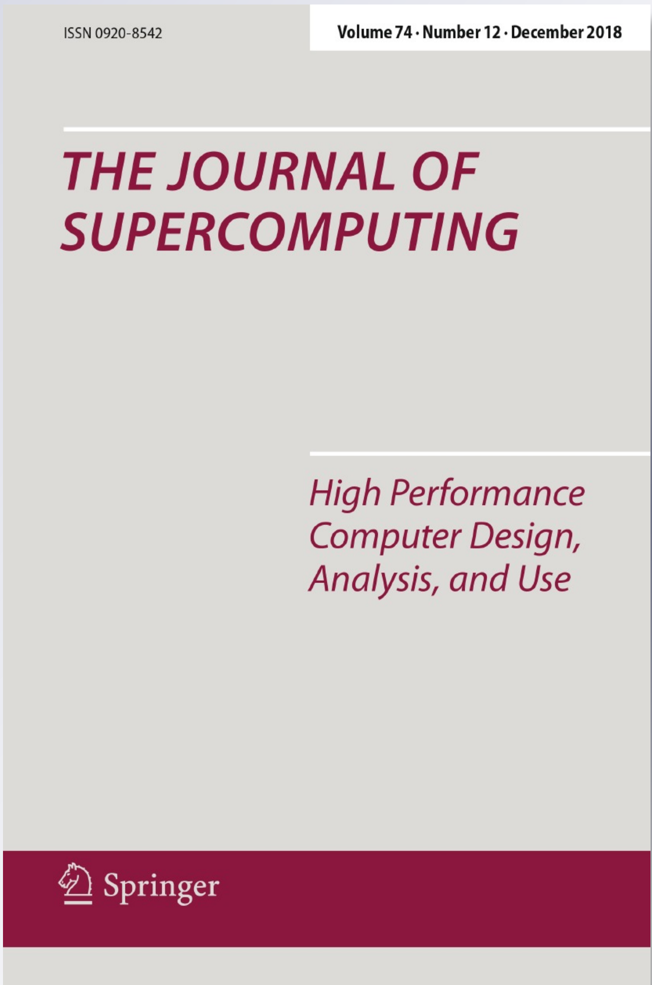
An efficient transmission method based on HEVC multi-view adaptive video streaming over P2P network in NFV

Linh Van Ma, Gwanghyun Yu, Jin-Young Kim, Yonggwan Won & Jinsul Kim

The Journal of Supercomputing
An International Journal of High-Performance Computer Design, Analysis, and Use

ISSN 0920-8542
Volume 74
Number 12

J Supercomput (2018) 74:6939-6959
DOI 10.1007/s11227-018-2594-0



Springer

The Journal of Supercomputing (2018) 74:6939–6959
https://doi.org/10.1007/s11227-018-2594-0

An efficient transmission method based on HEVC multi-view adaptive video streaming over P2P network in NFV

Linh Van Ma[1] · Gwanghyun Yu[1] · Jin-Young Kim[1] · Yonggwan Won[1] · Jinsul Kim[1]

Published online: 11 September 2018

Abstract

Researchers and entertainment companies have given lots of attention to virtual reality over the past decade. 3D multi-view is a technology that provides interactions that are similar to those in the real world. However, 3D video streaming has a high data transfer rate because we must transmit multimedia data at a rate several times higher than that used for regular streaming. Besides, network throughput is unstable due to the inherent limitations of network infrastructure, which degrades video streaming quality. Additionally, network failure can occur frequently, causing stalling in multimedia playback. Hence, a network system is required to have more than one backup route in order to successfully guarantee the reliability of a network at all times. Furthermore, in the field of multi-view transmission, not much research has been published that has been conducted in a network virtualization environment. Therefore, we present a study on adaptive-based, high-efficiency video coding with three-dimensional, multi-view streaming over a peer-to-peer network. First, we study adaptive bitrate streaming methods based on high-efficiency video coding. Then we research transmitting multi-view data over a multi-path system. In the experiment, we first record a video from different views using five cameras. Next we merge recorded videos from the five cameras into a file and encode it before transmitting it over the peer-to-peer network. Moreover, we build a virtualized system using Docker virtualization technology and network function virtualization. The results of the experiment show that transmitting high-volume data over a multi-path network channel increases the streaming buffer level, which is about 20% higher than an adaptive streaming 3D method. It also makes the video quality 4% higher than in an HEVC-based adaptive streaming method.

Keywords HEVC · 3D multi-view · Multi-path · CDN · NFV · Adaptive streaming · P2P · Video streaming

Extended author information available on the last page of the article

1 Introduction

People are especially interested in high-volume multimedia streaming, such as in ultra-high-definition video streaming over the internet. Users require high-quality video streaming and are eager for interactive and practical video technologies, such as multi-view streaming. Indeed, many service providers have recently begun to employ multi-view streaming due to its free-viewpoint switching, as shown in Fig. 1. The increasing popularity of smart portable devices, as well as the remarkable computing and networking capabilities available, opens a range of exciting prospects for the development of mobile 3D video applications and 3D reconstruction [1]. For instance, a spontaneous congregation of client devices can take place to independently record videos of a sporting event from different angles, at the same time. Those clients may be interested in sharing their respective video feeds among themselves. Consequently, we have different views of the event in those videos, which could be used to build a multi-view video with different qualities. Hence, we can exploit the video and build a streaming system that gathers data from users and broadcasts the stream to other users who are waiting to watch the event.

The demand for such high-volume multimedia transmission poses many challenges with regard to network delivery. As addressed in [2], we first had the problems of variable bandwidth, packets loss, and delay due to the best effort nature of the internet. Furthermore, compression techniques are incredibly vulnerable to missing data on the client side. For example, a file is transferred from a server to a client using a protocol to handle small packets. It may not have any problem if the packets arrive on time and in order. However, those packets might be dropped during a transmission session and conflicted by routing protocols while traversing different networks or firewalls. And the heterogeneity of devices necessitates that video streaming adapts the characteristics of the devices. More specifically, a smartphone model can only adapt to specific multimedia services because each different model has distinct computational capacities and pre-installed features.

Several approaches have been proposed to improve multi-view video streaming, services such as multi-view video coding (MVC). MVC aims to maximize the com-

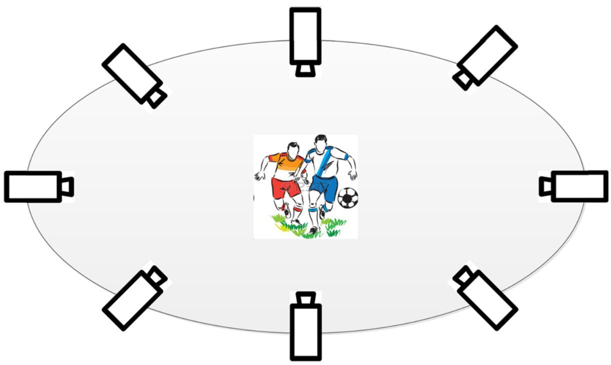


Fig. 1 Multi-view presented by multi-camera recording

pression efficiency of data features with inter- and intra-view dependencies. Another research [3] focused on stabilizing throughput for network connections to minimize packet loss and reduce network latency. Another approach examined the idea of choosing a platform for transmitting media data, such as a hybrid transmission mode [4]. However, these approaches still have some problems in delivering content over the internet. The two approaches mentioned above can provide high-quality streaming services for devices that have high computational resources, but they would have trouble with devices that cannot decode or encode video in a timely manner. To that end, in this article, we present a study on adaptive-based high-efficiency video coding (HEVC) [5,6] with multi-view streaming over a peer-to-peer network [7]. In brief, the contribution of this research mainly focuses on multi-view and multi-path data transmission. In detail, we divided this research into three small parts. First, we encode video before transmitting over a peer-to-peer network (P2P) by using the latest technologies of multimedia compression (HEVC) cooperating with multi-view video coding. Second, we research transmitting multimedia data over a multi-path system using content delivery network (CDN) service and a P2P network. Finally, we carry out additional research into adaptive streaming to deliver multi-view video in the same architecture as the P2P network. In the experiment, we build a virtual environment using Docker [8]. In particular, Docker is a category of recently emerged virtualized environment [9–12] that is utilized to evaluate our proposed system. We also employ network function virtualization (NFV) technology in our experimental research. By doing this, we can quickly implement the proposed system and present it as an outstanding system compared to those presented in previous studies.

The rest of the paper is organized as follows. First, we present background knowledge and our main research in Sect. 2. In Sect. 3, we present related works mainly focused on multi-view video streaming. In Sect. 4, we introduce the multi-view streaming system with multi-path. In Sect. 5, we conduct an experiment and discuss the results found. Section 6 concludes our study with a discussion of future research directions.

2 Background knowledge

2.1 P2P streaming

A new network architecture, called peer-to-peer (P2P), is proposed to overcome the limitations of the client–server model. In the P2P network, every client can be considered as a server and vice versa. Figure 2 presents the client–server model, in which a server receives and responds to all requests coming from clients. Conversely, Fig. 3 depicts the new P2P model, in which clients can interact with each other, and their roles are equal in the network. Peer-to-peer networks [13–15] are computer networks in which activity is primarily based on the computing power and bandwidth of participating peers (or computers). Moreover, the P2P network infrastructure does not focus on a small number of central servers, as is the case in client–server architecture. In the P2P system, we only need one server that supports communication between peers synchronously. Specifically, an actual P2P network does not rely on the concept of servers and clients. Put another way, all participating machines are equal and

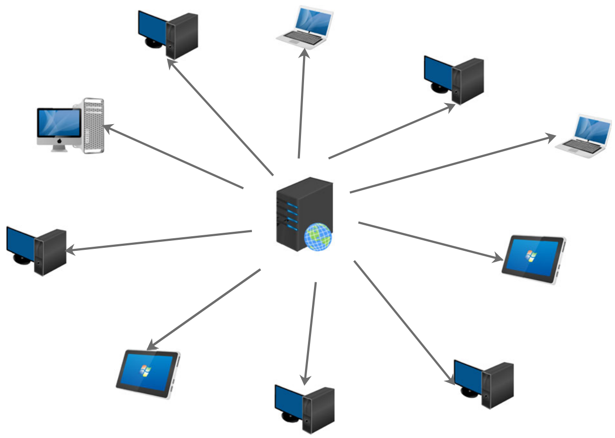


Fig. 2 Overview of server-based network where a server is in-charge of all request coming from clients

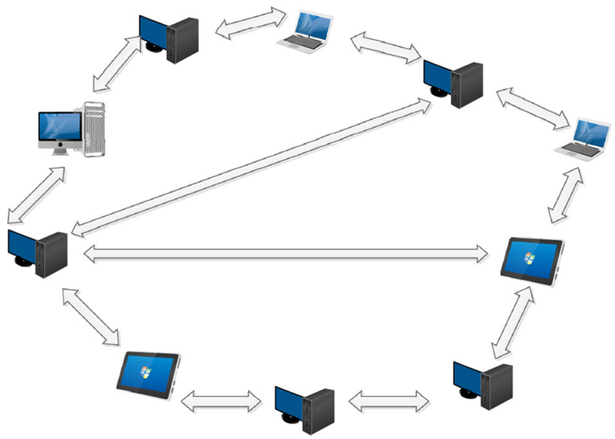


Fig. 3 Overview of P2P network where each client is treated equally in the network

are called peers, each of which is a network node acting as both a client and a server. Indeed, the peer not only retrieves data but shares data with other peers, which is called seeding. Seeding is critical in the development of P2P networks. In short, P2P is just like the client–server model, except that in the latter, there is no seeding. Peer-to-peer networks are often used to create ad hoc connections between machines. They have many applications. The most common of these are file sharing (audio, video, data, etc.) and real-time data transmission, such as in voice over internet protocol (VoIP) [16,17] telephony.

In an unstructured peer network, network nodes linked in the overlay network are randomly set without any given rules. This type of network can be built easily. Put another way, if a new peer wants to join the network, it can obtain the available links of another peer on the network and gradually add itself to the links. If a peer wants to find data in an unstructured peer-to-peer network, a search request is routed across the network to find as many peers that share common data as possible. This network structure clearly has some disadvantages. First, it does not guarantee that the search will be successful. If a large number of peers have common data, the probability of a successful search is quite high. Conversely, if the data are only shared with several peers, the probability of finding it is quite small. We can identify this problem because there is no correlation between a peer and the data it manages in an unstructured P2P. To that end, the search requests are randomly transferred to a number of peers in the network. If a large number of peers share common data, a successful search is more likely.

Another disadvantage of the unstructured P2P network structure is that a search request is usually forwarded to a large number of peers without direction in the network. This request consumes large amounts of network bandwidth, which results in low-performance search efficiency. In the system we present in this research, we prevent these kinds of issues by using a content delivery service (CDN) combined with the P2P network. The combination is discussed later, in the system overview section.

Structured P2P networks overcome the disadvantage of unstructured networks by using the distributed hash table (DHT) system. This system defines links between nodes in the network by using a hash function with key and value. Subsequently, it closely identifies each node responsible for a portion of the shared data in the network. With this structure, when a machine needs to find data, it only needs to apply a common protocol to determine which network node is responsible for the data. Then, it communicates directly to that network node to retrieve search results.

P2P hybrid models are the first-generation peer-to-peer network, which is still based on a central index server. The server stores two crucial tables. First, the file index table contains information about files shared by clients. Second, the connection table presents information about connections between clients, such as bandwidth and IP address. Once a search request is sent to the central server, it will be analyzed. If the request is resolved, the server responds to the request with the IP address of the machine containing resources such as files. In consequence, file transfer is performed in the same manner as peer-to-peer networks, without the server.

2.2 High-efficiency video coding

In the era of the modern internet, the transmission and storage of video and multimedia files have become a necessity of daily life. The performance of the hardware devices has been improved significantly and are available at low prices. For that reason, downloading a video or watching a TV program is more accessible and more convenient. Nowadays, quality of life is increasing, with the result that people are demanding a better audiovisual experience to match their increasing demand for a higher quality of experience (QoE) in general. A few years ago, enjoying a movie at 480p (progressive scanning) or 720p high-definition (HD) resolution was enough to satisfy the viewer. However, videos with 1080p (full HD) resolution have become popular today. Since the emerged of 4K (double full HD), the demand for a better user experience is

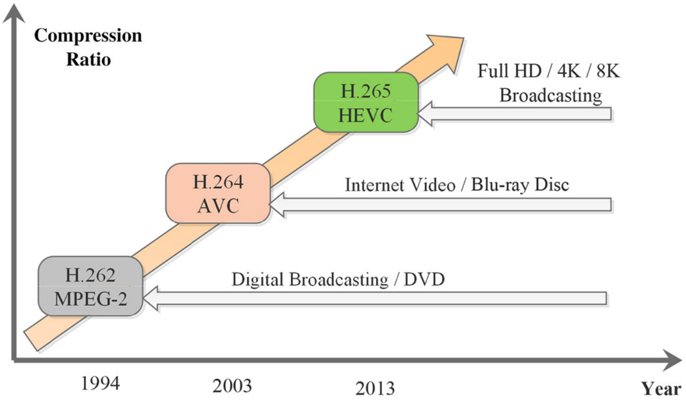


Fig. 4 Evolution of compression technology along years with a performance comparison

increasing. 4K movies are being released that give users an awe-inspiring experience, bringing viewers into another world of ultra-high-quality multimedia.

Researchers must work hard on the video compression problem to upgrade data storage as well as the internet connection to have such a great experience as described above. Therefore, we needed a new video compression standard that is more effective in reducing the data transmission rate. Subsequently, H.265/high-efficiency video coding (HEVC) was born to meet the requirement. Indeed, H.265/HEVC [18–22] is a new video compression standard adopted by the telecommunication standardization sector (ITU-T) and deployed commercially. H.265 promises twice as much compression as its predecessor, H.264/AVC (advanced video coding), which is currently in widespread use, accounting for 80% of current video content. Figure 4 illustrates the evolution of the above-mentioned compression technologies.

HEVC has twice the data compression rate as H.264/MPEG-4 AVC at the same level of video quality. In other words, it can improve picture quality significantly at the same bitrate. It is capable of supporting 8K ultra-high-definition television (UHDT) at up to 8192 × 4320 (4320p) resolution. H.265/HEVC also offers improvements in sound, space, color, and most importantly, it is capable of monitoring an enhanced range of performance. Furthermore, it not only provides the same image quality as H.264/AVC but has better compression than H.264/AVC. In addition, the high compression data rate also reduces the bandwidth required to transfer multimedia data, and storage space can be reduced. This results in lower costs for internet bandwidth and storage media. These remarkable advantages of HEVC are also a boost for the 4K/UHD audiovisual market. Therefore, the efficiency of data compression is crucial to the experience of 4K (ultra HD). Theoretically, H.265/HEVC is estimated to be 3050% more efficient than H.264/AVC. At higher resolutions, the compression efficiency of H.265/HEVC is higher also. However, H.265/HEVC has a structure similar to previous standards, such as MPEG-2 and H.264/AVC.

The basic coding processes of all compression techniques are shown in Fig. 5. Specifically, the above-mentioned coding standards are based on video coding techniques that can be described as follows. First, an image is partitioned into macroblocks,

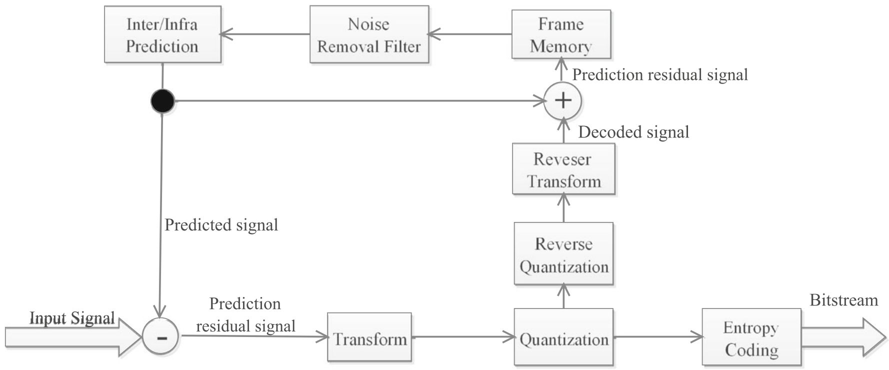


Fig. 5 Coding basic processes

and these macroblocks are further partitioned into smaller blocks. Second, redundant space is reduced by using compression techniques in the internal frame. Third, time consumption is also reduced by using inter-frame compression (motion prediction and motion compensation). Fourth, redundant data are compressed using mathematical and quantized transformations. Fifth, entropy coding is employed to reduce redundancy in motion vector transmissions and signals. Theoretically, entropy information is a concept of entropy in thermodynamics and statistical mechanics that has been extended into the field of information theory. Entropy information describes the degree of chaos in a signal taken from a random event. In other words, the entropy also indicates how much information is in the signal, in which the information is not random parts of the signal. A sequence of frames from the source video will be encoded or compressed by the H.265/HEVC encoder, resulting in a compressed video bitstream. This compressed bitstream can be stored or transmitted. A video decoder then decompresses this bitstream into a series of frames for video playback. In particular, the detail of the encoding processes of HEVC is shown in Fig. 6. HEVC first breaks down an image into coding tree units (CTU). The CTU can continue to be divided into coding units (CU), which can be split into transform units (TU) and prediction units (PU).

2.3 Multi-view video coding

Multi-view video coding (MVC) [2,23–25] is an extension of the H.264/MPEG-4 AVC video compression standards, popularly known as advanced video coding. In particular, MVC allows efficient encoding of sequences captured simultaneously from multiple camcorders with a single data stream. The MVC is oriented to the stereo video encoding, the free-viewpoint television (FTV), and the television 3D. The MVC encoding system has been designed to be compatible with H.264/AVC, which means that older devices can decode stereo video, ignore additional information from the second point of view, and display the image in 2D.

MVC arose with the large volume of data that involves the transmission of video from multiple points of view, which requires much more bandwidth than traditional videos, such as a single view videos. Consequently, compressing multi-point videos

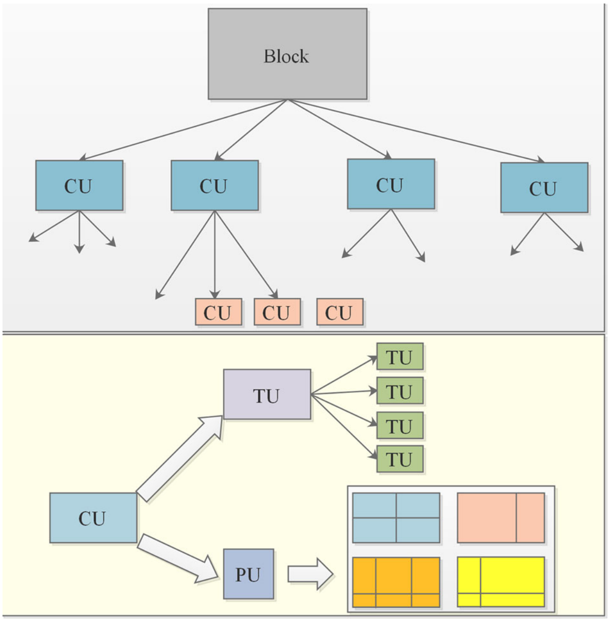


Fig. 6 Basic processes of HEVC with block structure

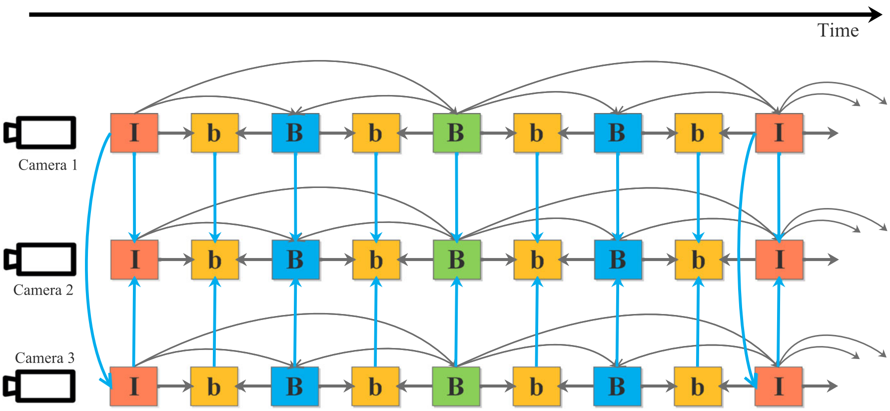


Fig. 7 Temporal/inter-view prediction structure for MVC where squares represent data units (video frames) and arrows depict encoding dependencies between frames

efficiently is crucial in order to reduce the bandwidth requirement for MVC. In fact, a multiple-view video consists of video sequences captured by multiple cameras from different angles and places. Hence, the temporary combination and the prediction of the temporary view are the keys to efficient coding in MVC. The prediction is based on the assumption that video frames from different points of view can be freely exchanged or simultaneously available in the encoder. In consequence, we can only predict a frame of a camera from frames of the same camera.

Multi-view video source is recorded from several cameras capturing the same scene from different angles. This leads to redundancy, which contains a large number of statistical dependencies between views. Consequently, combining temporal prediction and inter-view is crucial for improving the quality as well as the efficiency of MVC. Figure 7 depicts encoding processes using three cameras. It shows that we cannot predict a picture captured from a certain camera from the pictures captured from other cameras. However, we can use pictures of neighboring cameras to increase the efficiency of prediction.

2.4 Adaptive bitrate streaming

Adaptive bitrate streaming has recently emerged as a technique used for streaming multimedia over computer networks. In the past, most video streaming technologies utilized streaming protocols such as real-time transport protocol (RTTP) with real-time streaming protocol (RTSP). On the other hand, adaptive streaming technologies are almost exclusively based on hypertext transfer protocol (HTTP) and designed to work efficiently over large distributed HTTP networks such as the internet.

Adaptive bitrate streaming (ABS) includes the creation of instance entities that are small files. ABS makes these usable by many different users depending on bandwidth and computational resources. In other words, ABS technologies can switch streaming quality as needed to ensure continuous use of the files by adjusting the CPU usage or buffering state of a streaming device. The key to the difference between adaptive streaming technologies is in the streaming protocols employed. For example, Adobes RTMP-based dynamic streaming technology uses the real-time messaging (RTMP) protocol. Put another way, it requires a streaming server and a near-continuous connection between the server and users. These requirements not only increase the cost of execution but lead to potential blocking of the RTMP by a firewall.

A large number of internet companies are paying attention to HTTP streaming services, such as YouTube. All HTTP-based adaptive streaming technologies now use a combination of compressed media and uncompressed files. Dynamic adaptive streaming over HTTP (DASH) [26,27] is one version of ABS. It works to incorporate the best features of all HTTP adaptive streaming technologies into a standard for use on all interfaces, from mobile to server. An overview of DASH work flow is depicted in Fig. 8.

For DASH, encoded file streams are called media presentations (MP), while the manifest file is called a media presentation description (MPD). MP is a collection of structured audio and video content that has description fields, such as periods, adaptation sets, representations, and segments. MP defines video clips with one or several consecutive separated periods from the beginning to the end of the video. Each adaptation-set consists of several representations of a stream of adaptive streaming. For example, a representation is described as 640×480@ 500 kbps. Each representation is split into media segments (MS). In the DASH manifest file, MPD is an XML file that defines the various content elements and the location of video streams. This file allows DASH users to identify and start playing video, etc. easily. Figure 9 illustrates the working process of the MPD file.

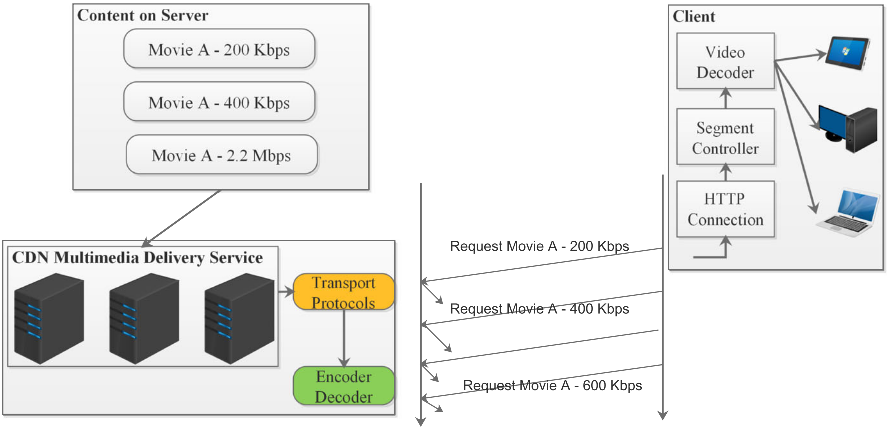


Fig. 8 Overview of dynamic adaptive streaming

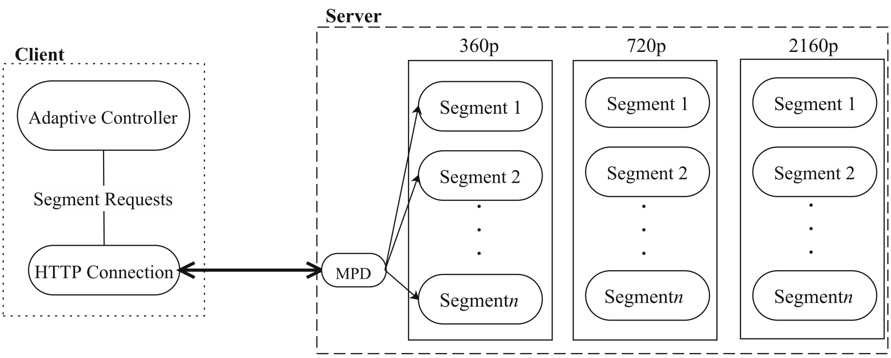


Fig. 9 Overview of media presentation description

3 Related works

Similar to our research, the authors [7] proposed a method which supported live streaming between different mechanisms over P2P. This technique could maintain high performance at a low overhead in highly network fluctuating environments.

Our approach is based on an adaptive streaming 3D multi-view [28] with an extension to a P2P network and multi-path streaming. In particular, they utilized the extra side information in view recovery of the media presentation description from dynamic adaptive streaming properties. As a result, they showed that 3D video quality is significantly better than in traditional streaming approaches.

In networking transmission, packet loss and latency may significantly degrade the multi-view video streaming experience. The authors [28] introduced an outstanding view recovery method for 3D, free-viewpoint, multimedia that is a cost-effective approach. First, the authors described the undelivered views as a result of adaptation in the network. Then they presented methods for recovering those views with high quality at the receiver side, using side information (SI) and the delivered frames of neighboring views. In the experiment, they evaluated their proposed adaptive 3D multi-view video streaming scheme using DASH. Their proposed adaptive technique significantly improved perceptual 3D video quality under adverse network conditions.

In another approach of adaptive streaming, the authors [29] presented an interactive, free-viewpoint video (FVV) streaming system based on DASH. They carried out a study on rate adaptation logic based on sampled rate–distortion (R–D) values. In particular, the research separately described two techniques. First, they presented the distortion of synthesized views to the bit rates of the texture. Second, they depicted depth components of the reference views. The two techniques are employed to maximize the quality of rendered virtual views. Their results showed that the proposed R–D-based rate adaptation strategy outperforms fair bit rate allocation among the reference streams components.

Regarding multi-view video combined with adaptive streaming, the authors [30] proposed a dynamic adaptive rate control system and its associated rate–distortion model for multi-view 3D video transmission. Their approach aims to improve the quality of the user experience in the face of varying network bandwidths. In the research, they exploited two techniques: high-efficiency video coding (HEVC) and DASH. They found that their proposed adaptive method has a higher quality of transmitted bitstream than non-dynamic adaptive rate control for given network bandwidth fluctuations.

In the field of 3D multimedia transmission, the authors [31] introduced an adaptive MVV streaming method. In particular, the approach addressed future generation 3D immersive MVV experiences with multi-view displays. The proposed method involved the calculation of low-overhead additional metadata that is delivered to a client. As a result, extensive evaluations, both objective and subjective, showed that the proposed method provides significant quality enhancement under adverse network conditions.

Similar to our approach, which employs the latest compression technology, HEVC, the authors [32] proposed an HEVC multi-view system using DASH. The system is supposed to overcome some challenges, such as network throughput and compatible video streaming with different devices. In fact, their system used an adaptive mechanism to adjust the video bit rate to the variations of bandwidth in the best effort networks. They also proposed a novel scalable method for handling multi-view videos and depth content for 3D videos with regard to the number of transmitted views. As a result, their method of transmitting MVV content could maximize the perceptual quality of virtual views after the rendering and hence increase QoE [33].

The authors [34] offered a good navigation quality to the different users as well as minimized the overall resource requirements of a streaming server to provide an adaptive streaming service. In the research, their proposed method preserved optimality in most of the 3D scenes while reducing the computational complexity of decoding and encoding processes. In the experiment, they simulated the proposed method with various classes of streaming users. The technique gained an optimal multi-view video representation selection that was better than that of the two recommended methods, which are a baseline representation selection algorithm and recommended representation sets. Indeed, the two compared methods use encoding parameters which decide a priority for all the views.

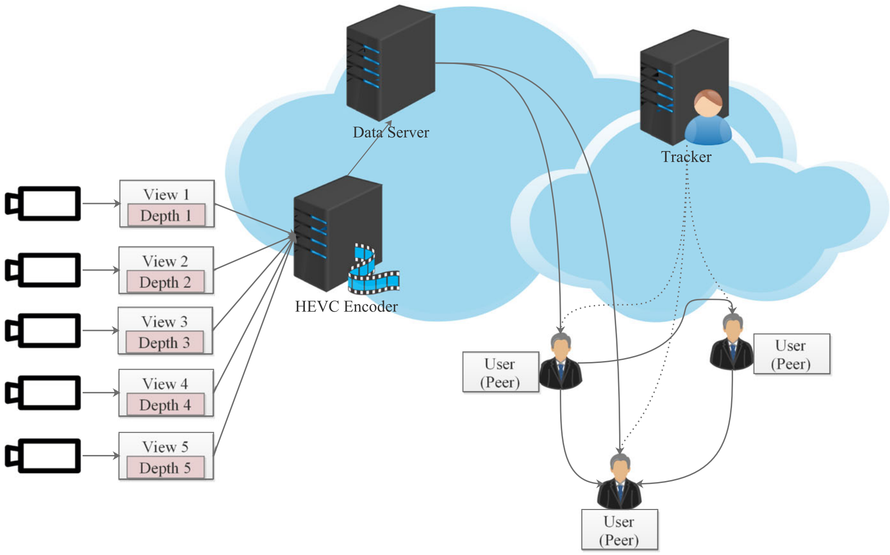


Fig. 10 HEVC-based multi-view P2P streaming overview multi-view P2P

4 System overview

We have four main components in the system. First, multi-camera records a scene from different views and depths. Second, the recorded data are transferred to encode and to minimize storage space before storing in data servers. Third, data servers represent an agent to deliver streaming data to a multi-user system using the peer-to-peer network. Finally, a tracker tracks the video streaming information of each participant in the P2P network. Figure 10 shows the system with the components described above.

As Fig. 11 shows, a set of CDNs deliver a video over HTTP (supposing that the video is available in all CDN of the set). Additionally, each streaming client is considered as a peer in a P2P network. Hence, a network connection from each peer to each CDN is different because they are from different networks. Consequently, there has to be a method to efficiently assist different peers working together or to maximize utilization of network resources. In the system, a peer downloads segment files while streaming with a CDN server. It stores those files temporally in its local file system and publishes the information in them to its manage tracker. If a new peer joins the P2P network, it first checks streaming metrics such as location, distance, and network latency to decide which peers or CDN can be connected to start the streaming. For instance, we have five peers with four stored files in each local peer and one file not yet available in their local storage, as shown in Fig. 11 with a white bar under five peers. If all the peers are near each other, the new peer will get each segment from each of the other peers. As a result, these peers could help CDN servers reduce traffic generating from the requests of peers.

We deliver the recorded video by employing two methods, as shown in Fig. 12. First, we use many data servers containing multiple parts of the content. Each part is

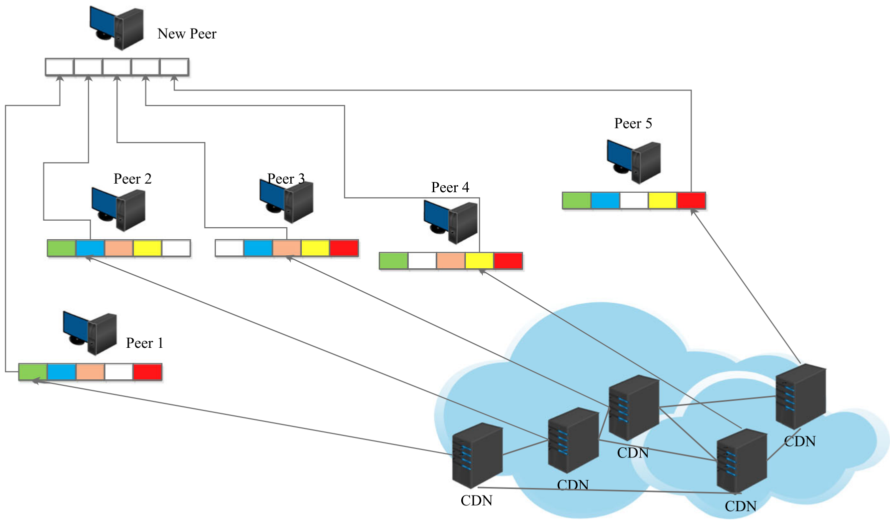


Fig. 11 Adaptive streaming over HTTP cooperating with P2P network

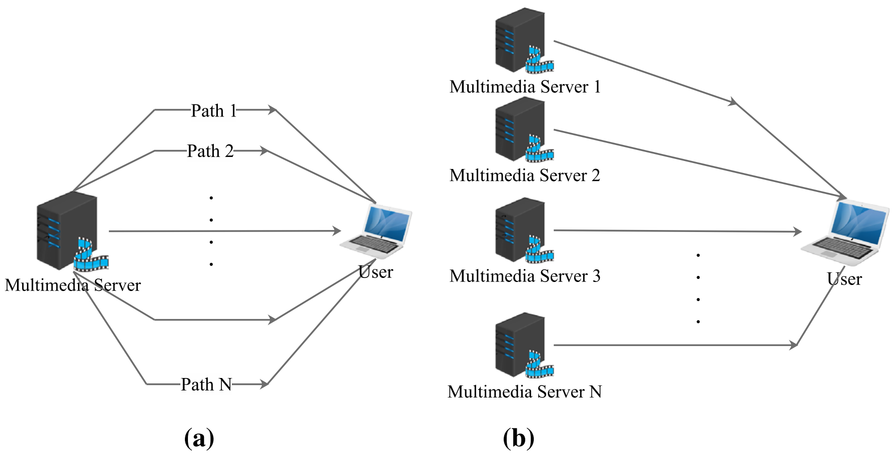


Fig. 12 Multi-path multimedia transmission

sent to a user via the multi-path transmission. For example, a peer in the P2P network can receive packets from different peers or CDN servers. Then a multimedia server can dispatch content by the multi-path transmission. These methods should ensure that more important data units have a high priority so that they will be sent on the high-quality path. In this way, we can guarantee that the streaming service is always available with the best quality.

The P2P system does not guarantee the quality of a streaming service, because video streaming applications require that packets arrive within strict constraints of timing and packet order. In addition, peers join and leave a P2P network randomly and frequently. These facts pose many problems in the effort to ensure a high or ultra-high quality of streaming service. First, there is the problem of finding a subset containing many peers that are nearest each other based on network metrics. Second is the problem of organizing the order of arriving packets within strict time constraints. Recall that the segment files in the adaptive streaming have video durations ranging from two seconds to ten seconds. In other words, the short duration could lead to reducing the transmission rate and vice versa. In P2P data transmission, we can consider each file as trunk data. Also, the P2P network might cause corrupted data, which is caused by a lack of some trunk data and the joining other trunks in the decoding process during playback at the client side. However, the adaptive streaming allows the client side to play the video seamlessly, with the indication of segments, which reduces problems with stalling and buffering.

We model the P2P network as follows. The P2P network with peers can be considered as a set in which we could exclude or add an element randomly. In particular, an element ith has several characteristics denoted as $E_i(d, t_c, n_c, f)$ with parameters described as follows. First, d is network metric distance such as bandwidth or latency. Second, t_c is the connection duration, measured since the peer has joined the network. Third, n_c is the number of connections a peer is making with other peers. Fourth, f is an array that contains available segment files in its local storage. Additionally, the P2P network is considered as a set E with the above-defined elements $E_i, i \in \mathbb{N}$. We define distance (Manhattan distance [35]) between elements as shown in (1).

$$d_{ij} = \sqrt{\sum_{k \in \{d, t_c, n_c, f\}} \left(\frac{E_i(k)}{\max_i E_i(k)} \right)^2}. \tag{1}$$

From the above-defined distance, we consider each peer as a point in a Voronoi region [36,37] to find a set of the nearest peers for a given peer. For example, Fig. 13 shows that we have eleven peers appearing as points. If any point belongs to the yellow region, it is considered the nearest neighbor of the point. Furthermore, a network administrator can define his distance to evaluate the system. Each definition can produce different research outcomes because it considers different network metrics.

We can generalize the Voronoi region concept as follows. The distance between a point p to a set S is defined as $d(p, S) = \inf\{d(p, s) \mid s \in S\}$ [37]. A Voronoi cell is considered as $R_i, i \in \mathbb{N}^+$. Each cell associates with a site $P_j, j \in \mathbb{N}^+$. Hence, the Voronoi diagram is the tuple of T_k in (2). In the summary of the Voronoi application for multi-path streaming, we briefly describe the neighbor discovery technique in P2P streaming by Algorithm 1.

$$T_k = \left\{ s \in S \mid d(s, P_k) \leq d(s, P_j) \text{ for all } j \neq k \right\}. \tag{2}$$

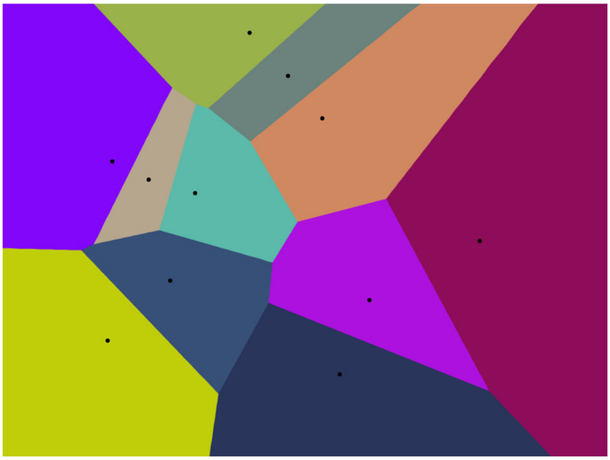


Fig. 13 Voronoi region to find the nearest neighbor in the P2P network

---

Algorithm 1: Voronoi Neighbor Discovery in P2P Streaming

---

while *not having a neighbor* **do**
    Calculate distance using (1);
    Aggregate Voronoi diagram (2);
    Initialize minimum distance for each site P_i, min_{ij};
    for *sites* P_i **do**
        for *regions* $R_j, i \neq j$ **do**
            if *distance* $(P_i, R_j) > min_{ij}$ **then**
                $min_{ij} = \text{distance } (P_i, R_j)$;
            end
        end
    end
end

---

5 Experiment and discussion

In this section, we first describe our virtualized system which is used to simulate our proposed method in multi-view video streaming. Subsequently, we present our experimental results and compare them to existing methods. Finally, we conclude this section with an analysis of our findings.

First of all, the management and orchestration (MANO) system is the heart of NFV that allows agile on-boarding and prevents chaos in the network virtualization field [38]. In our testing environment, we aggregate a set of physical computers by using Docker swam [8]. Then we create four virtualized components: HEVC encoder, data server, tracker, and user. Each component can have several instances. For example, we deploy the system (CDN streaming service in a P2P network) with one hundred users

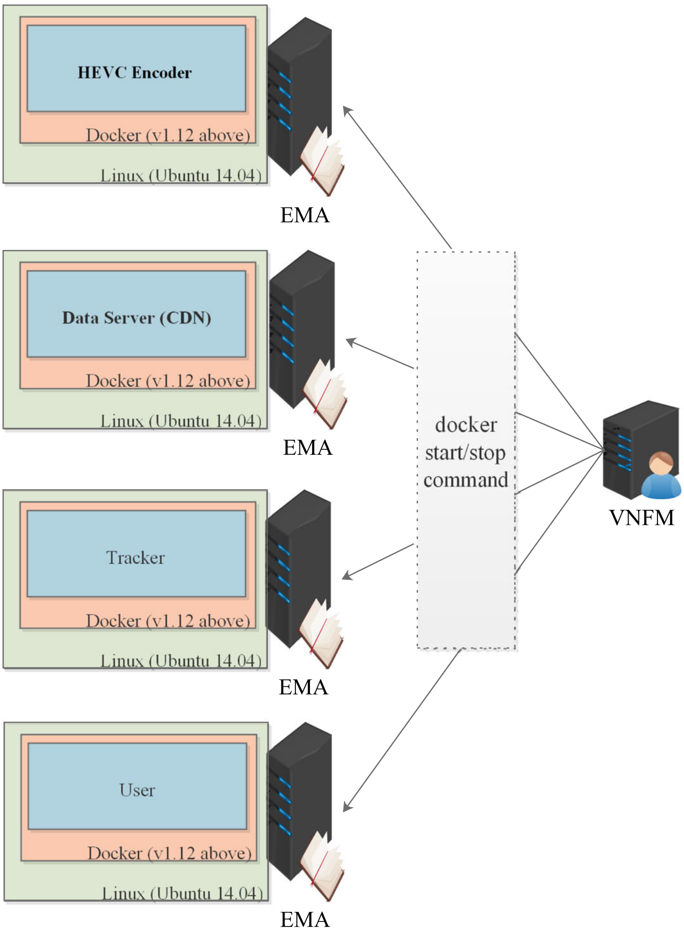


Fig. 14 Virtualization experiment system using MANO structure and Docker

who are peers, one tracker, and ten data servers. We implement adaptive streaming servers, delivery servers, and tracker servers based on Node.js [39]. In addition, we set up a cloud using Docker [40] to virtualize a network system and simulate delivery of multi-path video over the network. On the client side, we implement a web-based streaming application that is a hybrid content delivery network/P2P architecture for dynamic adaptive streaming over the hypertext transfer protocol. In detail, Fig. 14 shows our virtualized system in the experiment.

The overview of our experiment system is shown in Fig. 15 with six steps, as follows. We set up several cameras to record a scene from different views in a system that supports 3D and multi-view streaming. First, a file aggregation agent is employed to assemble files from different views. Second, the aggregation agent sends a request file upload to the management server which manages all actions of the whole system. Then the management server responds with an encoder server IP address. Third, the

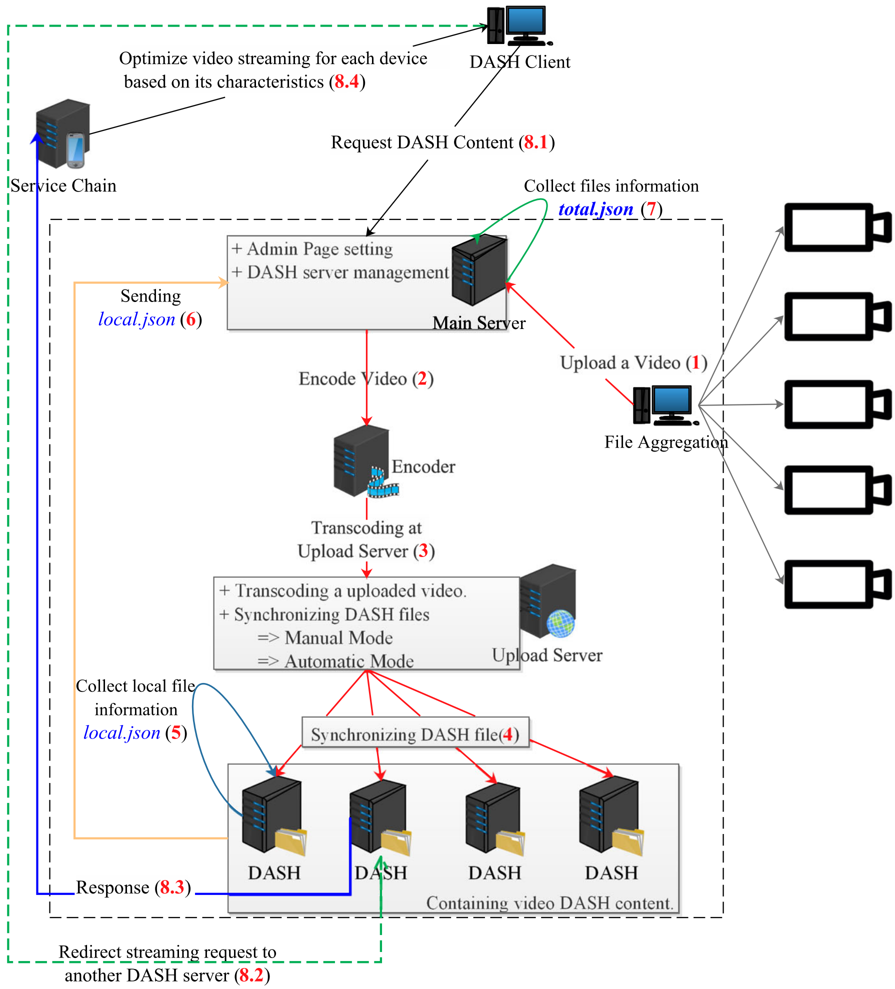


Fig. 15 A list of processes for dynamic adaptive streaming

file is encoded by using the latest encode/decode technology (HEVC). Subsequently, the encoded file is uploaded to an upload server where we transcode the video file into segments which support dynamic adaptive streaming over HTTP (DASH). Fourth, the transcoded files are synchronized to DASH servers in a manual or automatic mode. Functionally, a system administrator can control these modes. Furthermore, we implement four DASH severs to build a CDN service in which users can request content from the nearest CDN. Fifth, each DASH server collects its local information, such as available video streaming video in a JavaScript object notation (JSON) file and sends it to the management server. Sixth, the management server gathers all information from DASH servers and stores the information in a JSON file periodically [denoted as (7) in Fig. 15].

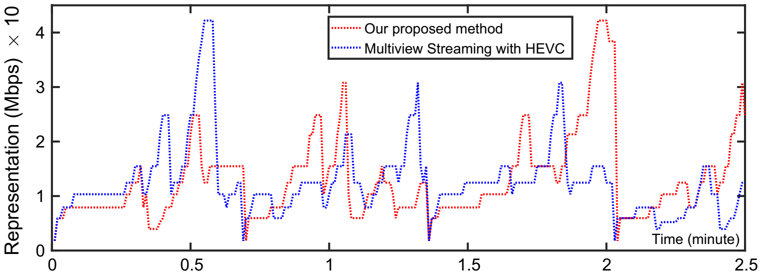


Fig. 16 Segment fetching in the P2P-based DASH streaming system

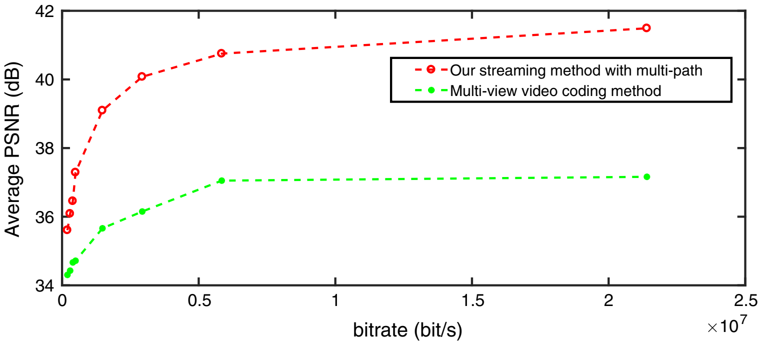


Fig. 17 Objective comparison of multi-view quality between our system streaming and multi-view video coding scheme

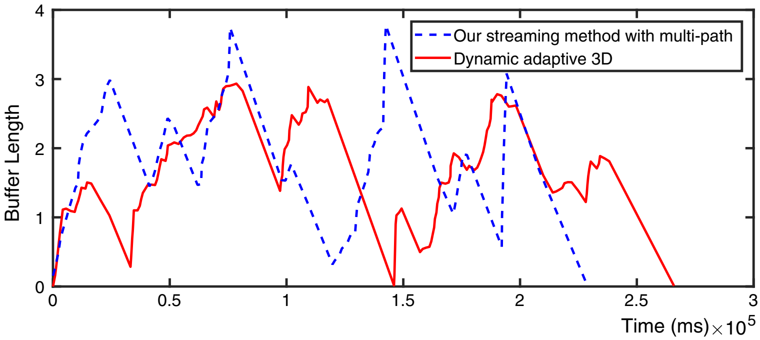


Fig. 18 Buffer state comparison between our proposed method and the approach in [28]

Regarding a streaming client, if it is a DASH client, we can consider it as a P2P participant request for streaming. It first makes a request to the management server. Then the server selects the best CDN server and redirects the request there. Also, we implement the CDN service that supports P2P streaming service and a service chain as well. Specifically, we place a streaming optimizer in the middle, between delivery servers and clients, which could substantially improve streaming quality.

We evaluate the system with a 3D video and compare the result with the streaming method in [28]. First, the segment fetching result is shown in Fig. 16 with the bandwidth fluctuation simulation from 3 to 40 Mbps. By applying multi-path transmission, our proposed method achieved a quality about 4% higher than the adaptive multi-view streaming method employing HEVC. As shown in Fig. 17, the peak signal-to-noise ratio (PSNR) measurement in our method is much higher than the base method for each bitrate {196,217, 295,360, 394,284, 493,986, 1,478,541, 2,934,266, 5,842,639 21,400,447} (bps).

In addition, we carry out one more experiment by simulating bandwidth fluctuation. Specifically, we compare our archived results with another method that has the same approach of adaptive streaming. By using a multi-view video with a duration of 2 min and 24 s, we obtain the comparison result shown in Fig. 18. As a result, our proposed method obtained a shorter duration (video playback) due to a higher level of buffer (about 20%) than the adaptive 3D method.

6 Conclusion

In this paper, we presented a study on adaptive-based HEVC with 3D multi-view streaming over a P2P network. First, we encoded a video before transmitting over the peer-to-peer network. Subsequently, we researched transmitting multi-view data over a multi-path channel. We observed that transmitting high-volume data over a multi-path channel provides streaming quality significantly better than traditional approaches, such as multi-view video encoding. In the future, we intend to investigate video quality evaluation methods that guarantee the quality of service as well as the quality of experience. More specifically, these methods will be used to evaluate transmission or encoding/decoding algorithms in the field of multimedia.

Acknowledgements This work was supported by 'The Cross-Ministry Giga KOREA Project' grant from the Ministry of Science, ICT and Future Planning, Republic of Korea (GK16P0100, Development of Tele Experience Service SW Platform based on Giga Media). Besides, this research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science, and Technology (MEST) (Grant No. NRF-2017R1D1A1B03034429). Furthermore, this research was supported by the Ministry of Science and ICT (MSIT), Korea, under the Information Technology Research Center (ITRC) support program (IITP-2018-2016-0-00314) supervised by the Institute for Information & communications Technology Promotion (IITP). Finally, this work was supported by Institute for Information & Communications Technology Promotion (IITP) Grant funded by the Korea government (MSIT). [2018-0-00691, Development of Autonomous Collaborative Swarm Intelligence Technologies for Disposable IoT Devices].

References

1. Khatamian A, Arabnia HR (2016) Survey on 3D surface reconstruction. J Inf Process Syst (JIPS) 12(3):338–357
2. Chakareski J (2013) Adaptive multiview video streaming: challenges and opportunities. IEEE Commun Mag 51(5):94–100
3. Basescu C, Reischuk RM, Szalachowski P, Perrig A, Zhang Y, Hsiao HC, Kubota A, Urakawa J (2015) Sibra: scalable internet bandwidth reservation architecture. arXiv preprint arXiv:1510.02696

4. Kang WM, Kim Mc, Jo BH, Park JH (2014) Hybrid transmission model based on thin client for efficient web platform service in media devices. J Converg (JoC) 5(3):37–42
5. Sullivan GJ, Ohm J, Han WJ, Wiegand T (2012) Overview of the high efficiency video coding (HEVC) standard. IEEE Trans Circuits Syst Video Technol 22(12):1649–1668
6. Migallón H, Galiano V, Piñol P, López-Granado O, Malumbres MP (2017) Distributed memory parallel approaches for HEVC encoder. J Supercomput 73(1):164–175
7. Wichtlhuber M, Richerzhagen B, Ruckert J, Hausheer D (2014) Transit: supporting transitions in peer-to-peer live video streaming. In: Networking Conference, 2014 IFIP. IEEE, pp 1–9
8. Naik N (2016) Building a virtual system of systems using Docker Swarm in multiple clouds. In: Systems engineering (ISSE), 2016 IEEE international symposium on, IEEE, pp 1–3
9. Fareghzadeh N, Seyyedi MA, Mohsenzadeh M (2018) Dynamic performance isolation management for cloud computing services. J Supercomput 74(1):417–455
10. Martin JP, Kandasamy A, Chandrasekaran K (2018) Exploring the support for high performance applications in the container runtime environment. Hum-Centric Comput Inf Sci (HCIS) 8(1):1
11. Manu A, Patel JK, Akhtar S, Agrawal V, Murthy KBS (2016) Docker container security via heuristics-based multilateral security-conceptual and pragmatic study. In: Circuit, Power and Computing Technologies (ICCPCT), 2016 International Conference on. IEEE, pp 1–14
12. Chamberlain R, Schommer J (2014) Using Docker to support reproducible research. https://doi.org/10.6084/m9.figshare.1101910
13. Imani M, Joudaki M, Arabnia HR, Mazhari N (2017) A survey on asynchronous quorum-based power saving protocols in multi-hop networks. J Inf Process Syst 13(6):1436–1458
14. Zhou Y, Fu TZ, Chiu DM (2015) A unifying model and analysis of P2P VoD replication and scheduling. IEEE/ACM Trans Netw (TON) 23(4):1163–1175
15. Degui Z, Geng Y (2014) Content distribution mechanism in mobile P2P network. J Netw 9(5):1229
16. Le Blond S, Choffnes D, Caldwell W, Druschel P, Merritt N (2015) Herd: a scalable, traffic analysis resistant anonymity network for VoIP systems. ACM SIGCOMM Comput Commun Rev ACM 45:639–652
17. Van Ma L, Kim J, Park S, Kim J, Jang J (2016) An efficient Session_Weight load balancing and scheduling methodology for high-quality telehealth care service based on WebRTC. J Supercomput 72(10):3909–3926
18. Sarif BA, Pourazad MT, Nasiopoulos P, Leung VC, Mohamed A (2015) Fairness scheme for energy efficient H. 264/AVC-based video sensor network. Hum-Centric Comput Inf Sci (HCIS) 5(1):7
19. Yan C, Zhang Y, Xu J, Dai F, Li L, Dai Q, Wu F (2014) A highly parallel framework for HEVC coding unit partitioning tree decision on many-core processors. IEEE Signal Process Lett 21(5):573–576
20. Sze V, Budagavi M, Sullivan GJ (2014) High efficiency video coding (HEVC). In: Integrated circuit and systems, algorithms and architectures, pp 1–375
21. Yan C, Zhang Y, Xu J, Dai F, Zhang J, Dai Q, Wu F (2014) Effin algorithm through inter-view RHEVC motion estimation on many-core processors. IEEE Trans Circuits Syst Video Technol 24(12):2077–2089
22. Vanne J, Viitanen M, Hämäläinen TD (2014) Efficient mode decision schemes for HEVC inter prediction. IEEE Trans Circuits Syst Video Technol 24(9):1579–1593
23. Pan Z, Zhang Y, Kwong S (2015) Efficient motion and disparity estimation optimization for low complexity multiview video coding. IEEE Trans Broadcast 61(2):166–176
24. Yeh CH, Li MF, Chen MJ, Chi MC, Huang XX, Chi HW (2014) Fast mode decision algorithm through inter-view rate–distortion prediction for multiview video coding system. IEEE Trans Ind Inf 10(1):594–603
25. Zeng H, Wang X, Cai C, Chen J, Zhang Y (2014) Fast multiview video coding using adaptive prediction structure and hierarchical mode decision. IEEE Trans Circuits Syst Video Technol 24(9):1566–1578
26. Ma LV, Park J, Nam J, Ryu H, Kim J (2017) A fuzzy-based adaptive streaming algorithm for reducing entropy rate of dash bitrate fluctuation to improve mobile quality of service. Entropy 19(9):477
27. Van Ma L, Park J, Nam J, Jang J, Kim J (2017) An efficient scheduling multimedia transcoding method for dash streaming in cloud environment. Clust Comput 1–11. https://doi.org/10.1007/s10586-017-1259-8
28. Ozcinar C, Ekmekcioglu E, Kondoz A (2013) Dynamic adaptive 3D multi-view video streaming over the internet. In: Proceedings of the 2013 ACM international workshop on immersive media experiences. ACM, pp 51–56

29. Hamza A, Hefeeda M (2014) A dash-based free viewpoint video streaming system. In: Proceedings of network and operating system support on digital audio and video workshop. ACM, p 55
30. Su T, Javadtalab A, Yassine A, Shirmohammadi S (2014) A dash-based 3D multi-view video rate control system. In: Signal Processing and Communication Systems (ICSPCS), 2014 8th International Conference on. IEEE, pp 1–6
31. Ozcinar C, Ekmekcioglu E, Ćalić J, Kondoz A (2016) Adaptive delivery of immersive 3D multi-view video over the internet. Multimed Tools Appl 75(20):12431–12461
32. Su T, Sobhani A, Yassine A, Shirmohammadi S, Javadtalab A (2016) A dash-based HEVC multi-view video streaming system. J Real-Time Image Process 12(2):329–342
33. Gupta R, Laghari K, Banville H, Falk TH (2016) Using affective brain–computer interfaces to characterize human influential factors for speech quality-of-experience perception modelling. Hum-Centric Comput Inf Sci 6(1):5
34. Toni L, Frossard P (2017) Optimal representations for adaptive streaming in interactive multi-view video systems. IEEE Trans Multimed 19:2775–2787
35. Torres-Sospedra J, Montoliu R, Trilles S, Belmonte Ó, Huerta J (2015) Comprehensive analysis of distance and similarity measures for wi-fi fingerprinting indoor positioning systems. Expert Syst Appl 42(23):9263–9278
36. Cai S, Pan H, Gao Z, Yao N, Sun Z (2014) Research of localization algorithm based on weighted Voronoi diagrams for wireless sensor network. EURASIP J Wirel Commun Netw 1:50
37. Aurenhammer F, Klein R, Lee D (2013) Voronoi diagrams and Delaunay triangulations. World Scientific Publishing Company. https://books.google.co.kr/books/about/Voronoi_Diagrams_and_Delaunay_Triangulat.html?id=Z849kgEACAAJ&source=kp_book_description&redir_esc=y. Accessed 10 Sept 2018
38. Huh JH, Seo K (2016) Design and test bed experiments of server operation system using virtualization technology. Hum-Centric Comput Inf Sci 6(1):1
39. Madsen M, Tip F, Lhoták O (2015) Static analysis of event-driven node. js JavaScript applications. In: ACM SIGPLAN notices, vol 50. ACM, pp 505–519
40. Merkel D (2014) Docker: lightweight linux containers for consistent development and deployment. Linux J 2014(239):2

Affiliations

Linh Van Ma[1] · Gwanghyun Yu[1] · Jin-Young Kim[1] · Yonggwan Won[1] · Jinsul Kim[1]

✉ Jinsul Kim
jsworld@jnu.ac.kr; jsworld@chonnam.ac.kr

Linh Van Ma
linh.mavan@gmail.com

Gwanghyun Yu
sayney1004@naver.com

Jin-Young Kim
beyondi@jnu.ac.kr

Yonggwan Won
ykwon@jnu.ac.kr

[1] School of Electronics and Computer Engineering, Chonnam National University, 77, Yongbong-ro, Buk-gu, Gwangju 500-757, Korea