IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 3, 2018, accepted October 18, 2018, date of publication October 29, 2018,
date of current version November 30, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2878519

Computing Offloading Over mmWave for Mobile VR: Make 360 Video Streaming Alive

TUAN THANH LE, DIEN VAN NGUYEN, (Student Member, IEEE), AND EUN-SEOK RYU, (Senior Member, IEEE)
Department of Computer Engineering, Gachon University, Seongnam 13120, South Korea

Corresponding author: Eun-Seok Ryu (esryu@gachon.ac.kr)

This work was supported in part by the Ministry of Science and ICT, South Korea, through the Information Technology Research Center Support Program, supervised by the Institute for Information and Communications Technology Promotion (IITP) under Grant IITP-2018-2017-0-01630, and in part by IITP through the Korea Government Development of Tiled Streaming Technology for High Quality VR Contents RealTime Service under Grant MSIP 2017-0-00307.

ABSTRACT Compared with previous video streaming technologies, 360 video streaming is an emerging technology in commercial platforms that promise greater immersive video service. The use of 360 video streaming requires larger bandwidth and rapid responsiveness to users. In addition, mobile virtual-reality (VR) devices locally process video decoding, post-processing, and rendering. However, its performance is insufficient to stream high-resolution videos such as 4k–8k. Therefore, we propose an adaptive computing offloading scheme using millimeter wave (mmWave) communication. This offloading scheme helps mobile device sharing video decoding tasks to a powerful GPU-based PC. It allows the mobile VR device enhancing the capability of playing high-resolution videos. The mmWave 802.11ad wireless technology promises the use of high-bandwidth wireless communication to improve the capacity of multimedia systems. However, the current studies on video transmission over 802.11ad mmWave are specifically focused on supporting the outdoor environment. Furthermore, synchronization is an essential issue for ensuring the quality of service (QoS) of 360 video transmission. Thus, the proposed scheme concentrates on two parts: 1) 360 video streaming in an indoor environment, and offloading mechanism using mmWave 802.11ad 60-GHz wireless communication to offload video coding tasks and 2) adaptive video synchronization mechanisms, and high-efficiency video coding (HEVC) using scalable HEVC extension for 360 videos with equirectangular projection. The experiments prove that the proposed scheme provides high performance 360 video streaming in QoS-sensitive video streaming applications.

INDEX TERMS 360 video VR, video streaming, SHVC, HEVC, mmWave, offloading.

I. INTRODUCTION

Currently, 360 video VR is the most interesting streaming service offered by video streaming systems. The compressed video transmission through 2.4/5 GHz wireless networks might not be reasonable for certain latency sensitive applications such as high-resolution video service or wireless VR. This is particularly challenging to ensure QoS can play high-resolution real-time video. Since 4K-8K video streaming services require much larger network capacity than previous high-definition video services did, so it increases the complexity of the transmission system to maintain the quality of the 360 video stream. Due to the varying requirements of the large spectrum of multimedia services, mmWave wireless technology is the most relevant technology for fulfilling the current specifications for the high-speed multimedia system. Recently, mmWave 802.11ad 60GHz has attracted attention as it can provide adjacent bandwidth up to 7GHz. Therefore, it has great capability for meeting the fast growing demand for the capacity of 5G network systems.

Recently, video streaming systems utilize an effective video coding technology, which was proposed standard in JCT-VC [1], and its standard name is HEVC. HEVC coding technology produces twice as much bandwidth as H.264/AVC coding technology. Otherwise stated, HEVC video coding can double the capacity of bandwidth to transmit the video content at the same data rate when compared to H.264/AVC. Figure 1 illustrates the conceptual architecture of our proposed system. The proposed system includes a 360 video server, powerful PC, and mobile VR based head-mounted device (HMD). The mobile

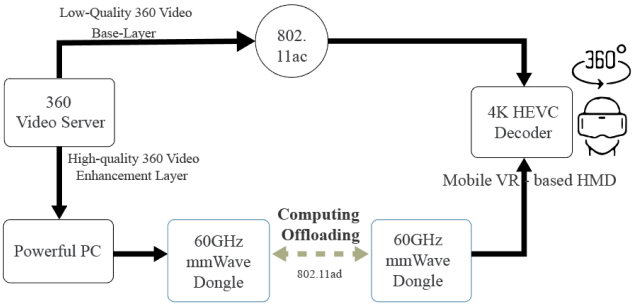


FIGURE 1. Conceptual diagram 360 video computing offloading.

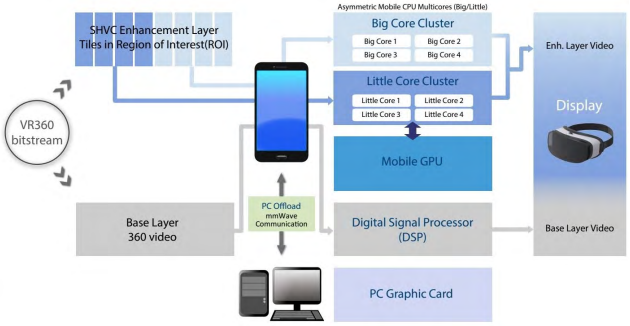


FIGURE 2. The conceptual architecture of the MVP project.

VR device supports 4K HEVC software/hardware decoders. In an effort to enhance video streaming services, our goal was to make 360 video VR look even more reasonable in high-bandwidth mmWave network systems.

Current studies on video transmission through mmWave link are primarily focused on supporting the outdoor environment. The support for the mmWave indoor environment has not been properly considered yet. Additionally, the use of SHVC coding also has not been addressed accordingly its necessity for 360 video VR on the mobile device. Moreover, mobile device VRs can handle local video coding and post-processing tasks, but its performance is insufficient to stream 4K-8K resolution videos. Thus, this article presents a more appropriate scheme to handle 360-degree video streams on mobile VR using 802.11ad 60GHz wireless communication. In order to perform 360 video streaming, we design a practical system based on the proposed scheme and demonstrate how to deploy computing offloading scheme. Our contribution consists of ideas as follows:

- Use of experimental tests of mmWave 802.11ad 60GHz wireless communication in an indoor environment to get optimal parameters for 360 video streaming.
- Verification of 360 videos streaming on mobile VR device using scalable video coding.
- Use of computing offloading scheme via a mmWave 802.11ad 60GHz wireless link in order to share video coding task.
- Use of synchronization mechanisms to handle real-time video transmission without packet lost.

The rest of the paper is organized as follows: section II describes current studies that related to aspects of proposed scheme. Section III addresses the challenges in 360 video streaming at high-resolution, and presents the computing offloading scheme in detail. Section IV provides the testbest and experimental scenarios and section V shows the performance evaluation. Finally, section VI presents conclusion related to proposed scheme and our future work.

II. RELATED WORK

A. PREVIOUS WORK FOR VIDEO STREAMING SYSTEM

The main objective of the Gachon University merciless video processing (MVP) project is to contribute the high-quality video service that can be approached in a limited performance of mobile VR. Figure 2 illustrates the architecture of the MVP. The region of interesting (ROI) is decoded using parallel processing according to mobile asymmetric big/little cores [2]. Viewport extraction and motion-constrained tile for mobile VR streaming were detailed as in [3]. The implementation of the proposed system is part of the MVP project and it is applicable to mobile VR as an extended version of the project in [4] and [5].

B. 802.11AD MILLIMETERWAVE FOR INDOOR ENVIRONMENTS

Nowadays, wireless networks are facing high demand for interacting data as customers progressively utilize mobile phone, tablet, and other smart devices to share high-definition multimedia content. In recent times, the use of high frequency higher up 24 GHz, also known as millimeter wave bands, mmWave is emergence as a key factor for enhancing communication in 5G wireless networks. The analyzed details of mmWave can be reviewed in [6]. Firstly, our goal is to verify the real-world efficiency of an indoor environment on the deployment of mmWave communication technology. While state-of-the-art studies have not yet directly addressed this issue, there have been various related papers [7]–[10], that have provided us with sufficient details to draw significant conclusions. These researches encourage us to focus the efforts on the deployment of indoor mmWave networks. The results in [11] and [12] provide useful data regarding object blockage that affects to the performance of mmWave link, and peer-to-peer indoor mmWave communication scenarios. These researches also verify which features are fundamental for the network designer to consider based on their intended clients.

Although there have been important insights gleaned regarding the impact of the affected factors, there is still little understanding of the operation of the 802.11ad 60GHz device in real-world situations. Therefore, a testbed for mmWave systems is provided for large indoor environments using 802.11ad mmWave products obtained from Dongle MLWGU3V2-D [13] and Tensorcom 3.0 TC60G-USB3.0 EVK [14]. Table 1 shows the basic specifications of the antenna of the Dongle MLWGU3V2-D device.

TABLE 1. mmWave antenna specifications.

Parameters	Type	Unit
Radiation type	Enfire	
Polarization	Linear	
2Tx Array Peak Gain	10	dBi
2Rx Array Peak Gain	10	dBi
3dB Beamwidth	110	degree

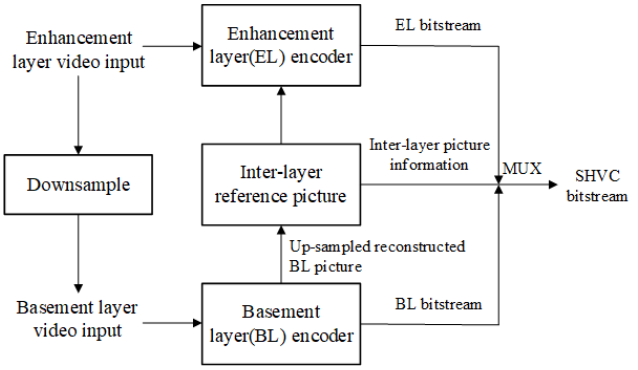


FIGURE 3. SHVC encoder architecture.

C. VIDEO CODING WITH HEVC-SHVC

Recently, HEVC (H.265 / MPEG-H part2) is an emerging technology in the video coding field. Therefore, in order to support HEVC, video service platforms are upgrading to support HEVC coding. The first version of HEVC achieved an approximately 50% bitrate reduction compared to its predecessor H.264 AVC with equivalent subjective quality [15]. The second version of the HEVC [16] includes the SHVC extension, format range extension, and multi-view extensions. The specifications of HEVC and SHVC extension was analyzed in [17] and [18]. The SHVC extension brings a choice of various resolution levels of decoded videos within unique input encoded bitstream. In an effort to improve streaming capabilities, [19]–[21] proposed advanced solutions. Based on these solutions, our work also looks at the factors that affect the quality of 360 videos streaming to mobile devices.

As shown in Figure 3, SHVC provides an adaptive scheme to support video encoding in multiple layers. Where each layer represents a different quality representation of the same video scene. Base-layer (BL) is of the lowest quality representation, and one enhancement-layer (EL) can be encoded by referencing to the lower layers to produce one encoded bitstream. In other words, client-side can receives various resolutions of decoded video from an input SHVC bitstream. Therefore, to improve video transmission capability, instead of using a single HEVC coder, the proposed scheme utilizes the SHVC coder as illustrated in Figure 4.

D. ENTROPY CODING

In the field of video technology, video data compression is one of the most important steps toward reducing the size of video content. In the area of data compression, the entropy

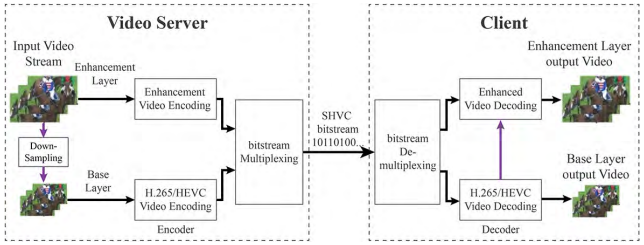


FIGURE 4. Video streaming processes using SHVC.

TABLE 2. Benchmarking of entropy coders.

Benchmarked file	Silesia compression corpus		
Algorithm	Ratio	Compress	Decompress
fse	2.778	345MB/s	490MB/s
fse-zstd	2.882	470MB/s	1160MB/s
zlibh	2.746	259MB/s	275MB/s

coding stage is generally the last stage of an algorithm where the benefits from the model are actualized. Currently, in order to solve the Shannon limit [22] in video coding area, based on prior research [23], we have a new trend for improving data compression using asymmetric numeral systems (ANS) theory [24]. The currently implemented version of ANS achieves approximately 50% faster decoding than the Huffman version and achieves rates similar to those in arithmetic encoding. This optimized entropy coder is called the Finite State Entropy (FSE) coder. In the proposed system, we applied an advanced version of the FSE coder, which was implemented by Facebook. Table 2 shows the benchmarking results for the FSE coder of Facebook (zstd) [25], a standard FSE coder and Huffman entropy coder (zlibh) on a platform Core i7-7700K 4.2GHz processor and Linux Ubuntu 64-bits gcc 6.3 OS using the lzbench tool [26].

III. 360 VIDEO STREAMING OVER mmWave - COMPUTING OFFLOADING

For the purpose of enhancing 360 video streaming on a mobile VR device, we firstly implemented computing offloading using an 802.11ac wireless communication. However, the throughput of a connected link was just around 70-80 Mbps. Therefore, our first target is to evolve to improve the bandwidth of network communication. Recently, various studies have analyzed the capacity of 802.11ad 60Ghz communication. Thus, we launched the offloading project with mmWave products, which are available on the commercial market now.

In order to facilitate high-resolution video streaming in indoor environments, the 802.11ad 60GHz device was considered to provide high-bandwidth video transmission. Additionally, effective synchronization mechanisms are necessary to adapt the real-time issue in high-bandwidth video transmission. Moreover, in video server-client models, mobile VR devices always encounters certain problems, including overflow or high-power consumption, when

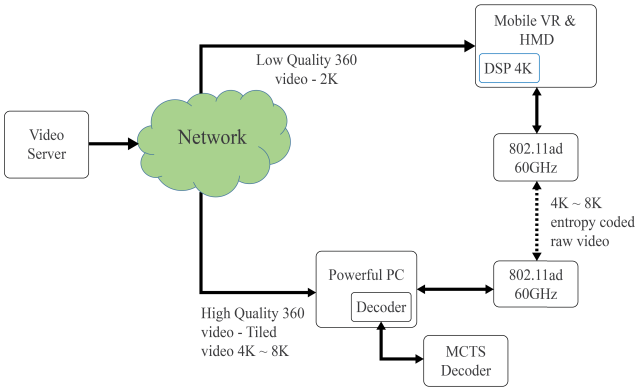


FIGURE 5. Proposed system for computing offloading.

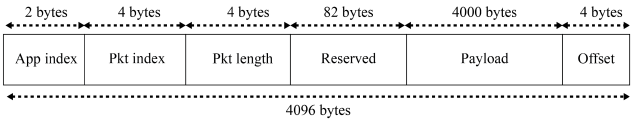


FIGURE 6. mmWave packet structure of streaming application.

they play high-resolution video content. Specifically, mobile phones utilizing next-generation video technologies, such as AR/VR or 360 videos, typically encounter poor performance. Therefore, in this section, we propose an offloading scheme and other essential mechanisms in order to ensure QoS for high-resolution video streaming.

According to the idea above, an implementation of the proposed system is essential for demonstrating how to deploy the proposed scheme in fact. The conceptual architecture of the proposed system was designed as shown in Figure 5. We focus on computing offload and synchronization between mobile device and PC based on the quality of a selected video stream. The low-quality video is encoded as base layer video and the high-quality video is encoded as an enhancement layer video. These layers are represented as HEVC tiles in the viewport. Motion constraint tile sets (MCTS) [27], [28] coding is also an objective topic in the field of video coding. In order to enhance the proposed system, powerful PC and video server are grouped into a single entity. This grouping was considered to be the best choice for the purposes of our implementation and experimental tests.

A. DATA PACKET STRUCTURE

As shown in Figure 6, in order to ensure data transmission via a mmWave link, the packet structure of video streaming applications had to be re-designed at the application layer. The designed data packet consists of 'app index' and 'Pkt index' fields to handle packet loss issue and real-time transmission through the use of synchronization mechanisms. The 'Pkt length' is the optimal value following the packet size and video resolution. Payload field contains the raw data of video. The last 'offset' 4 bytes is for ensuring communication between a streaming application and mmWave device. The 'reserved' portion includes reserved bytes to improve mmWave communication. In some cases, we use more bytes from the 'reserved' part to expand other fields based on certain aims. For example, we can add 4 bytes to the 'Pkt index' field, and the new 'Pkt index' field with 8 bytes can index 2^{64} packets compared to 2^{32} packets when it is of a size of 4 bytes. Furthermore, we also utilize this packet structure to build the acknowledgment (ACK) packet, which was applicable in synchronization mechanisms. The necessity for the new format arises from some issues of USB connection between hosts and commercial 802.11ad devices as follows:

- Mobile VR device and PC exchange packets to the mmWave device through USB ports. Currently, streaming application cannot deeply access to the firmware of 802.11ad 60GHz device in order to modify its internal functionalities.
- Streaming application can send control packets to the mmWave device so as to configure mmWave firmware within a given parameter set. Furthermore, we can get reply packets from firmware back to the hosts. These reply packets are particularly useful for confirmation or collecting statistical details such as SNR, PER, and packet loss.

B. COMPUTING OFFLOADING SCHEME

In the current streaming system, mobile device gets full SHVC coded sequence from the video server, then it locally proceeds video tasks (such as video decoding, post-processing and rendering). Mobile devices require a lot of resources to perform properly high-resolution video processing. In order to face this challenge, our target is offloading its decoding and post-processing stages to PC. Hence, the mobile device only produces rendering output decoded video on HMD. However, this offloading requires larger bandwidth than the 2.4GHz/5GHz wireless networks can provide. As mentioned above, we verified offloading using the 802.11ac wireless network, but it only provides offload transmission at low datarate approximate 80 Mbps. Thus, we implemented offloading over 802.11ad 60Ghz communication to adapt to high-bandwidth demand. Moreover, the bitstream separation was proposed for enhancing the flexibility of mobile VR device in playing various resolution videos. The proposed scheme includes an offloading mechanism to share enhancement layer video decoding between mobile VR and PC as shown in Figure 7. Furthermore, the ''option'' part is an as an extended part for loading encoded video from the mobile device to PC. In other words, the ''option'' part allows mobile device can transfer the enhancement-layer video bitstream to PC through 802.11ad 60GHz communication. If the PC cannot receive the enhancement-layer bitstream from the video server, it can get this sequence from the mobile device.

The offloading procedure can be described as follows: (1) The video server processes input SHVC encoded sequence into two or more new encoded sequences. Then, video server sends low-quality video sequence (BL bitstream) to mobile VR device. After that, the mobile device

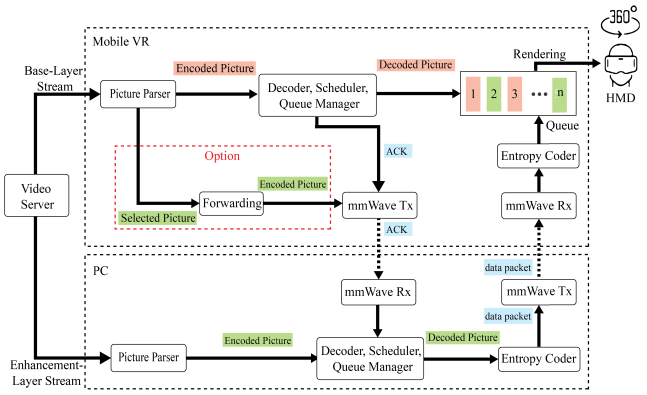


FIGURE 7. Computing offloading scheme.

locally processes BL bitstream, consists of video decoding, post-processing and reconstructing. These stages require low-loading, and the rest of the resources is the idle resource that can be saved for other tasks; (2) The high-quality video sequences (enhancement-layer bitstreams) are sent to the PC to decode and pass post-processing. Then, the output video will be transferred to mobile device via 802.11ad 60GHz communication. Because PC handles a large amount of data such as 4K-8K video data, its tasks incur a high-load, leaving the mobile device free to perform other tasks. Then we need a powerful PC that can process the high-load of ELs decoding tasks. This work of powerful PC is commonly referred to as computing offloading. Thus, the mobile device only launches video rendering task. We can confirm that computing offloading can support mobile device enhancing the capability of playing 4K or higher resolution videos. Additionally, the FSE entropy coder is also embedded in the system for decreasing the size of raw EL video. The compression of raw video may result in a maximum gain of approximately 40% bandwidth capacity for the offloading process.

C. SYNCHRONIZATION MECHANISMS

In order to ensure the QoS of video content delivery, especially in mmWave systems, the synchronization mechanism is the most important component. In our system, we provide two synchronization mechanisms so as to ensure performance based on pre-determined requirements. This is because of the features of commercial mmWave products of Dongle or Tensorcom, which only provide a UDP connection. As shown in algorithms 1 and 2, synchronization mechanisms were combined with computing offloading in order to support video transmission according to various strategies.

1) FIRST SYNCHRONIZATION MECHANISM

Figure 8 presents the first synchronization mechanism, which can be detailed as follows:

1) Every data packet accommodates an indexing number to identify different packet.
2) Mobile VR device sequentially receives data packets from PC via wireless 802.11ad communication.

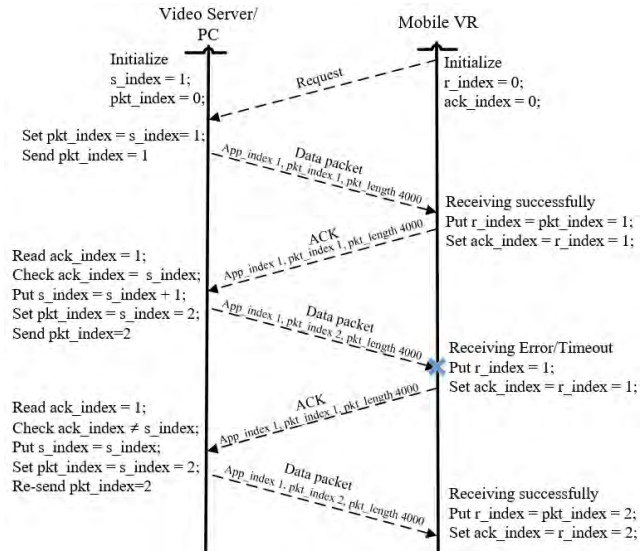


FIGURE 8. First synchronization mechanism.

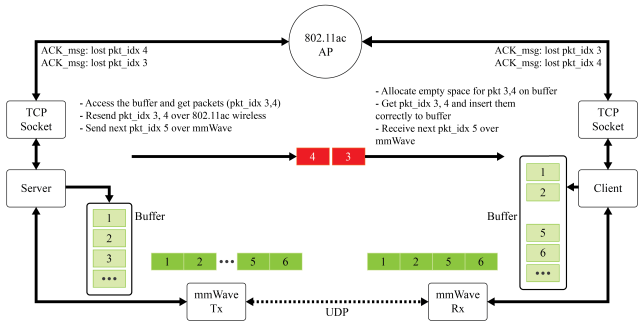


FIGURE 9. Second synchronization mechanism.

3) Mobile VR device builds ACK packet according to indexing number from the data packet. Then, it replies ACK packet back to the PC.
4) After PC received ACK packet, it processes 'pkt_index' confirmation of the ACK packet. Then PC decides to forward next packet to mobile VR device based on the status of confirmation "success". Otherwise, PC will re-send the current packet until it succeeds.

2) SECOND SYNCHRONIZATION MECHANISM

Figure 9 presents the second synchronization mechanism as follows:

1) Every data packet accommodates an indexing number for identification.
2) mobile VR device sequentially receives data packets from PC via wireless 802.11ad UDP communication, and PC temporarily stores transmitted packets on the buffer.
3) Mobile VR device builds ACK packet according to indexing number from the data packet. Then, it replies ACK packet back to the PC via an 802.11ac wireless network.

Algorithm 1 Computing Offloading and First Synchronization Mechanism

Data: Input SHVC bitstream
- N: number of coded pictures inside bitstream;
Init: Initialization
- Picture parser (PP); Max buffer size (MAXB); SHVC decoder; Decoded picture buffer (DPB); Data buffer (DB); Synchronizer with timer; ;
Main loop: **while** $i <$ *(Max of N)* **do**
  1. Load SHVC bitstream using PP to detach encoded picture n_i, and check that it belongs to BL or ELs;
  2. If picture n_i belongs to BL, write it to DB;
  3. If picture n_i belongs to EL, SHVC decoder process it and write output to DPB;
  4. Synchronizer builds data packet using picture n_i following new design of mmWave data packet. The raw data of picture n_i becomes a payload of new data packets;
  5.**while** *(j * 4000 bytes) < sizeof(picture n_i)* **do**
    6. Send the new data packet j to mobile VR device;
    7. Wait for ACK from mobile VR, check ''pkt index'' and decide between retransmission of packet j or moving to the next packet $(j + 1)$;
  end
  8. Open display picture from DPB and open decoded picture ni on the client side;
  9. **if** *(sizeof(DB) > MAXB or sizeof(DPB) > MAXB)* **then**
    Release packet data n_0 to n_i from DB and DPB;
  else
    Move to next picture n_{i+1};
  end
end
Result: Decoded Video

Algorithm 2 Computing Offloading and Second Synchronization Mechanism

Data: Input SHVC bitstream
- N: number of coded pictures inside bitstream;
Init: Initialization
- Picture parser (PP); Max buffer size (MAXB); SHVC decoder; Decoded picture buffer (DPB); Data buffer (DB); Synchronizer with timer; ;
Main loop: **while** $i <$ *(Max of N)* **do**
  1. Load SHVC bitstream using PP to detach coded picture n_i, and check that it belongs to BL or ELs;
  2. If picture n_i belongs to BL, write it to DB;
  3. If picture n_i belongs to EL, SHVC decoder process it and write output to DPB;
  4. Synchronizer builds data packet using picture n_i following new design of mmWave data packet. The raw data of picture n_i becomes a payload of new data packets;
  5.**while** *(j * 4000 bytes) < sizeof(picture n_i)* **do**
    6. Send the new data packet j to mobile VR device;
    7. Write packet j to file buffer;
  end
  8. Open display picture from DPB and open decoded picture n_i on the client side;
  9. **if** *(sizeof(DB) > MAXB or sizeof(DPB) > MAXB)* **then**
    Release packet data n_0 to n_i from DB and DPB;
  else
    Move to next picture n_{i+1};
  end
end
Second loop: **while** *TCP socket thread running* **do**
  10. TCP socket listen for ACK from mobile VR;
  11. Get the ACK packet with index m from mobile VR, then check ''pkt index m'';
  12. Access to file buffer, get data packet within index m and resend data packet m over 802.11ac network;
end
Result: Decoded Video

4) After confirmation of the ACK by PC, PC will resend lost packets to mobile VR device.
5) Mobile VR device receives the packets from a TCP socket and inserts them correctly into the buffer;
6) Although stages 1-5 are all processing stages, the UDP connection always exchanges raw video data from PC to mobile VR device.

IV. SETTING UP A TESTBED

A. SETTING UP HARDWARE AND SOFTWARE

The proposed system may significantly demonstrate real performance when transmitting video content, compared to mmWave network in simulations. In order to analyze the offloading scheme, we set up a testbed as illustrated in Figure 10. Additionally, we also made a real-world demo of the proposed system as in [29]. We set up a PC with a Core i7-7700 4.2GHz processor, GPU Nvidia GeForce 1080, and 32 GB of memory with Linux Ubuntu 64-bits gcc 6.3 OS. Mobile VR device was set up to consists of Samsung Gear VR and a Samsung Galaxy S7 phone. Additionally, we installed an 802.11 access point, which was used to handle synchronization in some of our configurations. We also configure out the powerful PC as a fully functional video server. Two Dongle *MLWGU3V2-D* devices were used to set up 802.11ad 60GHz communication. Table 3 provides the basic parameters of a testing system. Table 4 provides in detail the standard 4K 360 video test sequences by JCT-VC as shown in Figure 11.

Regarding software, we used HM software with 360 libraries [30] as a video encoder and openHEVC

TABLE 3. System parameters.

Parameters	Value
mmWave MCSL Level	$1 \rightarrow 7$ (level 7: physical rate 1.9 Gbps)
mmWave Antenna Sector	$0 \rightarrow 15$
Frequency Channel	$59.40 \rightarrow 61.56GHz$
Distance	$0 \rightarrow 3$ meters
Video Test Duration	$10 \rightarrow 20$ seconds
SHVC testsequences	DrivingInCity(3840×1920_1920×1080); KiteFlite(4096×2048_1920×1080); Harbor(4096 × 2048_1920 × 1080); GasLamp(4096 × 2048_1920 × 1080); Trolley(4096×2048_1920×1080)

software [31] as a video decoder. The entropy coder was the Facebook FSE coder. Besides that, FFmpeg [32] and equirectangular projection Peak Signal-to-Noise Ratio (PSNR) software [33] were used as evaluation tools. In order to produce BL and EL bitstreams, HM and its scalable extension were applied to process a spherical 360 video with the equirectangular projection (ERP) [34]. The ERP projection in detail was described in table 5. Additionally, coding configurations of BL and EL bitstream are also detailed in table 6 and 7.

360 videos allow the user to leave the fixed center position in the view space displayed in the HMD [35]. Unlike conventional video, the commonly used format of input 360 videos converted to HMD is an equirectangular plane that mapped from the sphere. Instead of calculating the PSNR in the rectangular plane to perform the quality of 360 videos, we can compute the distortion in the sphere to properly respond to the way we watch 360 videos. In here, we used a method called weighted in sphere PSNR (WS-PSNR) [35]. WS-PSNR is a weighted metric to compute distortion of reconstructed 360 videos in the spherical domain to show the difference between reconstructed video and original 360 videos. ERP WS-PSNR software [33] supports WS-PSNR computation for full ERP and windowed ERP (a part of ERP) as shown in Figure 12. The deformation of each frame is the error accumulation of all pixels. Errors at each pixel location on the projection format are calculated according to the spherical area corresponding to the function of the mapping between the format and the sphere. The weighted metric on position (i: longitude coordinate, j: latitude coordinate) for full ERP is calculated as:

$$w(i,j) = \cos(((j + 0.5 - N/2) \times \pi)/N). \quad (1)$$

where N is the height of full ERP. For windowed ERP, the weights equal to the weights of the corresponding location on full ERP.

B. TESTBED SCENARIO

The experiments of the proposed system focused on EL video bitstreams. This means the testbed focus on the decoding original SHVC bitstream and video post-processing on PC. Then, output video in YUV format (4:2:0) was sent to mobile VR device through the 802.11ad wireless link. Mobile VR devices can play low-quality (BL) video or render high-quality (EL) YUV video according to its selection. We set a scenario as stages as follows:

1) UDP 802.11ad only: Take experiments without synchronization mechanisms or entropy coder. We perform this stage in order to show the capability of mmWave communication for data transmission. Based on the results, we can optimize the parameters of the proposed system for testing in the second stage and third stage.
2) mmWave ACK: UDP 802.11ad, first synchronization mechanism, and FSE entropy coder were used.
3) TCP ACK: UDP 802.11ad, second synchronization mechanism, and FSE entropy coder were applied to stream 360 video.

TABLE 4. JCT-VC 360 videos bitstreams.

Parameters	Value
InputFile	DrivingInCity_3840 × 1920_30fps_8bit_420_$erp.yuv$
InputBitDepth	8 bits
InputChroFormat	420
FrameRate	30 fps
FrameSkip	0
Width	3840
Height	1920
Frames	300
Level	5.2
InputFile	KiteFlite_4096 × 2048_30fps_8bit_420_$erp.yuv$
InputBitDepth	8
InputChromaFormat	420
FrameRate	60
FrameSkip	0
SourceWidth	4096
SourceHeight	2048
FramesToBeEncoded	300
Level	5.2
InputFile	Harbor_4096 × 2048_30fps_8bit_420_$erp.yuv$
InputBitDepth	8
InputChromaFormat	420
FrameRate	30
FrameSkip	0
SourceWidth	4096
SourceHeight	2048
FramesToBeEncoded	300
Level	5.2
InputFile	GasLamp_4096 × 2048_30fps_8bit_420_$erp.yuv$
InputBitDepth	8
InputChromaFormat	420
FrameRate	30
FrameSkip	0
SourceWidth	4096
SourceHeight	2048
FramesToBeEncoded	300
Level	5.2
InputFile	Trolley_4096 × 2048_30fps_8bit_420_$erp.yuv$
InputBitDepth	8
InputChromaFormat	420
FrameRate	30
FrameSkip	0
SourceWidth	4096
SourceHeight	2048
FramesToBeEncoded	300
Level	5.2

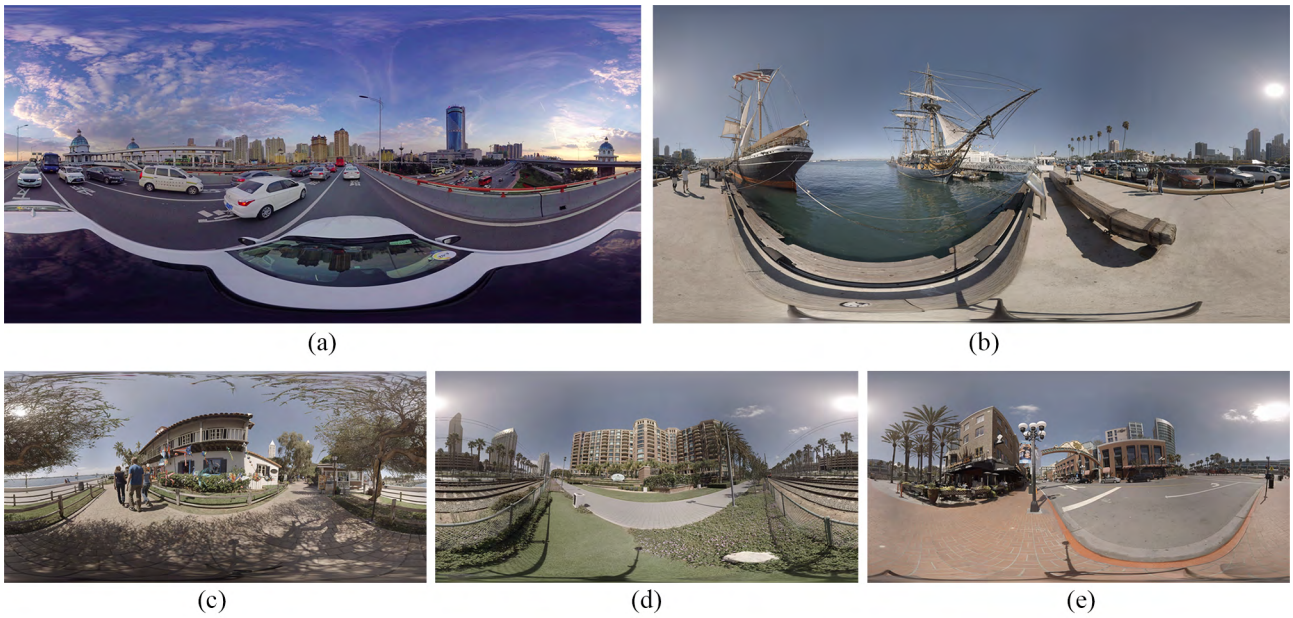


FIGURE 11. JCT-VC test sequences (a) DrivingInCity_3920x1080 (b) Harbor_4096x2048 (c) KiteFlite_4096x2048 (d) Trolley_4096x2048 (e) GasLamp_4096x2048.

V. PERFORMANCE EVALUATION

In order to evaluate the performance of the proposed system, experiments was performed various scenarios to verify some aspects. All tools and software for video coding were used to evaluate performance according to [30]–[33].

A. UDP 802.11AD ONLY

As shown in table 8, based on the distance between mobile device and PC, end-to-end throughput of the mmWave UDP connection varied from 500 Mbps to 930 Mbps. The experiments proved that the throughput decreased to near zero at distances over10 m. Thus, the adaptive distance for 360 video streaming was set in the range of zero to three meters for several reasons: 1) A mobile VR device can change the location as around the PC or just change the viewing direction. 2) This range yields the highest and stable throughput. 3) This range is a satisfactory distance from the human's face and PC in an indoor environment. In distances from 3 meters to10 meters, the throughput of mmWave link seriously decreased.

Figure 13 shows the effect of different obstacles when they are located between mobile VR and PC. Additionally, this experiment confirms that if the obstacle is in a fixed position, the throughput will drop to near zero. At the moment 10 seconds the human's head moves through the LOS area, this leads to throughput rapidly drop to zero. Therefore, we set up obstacles that only move through the video streaming area in a short time. The recovery from obstacle issue will be an engaging research of mmWave communication in the future. The results of the experiments show that packet loss of the UDP mmWave link varies from 10% to 25%. This suggests that more effort is needed to address both bandwidth and packet loss through the use of advanced synchronization mechanisms.

To verify the support for the movement of mobile device, we perform experiments as shown in Figure 14. Table 9 proved that the 802.11ad wireless link as in Figure 14(a) provides a high throughput in most cases. However, when α_i was in a range of 25° to 35°, end-to-end throughput rapidly drops. Especially, at location 30°, end-to-end throughput approached zero. An applicable mechanism of sector switching should be implemented in order to address this problem. Additionally, Figure 15 presents results, where the mobile VR device was in a fixed position. The human's head only changes viewing direction from (0°) light-of-sight to 90° (byβ_i) offset as illustrated in Figure 14(b). We indicated that 75° is the threshold to maintain the quality of 802.11ad wireless communication at any sector 0-15. Therefore, we set up the mmWave communication for 360 videos streaming in range of LOS to 70°, and 802.11ad wireless link can work in a stable manner.

B. UDP 802.11AD AND FIRST SYNCHRONIZATION MECHANISM

Next, we tried to apply the first synchronization mechanism. The results are presented in Figure 16, which shows that the end-to-end throughput for both 2K and 4K video is approximately 280 Mbps. The latency time is caused by around three seconds of delays for pre-loading rendered video on mobile VR device. This latency results in the number of ACK packets increasing significantly based on the number of data packets while playing 4K-8K video. We then applied the FSE coder in order to reduce the size of the raw video data. This allowed us to gain more bandwidth, and the video data was

TABLE 5. Equirectangular projection for 360 video.

Parameters	Value	Note
SphereVideo	1	1: 360 video; 0: traditional video;
InputGeometryType	0	0: equirectangular; 1: cubemap; 2: equalarea;
SourceFPStructure	1 1 0 0	frame packing order
CodingGeometryType	0	
CodingFPStructure	1 1 0 0	frame packing order
SVideoRotation	0 0 0	rotation along X, Y, Z;
CodingFaceWidth	0	0: automatic calculation; 4096 for 8K; 3328 for 4K;
CodingFaceHeight	0	0: automatic calculation; 2048 for 8K; 1664 for 4K;
InterpolationMethodY	5	interpolation method for luma, 0: bicubic; 1:NN, 2: bilinear, 3: bicubic, 4: lanczos2, 5: lanczos3
InterpolationMethodC	4	interpolation method for chroma, 0: bicubic; 1:NN, 2: bilinear, 3: bicubic, 4: lanczos2, 5: lanczos3
InternalChromaFormat	420	internal chroma format for the conversion process;
SPSNR_NN	1	enable end-to-end S-PSNR-NN calculation;
SPSNR_I	0	enable end-to-end S-PSNR-I calculation;
CPP_PSNR	0	enable end-to-end CPP-PSNR calculation;
E2EWSPSNR	1	enable end-to-end WS-PSNR calculation;
CODEC_SPSNR_NN	1	enable codec S-PSNR-NN calculation;
WSPSNR	1	enable codec WS-PSNR calculation;
CF_SPSNR_NN	1	enable cross-format S-PSNR-NN calculation;
CF_SPSNR_I	0	enable cross-format S-PSNR-I calculation;
CF_CPP_PSNR	1	enable cross-format CPP-PSNR calculation;
SphFile		../cfg-360Lib/360Lib/sphere_655362.txt
ViewPortPSNREnable	0	1: Yes; 0: No
ViewPortList		2 75.0 75.0 0.0 0.0 75.0 75.0 -90.0 0.0
ViewPortWidth	428	1816 for 8K; 856 for 4K; 428 for Full HD
ViewPortHeight	428	1816 for 8K; 856 for 4K; 428 for Full HD
DynamicViewPort PSNREnable	1	1: Yes; 0: No
DynamicViewPortList		2 75.0 75.0 0 -45.0 -15.0 299 45.0 15.0 75.0 75.0 0 -135.0 -15.0 299 -45.0 15.0
DynamicViewPortWidth	428	1816 for 8K; 856 for 4K; 428 for Full HD;
DynamicViewPortHeight	428	1816 for 8K; 856 for 4K; 428 for Full HD;

TABLE 6. Coding configuration full HD 360 video - Base layer.

Parameters	Value
InputFile	sequence_1920x1080_30fps_8bit_420_erp.yuv
InputBitDepth	8 (Input bitdepth
InputChromaFormat	420 (Ratio of luminance to chrominance samples)
FrameRate	30 (Frame Rate per second)
FrameSkip	0 (Number of frames to be skipped in input)
SourceWidth	1920 (Input frame width)
SourceHeight	1080 (Input frame height)
FramesToBeEncoded	300 (Number of frames to be coded)
Level	5.2
DynamicViewPort PSNREnable	1
DynamicViewPortList	2 75.0 75.0 0 210 -18 299 300 12 75.0 75.0 0 30 -44 299 120 -14
DynamicViewPortWidth	428 (1816 for 8K; 856 for 4K; 428 for Full HD)
DynamicViewPortHeight	428 (1816 for 8K; 856 for 4K; 428 for Full HD)

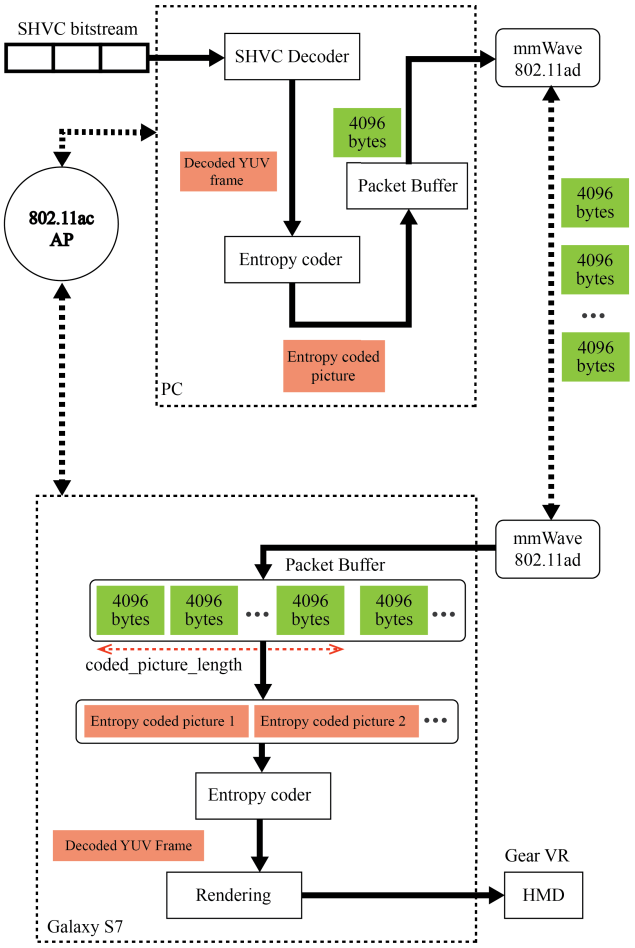


FIGURE 10. Testbed for the proposed system.

TABLE 7. Coding configuration 4K 360 video - Enhancement layer.

Parameters	Value
InputFile	sequence_4096x2048_30fps_8bit_420_erp.yuv
InputBitDepth	8 (Input bitdepth)
InputChromaFormat	420 (Ratio of luminance to chrominance samples)
FrameRate	30 (Frame Rate per second)
FrameSkip	0 (Number of frames to be skipped in input)
SourceWidth	4096 (Input frame width)
SourceHeight	2048 I(nput frame height)
FramesToBeEncoded	300 (Number of frames to be coded)
Level	5.2
DynamicViewPort PSNREnable	1
DynamicViewPortList	2 75.0 75.0 0 210 -18 299 300 12 75.0 75.0 0 30 -44 299 120 -14
DynamicViewPortWidth	856 (1816 for 8K; 856 for 4K; 428 for Full HD)
DynamicViewPortHeight	856 (1816 for 8K; 856 for 4K; 428 for Full HD)

transmitted faster than it was before applying the FSE coder, as shown in figure 17. Additionally, the problem of packet loss is overcome by using an ACK exchanging mechanism. By applying adaptive synchronization mechanism, all lost packets are retransmitted over the 802.11ad wireless link. In other words, mobile devices always get all the data without losing any packets. As shown in table 10, the packet loss rate for 2K and 4K videos are 18.64% and 22.45%, and the number of retransmitted packets are 4.76% and 9.84%, respectively.

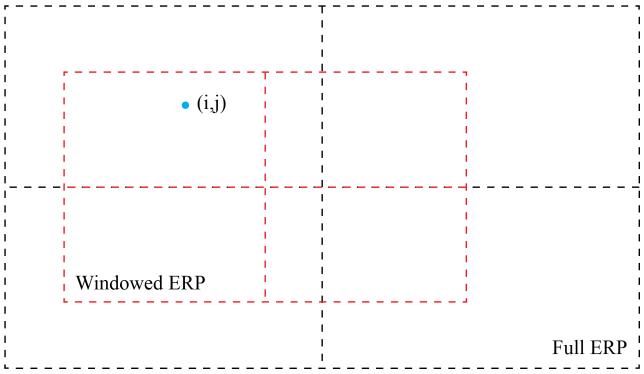


FIGURE 12. Windowed ERP and full ERP.

TABLE 8. Data rate of mmWave connection affected by distance.

Video Time	1 meter	2 meters	3 meters
2 secs	804 Mbps	890 Mbps	637 Mbps
4 ses	847 Mbps	899 Mbps	666 Mbps
6 secs	868 Mbps	893 Mbps	628 Mbps
8 secs	893 Mbps	914 Mbps	642 Mbps
10 secs	901 Mbps	923 Mbps	618 Mbps

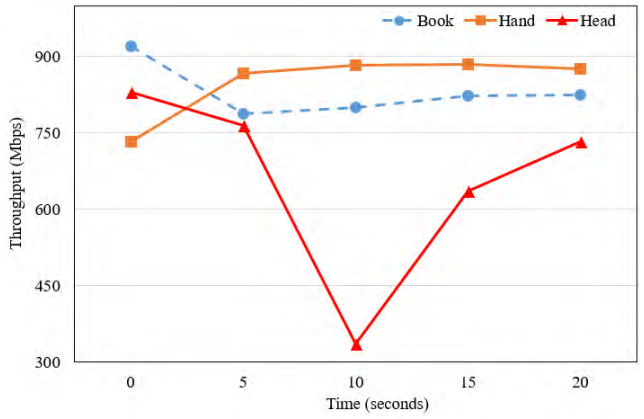


FIGURE 13. End-to-end throughput as affected by obstacles.

TABLE 9. End-to-end throughput (Mbps): mobile VR device moves around PC.

Video Time	$\alpha_i = 0°$	$\alpha_i = 15°$	$\alpha_i = 25°$	$\alpha_i = 30°$	$\alpha_i = 35°$	$\alpha_i = 45°$
1 sec	923	916	606	0	260	830
5 secs	922	912	644	0	268	838
10 secs	922	913	635	0	273	846
15 secs	927	907	659	0	300	848
20 secs	921	907	647	0	234	846
25 secs	922	905	655	0	270	847
30 secs	922	907	650	0	301	850

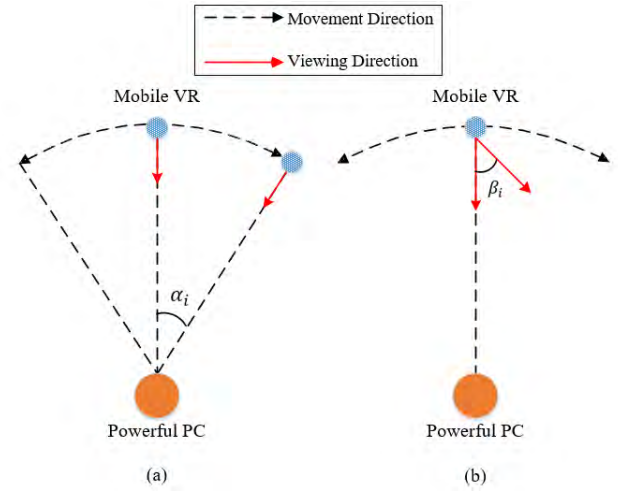


FIGURE 14. Movement experiments (a) mobile VR moves around PC (b) mobile VR changes viewing direction.

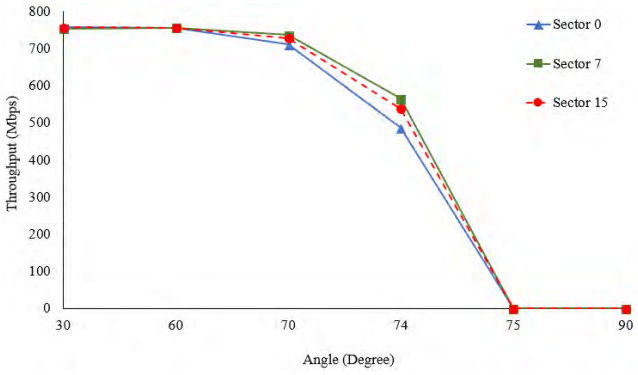


FIGURE 15. Mobile VR device changes viewing direction.

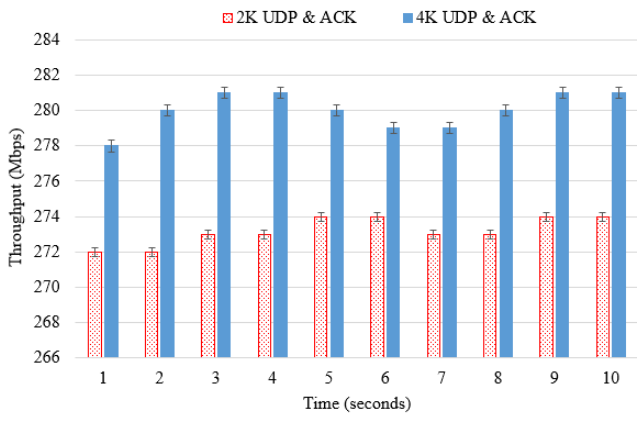


FIGURE 16. End-to-end throughput with mmWave ACK.

C. UDP 802.11AD AND SECOND SYNCHRONIZATION MECHANISM

In order to enhance the performance of 360 video streaming, we considered using TCP over the 802.11ac wireless network for packet retransmission using the second synchronization mechanism. Therefore, any packet loss resulting from UDP mmWave communication can be recovered by the second synchronization mechanism, and the throughput of video data transmission can be improved in order to handle the large data size of 4K or 8K video. As shown in Figure 18, the throughput, in this case, was improved to over 500 Mbps, meaning the proposed system could transfer 4K raw YUV 4:2:0 pictures (size of one picture: $(3/2) \times (3840 \times 2048) \times 8\ bits = 94.37\ Mbits$) at a rate of five pictures per second, while

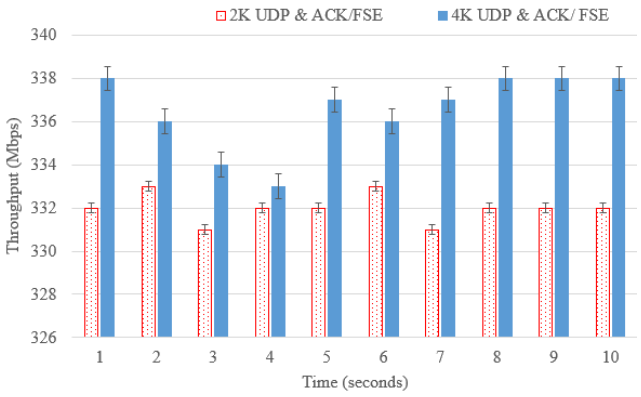


FIGURE 17. End-to-end throughput with mmWave ACK/FSE.

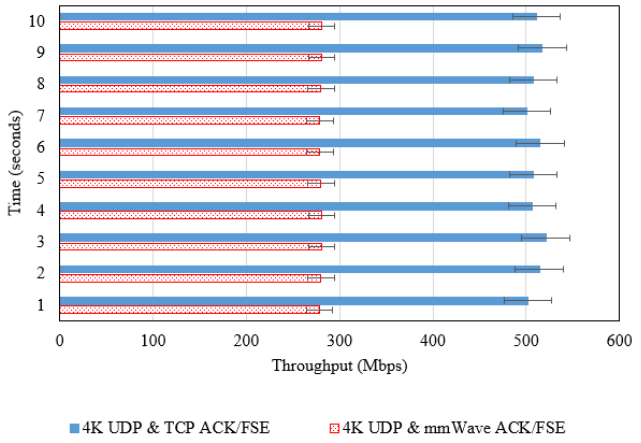


FIGURE 18. End-to-end throughput comparison between mmWave ACK and TCP ACK.

the mmWave ACK system could only transfer three such pictures per second. However, the 802.11ac link provides just 30-40 Mbps throughput on average for both ACK message exchange and the retransmission of data packets. Therefore, packet retransmission over the 802.11ac network increases the processing time at the mobile VR device. This means that the mobile VR device requires a video pre-loading time of around five seconds in order to ensure the QoS of the 4K video display. Therefore, the deployment of 802.11ad networks must also cover existing 802.11ac wireless networks.

D. PERFORMANCE COMPARISON

As shown in table 11 and table 12 the Structural Similarity (SSIM) and PSNR values were obtained through a comparison between the original YUV video [1] and the decoded YUV video, which was rendered on mobile VR device. In order to evaluate the quality of the decoded video, WS-PSNR has been shown to perform poorly compared to other quality metrics when it comes to estimating the quality of images and videos in particular, as realized by humans. Since PSNR values are higher than 38dB, we can determine that the quality of 360 video stream is reasonable to feel fully immersed in 360 videos on mobile VR device. In order to meet the demand of SHVC coding with the base layer and enhancement layer, we processed the comparison for both 1080p resolution and 4K resolution.

Through the SSIM and PSNR analysis, we can detect that the *DrivingInCity* test sequence gives the result a little bit different with other test sequences. The reason comes from the compatibility of HM software & library 360 with the various 4K resolutions in the encoding task. The test sequences of 4098 × 2048 resolution are more preferable with standard encoding configuration of HM & lib 360. The *DrivingInCity* test sequence is 3840 × 1920 resolution, and we already changed some options of the standard configuration to adapt the encoding procedure of 3840 × 1920 test sequences.

The quality of DrivingIncity 4K 360 video sequence was performed in the real-world experiment as shown in Figure 19. Additionally, the latency of video transmission is accepted to be QoS-sensitive as in the demonstration [29]. Additionally, Table 13 presents the performance comparison between the streaming system using an 802.11ac wireless network (SS802.11ac) and our proposed system. In the SS802.11ac, the mobile VR device gets an SHVC bitstream

TABLE 10. Packet loss and packet retransmission.

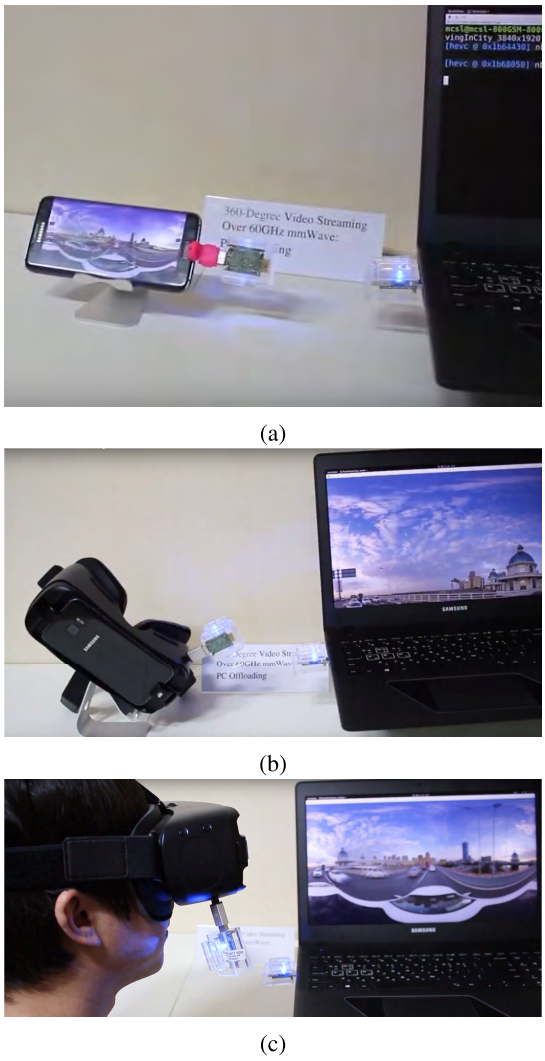
2K UDP Packet Loss	4K UDP Packet Loss	2K UDP mmWave ACK - Retransmission	4K UDP mmWave ACK - Retransmission
18.64%	22.45%	4.76%	9.84%

TABLE 11. Structural similarity comparison.

Test_sequence	Y Channel	U Channel	V Channel	All
*DrivingInCity*_1920 × 1080	0.9734	0.9819	0.9793	0.9758
*DrivingInCity*_3840 × 1920	0.9654	0.9839	0.9810	0.9711
*GasLamp*_1920 × 1080	0.9755	0.9871	0.983787	0.9788
*GasLamp*_4096 × 2048	0.9793	0.9918	0.9900	0.9831
*Harbor*_1920 × 1080	0.9736	0.9893	0.9894	0.9788
*Harbor*_4096 × 2048	0.9779	0.9928	0.9922	0.9828
*KiteFlite*_1920 × 1080	0.9781	0.9771	0.9819	0.9786
*KiteFlite*_4096 × 2048	0.9802	0.9863	0.9881	0.9825
*Trolley*_1920 × 1080	0.9783	0.9816	0.9869	0.9803
*Trolley*_4096 × 2048	0.9810	0.9889	0.9909	0.9840

TABLE 12. WS-PSNR comparison.

Test_sequence	WS-PSNR Y channel	WS-PSNR U channel	WS-PSNR V channel
*DrivingInCity*_1920 × 1080	40.32	46.18	45.55
*DrivingInCity*_3840 × 1920	39.69	46.31	45.64
*GasLamp*_1920 × 1080	42.49	47.78	47.05
*GasLamp*_4096 × 2048	43.14	49.31	48.69
*Harbor*_1920 × 1080	41.90	48.38	48.53
*Harbor*_4096 × 2048	42.27	49.41	49.23
*KiteFlite*_1920 × 1080	39.01	45.27	46.44
*KiteFlite*_4096 × 2048	39.98	47.09	47.80
*Trolley*_1920 × 1080	40.43	46.47	47.72
*Trolley*_4096 × 2048	40.73	48.08	48.70



(a)

(b)

(c)

FIGURE 19. Demo of (a) 4K DrivingInCity 360 video streaming via 802.11ad 60GHz wireless connection (b) 360 video streaming on Mobile VR device based HMD (c) Watching 360 video streaming.

from the video server, then it handles locally this bitstream by itself. In order to evaluate streaming systems, we performed 4K_30fps_300frames_8bit_420_erp_27qp (quantization parameter of 27) test sequences as shown in table 4 for tile partitioning. The openHEVC software was used as a video decoder to decode 360 videos. The experiments prove that the proposed scheme can enhance the overall performance of the system while streaming 360 videos. Especially, its offloading mechanism can help speed up the system in video decoding task up to 67.4%.

Through implementation and demonstration, we confirmed that these commercial 802.11ad 60GHz devices are currently under development. Thus, a few components are not as advanced as we expected. Dongle *MLWGU3V2-D* and Tensorcom *3.0 TC60G-USB3.0 EVK* devices have some disadvantageous points as follows:

- They only operate well in the case using peer-to-peer mode. This means that it can only be active with one connected-pair. These devices also give the network mode option, but this option cannot work properly. Although we put a lot of effort into fixing bugs, it has not been completely activated. This problem should be solved by manufacturers.
- The information regarding mmWave devices is not detailed, especially that about the MAC layer and the Physical layer. This is an inconvenience for implementing internal protocols or analyzing it in detail.

TABLE 13. Overall performance comparison.

Test Sequence	Option	**Proposed system**	**SS802.11AC**
DrivingInCity	Throughput	330 Mbps	78.1 Mbps
	Video coding task: time - FPS	14.16 secs - 21 FPS	57.14 secs - 5.25 FPS
	Average PSNR	41.86 dB	39.25 dB
GasLamp	Throughput	336 Mbps	77 Mbps
	Video coding task: time - FPS	12.72 secs - 24 FPS	55.42secs - 5.39 FPS
	Average PSNR	45.14 dB	41.59 dB
Harbor	Throughput	341 Mbps	74 Mbps
	Video coding task: time - FPS	13.24 secs - 23 FPS	56.28 secs - 5.31 FPS
	Average PSNR	44.40 dB	41.05 dB
Trolley	Throughput	328 Mbps	69.3 Mbps
	Video coding task: time - FPS	13.93 secs - 22 FPS	54.39 secs - 5.49 FPS
	Average PSNR	42.88 dB	39.71 dB

The demonstration [29] was performed in a real-world environment. There are many papers that have been published related to the research area of mmWave. However, the proposed approaches in these articles show the experimental results from simulation software or other hardware platforms, which are only specified to research at the MAC layer or the Physical layer. Support for applications using mmWave, however, is little explored. Thus, in this paper, we focused on a real-world scenario with mmWave communication in order to answer how mmWave is possible for high-quality video streaming with the constrained resource of the mobile VR device. All of these experiments encouraged us to implement advanced strategies to facilitate 360 video VR system in the future.

VI. CONCLUSION

We designed and implemented a 360 video transmission system using SHVC video coding, FSE coder and computing offloading over mmWave communication. We determined that 802.11ad wireless communication can improve the overall performance of 360 videos streaming in the indoor environment. Moreover, our adaptive proposed scheme also indicates that 802.11ad wireless communication can facilitate 4K-8K video transmission in real-time through the application of optimization mechanisms.

In future work, we will consider implementing relay prototypes in order to support blockage cases. Another future research direction is to design a new, more reliable communication technique for our system based on 802.11ad devices in order to improve the QoS and quality of experience (QoE)

of 360 video streaming. Furthermore, a sector switching mechanism to ensure QoE of the user is also a delightful topic of our research in the future. Regarding the real-time issue, more handshake policies can be improved efficiently to enhance 360 video streaming.

REFERENCES

[1] *JCT-VC—Joint Collaborative Team on Video Coding*. Accessed: Nov. 3, 2018. [Online]. Available: https://www.itu.int/en/ITU-T/studygroups/2017-2020/16/Pages/video/jctvc.aspx

[2] Y. Ryu and E. S. Ryu, "Video on mobile CPU: UHD Video parallel decoding for asymmetric multicores," in *Proc. 8th ACM Int. Conf. Multimedia Syst. (MMSys)*, Taipei, Taiwan, 2017, pp. 229–232.

[3] J. W. Son, D. M. Jang, and E. S. Ryu, "Implementing motion-constrained tile and viewport extraction for VR streaming," in *Proc. ACM Netw. Oper. Syst. Support Digit. Audio Video (NOSSDAV)*, Amsterdam, The Netherland, Jun. 2018, pp. 61–66.

[4] H. W. Kim, T. T. Le, and E. S. Ryu, "360-degree video offloading using millimeter-wave communication for cyberphysical system," in *Transactions on Emerging Telecommunications Technologies*. Hoboken, NJ, USA: Wiley, Jul. 2018.

[5] T.-T. Le, D. N. Van, and E.-S. Ryu, "Real-time 360-degree video streaming over millimeter wave communication," in *Proc. Int. Conf. Inf. Netw. (ICOIN)*, Jan. 2018, pp. 5–17.

[6] T. S. Rappaport, Y. Xing, and G. R. MacCartney, "Overview of millimeter wave communications for fifth-generation (5G) wireless networks—With a focus on propagation models," *IEEE Trans. Antennas Propag.*, vol. 65, pp. 6213–6230, Dec. 2017.

[7] X. Wu *et al.*, "60-GHz millimeter-wave channel measurements and modeling for indoor office environments," *IEEE Trans. Antennas Propag.*, vol. 65, no. 4, pp. 1912–1924, Apr. 2017.

[8] P. Liu, M. D. Renzo, and A. Springer, "Line-of-sight spatial modulation for indoor mmWave communication at 60 GHz," *IEEE Trans. Wireless Commun.*, vol. 15, no. 11, pp. 7373–7389, Nov. 2016.

[9] G. R. Maccartney, T. S. Rappaport, S. Sun, and S. Deng, "Indoor office wideband millimeter-wave propagation measurements and channel models at 28 and 73 GHz for ultra-dense 5G wireless networks," *IEEE Access*, vol. 3, pp. 2388–2424, 2015.

[10] Y. Wang and G. D. Veciana, "Dense indoor mmWave wearable networks: Managing interference and scalable MAC," in *Proc. 14th Int. Symp. Modeling Optim. Mobile Ad Hoc Wireless Netw. (WiOpt)*, Tempe, AZ, USA, May 2016, pp. 1–8.

[11] J. S. Lu, D. Steinbach, P. Cabrol, and P. Pietraski, "Modeling human blockers in millimeter wave radio links," *ZTE Commun.*, vol. 10, no. 4, pp. 23–28, 2012.

[12] M. Gapeyenko *et al.*, "Analysis of human-body blockage in urban millimeter-wave cellular communications," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2016, pp. 1–7.

[13] Seed Studio. *WiGig USB3.0 Dongle, MLWGU3V2-D*. Accessed: Nov. 3, 2018. [Online]. Available: https://www.seeedstudio.com/WiGig-USB3.0-Dongle-p-2827.html

[14] Tensorcom. *TC60G-USB3.0 EVK*. Accessed: Nov. 3, 2018. [Online]. Available: http://tensorcom.com/products-1/

[15] T. K. Tan, M. Mrak, V. Baroncini, and N. Ramzan, *Report on HEVC Compression Performance Verification Testing*, document JCTVC-Q1011, Valencia, Spain, Mar./Apr. 2014.

[16] J. Boyce *et al.*, *Draft High Efficiency Video Coding (HEVC) Version 2, Combined Format Range Extensions (RExt), Scalability (SHVC), and Multi-View (MV-HEVC) Extensions*, document JCTVC-R1013_v6, Sapporo, Japan, Jun./Jul. 2014.

[17] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

[18] G. J. Sullivan, J. M. Boyce, Y. Chen, J.-R. Ohm, C. A. Segall, and A. Vetro, "Standardized extensions of high efficiency video coding (HEVC)," *IEEE J. Sel. Topics Signal Process*, vol. 7, no. 6, pp. 1001–1016, Dec. 2013.

[19] E. S. Ryu and J. H. Kim, "Error concealment mode signaling for robust mobile video transmission," *AEU-Int. J. Electron. Commun.*, vol. 69, no. 7, pp. 1070–1073, Jul. 2015.

[20] E. S. Ryu and C. Yoo, "An approach to interactive media system for mobile devices," in *Proc. 12th Annu. ACM Int. Conf. Multimedia*, Oct. 2004, pp. 160–161.

[21] J. Kim and E. S. Ryu, "Quality analysis of massive high-definition video streaming in two-tiered embedded camera-sensing systems," in *Proc. Int. J. Distrib. Sensor Netw.*, vol. 10, no. 3, p. 634191, Mar. 2014.

[22] *Shannon Limit Theory*. Accessed: Nov. 3, 2018. [Online]. Available: https://en.wikipedia.org/wiki/Entropy_(information_theory)

[23] J. Duda. (May 2009). "Asymmetric numeral systems: Entropy coding combining huffman coding with compression rate of arithmetic coding." [Online]. Available: https://arxiv.org/abs/0902.0271

[24] *Asymmetric Numeral Systems*. Accessed: Nov. 3, 2018. [Online]. Available: https://en.wikipedia.org/wiki/Asymmetric_numeral_systems

[25] *Facebook Zstandard*. Accessed: Nov. 3, 2018. [Online]. Available:http://facebook.github.io/zstd/

[26] *Lzbench Benchmark Tool*. Accessed: Nov. 3, 2018. [Online]. Available: https://github.com/inikep/lzbench

[27] R. Ghaznavi-Youvalari, "Comparison of HEVC coding schemes for tile-based viewport-adaptive streaming of omnidirectional video," in *Proc. IEEE 19th Int. Workshop Multimedia Signal Process. (MMSP)*, Oct. 2017, pp. 1–6.

[28] R. Skupin, "ISO/IEC JTC1/SC29/WG11/N16499: Working draft 1 of motion constrained tile sets extraction information SEI messages," MPEG 116-Chengdu, Chengdu, China, Oct. 2016. Accessed: Nov. 3, 2018. [Online]. Available: https://mpeg.chiariglione.org/meetings/116

[29] *360 Video Streaming Over mmWave Demo*. Accessed: Nov. 3, 2018. [Online]. Available: https://youtu.be/7L8wGU50CZ8

[30] *High Efficiency Video Coding (HEVC) JCT-VC*. Accessed: Nov. 3, 2018. [Online]. Available: https://hevc.hhi.fraunhofer.de/

[31] *Open HEVC Decoder*. Accessed: Nov. 3, 2018. [Online]. Available: http://openhevc.github.io/openHEVC/

[32] *FFmpeg Tool*. Accessed: Nov. 3, 2018. [Online]. Available: https://www.ffmpeg.org/

[33] *ERP WS-PSNR Software*. Accessed: Nov. 3, 2018. [Online]. Available: http://mpegx.int-evry.fr/software/MPEG/Explorations/3DoFplus/ERP_WS-PSNR

[34] M. Yu, H. Lakshman, and B. Girod, "A framework to evaluate omnidirectional video coding schemes," in *Proc. IEEE Int. Symp. Mixed Augmented Reality (ISMAR)*, Sep./Oct. 2015, pp. 31–36.

[35] Y. Sun, A. Lu, and L. Yu, "Weighted-to-spherically-uniform quality evaluation for omnidirectional video," *IEEE Signal Process. Lett.*, vol. 24, no. 9, pp. 1408–1412, Sep. 2017.

TUAN THANH LE received the B.S. degree from the Department of Electronic and Telecommunication Engineering, Hanoi University of Sciences and Technology, Vietnam, in 2010, and the M.S. degree from the Department of IT Engineering, Kongju National University, South Korea, in 2013. He is currently pursuing the Ph.D. degree with the Department of Computer Engineering, Gachon University, Seongnam, South Korea. Prior to joining Gachon University for the Ph.D. degree in 2017, he was a Network System Engineer with Future Systems, Seongnam, from 2014 to2017. His research interests are in the areas of multimedia communications, including video source coding and wireless mobile systems.

DIEN VAN NGUYEN received the B.S. degree from the Department of Electronic and Telecommunication Engineering, Hanoi University of Sciences and Technology, Vietnam, in 2013, and the M.S. degree from the Department of IT Engineering, Kongju National University, South Korea, in 2015. He is currently pursuing the Ph.D. degree with the Department of Computer Engineering, Gachon University. His research interests are in the areas of multimedia communications, including video source coding and wireless mobile systems.

EUN-SEOK RYU (SM'16) received the B.S., M.S., and the Ph.D. degrees in computer science from Korea University, Seoul, South Korea, in 1999, 2001, and 2008, respectively. He was also a Staff Engineer at InterDigital Labs, San Diego, CA, USA, from 2011 to 2014, where he researched and contributed to next-generation video coding standards, such as high-efficiency video coding and scalable HEVC. From 2008 to 2010, he was a Post-Doctoral Research Fellow with the Georgia Centers for Advanced Telecommunications Technology, School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, USA. In 2008, he was a Research Professor with the Research Institute for Information and Communication Technology, Korea University. He is currently an Assistant Professor with the Department of Computer Engineering, Gachon University, Seongnam, South Korea. Prior to joining Gachon University in 2015, he was a Principal Engineer at Samsung Electronics, Suwon, South Korea, where he led the multimedia team. His research interests are in the areas of multimedia communications, including video source coding and wireless mobile systems.