Enhancement of Motion Feedback Latency for Wireless Virtual Reality in IEEE 802.11 WLANs

Dang Tai Tan, Sanghyun Kim, and Ji-Hoon Yun
Department of Electrical and Information Engineering, Research Center for Electrical and Information Technology
Seoul National University of Science and Technology, Seoul, Korea
e-mail: {taitan7411, shk0787, jhyun}@seoultech.ac.kr

***Abstract*—Wireless virtual reality (VR) that offloads VR processing to a powerful PC and streams rendered VR image frames to a VR headset wirelessly is a promising technology for high-quality interactive VR experiences with free mobility and high immersiveness. Realizing wireless VR in IEEE 802.11 WLANs is feasible when combined with the *Timewarp* technique, but minimal latency is still of importance for higher quality of VR service. In this paper, we reveal that the bi-directional transmission nature of wireless VR (VR frame data in downlink and motion-feedback data in uplink) and the resulting delay of motion feedback is a major challenge of wireless VR in IEEE 802.11 WLANs. To combat this challenge, we propose three basic methods—(1) prioritizing aged motion data, (2) using reverse direction, and (3) limiting the aggregation size of downlink transmission—and demonstrate that all enhance the latency and jitter of motion feedback, and, among these, limiting the aggregation size is most effective. We then design a scheme to adjust the aggregation size of downlink transmission in conjunction with the use of reverse direction to best support responsive and frequent motion feedback while preserving the downlink transmission of VR frame data.**

***Index Terms*—virtual reality, wireless VR, motion feedback, IEEE 802.11**

I. INTRODUCTION

Virtual reality (VR) is realistic and immersive simulation of a three-dimensional environment, created using interactive software and hardware, and explored by the movement of a user's body. VR services are delivered to a user through a VR-dedicated headset device equipped with a high pixel-resolution display and an inertial measurement unit (IMU) to capture the user's head motion. A user wearing a VR headset sees the stereoscopic image of the virtual world that is rendered corresponding to the user's current viewport (estimated from the latest head tracking data)[1] on a display panel at an ultrashort viewing distance (several centimeters) through binocular magnifying lenses to allow a large field of view (FOV).

There are two types of VR headset system: *tethered* and *untethered*. VR processing of a tethered headset (e.g. Oculus Rift and HTC VIVE) is done at a powerful PC (or a gaming console), thus offering the highest-quality interactive VR experiences of all kinds with high resolution and frame rate. Today's tethered headsets, however, accompany cables for display and motion data, and such a wire harness disturbs a user's mobility, thus degrading immersiveness and creating a tripping hazard. Untethered headsets (e.g. Samsung Gear VR and Oculus Go) have a processing unit within a headset (either an installed smartphone or a built-in unit) and give VR services with no wires, thus are portable and convenient, but for lower quality of contents due to limited processing power.

A promising direction of evolution for both types of VR headset system is to offload VR processing to a high-end host PC (like the tethered case) and stream rendered VR image frames to a VR headset wirelessly, which we call *wireless VR* in the paper. The data traffic of wireless VR has a bi-directional nature; VR image frames rendered by the host PC are transferred to the VR headset in downlink and the motion data of the headset is fed back to the host in uplink so that the next VR image frame is rendered for the user's latest viewport.

However, *latency* is a major challenge of wireless VR, which is also the problem of the conventional wired VR system [1], but gets severer in wireless VR due to wireless transmission. As the latency increases, the inconsistency between a rendered VR image and the user's viewport at the time of scan-out also increases. Large inconsistency leads to motion sickness and makes the user quit the service finally. The inconsistency comes not only from the latency of the downlink transmission of VR frame data, but also from the latency and jitter of motion feedback in uplink.

There have been some attempts to make tethered headsets wireless using 60 GHz radio technology; TPCast (based on a proprietary technology) and the VIVE wireless adapter (based on WiGig) are representative cases. There have also been some research works on the feasibility of wireless VR using 60 GHz transmission [2]. Thanks to the ultrawide bandwidth (2.16 GHz per channel) and resulting multi-Gbps transmission speed, the 60 GHz technology is suitable for streaming high-resolution VR image frames with no or light compression at low latency. However, the inherent characteristics of 60 GHz spectrum result in short transmission distance and unstable connection due to blockage [3]. High battery consumption is another problem, which results from the use of higher emission power[2] than for other unlicensed bands.

---

[1] We focus on interactive VR service in this paper, which is more challenging for wireless VR.

[2] The US Federal Communications Commission (FCC) specified the total maximum transmit power of 500 mW for an emission bandwidth greater than 100 MHz in 54-66 GHz [4] .

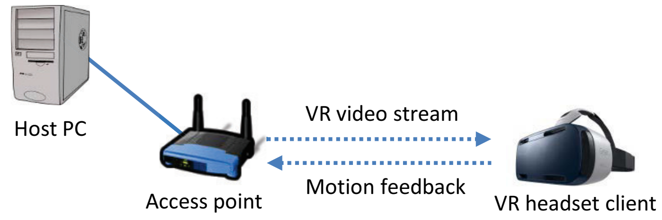


Fig. 1. Wireless VR system in an IEEE 802.11 WLAN

IEEE 802.11 wireless LAN (WLAN) is an alternative to implement wireless VR [5], [6]. It has not been considered as a viable solution for wireless VR due to the insufficient transmission speed, thus inevitable latency and motion sickness. Recently, however, some solutions like onAirVR [7] proved the feasibility of wireless VR in IEEE 802.11 WLANs. The key to the success of the solution is the adoption of the *Timewarp* technique [8], [9] by which every received image is reprojected according to the latest pose of a user and thus the inconsistency that the user perceives is minimized. However, as the total latency gets excessive, Timewarp produces noticeable black borders and reprojected images get much different from ideal ones, thus disturbing immersion. Therefore, redesigning IEEE 802.11 for wireless VR is still needed.

In this paper, we reveal that the bi-directional transmission nature of wireless VR results in deterioration of motion feedback in IEEE 802.11 WLANs; downlink transmission of periodic VR frame data with a large size occupies the channel medium for a long time and thus uplink transmission of motion-feedback data fails to be responsive and as frequent as the feedback-data generation rate. In order to better handle motion-data transmission, we propose three basic methods: (1) prioritizing aged motion data; (2) using reverse direction; (3) limiting the aggregation size of downlink transmission, and show through simulation that all methods enhance the performance of motion feedback and the last one achieves the highest gain. We then design a scheme to configure the best aggregation size of downlink transmission to maximize the motion-feedback performance while preserving the downlink transmission performance of VR frame streaming. To the best of our knowledge, this is the first in-depth study on latency enhancement of wireless VR in IEEE 802.11 WLANs.

The rest of the paper is organized as follows. In Section II, we describe the wireless VR system in IEEE 802.11 WLANs. Section III presents important observations on the problem of motion feedback. We then propose and evaluate basic enhancement methods in Section IV. The proposed scheme of aggregation size adjustment is described and its performance results are given in Section V. Finally, Section VI concludes the paper.

II. WIRELESS VR SYSTEM IN IEEE 802.11 WLAN

A wireless VR system in an IEEE 802.11 WLAN is illustrated in Fig. 1; A local host PC connected to an access point (AP) of IEEE 802.11 generates and streams VR image frames to a VR headset, which is the downlink transmission of the WLAN. The VR headset keeps reporting motion data to the host PC, which corresponds to the uplink transmission of the WLAN.

Due to the insufficient transmission speed of IEEE 802.11, VR image frames are compressed as a video stream using a compression codec such as H.265. But, each of compressed VR frames, which we now call a *VR video frame*, is still large in size, thus are transmitted as multiple packets. In order to transmit such a set of packets for a VR video frame efficiently in WLAN, *frame aggregation* will be used, which is the key feature of IEEE 802.11 for throughput enhancement. Then, a set of multiple packets are aggregated[3] and transmitted at once, thus reducing the overhead of channel access and physical-layer headers.

In order to track a user's viewport, the VR headset has a built-in inertial measurement unit (IMU) typically composed of gyroscope, accelerometer and magnetometer sensors, and feeds back the user's head pose (yaw, pitch and roll orientations) measured by the IMU along with raw sensor data to the host PC. The data size of each feedback is several tens of bytes only (44 bytes for Oculus Rift [10]). However, for accurate identification of the current head pose, the rate of motion feedback is as high as several hundreds of times per second.[4] Obviously, the latency of motion feedback must be minimized and the rate must be maximized [11].

III. PERFORMANCE EVALUATION OF WIRELESS VR IN IEEE 802.11 WLAN

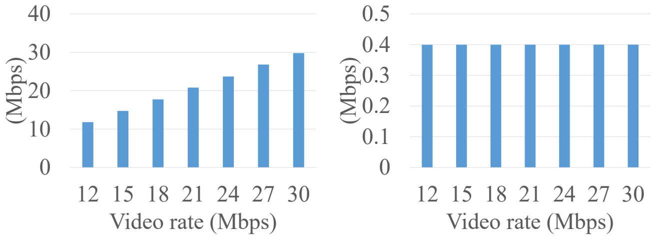
In this section, we evaluate the performance of wireless VR transmission in an IEEE 802.11ac WLAN via simulation. In the simulation, there is one VR service session for a VR headset paired with a host PC. The downlink VR frame data is generated at 60 Hz with an exponentially-distributed size for a given VR video rate and packetized with the maximum packet size of 1500 bytes. The uplink motion-feedback data with a payload size of 44 bytes is generated at the rate of 500 Hz. The maximum number of packets that a transmission can aggregate is set to 64. The transmission bit rate of the WLAN is 65 Mbps. We vary the mean VR video rate and evaluate the system performance in terms of throughput, motion-feedback latency and jitter. The jitter is obtained as the gap between the interval of contiguous motion-data generations and the interval of their receptions at the host PC.

The results are shown in Figs. 2, 3 and 4. In Fig. 2, both downlink and uplink transmissions meet their throughput requirements for all VR video rates under consideration. Motion-feedback latency, however, increases as the VR video rate increases. Its jitter increases as well. The cumulative distribution functions (CDFs) in Fig. 3 show that the range

---

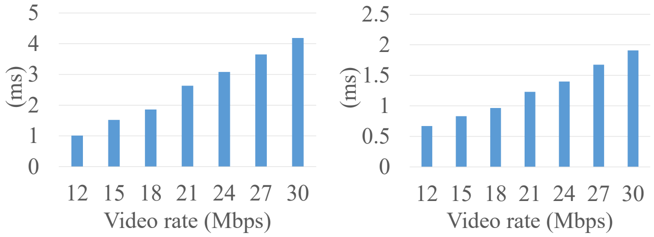
[3] We assume that one packet constructs one medium access control protocol data unit (MPDU). For simplicity of exposition, we use the term "packet" instead of "MPDU" throughout the paper.

[4] According to our experimental measurement of Oculus Rift DK1, the rate of motion feedback reaches 500 Hz.



(a) Throughput of VR video streaming

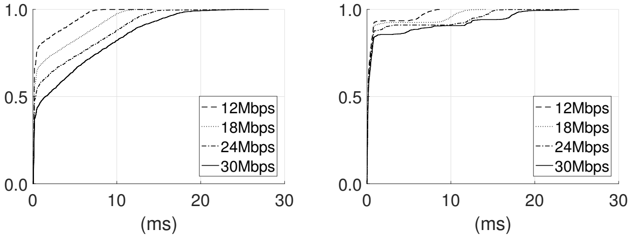
(b) Throughput of motion feedback



(c) Motion-feedback latency

(d) Motion-feedback jitter

Fig. 2. Performance of wireless VR transmission



(a) Latency

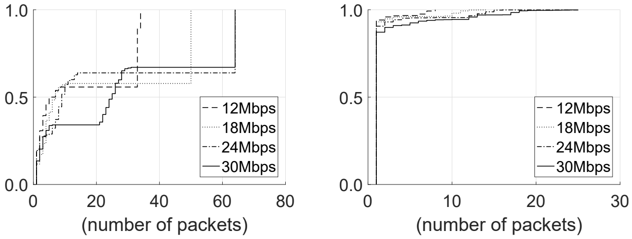
(b) Jitter

Fig. 3. CDF of motion-feedback latency and jitter for various VR video rates

of motion-feedback latency is wide; for the VR video rate of 30 Mbps, the average of motion-feedback latency is 4.2 ms, but more than 20% of motion-feedback reports experience latency higher than 10 ms. The root cause of such deteriorated motion-feedback latency is that the AP and the headset client contend with each other for the same channel. In particular, the downlink transmission conveying large VR frame data occupies the channel medium for a long time. Therefore, once the channel is occupied by a downlink transmission, the headset client has to wait until the transmission finishes. Fig 4 shows how large each downlink-transmission data is in terms of the aggregation size; for the video rate of 12 Mbps, 45% of transmissions aggregate over 30 packets and, for the video rates of 24 and 30 Mbps, over 30% of transmissions reach the maximum aggregation size, thus leading to long channel occupation time. This results in increased latency of motion feedback and even aggregated transmission of multiple motion reports to some degree (over 10% of uplink transmissions for the video rate of 30 Mbps).

IV. BASIC ENHANCEMENT METHODS

We propose three basic methods to reduce motion-feedback latency and jitter, and evaluate each's gain.



(a) VR video streaming

(b) Motion feedback

Fig. 4. CDF of aggregation size for various VR video rates

A. *Prioritizing Aged Motion Data*

The first method is to control the priority of the head-of-line (HOL) motion data packet according to its age after generation. The age of motion data is counted as the period from the time when it is generated to the present time. As the age of HOL motion data gets older, it is given higher priority and the backoff count is decreased faster. We define N stages of age according to configured age thresholds (α_n, $n = 1, \cdots, N$); if the age of HOL motion data is lower than α_n, it is in the n-th stage of age and a backoff-count decrease is done by a specified ratio, denoted by β_n, of the contention window (CW) size of the current backoff stage.

B. *Motion Feedback using Reverse Direction*

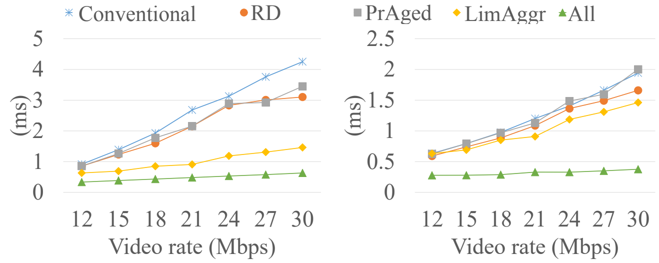
One cause of long motion-feedback latency is the backoff procedure needed for every motion-feedback transmission. So, we propose to exploit the optional feature of IEEE 802.11 called *reverse direction* (RD) [12] for the transmission of motion-feedback data. RD allows the receiver of a data frame to reply back to the sender with its data immediately within the sender's transmission opportunity (TXOP) time, thus reducing reply latency. With RD, a VR headset client can transmit its motion data right after a transmission of a VR downlink frame (in short interframe space (SIFS) with no completion of its backoff procedure) if the TXOP of the downlink transmission is not fully used.

C. *Limiting Aggregation Size*

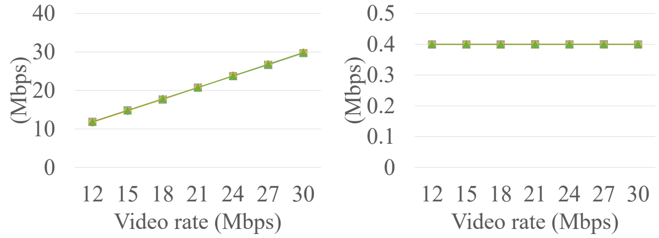
Prioritizing aged motion data reduces the channel access time of motion feedback and RD even eliminates it, but a long channel occupation of a VR-frame data transmission still delays motion feedback. Moreover, motion feedback does not always benefit from RD; If TXOP of a downlink transmission is fully used by itself or a motion report is generated between downlink transmissions, the headset client still has to complete the backoff procedure to transmit its motion data. Therefore, limiting the channel occupation time of a VR frame transmission will be effective, which can be realized by limiting the aggregation size of a downlink transmission.

D. *Performance Evaluation of Enhancement Methods*

We now evaluate the gain of each method. For the first method (prioritizing aged motion data, denoted as PrAged), based on the CDF of latency in Fig. 3, we configure five



(a) Motion-feedback latency (b) Motion-feedback jitter



(c) Throughput of VR video streaming (d) Throughput of motion feedback

Fig. 5. Performance of enhancement methods

stages of age as $(\alpha_1, \cdots, \alpha_5) = (3, 6, 9, 12, \infty)$ ms; the ratios of backoff-count decrease to the CW size are configured as $(\beta_1, \cdots, \beta_5) = (\mathrm{CW}^{-1}, 0.3, 0.45, 0.7, 0.85)$ (i.e., for the first stage of age, we use the legacy backoff procedure decreasing the backoff count by one). For the third method (limiting aggregation size, denoted as LimAggr), we limit the aggregation size of a downlink transmission as 4, 12 and 18 packets for the VR video rates of 12∼17, 18∼29 and 30 Mbps, respectively.

The performance results of the enhancement methods are shown in Fig. 5. All methods reduce the average latency of motion feedback compared to the conventional case. Among all methods, limiting aggregation size (LimAggr) achieves the highest gain. The proposed methods except prioritizing aged motion data (PrAged) also enhance jitter. Finally, combining all methods (All) at the same time dramatically decreases latency and jitter, both below 1 ms for all video rates under consideration. Despite the reduction of motion-feedback latency and jitter, the individual methods as well as their combination do not affect the throughput of VR video streaming. This is because each motion feedback has a small data size and its transmission occupies the channel for a short period only.

V. LIMITING AGGREGATION SIZE: OPTIMAL CONFIGURATION OF THE LIMIT

In the previous section, we recognized that limiting aggregation size is the best method for reducing latency and jitter. However, this method needs a proper configuration of the limit of the aggregation size. If the limit is set too small, the efficiency of downlink transmission is degraded (throughput is reduced) and thus VR video streaming will deteriorate. To the contrary, if the limit is too large, motion feedback will deteriorate. This motivates us to design a scheme to find an optimal limit value of the aggregation size.

A. *Design of The Optimal Configuration Scheme*

First, we formulate the target problem of the optimal configuration. The problem variable is the maximum aggregation size of a downlink transmission, which we denote by n^h_{pkt}.[5] Since larger n^h_{pkt} deteriorates motion feedback more, we have to minimize it while meeting the transmission requirement of VR video streaming, which can be described as that a VR video frame has to be transmitted completely before the next frame is generated. Therefore, the target problem P is expressed as

$$P: \quad \begin{array}{l} \text{minimize } n^h_{pkt} \\ \text{s.t.} \\ \text{Every VR video frame data is transmitted} \\ \text{before the next frame is generated.} \end{array} \tag{1}$$

If there exists no feasible solution of the problem, the wireless VR system cannot provide a stable VR service with the current video rate, thus has to lower the rate by changing the compression configuration (e.g., resolution, quantizer parameter).

For the simplicity of the scheme design, we make the following assumptions:

Assumption 1 (Saturated traffic condition)**.** Based on the problem formulation, the scheme aims to minimize the amount of data for each downlink transmission, thus is likely to make the AP always have VR frame data to transmit. Due to the high rate of motion-feedback, the VR headset client is also likely to always have data to transmit.

Assumption 2 (Same channel condition)**.** The AP and VR headset client are assumed to be under the same channel condition, i.e., the same transmission failure probability, which thus leads to the same average CW size and channel access probability in the long term.

These assumptions lead to the equal number of transmissions by the AP and the headset client during a frame interval in the average sense.

The proposed scheme considers the combination of two enhancement methods: LimAggr and RD. Hence, there are two transmission cases of motion feedback: (1) in the client's own TXOP and (2) in the AP's TXOP via RD. Let R_{fr} be a VR frame rate (the number of frames per unit time; 60 or 90 Hz is typical). Assume that, during a VR frame interval $(1/R_{fr})$, there exist N_{tx} downlink and uplink transmissions, respectively. Let T^h_{tx} and T^c_{rd} be the average times of the AP's transmission and the client's transmission (via RD), respectively, within the AP's TXOP. Similarly, T^c_{tx} is the average time of the client's transmission within the TXOP of its transmission. T_{idle} is the idle time of the WLAN between two consecutive transmissions. Then, we have an inequality of the above variables as

$$N_{tx}(T^h_{tx} + T^c_{rd} + T_{idle} + T^c_{tx} + T_{idle}) \leq R^{-1}_{fr}. \tag{2}$$

---

[5] The superscript h of all notations stands for "host" and c stands for "client".

We obtain the variables of Ineq. (2) in the following. Let us denote the total number of packets to be transmitted by the host during a frame interval by N^h_{pkt}, which is obtained as

$$N^h_{pkt} = \frac{1}{R_{fr}} \frac{R_v}{L^h_P} \frac{1}{1-p} \tag{3}$$

where $1/R_{fr}$ is the VR frame interval, R_v is the average date rate of compressed VR frames (the amount of data per unit time), L^h_P is the packet size of the host and p is the transmission failure probability of a packet in the WLAN. Then, we obtain N_{tx} in terms of N^h_{pkt} and n^h_{pkt} (aggregation size) as

$$N_{tx} = N^h_{pkt}/n^h_{pkt}. \tag{4}$$

Next, we obtain T^h_{tx} as

$$\begin{aligned} T^h_{tx} &= T_{phy} + T^h_{mpdu} \times n^h_{pkt} + T_{ba-ex} + T_{sifs} \\ &= T^h_{mpdu} \times n^h_{pkt} + T_{ov} \end{aligned} \tag{5}$$

where T_{phy}, T^h_{mpdu}, T_{ba-ex} and T_{sifs} are the times of the physical-layer preamble and header, one packet transmission, block acknowledgement (BA) exchange procedure and SIFS, respectively, and T_{ov} is the sum of all overhead times (T_{phy}, $T_{ba\text{-}ex}$ and T_{sifs}). T^h_{mpdu} and T_{ba-ex} are further expressed as

$$T^h_{mpdu} = \frac{L_h + L^h_P + L_{delim}}{r^h} \tag{6}$$

and

$$T_{ba-ex} = T_{sifs} + T_{bar} + T_{sifs} + T_{ba} \tag{7}$$

where L_h and L_{delim} are the lengths of the medium access control (MAC) header and delimiter, respectively; r^h is the AP's transmission bit rate; T_{bar} and T_{ba} are the transmission times of BA request and BA frames, respectively. In the same way, T^c_{rd} is expressed as

$$T^c_{rd} = T_{phy} + T^c_{mpdu} \times n^c_{pkt-rd} + T_{ba-ex} + T_{sifs} \tag{8}$$

where n^c_{pkt-rd} is the aggregation size of the client's motion-feedback data which is transmitted within the AP's TXOP via RD; we obtain it in the following.

Let R_{fb} be the rate of motion feedback (the number of motion-feedback reports per unit time). Then, the total number of motion-feedback packets (assuming that one report generates one packet) to transmit during a VR frame interval, denoted by N^c_{pkt}, is obtained as

$$N^c_{pkt} = \frac{1}{R_{fr}} R_{fb} \frac{1}{1-p} = N_{tx}(n^c_{pkt-rd} + n^c_{pkt-tx}) \tag{9}$$

where n^c_{pkt-tx} is the aggregation size of the client's motion-feedback data which is transmitted within its own TXOP. We assume that each transmission of the client's own TXOP flushes all motion-feedback packets in its queue (note in Fig. 3 that the uplink transmission does not reach the maximum aggregation size for all cases). Then, among N^c_{pkt} motion-feedback packets, those generated during the period of the AP's channel occupancy, i.e., $N_{tx}n^c_{pkt-rd}$ packets, are transmitted via RD right after the AP's transmission. Thus we obtain n^c_{pkt-rd} as

$$n^c_{pkt-rd} = \frac{R_{fb}}{N_{tx}}(T^h_{bo} + T^h_{tx}). \tag{10}$$

From Eq. (9), we express n^c_{pkt-tx} as

$$n^c_{pkt-tx} = \frac{N^c_{pkt}}{N_{tx}} - n^c_{pkt-rd} = \frac{R_{fb}}{N_{tx}} \left[\frac{R^{-1}_{fr}}{1-p} - (T_{idle} + T^h_{tx}) \right]. \tag{11}$$

The transmission time T^c_{tx} of the client is expressed as

$$\begin{aligned} T^c_{tx} &= T_{phy} + T^c_{mpdu} \times n^c_{pkt-tx} + T_{ba-ex} + T_{sifs} \\ &= T^c_{mpdu} \times n^c_{pkt-tx} + T_{ov} \end{aligned} \tag{12}$$

where T^c_{mpdu} is obtained in the same manner as Eq. (6).

In the average sense, $2T_{idle}$ is equal to the average backoff counts that both the AP and client go through before a transmission and obtained as

$$2T_{idle} = \frac{\widehat{\text{CW}}}{2} \times T_s \tag{13}$$

where $\widehat{\text{CW}}$ is the average CW of both the AP and client, and obtained from Bianchi's WLAN model [13] with p; T_s is the backoff slot time.

Finally, by applying the above equations into Ineq. (2), we rewrite it in terms of the problem variable n^h_{pkt} as

$$a_2(n^h_{pkt})^2 + a_1 n^h_{pkt} + a_0 \leq 0 \tag{14}$$

where

$$\begin{aligned} a_2 &= \frac{2R_{fb}T^h_{mpdu}T^c_{mpdu}}{N^h_{pkt}}, \\ a_1 &= T^h_{mpdu} + \frac{2R_{fb}T^c_{mpdu}(2T_{idle} + T_{ov})}{N^h_{pkt}} - \frac{R^{-1}_{fr}}{N^h_{pkt}}, \\ a_0 &= 4T_{idle} + 3T_{ov}. \end{aligned} \tag{15}$$

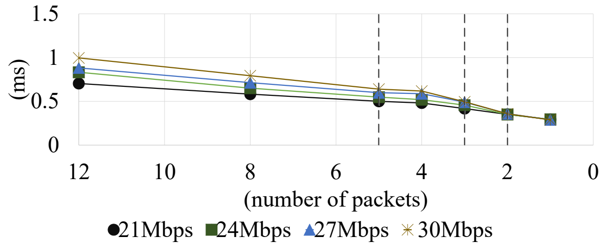
Note that a_2, a_1 and a_0 are all composed of the variables that are independent from n^h_{pkt}. Then, the minimum integer (greater than zero) of n^h_{pkt} meeting Ineq. (14), i.e., the solution of the target problem P, which we denote by n^{h*}_{pkt}, is obtained as

$$n^{h*}_{pkt} = \max\left\{ \left\lceil \frac{-a_1 - \sqrt{a_1^2 - 4a_2a_0}}{2a_2} \right\rceil, 1 \right\}. \tag{16}$$

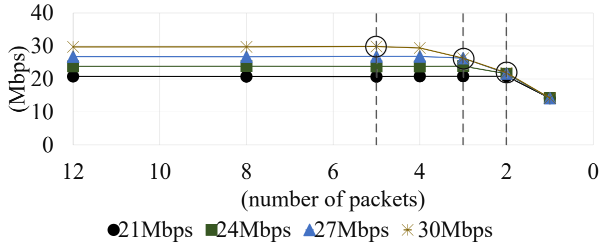
B. Performance Evaluation

For the network environment of Section IV.D, the proposed configuration scheme yields the aggregation size limit of VR video streaming as 2, 3, 3 and 5 for the video rates of 21, 24, 27 and 30 Mbps, respectively.

Fig. 6 shows the motion-feedback latency and the throughput of VR video streaming for a varying aggregation size and different video rates. As expected, the motion-feedback latency gets smaller as the aggregation size is decreased. However, when the aggregation size is decreased excessively,



(a) Motion-feedback latency



(b) Throughput of VR media streaming

Fig. 6. Motion-feedback latency and the throughput of VR video streaming for a varying aggregation size and the aggregation limits determined by the proposed scheme for different video rates

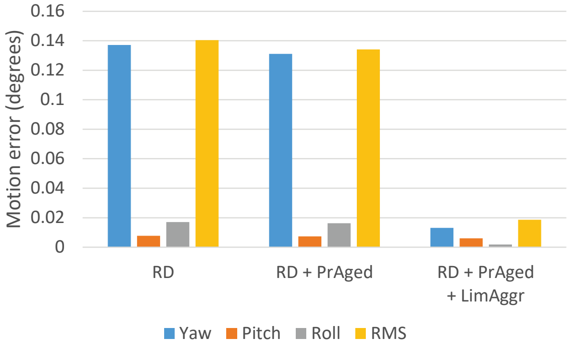


Fig. 7. Motion error for the combinations of the proposed methods based on VR video and motion-data traces captured during a VR game play

the throughput of VR video streaming gets lower than the generated video rate, thus deteriorating VR video streaming. As we formulate the target problem P in Section V.A, the goal of the proposed scheme is to find the minimum aggregation size while not degrading VR video streaming. As we see in Fig. 6(b), the aggregation size limits that the scheme finds are the minimum ones that do not decrease the throughput of VR video streaming; if we use a smaller aggregation-size limit than these, the throughput gets lowered immediately.

Finally, we evaluate the combinations of the proposed methods based on the VR video and motion-data traces captured during a play of a VR game (Great Power [14]) and show the results in Fig. 7 (LimAggr is with the optimal configuration scheme). A host PC with a tethered Oculus Rift DK1 headset runs the game in the resolution of 1280×800 (the headset's native one) and compresses rendered frames using the x264 codec library as a video stream. The resulting rate of the VR video stream is around 30 Mbps on average. The captured traces are fed into the simulator so that downlink and uplink traffic are generated accordingly. Then, upon reception of each motion-feedback report, it is compared against the true pose data of the present time to calculate an error in each axis (yaw, pitch and roll) and the root mean square (RMS) as well. The figure again shows that LimAggr in combination with the other two methods reduces motion error significantly compared to the cases without it.

VI. CONCLUSION

In this paper, we revealed that the bi-directional transmission nature of wireless VR results in deterioration of motion feedback in IEEE 802.11 WLANs. We proposed three basic methods for enhancement and showed that all enhance motion-feedback performance. We also developed the scheme to configure the best aggregation size of downlink transmission while preserving downlink performance.

ACKNOWLEDGEMENT

This work was supported in part by the Institute for Information and Communications Technology Promotion (IITP) funded by the Korea Government (MSIT) under Grant 2017-0-00650 and in part by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education under Grant 2019R1A6A1A03032119.

REFERENCES

[1] M. Abrash. *Latency - the sine qua non of AR and VR*. Accessed: May 2019. Valve Software Blog. [Online]. Available: http://blogs.valvesoftware.com/abrash/latency-the-sine-qua-non-of-ar-and-vr/

[2] O. Abari, D. Bharadia, A. Duffield, and D. Katabi, "Enabling high-quality untethered virtual reality," in *Proc. USENIX NSDI'17*, 2017, pp. 531–544.

[3] S. Sur, V. Venkateswaran, X. Zhang, and P. Ramanathan, "60 GHz indoor networking through flexible beams: A link-level profiling," in *Proc. ACM SIGMETRICS'15*, Jun. 2015, pp. 71–84.

[4] *60 GHz - kick-off of the analysis*, Electronic Communications Committee, European Conference of Postal and Telecommunications Administrations (CEPT) Doc. SRDMG(17)046, Feb. 2017.

[5] WILUS and SeoulTech, *Experiments on Wireless VR for EHT*, IEEE 802.11 EHT TIG/SG Contribution 802.11-18/1606r1, Sep. 2018.

[6] Huawei, *EHT Use Case Discussion: VR Requirement Follow Up*, IEEE 802.11 EHT TIG/SG Contribution 802.11-18/1954r21, Jan. 2019.

[7] *onAirVR*. Accessed: May 2019. Clicked. [Online]. Available: https://onairvr.io

[8] Oculus. *Asynchronous Timewarp Examined*. Accessed: May 2019. [Online]. Available: https://developer.oculus.com/blog/asynchronous-timewarp-examined/

[9] T. C. Nguyen, S. Kim, J. Son, and J.-H. Yun, "Selective timewarp based on embedded motion vectors for interactive cloud virtual reality," *IEEE Access*, vol. 7, pp. 3031–3045, 2019.

[10] Oculus. *Low-Level Sensor Details*. Accessed: May 2019. [Online]. Available: https://developer.oculus.com/documentation/pcsdk/0.8/concepts/dg-low-level-sensor/

[11] ——. *Building a Sensor for Low Latency VR*. Accessed: May 2019. [Online]. Available: https://www.oculus.com/blog/building-a-sensor-for-low-latency-vr/

[12] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications*, IEEE Std. 802.11, 2016.

[13] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE Journal on Selected Areas in Communications*, vol. 18, pp. 535–547, Mar. 2000.

[14] *Great Power Demo*. Accessed: May 2019. [Online]. Available: https://forums.oculusvr.com/developer/discussion/3507/release-great-power/