Digital Object Identifier 10.1109/ACCESS.2020.3038854

Multipath-Based HTTP Adaptive Streaming Scheme for the 5G Network

HEEKWANG KIM, AND KWANGSUE CHUNG, (Senior Member, IEEE)
Department of Electronics and Communications Engineering, Kwangwoon University, Seoul 01897, South Korea
Corresponding author: Kwangsue Chung (kchung@kw.ac.kr)

ABSTRACT Fifth-generation (5G) communication has a non-line-of-sight (non-LOS) property, in that it cannot pass through obstacles due to the characteristics of ultra-high frequencies. This can lead to a high packet loss rate, which results in loss-based congestion. As the demand for high-quality video streaming services increases, multipath-based Hypertext Transfer Protocol (HTTP) adaptive streaming has been widely studied. It has the advantage of improving stability using bandwidth aggregation and alternative transmission. The request policies of existing systems degrade bandwidth utilization by the ON-OFF traffic pattern and degrade the quality-of-experience (QoE) due to incorrect bandwidth estimation. In the 5G environment, the above problems are exacerbated by severe network asymmetry, and buffer underflow occurs due to the drastic bandwidth changes in a non-LOS environment. We propose a multipath-based transmission scheme to solve the problem of HTTP adaptive streaming in a 5G environment. The proposed scheme presents a multipath-based collective segment request policy for accurate aggregated bandwidth estimation and bandwidth utilization improvement. The aggregated bandwidth estimation measures both block-based and segment-based bandwidth estimation values to improve network responsiveness and stability. We propose a rate adaptation scheme to improve bandwidth utilization and QoE. We also present a segment scheduler to solve the out-of-order problem in multipath-based transmission, and propose an offloading control scheme based on a partial-segment-based request policy to prevent buffer underflow due to sudden bandwidth reduction in the non-LOS 5G environment. Quantitative results from simulations suggest that our scheme solves the shortcomings of the existing solutions improving both bandwidth utilization and QoE.

INDEX TERMS 5G mmWave, bandwidth utilization, HTTP adaptive streaming, multipath environment, segment scheduling policy.

I. INTRODUCTION
Recently, fifth-generation (5G) mobile communication networks have been developed and commercialized. The design goals of 5G are to provide a high data rate, high mobility, and low latency. Notably, 3GPP standardized 5G new radio (NR) technology for these design goals. To achieve these goals, 5G mainly uses the millimeter wave (mmWave) signal of ultra-high frequencies. mmWave signals have the characteristic of generating frequent packet loss in a non-line-of-sight (non-LOS) region because of their short reach, strong linearity, and weak obstacle penetration. [1]. These mmWave features have a significant impact on the transport layer. These factors can lead to a high packet loss rate through the mmWave links, which results in TCP throughput degradation due to additive increase multiplicative decrease congestion control.

Due to the development of mobile communication technology (e.g., 3G, LTE, 5G, and Wi-Fi) and the spread of high-performance mobile devices, the demand for multimedia streaming services has increased. To provide an efficient video streaming service, HTTP (Hypertext Transfer Protocol) adaptive streaming is actively being studied [2]. HTTP adaptive streaming controls video quality based on various context information in network environments. Many studies have adopted control-theoretic approach to context-aware adaptive streaming in a changing network [3]–[5]. Commercial HTTP adaptive streaming services include Microsoft’s Smooth Streaming [6], Apple’s HTTP Live Streaming [7], and Adobe’s Dynamic Streaming [8]. Also, Dynamic Adaptive Streaming over HTTP has been standardized by the Moving Picture Experts Group [9]. Commercialized services based on these systems include YouTube, Netflix, and

The associate editor coordinating the review of this manuscript and approving it for publication was Dipankar Deb,
keeping the buffer stable.

posed scheme improves bandwidth utilization and QoE while scheme using both LTE and mmWave scenarios. The pro-

tion in the 5G NR environment. We evaluated the proposed policy to prevent buffer underflow due to bandwidth reduc-

offloading scheme based on a partial-segment-based request level to improve bandwidth utilization. Finally, we present an algorithm for the proposed segment request policy, which determines the number of request segments needed to be downloaded sequentially when multiple segments are requested from multiple paths. We also propose a rate adap-

tion algorithm to prevent the reordering problem in multipath environments. This allows segments to be downloaded sequentially when multiple segments are requested from multiple paths. We also propose a rate adap-

tation algorithm for the proposed segment request policy, which determines the number of request segments needed to manage the buffer stably and selects the quality level to improve bandwidth utilization. Finally, we present an offloading scheme based on a partial-segment-based request policy to prevent buffer underflow due to bandwidth reduc-

tion in the 5G NR environment. We evaluated the proposed scheme using both LTE and mmWave scenarios. The proposed scheme improves bandwidth utilization and QoE while keeping the buffer stable.

Hulu [10]. Each service guarantees the quality of experience (QoE) using different quality control methods [11].

As the demand for high-quality video streaming services such as ultra-high definition, virtual reality, and augmented reality increases, service providers can use multipath-based transmission to provide better QoE [12]. Multipath-based schemes download through multiple paths simultaneously, it can provide higher available bandwidth than single-path-based schemes. It also improves stability by allowing redundant paths to be continuously transmitted when problems occur on the primary path. In recent years, many studies have been conducted on multipath-based HTTP adaptive streaming schemes [13]–[19].

In conventional multipath-based HTTP adaptive streaming systems, the client simultaneously requests and downloads video segments of lower quality than the available bandwidth through multiple paths [20]. The conventional request policy downloads video segments using the ON-OFF traffic pattern to manage the buffer reliably, but this reduces bandwidth utilization. Furthermore, the bandwidth is underestimated according to the request policy, because existing bandwidth estimation methods predict the aggregate available bandwidth when a playable segment is received on multiple paths. In addition, out-of-order problems occur because the download times of segments received through multiple paths are different. It manages the buffer unstably, which causes the playback to be interrupted. These problems degrade the user’s QoE. The problems of multipath-based conventional HTTP adaptive streaming systems are exacerbated by severe network asymmetry in the 5G NR environment. In addition, when bandwidth is significantly decreased in a non-LOS 5G environment, the existing HTTP adaptive streaming systems do not respond quickly, resulting in buffer underflow.

In this paper, we present a multipath-based HTTP adaptive streaming scheme for a 5G NR environment. First, we propose a collective segment request policy to measure the aggregate available bandwidth accurately. The proposed policy requests multiple segments simultaneously for each path based on a group during the request interval. The purpose of this method is to continuously download multiple segments, reducing the frequency of OFF periods and improving the bandwidth utilization of each path. Subsequently, we propose a segment scheduling scheme to prevent the reordering problem in multipath environments. This allows segments to be downloaded sequentially when multiple segments are requested from multiple paths. We also propose a rate adaptation algorithm for the proposed segment request policy, which determines the number of request segments needed to manage the buffer stably and selects the quality level to improve bandwidth utilization. Finally, we present an offloading scheme based on a partial-segment-based request policy to prevent buffer underflow due to bandwidth reduc-

tion in the 5G NR environment. We evaluated the proposed scheme using both LTE and mmWave scenarios. The proposed scheme improves bandwidth utilization and QoE while keeping the buffer stable.

The rest of this paper is organized as follows. In Section II, we provide an overview of the multipath-based HTTP adaptive streaming system and review the characteristics of the 5G NR environment. In Section III, we present the proposed multipath-based HTTP adaptive streaming system. In Section IV, we evaluate the performance of the proposed scheme through simulations. Finally, in Section V, we conclude the paper and discuss future work.

II. BACKGROUND AND RELATED WORK

A. FEATURES OF 5G NR

Overall, mmWave networks are expected to reach an order of magnitude increase in throughput over current systems owing to the larger bandwidth available, but this improvement is associated with more troublesome propagation conditions. The main differences between the mmWave and LTE channels are as follows. The path loss during LOS to non-LOS transition is deeper for mmWave than for current systems. The mmWave networks are small cell networks, and mmWave links are sensitive to blockages from buildings, human bodies, moving obstacles, and so on. Thus, the transitions from LOS to non-LOS for mmWave connections are much more frequent than those in LTE. Furthermore, given the short range of mmWave communication, it is more likely for mmWave links than for LTE links to experience an outage (where no signal is received) because of shadowing [21]. Fig. 1 shows the mmWave throughput changes in a scenario where there is a building between the user equipment (UE) and gNode B (gNB).

![FIGURE 1. mmWave throughput changes.](image-url)
mmWave degrades the QoE of multipath-based video streaming schemes.

B. OVERVIEW OF MULTIPATH-BASED HTTP ADAPTIVE STREAMING

Depending on the number of network paths used for downloading the video segments, HTTP adaptive streaming schemes have been classified as single-path-based HTTP adaptive streaming approaches and multipath-based HTTP adaptive streaming approaches. For continuous video playback, a single-path-based approach uses a single network path for downloading video segments by selecting the appropriate video quality in a changing network. However, due to the limited bandwidth, the single-path-based approach has difficulty in continuously playing high-quality video. The multipath-based approach using multiple network interfaces has been developed to solve the problem of QoE degradation caused by the limited bandwidth of the single-path-based approach [13]–[20].

The multipath-based HTTP adaptive streaming scheme is shown in Fig. 2. The multipath-based scheme downloads video segments by requesting them simultaneously through multiple network interfaces, which provide higher bandwidth and enable concurrent higher quality video segments within the same time as required by the single-path-based scheme. Therefore, the client of the multipath-based scheme obtains a higher network bandwidth following multiple paths to prevent QoE degradation. However, because segments are downloaded through multiple paths simultaneously, studies on segment request policies for multipath environments and available bandwidth measurement methods according to request policy are required. In addition, research on techniques for solving the out-of-order problem, which is a typical problem of multipath transmission technology, is required.

C. REQUEST POLICY OF MULTIPATH-BASED HTTP ADAPTIVE STREAMING SCHEMES

Fig. 3 illustrates the request policy in conventional HTTP adaptive streaming for managing the playback buffer stably. HTTP adaptive streaming operates in a buffering state and a steady state. At the beginning of streaming, HTTP adaptive streaming operates in a buffering state that requests the lowest quality to fill the playback buffer quickly. In the buffering state, when the requested segment is downloaded, the next segment is continuously requested. When the buffer reaches a threshold, it operates in the steady state, and periodically requests each segment for stable buffer management. In the steady state, the ON-OFF pattern is repeated by the download section due to the difference between the bandwidth and the bitrate. The ON period is the time when the client downloads the segment, and both buffer consumption and charging occur. The OFF period is a section that consumes the buffer after downloading all requested segments. The OFF periods are periodically generated by existing rate adaptation algorithms [22], [23]. In the 5G environment, the client has a long OFF period due to the large difference between the available bandwidth and the bitrate.

The existing request policies for multipath-based HTTP adaptive streaming are shown in Fig. 4. Fig. 4(a) shows a segment-based request policy in which different segments are requested for each path. Because the playable segments are downloaded through multiple paths, the available bandwidth of each path can be accurately measured using a single-path-based bandwidth measurement. However, because this request policy does not aggregate multipath bandwidth, a low-quality segment is requested, and frequent quality changes occur because the quality requested for each path is different due to network asymmetry. Fig. 4(b) shows a partial segment-based request policy in which a single segment is simultaneously requested across multiple paths. The advantages of this approach over segment-based policies are that the client can request higher-quality segments by aggregating the multipath bandwidth and avoiding the reordering problem. However, there is a problem of incorrectly measuring the available bandwidth. The HTTP adaptive streaming estimates the available bandwidth when one playable segment is received. Because the partial segment-based request policy delays the download completion time of one path due to network asymmetry, the aggregated available bandwidth is measured to be less than the sum of the available bandwidths of each path. This degrades bandwidth utilization and user QoE.

Gouache et al. [13] proposed a distributed and adaptive HTTP streaming (DAHS) scheme that includes a segment-based request policy as shown in Fig. 4(a). DAHS improves the video playback quality of the client using the existing distributed server in the HTTP adaptive streaming system. DAHS requests the same segment at the same time, adapting to the network situation of each path from the distributed HTTP servers. The available bandwidth of each
path is measured, and different qualities of segments in the same order are downloaded based on the measured available bandwidth. By playing the highest-quality segment among the received segments, the client can play higher media quality when only a single server is used. However, this scheme wastes bandwidth because the same segment is repeatedly received through each path with a low-quality segment that cannot be played. Moreover, DAHS provides a lower quality video than other multipath-based schemes because it does not use the aggregated bandwidth.

Chen et al. [14] proposed a scheme for adjusting the sizes of partial segments, called the MSPlayer. This scheme uses Wi-Fi and LTE networks to download video segments simultaneously from different servers. The available bandwidth is estimated using the harmonic-mean to account for bandwidth variation in wireless environments. The size of the video segment to be requested for each path is determined by dynamic scheduling based on the ratio between estimated bandwidth of each path to achieve the same download time for both paths. However, a drawback is that MSPlayer takes a long time to measure the bandwidth accurately in 5G networks with high bandwidth due to the harmonic-mean-based estimation method.

Zhou et al. [18] proposed a block-based rate adaptation scheme called control-theoretic rate adaptation (CTRA). This scheme adopts fixed block lengths and determines block-based quality according to network bandwidth and buffer occupancy. The aggregated bandwidth is accurately measured because the CTRA requests segments simultaneously through a segment-based request policy via multiple paths. However, CTRA is less responsive to network changes than other methods because of the fixed and long block length, which results in buffer underflow when the bandwidth is drastically reduced in the 5G NR environment.

In the previous work, we proposed a segment scheduling scheme for efficient bandwidth utilization of HTTP adaptive streaming in the multipath environment [19]. We proposed the collective segment-based request policy and a quality control scheme to improve the bandwidth utilization in multipath environments using LTE and WiFi simultaneously. We simulated to evaluate the performance of the previous scheme in the 5G NR environment, and the results are shown in Fig. 5. Since the previous work only considered bandwidth utilization in the rate adaptation scheme, it shows that unnecessary quality changes occur due to frequent network changes of 5G NR according to the video quality configuration. In addition, when entering the non-LOS environment after 80 seconds, the download completion time of the segment being downloaded is delayed due to the rapid bandwidth reduction of 5G NR. It causes buffer underflow and degrades the QoE.
for improved bandwidth utilization is discussed. Finally, an offloading control scheme is presented.

A. ARCHITECTURE OF THE PROPOSED SYSTEM

The client architecture of the proposed multipath-based HTTP adaptive streaming system is shown in Fig. 6. In the proposed system, the client consists of seven modules.

- The network monitor estimates the available bandwidth of each route and measures the integrated available bandwidth using the predicted bandwidth of each path.
- The buffer monitor periodically monitors the client’s buffer occupancy and rearranges the downloaded segments in each path to deliver them to the playback buffer.
- The parser analyzes the XML-based manifest file received from the server, delivers segment information such as the quality of the video contents and addresses the quality controller module.
- The quality controller module determines the quality and the number of segments to request for each path to improve the user QoE and bandwidth utilization considering the network information and buffer status.
- The request controller calculates the length of the request interval to improve responsiveness and determines the range of segments to request during the request interval to prevent buffer underflow and overflow.
- The segment scheduler distributes the order of request messages to each path to alleviate the reordering problem in a multipath environment.
- The offloading controller checks the download status of each path and requests an alternate transmission using a partial-segment-based request policy to prevent performance degradation due to any drastic decrease in bandwidth in a 5G environment.

The proposed scheme consists of a segment request policy, bandwidth measurement, determining the quality and number of request segments, reordering request messages, and offloading control. The segment request policy is to request multiple segments for each path simultaneously using a collective segment request policy to improve bandwidth utilization. To improve network responsiveness and manage the buffer stably, the length of the request interval and the range of request segments are determined. To adapt to frequent and rapid bandwidth changes in the 5G environment, the aggregated bandwidth estimation reflects the block-based bandwidth prediction and the segment-based bandwidth prediction at the same time to measure the integrated available bandwidth. Using the measured available bandwidth and request segment range, the quality during the request interval, the number of segments to request for each path, and the order of the request messages are determined. The offloading control operation redistributes the segment being downloaded using a partial-segment-based request policy to minimize the download completion delay time by measuring the download state of each path during the request interval.

B. COLLECTIVE SEGMENT REQUEST POLICY WITH OFFLOADING CONTROLLER

In multipath based HTTP adaptive streaming, a request policy is an important research area that affects performance, such as by measuring the available bandwidth and reordering. Existing multipath-based HTTP adaptive streaming schemes download segments using segment-based request policies and partial-segment-based request policies. However, segment-based request policies cause a reordering problem due to asymmetry in 5G and LTE networks, which deteriorates the QoE. The partial-segment-based request policy has a problem in that download completion time is delayed according to how a segment is distributed to each path, and, accordingly, the aggregated bandwidth is underestimated. If the quality is selected using the low measured bandwidth, the length of the OFF period becomes longer, and thus the bandwidth utilization decreases.

To solve these problems, we propose a collective segment request policy that requests multiple segments as a group during the request interval. Fig. 7 shows the proposed collective segment request policy. It operates in a buffering state at the beginning of streaming, and when the buffer reaches the threshold, it is converted into a collective state. In the buffering state, the client continuously requests segments of the lowest quality to fill the buffer quickly. In the collective state, it operates by a collective request policy to manage the buffer stably. The collective request policy periodically requests one or more playable segments in a block form for each path in every request interval. The request interval is the period from when the client requests segments until it completes the download from the server. The proposed request policy can measure bandwidth accurately regardless of the performance of other paths by distributing playable segments to each path. In addition, by requesting multiple segments at once in block form, the number of request messages transmitted to each path is reduced. This reduction of the request events reduces the OFF period of the entire
streaming and improves bandwidth utilization, so that high-quality segments can be requested. However, the collective segment request policy suffers from the reordering problem because it is a segment-based request policy. To solve the reordering problem, the segment scheduling scheme to be described in Subsection D is used to alleviate the reordering problem according to the order of the segments requested in each path. In the 5G environment, if all previous request segments are not downloaded during the request period due to a rapid decrease in bandwidth, to alleviate the problem of order rearrangement, the delayed segments are preferentially requested through the LTE path with a relatively stable network state in the next request interval. The delay in download means that the 5G network is unstable. Therefore, regardless of the throughput of 5G NR, delayed segments are received through the stable LTE network path to alleviate the reordering problem.

The length of the request interval is an important factor in improving network responsiveness and bandwidth utilization. When the request interval is set long, the client does not respond quickly to rapid changes in the network, but it reduces the number of request events and aggregates the bandwidth for a long period to improve bandwidth utilization. On the other hand, if the request interval is set short, network responsiveness improves, but OFF periods occur frequently due to numerous request events, thereby reducing bandwidth utilization. To solve the trade-off relationship between bandwidth utilization and network responsiveness, the proposed scheme adaptively determines request interval according to the network status. The condition for judging the network status is measured using the buffer filling rate, as in (1).

\[
B_c[i] = \frac{N_{\text{actual}}[i - 1]}{N_{\text{total request}}[i - 1]} \tag{1}
\]

\(N_{\text{actual}}\) denotes the number of segments downloaded in the previous interval, and \(N_{\text{total request}}\) denotes the number of segments requested in the previous request interval. That is, the buffer filling rate is calculated using the ratio of the number of downloaded segments to the number of requested segments during the request period. If all requested segments are downloaded in the previous request interval, the buffer filling rate is 1. This means that the previously aggregated bandwidth has been accurately estimated or no network changes have occurred. In this case, to improve bandwidth utilization, the length of the request interval \(T[i]\) is increased by the segment duration \(\tau\), as shown in (2).

\[
T[i] = T[i - 1] + \tau, \quad \text{if } B_c[i] = 1 \tag{2}
\]

On the other hand, if the buffer filling rate is less than 1, it means that all segments requested in the previous request interval have not been downloaded. This means that the previously estimated bandwidth is unreliable, or the bandwidth has changed dramatically. The length of the request interval is reduced to measure the aggregated bandwidth accurately, as shown in (3).

\[
T[i] = [B_c[i] \times T[i - 1]], \quad \text{if } B_c[i] < 1 \tag{3}
\]

The length of the final request interval \(T_{\text{request}}[i]\) is determined to periodically request the request interval by reflecting the delay time \(T_{\text{dy}}\) of the previous request interval, as in (4) and (5).

\[
T_{\text{request}}[i] = \min(T[i] - T_{\text{dy}}, T_{\text{max}}[i]) \tag{4}
\]

\[
T_{\text{dy}} = T_{\text{complete}}[i - 1] - T_{\text{request}}[i - 1] \tag{5}
\]

\(T_{\text{complete}}\) denotes the time when a segment currently being downloaded are received and \(T_{\text{max}}[i]\) denotes the maximum request interval, as in (6). The maximum request interval is the time required to request one segment of the highest quality in the path with lower throughput.

\[
T_{\text{max}}[i] = \left\lceil \frac{R_{\text{max}}}{Th[i, 1 - j]} \right\rceil \tag{6}
\]

\(R_{\text{max}}\) denotes the bitrate of the server’s maximum quality. After determining the request interval, the client determines the range of the total request segments based on the current buffer occupancy, to prevent buffer underflow and overflow, as in (7).

\[
N_{\text{th min}}[i] - N_{\text{dy}} < N_{\text{total request}}[i] < N_{\text{th max}}[i] - N_{\text{dy}} \tag{7}
\]

\(N_{\text{th min}}[i]\) denotes the minimum threshold to prevent buffer underflow, \(N_{\text{dy}}\) denotes the number of segments that were not downloaded in the previous request interval and \(N_{\text{th max}}\) denotes the maximum threshold to prevent buffer overflow. In (8), the minimum threshold is determined using the estimated buffer state during the next request interval. The minimum threshold is set to 2 to ensure that at least one segment can be requested for each path during the request interval.

\[
N_{\text{th min}}[i] = \max \left(\frac{T_{\text{request}}[i] - \overline{B}_{\text{OoO}}}{\tau}, 2\right) \tag{8}
\]

\(\overline{B}_{\text{OoO}}\) denotes the playback time of segments that have been downloaded in the previous request interval but cannot be played due to the reordering problem. The maximum threshold is calculated using the target buffer \(B_{\text{target}}\), as in (9).

\[
N_{\text{th max}}[i] = \max \left(\frac{B_{\text{target}} - B[i] - \overline{B}_{\text{OoO}} + T_{\text{request}}[i]}{\tau}, N_{\text{th min}}[i]\right) \tag{9}
\]
The maximum threshold means the number of segments to be downloaded to guarantee the minimum threshold and maintain the buffer occupancy.

In the conventional HTTP adaptive streaming system, when the available 5G NR bandwidth decreases drastically in a non-LOS environment, the completion time of the downloading segment is delayed. Due to the time delay, the LTE path, which has already been downloaded, also interrupts the request event. Buffer underflow occurs because the next request interval cannot proceed. This degrades the performance of the existing HTTP adaptive streaming system.

To solve this problem, we propose an offloading controller using an alternative transmission of multipath scheme, based on the partial segments, that includes a mechanism to reduce delays due to sudden changes in available 5G NR bandwidth. Fig. 8 shows the concept of the proposed offloading controller. The offloading controller operates when all the allocated request segments in one path are downloaded. The LTE path that downloads all request segments checks the status of the 5G NR and offloads the remaining data via 5G NR, based on the partial request policy, if the completion time on 5G NR exceeds the request interval.

Measurements of the path status are used to predict the download completion time using the segment-based throughput of each path, as in (10).

\[
R[i] \times \tau - D_{\text{complete}}^{i-j} > T_{\text{remain request}}
\]  

\(D_{\text{complete}}^{i-j}\) denotes the completed download size, \(Th_{\text{seg}}[k - 1, i, 1 - j]\) denotes the segment-based available bandwidth of \(k - 1\)th segment in \(i\) request interval on 5G NR, and \(T_{\text{remain request}}\) denotes the remaining time in the request interval. If the completion time is longer than the remaining request interval, the data to be downloaded on 5G NR is offloaded to the LTE path. The offloading size is determined proportionally to the bandwidth of each path to reduce the difference in download completion time, as shown in Fig. 9. It alleviates the underestimation problem of the aggregated bandwidth in partial-segment-based request policy.

If the bandwidth of the LTE path is sufficient, the remaining data are distributed to the relatively stable LTE path as much as possible, as in (11).

\[
S_{\text{partial}}[i, j] = \max(w[i, j] \times S_{\text{remain}}, Th_{\text{seg}}[i, j] \times T_{\text{remain request}})
\]

In (11), \(w\) denotes the bandwidth ratio between LTE and 5G NR, as in (12), and \(S_{\text{remain}}\) denotes the remaining data size.

\[
w[i, j] = \frac{Th_{\text{seg}}[i, j]}{Th_{\text{seg}}[i, j] + Th_{\text{seg}}[i, 1 - j]}
\]

As shown in Fig. 10, by using the HTTP range request method included in the HTTP/1.1 standard, partial segments are requested simultaneously through multipath.

\[
S_{\text{partial}} = \max(w[i, j] \times S_{\text{remain}}, Th_{\text{seg}}[i, j] \times T_{\text{remain request}})
\]

For example, the HTTP rage message header requested to the server connected to the 5G NR path is shown in Fig. 11(a). A part of the “seg1.m4v” segment can be requested in byte units in the Range field of the request message. As a response, the byte of the data downloaded in the Content-Range field is displayed as shown in Figure 11(b). As a result, the same segment sequence number is assigned to each path, but the partial segments to be downloaded through each path are described in the Range field.

C. AGGREGATED BANDWIDTH ESTIMATION AND RATE ADAPTATION ALGORITHM

In a 5G environment, the simultaneous download method through multiple paths degrades QoE by incorrectly measuring the bandwidth according to request policies. To solve this problem, the proposed scheme predicts the available...
bandwidth of each path by simultaneously reflecting the block-based available bandwidth in \( i \)th request interval on \( j \) path \( Th_{\text{block}} [i, j] \) and segment-based available bandwidth \( Th_{\text{seg}} [i, j] \) to improve network responsiveness and stability by the exponential weighted moving average (EWMA), as shown in (13).

\[
Th[i, j] = (1 - \alpha) Th_{\text{block}}[i, j] + \alpha Th_{\text{seg}}[i, j] \tag{13}
\]

\( \alpha \) is the parameter for improving the stability or responsiveness. The \( \alpha \) is closer to 1, the more segment-based bandwidth is reflected. It means that the estimated available bandwidth of each path fluctuates significantly. In this paper, in order to mitigate the frequent bandwidth changes of 5G NR, \( \alpha \) is set to 0.3. The block-based available bandwidth is calculated by dividing the total data size of the segments downloaded during the previous request interval by the time required for download, as shown in (14). By estimating the throughput while downloading multiple segments, it is possible to improve the stability of network bandwidth fluctuations in a 5G environment by smoothing.

\[
Th_{\text{block}}[i, j] = \frac{R[i - 1] \times \tau \times N[i - 1, j]}{t_{\text{down}}[i - 1, j]} \tag{14}
\]

The segment-based available bandwidth is the throughput for the last segment received in each path during the previous request interval. It is calculated by dividing the size of the last segment by the downloading time as shown in (15). It improves network responsiveness in the non-LOS environment of 5G.

\[
Th_{\text{seg}}[k, i, j] = \frac{R[i - 1] \times \tau}{t_{\text{down}}[k, i, j]} \tag{15}
\]

The proposed request policy requests playable segments on each path as in the existing segment-based request policy. Unlike in the partial-segment-based request policy, the available bandwidth can be estimated using the segments received in each path. It can be expressed as the sum of the estimated bandwidth of each path, as in (16).

\[
Th_{\text{agg}}[i] = \sum Th[i, j] \tag{16}
\]

Using the estimated available bandwidth and the range of the number of request segments in (7), the quality of collective segments and the number of segments to request for each path are determined to improve bandwidth utilization and QoE. Bandwidth utilization \( U \) is the ratio of the size of data to download the actual segments to the total size of data \( S_{\text{avail}} \), as in (18), which can be downloaded during the request interval. The bandwidth utilization function is shown in (17).

\[
U = \frac{R[i] \times \tau \times (N_1 + N_{1-j})}{S_{\text{avail}}} \tag{17}
\]

\[
S_{\text{avail}}[i] = Th_{\text{agg}} \times T_{\text{request}}[i] \tag{18}
\]

In general, in the HTTP adaptive streaming, an improvement in QoE as it relates to video quality means higher average video quality and smaller quality changes. Therefore, the QoE function can be defined using the difference between the bitrate at which the quality change occurs and the average bitrate of segments requested in the entire video stream, as shown in (19).

\[
QoE = \sum_{i=1}^{l} q(R[i]) - \sum_{i=1}^{l} |q(R[i]) - q(R[i - 1])| \tag{19}
\]

Finally, the proposed scheme determines the pairs of quality and number of segments to request for each path that maximizes bandwidth utilization and QoE as shown in (20).

\[
\pi = \{N_k, N_{1-k}, R[i]\}, \max_{\pi[i]} [QoE \times \beta U], \tag{20}
\]

\[
\text{Subject to } N_k^{\text{max}}[i] - N_k^{\text{delay}} \leq N_k + N_{1-k} \leq N_k^{\text{max}}[i] - N_k^{\text{delay}} \leq R[i](N_k + N_{1-k}) < S_{\text{avail}}[i] \tag{21}
\]

\[
R[i] \times N_k < Th[i, j] \times T_{\text{request}}[i] \tag{22}
\]

\[
Th[i, j] < R[i] \tag{23}
\]

\( \beta \) denotes the margin factor for the OFF period while delivering the request message.

**D. SEGMENT SCHEDULING**

The proposed request policy suffers from the reordering problem because it downloads multiple segments at the same time through multiple paths during the request interval. The reordering problem means that if the segment to be played first is not received until the later segments are downloaded, the later segments are stored without being played until the previous segments are downloaded. This problem is caused by the asymmetry of the network in a multipath environment. It is exacerbated in an environment that uses 5G and LTE networks simultaneously. To solve this issue, we propose a segment scheduler requesting segments for each path proportionally, based on the expected download time, as shown in Fig. 12.

The expected download time is calculated using the determined quality and estimated throughput of each path by indirectly reflecting the network congestion information, as in (21).

\[
e T_{\text{down}}^j = \frac{R[i] \times \tau}{Th[i, j]} \tag{21}
\]

To prevent the out-of-order problem, segments are requested using the ratio of the expected download time,
as in (22).

\[ e_{T_{\text{ratio}}} = \left\lfloor \frac{eT_{\text{down}}^{1-j}}{eT_{\text{down}}^j} \right\rfloor, \text{if} \quad eT_{\text{down}}^{1-j} > eT_{\text{down}}^j \quad (22) \]

The requested segment is prioritized on a path with a fast download time. It is stored in order in a request buffer for each 5G NR and LTE path according to the download time ratio. It is possible to request segments to be played first on the 5G NR path and to solve the out-of-order problem by requesting segments in proportion to the download time.

A flowchart of the proposed multipath-based HTTP adaptive streaming system is shown in Fig. 13. In the collective state, the proposed scheme checks the status of another path when all segments requested during the request interval are received in each path. If another path does not complete the download within the request interval, the offloading controller is used to reduce the delay in the non-LOS environment. When all requested segments on multiple paths are downloaded and the request interval ends, the proposed scheme determines the next request interval and estimates the network bandwidth. Then, the video quality and the number of request segments are selected to improve bandwidth utilization and QoE. Finally, the segment scheduler prevents the out-of-order problem.

IV. PERFORMANCE EVALUATION

To evaluate the performance of the proposed scheme, we implemented a simulation environment using the ns-3 network simulator [24]. In this section, we present and discuss the results of the simulation.

A. SIMULATION SETUP

Fig. 14 shows the simulation topology. The client downloads the segments using both 5G mmWave and LTE wireless network interfaces simultaneously. The average throughput when transmitting over mmWave is 2.3 Gbps, while that over LTE is 72 Mbps. The two servers provide 14 different pre-encoded qualities of video: 1000, 2000, 4000, 6000, 10000, 12000, 16000, 20000, 23000, 26000, 100000, 130000, 160000, and 200000 Kbps. \( \alpha \) is set to 0.3; \( \beta \) is set to 0.9; \( \tau \) is set to 2 s, and \( B_{\text{target}} \) is set to 20 s.

For comparison, we implemented bandwidth aggregation models and an independent model in multipath transmission environments. We implemented DAHS [13] as the independent model, Conventional-Partial and MSPlayer [14] as the bandwidth aggregation models, and CTRA [18] as the block-based model. The simulation was run in three scenarios. Fig. 15 shows the bandwidth changes according to each scenario. Scenario 1 is a network condition without buildings. Scenario 2, there is one building between the UE device and gNode B, which causes a drastic bandwidth reduction in mmWave. Finally, Scenario 3 is a condition in which many buildings are located between the UE device and gNode B.

B. EVALUATION OF BANDWIDTH ESTIMATION SCHEME

Fig. 16 shows the aggregated throughput according to the segment request policy and bandwidth estimation schemes in Scenario 1. The Conventional scheme requests segments of the same size on each path regardless of the mmWave and LTE network conditions. Thus, mmWave must wait unnecessarily until the slower LTE connection completes the download. As a result, the bandwidth was underestimated. The MSPlayer client downloads partial segments on multiple
paths. However, the size of the partial segment to be requested on each path is adaptively adjusted, unlike in the Conventional scheme. The MSPlayer scheme uses a harmonic-mean for smoothing method, as shown in (23).

\[
Th_{avg} = \frac{\sum_{k=1}^{n} 1/Th_{agg}[k]}{n}
\]  

(23)
In (23), \( n \) denotes the number of bandwidth samples. The harmonic mean responds slowly to frequent bandwidth changes; this approach has the lowest average and maximum variations compared to the other bandwidth estimation schemes. The available bandwidth is underestimated due to the conservative estimation scheme of the harmonic mean. CTRA estimates the bandwidth closest to the aggregated bandwidth using the segment-based estimation method. However, because CTRA lacks consideration of the wireless environment, the estimated bandwidth changes frequently without smoothing. The proposed scheme achieves a similar estimation to that of the aggregated bandwidth and displays better smoothing performance. The proposed collective-based bandwidth estimation method ensures that the available bandwidth is more accurately measured than in other schemes to improve the QoE.

The estimated bandwidth in Scenario 2 is shown in Fig. 17. The Conventional and MSPlayer methods yield a bandwidth that is not fully used, and these schemes are not affected by the sudden bandwidth changes of mmWave. CTRA responds to the bandwidth reduction of mmWave. However, this response takes a long time, because the time to download the requested segments runs beyond the previous request interval, and this amount is increased by the reduced bandwidth and...
FIGURE 19. Comparison of video quality and buffer occupancy in Scenario 1.
FIGURE 20. Comparison of video quality and buffer occupancy in Scenario 2.
FIGURE 21. Comparison of video quality and buffer occupancy in Scenario 3.
FIGURE 22. Video quality and buffer occupancy of the proposed scheme without an offloading controller.

long block length. Therefore, CTRA has less responsiveness than the other methods. However, the proposed scheme shows better responsiveness and accurate bandwidth estimation because the delay is reduced by the proposed request policy and the offloading controller.

Fig. 18 shows the estimated bandwidth in Scenario 3. CTRA estimates high bandwidth by responding less to frequent and sudden network changes than other methods, due to the longer block length. However, the proposed scheme responds to three sudden changes in bandwidth and adapts faster than CTRA to increased bandwidth.

C. EVALUATION OF RATE ADAPTATION ALGORITHM

In a video streaming system, QoE enhancement means the seamless playback of high-quality video content. To evaluate the performance of the proposed rate adaptation algorithm, we compare with the request quality and buffer level changes in all scenarios. The video quality and buffer level changes in Scenario 1 are shown in Fig. 19. The Conventional and MSPlayer methods do not fully use the bandwidth of the mmWave path due to bandwidth underestimation. Even the MSPlayer requests lower-quality video than Conventional video quality until 200s, because it takes a long time to increase the video quality with the harmonic mean method. In DAHS, segments downloaded over the mmWave network are played while there is no change in available bandwidth. However, frequent quality changes occur due to inaccurate bandwidth estimations that do not consider the 5G NR environment. CTRA and the proposed scheme stably request videos of the highest quality. In CTRA, the OFF period, which consumes only the buffer due to a long request period, periodically occurs for a long period time, and the buffer is charged in two stages because the reordering problem is not considered. In these two steps, the buffer is charged in such a way that the segment requested by the mmWave path is first charged by the received segment, and when the segment requested by the LTE path is received, the remaining segments that have already been downloaded by mmWave are charged. The proposed scheme takes more time than CTRA to reach the maximum quality and improve the QoE by preventing sudden quality changes at the beginning of streaming. However, the proposed scheme keeps the buffer stable and does not cause the problem of reordering, so that the buffer is charged and consumed only once during the request cycle.

Fig. 20 shows the video quality and buffer level changes in Scenario 2. The MSPlayer requests a lower-quality video than that in Scenario 1 during sudden bandwidth reduction and does not switch the quality. CTRA requests segments of maximum quality, but because of its low responsiveness, it reacts slowly to bandwidth changes, resulting in buffer
underflow for a long time. After buffer underflow, there is no mechanism for adjusting to the frequent bandwidth changes of mmWave, so buffer underflow occurs continuously without filling the buffer. On the other hand, the proposed scheme minimizes the delay in download completion time through the proposed request policy and offloading scheme. Besides, our scheme responds to network changes quickly through adaptive request interval adjustment based on the buffer filling rate and measures the available bandwidth accurately. Also, since multiple segments are requested simultaneously according to the buffer status, the buffer is stably maintained even in the non-LOS environment.

The video quality and buffer level changes in Scenario 3 are shown in Fig. 21. CTRA does not change the quality due to the block-based request policy, but it maintains a low buffer occupancy due to the problem of reordering. This causes a buffer underflow for a short time when the user enters the non-LOS area. Additionally, the proposed scheme maintains the buffer stably and requests the maximum quality because of the shorter request interval than that for the block length of CTRA. Other schemes in this scenario had performances similar to those in previous scenarios.

D. EVALUATION OF THE REQUEST SCHEDULER AND OFFLOADING CONTROLLER

To evaluate the request scheduler and offloading controller, their performance was compared with that of CTRA in requesting multiple segments in Scenario 2, in which a sudden bandwidth change occurred. Fig. 22 shows the request quality and buffer occupancy of the proposed scheme without the offloading controller. Buffer underflow occurs due to the increased delay compared to that in the previous experiment.

Fig. 23 shows the time at which the request message is transmitted for each scheme. CTRA has a long delay because the request event does not occur until all previous request segments are downloaded. In the case of the proposed scheme without an offloading controller, the delayed segment among the requested segments in the previous request interval has a shorter delay time than in CTRA because it is downloaded first in the next request interval. The proposed scheme with an offloading controller has the shortest request interval because the load is distributed to the LTE path.

E. EVALUATION OF BANDWIDTH UTILIZATION

Fig. 24 presents the bandwidth utilization according to each scheme in all three Scenarios. The bandwidth utilization levels are calculated using (24).

\[
\text{Utilization} = \frac{\sum R}{N \times R_{\text{max}}} \quad (24)
\]

In (24), \( N \) denotes the number of downloaded segments; \( R \) denotes the bitrate of the requested video quality, and \( R_{\text{max}} \) denotes the bitrate of the maximum video quality. If \( \text{Utilization} \) is close to 1, the bandwidth is fully utilized. Other schemes, except for CTRA and the proposed scheme, operate in a steady state. These schemes underestimate the aggregated bandwidth due to the use of a partial segment-based scheme. However, DAHS shows higher utilization than other partial-segment-based schemes because DAHS fully utilizes the bandwidth of mmWave while discarding the low-quality segments of LTE. Comparatively, both CTRA and the proposed scheme reduce the frequency of the OFF period using a block-based request policy with a long request interval. As a result, these methods utilize more bandwidth than the others. The proposed scheme achieves the highest bandwidth utilization by improving the responsiveness and distribution of data through offloading.

V. CONCLUSION

In conventional multipath-based HTTP adaptive streaming systems, the drawback of the segment request policy is that bandwidth utilization is degraded by the frequency of OFF periods. This problem leads to inaccurate bandwidth estimation and degrades the quality of experience (QoE). These issues are exacerbated by increased network asymmetry in the 5G new radio (NR) environment. In this paper, we proposed a multipath-based transmission scheme for efficient bandwidth utilization of HTTP adaptive streaming in a 5G NR environment. The proposed scheme consists of a collective segment request policy,
aggregated bandwidth estimation, segment scheduling to prevent the out-of-order problem, a rate adaptation algorithm to improve both bandwidth utilization and QoE, and an offloading controller to prevent buffer underflow in non-LOS 5G NR environments. The simulation results confirm that the proposed scheme accurately estimates aggregate throughput and improves the QoE. Furthermore, we demonstrated that this approach efficiently uses the network bandwidth. In future work, we will focus on evaluating the performance in real network environments by implementing the proposed scheme in a real video player.

REFERENCES

[1] M. Zhang, M. Polese, M. Mezzavilla, J. Zhu, S. Rangan, S. Panwar, and M. Zorzi, “Will TCP work in mmWave 5G cellular networks?” IEEE Commun. Mag., vol. 57, no. 1, pp. 65–71, Jan. 2019.

[2] M. Seufert, S. Egger, M. Slanina, T. Zinner, T. Höffeld, and P. Tran-Gia, “A survey on quality of experience of HTTP adaptive streaming.” IEEE Commun. Surveys Tuts., vol. 17, no. 1, pp. 469–492, 1st Quart., 2015.

[3] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli, “A control-theoretic approach for dynamic adaptive video streaming over HTTP.” in Proc. ACM Conf. Special Interest Group Data Commun. SIGCOMM, 2015, pp. 325–338.

[4] L. DeCicco, S. Mascolo, and V. Palmisano, “Feedback control for adaptive live video streaming,” in Proc. 2nd Annu. ACM Conf. Multimedia Syst., Feb. 2011, pp. 145–156.

[5] R. Patel and D. Deb, “Nonlinear adaptive control of microbial fuel cell with two species in a single chamber,” J. Power Sources, vol. 434, pp. 1–7, Sep. 2019.

[6] Microsoft, Smooth Streaming. [Online]. Available: https://www.microsoft.com/silverlight/smoothstreaming

[7] Apple, HTTP Live Streaming. [Online]. Available: https://developer.apple.com/streaming

[8] Adobe, HTTP Dynamic Streaming. [Online]. Available: https://www.adobe.com/products/hds-dynamic-streaming.html

[9] T. Stockhammer, “Dynamic adaptive streaming over HTTP :- Standards and design principles,” in Proc. 2nd Annu. ACM Conf. Multimedia Syst., Feb. 2011, pp. 133–144.

[10] T.-Y. Huang, N. Handigol, B. Heller, N. McKeown, and R. Johari, “Confused, timid, and unstable: Picking a video streaming rate is hard,” in Proc. ACM Conf. Internet Meas. Conf. IMC, 2012, pp. 225–238.

[11] R. K. P. Mok, W. Li, and R. K. C. Chang, “Rate: Initial video bitrate selection system for HTTP streaming,” IEEE J. Sel. Areas Commun., vol. 34, no. 6, pp. 1914–1928, Jun. 2016.

[12] J. Wu, C. Yuen, B. Cheng, M. Wang, and J. Chen, “Streaming high-quality mobile video with multipath TCP in heterogeneous wireless networks,” IEEE Trans. Mobile Comput., vol. 15, no. 9, pp. 2345–2361, Sep. 2016.

[13] S. Gouache, G. Bichot, A. Bsilà, and C. Howson, “Distributed & adaptive HTTP streaming,” in Proc. IEEE Int. Conf. Multimedia Expo, Barcelona, Spain, Jul. 2011, pp. 1–6.

[14] Y.-C. Chen, D. Towsley, and R. Khalili, “MPlayer: Multi-source and multi-path video streaming,” IEEE J. Sel. Areas Commun., vol. 34, no. 8, pp. 2198–2206, Aug. 2016.

[15] K. Evensen, D. Kaspar, C. Griwodz, P. Halvorsen, A. Hansen, and P. Engelstad, “Improving the performance of quality-adaptive video streaming over multiple heterogeneous access networks,” in Proc. 2nd Annu. ACM Conf. Multimedia Syst., Feb. 2011, pp. 57–68.

[16] B. Han, F. Qian, L. Ji, and V. Gopalakrishnan, “MP-DASH: Adaptive video streaming over preference-aware multipath,” in Proc. 12th Int. Conf. Emerg. Netw. Exp. Technol., Dec. 2016, pp. 129–143.

[17] Y. Go, O. C. Kwon, and H. Song, “An energy-efficient HTTP adaptive video streaming with networking cost constraint over heterogeneous wireless networks,” IEEE Trans. Multimedia, vol. 17, no. 9, pp. 1646–1657, Sep. 2015.

[18] C. Zhou, C.-W. Lin, X. Zhang, and Z. Guo, “A control-theoretic approach to rate adaption for DASH over multiple content distribution servers,” IEEE Trans. Circuits Syst. Video Technol., vol. 24, no. 4, pp. 681–694, Apr. 2014.

[19] H. Kim and K. Chung, “Segment scheduling scheme for efficient bandwidth utilization of HTTP adaptive streaming in multipath environments,” IEEE Access, vol. 7, pp. 36910–36920, 2019.

[20] J. Kua, G. Armitage, and P. Branch, “A survey of rate adaptation techniques for dynamic adaptive streaming over HTTP,” IEEE Commun. Surveys Tuts., vol. 19, no. 3, pp. 1842–1866, 3rd Quart., 2017.

[21] M. Polese, R. Jana, and M. Zorzi, “TCP and MP-TCP in 5G mmWave networks,” IEEE Internet Comput., vol. 21, no. 5, pp. 12–19, Sep. 2017.

[22] J. Park and K. Chung, “Layer-assisted video quality adaptation for improving QoE in wireless networks,” IEEE Access, vol. 8, pp. 77518–77527, 2020.

[23] X. Zhu, Z. Li, R. Pan, J. Gahm, and H. Hu, “Fixing multi-client oscillations in HTTP-based adaptive streaming: A control theoretic approach,” in Proc. IEEE 15th Int. Workshop Multimedia Signal Process. (MMSP), Sep. 2013, pp. 230–235.

[24] The Network Simulator NS-3. [Online]. Available: https://www.nsnam.org

HEEKWANG KIM received the B.S. and Ph.D. degrees from the Electronics and Communications Engineering Department, Kwangwoon University, Seoul, South Korea, in 2015 and 2020, respectively. He is currently an Associate Researcher with the Telecommunications Technology Association (TTA). His research interests include QoS, QoE support, multimedia systems, and streaming protocols.

KWANGSUE CHUNG (Senior Member, IEEE) received the B.S. degree from the Electrical Engineering Department, Hanyang University, Seoul, South Korea, the M.S. degree from the Electrical Engineering Department, Korea Advanced Institute of Science and Technology (KAIST), Seoul, and the Ph.D. degree from the Electrical Engineering Department, University of Florida, Gainesville, FL, USA. He spent ten years as a Research Staff with the Electronics and Telecommunications Research Institute (ETRI). From 1991 to 1992, he was also an Adjunct Professor with KAIST. From 2003 to 2004, he was a Visiting Scholar with the University of California at Irvine, Irvine, CA, USA. He joined Kwangwoon University, in 1993. His research interests include communication protocols and networks, QoS mechanism, and video streaming.