

적응형 세그먼트 길이 기반의 DASH 볼류메트릭 영상 스트리밍

유 상 우*, 홍 원 기°

Adaptive Segment Duration-based DASH Volumetric Video Streaming

Sangwoo Ryu*, James Won-Ki Hong°

요 약

볼류메트릭 영상은 몰입감 있는 경험을 제공하지만, 성공적인 스트리밍을 위해서는 높은 수준의 사용자 경험 (QoE)을 보장하고 다양한 네트워크 환경과 서비스 및 사용자 시나리오에 유연하게 대응해야 한다. 특히 볼류메트릭 영상 서비스는 수동적 시청과 사용자 상호작용 단계가 공존하는 경우가 많아, 재생 안정성과 저지연 반응성 사이의 균형이 요구된다. 2D 영상과 다른 특성 때문에 기존의 DASH 방식으로는 해당 가변성을 효율적으로 처리하는 데 한계가 있다. 본 연구에서는 볼류메트릭 영상을 위한 DASH 시스템 내에서 세그먼트 길이를 동적으로 조절하는 적응형 전략을 제안한다. 이를 위해 다양한 세그먼트 길이를 지원하도록 확장된 MPD 구조를 도입하고, 실시간으로 세그먼트 길이를 조절하며 버퍼를 관리하여 반응성과 안정성 간의 균형을 유지하는 세그먼트 적응 엔진을 제시한다. 구현된 시스템을 다양한 시나리오에서 평가해 시스템의 가변적인 환경에 대한 적응성을 분석한다.

Key Words : Volumetric Video Streaming, Adaptive Bitrate Streaming, MPEG-DASH

ABSTRACT

While volumetric video provides an immersive experience, successful volumetric video streaming requires high quality of experience (QoE) and robust adaptation to diverse network environments and user scenarios. Moreover, volumetric video services often include both passive viewing and user interaction phases, demanding a balance between playback stability and low-latency responsiveness. Due to the differences from 2D video, existing Dynamic Adaptive Streaming over HTTP (DASH) method often struggles to address this variability efficiently. In this paper, we propose an adaptive strategy that dynamically adjusts the segment duration within the DASH system for volumetric video. To this end, we first introduce an extended Media Presentation Description (MPD) structure to support multiple segment durations. We then present a segment adaptation engine that adjusts the duration of the segment in real time, manages buffers, and pursues a balance between responsiveness and stability. We implemented our system and conducted extensive evaluations across diverse scenarios. The results illustrate that our system effectively adapts to varying conditions while balancing stability and responsiveness.

* 이 논문은 2026년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원-대학ICT연구센터(ITRC)의 지원(IITP-2026-RS-2024-0-0437866, 50%)과 정보통신기획평가원의 지원을 받아 수행된 연구임 (RS-2024-00392332, 6G 네트워크 통합 지능평면 기술 개발, 50%)

• First Author : Pohang University of Science and Technology (POSTECH), Graduate School of Artificial Intelligence, rswoo@postech.ac.kr

° Corresponding Author : POSTECH, Department of Computer Science and Engineering, jwkhong@postech.ac.kr

논문번호 : KNOM2026-01-003, Received July 22, 2026; Revised August 10, 2026; Accepted August 18, 2026

I. 서 론

확장현실(eXtended Reality)은 차세대 실감형 미디어 서비스로, 메타버스 및 네트워크 기술의 발전과 함께 성장하고 있다. 특히 볼류메트릭 영상(Volumetric Video)은 현실 세계의 물체와 장면을 3D로 재구성해 디지털 공간으로 가져와 사용자에게 몰입감 있고 상호작용이 가능한 경험을 제공하는 확장현실의 핵심 기술이다. 이를 토대로 볼류메트릭 영상 서비스는 더욱 다양하고 복잡한 응용 시나리오로 확장되고 있다.

볼류메트릭 영상 서비스의 요구사항은 시나리오에 따라 크게 달라진다. 시청 중심의 수동적 시나리오인 재생 안정성이 중요하지만, 완전한 상호작용이 필요한 경우에는 사용자의 지속적인 움직임에 대한 반응성이 우선시된다. 두 방식이 혼재된 하이브리드 서비스도 존재하는데, 온라인 교육(강연자의 설명/3D 모델 상호작용), 스포츠 하이라이트(수동적 재생/자유 시점 탐색), 전자상거래(제품 시연/제품 탐색)가 대표적이다. 이러한 시나리오에서는 사용자의 현재 상태에 따라 시스템의 우선순위를 유연하게 조정하여 요구사항을 충족해야 한다.

일반적으로 비디오 스트리밍은 가변적인 네트워크 환경에서도 안정적인 콘텐츠 전송을 위해 MPEG-DASH [1]와 같은 적응형 비트레이트(ABR) 기술을 활용하며, 볼류메트릭 영상 스트리밍에서도 적응형 재생을 제공하기 위해 사용된다. 하지만 기존의 ABR 방식은 대부분 세그먼트 길이를 고정하기 때문에 하이브리드 서비스에서 상충하는 문제에 직면하게 된다. 예를 들어, 10초와 같은 긴 세그먼트는 HTTP 요청 오버헤드를 줄이고 영상 품질을 지속적으로 유지하여 안정성을 높이지만, 사용자 상호작용을 반영하기 위해 다음 세그먼트까지 기다려야 하므로 높은 지연 시간이 발생한다. 반면 1초 내외의 짧은 세그먼트는 즉각적인 상호작용이 가능하지만, 빈번한 HTTP 요청으로 인한 오버헤드가 발생하며 네트워크 환경이 불안정할 경우 버퍼링에 취약하다는 단점이 있다.

이러한 한계는 고정된 세그먼트 방식 대신 동적인 세그먼트 접근 방식의 필요성을 제시한다. 기존의 2D 영상용 동적 세그먼트 기법들 역시 이를 탐구한 바 있으나, 수동적인 시청 경험을 전제로 하여 안정적인 재생에만 최적화되어 있다. 따라서 상호작용이 핵심 요소인 볼류메트릭 영상 스트리밍에서는

재생 안정성과 상호작용 반응성 사이의 상충 관계를 효율적으로 관리하는 기술이 필요하다.

따라서 본 연구는 시청 중심의 수동적 환경과 사용자 상호작용이 공존하는 하이브리드 시나리오에서 안정성과 반응성을 효과적으로 확보하기 위해, 적응형 세그먼트 길이를 활용한 DASH 기반 볼류메트릭 영상 시스템을 제안한다. 이를 위해 세그먼트 길이를 결정하고 버퍼를 관리하는 세그먼트 적응 엔진을 설계한다. 해당 엔진은 네트워크 효율과 재생 안정성이 필요한 경우 긴 세그먼트를 우선 선택하며, 사용자 상호작용이 감지되면 짧은 세그먼트로 전환해 적절한 버퍼 관리를 수행하여 상호작용 지연 시간을 최소화한다. 다양한 네트워크 환경과 상호작용 시나리오를 바탕으로 제안하는 시스템의 성능을 평가하였다. 본 연구의 기여는 다음과 같다.

- 1) 상호작용 및 버퍼 상태에 따라 세그먼트 길이를 결정하는 적응형 엔진을 포함한 DASH 기반 볼류메트릭 영상 시스템을 제안
- 2) 다양한 네트워크 환경과 상호작용 시나리오를 바탕으로 시스템을 평가하여 제안 기법의 적응성을 분석

II. 관련 연구

본 장에서는 볼류메트릭 영상 스트리밍 연구 현황과 적응형 비트레이트 스트리밍을 설명한다.

1) 볼류메트릭 영상 스트리밍

볼류메트릭 영상 스트리밍은 지난 몇 년 간 꾸준히 연구되어 왔으며, 다양한 네트워크 및 서비스 요구사항을 만족시키기 위해 압축 코덱, 뷰포트 적응, 전송 최적화 등 다양한 기술들이 적용되었다. PCL [2], Draco [3], V-PCC [4] 등의 압축 코덱은 전송 효율성을 높이는 핵심 요소이며, 이를 바탕으로 전체적인 스트리밍 시스템을 발전시키기 위한 다양한 연구들이 수행되었다. ViVo [5] 및 GROOT [6]는 효율적인 사용자 시점 처리 방식을 사용했으며, YuZu [7]와 VoLUT [8]는 3D 초해상화 기반 대역폭 절감 방식을 제안하였다. MuV2 [9]는 다중 사용자 시나리오에서 엣지 기반의 콘텐츠 결합 및 공유 기술을 제안하였다. 이처럼 많은 연구들이 대역폭 절감을 위한 기술에 집중하고 있다. 본 연구는 대역폭 절감보다 스트리밍 시스템의 적응성 측면에 중점을 둔다.

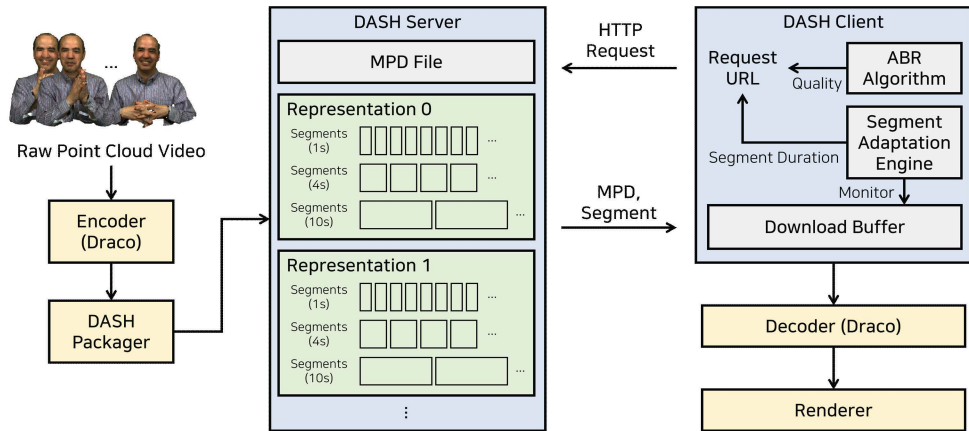


그림 1. 적응형 세그먼트 길이를 활용한 DASH 기반 볼류메트릭 영상 스트리밍 시스템 구조
Fig. 1. System architecture for DASH-based volumetric video streaming with adaptive segment duration

2) 적응형 비트레이트(ABR) 스트리밍

2D 영상 전송을 위한 적응형 비트레이트(ABR) 스트리밍 기술은 네트워크 환경 변화에 대응하기 위해 널리 연구되어 왔으며, 대부분 HLS [10]나 MPEG-DASH [1]와 같이 영상을 작은 단위로 분할해 적절한 비트레이트를 선택하는 방식을 따른다. LL-DASH [11]나 WebRTC [12]와 같이 저지연 요구사항을 만족하기 위한 방식들도 제안되었다. 볼류메트릭 영상 스트리밍 분야에서도 관련 연구들이 진행되었는데, 주로 PCC-DASH [13], DASH-PC [14], VV-DASH [15]와 같이 기존 DASH를 확장하여 포인트 클라우드 전송을 지원하거나, 다중 설명(multiple description) 기법과 WebRTC 혼잡 제어를 결합한 방식도 제안되었다 [16, 17].

2D 영상 스트리밍에서 세그먼트 길이를 가변적으로 조절하는 연구들이 존재하지만, 접근 방식과 최적화 목표는 볼류메트릭 영상의 요구사항과 차이가 있다. Schwarzmann 등 [18]은 장면 전환과 같은 콘텐츠 특성을 기반으로 세그먼트 길이를 조절하여 인코딩 효율을 개선했으나, 클라이언트의 실시간 네트워크나 버퍼 상태보다는 콘텐츠 자체에 집중한 방식이다. 또한, 다른 연구들 [19, 20, 21]은 클라이언트 상태에 따라 세그먼트 길이를 동적으로 변화시키는 방안을 제안하였으나, 최적화 목표는 재생 안정성 확보에 초점을 두고 있다. 반면, 볼류메트릭 영상은 사용자의 상호작용을 지원해야 하므로 안정성과 반응성 사이의 균형 및 버퍼 관리가 필수적이다. 본 논문에서는 수동적 시청과 능동적 상호작용 단계가 공존하는 환경에서, DASH의 세그먼트 기반

적응 방식을 확장하여 재생 안정성과 상호작용 반응성을 동시에 고려한 스트리밍 기법을 제안한다.

III. 시스템 설계

이 장에서는 제안하는 DASH 기반 볼류메트릭 영상 스트리밍 시스템 핵심 설계들을 서술한다.

1) 볼류메트릭 영상 스트리밍 시스템

그림 1은 제안하는 시스템의 전체 구조를 나타낸다. 서버는 온디맨드(On-Demand) 형식으로 콘텐츠를 전송하도록 설계되었다. 원본 포인트 클라우드 영상은 Draco [3] 인코더로 압축된 후 다양한 길이의 세그먼트로 패키징된다. 이때 MPD(Media Presentation Description) 형식의 메타데이터 파일이 생성되는데, 본 연구에서는 다중 세그먼트 길이를 지원하도록 MPD 구조를 확장하였다. 이 세그먼트와 메타데이터 파일은 HTTP 서버를 통해 제공된다. DASH 클라이언트는 세그먼트 적응 엔진과 ABR 알고리즘을 사용하여 원하는 길이와 품질의 세그먼트를 다운로드하고, 이를 프레임 단위로 추출한 뒤 Draco 디코더와 렌더러로 전달하여 재생한다. 결과적으로 본 시스템은 온디맨드 시나리오에서 볼류메트릭 영상을 스트리밍하기 위한 구조로 설계되었다.

2) MPD 구조 및 세그먼트

DASH 기반 영상 스트리밍에서 Media Presentation Description(MPD)은 영상 메타데이터

를 XML 형식으로 저장하는 표준적인 방식이다. DASH 클라이언트는 MPD 파일에 포함된 메타데이터를 활용하여 원하는 품질의 세그먼트를 다운로드한다. 일반적으로 MPD 파일은 AdaptationSet 요소 내부에 품질을 나타내는 여러 Representation 요소를 가지며, 각 Representation 요소는 세그먼트 정보를 담고 있는 SegmentList 요소를 포함한다. 포인트 클라우드용 DASH 스트리밍 시스템을 제안한 기존 연구들도 이러한 기본 구조를 따른다.

본 연구에서는 서로 다른 길이의 세그먼트를 지원하기 위해 하나의 Representation 요소가 다중 SegmentList를 가질 수 있도록 설계하였다. 이에 따라 각 SegmentList는 segmentSize 속성을 가지며, SegmentUrl은 세그먼트 길이에 따라 패키징된 파일의 URL과 메타데이터를 속성으로 포함한다. 그림 2는 이러한 설계를 반영한 MPD 구조의 예시이다.

```
<?xml version='1.0' encoding='UTF-8'?>
<MPD xmlns="urn:mpeg:dash:schema:mpd:2011" type="static"
  profiles="urn:mpeg:dash:profile:isoff-live:2011"
  mediaDuration="900" mediaFPS="30">
  <BaseURL>https://www.example.com/</BaseURL>
  <Period>
    <AdaptationSet id="0" codecs="draco" mimeType="3d/ptcl">
      <Representation id="0" bandwidth="50569456"
        availableDurations="30, 120, 300">
        <FrameTemplate
          media="/ph19/frames/compressed_ext/0/frame%04d.drc"
          startNumber="0" />
        <SegmentList startNumber="0" segmentNum="30" segmentSize="30">
          <SegmentUrl
            media="/ph19/segments/segments-30f/0/segment_00000.bin"
            startNumber="0" duration="30" />
          <SegmentUrl
            media="/ph19/segments/segments-30f/0/segment_00001.bin"
            startNumber="30" duration="30" />
          ...
        </SegmentList>
        <SegmentList startNumber="0" segmentNum="8" segmentSize="120">
          ...
        </SegmentList>
      </Representation>
      <Representation id="1" bandwidth="180941216"
        availableDurations="30, 120, 300">
        ...
      </Representation>
    </AdaptationSet>
  </Period>
</MPD>
```

그림 2. DASH MPD 구조 예시
Fig. 2. An example DASH MPD Structure

세그먼트 길이에 따른 특성(III-3장에서 상세 설명)을 고려하여 1초, 4초, 10초 세 가지 길이로 세그먼트를 구성하였다. VV-DASH [15]에서 제안하는 DVV 세그먼트를 참조하여, 각 세그먼트는 클라이언트에서 추출 및 디코딩이 가능하도록 설계하였다. 세그먼트 헤더에는 시작 프레임 인덱스, 포함된 프레임 수, 전체 데이터 길이, 그리고 프레임별 오프셋 정보가 포함된다. 페이로드에는 압축 효율과 디코딩 성능 간의 균형이 뛰어나 연구에서 일반적

으로 사용되는 Google Draco로 압축한 프레임별 데이터가 담겨 있다.

3) 세그먼트 적응 엔진

세그먼트 적응 엔진은 상태 머신(state machine) 방식으로 작동하며, 현재 상태에 기반하여 최적의 결정을 내리기 위해 STARTUP, VOD, INTERACTION의 세 가지 모드 중 하나를 선택한다. 세그먼트 길이에 따른 특성을 고려하여 1초(30 프레임), 4초(120프레임), 10초(300프레임) 단위의 세그먼트를 구성하였다. 1초 세그먼트는 사용자의 상호작용이 빠르게 반영되어 반응성 측면에서 유리하지만, 잦은 HTTP 요청과 세그먼트 추출 과정으로 인해 클라이언트 오버헤드가 발생할 수 있다. 반면 10초 세그먼트는 HTTP 요청 횟수가 적어 오버헤드가 낮고 품질이 안정적이나, 상호작용에 대한 반응이 느리고 큰 세그먼트 크기로 인해 추출 과정에서 오버헤드가 발생할 수 있다. 4초 세그먼트는 1초와 10초 세그먼트의 절충점 역할을 한다.

```
1 Input: is_interacting, buffer_level ∈ [0, 1]
2 Output: segment_duration, need_flush

3 now ← time()
4 need_flush ← False
5 if is_interacting then
6   if current_state ≠ 'INTERACTION' then
7     current_state = 'INTERACTION'
8     max_buf_size ← 2 seconds
9     need_flush ← True
10  if current_state = 'INTERACTION' and
    now - first_interaction ≥ cooldown then
11    current_state = 'VOD'
12    max_buf_size ← original_max_buf_size
13  if current_state = 'STARTUP' then
14    return (1s, False)
15  if current_state = 'INTERACTION' then
16    return (1s, need_flush)
17  if buffer_level < buffer_lower_threshold then
18    return (1s, False)
19  if buffer_level > buffer_upper_threshold then
20    return (10s, False)
21  return (4s, False)
```

그림 3. 세그먼트 길이 적응 알고리즘
Fig. 3. Segment Adaptation Algorithm

시스템이 시작되면 STARTUP 모드를 유지하며 짧은 1초 세그먼트를 이용해 빠르게 버퍼를 채우고, 첫 프레임이 렌더링된 직후 VOD 모드로 전환된다.

이러한 초기 단계 이후, 세그먼트 적응 엔진은 상호 작용성과 다운로드 버퍼 수준이라는 두 가지 주요 신호를 기준으로 세그먼트 길이를 전환한다. 먼저 상호작용성의 경우, 카메라 움직임이 임계값을 초과하면 엔진은 1초 세그먼트를 사용하는 INTERACTION 모드로 진입한다. 이 모드에서는 최신 정보를 즉각 반영하기 위해 최소한의 안전 프레임(need_flush)을 제외한 나머지 다운로드 버퍼를 비우며, 반응성과 안정성 간의 균형을 맞추기 위해 최대 버퍼 크기를 2초로 제한하여 지연 시간을 최소화한다. 일정 시간 동안 상호작용이 감지되지 않으면 VOD 모드로 복귀하고 최대 버퍼 크기도 재설정한다. 다음으로 다운로드 버퍼 수준의 경우, 버퍼가 낮으면 1초 세그먼트를 사용하여 네트워크 상태를 빠르게 갱신하고 버퍼를 채운다. 반면 버퍼 수준이 높으면 시스템이 안정적인 것으로 판단하여 10초 세그먼트를 사용하여 효율성을 높인다.

세그먼트 선택 외에도 엔진은 버퍼 수준을 지속적으로 모니터링하며 위험 상황을 감지한다. 고품질의 큰 세그먼트를 다운로드하는 도중 버퍼 수준이 최대 버퍼 크기 대비 특정 임계값(20%) 밑으로 떨어지면, 진행 중인 다운로드를 취소하고 더 짧은 세그먼트로 강제 전환하여 네트워크 대역폭을 빠르게 다시 계산한다. 그림 3은 본 연구의 세그먼트 길이 적응 방식이다. 임계값은 실험을 통해 경험적으로 결정하였다. 실험에서는 버퍼 수준(buffer_level)을 최대 버퍼 크기를 기준으로 0~1로 정규화하고 버퍼 하한 임계값(buffer_lower_threshold)과 상한 임계값(buffer_upper_threshold)은 각각 0.3과 0.8로 설정하였으며, 쿨다운(cooldown)과 최대 버퍼 크기(max_buf_size)는 각각 5초와 15초로 설정하였다.

IV. 시스템 구현

이 장에서는 DASH 기반 볼류메트릭 영상 스트리밍 시스템의 서버, 클라이언트에 대한 구체적인 구현 방식을 설명한다.

1) 서버

볼류메트릭 영상의 적응형 스트리밍을 위한 DASH 서버는 Go 언어 [22]로 구현하였다. 볼류메트릭 영상으로는 MVUB(Microsoft Voxelized Upper Bodies) 데이터셋 [23]의 Phil-9 영상을 사용했으며, 프레임 당 평균 33.4만 개의 포인트로 구성

되어 있다 (평균 7.09 MB, 420 Mbps). VV-DASH [15]를 참고해 245 프레임 길이의 영상을 반복해 총 900 프레임 길이의 영상으로 수정하였다. Open3D[24]를 사용하여 포인트 수를 10%, 40%, 70%, 100%로 샘플링해 네 가지 품질의 데이터를 생성하였고, DracoPy [25]를 이용해 인코딩하였다.

2) 클라이언트

볼류메트릭 영상의 적응형 스트리밍을 위한 DASH 클라이언트는 Python으로 구현하였다. 세그먼트 다운로드, 디코딩, 렌더링으로 구성된 클라이언트 파이프라인을 효율적으로 실행하기 위해 asyncio(Python 내장)와 aiohttp [26]를 활용해 비동기 방식으로 동작하도록 설계하였다. 실험에는 대역폭 기반의 ABR 알고리즘을 적용하여 세그먼트 길이 변화와 독립적으로 비트레이트를 제어하도록 구성하였으며, 프레임 디코딩에는 DracoPy를 사용하였다. 또한, 연산 집약적인 작업을 효율적으로 처리하기 위해 최대 6개의 프로세스를 사용하는 멀티프로세싱을 사용하였다. 평가를 위해 디코딩된 포인트 클라우드는 Open3D의 헤드리스(headless) 모드로 렌더링하였으며, 카메라 움직임을 통해 상호작용을 구현하였다.

V. 실험 환경

이 장에서는 제안하는 적응형 세그먼트 기반의 볼류메트릭 영상 스트리밍 시스템을 분석하기 위한 실험 환경을 설명한다.

1) 테스트베드 구성

서버, 클라이언트, 그리고 네트워크 에뮬레이터는 단일 워크스테이션에서 Docker [27] 컨테이너를 이용하여 구동하였다. 워크스테이션은 Intel Xeon(R) Silver 4215R CPU, Quadro RTX 5000 GPU, 64GB RAM으로 구성되어 있다. 네트워크 환경 제어를 위해 Docker 네트워크 내부에서 Linux tc 명령어를 사용하였으며, 콘텐츠 품질과 시나리오를 바탕으로 설정된 대역폭 트래이스를 활용하여 클라이언트의 트래픽을 제어하였다.

2) 네트워크 시나리오

제안하는 시스템을 평가하기 위해 네 가지 시나리오를 정의하였다. 각 시나리오를 수행하는 동안

표 1. 시나리오 1(정적 네트워크 환경)에서의 측정 결과
Table 1. Evaluation results of Scenario 1 (Stable Network)

Segment Duration	Bitrate (Mbps)	Startup Time (s)	Stall Time (s)	Num. of Requests
Fixed-1s	414.54	0.18	0.0	61
Fixed-4s	396.05	0.32	0.0	17
Fixed-10s	339.45	0.78	0.0	7
Adaptive (Proposed)	414.54	0.16	0.0	16

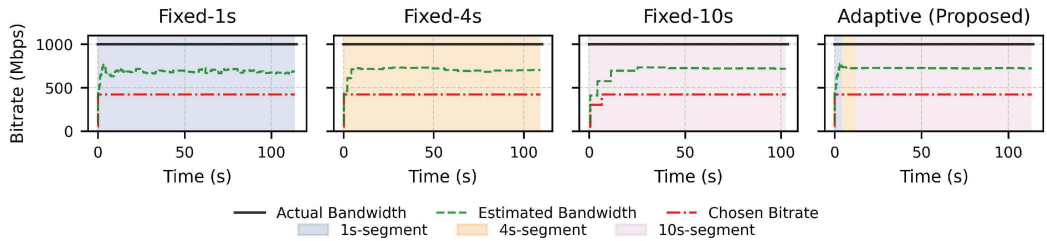


그림 4. 시나리오 1에서 시간의 흐름에 따른 비트레이트 및 세그먼트 길이 선택
Fig. 4. Bitrate and segment duration selection over time on Scenario 1

900 프레임의 포인트 클라우드 영상을 2회 반복 재생하여(30fps 기준 1분) 평가 지표를 측정하였다.

- 시나리오 1 (정적 네트워크): 대역폭을 1Gbps로 고정하여 모든 품질의 볼류메트릭 영상을 제공할 수 있게 하였다.
- 시나리오 2 (정적 네트워크): 시나리오 1의 네트워크 상황에서 정해진 시간대(그림 5의 빗금 영역)에 상호작용을 추가하였다.
- 시나리오 3 (동적 네트워크): 볼류메트릭 영상의 화질 별 대역폭을 고려해, (600Mbps, 400Mbps, 200Mbps, 100Mbps)의 대역폭을 (5초, 5초, 5초, 10초) 주기로 반복하는 시나리오를 구성하였다.
- 시나리오 4 (동적 네트워크): 시나리오 3의 네트워크 상황에서 정해진 시간대(그림 7의 빗금 영역)에 상호작용을 추가하였다.

3) 실험 과정 및 평가 지표

제안하는 세그먼트 적응 기법(Adaptive)을 세 가지 고정 길이 세그먼트 방식(Fixed-1s, Fixed-4s, Fixed-10s)과 비교하였다. Fixed-1s는 짧은 세그먼트나 청크(chunk)를 사용하는 LL-DASH [11]와 같은 저지연 프로토콜을 상정하였으며, Fixed-10s는 안정적인 VoD 스트리밍을 위한 기존 DASH 표준을, Fixed-4s는 중간 설정을 나타낸다. 2D 영상 스트리밍의 가변 세그먼트 방식[19, 20, 21]은 상호작용 상황이 존재하지 않아 반응성을 고려하지 않고, 대역폭 및 시스템 부담이 볼류메트릭 영상과 다른 특

성을 가지고 있어 포함하지 않았다.

평가 지표로는 평균 비트레이트(수신 품질), 시작 시간(첫 프레임 지연 시간), 스톨 시간(버퍼가 비어 있는 총 시간), 그리고 HTTP 요청 횟수(자원 효율성)를 측정하였다. 상호작용을 포함하는 시나리오 2과 4에 대해서는 상호작용이 감지된 시점부터 해당 프레임이 렌더링될 때까지 걸리는 시간인 상호작용 응답 지연 시간을 추가로 측정하였다. 프레임 속도(FPS)는 클라이언트 기기 성능의 영향을 크게 받는 연산 집약적인 디코딩 및 렌더링 이후에 측정되었으므로 평가 및 분석에는 활용하지 않았다 [28].

VI. 실험 결과 및 분석

1) 시나리오 1 실험 결과

표 1과 그림 4는 시나리오 1의 결과를 보여준다. 시나리오 1은 네트워크 상태가 안정적인 환경이므로, 대부분의 경우 높은 품질의 세그먼트가 선택되었다. Fixed-10s의 경우 세그먼트 길이가 길어 초기 선택된 저품질 세그먼트가 평균 미디어 비트레이트를 낮추는 결과를 보였다. 반면, 제안하는 적응형 방식(Adaptive)은 초기에 1초 세그먼트를 사용하다가 시스템이 안정화됨에 따라 순차적으로 4초 및 10초 세그먼트를 선택하였다. Fixed-1s와 적응형 방식은 초기 단계에서 짧은 세그먼트를 사용하므로 시작 시간이 짧게 측정되었다.

표 2. 시나리오 2(정적 네트워크 환경 및 상호작용)에서의 측정 결과
Table 2. Evaluation results of Scenario 2 (Stable Network + Interaction)

Segment Duration	Bitrate (Mbps)	Startup Time (s)	Stall Time (s)	Num. of Requests	Interaction Latency (s)
Fixed-1s	414.54	0.18	0.0	61	27.71
Fixed-4s	396.05	0.33	0.0	17	32.69
Fixed-10s	359.08	0.73	0.0	7	38.79
Adaptive (Proposed)	411.66	0.12	0.0	42	5.39

표 3. 시나리오 3(동적 네트워크 환경)에서의 측정 결과
Table 3. Evaluation results of Scenario 3 (Dynamic Network)

Segment Duration	Bitrate (Mbps)	Startup Time (s)	Stall Time (s)	Num. of Requests
Fixed-1s	197.01	0.20	0.0	61
Fixed-4s	277.27	0.44	0.0	17
Fixed-10s	199.94	1.16	1.30	7
Adaptive (Proposed)	225.00	0.16	0.0	31

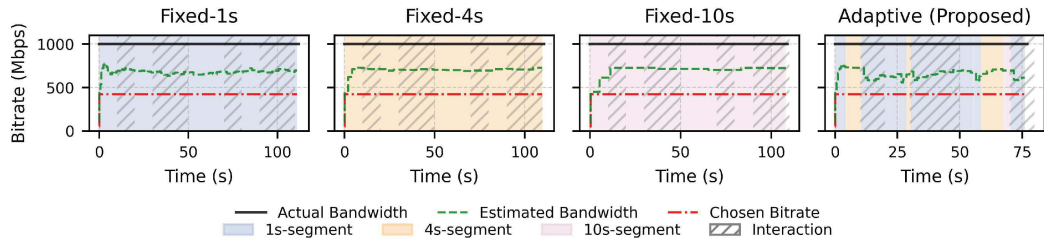


그림 5. 시나리오 2에서 시간의 흐름에 따른 비트레이트 및 세그먼트 길이 선택
Fig. 5. Bitrate and segment duration selection over time on Scenario 2

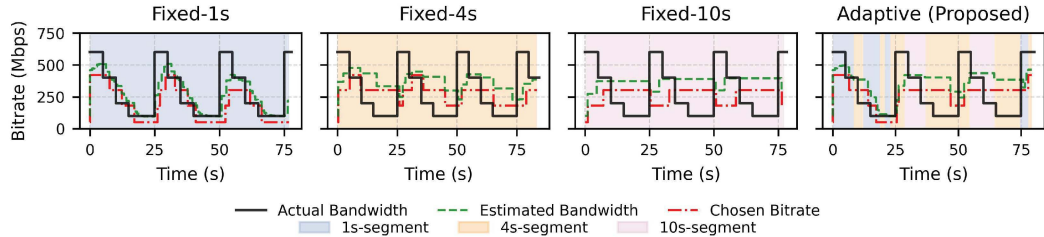


그림 6. 시나리오 3에서 시간의 흐름에 따른 비트레이트 및 세그먼트 길이 선택
Fig. 6. Bitrate and segment duration selection over time on Scenario 3

2) 시나리오 2 실험 결과

표 2과 그림 5는 시나리오 2의 결과를 보여준다. 시나리오 2는 시나리오 1과 유사하게 안정적인 네트워크 환경을 가정하므로 비트레이트 결과 역시 발생하는 구간에서 적응형 방식(Adaptive)이 1초 세그먼트를 선택한다는 점이다. 적응형 방식은 적절한 버퍼 관리를 통해 낮은 응답 지연 시간을 보인 반면, 고정 길이 방식들은 이미 버퍼에 쌓여 있는 프레임들로 인해 높은 응답 지연 시간이 측정되었다.

그림 8은 시나리오 2에서 적응형 방식의 시간에 따른 버퍼 수준 변화를 보여준다. 상호작용이 감지 되면(빛금 영역), 진행 중인 다운로드를 취소하고 1초 세그먼트를 선택하며 버퍼를 비운다(일부 안전 프레임은 유지). 또한, 응답 지연 시간을 낮게 유지하기 위해 최대 버퍼 크기를 기존 수준의 20%로 조정한다. 상호작용이 종료된 후에는 버퍼를 다시 안정적인 수준까지 채우는 과정을 거친다. 이처럼 초기 상호작용 지연시간은 버퍼에 남아있는 상호작

표 4. 시나리오 4(동적 네트워크 환경 및 상호작용)에서의 측정 결과
Table 4. Evaluation results of Scenario 4 (Dynamic Network + Interaction)

Segment Duration	Bitrate (Mbps)	Startup Time (s)	Stall Time (s)	Num. of Requests	Interaction Latency (s)
Fixed-1s	191.00	0.14	0.0	61	14.06
Fixed-4s	277.48	0.46	0.0	17	15.58
Fixed-10s	240.51	0.97	0.0	7	22.93
Adaptive (Proposed)	207.80	0.16	0.0	58	3.80

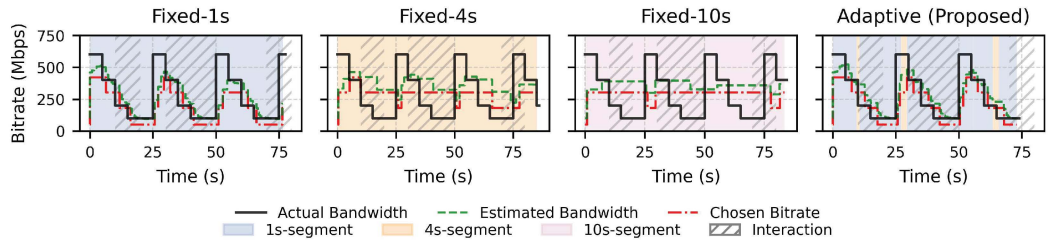


그림 7. 시나리오 4에서 시간의 흐름에 따른 비트레이트 및 세그먼트 길이 선택
Fig. 7. Bitrate and segment duration selection over time on Scenario 4

용 미반영 프레임의 양에 주로 영향을 받으며, 따라서 상호작용이 존재하는 경우 제안하는 시스템에서는 적응형 세그먼트 길이보다 버퍼 비우기 등의 버퍼 관리의 역할이 크게 작용하게 된다. 적응형 세그먼트 길이는 상호작용 구간 동안 지속되는 변화에 빠르게 반응하고, 상호작용 이후의 VOD 구간에서 다시 안정적인 재생 상태를 제공할 수 있도록 한다.

적응형 방식의 시작 시간은 Fixed-1s와 유사하게 측정되었으며, 상호작용으로 인해 이전 시나리오보다 HTTP 요청 횟수가 높게 나타났다.

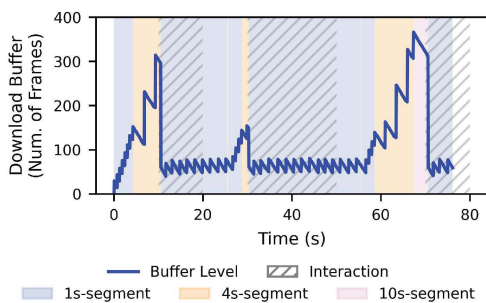


그림 8. 시나리오 2에서 시간의 흐름에 따른 버퍼 수준
Fig. 8. Buffer level over time on Scenario 2

3) 시나리오 3 실험 결과

표 3과 그림 6은 시나리오 3의 결과를 보여준다. 시나리오 3은 네트워크 상태가 변동적인 환경이므로, 네트워크에 적응하거나 미리 다운로드된 세그먼트를

를 활용하여 재생 안정성을 확보해야 한다. Fixed-1s는 세그먼트 길이가 짧아 대역폭 갱신이 빈번하게 일어나며 현재 네트워크 환경에 빠르게 적응한다. 반면, Fixed-4s와 Fixed-10s는 세그먼트 길이가 길어 대역폭 갱신이 즉각적이지 않으며, 이전에 다운로드된 세그먼트를 사용한다. 그림 6에서 Fixed-4s와 Fixed-10s는 최저 품질로 하향 조정되지 않고 영상을 재생하였다. 적응형 방식(Adaptive)은 초반(0~25초)에는 대역폭에 적응하고, 중반 이후(25초~)에는 10초 세그먼트를 사용하여 품질을 유지하는 두 가지 양상을 모두 보이며 초기 적응성과 재생 안정성을 보여주었다. 시작 시간은 시나리오 1과 유사한 결과를 보였으며, 대역폭 변동 구간에서 세그먼트를 활용함에 따라 요청 횟수는 증가하였다. Fixed-10s에서 스톨(stall)이 발생한 것과 달리, 적응형 방식은 10초 세그먼트를 사용하는 구간에서도 스톨 발생을 효과적으로 방지하였다.

4) 시나리오 4 실험 결과

표 4와 그림 7은 시나리오 4의 결과를 보여준다. 시나리오 4는 상호작용과 네트워크 상태를 모두 고려해야 하는 가장 복잡한 환경이다. 중반 이후 긴 세그먼트를 활용했던 시나리오 3과 달리, 시나리오 4의 적응형 방식(Adaptive)은 상호작용으로 인해 지속적으로 짧은 세그먼트를 선택하며 대역폭에 적응하였다. Fixed-4s 및 Fixed-10s은 낮은 대역폭과 상

호작용이 겹치는 구간에서 상호작용이 반영되지 않은 기존 프레임을 사용하여 품질을 유지한 것과 다르게, 적응형 방식(Adaptive)은 상호작용이 반영된 프레임들을 대역폭에 맞게 새로 다운받아 평균 비트레이트는 낮게 측정되었다. 시나리오 2와 유사하게 적응형 방식은 낮은 상호작용 응답 지연 시간을 보였다. 여기서 측정된 지연 시간의 절대값이 시나리오 2보다 낮은 이유는, 품질이 낮아 프레임 속도가 높게 유지됨에 따라 버퍼가 더 빠르게 소진되었기 때문이다. 적응형 방식의 시작 시간은 Fixed-Is와 유사하였으며, 시나리오 전반에 걸쳐 주로 1초 세그먼트를 사용함에 따라 HTTP 요청 횟수 또한 Fixed-Is와 비슷한 수준으로 나타났다.

5) 종합 결과

네 가지 시나리오에서의 평가를 통해, 제안하는 적응형 세그먼트 시스템이 재생 안정성과 상호작용 반응성 사이의 균형을 유지할 수 있음을 보여주었다. 상호작용이 없는 안정적(시나리오 1) 및 가변적(시나리오 3) 네트워크 환경에서, 시스템은 안정성과 효율성을 우선시한다. 평가 결과에 따르면, 시스템은 빠른 적응을 위해 짧은 세그먼트를 활용하는 동시에, 안정적인 재생과 요청 오버헤드 감소를 위해 긴 세그먼트를 적절히 혼용한다. 상호작용이 포함된 시나리오 2와 4에서는 짧은 세그먼트와 버퍼 관리를 통해 짧은 상호작용 응답 시간을 보였으며, 동시에 네트워크 상태에 대한 적절한 적응 성능을 보여주었다. 종합적으로, 본 시스템은 각 조건 및 시나리오에서 요구되는 안정성과 반응성 간의 목표 균형을 효과적으로 달성하며 적응성을 보여주었다.

VII. 논 의

본 연구는 적응형 세그먼트 시스템의 잠재력을 보여주었지만, 몇 가지 한계점이 존재한다. 우선, 본 연구에서는 제안 기법의 효과에 집중하기 위해 단순 대역폭 기반 ABR 알고리즘을 사용하였으나, 실제 환경에서 널리 쓰이는 버퍼 기반(BOLA [29])이나 하이브리드(MPC [30]), 혹은 덤핑 기반(Pensieve [31]) 알고리즘과의 상호작용 및 최적화가 추가로 검증되어야 한다. 또한 클라이언트 측의 디코딩 및 렌더링 부하가 전체 파이프라인에 미치는 영향을 최소화하기 위해 스레드 및 멀티프로세싱 등을 사용하였지만, 여전히 일부 처리에 따른 병목 현상과 지연 문제가 남아있다. 기기의 실제 성능

상태를 ABR과 세그먼트 선택 과정에 실시간으로 피드백하는 구조가 동반되어야 한다 [28]. 마지막으로, 성능 지표 중심의 평가를 넘어 실제 사용자 체감을 반영하는 QoE 평가가 수반되어야 한다.

향후 연구에서는 이러한 한계를 극복하기 위해 다양한 접근이 가능하다. 현재는 세그먼트 길이 제어 로직이 ABR 알고리즘과 분리되어 동작하지만, 다양한 ABR 알고리즘의 통합 및 분석을 통해 서로 간의 상호작용 및 영향을 정밀하게 분석하고, 이와 함께 프레임 속도(FPS) 및 디코딩 시간 등 기기 성능 상태까지 활용하는 통합 알고리즘으로 발전시킬 수 있다. 특히 현재의 규칙 기반(rule-based) 세그먼트 선택 방식을 개선하여, 보다 정교한 QoE 공식을 실시간으로 반영하는 최적화 모델이나 인공지능 기반의 지능형 적응 모델을 도입하는 것도 한 가지 연구 방향이다. 아울러 실제 네트워크 대역폭 트레이스 및 콘텐츠를 적용하거나 인코딩 과정을 포함한 라이브 스트리밍 시나리오로 범위를 확장함으로써, 더욱 실질적인 물입 경험을 제공하는 시스템으로 고도화할 수 있을 것이다.

VIII. 결 론

본 연구는 DASH 시스템에서 볼류메트릭 영상의 새로운 요구 사항을 해결하기 위해 세그먼트 길이를 적응적으로 조절하는 DASH 기반 볼류메트릭 영상 스트리밍 시스템을 제안하였다. 주로 2D 영상을 위해 설계된 기존의 고정 길이 세그먼트 DASH 시스템은 볼류메트릭 스트리밍에 필요한 안정성과 반응성 간의 균형을 효과적으로 유지하는 데 어려움이 있었다. 이러한 적응을 구현하기 위해, 우선 클라이언트 측 세그먼트 적응을 지원하는 MPD 구조를 정의하였다. 이어 제안하는 세그먼트 적응 엔진은 MPD 파일에 정의된 세그먼트 길이 옵션을 활용하여 사용자의 상호작용, 네트워크, 버퍼 상태를 기반으로 최적의 세그먼트를 선택하며, 이와 더불어 적절한 버퍼 관리를 수행한다. 서로 다른 네트워크 환경과 상호작용 조건을 갖춘 네 가지 시나리오를 통해 시스템을 평가하였으며, 각 시나리오에서 시스템이 상황에 맞게 반응성과 안정성을 효과적으로 조정하며 적응적으로 동작하는 것을 확인하였다. 향후 연구로는 고도화된 적응형 비트레이트 알고리즘, 기기 성능 피드백 및 주관적 품질 평가와 더불어 더욱 실용적인 시스템을 위한 중단 간 라이브 스트리밍 환경으로의 확장을 제시하였다.

References

- [1] T. Stockhammer, "Dynamic adaptive streaming over HTTP-- standards and design principles." Proceedings of the second annual ACM conference on Multimedia systems. 2011.
- [2] R. B. Rusu and S. Cousins, "3d is here: Point cloud library (pcl)." 2011 IEEE international conference on robotics and automation. IEEE, 2011.
- [3] Draco 3D Data Compression, Retrieved June., 25, 2026, <https://google.github.io/draco>.
- [4] MPEG V-PCC, Retrieved June., 25, 2026, <https://www.iso.org/standard/83535.html>.
- [5] B. Han et al., "ViVo: Visibility-aware mobile volumetric video streaming." In Proceedings of the 26th annual international conference on mobile computing and networking (pp. 1–13). 2020.
- [6] K. Lee et al., "GROOT: a real-time streaming system of high-fidelity volumetric videos." In Proceedings of the 26th Annual International Conference on Mobile Computing and Networking (pp. 1–14). 2020.
- [7] A. Zhang et al., "YuZu: Neural-Enhanced volumetric video streaming." 19th USENIX Symposium on Networked Systems Design and Implementation (NSDI 22). 2022.
- [8] C. Wang et al., "Volut: Efficient volumetric streaming enhanced by lut-based super-resolution." Proceedings of Machine Learning and Systems 7 (2025).
- [9] Y. Liu et al., "Muv2: Scaling up multi-user mobile volumetric video streaming via content hybridization and sharing." Proceedings of the 30th Annual International Conference on Mobile Computing and Networking. 2024.
- [10] R. Pantos and W. May, HTTP live streaming. No. rfc8216. 2017.
- [11] LL-DASH, Retrieved June., 25, 2026, <https://dashif.org/news/low-latency-dash>.
- [12] WebRTC, Retrieved June., 25, 2026, <https://webrtc.org>.
- [13] J. Van Der Hooft et al., "Towards 6dof http adaptive streaming through point cloud compression." Proceedings of the 27th ACM International Conference on Multimedia. 2019.
- [14] M. Hosseini and C. Timmerer, "Dynamic adaptive point cloud streaming." Proceedings of the 23rd Packet Video Workshop. 2018.
- [15] H. Heidarirad and M. Wang, "VV-DASH: A Framework for Volumetric Video DASH Streaming." Proceedings of the 16th ACM Multimedia Systems Conference. 2025.
- [16] M. De Fré et al., "Scalable mdc-based volumetric video delivery for real-time one-to-many webrtc conferencing." Proceedings of the 15th ACM multimedia systems conference. 2024.
- [17] M. De Fré et al., "Demonstrating adaptive many-to-many immersive teleconferencing for volumetric video." Proceedings of the 15th ACM multimedia systems conference. 2024.
- [18] S. Schwarzmann et al., "Comparing fixed and variable segment durations for adaptive video streaming: A holistic analysis." Proceedings of the 11th ACM Multimedia Systems Conference. 2020.
- [19] J. van der Hooft et al., "Low-latency delivery of news-based video content." Proceedings of the 9th ACM Multimedia Systems Conference. 2018.
- [20] I. Ironi et al., "Optimized adaptation algorithm for HEVC/H. 265 dynamic adaptive streaming over HTTP using variable segment duration." Real-Time Image and Video Processing 2016. Vol. 9897. SPIE, 2016.
- [21] B. J. Villa and P. E. Heegaard, "Group based traffic shaping for adaptive HTTP video streaming by segment duration control." 2013 IEEE 27th International Conference on Advanced Information Networking and Applications (AINA). IEEE, 2013.
- [22] The Go Programming Language, Retrieved June., 25, 2026, <https://go.dev/>
- [23] C. Loop et al., Microsoft Voxelized Upper Bodies - A Voxelized Point Cloud Dataset, Retrieved June., 25, 2026, <https://plenodb.jpeg.org/pc/microsoft>. 2016.
- [24] Q.-Y. Zhou et al., "Open3D: A modern library for 3D data processing." arXiv preprint arXiv:1801.09847 (2018).
- [25] DracoPy, Retrieved June., 25, 2026,

<https://github.com/seung-lab/DracoPy>.

- [26] Async http client/server framework, Retrieved June., 25, 2026, <https://github.com/aio-lib/aiohttp>.
- [27] D. Merkel, "Docker: lightweight linux containers for consistent development and deployment." Linux j 239.2 (2014): 2.
- [28] S. Ryu, and J.W.K. Hong, "Analysis of the Impact of Client Computing Capability on ABR Algorithm Behaviors in Volumetric Video Streaming", KNOM Review, Vol. 28, No. 2, Dec. 2025, pp.23-32.
- [29] K. Spiteri et al., "BOLA: Near-optimal bitrate adaptation for online videos." IEEE/ACM transactions on networking 28.4 (2020): 1698-1711.
- [30] X. Yin et al., "A control-theoretic approach for dynamic adaptive video streaming over HTTP." Proceedings of the 2015 ACM conference on special interest group on data communication. 2015.
- [31] H. Mao et al., "Neural adaptive video streaming with pensieve." Proceedings of the conference of the ACM special interest group on data communication. 2017.

유 상 우 (Sangwoo Ryu)



멀티미디어

2020 포항공과대학교, 컴퓨터 공학과 학사
 2020 ~ 포항공과대학교, 인공 지능대학원 석/박사 통합과정
 <관심분야> 2D/3D 비디오 스트리밍, 영상회의, 인공지능,

홍 원 기 (James Won-Ki Hong)



2020~현재 포스텍 크립토블록체인연구센터 공동센터장
 2019-2023 포스텍 교육혁신센터장
 2012~2014 KT 종합기술원 원장
 2008~2012 포스텍 정보전자융

합공학부장

2008~2010 포스텍 컴퓨터공학과 주임교수
 2007~2010 포스텍 정보통신연구소 연구소장
 2007~2011 포스텍 정보통신대학원장
 1995~현재 포스텍 컴퓨터공학과 교수
 1992~1995 Univ. of Western Ontario, 연구교수
 <관심분야> AI 기반 네트워크관리, SDN/NFV, 무크기반 교육, 블록체인 및 암호화폐, Vmeeting 화상회의시스템