Edge-Assisted 360° Videos Streaming to Head-Mounted Virtual Reality

利用邊緣運算最佳化360度全景影片之頭戴式虛擬實境串流

Wen-Chih Lo
Advisor: Cheng-Hsin Hsu

Networking and Multimedia Systems Lab,
CS Dept., National Tsing Hua University

Outline

- Introduction
- Challenges
- System Architecture
  - Tile Rewriting
  - Viewport Rendering
- Optimal Edge-Assisted Streaming to HMDs
- 360° Viewing Dataset
- Evaluations
- Conclusion

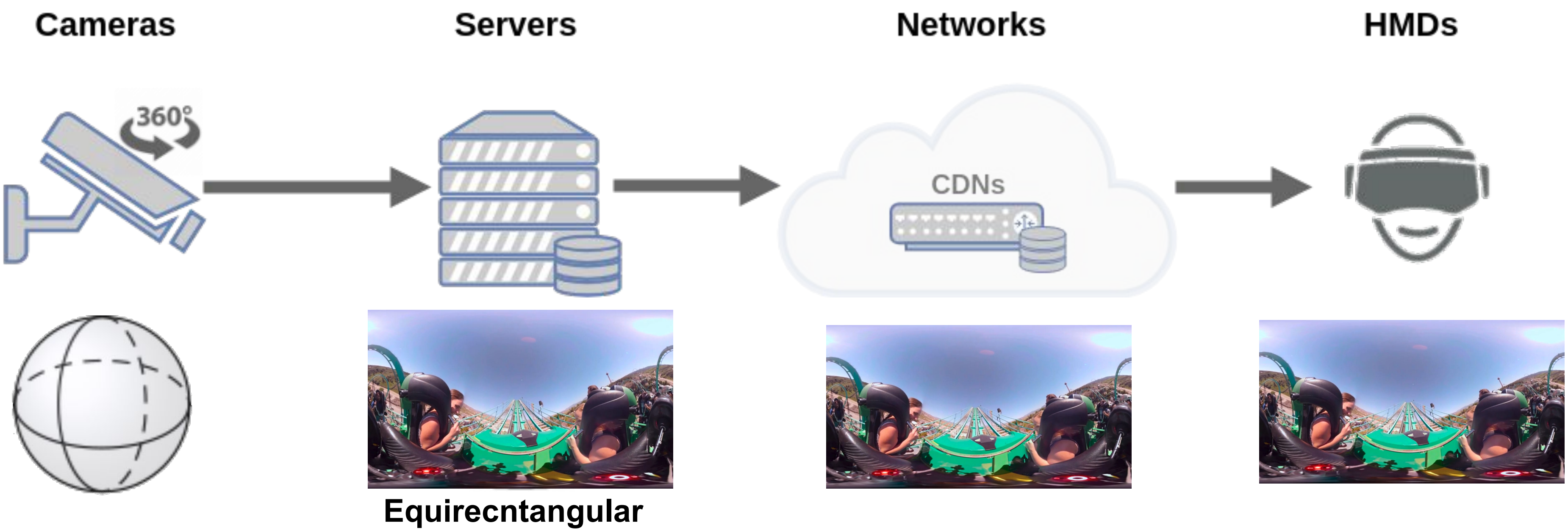
Introduction

Explore the World from Your Sofa

- Virtual Reality (VR) in business, education, film, media, entertainment, healthcare, and etc.
- **360° video** (aka spherical or omnidirectional video)
  - Every direction is recorded at the same time
  - Viewers dynamically change their viewports at playout time

⇨ **Offer better immersive experience**

Current Streaming Approach: How Does It Work?



Everything seems to be good

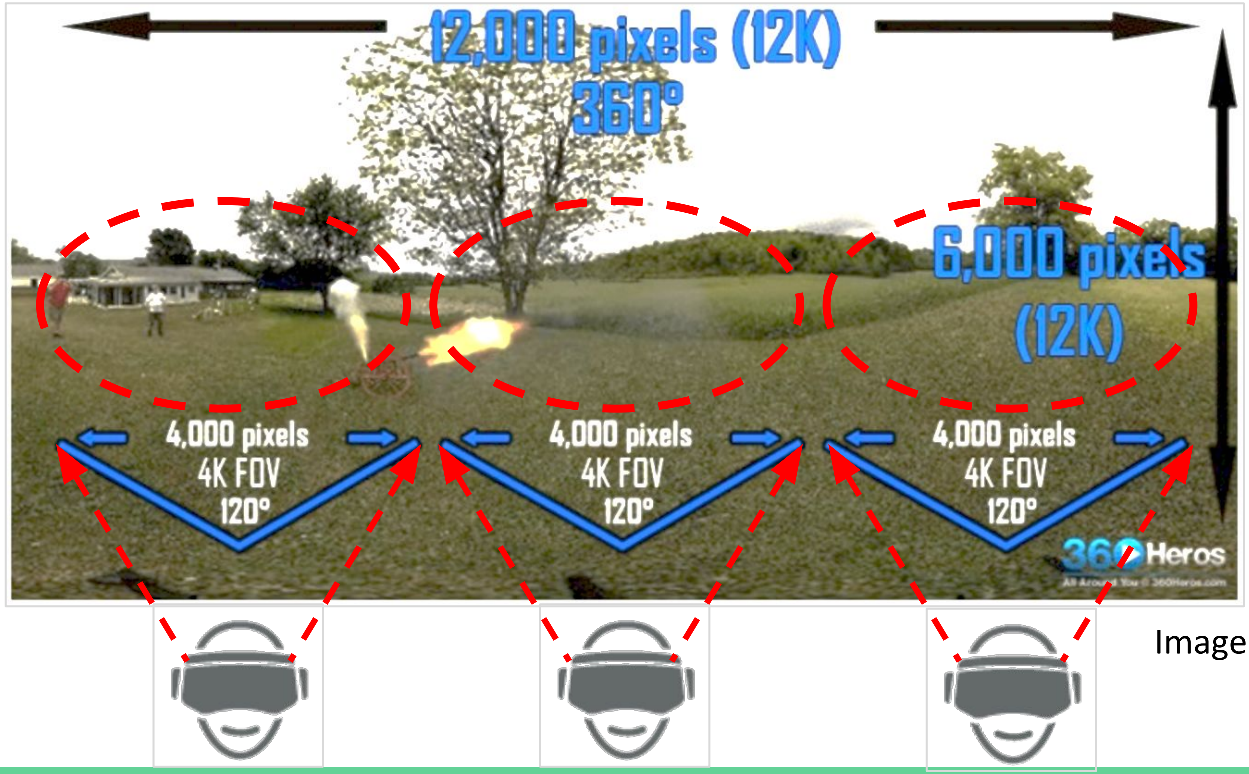
What's the Problem?

Streaming 360° Video is Challenging

- **High Bandwidth Demand**
  - contains wider view than conventional videos
  - very high resolution, such as 4K, 8K, and higher
- **Latency Sensitive**
  - human perception requires accurate and smooth movements
  - to avoid motion sickness
- **Heterogeneous HMD devices**
  - different HMDs has different computing power and network condition

Challenge #1: High Bandwidth Demand

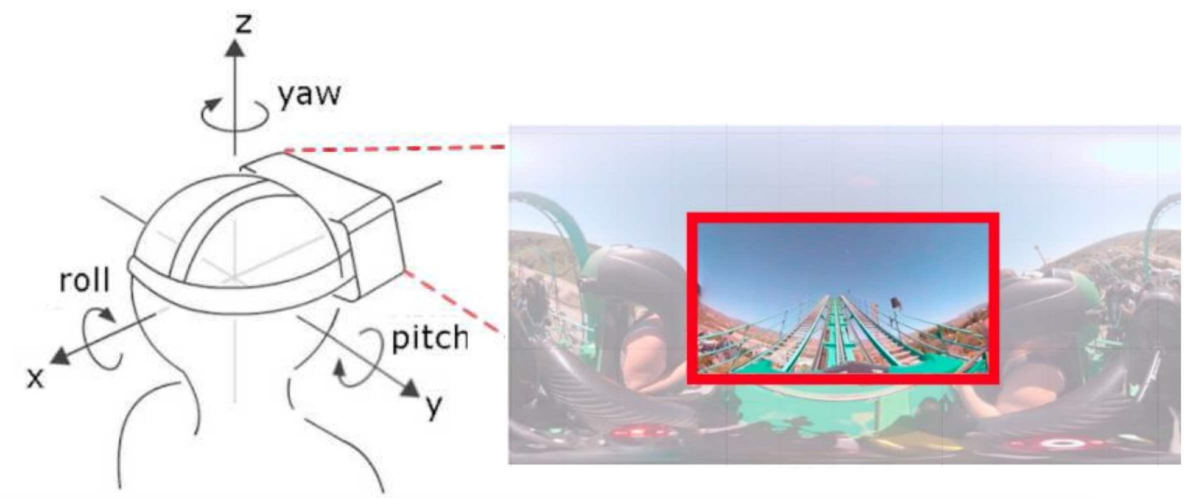
- We assume Field-of-View (FoV) is 100°x100°
- 4K resolution in FoV requires 12k/30fps resolution for the whole 360° videos (≈ 200 Mbps with HEVC)



Images source: 360Heros

Only Stream Field-of-View (FoV)

- Viewer actively changes viewing orientation when rotating his/her head
- HMD viewer only gets to see a small part of the whole 360° video (< 1/3)



Challenge #2: Latency Sensitive

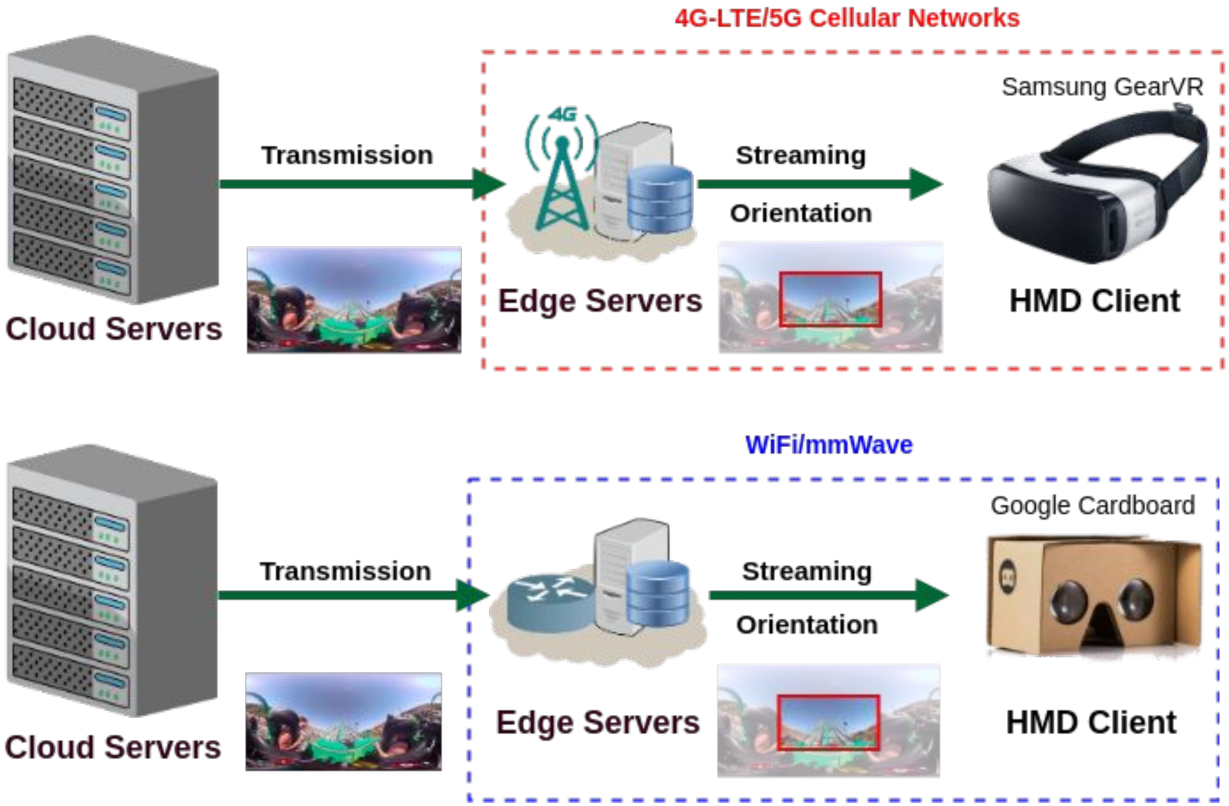
- Severe latency can lead to detached experience and motion sickness (latency < 60ms)[1] [2]

Resolution/FPS	Equivalent TV Res.	Bandwidth	Latency
4K/30fps	240p	25Mbps	**60ms**
8K/30fps	SD (480p)	100Mbps	**50ms**
12K/60fps	HD (720p)	400Mbps	**30ms**
24K/120fps	4K UHD (2160p)	2.35Gbps	**10ms**

[1] S. LaValle et al. “Head tracking for the Oculus Rift,” in Proc. of IEEE ICRA’14
[2] S. Mangiante et al. “VR is on the Edge: How to Deliver 360° Videos in Mobile Networks,” in Proc. of ACM VR/AR Network '17

Leverage Edge Devices

- Located closer to end users
- Cellular network & WLAN[1]



[1] R. Ford et al., "Achieving Ultra-Low Latency in 5G Millimeter Wave Cellular Networks," in IEEE Communications Magazine, vol. 55, no. 3, pp. 196-203, March 2017.

Challenge #3: Heterogeneous Devices

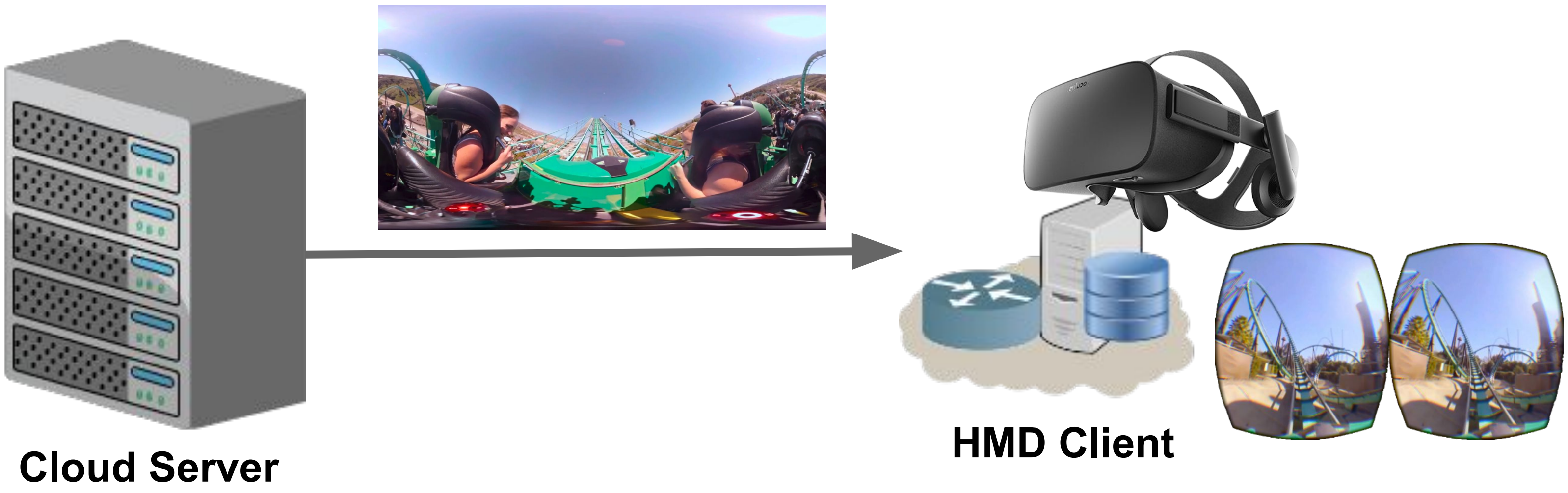
- Different HMD types
- There are mainly two types of HMDs

HMDs	Computing Power	Bandwidth	Battery powered	Mobility
PCs	powerful	High	No	No
Mobiles	weak	Medium/Low	Yes	Yes



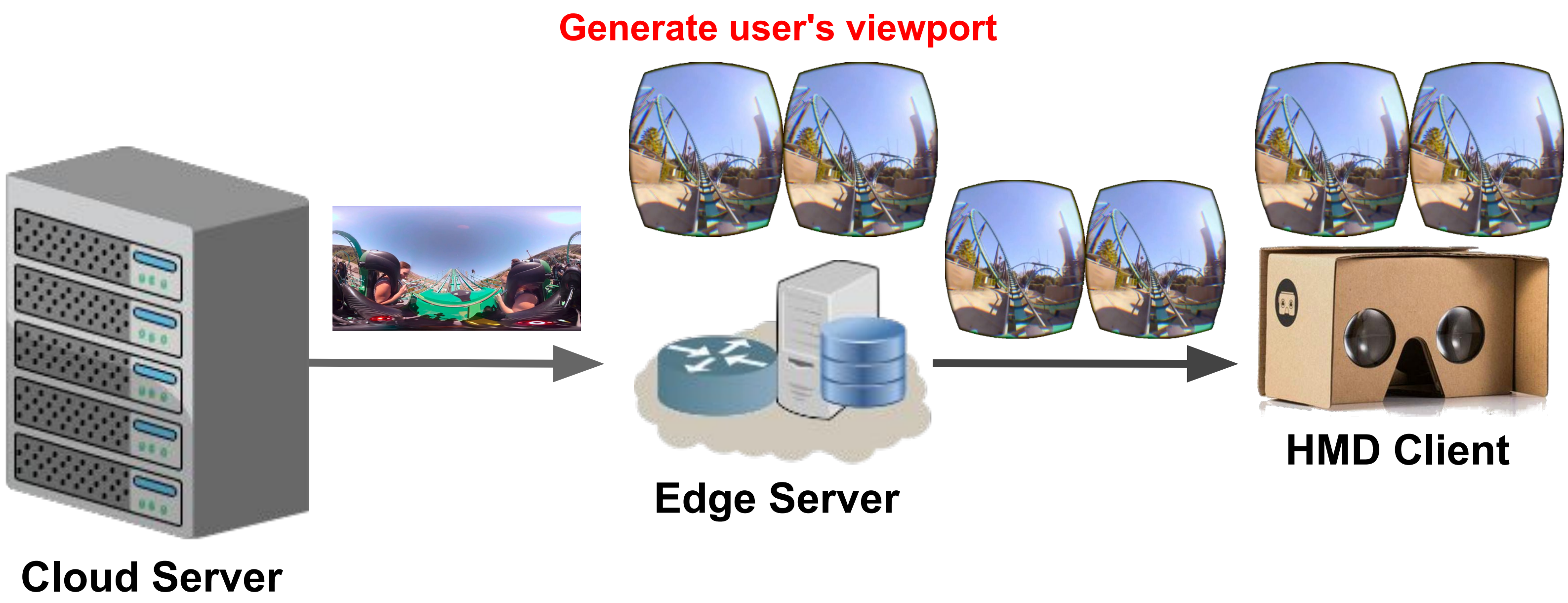
Edges Offloading

- Current streaming approach

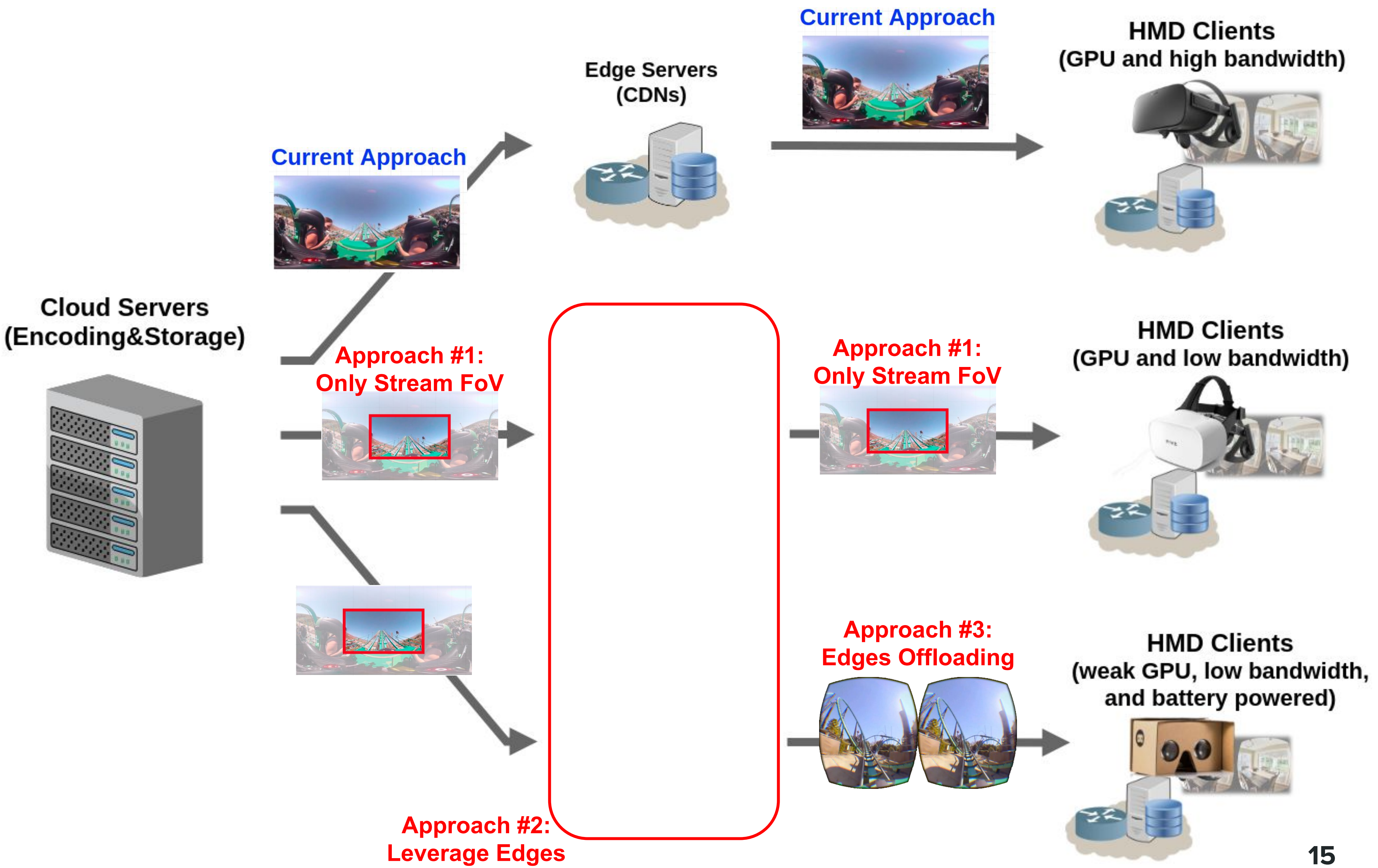


Edges Offloading

- Current Approach
- Edge Offloading
  - generate user's viewport on edges



Edge-assisted Streaming

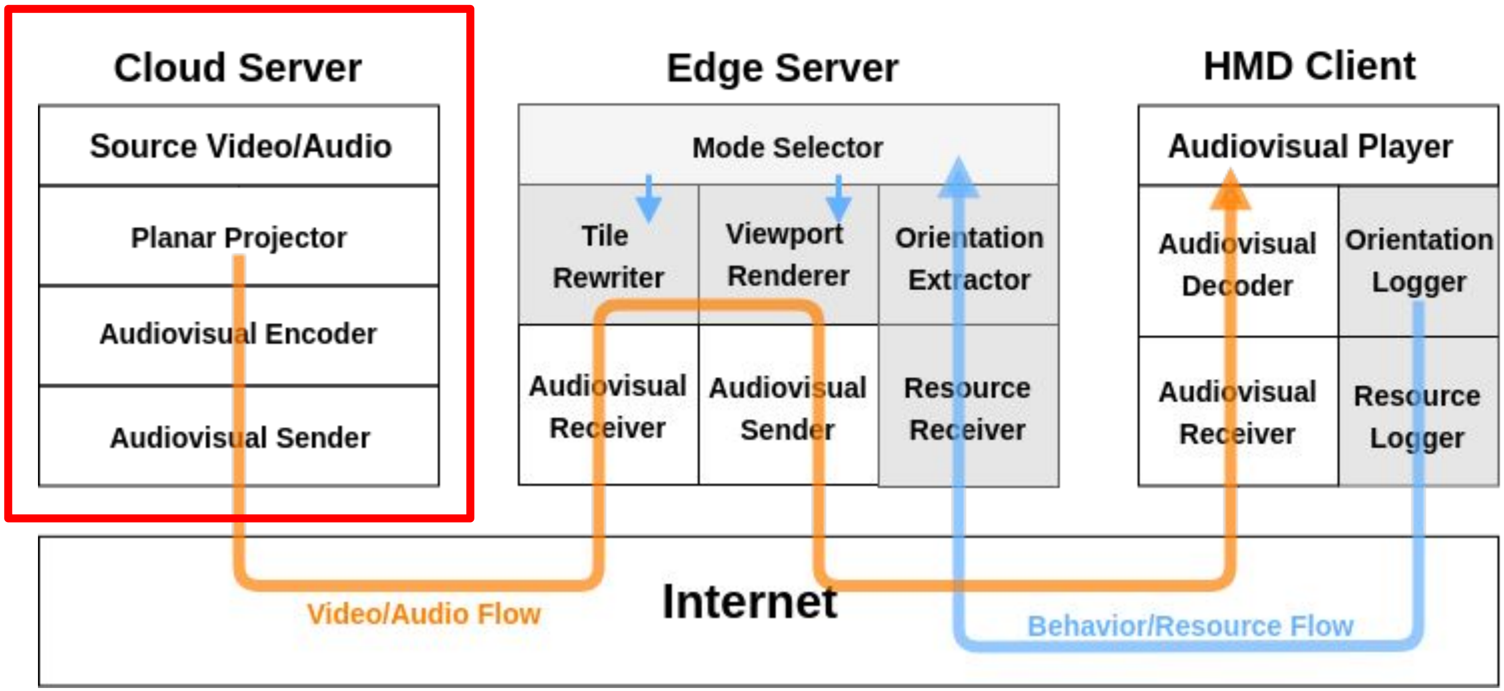


Contributions

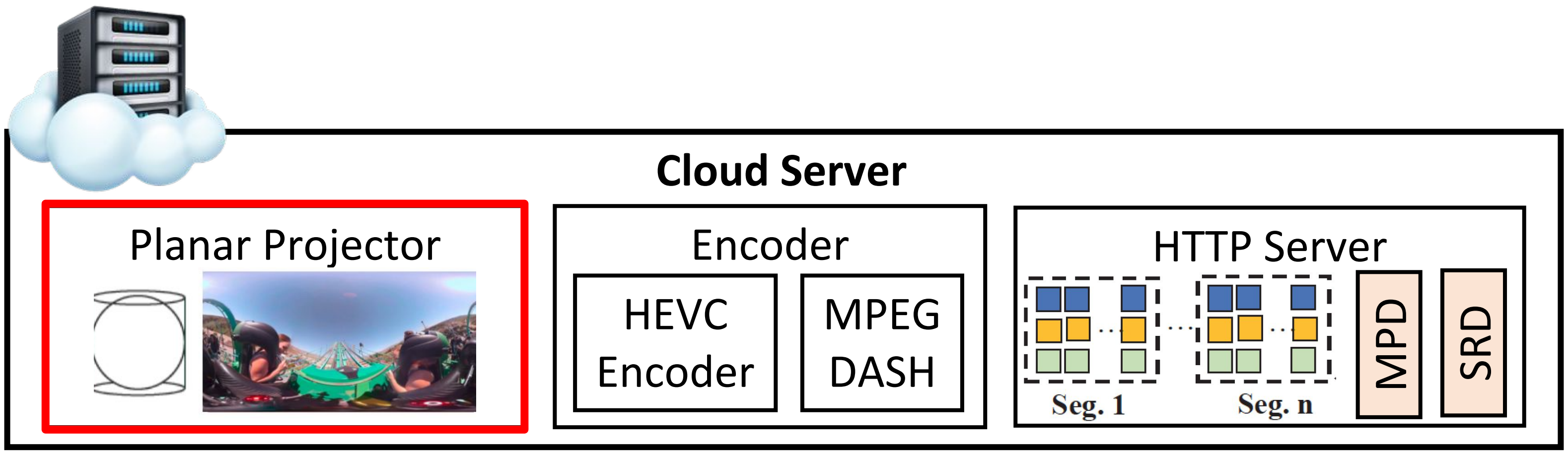
- Propose edge-assisted 360° video streaming system supporting aforementioned approaches
- Formulate and design an algorithm to slove edge-assisted streaming problem
- Quantify the performance of our proposed algorithm using an open-sourced 360° viewing dataset

System Architecture

System Overview



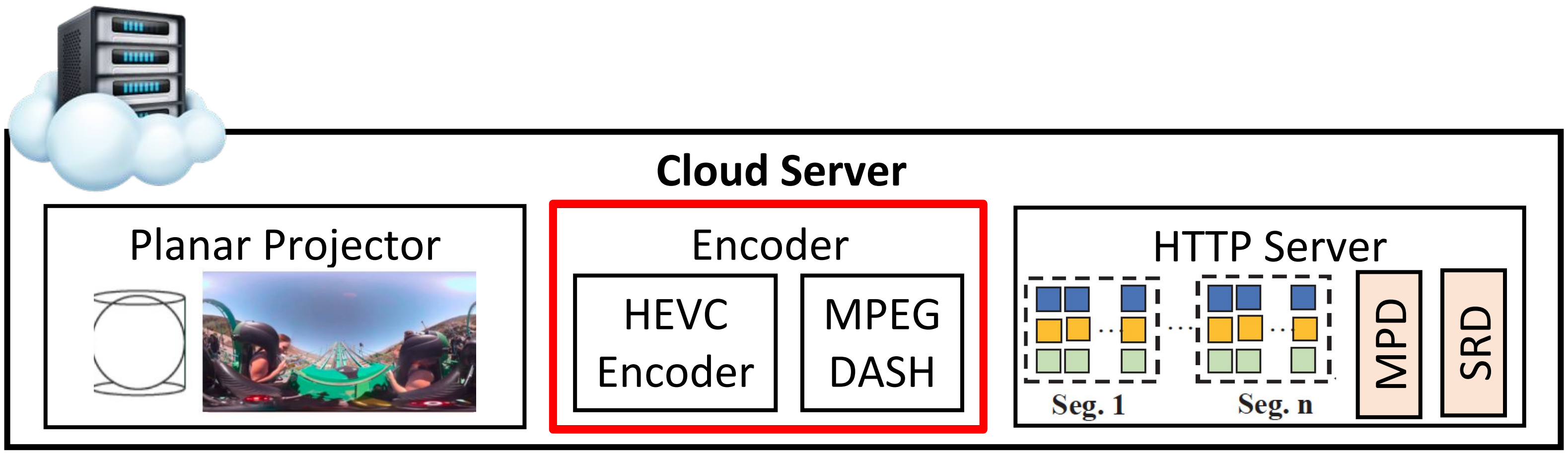
Cloud Server[1]



- Cloud Server
  - Planar projector

[1] W. Lo et al., "Performance Measurements of 360◦ Video Streaming to Head-Mounted Displays Over Live 4G Cellular Networks," in Proc. of APNOMS'17
[2] G. Sullivan et al. "Overview of the high efficiency video coding (HEVC) standard," in *IEEE Transactions on circuits and systems for video technology, vol.* 22, no. 12, pp. 1649-1668.
[3] ISO/IEC DIS 23009-1.2, "Dynamic adaptive streaming over HTTP (DASH)"

Cloud Server[1]



- Cloud Server
  - Planar projector
  - HEVC[2] encoder
  - MPEG DASH[3] content generator

[1] W. Lo et al., "Performance Measurements of 360◦ Video Streaming to Head-Mounted Displays Over Live 4G Cellular Networks," in Proc. of APNOMS'17
[2] G. Sullivan et al. "Overview of the high efficiency video coding (HEVC) standard," in *IEEE Transactions on circuits and systems for video technology, vol.* 22, no. 12, pp. 1649-1668.
[3] ISO/IEC DIS 23009-1.2, "Dynamic adaptive streaming over HTTP (DASH)"

Cloud Server

Tiles in HEVC

- Video is split into tiles of subvideos
- Compress with motion-constrained HEVC encoder[1]

[1] M. Viitanen, A. Koivula, A. Lemmetti, A. Ylä-Outinen, J. Vanne, and T. Hämäläinen, "Kvazaar: Open-Source HEVC/H.265 Encoder," in Proc. of ACM MM '16

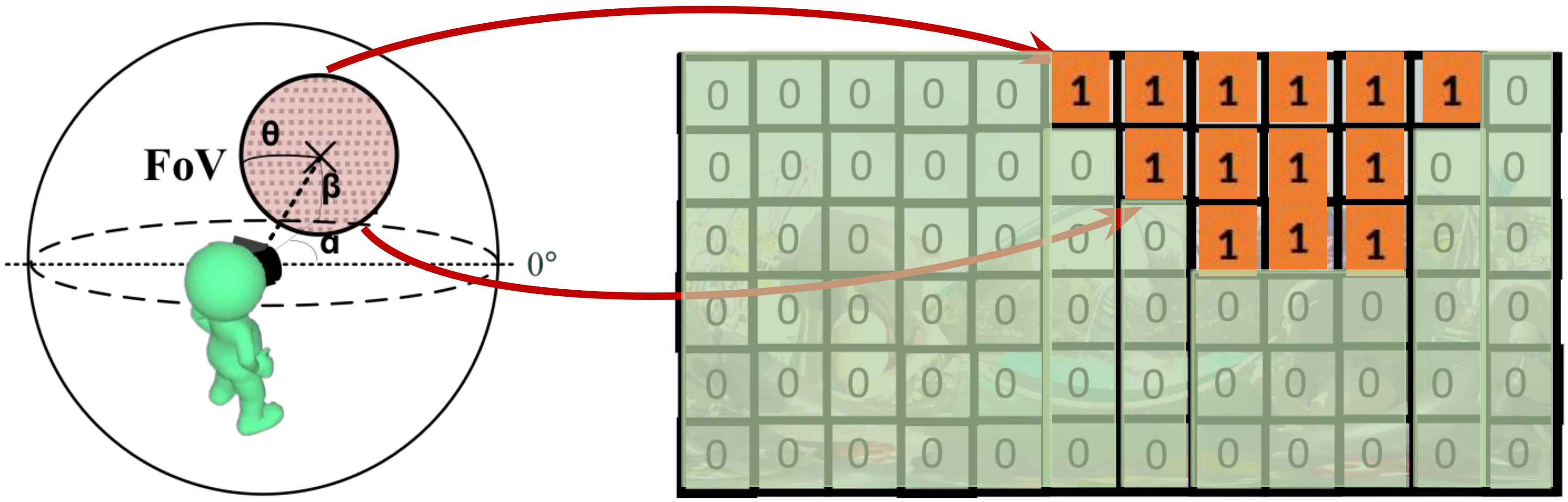
Cloud Server

Tiling with Dynamic Adaptive Streaming over HTTP (DASH)

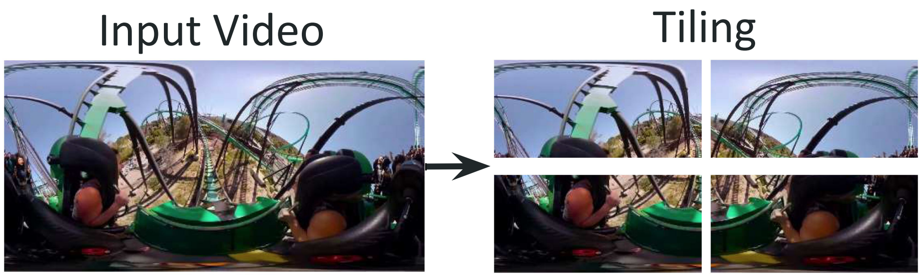
- Tiles are split into temporal segments (e.g., 2 secs)
    - qualities can be change at segment bundary

Tiling with Dynamic Adaptive Streaming over HTTP (DASH)

- Tiles overlapped with FoV are streamed in high-quality
- Others are streamed in low-quality



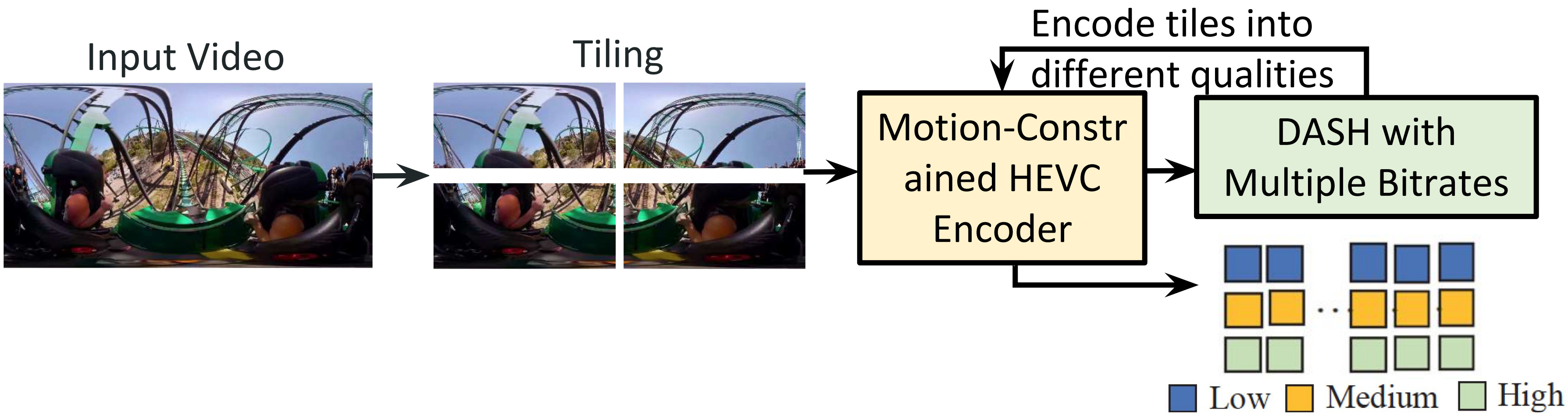
Cloud Server
Encoding Procedure



- Split the videos into tiles of sub-videos

Cloud Server

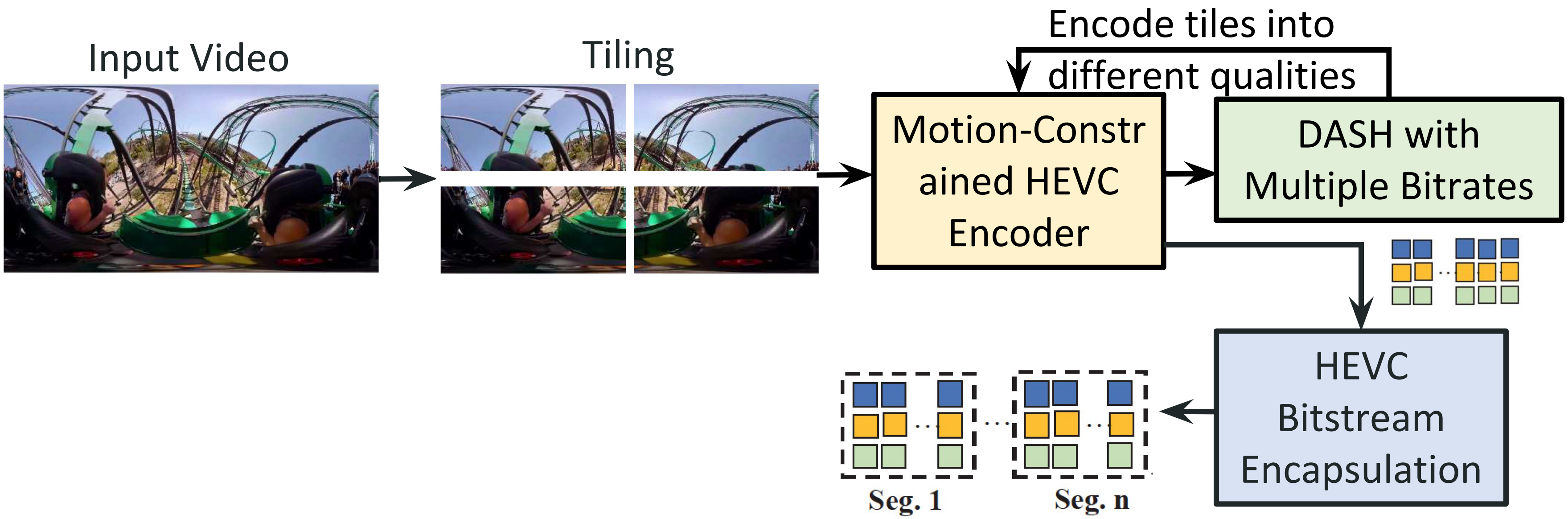
Encoding Procedure



- Split the videos into tiles of sub-videos
- Encode the tiles using motion-constrained HEVC encoder with different bitrates (qualities)

Cloud Server

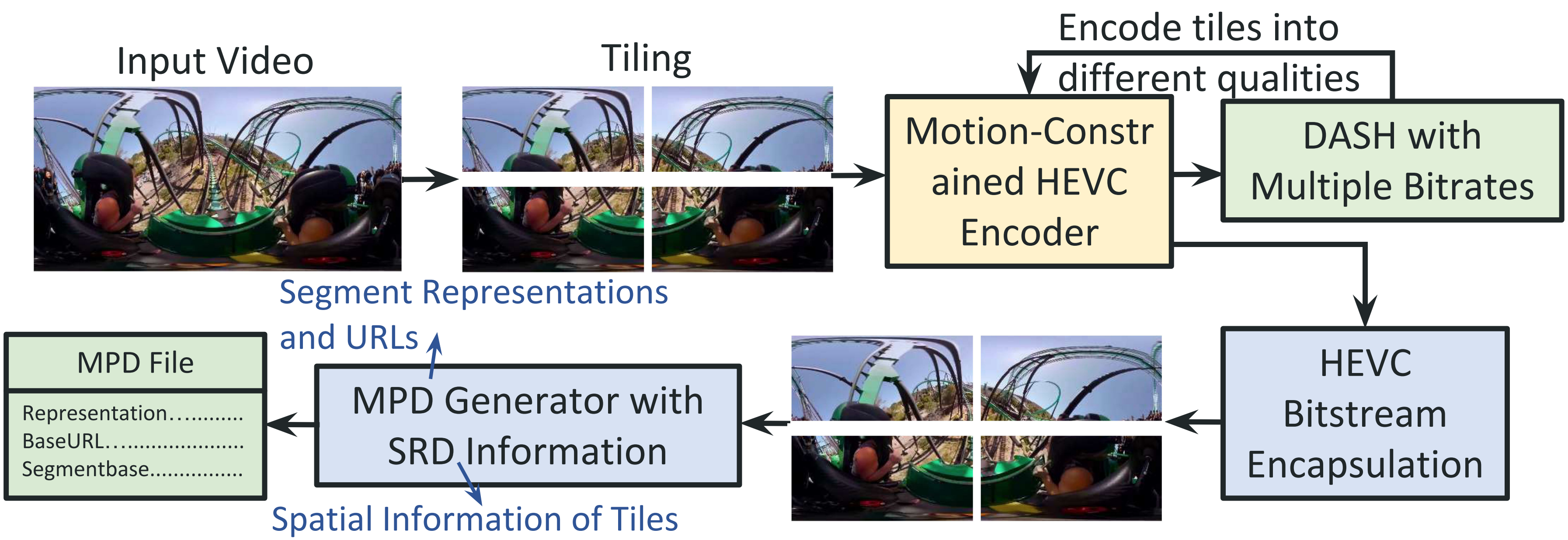
Encoding Procedure



- Split the videos into tiles of sub-videos
- Encode the tiles using motion-constrained HEVC encoder with different bitrates (qualities)
- Encapsulate tiles into HEVC bitstreams

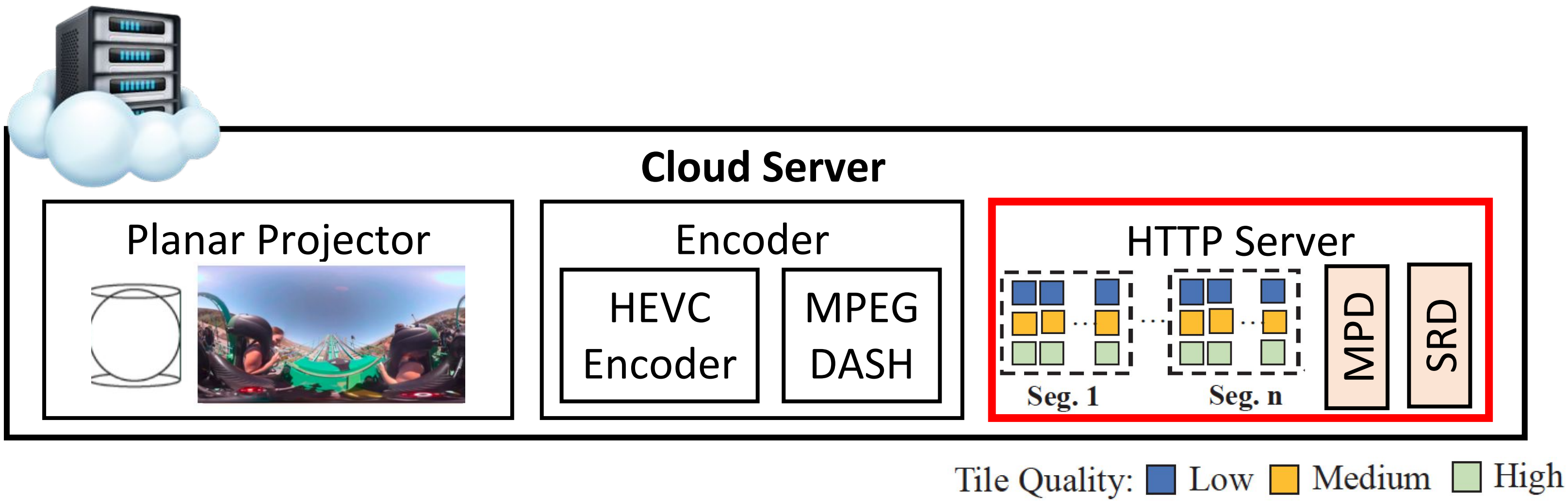
Cloud Server

Encoding Procedure



- Split the videos into tiles of sub-videos
- Encode the tiles using motion-constrained HEVC encoder with different bitrates (qualities)
- Encapsulate tiles into HEVC bitstreams
- Integrate with DASH for spatial index generation (MPD and SRD)

Cloud Server[1]



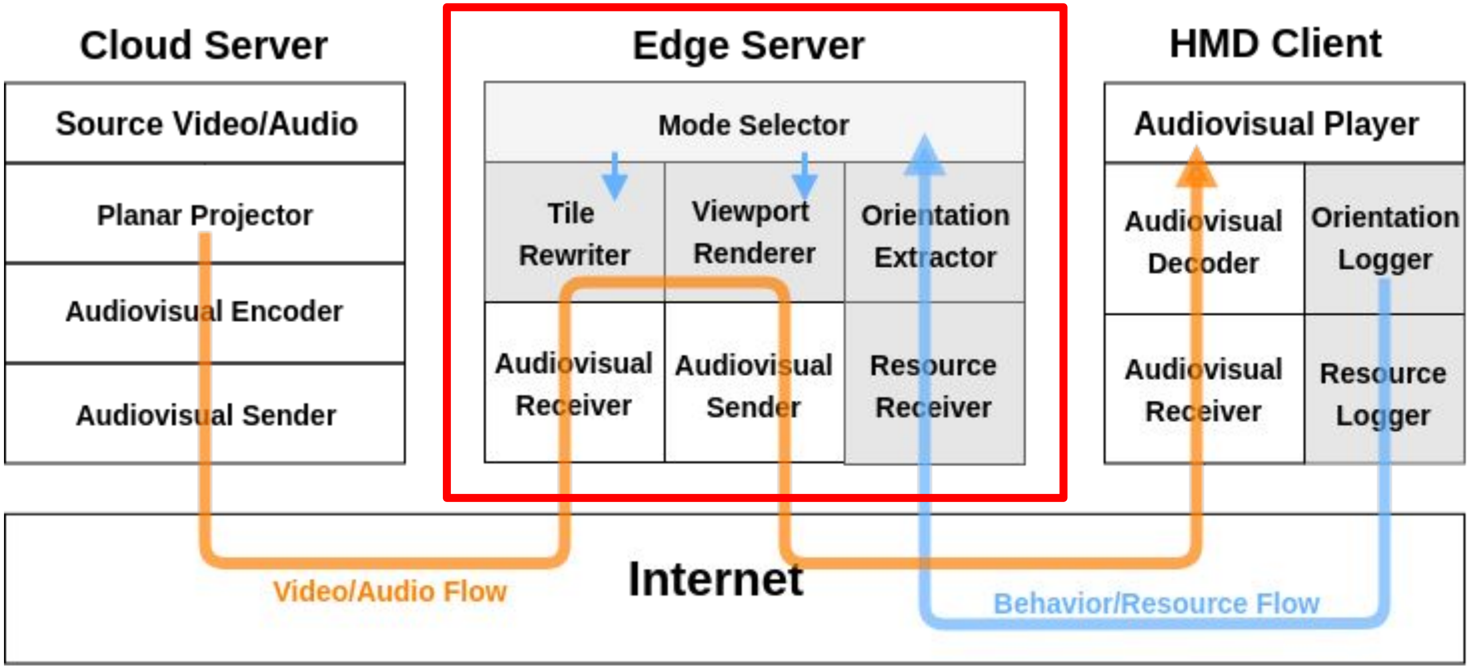
- Cloud Server
  - Planar projector
  - HEVC[2] encoder
  - MPEG DASH[3] content generator
  - HTTP Server

[1] W. Lo et al. "Performance Measurements of 360◦ Video Streaming to Head-Mounted Displays Over Live 4G Cellular Networks," in Proc. of APNOMS'17

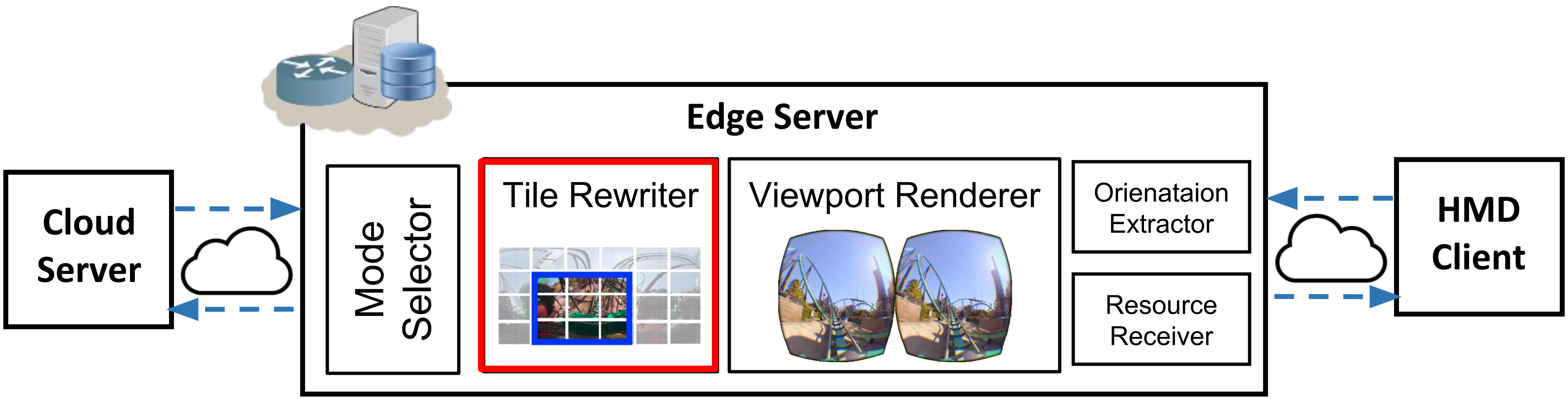
[2] G. Sullivan et al. "Overview of the high efficiency video coding (HEVC) standard." Sullivan, Gary J., et al. "Overview of the high efficiency video coding (HEVC) standard. " *IEEE Transactions on circuits and systems for video technology* 22 (12), 2012, 1649-1668.

[3] ISO/IEC DIS 23009-1.2 Dynamic adaptive streaming over HTTP (DASH)

System Overview



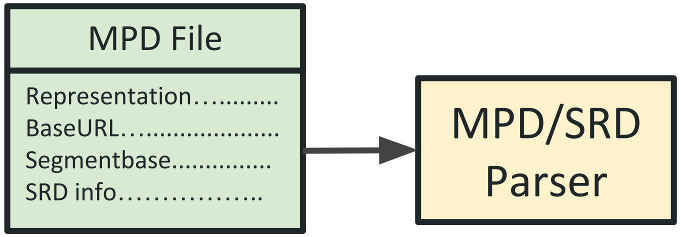
Edge Server



- **Edge Server**
  - **Tile Rewriter**

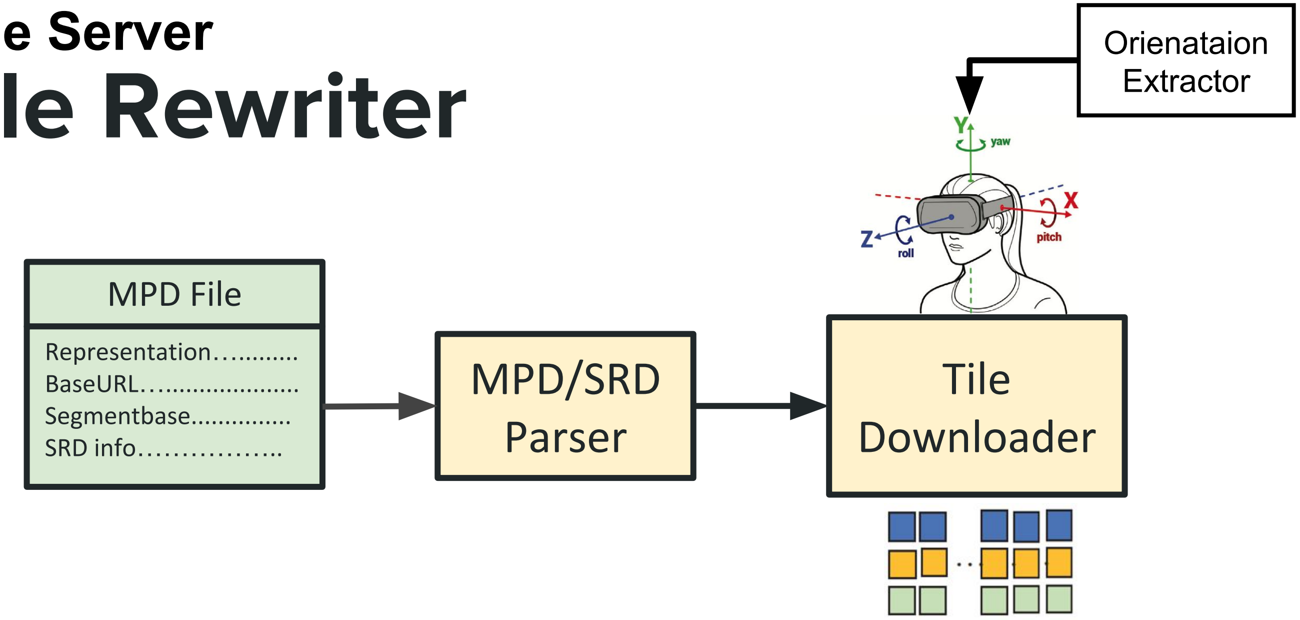
Edge Server

Tile Rewriter



- Parse MPD (with SRD info)

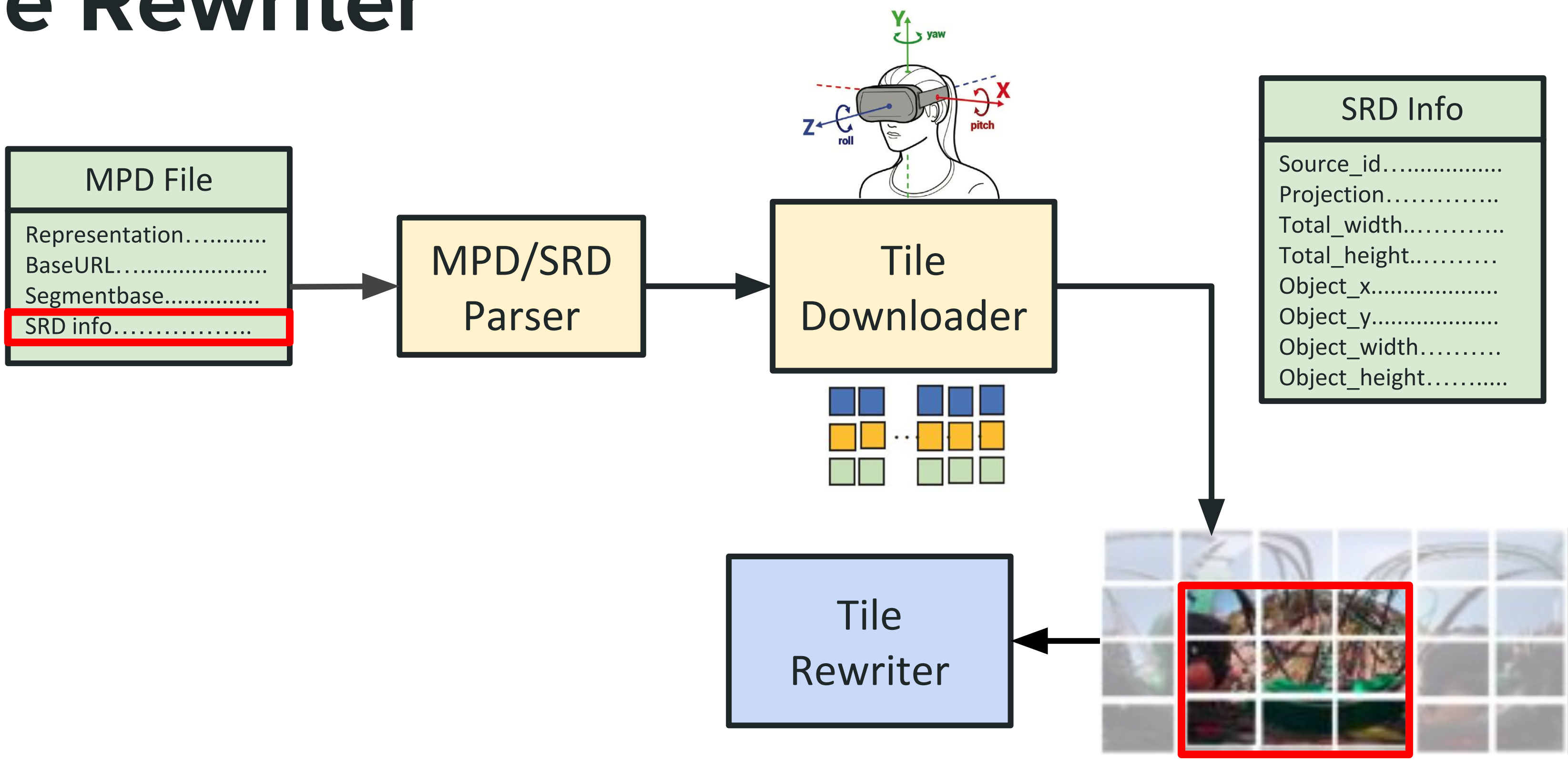
Edge Server
Tile Rewriter



- Parse MPD (with SRD info)
- Download tiles with different qualities

Edge Server

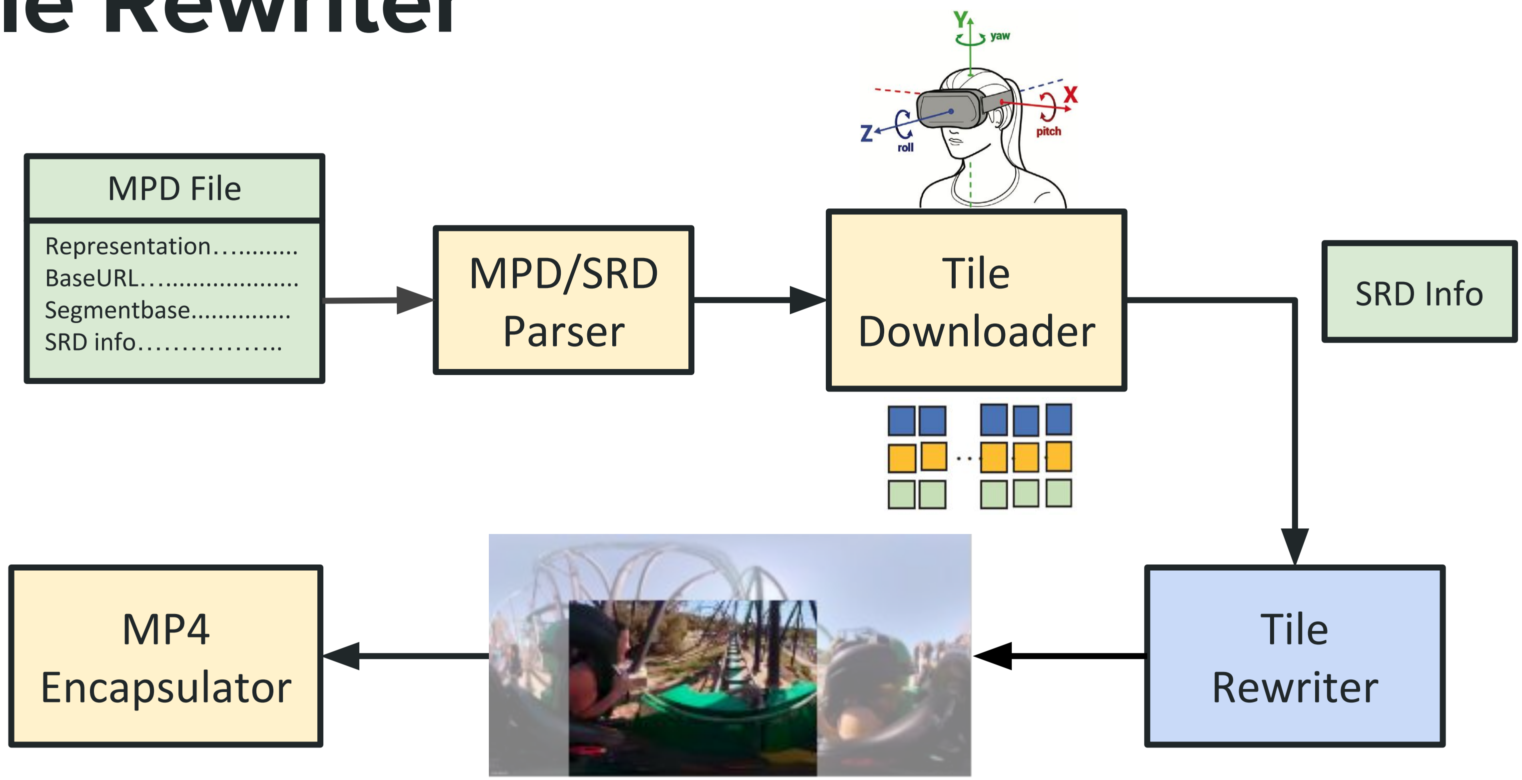
Tile Rewriter



- Parse MPD (with SRD info)
- Download tiles with different qualities
- Combine tiles into single HEVC bitstream based on SRD

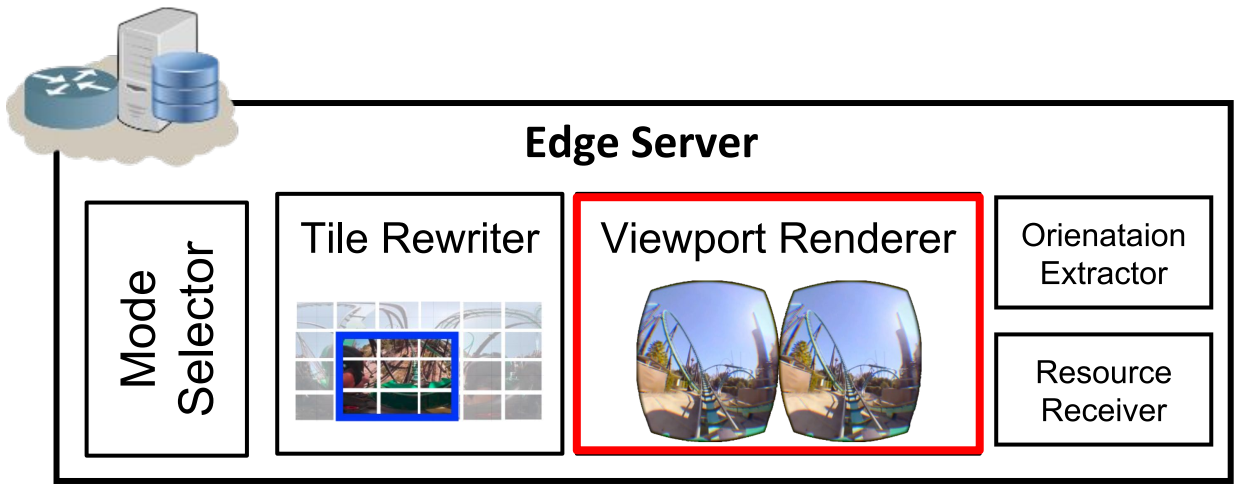
Edge Server

Tile Rewriter



- Parse MPD (with SRD info)
- Download tiles with different qualities
- Combine tiles into single HEVC bitstream based on SRD
- Encapsulate HEVC bitstream into MP4 container

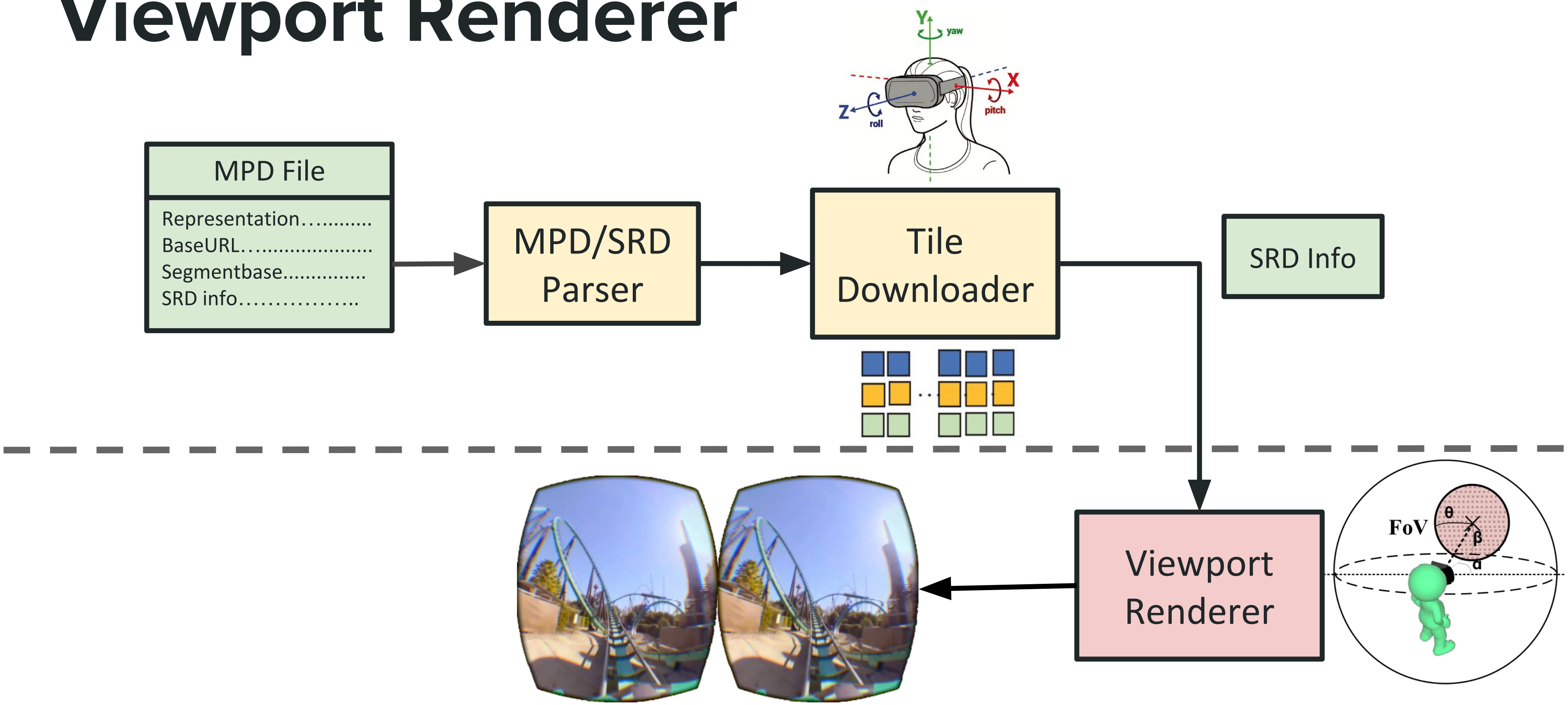
Edge Server



- **Edge Server**
  - Tile Rewriter
  - Viewport Renderer

Edge Server

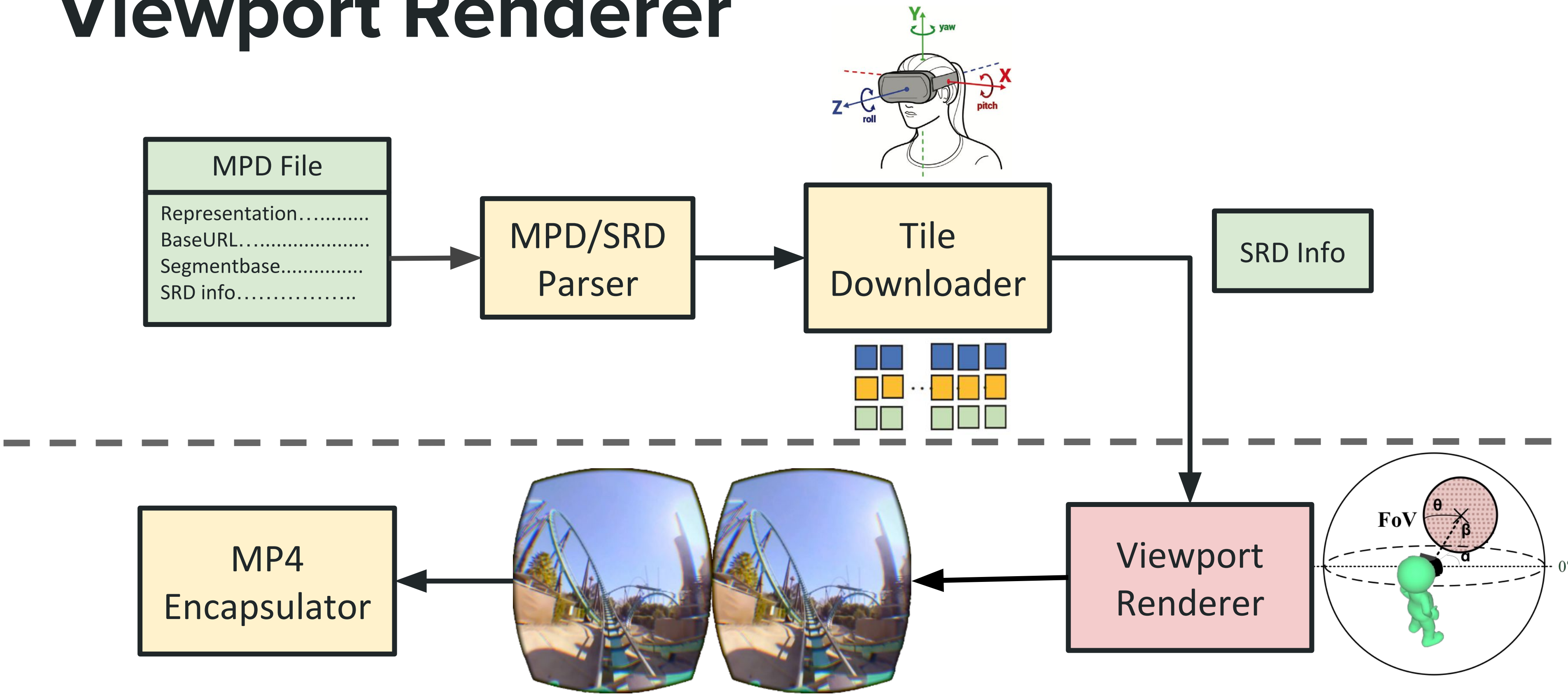
Viewport Renderer



- Parse MPD (with SRD info)
- Download tiles with different qualities
- Render user’s viewport scene (FoV size, FPS, and resolution)

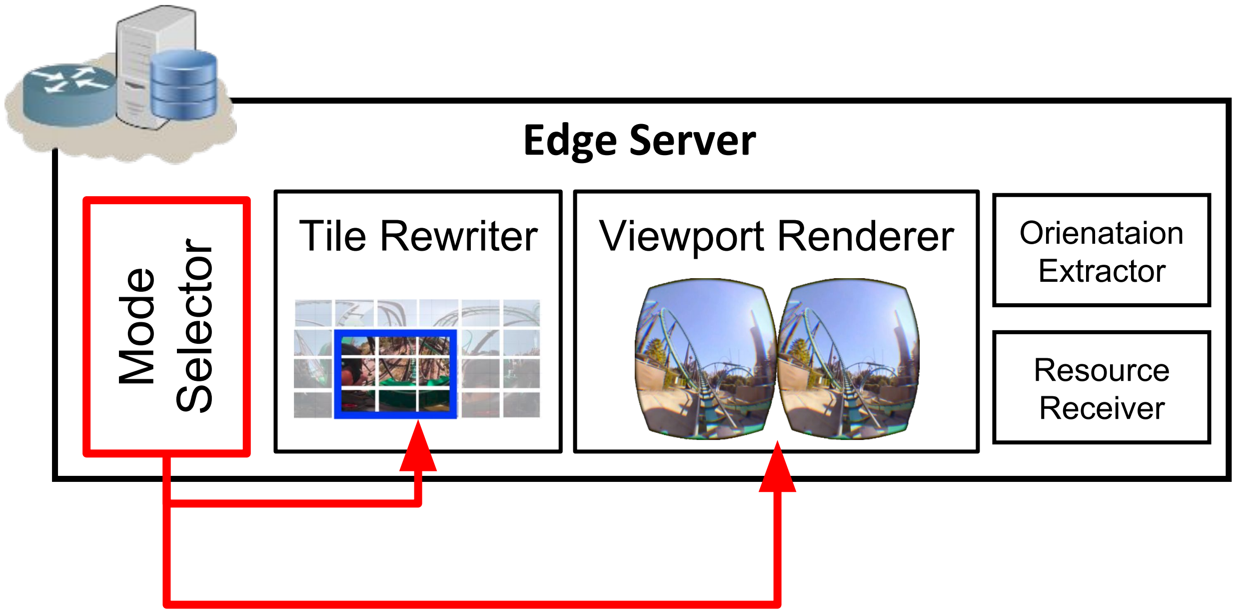
Edge Server

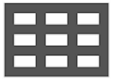
Viewport Renderer



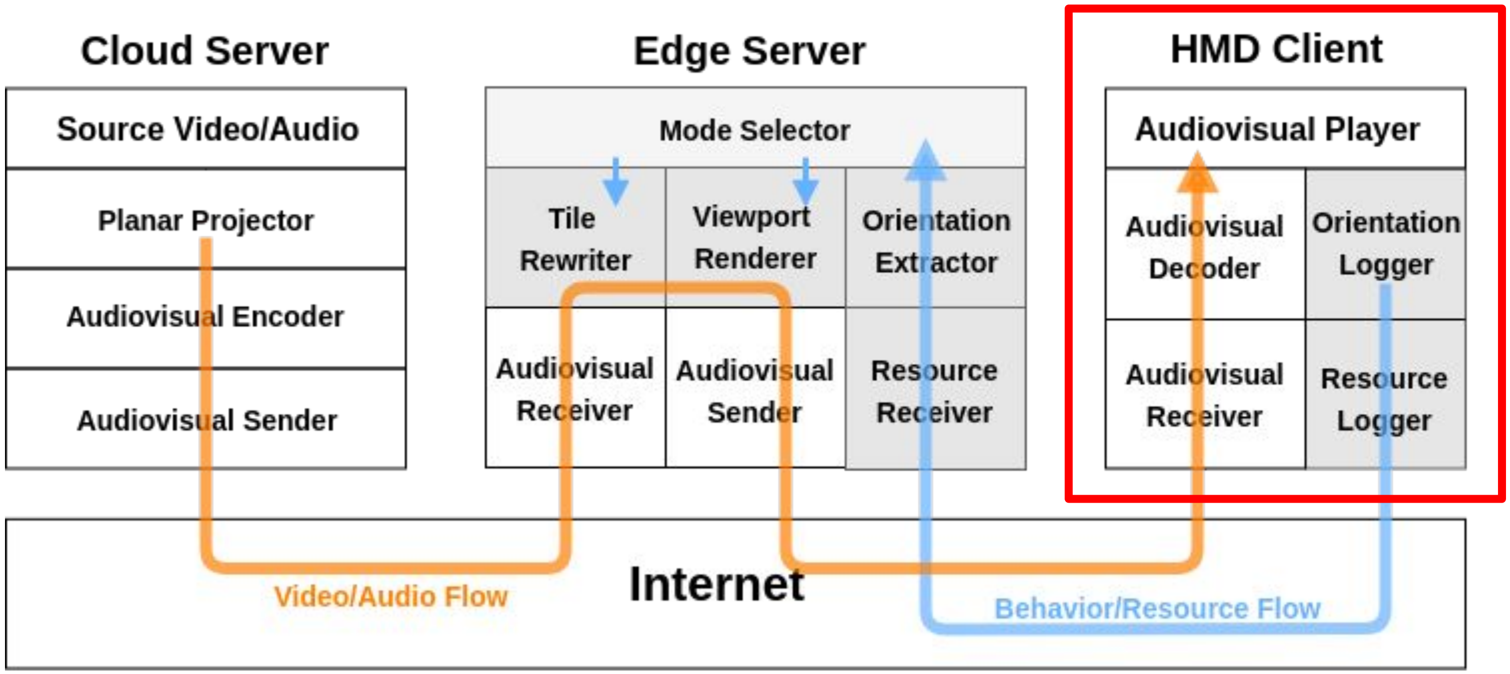
- Parse MPD (with SRD info)
- Download tiles with different qualities
- Render user's viewport scene (FoV size, FPS, and resolution)
- Encapsulate HEVC bitstream into MP4 container

Edge Server



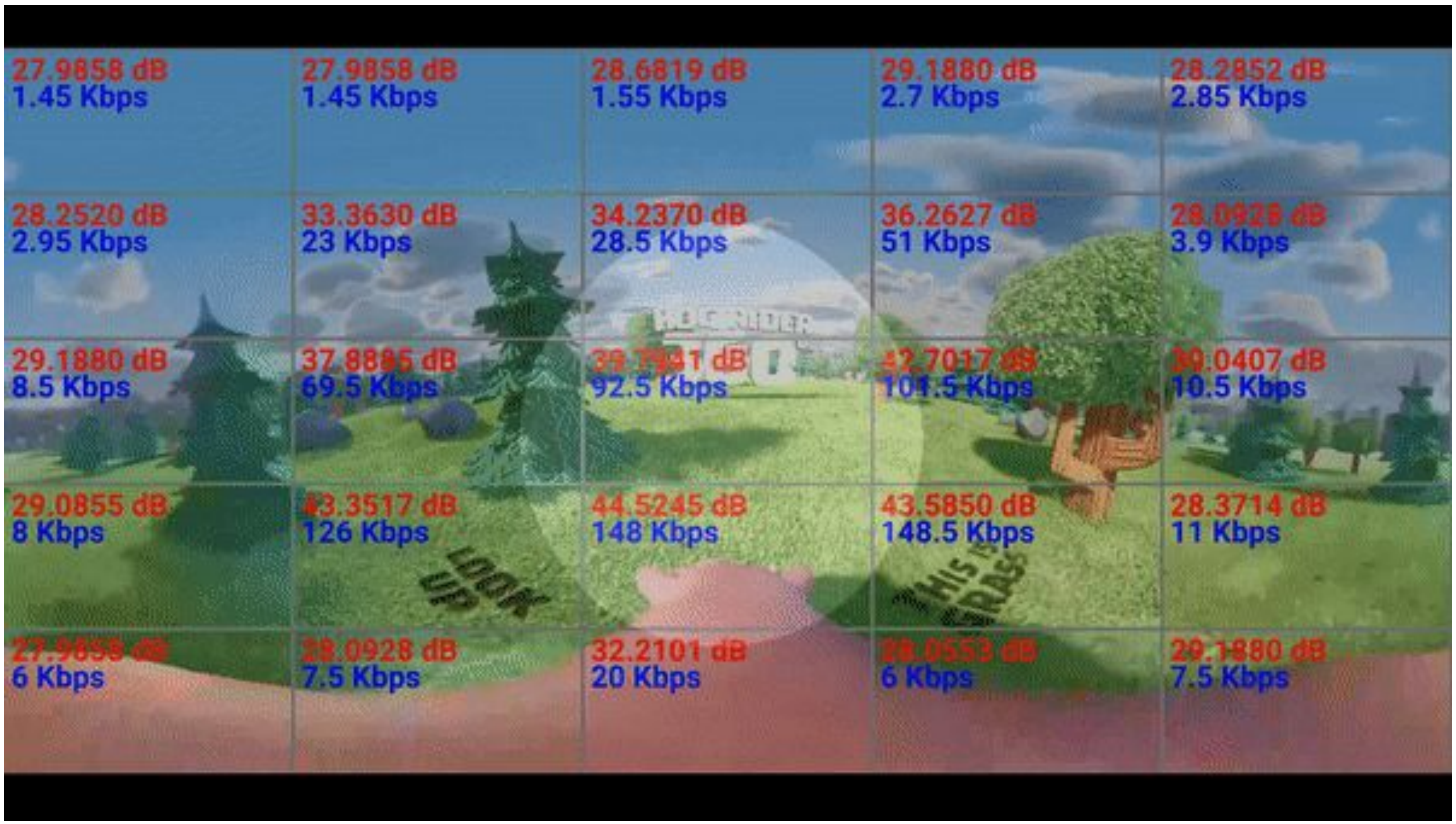
- **Edge Server**
  - **Tile Rewriter**
  - **Viewport Renderer**
  - **Mode Selector**
    - **Tile Rewriting (TR), as default setting** 
    - **Viewport Rendering (VPR)** 

System Overview



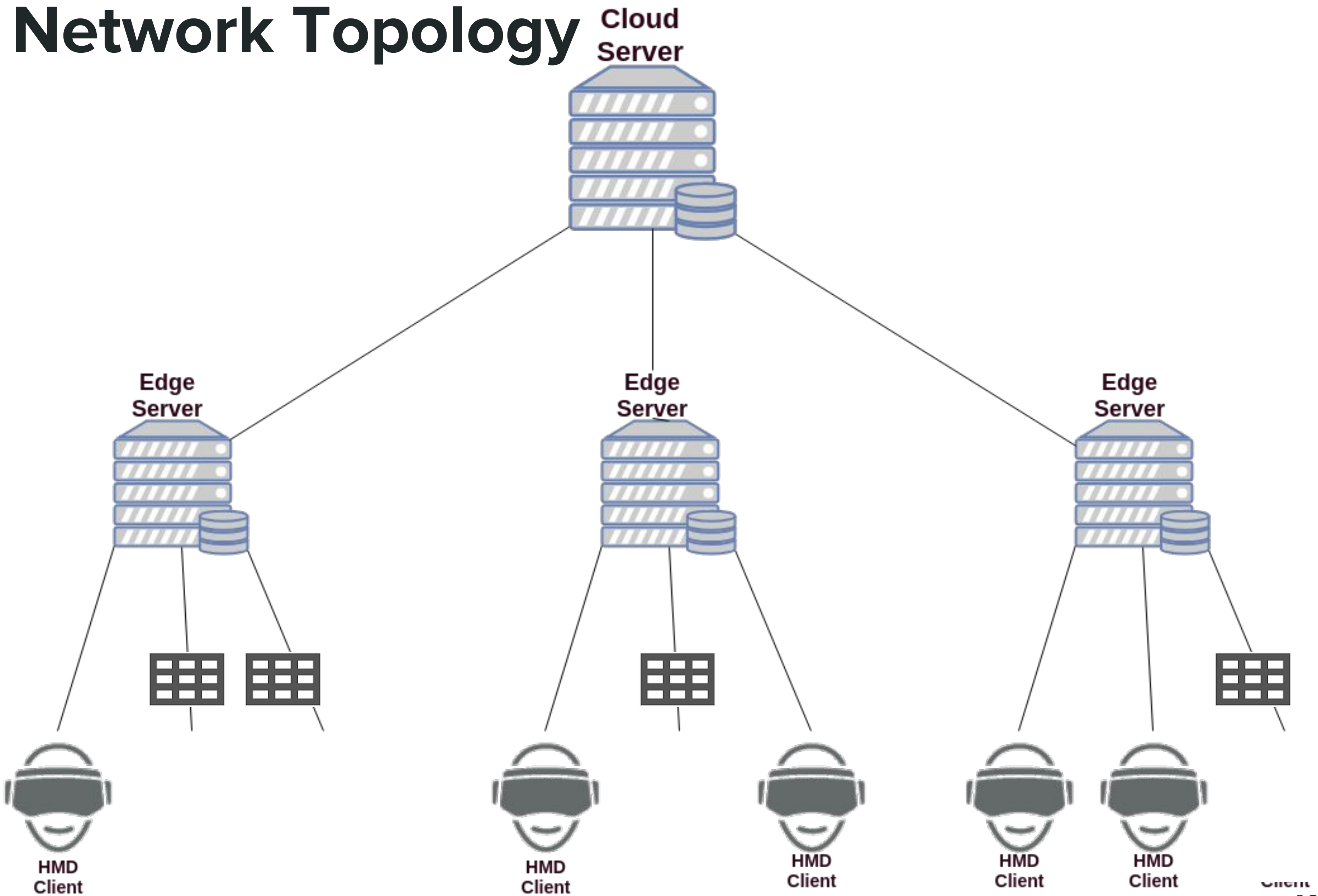
Demo

Demo


27.9858 dB
1.45 Kbps
27.9858 dB
1.45 Kbps
28.6819 dB
1.55 Kbps
29.1880 dB
2.7 Kbps
28.2852 dB
2.85 Kbps
28.2520 dB
2.95 Kbps
33.3630 dB
23 Kbps
34.2370 dB
28.5 Kbps
36.2627 dB
51 Kbps
28.0928 dB
3.9 Kbps
29.1880 dB
8.5 Kbps
37.8885 dB
69.5 Kbps
92.5 Kbps
42.7017 dB
101.5 Kbps
39.0407 dB
10.5 Kbps
29.0855 dB
8 Kbps
43.3517 dB
126 Kbps
44.5245 dB
148 Kbps
43.5850 dB
148.5 Kbps
28.3714 dB
11 Kbps
LOOK UP
6 Kbps
7.5 Kbps
32.2101 dB
20 Kbps
6 Kbps
7.5 Kbps

Optimal Edge-Assisted Streaming to HMDs

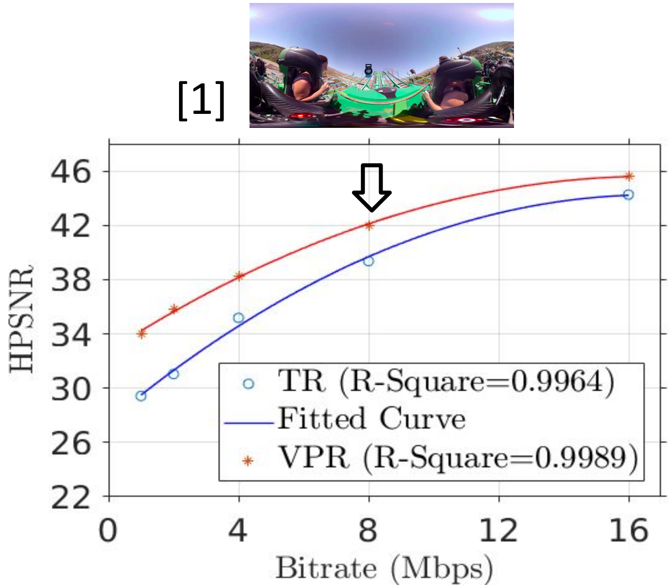
Network Topology



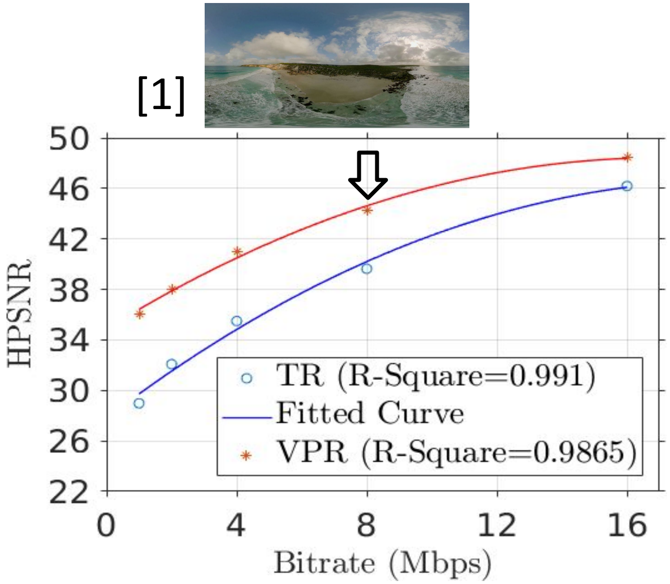
Problem Statement

- Limited resources of edge server
  - Computing power
  - Network bandwidth
- Capitalize edge server to assist HMDs to render scenes for maximizing the overall video quality improvement

[1]



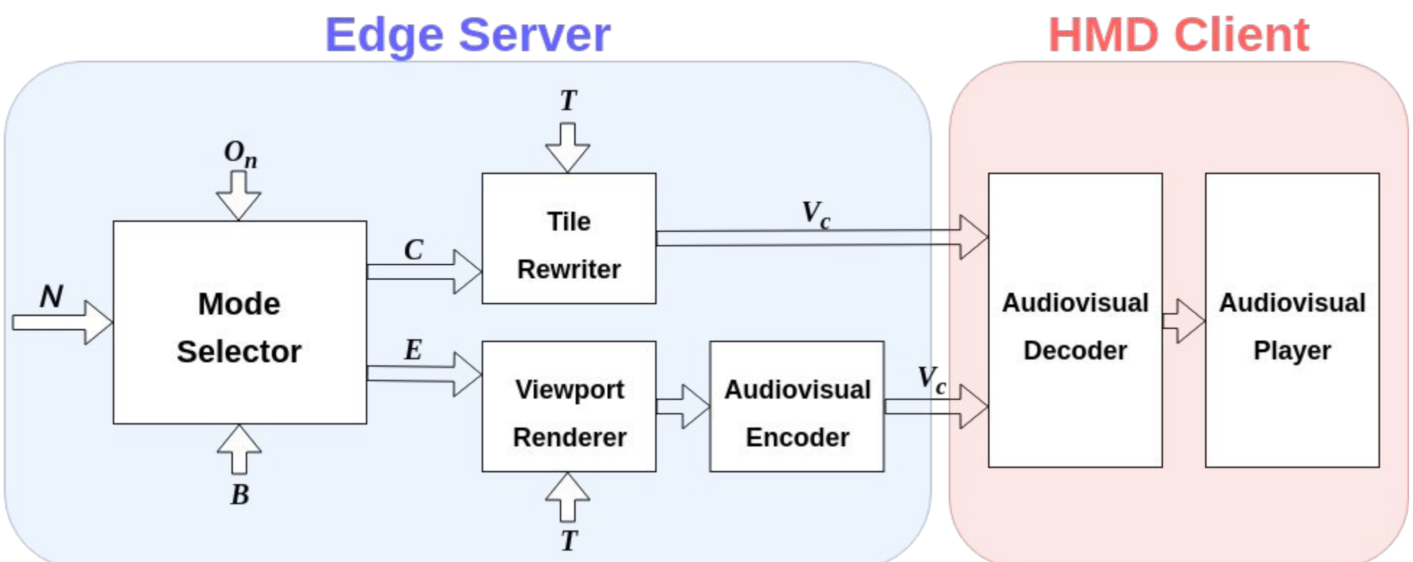
[1]



[1] W. Lo et al. "360° Video Viewing Dataset in Head-Mounted Virtual Reality," in Proc. of MMSys’17

Mode Selector

- **Goal**: maximize overall video quality imporvement Δq
  - avoid to overload edge server and exceed available bandwidth
  - fast and reliable
- We classify **N** HMD clients into two groups:
  - **C**, Tile Rewriting
  - **E**, Viewport Rendering



Problem Formulation

$$\text{maximize } 1/N \cdot \sum_{n=1}^{N} x_n (q'_n - q_n) \qquad (1a)$$

Objective: Maximize overall video quality improvement

$$s.\ t. \sum_{n=1}^{N} x_n \le E \qquad (1b)$$

Avoid to overload the edge server

$$\sum_{n=1}^{N} [x_n (f_n^w f_n^h b_h) + (1 - x_n)(f_n^w f_n^h b_h + (T - f_n^w f_n^h) b_l)] \le B \qquad (1c)$$

Consumed bandwidth of VPR

Consumed bandwidth of TR

$$x_n \in \{0, 1\}, \forall n = 1, 2, \ldots, N \qquad (1d)$$

Outbound bandwidth of edge server doesn’t exceed the available bandwidth

Proposed Algorithm

Algorithm 1 Mode Selector.

// We first initialize variables
for each n in N **do**
  $\mathbf{x}[\mathbf{n}] \leftarrow 0$; $\mathbf{Qual}[\mathbf{n}] \leftarrow q'_n - q_n$; $\mathbf{Band}[\mathbf{n}] \leftarrow \beta_n$; $\mathbf{Rati}[\mathbf{n}] \leftarrow (q'_n - q_n)/(\beta_n - \alpha_n)$
sort $\mathbf{Qual}[\mathbf{N}]$, $\mathbf{Band}[\mathbf{N}]$, $\mathbf{Rati}[\mathbf{N}]$ in desc. order
while $E > 0$ **do**
  pop an HMD client n with the maximal $\mathbf{Qual}[\mathbf{n}]$
  $\mathbf{x}[\mathbf{n}] \leftarrow 1$
  $\mathbf{Band}[\mathbf{n}] \leftarrow \alpha[n]$
  $E = E - 1$
if $\mathbf{sum}(\mathbf{Band}[\mathbf{N}]) \leq B$ **then**
  return $\mathbf{x}[\mathbf{N}]$
Initialize E, $x[N]$, and $\mathbf{Band}[\mathbf{N}]$
while $E > 0$ **do**
  pop an HMD client n with the maximal $\mathbf{Rati}[\mathbf{n}]$
  $\mathbf{x}[\mathbf{n}] \leftarrow 1$
  $\mathbf{Band}[\mathbf{n}] \leftarrow \alpha[n]$
  $E = E - 1$
if $\mathbf{sum}(\mathbf{Band}[\mathbf{N}]) \leq B$ **then**
  return $\mathbf{x}[\mathbf{N}]$
Initialize E. $x[N]$, and $\mathbf{Band}[\mathbf{N}]$
while $E > 0$ **do**
  pop an HMD client n with the maximal $\mathbf{Band}[\mathbf{n}]$
  $\mathbf{x}[\mathbf{n}] \leftarrow 1$
  $\mathbf{Band}[\mathbf{n}] \leftarrow \alpha[\mathbf{n}]$
  $E = E - 1$
if $\mathbf{sum}(\mathbf{Band}[\mathbf{N}]) \leq B$ **then**
  return $\mathbf{x}[\mathbf{N}]$
return no feasible solution

Calculate Δq and saved bandwidth

Sort in desc. order

Select with maximal video quality improvement

Check if exceed available bandwidth

Select with maximal ratio of quality improvment to saved bandwidth

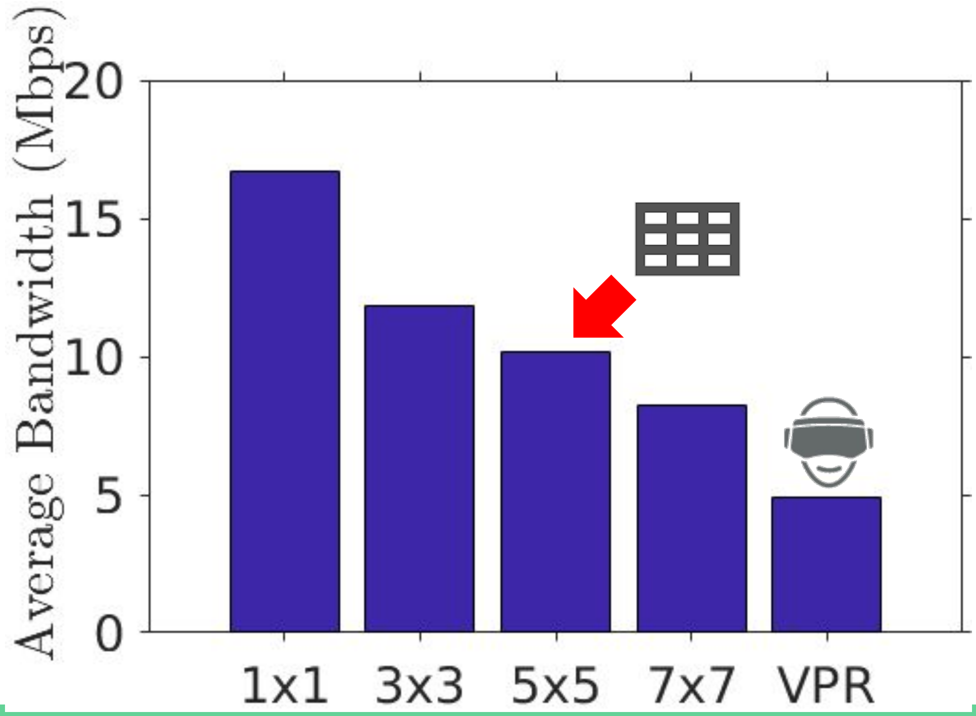
Check if exceed available bandwidth

Select with maximal saved bandwidth

Check if exceed available bandwidth

Lemma 1: Optimal Quality Improvement

- Bandwidth constraint is loose
  - if consumed bandwidth of all HMDs adopting TR does not exceed available bandwidth
- 0-1 Knapsack problem
  - each item has the same weight



Tile Rewriting = {3x3, 5x5, 7x7}

Viewport Rendering (VPR)

Encoding bitrate: 8 Mbps / 1Mbps

Lemma 1: Optimal Quality Improvement

- Greedy method
  - we always take whatever items (i.e., HMDs) are the most valuable (i.e., maximum video quality improvement)

- **Proof** (Contrapositive):

Let $\mathbf{Z}$ be the maximal video quality improvement set, where $\mathbf{Z} = \{z_1, z_2, \cdots z_E\}$, $z_i = q'_j - q_j$, and $\forall j \in \{1, 2, \cdots, N\}$. Then, the optimal solution is $\sum_{i=1}^{E}(z_i)$.

Suppose that $\exists z_m$, then we can replace z_i with z_m, where $z_i \in \mathbf{Z}$. Then, we denote $\hat{\mathbf{Z}} = \{z_1, z_2, \cdots, z_m, \cdots z_E\}$. So that $\sum_{i=1}^{E}(\hat{z}_i) > \sum_{i=1}^{E}(z_i)$
Therefore, $z_m > z_i$. This is contradiction.

Lemma 2: Runs in Polynomial Time

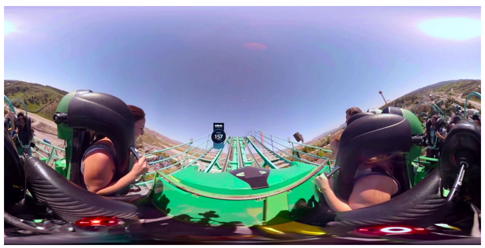
- Each round need to do
  - Calculate video quality improvement / bandwidth saving **O(N)**
  - Sort **Qual[N]**, **Band[N]**, **Ratio[N]** in desc. order **O(NlogN)**
  - Pick first **E** HMD clients **O(E)**
  - Calculate the total consumed bandwidth **O(N)**

(Complexity). *Algorithm 1 runs* ***O(E + NlogN)***

360 Viewing Dataset

360° Viewing Dataset[1]

- We collect ten 360° videos from YouTube
- 4K resolution, 30 fps, and 1 minute



Nature Image, fast-paced

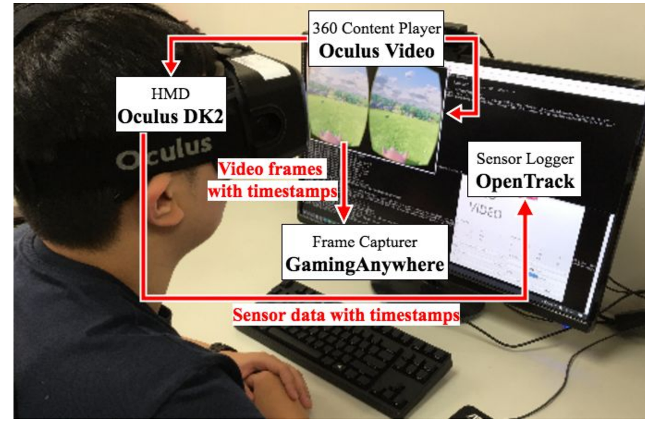
Nature Image, slow-paced

Computer Grapic, fast-paced

Category	Videos	Used Segment	Size (MB)	Link
NI, fast-paced	Mega Coaster	1:30 - 2:30	160	https://youtu.be/-xNN-bJQ4vI
	Roller Coaster	0:20 - 1:20	153	https://youtu.be/8lsB-P8nGSM
	Driving with	0:48 - 1:48	117	https://youtu.be/LKWXHKFCMO8
NI, slow-paced	Shark Shipwreck	0:30 - 1:30	114	https://youtu.be/aQd41nbQM-U
	Perils Panel	0:10 - 1:10	60	https://youtu.be/kiP5vWqPryY
	Kangaroo Island	0:01 - 1:01	126	https://youtu.be/MXlHCTXtcNs
	SFR Sport	0:16 - 1:16	51	https://youtu.be/lo5N90TlzwU
CG, fast-paced	Hog Rider	0:00 - 1:00	138	https://youtu.be/yVLfEHXQk08
	Pac-Man	0:10 - 1:10	50	https://youtu.be/p9h3ZqJa1iA
	Chariot Race	0:02 - 1:02	149	https://youtu.be/jMyDqZe0z7M

[1] W. Lo et al. "360° Video Viewing Dataset in Head-Mounted Virtual Reality," in Proc. of MMSys’17

360° Viewing Dataset



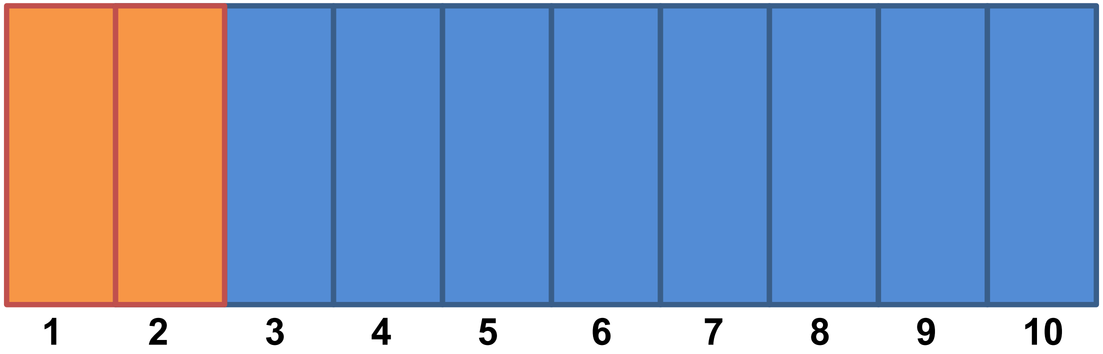
- 50 subjects
- Collect from HMDs while viewers are watching 360° videos
- Frame Capturer: GamingAnywhere[1]
- Sensor Logger: OpenTrack[2]

```
no. frames, raw x, raw y, raw z, raw yaw, raw pitch, raw roll, cal. yaw, cal. pitch, cal. roll
00001, 16.458 ,30.032, -19.276, -9.661, 5.853, -3.068, -4.65473888889, 4.06641388889, -3.068
00002, 16.458 ,30.032, -19.276, -9.661, 5.853, -3.068, -4.65473888889, 4.06641388889, -3.068
00003, 16.449 ,30.02, -19.362, -9.763, 5.746, -3.184, -4.75673888889, 3.95941388889, -3.184
00004, 16.449 ,30.02, -19.362, -9.763, 5.746, -3.184, -4.75673888889, 3.95941388889, -3.184
00005, 16.433 ,30.007, -19.473, -9.676, 5.659, -3.308, -4.66973888889, 3.87241388889, -3.308
...
```

[1] GamingAnywhere, http://gaminganywhere.org/
[2] OpenTrack, https://github.com/opentrack/opentrack

Partition Dataset

- 80% as training set
  - 40 subjects x 10 videos = 400 samples
  - generates video qaulity model of our system
- 20% as evaluation set
  - 10 subjects x 10 videos = 100 samples
  - conducts the experiments



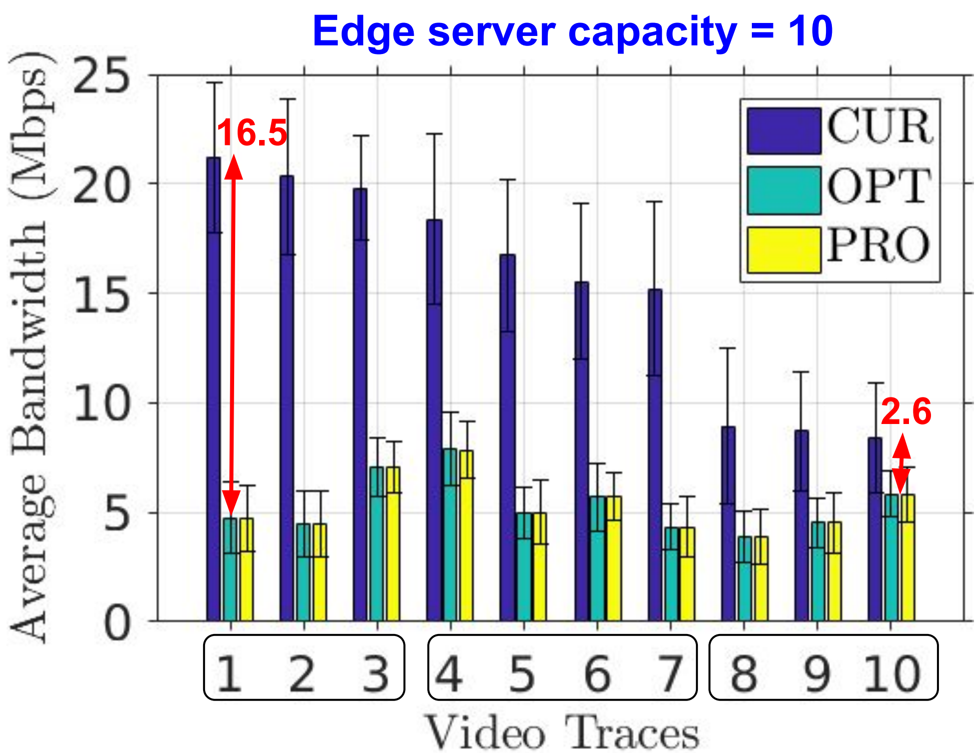
Evaluation

Setup

- Environment
  - Cloud server, Intel 60-cores workstation with 256 GB RAM
  - Edge server, Intel 40-cores workstation with 256 GB RAM
  - HMD client, Intel i7 CPU desktop with 16 GB RAM
- Tiling/Encoding/DASH
  - No. tiles = {5x5}
  - DASH segment length = {2} secs
  - Video bitrate outside/inside viewport = {1, 8} Mbps
  - FoV size = $\{100° \times 100°\}$
- Viewers
  - Randomly select 40 traces from the dataset (40/100)
- Baselines
  - Current streaming approach (**CUR**)
  - IBM CPLEX Solver (**OPT**)

Consumed Bandwidth

- We vary video sequences
- Saves bandwidth consumption from 31% to 78% Mbps
- Only stream the FoV saves lots of consumed bandwidth



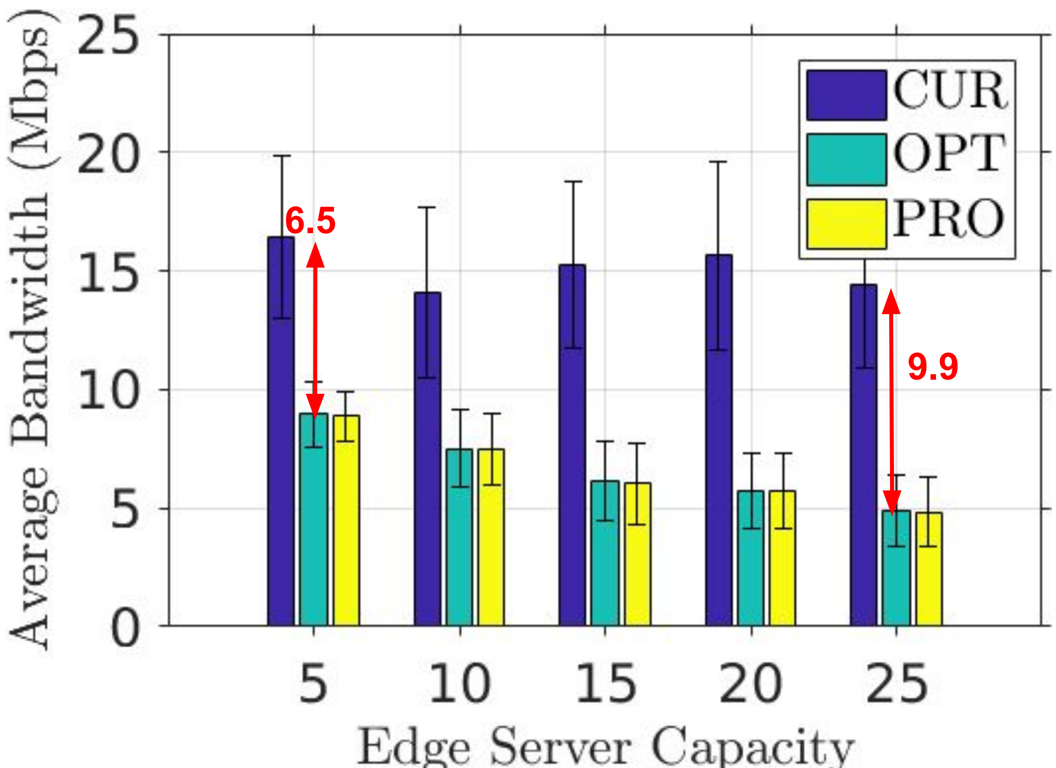
{1, 2, 3}: NI, fast-paced

{4, 5, 6, 7}: NI, slow-paced

{8, 9, 10}: CG, fast-paced

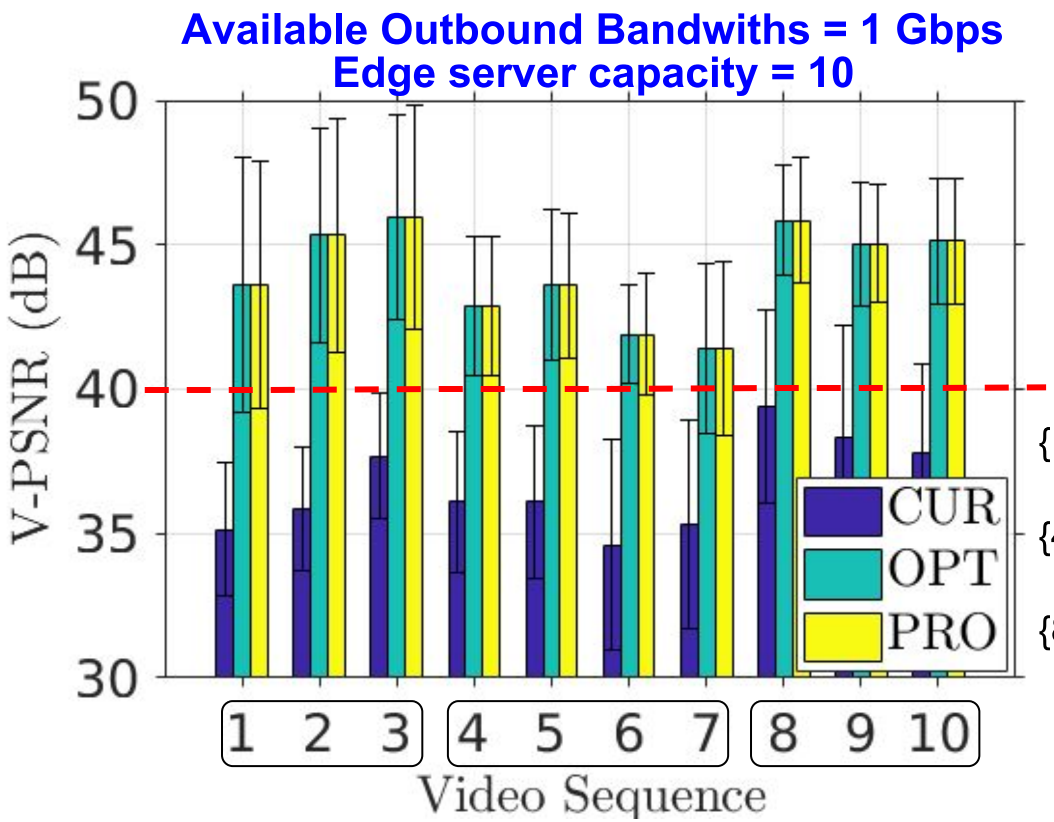
Consumed Bandwidth

- We vary edge capacities in {5, 10, 15, 20, 25}
- Save min/avg/max 35%/56%/62% bandwidth consumption
- Higher edge capacity, more consumed bandwidth we can save



Overall Video Quality (V-PSNR)

- We vary video sequences
- Constantly deliver high video quality (V-PSNR ≥ 40 dB)



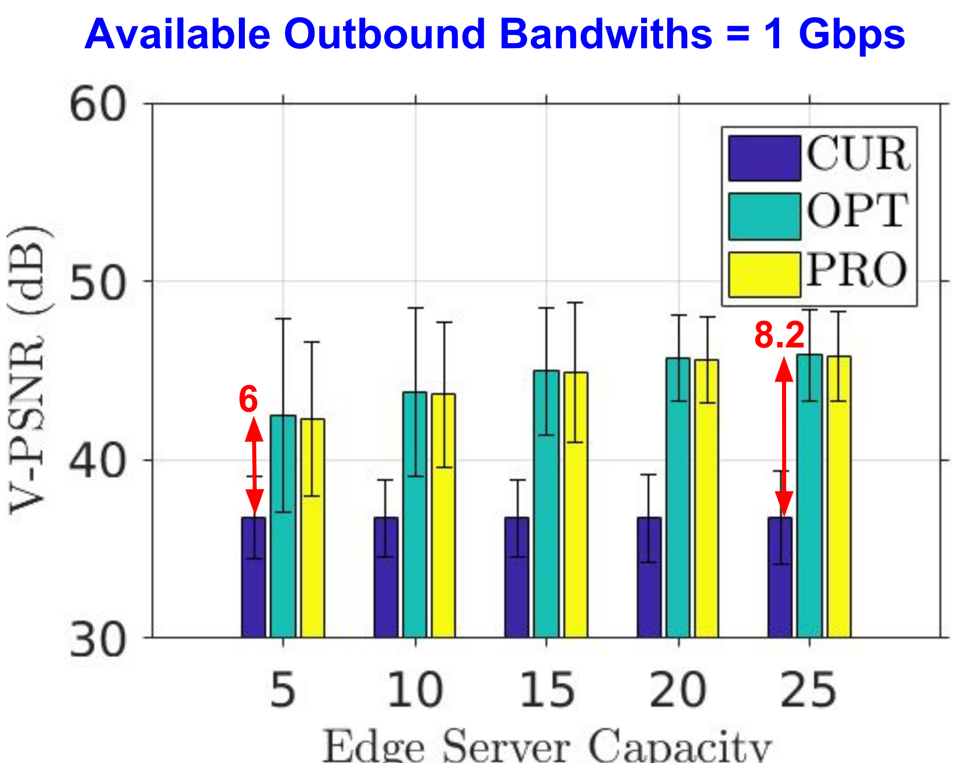
{1, 2, 3}: NI, fast-paced

{4, 5, 6, 7}: NI, slow-paced

{8, 9, 10}: CG, fast-paced

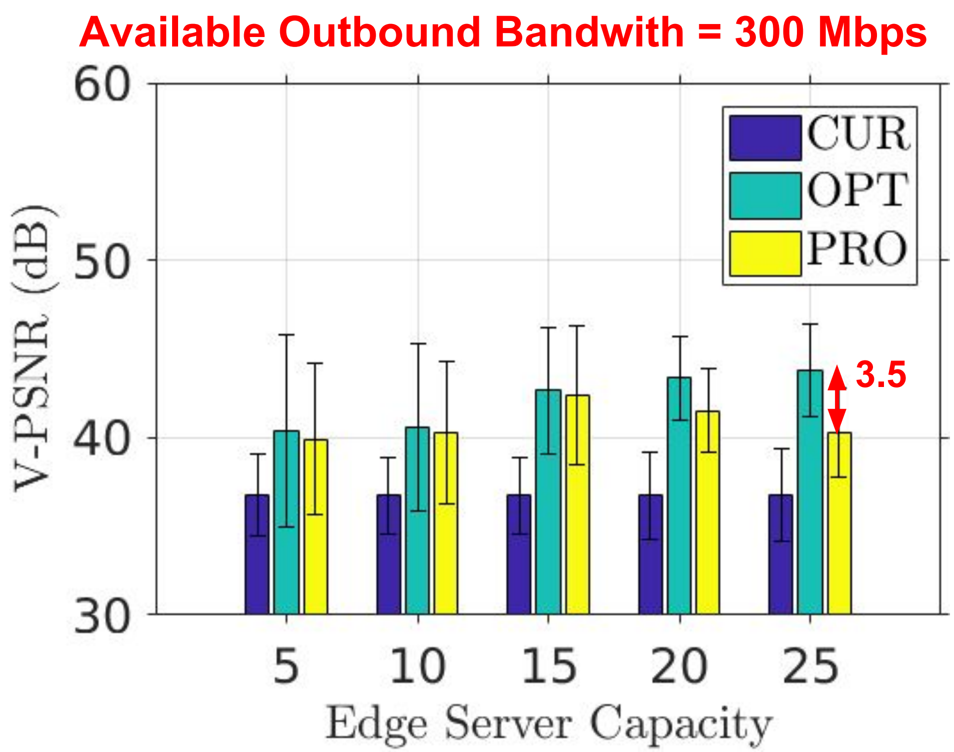
Overall Video Quality (V-PSNR)

- We vary different edge capacities in {5, 10, 15, 20, 25}
- Min/avg/max of video quality improvement is 6/7.4/8.2 dB
- Higher edge capacity, higher overall video quality we can get



If Available Bandwidth is Low...

- OPT delivers better video quality when edge capacity > 15
- OPT outperforms than PRO in video quality improvement by up to 3.5 dB



Proposed Runs Faster than OPT

- OPT suffers from exponential running time
  - It is not suitable to real-time systems
- PRO runs in polynomial time
  - It still outperfroms than CUR, and
  - produces good video quality improvement

Capacity	5	10	15	20	25
OPT	3.92 s	316.46 s	**531.62 s**	≥ 600 s	≥ 600 s
PRO	0.193 s	0.203 s	**0.229 s**	0.508 s	0.522s

Conclusion

Conclusion

- We propose an edge-assisted 360° video streaming system
- We design an algorithm for the optimal edge-assisted rendering to HMDs

- Compared to current streaming approach, our edge-assisted system:
  - saves bandwidth consumption by up to 62%
  - achieves higher video quality at the same bitrate
  - reduces weight of HMDs and offers better viewing experience

Future Work

- Instrumentation streaming system, not fully optimized
  - 40 CPUs, Intel(R) Xeon(R) CPU E5-2650 v3 @ 2.30GHz
  - Multithreading: 30 threads

	FPS	Time per Frame (s)
TR	49.84	0.02 s
VPR	0.89	1.12 s

- Leverage GPUs to fulfill real-time computing
- Model computing cost of VPR running on an edge server

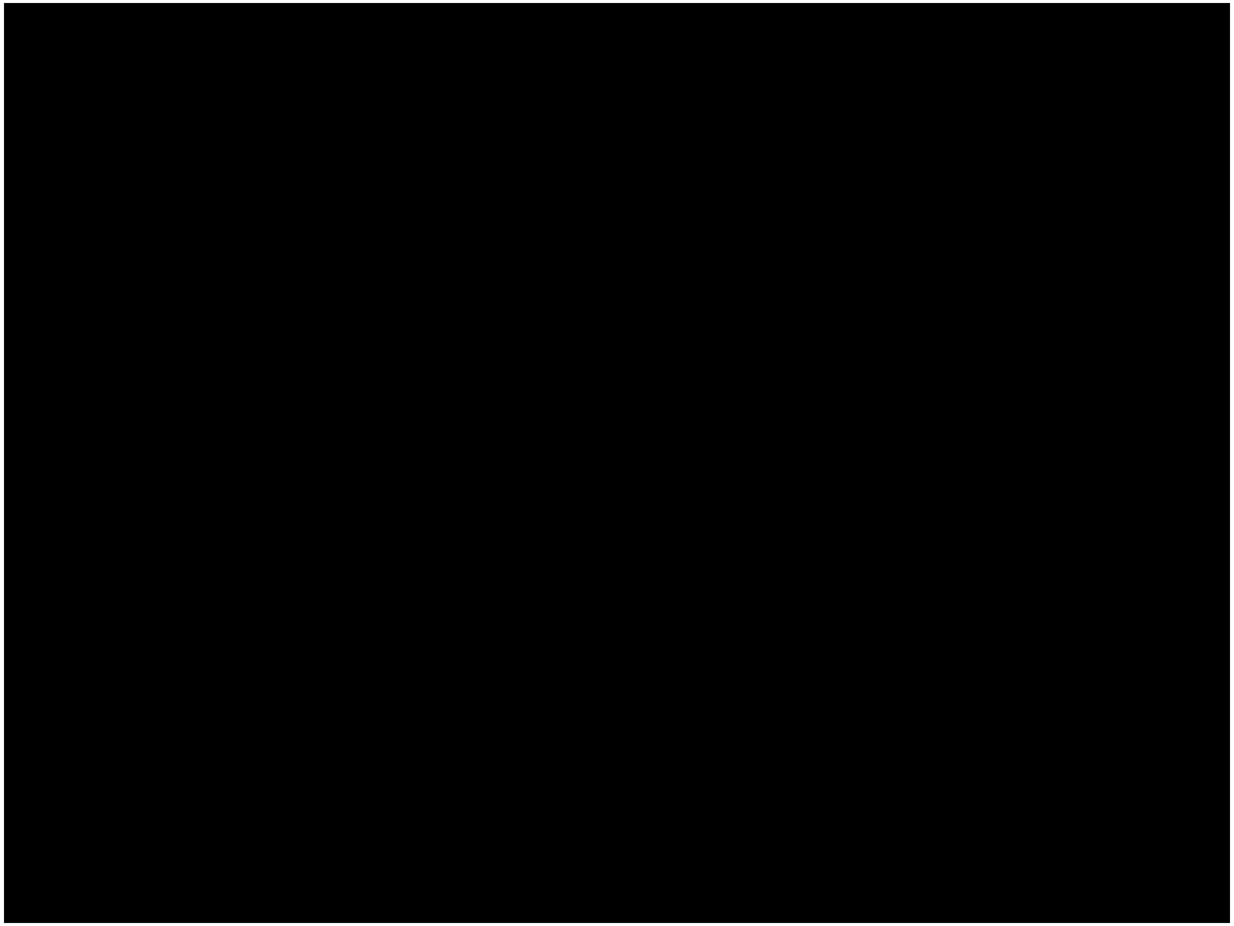
Research Highlight

- <u>**W. Lo**</u>, C. Fan, J. Lee, C. Huang, K. Chen, and C. Hsu, "360° Video Viewing Dataset in Head-Mounted Virtual Reality," in **Proc. of ACM on Multimedia Systems Conference** (**MMSys'17**), Dataset Track
- C. Fan, J. Lee, <u>**W. Lo**</u>, C. Huang, K. Chen, and C. Hsu, "Fixation Prediction for 360° Video Streaming in Head-Mounted Virtual Reality," in Proc. of **Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV'17)**
- <u>**W. Lo**</u>, C. Fan, S. Yen and C. Hsu, "Performance measurements of 360° video streaming to head-mounted displays over live 4G cellular networks," in **Proc. of Asia-Pacific Network Operations and Management Symposium (APNOMS'17)**
- C. Fan, <u>**W. Lo**</u>, Y. Pai, and C. Hsu, "A Survey on 360° Video Streaming: Acquisition, Transmission, and Display," in **ACM Computing Survey** (submitted)
- ISM'18 submission (based from this thesis)

Thanks for listening
Q & A

Backup Slides

Demo



Viewing Heatmap

- NI (fast-paced), NI (slow-paced), and CG (fast-paced)
- Leverage training set to draw viewing heatmap

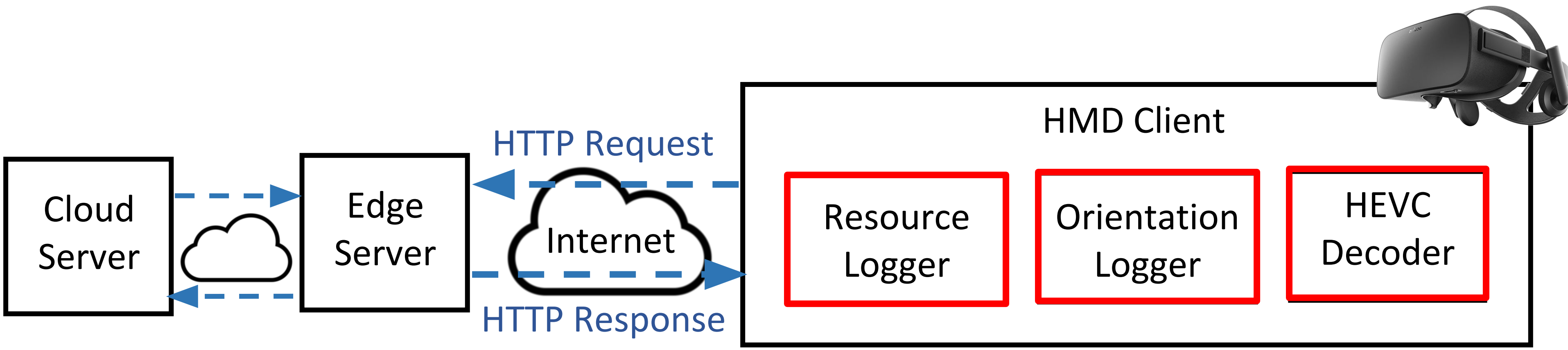
Heatmap

- coaster

Heatmap

- panel

HMD Client



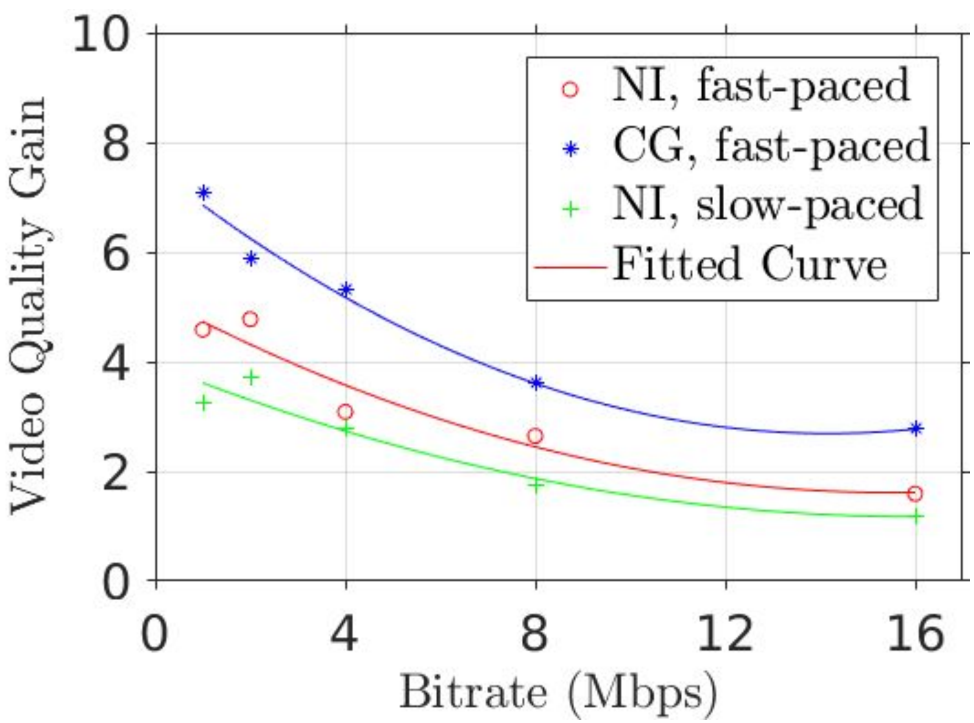
- HMD Client
  - Resource Logger
  - Orientation Logger
  - HEVC Deocder

Symbol Table

Symbol	Description
N	Set of all HMD clients
n	Index of a HMD client
B	Outbound bandwidth of an edge server
T	Number of tiles
t	Index of a tile
S	Video segment length in second
f_n^w	The width of tiles of HMD client n's viewport
f_n^h	The height of tiles of HMD client n's viewport
$\mathbf{V}_n$	Set of tiles overlapped with the viewer's FoV
b_h, b_l	High/Low encoding bitrate
O_n	Viewer's orientation collected from HMD client n
α_n	Consumed bandwidth of HMD client n for Viewport Rendering (VPR)
β_n	Consumed bandwidth of HMD client n for Tile Rewriting (TR)
q_n	Video quality of HMD client n for TR
q'_n	Video quality of HMD client n for VPR
E	Maximum number of HMD clients that an edge server can serve
x_n	Decision variable of the problem formulation

Lemma 1: Optimal Quality Improvement

- Video quality improvement $q_n' - q_n$ is monotonically decreasing



Lemma 2: Runs in Polynomial Time

O(E + NlogN)

Algorithm 1 Mode Selector.

// We first initialize variables
for each n in N **do**
  $\mathbf{x[n]} \leftarrow 0$; $\mathbf{Qual[n]} \leftarrow q'_n - q_n$; $\mathbf{Band[n]} \leftarrow \beta_n$; $\mathbf{Rati[n]} \leftarrow (q'_n - q_n)/(\beta_n - \alpha_n)$ — O(N)
sort Qual[N], Band[N], Rati[N] in desc. order — O(NlogN)
while $E > 0$ **do**
  pop an HMD client n with the maximal **Qual[n]**
  $\mathbf{x[n]} \leftarrow 1$
  $\mathbf{Band[n]} \leftarrow \alpha[n]$
  $E = E - 1$
— O(E)
if sum(Band[N]) $\leq B$ **then**
  return **x[N]**
— O(N)

(lines 4–11: O(E + NlogN))

Initialize E, $x[N]$, and **Band[N]**
while $E > 0$ **do**
  pop an HMD client n with the maximal **Rati[n]**
  $\mathbf{x[n]} \leftarrow 1$
  $\mathbf{Band[n]} \leftarrow \alpha[n]$
  $E = E - 1$
if sum(Band[N]) $\leq B$ **then**
  return **x[N]**

(O(E + NlogN))

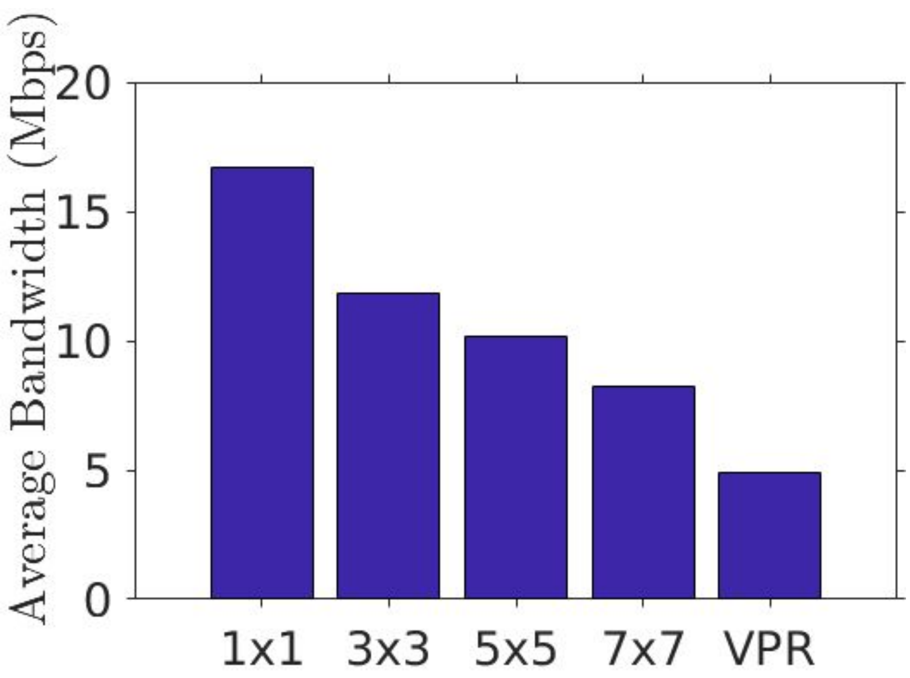
Initialize E. $x[N]$, and **Band[N]**
while $E > 0$ **do**
  pop an HMD client n with the maximal **Band[n]**
  $\mathbf{x[n]} \leftarrow 1$
  $\mathbf{Band[n]} \leftarrow \alpha[\mathbf{n}]$
  $E = E - 1$
if sum(Band[N]) $\leq B$ **then**
  return **x[N]**

(O(E + NlogN))

return no feasible solution

Observation: Consumed Bandwidth

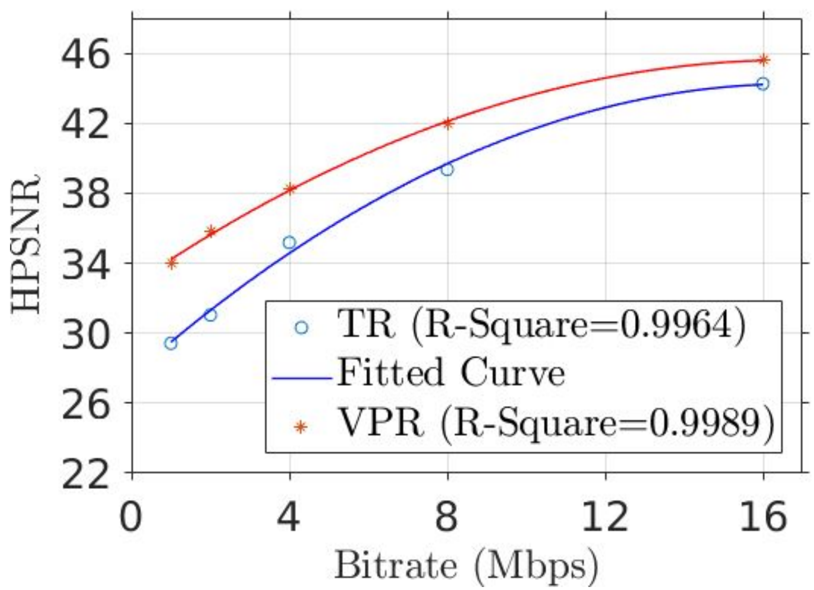
- Tile Rewriting = {3x3, 5x5, 7x7}
- Viewport Rendering (VPR)
- Encoding bitrate: 8 Mbps / 1Mbps



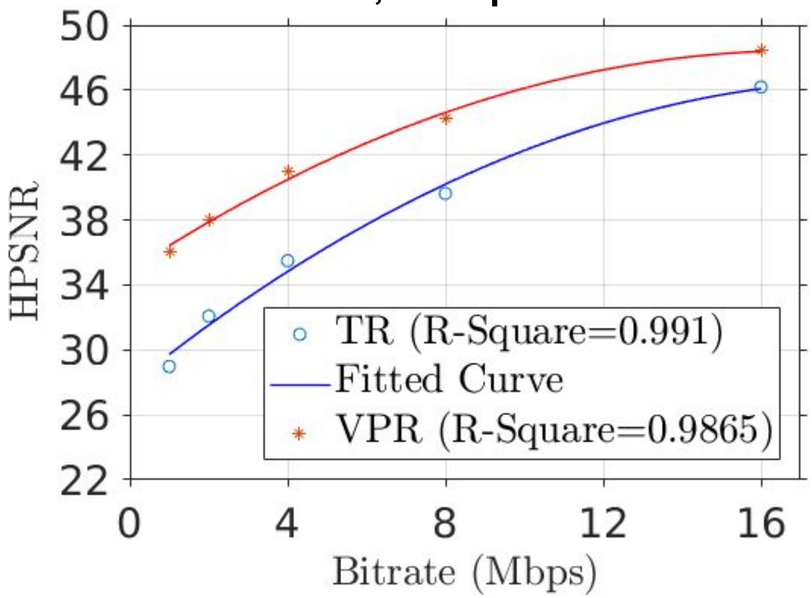
Observation: Video Qulaity Gain

- Peak signal-to-noise ratio (PSNR) wighted by viewing heatmap (HPSNR)

NI, fast-paced

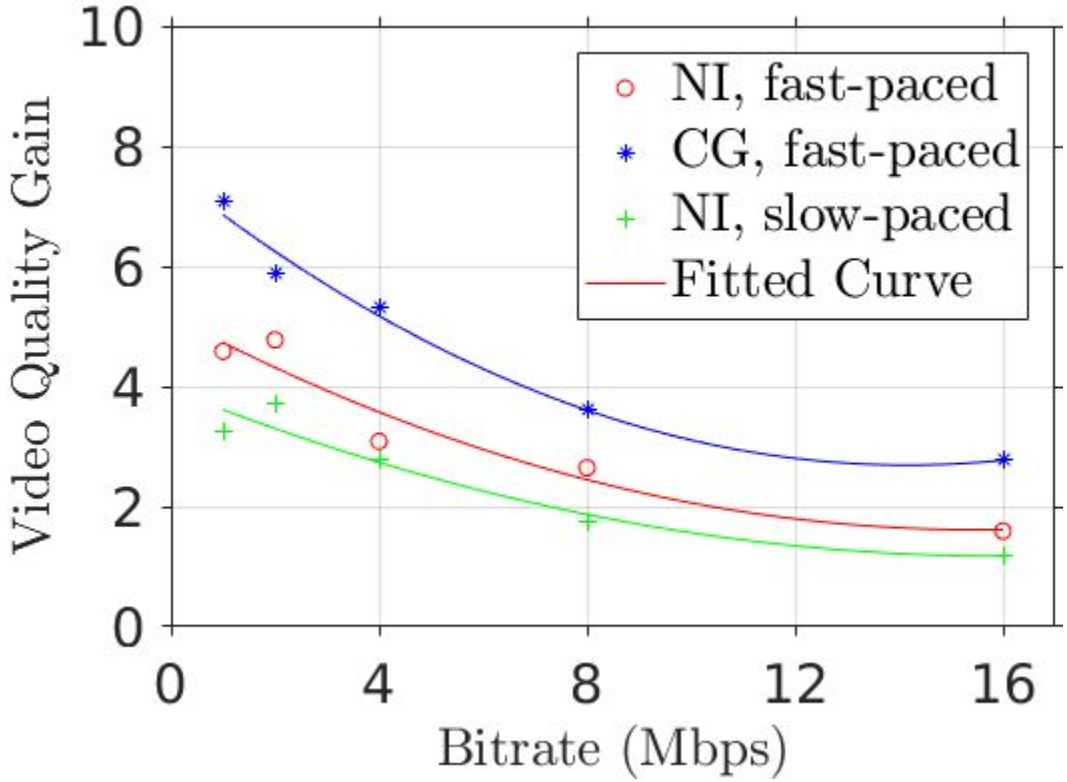


NI, slow-paced



Video Quality Gain

- Differentiate video quality of TR and VPR
- VPR gets better video quality



Rendering

- Computer Graphics
  - a process of generating a 2D/3D image

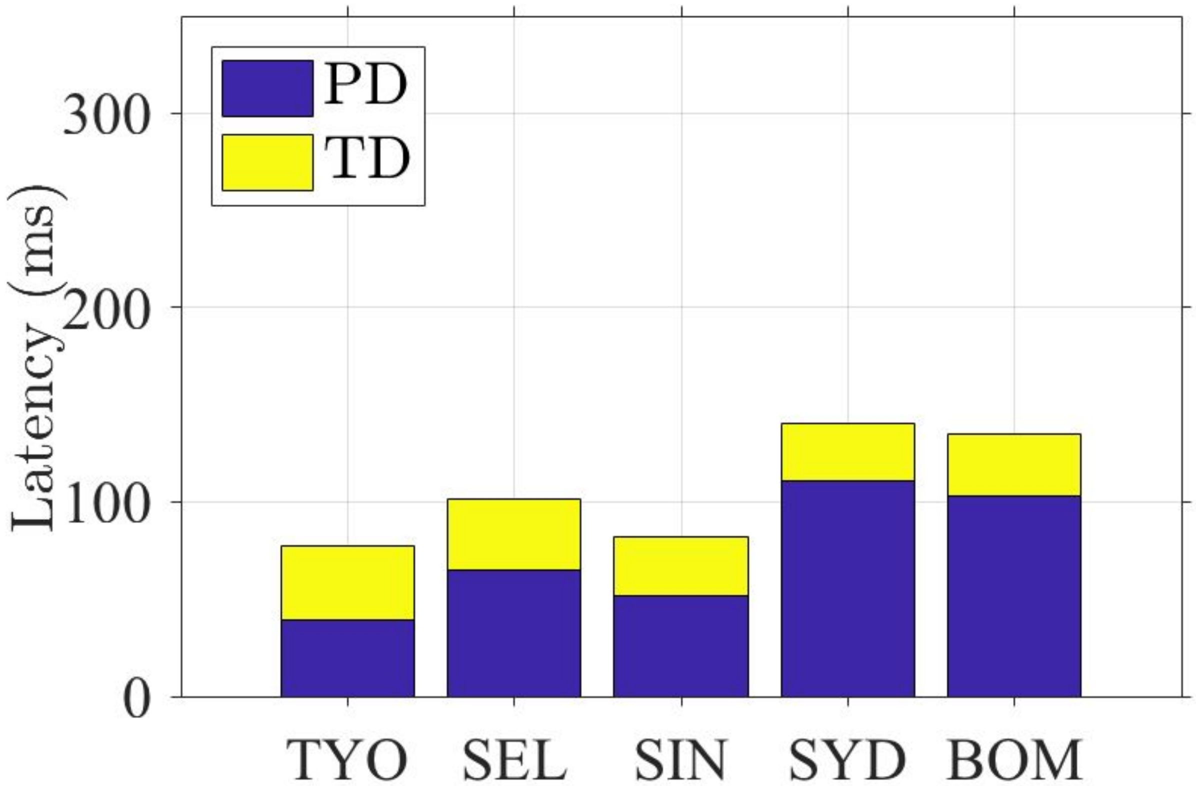
Rendering

- Computer Graphics
  - a process of generating a 2D/3D image
- Augmented/Virtual Reality
  - a process of generating an user's viewport

Cloud/Edge Latency & Bandwidth

- AWS clouds
  - US East/N. Virginia
  - US East/N. California
  - Canada/Montreal
  - EU/Frankfurt
  - EU/London
- AWS edges
  - Asia/Seoul
  - Asia/Singapore
  - Asia/Sydney
  - Asia/Tokyo
  - Asia/Mumbai

Latency



Latency