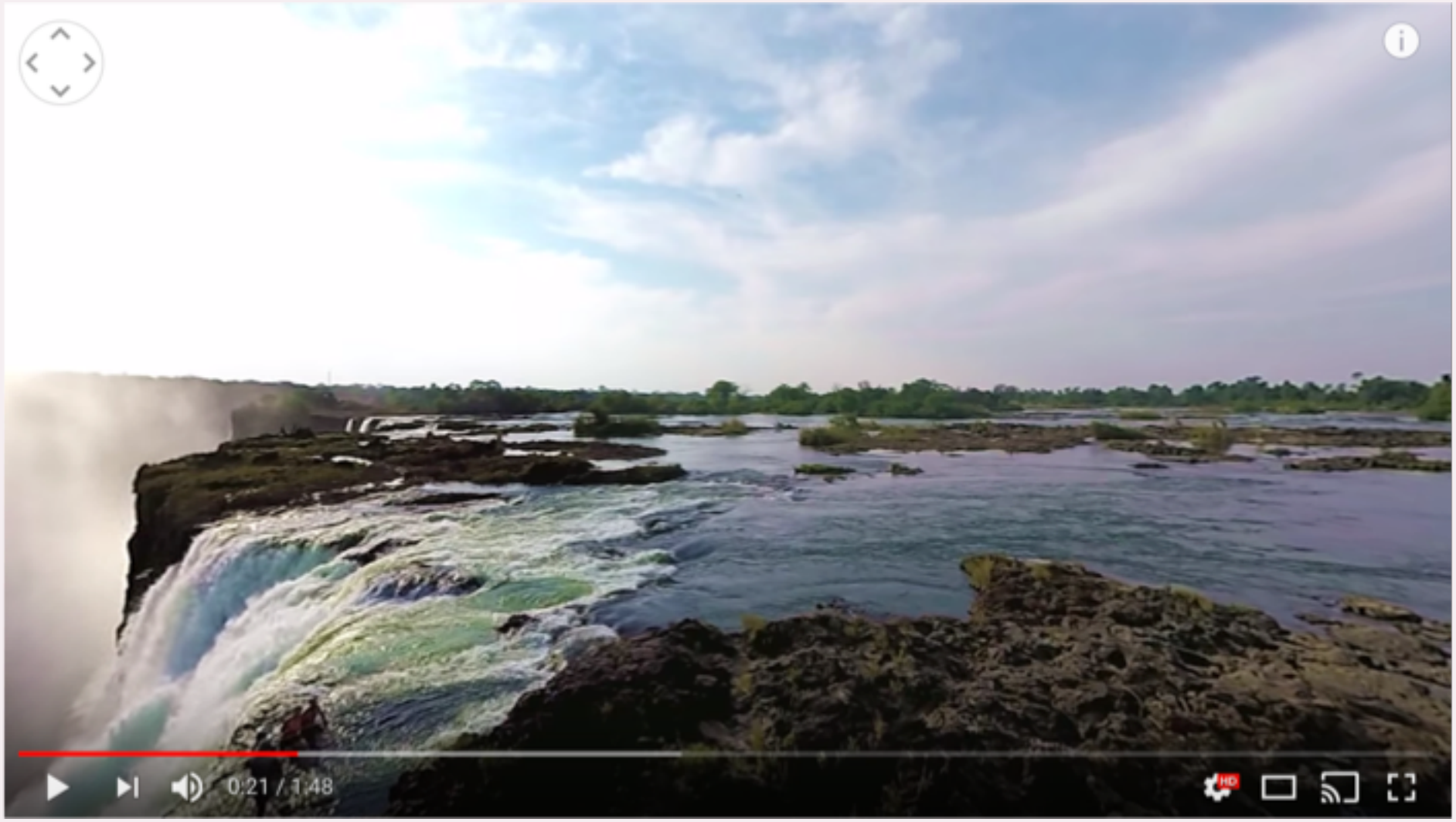
360 Degree Video Streaming

Yao Wang
Dept. of Electrical and Computer Engineering
Tandon School of Engineering
New York University
http://vision.poly.edu

360 Video Streaming

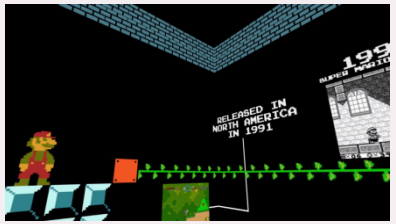


- https://www.youtube.com/watch?v=WsMjBMxpUTc

Applications of 360 Video Streaming

Virtual Tour

Gaming



Sports

Show

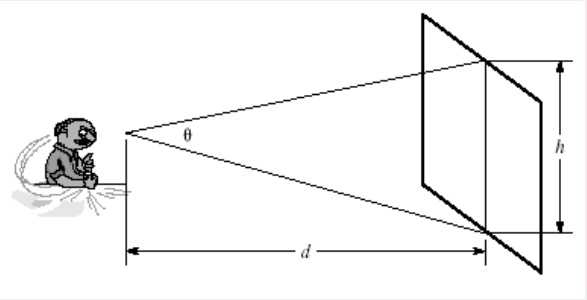
Entertainment

Training

- Interactive streaming
  - Video conferencing
  - Gaming
- Live Streaming
  - Live concert / sports
  - Training/education (surgery, flight, …)
- On-demand streaming
  - Entertainment
  - Training /education
  - Tourism
  - Youtube, Facebook, …

What resolution is needed?

- Perceived resolution depends on view angle span!
- Retina resolution: up to 60 pixel per degree (PPD)
- TV/computer/phone display is designed to cover about 36°
  - HD video 4096x2048: 4096/36~100 PPD ☺
  - Same format for 360°x180° ~ 11 PPD ☹



- HMD covers 90° to 120° FoV, with 60 PPD =>5400^2~7200^2 pels
- 360 video with 60 PPD => 21600*10800 pixels
- Also needs high frame rate (120 fps) & color depth (12 bits)!
- Stereo display (3D) further doubles!
- What bandwidth are we talking about?

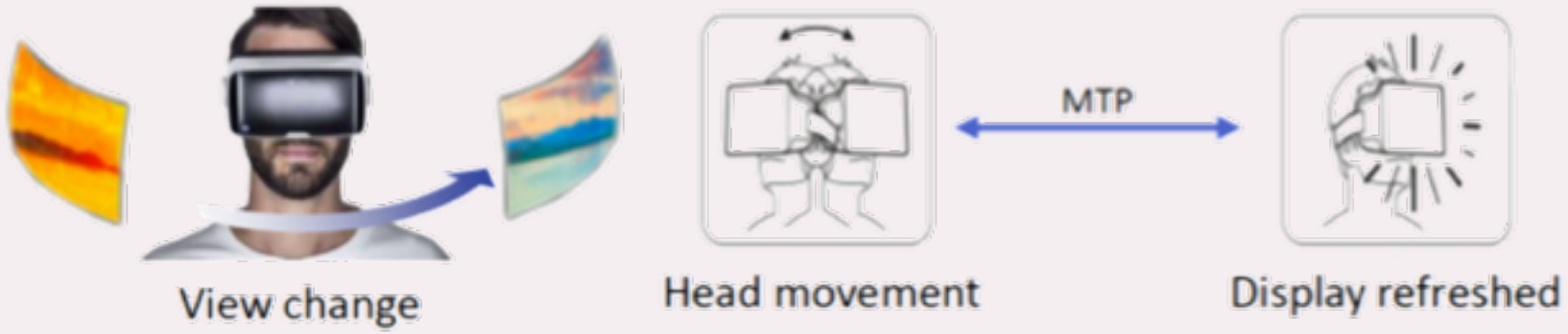
Resolution and Network Requirement

http://www-file.huawei.com/~/media/CORPORATE/PDF/white%20paper/whitepaper-on-the-vr-oriented-bearer-network-requirement-en.pdf

	Entry-Level	Advanced	Ultimate
Resolution	8K 2D (7680*3840)	12K 2D (11520*5760)	24K 3D (23040*11520)
HMD FoV (resolution)	90x90 (1920x1920)	120x120 (3840x3840)	120x120 (7680x7680)
PPD	21	32	64
Color representation	8 bit, 4:2:0	10, 4:2:0	12, 4:2:0
Frame Rate	30	60	120
Compression Ratio (Estimated)	165:1 (H264)	215:1 (HEVC/VP9)	350:1 (H.266)
Compressed Bitrate	64 Mbps	279 Mbps	3.29 Gbps
Bandwidth for smooth play	100 Mbps	418 Mbps	4.93 Gbps (Full 360) 1 Gbps (FoV only) 2.35 (FOV interactive)
Network Latency	30 ms	20 ms	10 ms

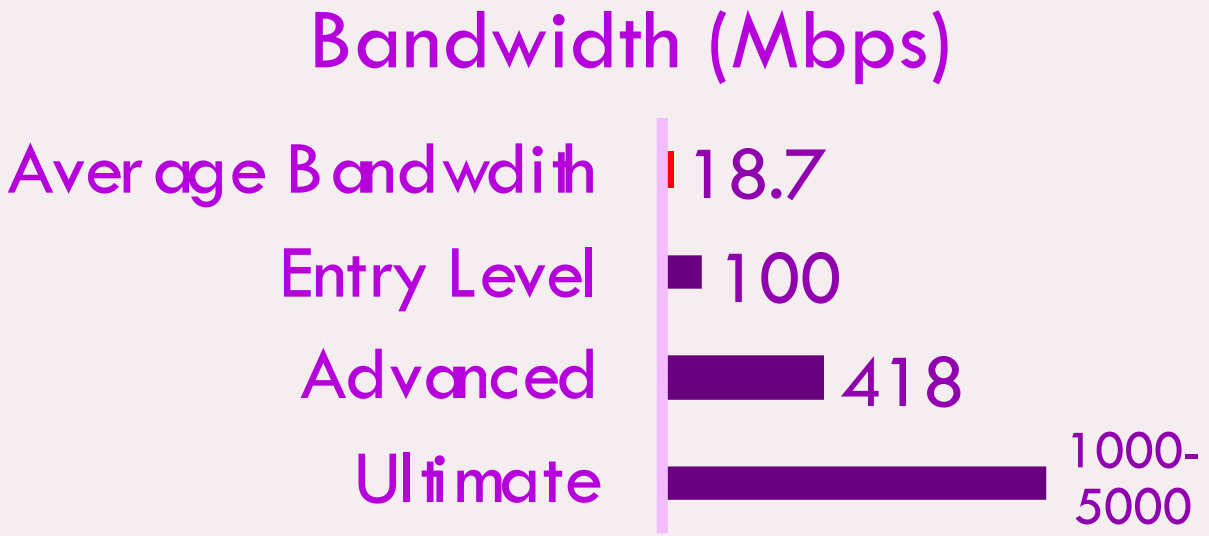
Latency Requirement

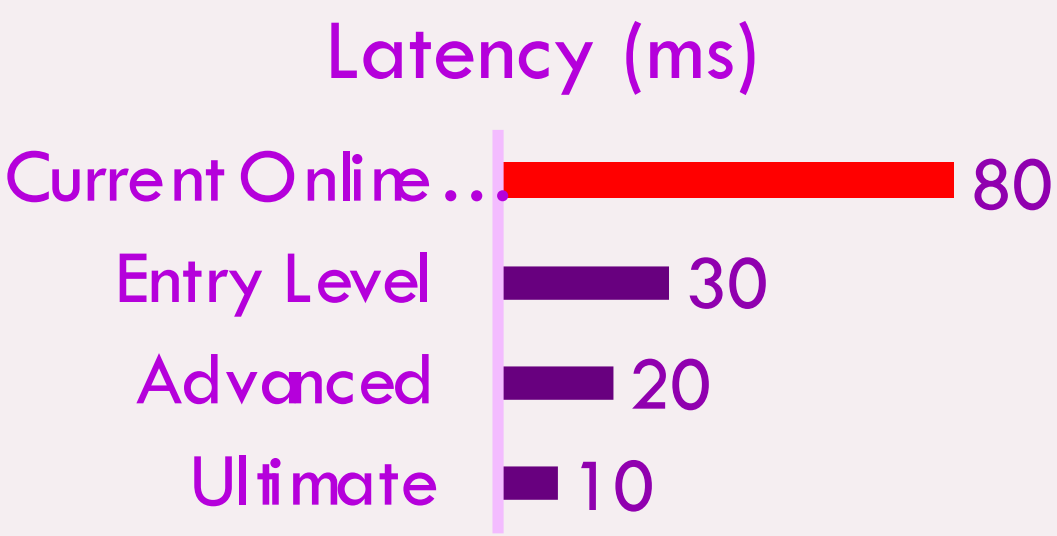
360 source => FOV based on the detected head position => FoV display



- Motion to Photon (MTP) delay should not be greater than 20ms!
  - Rendering done at a nearby computer or on HMD: <=10ms
  - Transmission delay (if rendering and deliver FoV remotely) <=10ms

Network Challenges





- Current network bandwidth and delay from https://www.akamai.com/us/en/about/our-thinking/state-of-the-internet-report/global-state-of-the-internet-connectivity-reports.jsp

DASH for On-Demand Streaming of 2D Video

- What happens after you click on a video on Youtube?
- Video divided into short segments (e.g. 4 sec.), each segment precoded into multiple chunks with different bit rates and saved on a server
- Initial buffering up to 20 sec. of video (while you watch commercials ☺)
- The client request next chunk based on the estimated network throughput and target buffer length
- Prefetching absorbs the bandwidth variation
  - Can stream the video at about average bandwidth even when the actual throughput fluctuates up and down
  - Stalls if a chunk arrives later than its display time (rebuffering ☹)

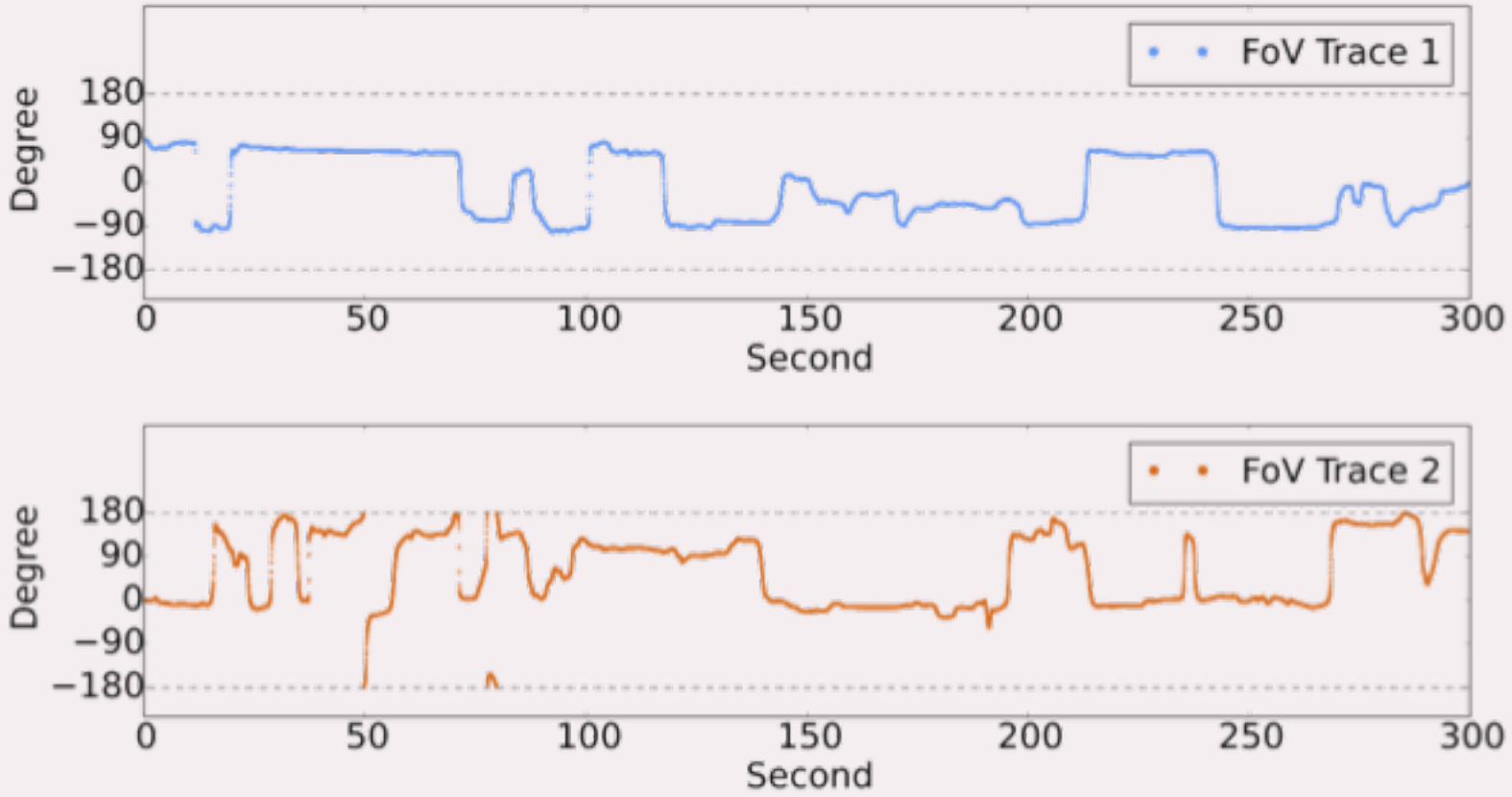
How to Stream 360 Video?

- Send entire 360 view span
  - A user only watches a small portion (Field of View) at any time! → Waste of bandwidth
  - Low quality under limited bandwidth
- Send only the predicted FoV (and possibly low quality for other areas)
  - FoV prediction over long time horizon (more than 5 sec) is hard!
  - With short pre-fetching buffer, requested video chunks may not arrive in time → video freezing
  - Predicted FoV can be wrong → missing part or all of the desired FoV
- How to take advantage of prefetching AND FoV prediction?

Typical FoV Dynamics

- Steady
- Easy to predict except during sudden transitions

- Fluctuant
- Hard to predict

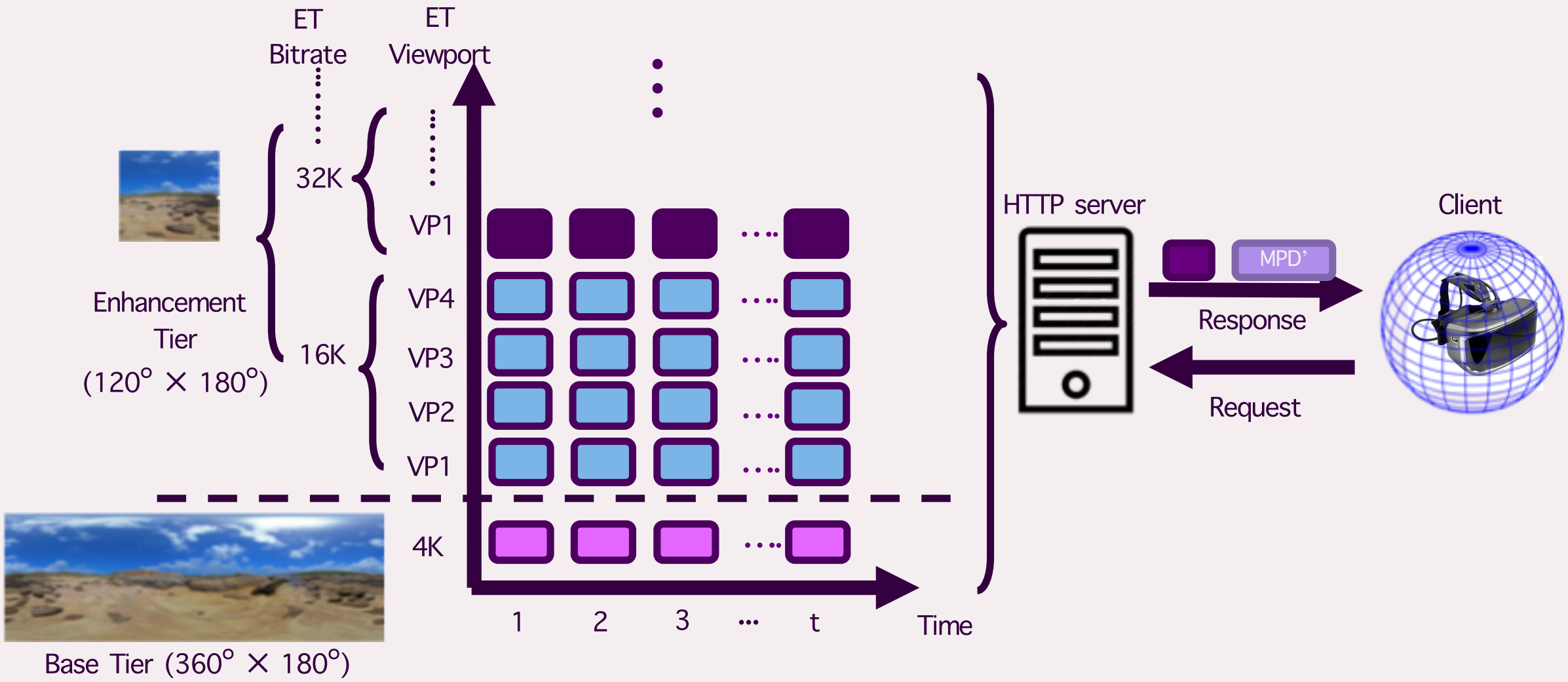


FoV Trace from: Chenglei Wu, Zhihao Tan, Zhi Wang, and Shiqiang Yang, "A Dataset for Exploring User Behaviors in VR Spherical Video Streaming," In Proc. of the 8th ACM on Multimedia Systems Conference (MMSys'17). 2017.

Our work: Two-Tier 360V Streaming

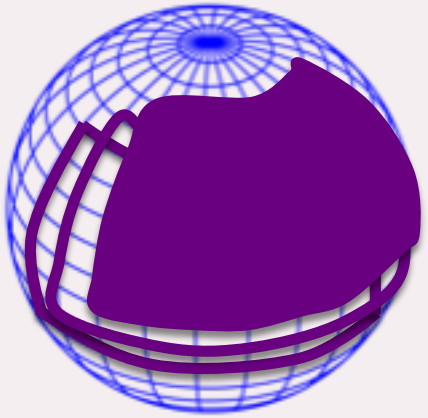
- Two-tier video encoding:
  - Base-tier (BT) chunks cover entire 360° scene in low quality
  - Enhancement-tier (ET) chunks cover different viewports at multiple rates
- Two-tier video streaming
  - Download BT chunks with long prefetching buffer (10-20s)
  - Download ET chunks based on predicted FoV with short prefetching buffer
- Two-tier video rendering
  - If buffered ET chunks match user actual FoV, render high quality video in FoV
  - Otherwise, render low quality video in FoV based on buffered BT chunks
- Base tier provides robustness to both network dynamics and view dynamics

Two-Tier Streaming System



ET View Partition and Coding

- Tiling/Striping: encode non-overlapping tiles/stripes
  - No storage redundancy ☺
  - Low coding/b.w. efficiency ☹
- Encode overlapped viewports
  - high coding/b.w. efficiency ☺
  - high storage redundancy ☹
- Layered vs. non-layered coding between ET and BT
  - Coding efficiency vs. complexity



Multi-Dimensional Problems

- Rate allocation:
  - How to set rates for base tier and enhancement tier?
- Video coding:
  - Layered or non-layered coding? Tile or Viewport coding?
- Streaming decisions:
  - What should be the target buffer length for BT and ET?
  - Download ET or BT chunks? (instant quality vs. long-term robustness)
  - Which BT/ET chunks? (Rate and viewport for ET)
- Multi-objective optimizations:
  - Rendered video quality & continuity, responsiveness to network & FoV dynamics

Rendered Video Quality

Assuming the base tier is always delivered before display deadline:

$$Q(R_b; \alpha, \gamma, R_t) = \alpha\gamma Q_e(\tilde{R}_e) + (1 - \alpha\gamma) Q_b(\tilde{R}_b)$$

ET chunk delivered and FoV correct With BT chunk only

- α: FoV hit rate
  - Average overlapping ratio between the requested viewport and actual FOV
- γ: Chunk delivery rate
  - Likelihood that a requested chunk is delivered before its display deadline
- Both depend on target ET buffer length

- $R_b + R_e = R_t = \eta\overline{BW}$
- $\tilde{R}_b = \frac{R_b}{A_b}$
- $\tilde{R}_e = \frac{R_b}{A_b} + \frac{R_t - R_b}{A_e}$

A_b: BT view coverage area

A_e: ET view coverage area

Ex: A_b =360x180, A_e =120x120, A_e/A_b=2/9

Rate Allocation Optimization

$$Q(R_b;\alpha,\gamma,R_t) = \alpha\gamma Q_e(\tilde{R}_e) + (1-\alpha\gamma)Q_b(\tilde{R}_b) = \alpha\gamma Q_e\left(\frac{R_b}{A_b} + \frac{R_t - R_b}{A_e}\right) + (1-\alpha\gamma)Q_b\left(\frac{R_b}{A_b}\right)$$

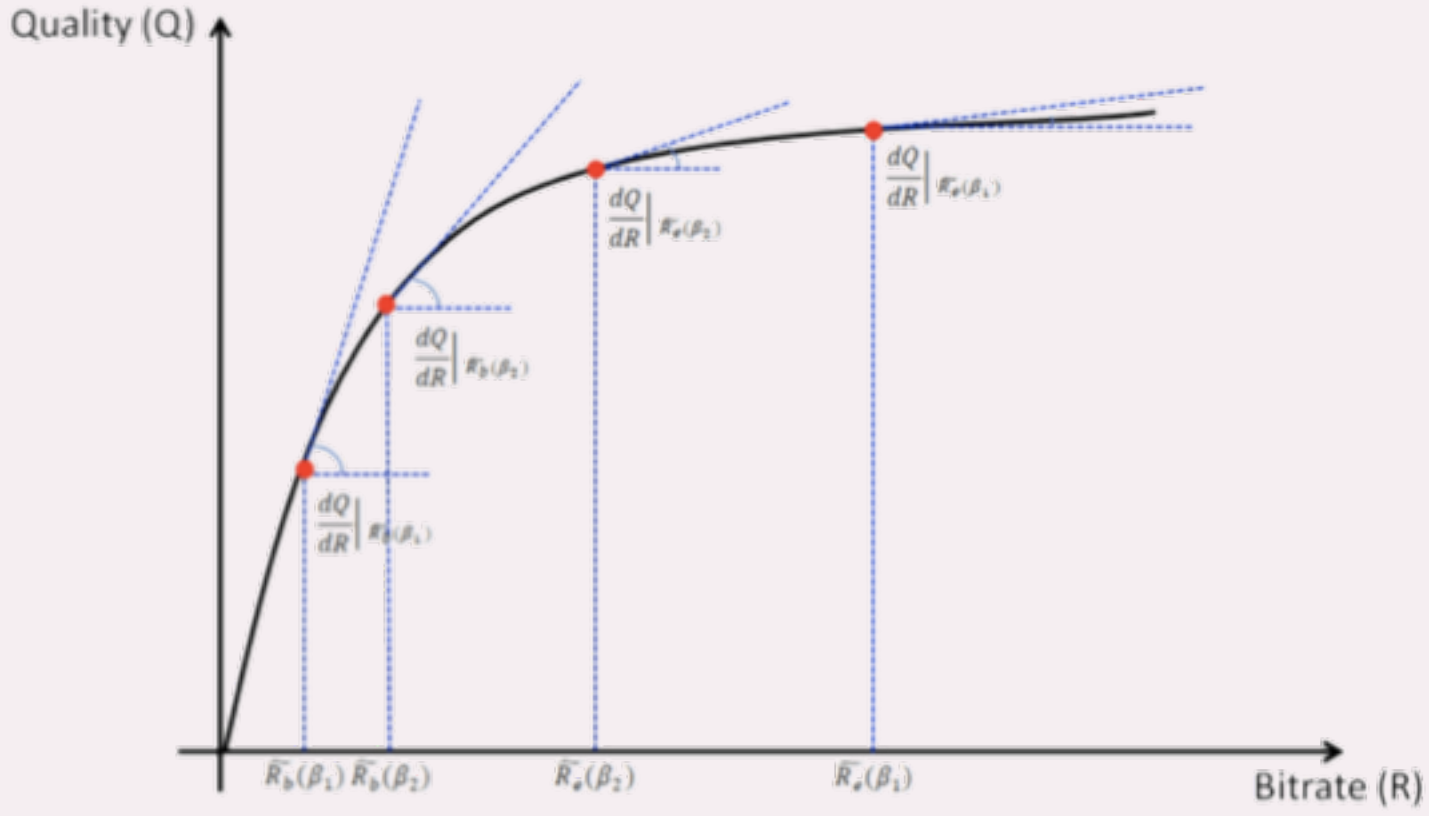
Setting $\frac{\partial Q}{\partial R_b} = 0 \rightarrow$

$$\left.\frac{\partial Q_e}{\partial R}\right|_{R_e^*} = \left(\frac{1-\alpha\gamma}{\alpha\gamma}\right)\frac{A_e}{A_b - A_e}\left.\frac{\partial Q_b}{\partial R}\right|_{R_b^*}$$

$$= \beta \left.\frac{\partial Q_b}{\partial R}\right|_{R_b^*}$$

Assuming $Q(R) = a + b\cdot\log(R)$

$$R_b^* = \beta R_e^*, R_b + R_e = R_t$$



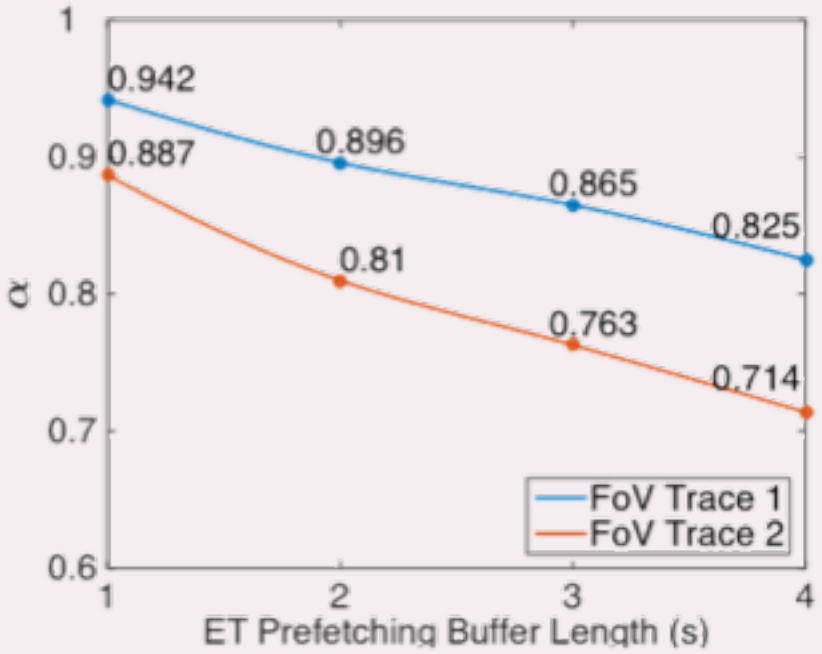
Ex 1: $\alpha\gamma$=0.9, $\frac{A_e}{A_b} = \frac{2}{9}$, $\beta = 0.03$

Ex 2: $\alpha\gamma$=0.7, $\frac{A_e}{A_b} = \frac{2}{9}$, $\beta = 0.12$

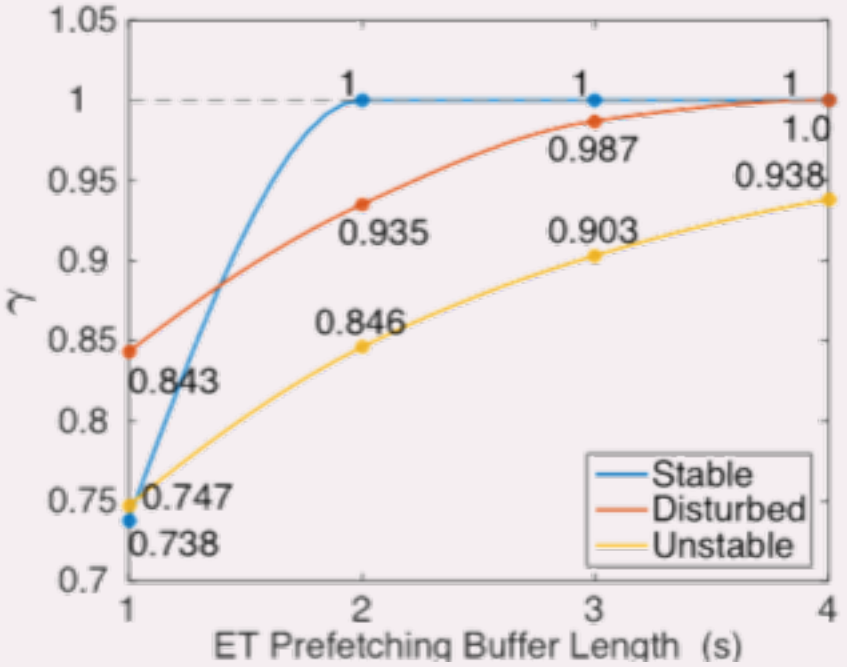
ET Buffer Length Optimization

$$Q(R_b;\alpha,\gamma,R_t)=\alpha\gamma Q_e(\tilde{R}_e)+(1-\alpha\gamma)Q_b(\tilde{R}_b)$$

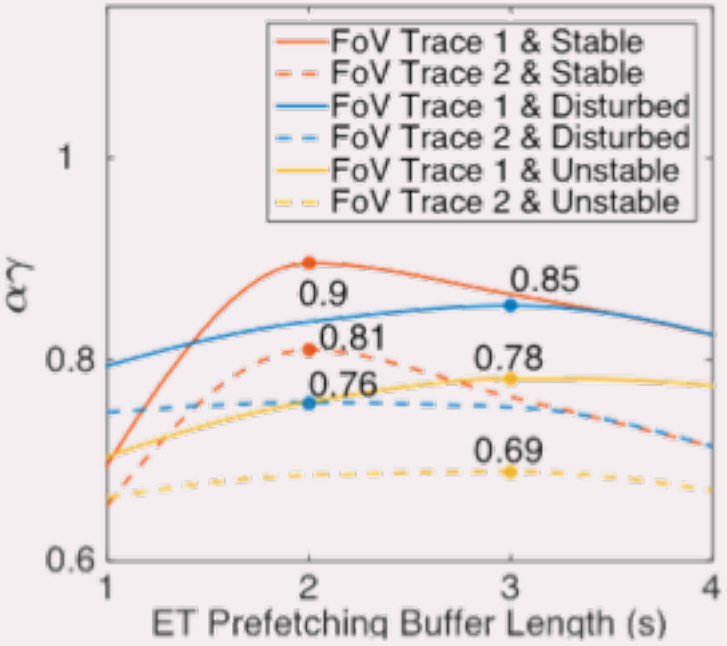
Given rate allocation, $\max Q \rightarrow \max\{\alpha\gamma\}$



FoV hit rate $\alpha \downarrow$ with buffer length



Chunk delivery rate $\gamma \uparrow$ with buffer length



$\alpha\gamma \rightarrow max$
at medium buffer length

Preliminary results

- Benchmark System 1 (BS1): Sending entire 360 view span
- Benchmark System 2 (BS2): Sending predicted FoV only
- Using bandwidth trace collected over a 3.5G HSPA cellular network
- Using FoV traces captured using Google Cardboard with a smart phone
- Using a PI controller for selecting ET rate to maintain the target ET buffer length
- Using a simple linear predictor for FoV
- No optimization of rate allocation

Fanyi Duanmu, Eymen Kurdoglu, S. Amir Hosseini, Yong Liu and Yao Wang, Prioritized Buffer Control in Two-tier 360 Video Streaming, ACM SigComm Workshop on VR/AR Network, 2017.

Preliminary Results

Network Trace Solution	1 *BS1*	1 *BS2*	1 TTS	2 *BS1*	2 *BS2*	2 TTS
RollerCoaster	2.8/12%	10.8/27%	7.9/6%	5.9/4%	27.0/10%	21.1/4%
Amsterdam	2.8/12%	10.5/25%	7.7/6%	5.9/4%	27.3/10%	22.3/3%

Video rendering rate (VRR, Mbps)

Freeze Ratio

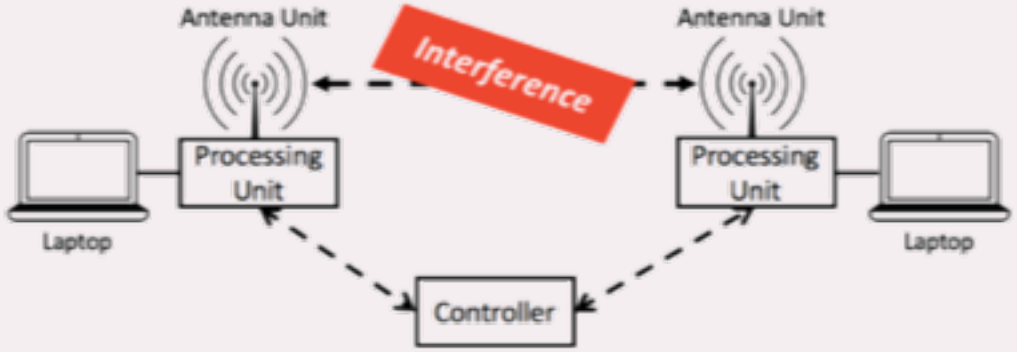
Network Trace Solution	3 *BS1*	3 *BS2*	3 TTS	4 *BS1*	4 *BS2*	4 TTS
RollerCoaster	9.0/1%	40.2/2%	36.6/0%	23.0/0%	93.1/8%	108.0/0%
Amsterdam	9.0/1%	39.3/2%	36.1/0%	23.0/0%	91.5/7%	106.3/0%

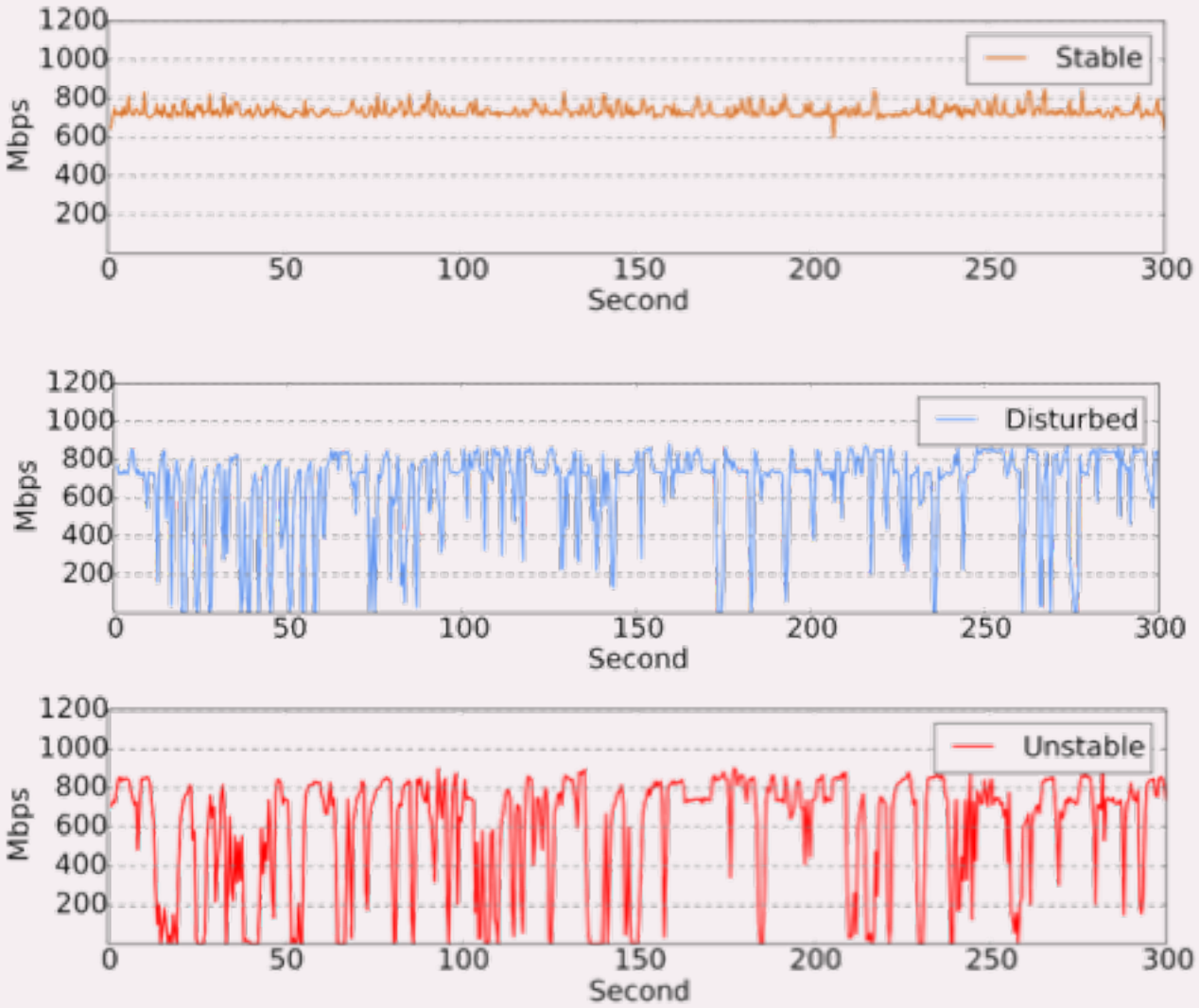
- Two-tier system achieves higher VRR and similar freeze ratio as BS1
- Two-tier system achieves comparable VRR but lower freeze ratio than BS2

Simulations using 5G Network Testbed

- Properties of 5G
  - High speed up to be 10 Gbps
  - Low latency down to be 1ms
  - High volatility, on-off

- Testbed: WiGig (802.11ad, multi-Gbps, 60Ghz)

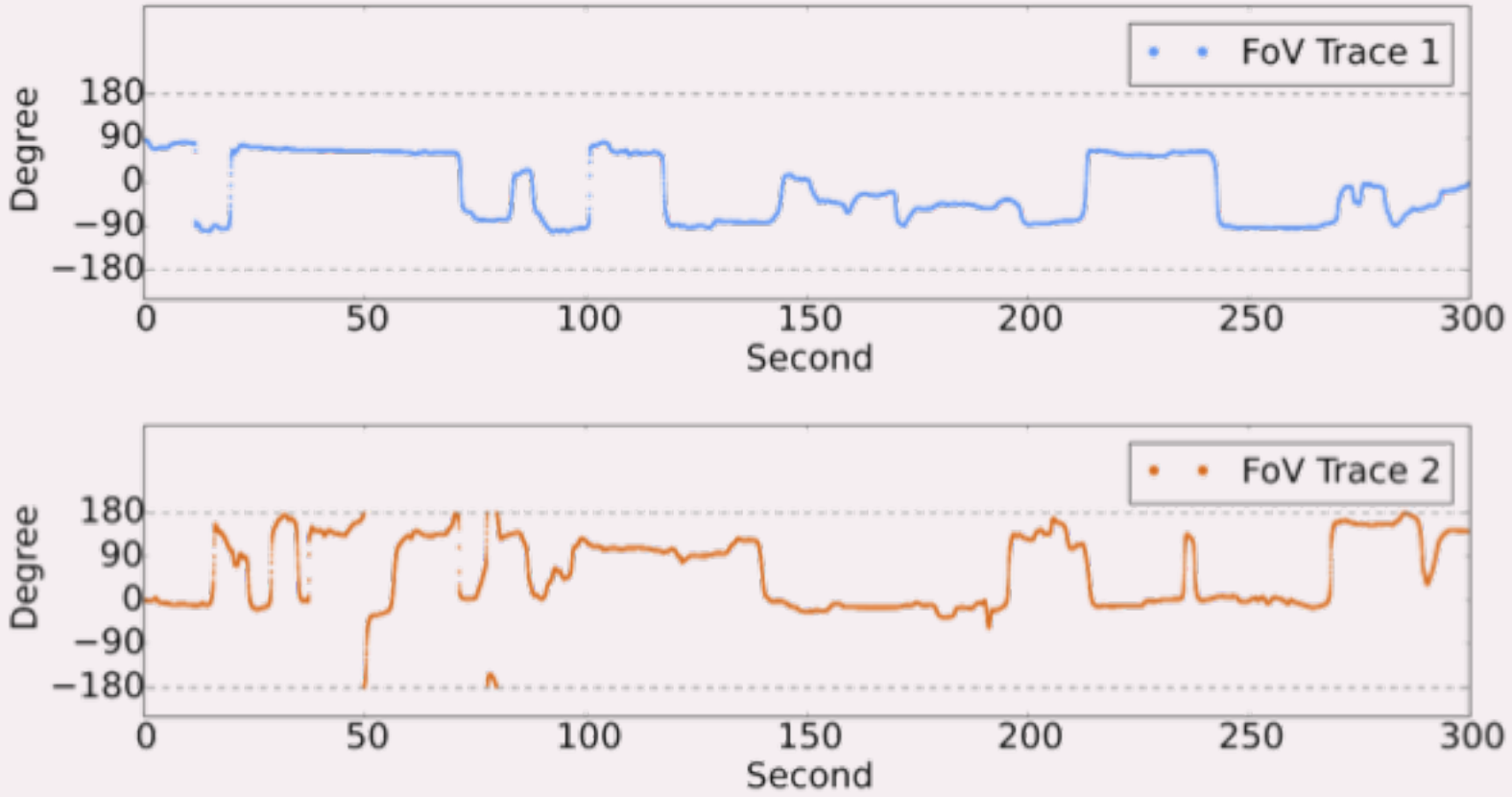




360° Video FoV Traces

- Steady
- High Prediction Accuracy

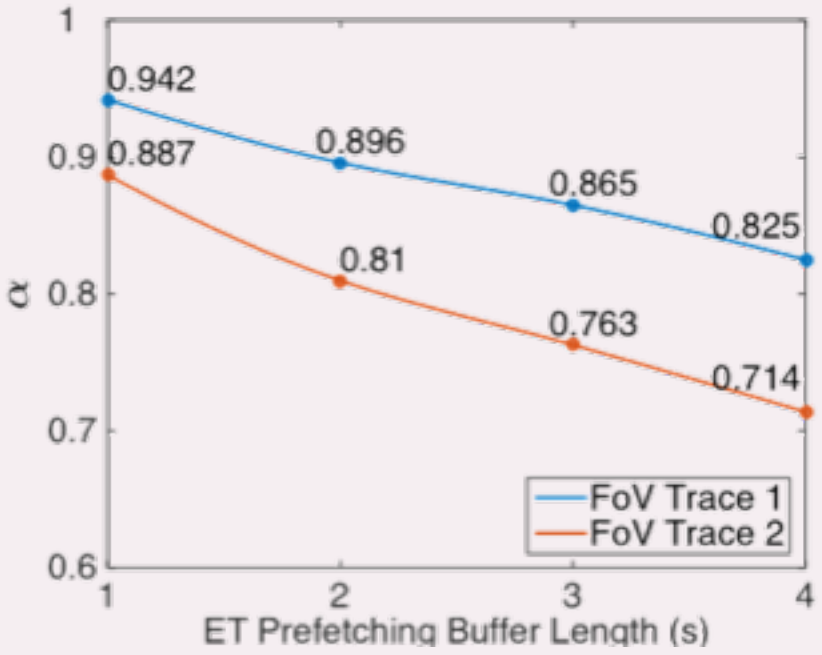
- Fluctuant
- Hard to predict



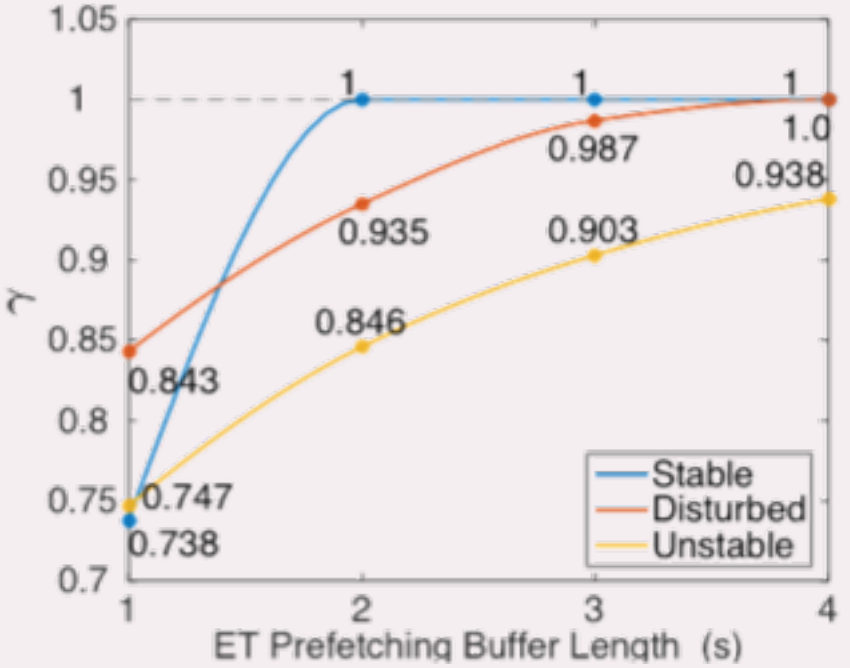
ET Buffer Length Optimization

$$Q(R_b; \alpha, \gamma, R_t) = \alpha\gamma Q_e(\tilde{R}_e) + (1 - \alpha\gamma) Q_b(\tilde{R}_b)$$

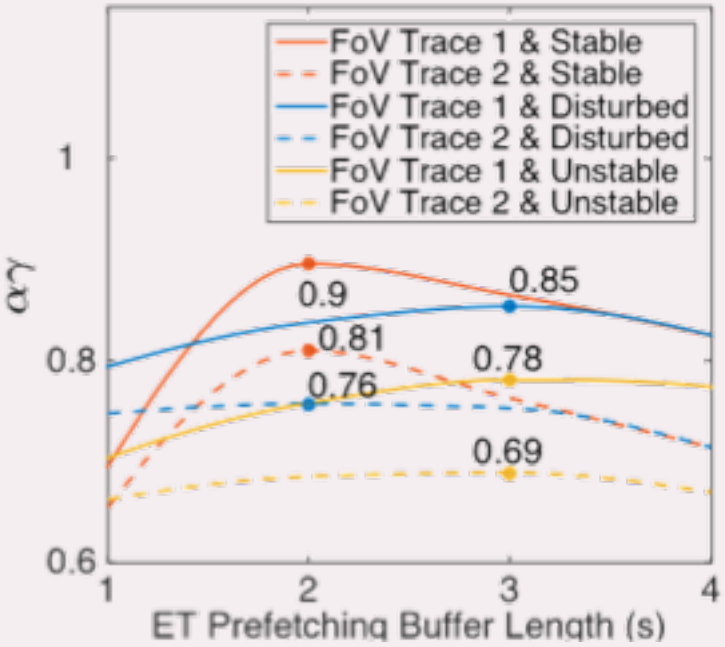
Given rate allocation, $\max Q \rightarrow \max\{\alpha\gamma\}$



FoV prediction accuracy



ET chunk delivery ratio



Effectiveness of ET chunk

Optimal ET Buffer Length & Rate Allocation

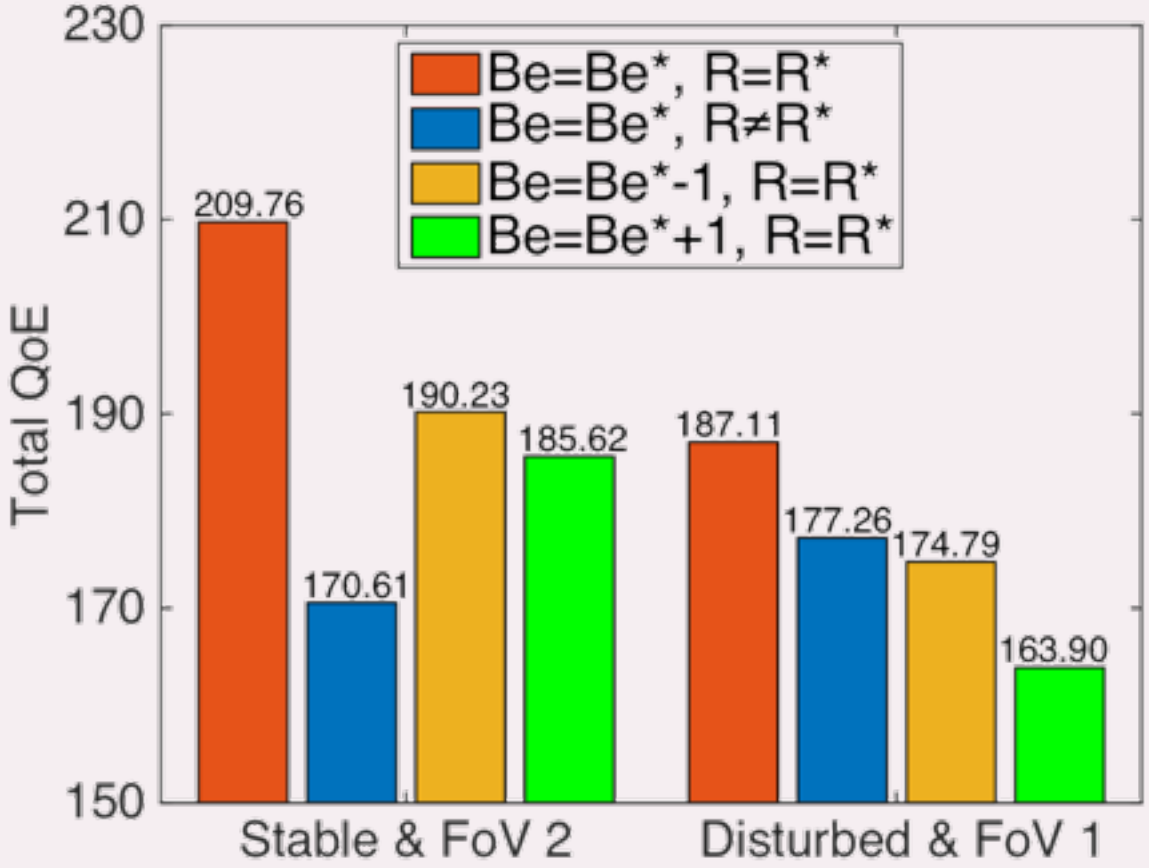
Bandwidth Traces	FoV Traces	B_e^* (s)	$\alpha\gamma$	Rate Allocation (Mbps)			
				R_b	R_{e1}	R_{e2}	R_{e3}
Stable	Fov 1	2	0.90	45.7	433.9	578.5	723.1
	Fov 2	2	0.81	89.8	400.8	534.4	668.0
Disturbed	Fov 1	3	0.85	62.7	373.5	498.0	622.5
	Fov 2	2	0.76	103.4	343.0	457.4	571.7
Unstable	Fov 1	3	0.78	83.3	310.7	414.3	517.8
	Fov 2	3	0.69	120.9	282.5	376.6	470.8

Network utilization rate $\eta = 85\%$

$\overline{BW_S} = 743.3$ Mbps, $\overline{BW_D} = 659.7$ Mbps, $\overline{BW_U} = 585.3$ Mbps

$R_{e2} = R_e^*$, $R_{e1} = 0.75 \cdot R_{e2}$, $R_{e1} = 1.25 \cdot R_{e2}$

Optimality Validation



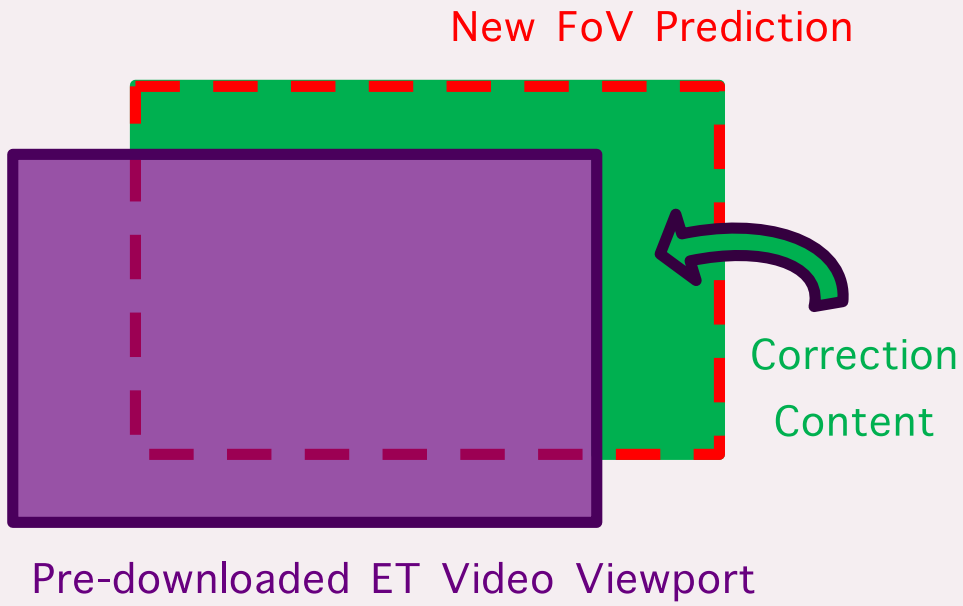
- Case 1: Both B_e and R are optimal
- Case 2: B_e is optimal, R is non-optimal
- Case 3: $B_e = B_e^* - 1$, R is optimal
- Case 4: $B_e = B_e^* + 1$, R is optimal

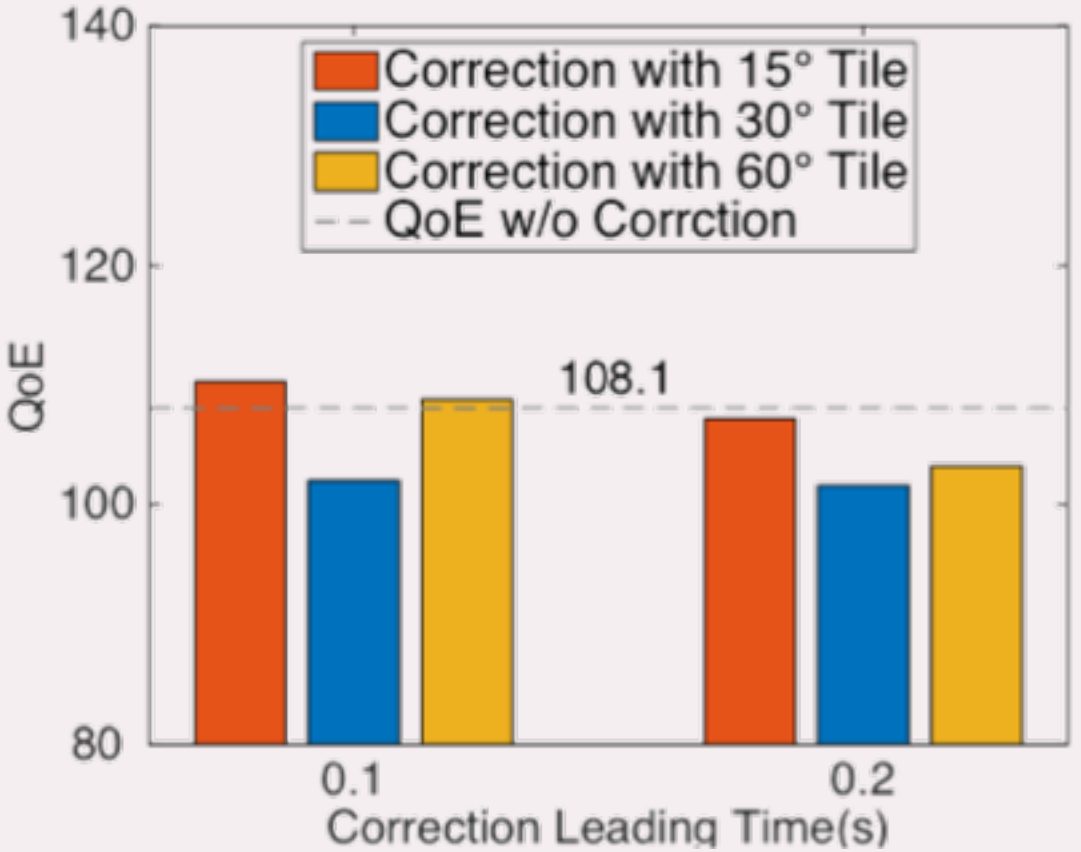
- QoE = a log ($R_{rendered}$)-b FreezRatio

- Optimized rate allocation and buffer length provides higher QoE!

Beyond Two Tier: FoV Correction

- Repredict the FoV for the segment to be displayed within next second.
- Request missing portion in tiles
- 5G network can have<10ms latency





$B_e^* = 3\text{s}$

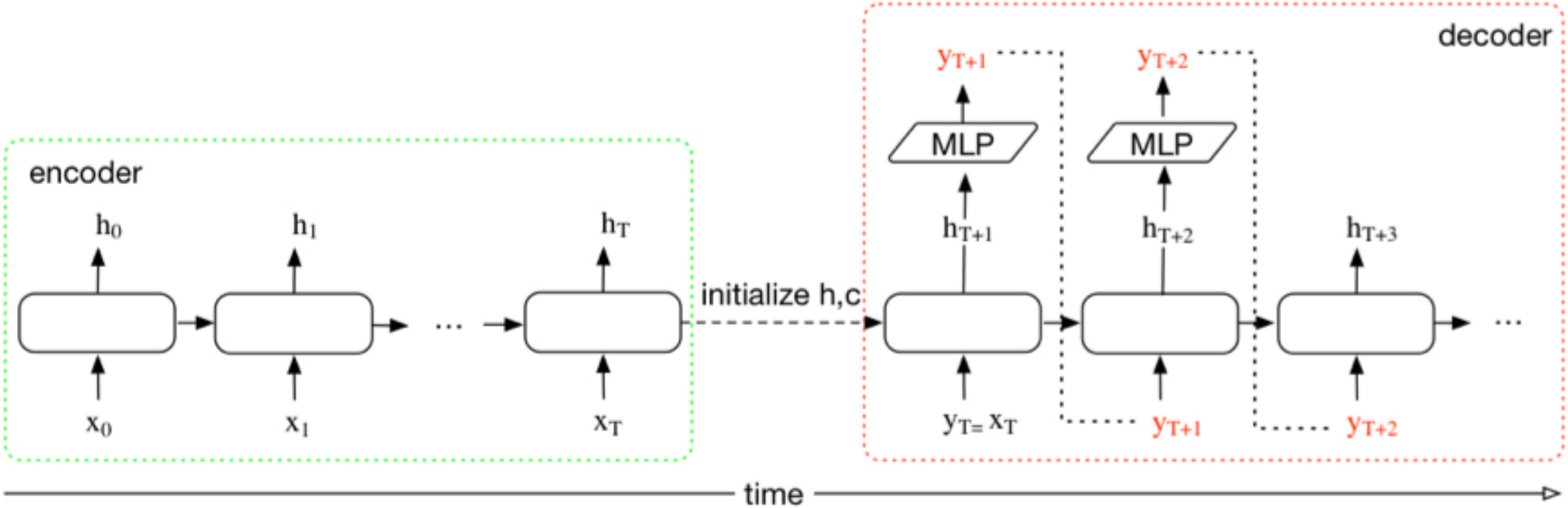
FoV Prediction for On-demand Streaming

- Each chunk covers a future video segment (1s long)
  - Need to predict the FoV span over the entire segment
  - Not necessary to predict framewise trajectory
  - Ex. Just predict mean and variance of FoV centers
- To provide robustness against network dynamics, want to prefetch as far ahead as possible (multiple seconds ahead!)
- Predicting where I will look seconds ahead is hard!

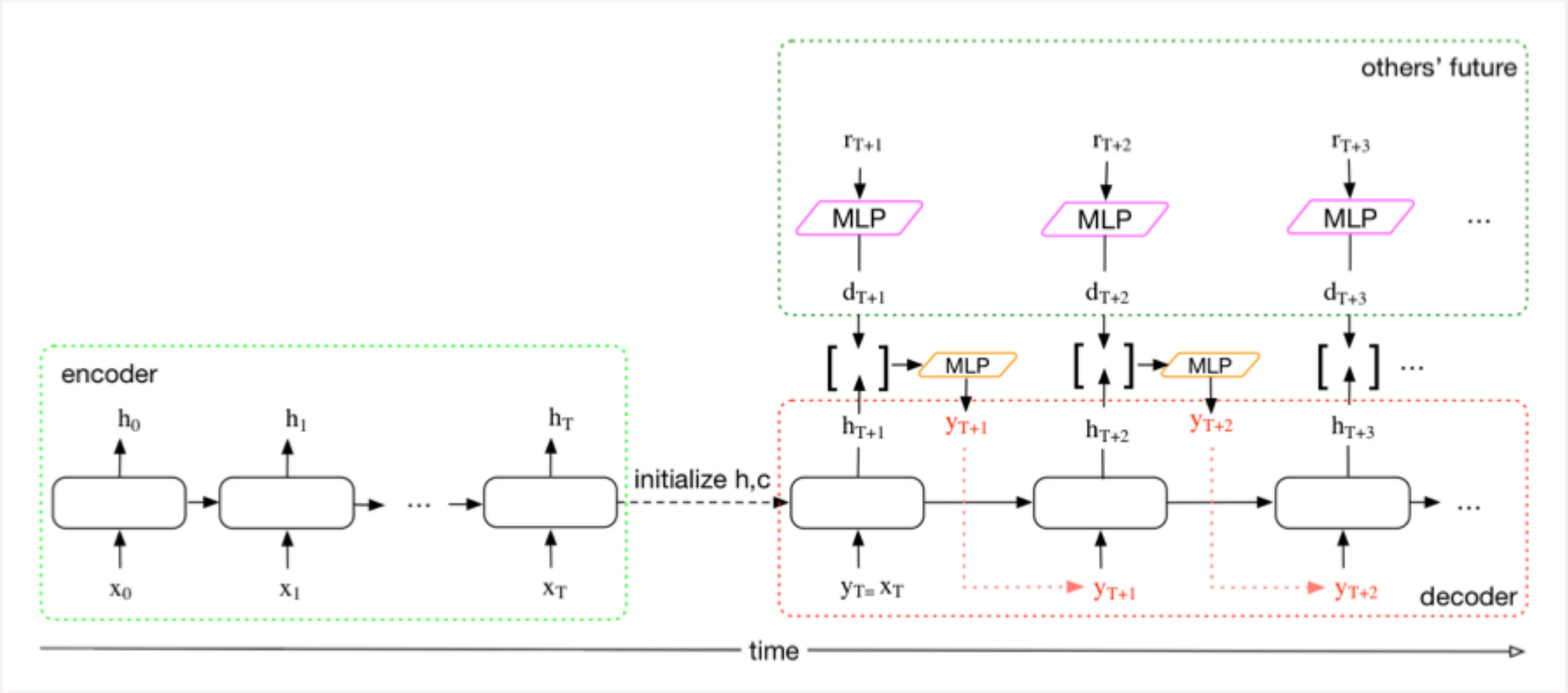
Possible Avenues for FoV Prediction

- From the past FoV center trajectory of the viewer only
- Using video content (at server) as well as the past trajectory
- Using other viewers' past and future trajectories (collected at server) as well as the target viewer's past trajectory
  - The same video may have been watched by many viewers
  - Distribution of other viewers' FoV centers ~ Saliency maps
- Leveraging machine learning for each

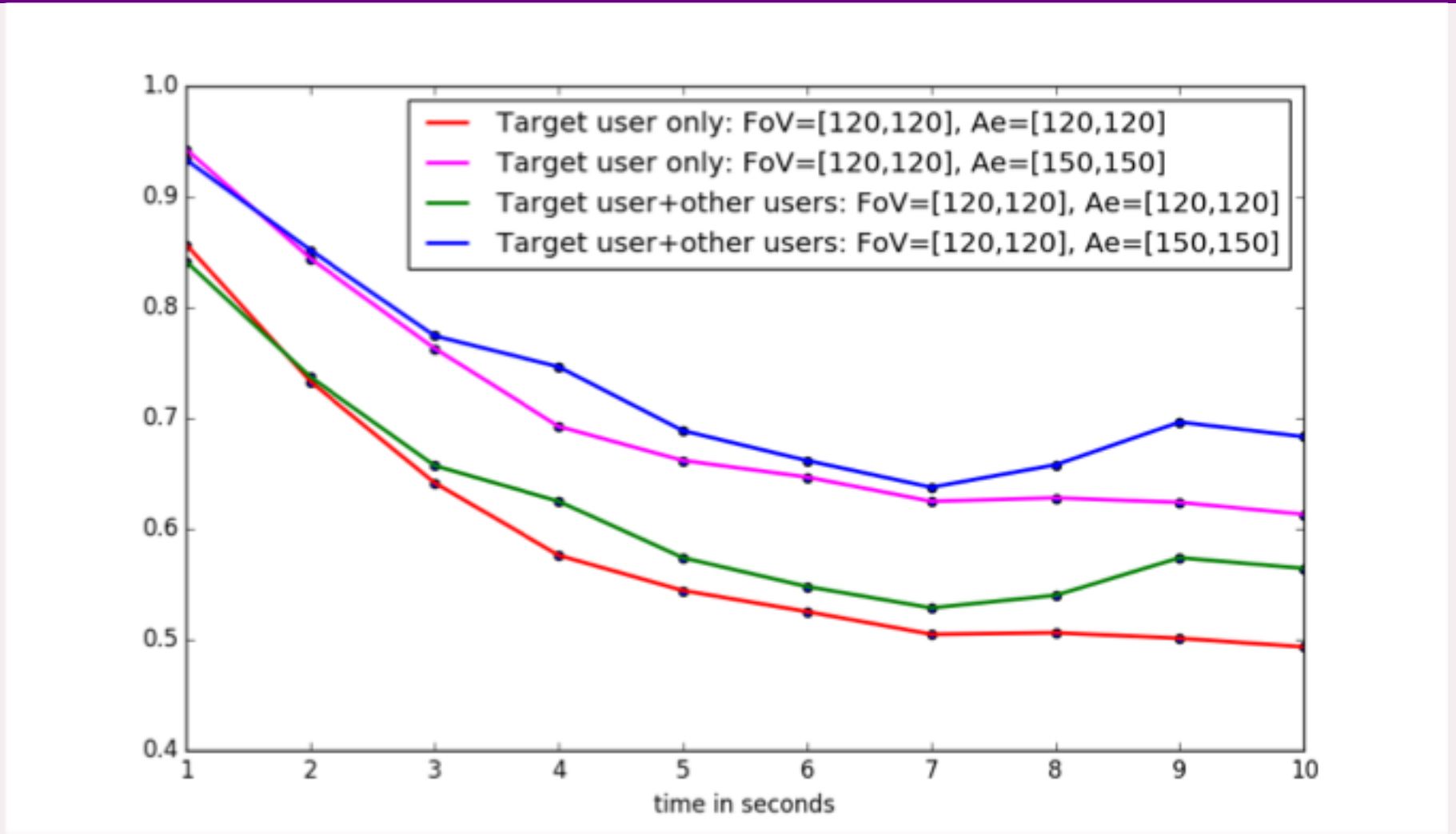
Using Viewer's Past Trajectory



Leveraging Other Viewers' Trajectories



Preliminary FoV Prediction Results



Trained and tested on FoV traces from: Chenglei Wu, Zhihao Tan, Zhi Wang, and Shiqiang Yang, "A Dataset for Exploring User Behaviors in VR Spherical Video Streaming," In Proc. (MMSys'17). 2017.

Streaming decision (Chunk Scheduling)

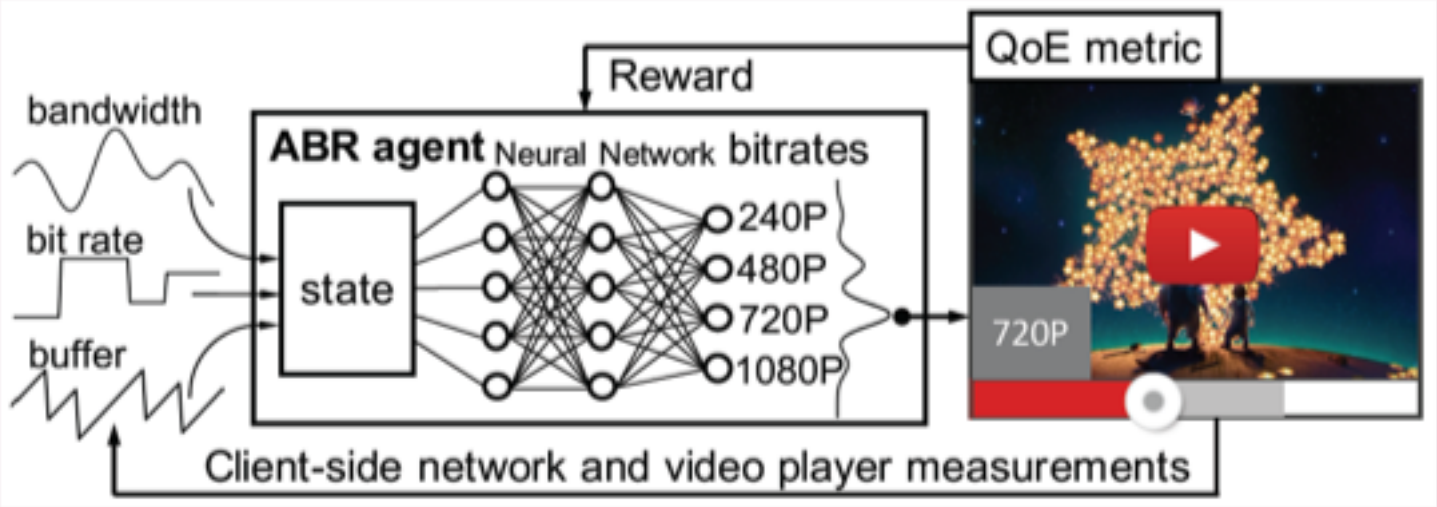
- After the arrival of each previously requested chunk
  - Download next ET or BT chunks? (instant quality vs. long-term robustness)
  - Which BT/ET chunks?
    - Which Rate/quality level? (DASH problem)
    - Which Viewport ? (FoV prediction)
- Simplification:
  - Perform FoV prediction independently
  - Streaming decision only decide
    - BT or ET chunk?
    - Which rate version?

Chunk Scheduling as Reinforcement Learning

- Average rate should match network bandwidth
  - When buffer is high, can download at rate higher than predicted throughput!
  - When buffer is low, should download at lower rate to prevent video freezing
- Each action affects future and should contribute to long term reward
- Formulation as Reinforcement learning
  - State variables: current buffer status for each buffer, past throughput based on the download time, possible chunk rates, etc.
  - Actions: which tier? Which rate version?
  - Reward: quality of rendered video, freezing (stall), smoothness of quality over entire streaming session

Deep Reinforcement Learning for Planar Video Streaming

- Deep reinforcement learning
  - Can handle continuous and many state variables
  - Do not require explicit models (data driven)
- Pensieve: DRL for DASH
  - Train a Critic Network and a Actor Network
  - Streaming using only the critic network



Hongzi Mao, Ravi Netravali, Mohammad Alizadeh, "Neural Adaptive Video Streaming with Pensieve." In Proceedings of SIGCOMM '17, August 21-25, 2017, Los Angeles, CA, USA,

"Pensieve" for Two-Tier Streaming

- More state variables and actions:
  - Need to consider both BT and ET buffers and their rate versions
- More complicated reward evaluation
  - The reward of a downloaded ET chunk depends on its FoV hit rate
  - A ET chunk is not useful if the corresponding (in time) BT chunk is not available
  - A ET chunk should be skipped if it cannot arrive before display time (no freezing caused by late ET)
- Work in progress …

Other streaming applications

- Interactive streaming
- Live streaming

Interactive 360 Streaming

- Video call or conference or gaming in 360!
  - Recipient can watch any view of the remote site
- Live encoding and delivery within 150 ms to enable interactions
  - Frame-based encoding and delivery (instead of segment-based)
- Three options
  - Detect and feedback recipient FoV, code and deliver only the FoV!
    - Only if the network delay between two parties <=10 ms
  - Predict FoV in next few frames at sender and code/send predicted FoV
    - How to code/deliver to mitigate effect of FoV prediction error?
  - Send 360 video directly
    - Will we ever have enough bandwidth?

Challenges for Frame-Based FoV Coding

- How to do temporal prediction if FoV keeps changing ?
  - How about sending periodic I frame with 360 span?
  - How about sending rolling I-regions?
- How to anticipate FoV prediction error?
  - How about coding an extended FoV with variable quality?
  - How to do bit allocation based on estimated FoV hit rate at each region?
- What if the delivered frame does not contain the entire FoV?
  - Error concealment in the FoV context!
- Others ?

Live 360 Streaming

- Sports/concert/major events, education/training in 360!
- Multicast (one to many)
- Use DASH framework (time-shifted live event):
  - Sender prepares multiple viewports, each at different rates
  - Recipient requests viewport and rate based on its predicted FoV and network/buffer conditions
  - Must use a short prefetching buffer to reduce the delay from the live event
  - Can incorporate two tier structure or its variants to coop with FoV prediction error and bandwidth variation

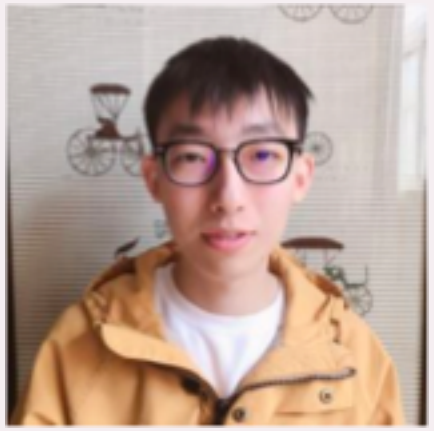
Concluding remarks

- 360 video brings rich and challenging research problems for video coding and streaming researchers!
  - Different constraints and challenges for interactive, live, and on-demand applications
- Tight coupling between coding and streaming
  - Robust-first design principle! (robust to FoV and network dynamics)
- New challenges for video coding
- New challenges for quality assessment
  - Viewing predownloaded 360 video
  - Viewing 360 video in FoV-based streaming sessions
  - Viewing using HMD vs. laptop/phone

Thanks

Prof. Yong Liu

Dr. Fanyi Duanmu



Liyang Sun

Chenge Li

Yixiang Mao

Weixi Zhang

Apurv Gupta

Project Sponsors

HUAWEI

NYU WIRELESS

Related publications

- Fanyi Duanmu, Eymen Kurdoglu, Yong Liu, Yao Wang, View Direction and Bandwidth Adaptive 360 Degree Video Streaming Using a Two-Tier System, IEEE International Symposium on Circuits and Systems (ISCAS), 2017.
- Fanyi Duanmu, Eymen Kurdoglu, S. Amir Hosseini, Yong Liu and Yao Wang, Prioritized Buffer Control in Two-tier 360 Video Streaming, ACM SigComm Workshop on VR/AR Network, 2017.
- Liyang Sun, Fanyi Duanmu, Yong Liu, Yao Wang, Yinghua Ye, Hang Shi, David Dai, Multi-path Multi-tier 360-degree Video Streaming in 5G Networks, in ACM Multimedia System Conference (MMSys), Amsterdam, Netherland, 2018.
- Fanyi Duanmu, Yixiang Mao, Shuai Liu, Sumanth Srinivasan, and Yao Wang, A Subjective Study of Viewer Navigation Behaviors When Watching 360-degree Videos on Computers, IEEE International Conference on Multimedia Expo (ICME), San Diego, California, USA, 2018.