Assignment-3 Checkpoint presentation

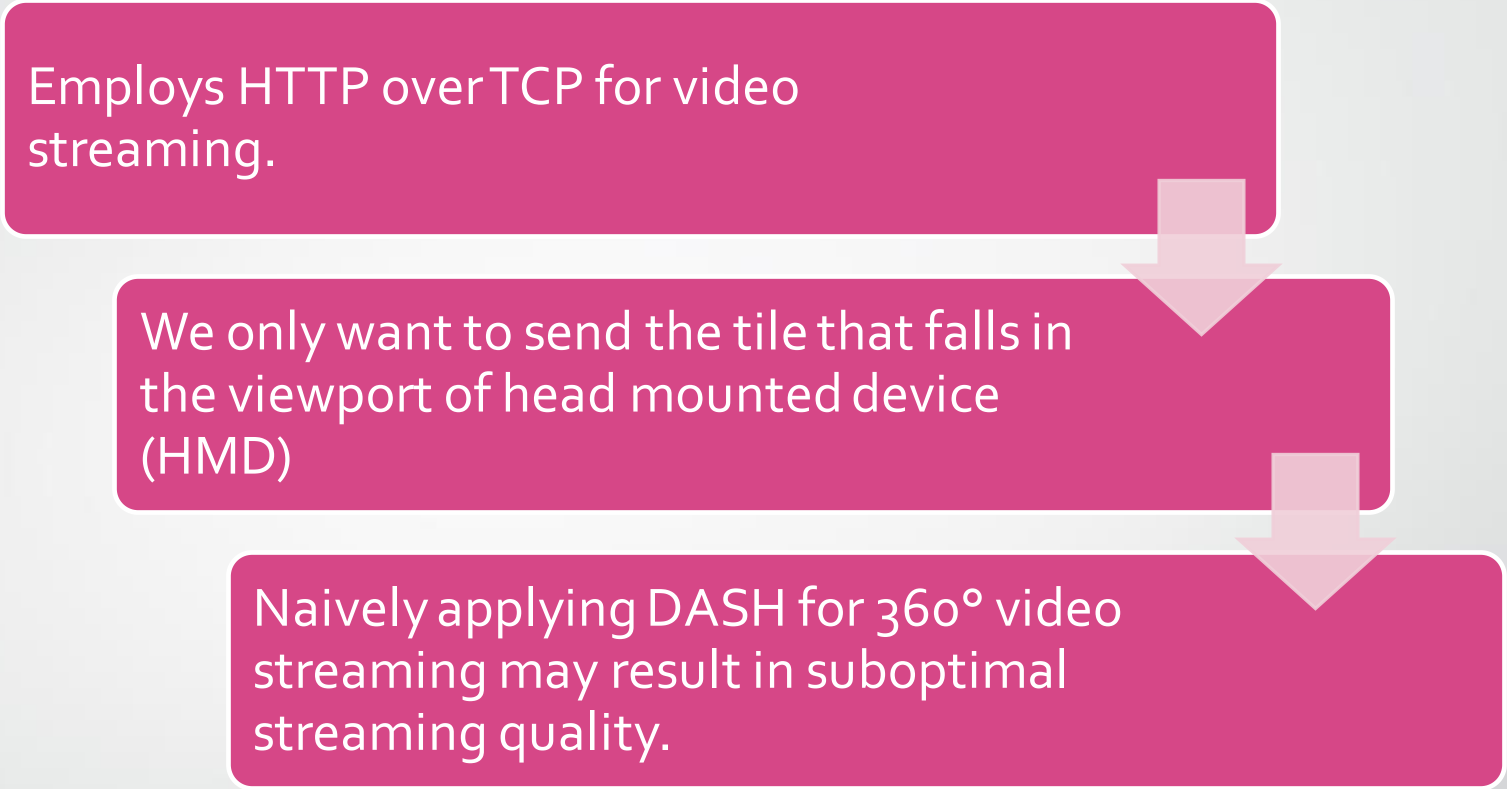
Group 2

Efficient streaming of 360° video using DASH over QUIC

Problem Statement

- Streaming VR video for headsets under the current network infra is suboptimal.
- We aim to follow and improve upon the paper "Streaming 360° videos to head-mounted virtual reality using DASH over QUIC transport protocol"
- Experiment with adaptive bitrate algorithms to improve FPS
- Modify DASH for ease of use and implementation

DASH Limitations



Employs HTTP over TCP for video streaming.

We only want to send the tile that falls in the viewport of head mounted device (HMD)

Naively applying DASH for 360° video streaming may result in suboptimal streaming quality.

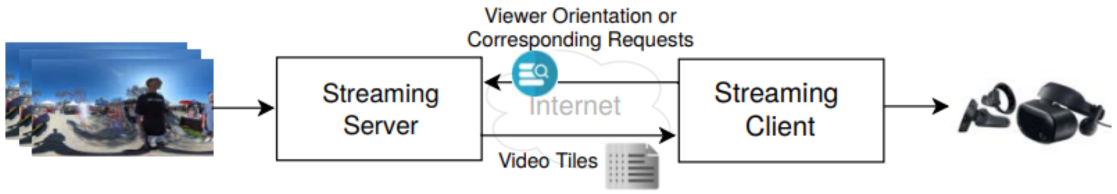
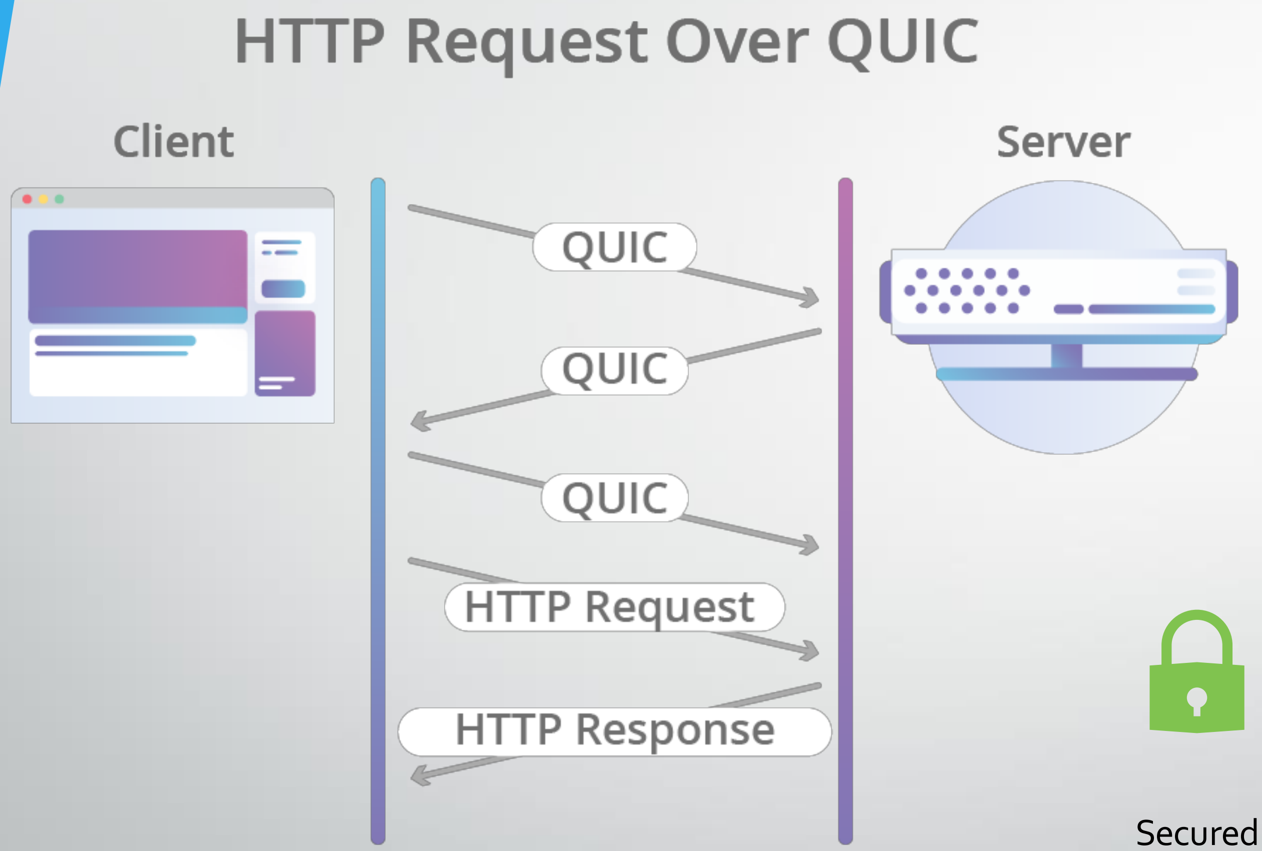


Figure 1: 360° video streaming needs high bandwidth, low latency.

Quick UDP Internet Connections (**QUIC**)



Secured communications.

Multiplexed streams with prioritized schedulers.

Low latency.

System Design (from paper)

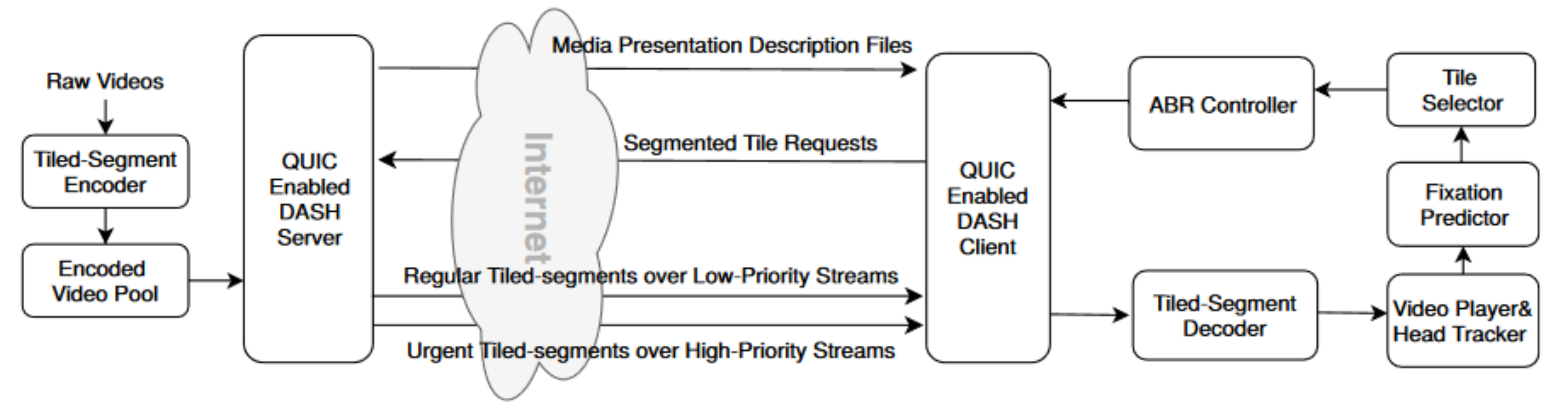


Figure 2: Architecture of the 360° video streaming using DASH over QUIC protocol.

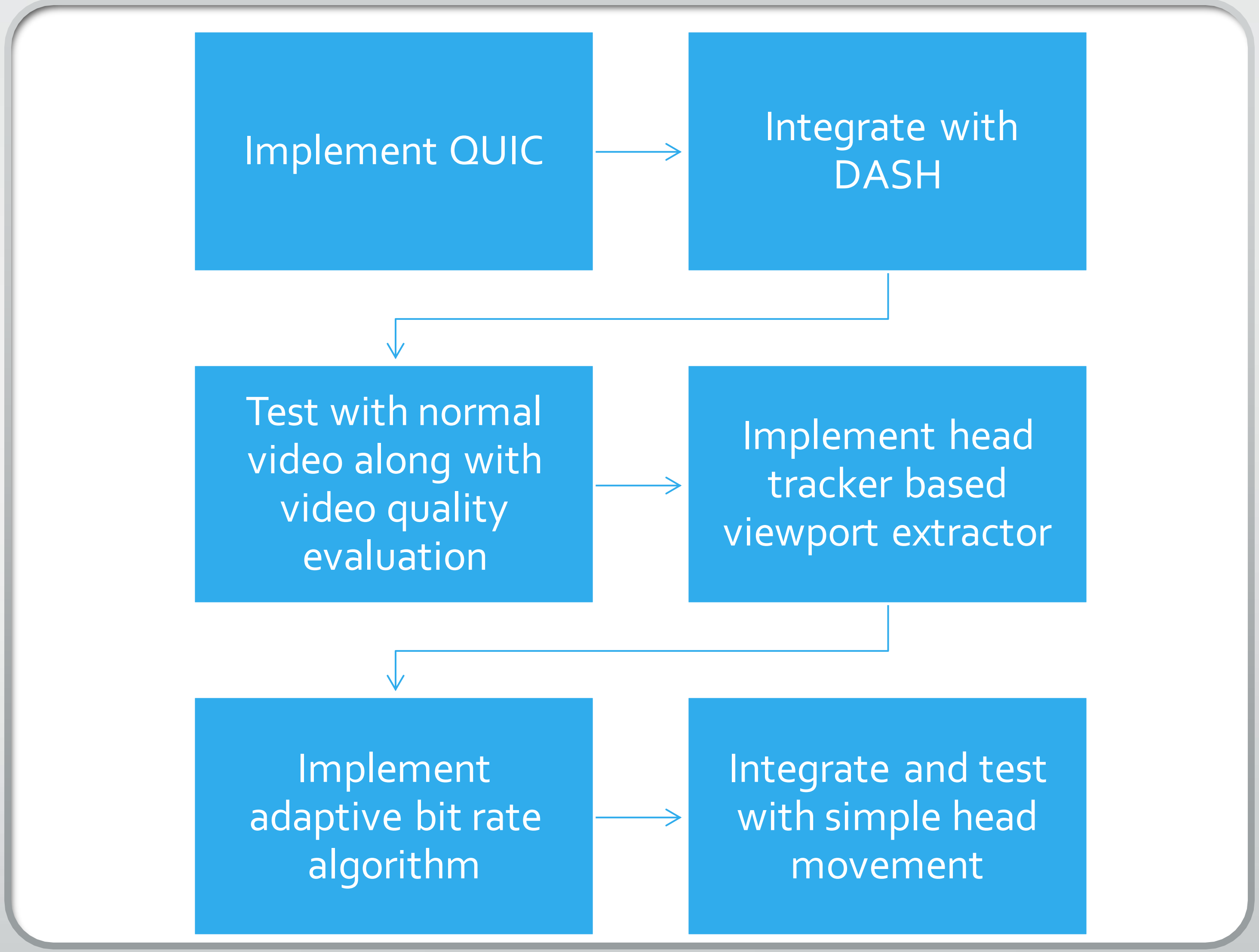
Shou-Cheng Yen, Ching-Ling Fan, and Cheng-Hsin Hsu. 2019. Streaming 360° videos to head-mounted virtual reality using DASH over QUIC transport protocol. In Proceedings of the 24th ACM Workshop on Packet Video (PV '19).

Project deliverables

Modules implemented by the paper

- Video streaming via DASH over QUIC
- Integrating head-tracker for 360° video
- ✓ Implementing bit-rate controller and view port extractor from video

Roadmap



Implement QUIC

Integrate with DASH

Test with normal video along with video quality evaluation

Implement head tracker based viewport extractor

Implement adaptive bit rate algorithm

Integrate and test with simple head movement

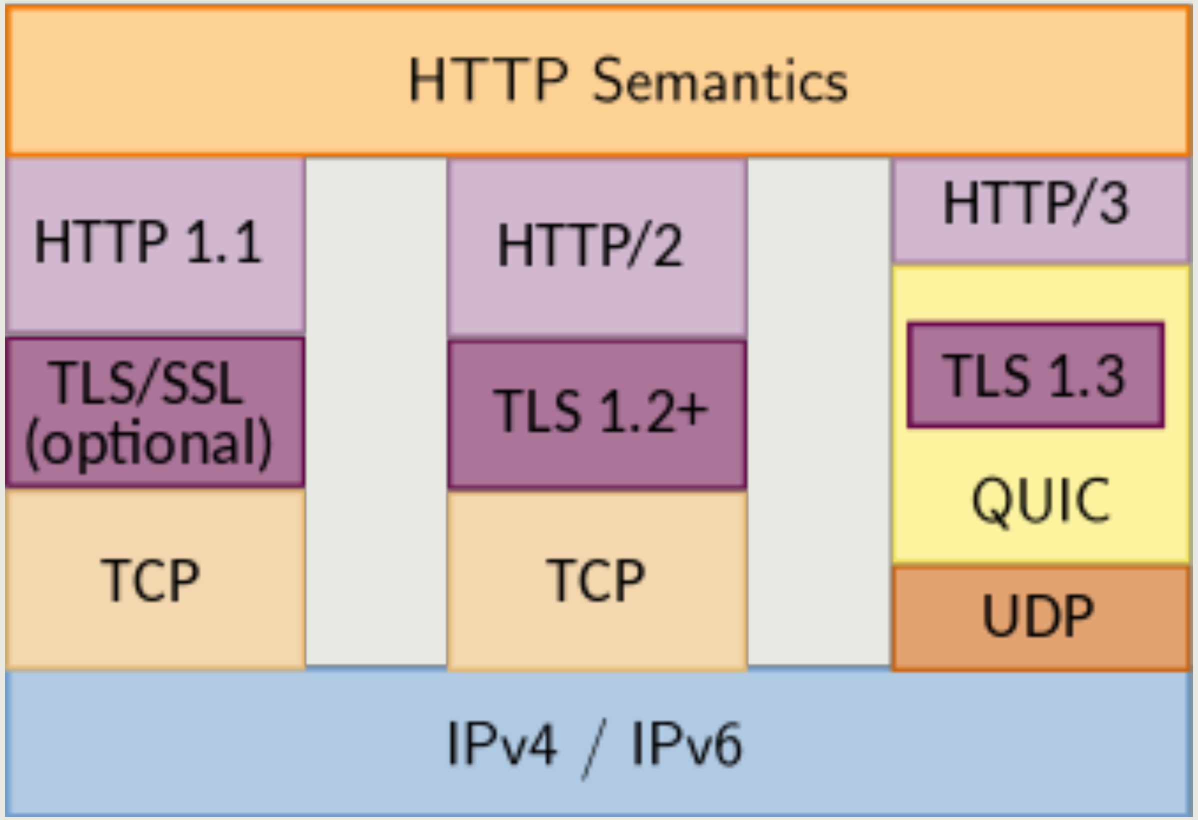
360° Video Streaming via DASH over QUIC

Final Presentation

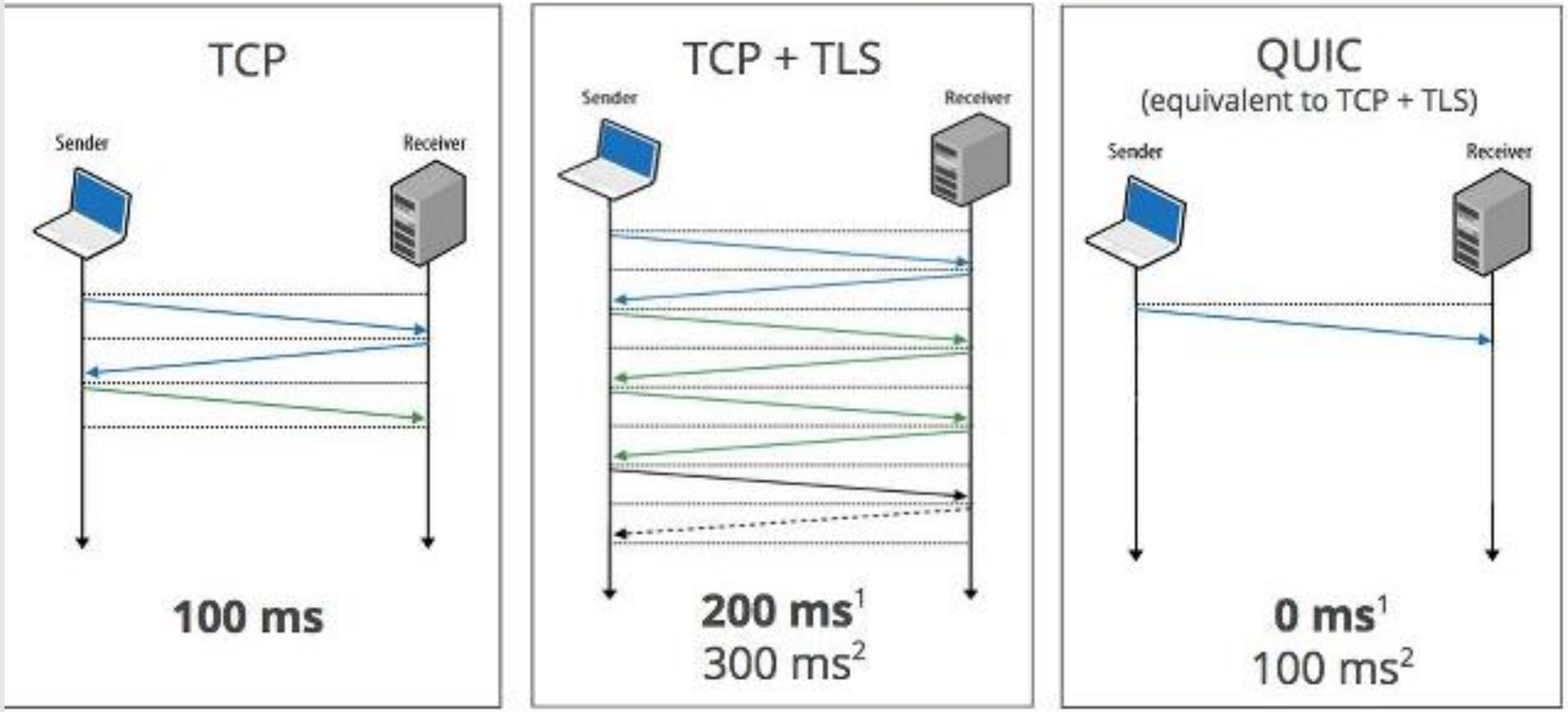
By: Group 2

QUIC

- Created by Google has now become IETF standard (RFC9000)
- Built over UDP so it is easy to integrate into the already present infrastructure.
- Used **aioquic** for implementation.

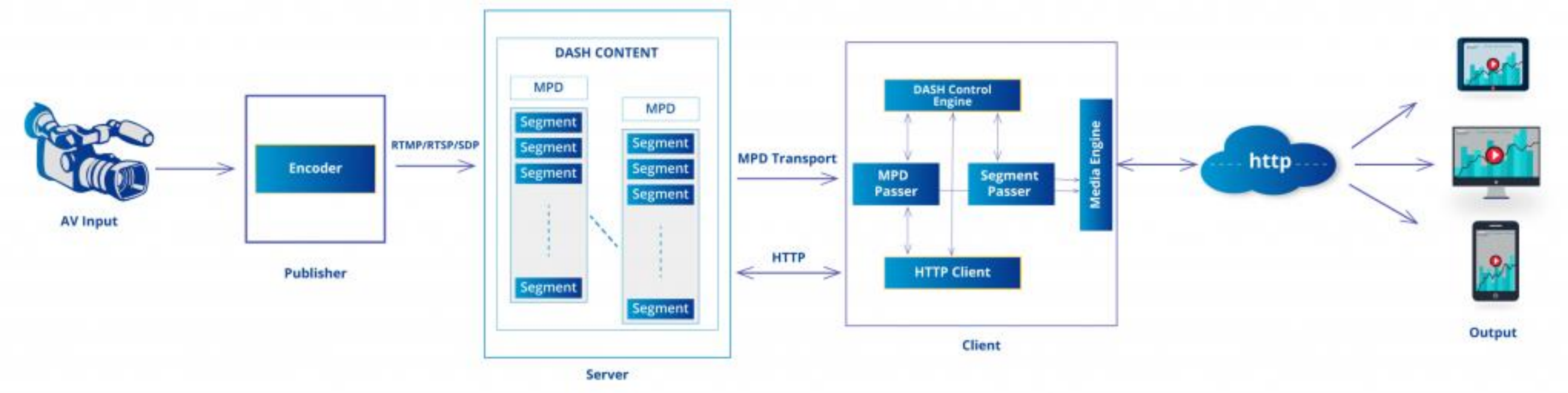


Zero RTT Connection Establishment



1. Repeat connection
2. Never talked to server before

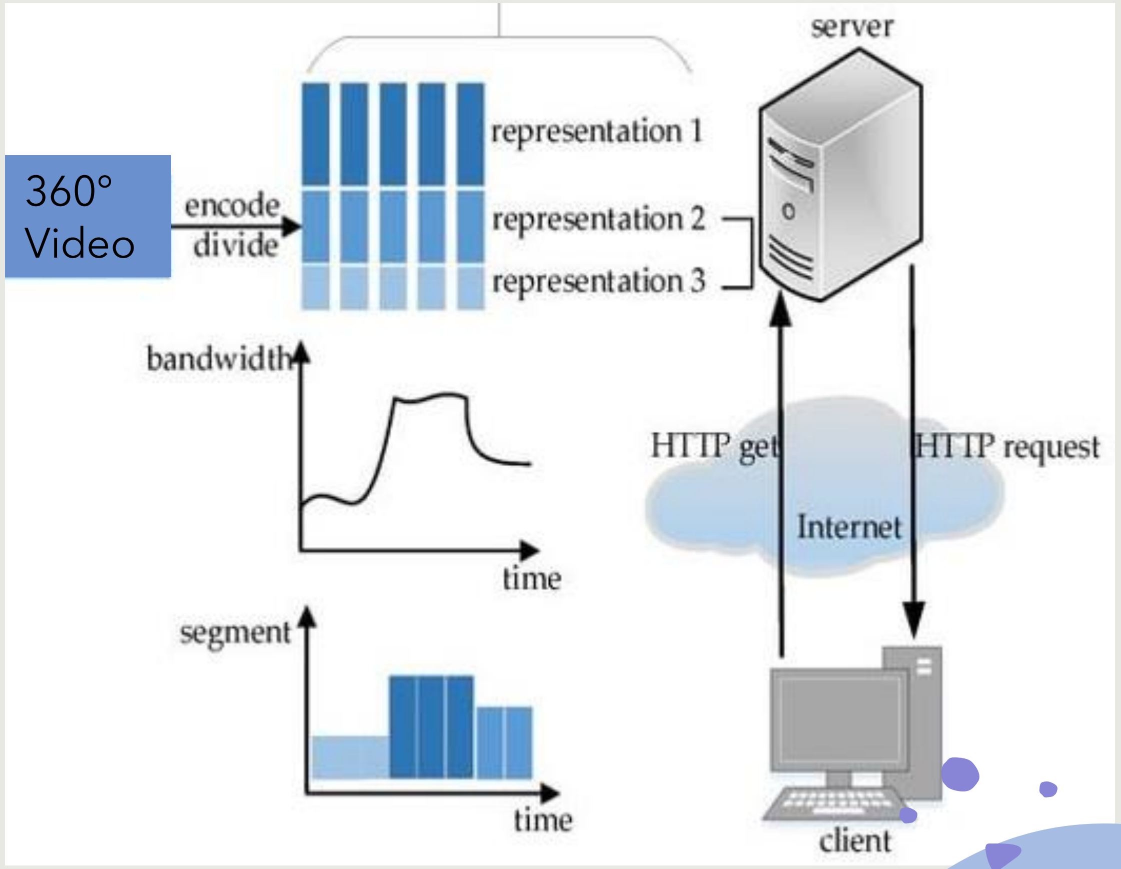
***DASH** (Dynamic Adaptive Streaming over HTTP)*



- Information of different video formats stored in .mpd file
- MPD file is communicated before the streaming starts
- Client uses MPD file as lookup table

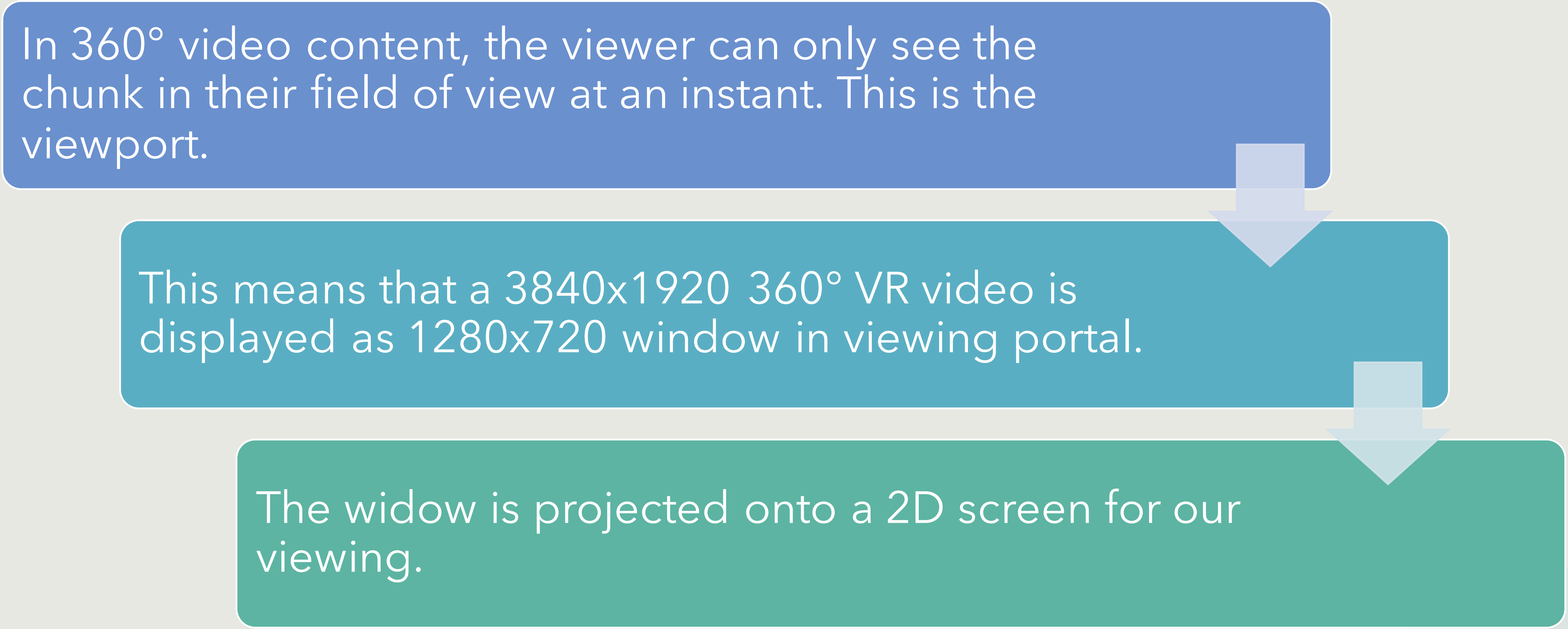
Our DASH Implementation

- We have encoded video at 3 qualities.
- When requesting for next frame, client shares the bandwidth information
- Given the available bandwidth, server will decide quality.
- ABR is based on switching using thresholds.
- Primitive version of industry standard DASH.



360° Video Encoding

Viewport of monoscopic 360° *VR*



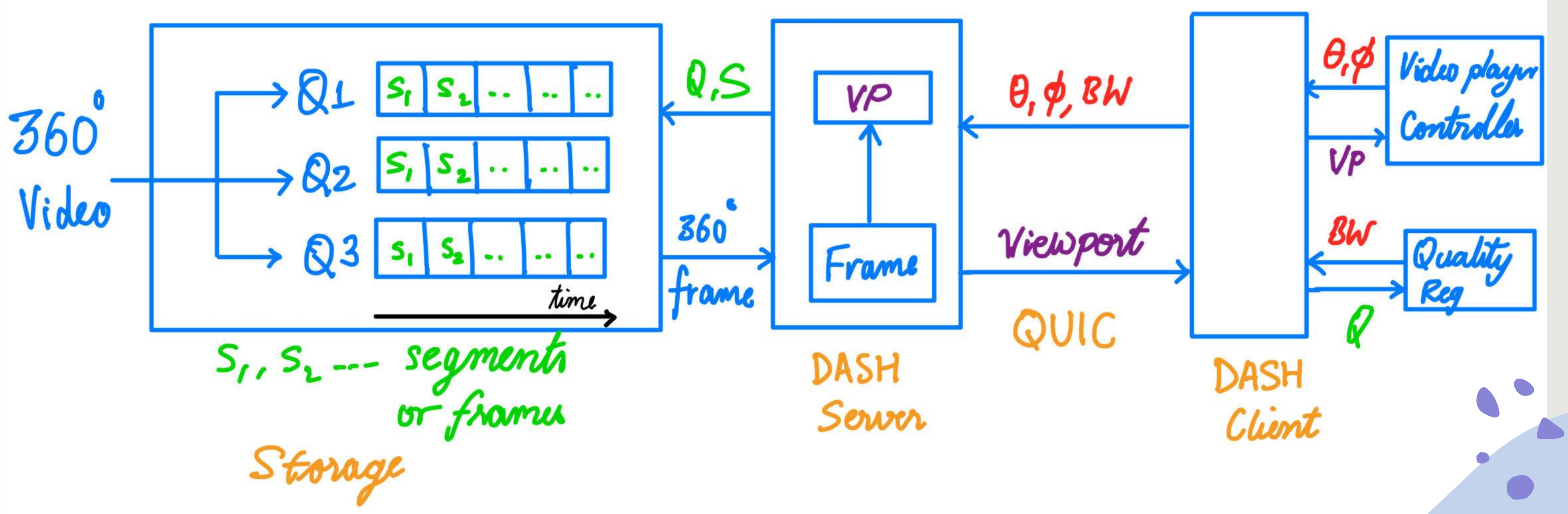
In 360° video content, the viewer can only see the chunk in their field of view at an instant. This is the viewport.

This means that a 3840x1920 360° VR video is displayed as 1280x720 window in viewing portal.

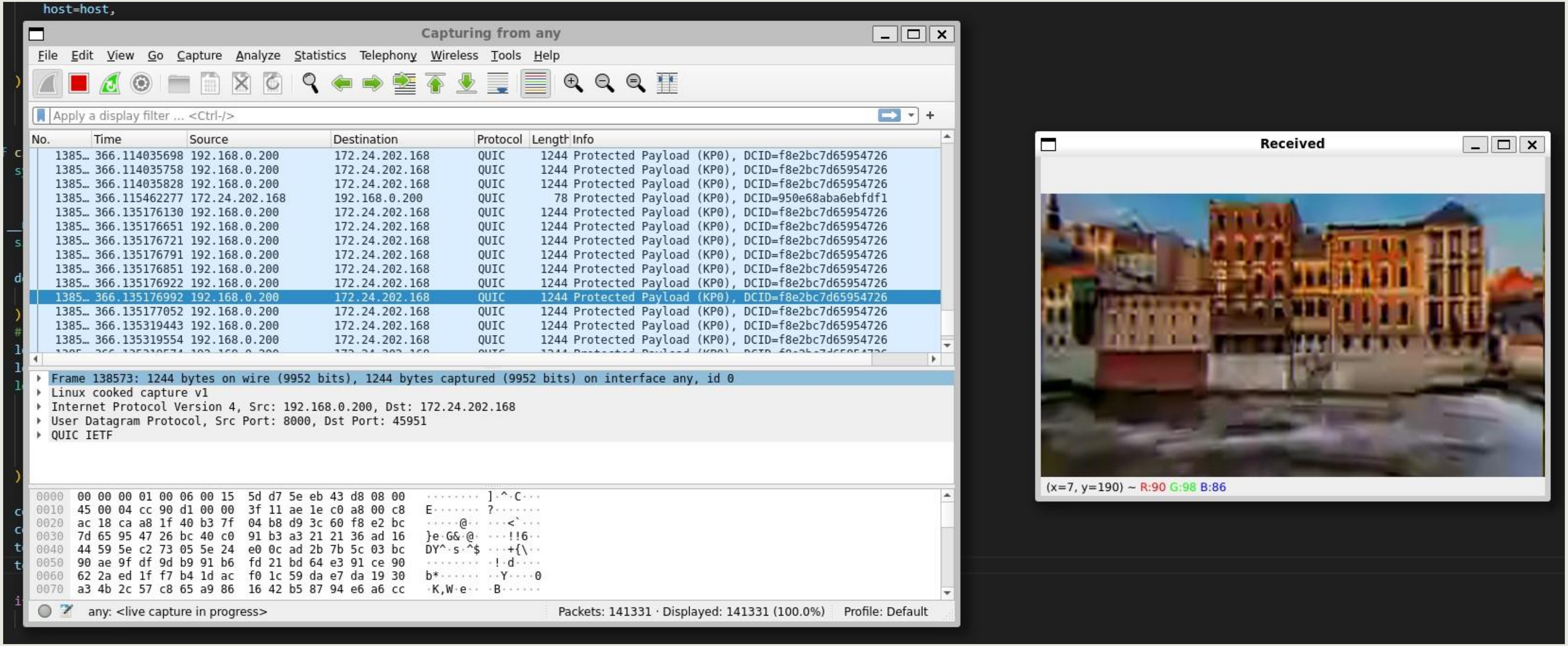
The widow is projected onto a 2D screen for our viewing.

ND5
PON

System Design



Wireshark Capture



```
host=host,
```

Capturing from any

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

```
No.     Time          Source          Destination     Protocol Length Info
1385… 366.114035698 192.168.0.200  172.24.202.168  QUIC     1244 Protected Payload (KP0), DCID=f8e2bc7d65954726
1385… 366.114035758 192.168.0.200  172.24.202.168  QUIC     1244 Protected Payload (KP0), DCID=f8e2bc7d65954726
1385… 366.114035828 192.168.0.200  172.24.202.168  QUIC     1244 Protected Payload (KP0), DCID=f8e2bc7d65954726
1385… 366.115462277 172.24.202.168 192.168.0.200   QUIC       78 Protected Payload (KP0), DCID=950e68aba6ebfdf1
1385… 366.135176130 192.168.0.200  172.24.202.168  QUIC     1244 Protected Payload (KP0), DCID=f8e2bc7d65954726
1385… 366.135176651 192.168.0.200  172.24.202.168  QUIC     1244 Protected Payload (KP0), DCID=f8e2bc7d65954726
1385… 366.135176721 192.168.0.200  172.24.202.168  QUIC     1244 Protected Payload (KP0), DCID=f8e2bc7d65954726
1385… 366.135176791 192.168.0.200  172.24.202.168  QUIC     1244 Protected Payload (KP0), DCID=f8e2bc7d65954726
1385… 366.135176851 192.168.0.200  172.24.202.168  QUIC     1244 Protected Payload (KP0), DCID=f8e2bc7d65954726
1385… 366.135176922 192.168.0.200  172.24.202.168  QUIC     1244 Protected Payload (KP0), DCID=f8e2bc7d65954726
1385… 366.135176992 192.168.0.200  172.24.202.168  QUIC     1244 Protected Payload (KP0), DCID=f8e2bc7d65954726
1385… 366.135177052 192.168.0.200  172.24.202.168  QUIC     1244 Protected Payload (KP0), DCID=f8e2bc7d65954726
1385… 366.135319443 192.168.0.200  172.24.202.168  QUIC     1244 Protected Payload (KP0), DCID=f8e2bc7d65954726
1385… 366.135319554 192.168.0.200  172.24.202.168  QUIC     1244 Protected Payload (KP0), DCID=f8e2bc7d65954726
```

```
Frame 138573: 1244 bytes on wire (9952 bits), 1244 bytes captured (9952 bits) on interface any, id 0
Linux cooked capture v1
Internet Protocol Version 4, Src: 192.168.0.200, Dst: 172.24.202.168
User Datagram Protocol, Src Port: 8000, Dst Port: 45951
QUIC IETF
```

```
0000  00 00 00 01 00 06 00 15  5d d7 5e eb 43 d8 08 00   ········ ]·^·C···
0010  45 00 04 cc 90 d1 00 00  3f 11 ae 1e c0 a8 00 c8   E······· ?·······
0020  ac 18 ca a8 1f 40 b3 7f  04 b8 d9 3c 60 f8 e2 bc   ·····@·· ···<`···
0030  7d 65 95 47 26 bc 40 c0  91 b3 a3 21 21 36 ad 16   }e·G&·@· ···!!6··
0040  44 59 5e c2 73 05 5e 24  e0 0c ad 2b 7b 5c 03 bc   DY^·s·^$ ···+{\··
0050  90 ae 9f df 9d b9 91 b6  fd 21 bd 64 e3 91 ce 90   ········ ·!·d····
0060  62 2a ed 1f f7 b4 1d ac  f0 1c 59 da e7 da 19 30   b*······ ··Y····0
0070  a3 4b 2c 57 c8 65 a9 86  16 42 b5 87 94 e6 a6 cc   ·K,W·e·· ·B······
```

any: <live capture in progress> Packets: 141331 · Displayed: 141331 (100.0%) Profile: Default

Received

(x=7, y=190) ~ R:90 G:98 B:86

Demo

Learnings

1. Better understanding of different HTTP versions
2. Asynchronous programming
3. DASH
4. QUIC protocols
5. Basic Concepts of videos and images
6. Fundamentals of 360° VR videos
7. Extraction of viewport from 360° VR video frame

Challenges

1. No good reference for DASH implementation. The one we found was old and did not work due to deprecated libraries.
2. QUIC is relatively newer technology and hence there is a lack of good tutorials. (As of April 2023, 8.9% of all websites use QUIC) (Wikipedia)
3. Limited sources and references for handling of 360° degree VR videos.
4. Academic challenges: Placement exams and University application deadlines.

References

The QUIC Protocol, HTTP3, and How HTTP Has Evolved | YouTube

https://github.com/aiortc/aioquic (QUIC python library)

https://github.com/najaco/quic-v-stream (Video Streaming)

https://github.com/pari685/AStream (DASH)

OmniCV 0.0.1 documentation (Viewport extraction)

Thank You