Ultra-low Latency Point Cloud Streaming in 5G

Yannis Thomas and George Xylomenos, AUEB

EUROXR Winterthur 2025

Motivation

NMP needs what 5G can provide

- NMP: Network Music Performance
- Ultra-low latency
  - NMP requires 30-40ms one way
- Very high bandwidth
  - Especially with volumetric video
- Processing at the edge
  - To relay or process media streams

Is NMP with PC streaming feasible with 5G?

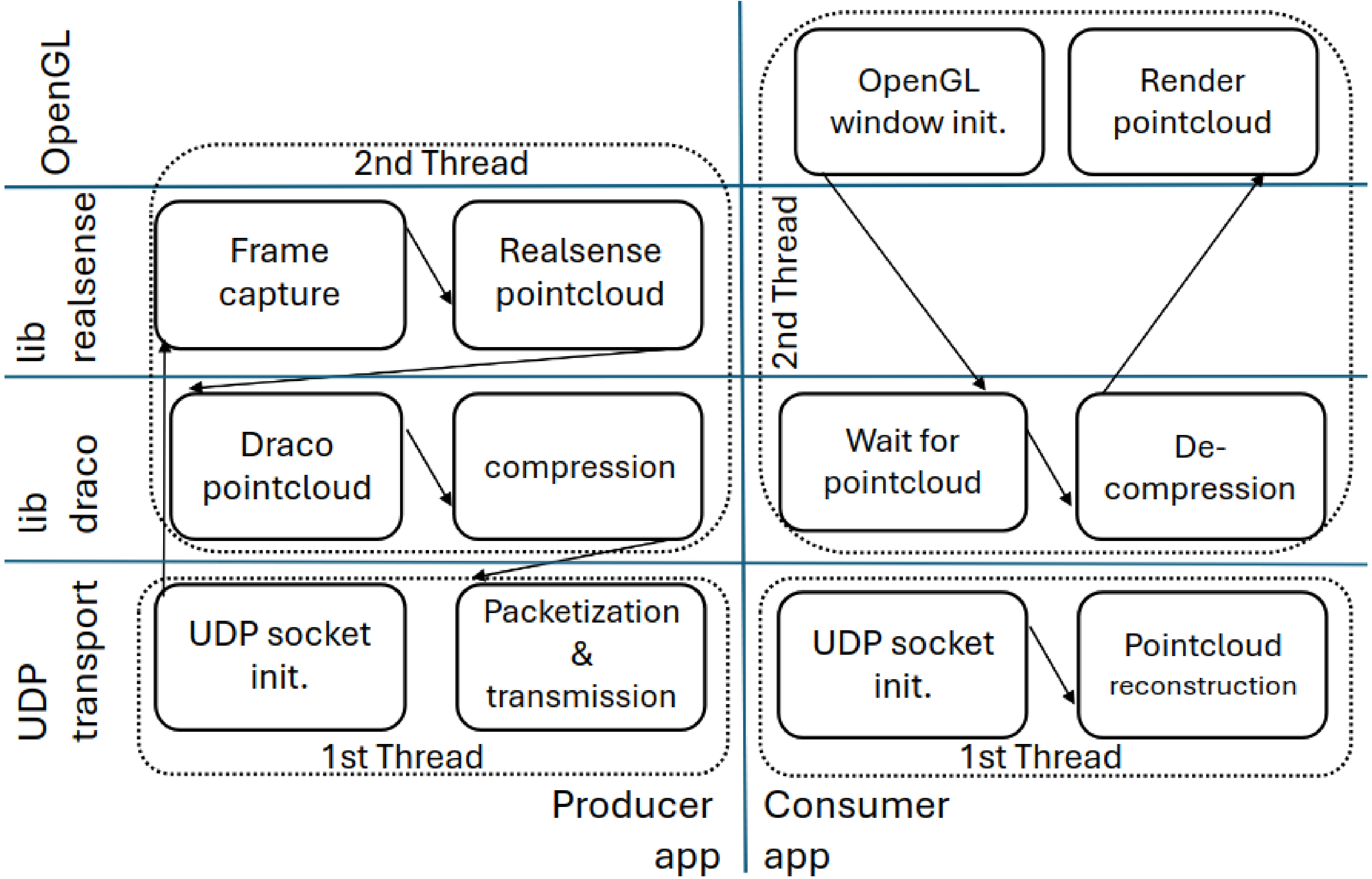
- Designed our own tool
  - Captures video from depth camera
  - Compresses PC with Draco
  - Frames and transmits the video
  - Renders with OpenGL
- Evaluated in 5G-SA testbed
  - Focus on latency reduction
  - Multithreading
  - Color drop
  - Resolution drop

Design & Implementation

Network setup

- SPIRIT Berlin 5G-SA testbed
  - Band N78, indoor area
  - 20.1 MBps throughput
  - 12.2 ms latency
- Endpoints
  - ASUS TUF A15 Ryzen 9 Ubuntu 24.04
  - Teltonika RUTX50 5G modems
  - Private IP network

Software setup: overview



EUROXR Winterthur 2025

Software setup: details

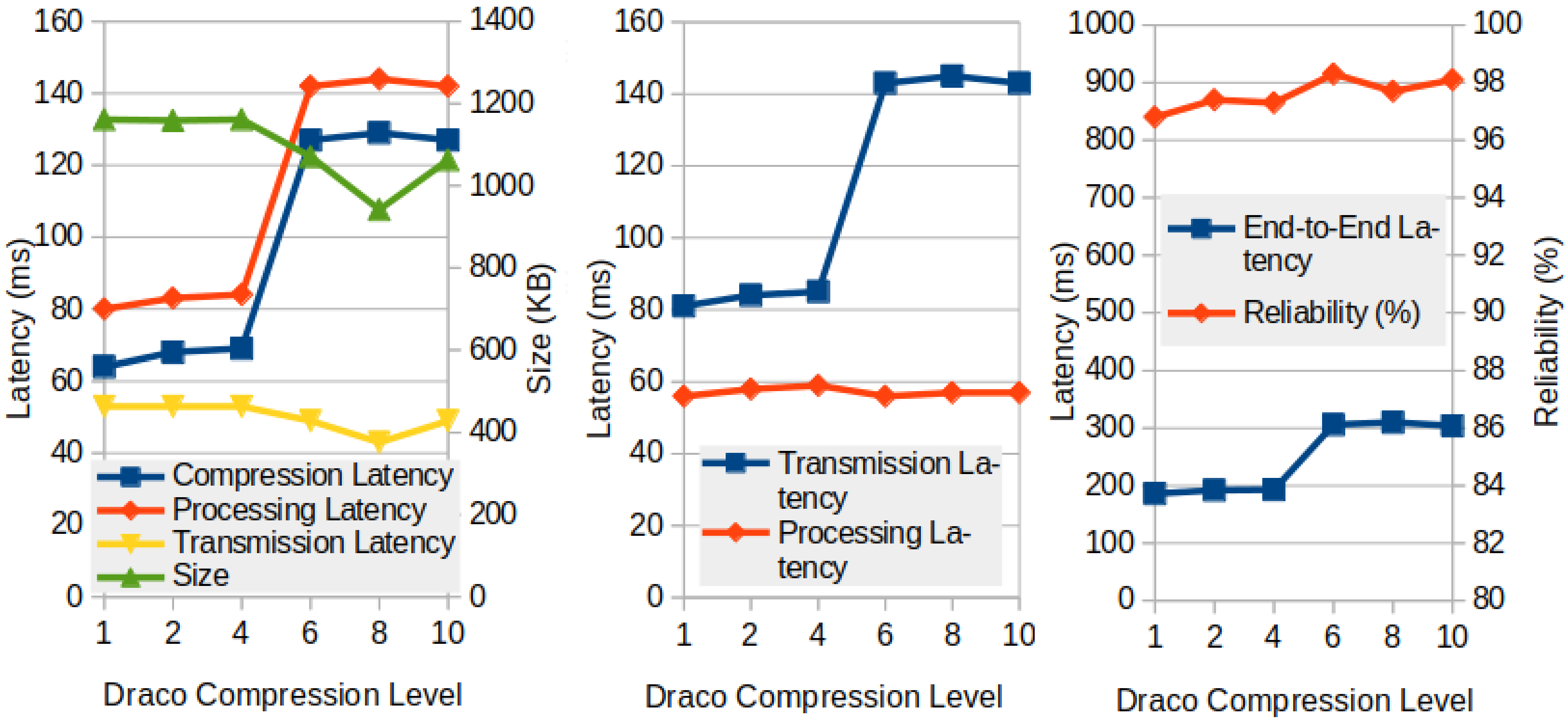
- Producer and consumer in C++
  - Network and processing threads
  - Producer can add more processing threads
  - Consumer does not seem to need it
  - FPS depends on overall latency
- Simple framing protocol
  - 1400 byte UDP packets
  - 4 bytes frame ID
  - 4 bytes chunk ID
  - Pacing to prevent drops

Performance metrics

- Processing latency (ms)
- Compression latency (ms)
- PC size (Bytes)
- Transmission latency (ms)
- End-to-end latency (ms)
- Reliability (%)
- Deployability limits
  - Processing latency < 33 ms
  - Bitrate < 20 MBps
  - End-to-end latency < 100 ms

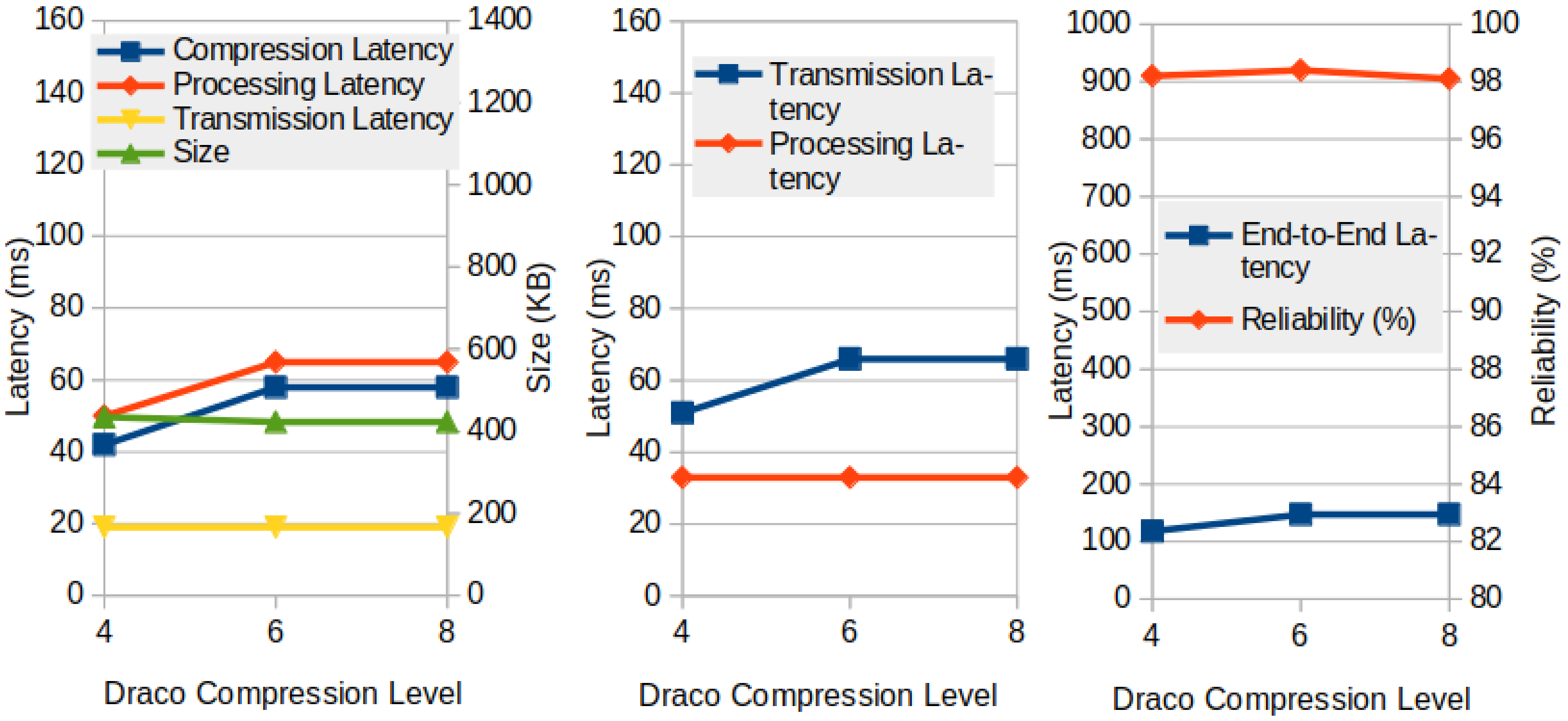
Results

Compression level



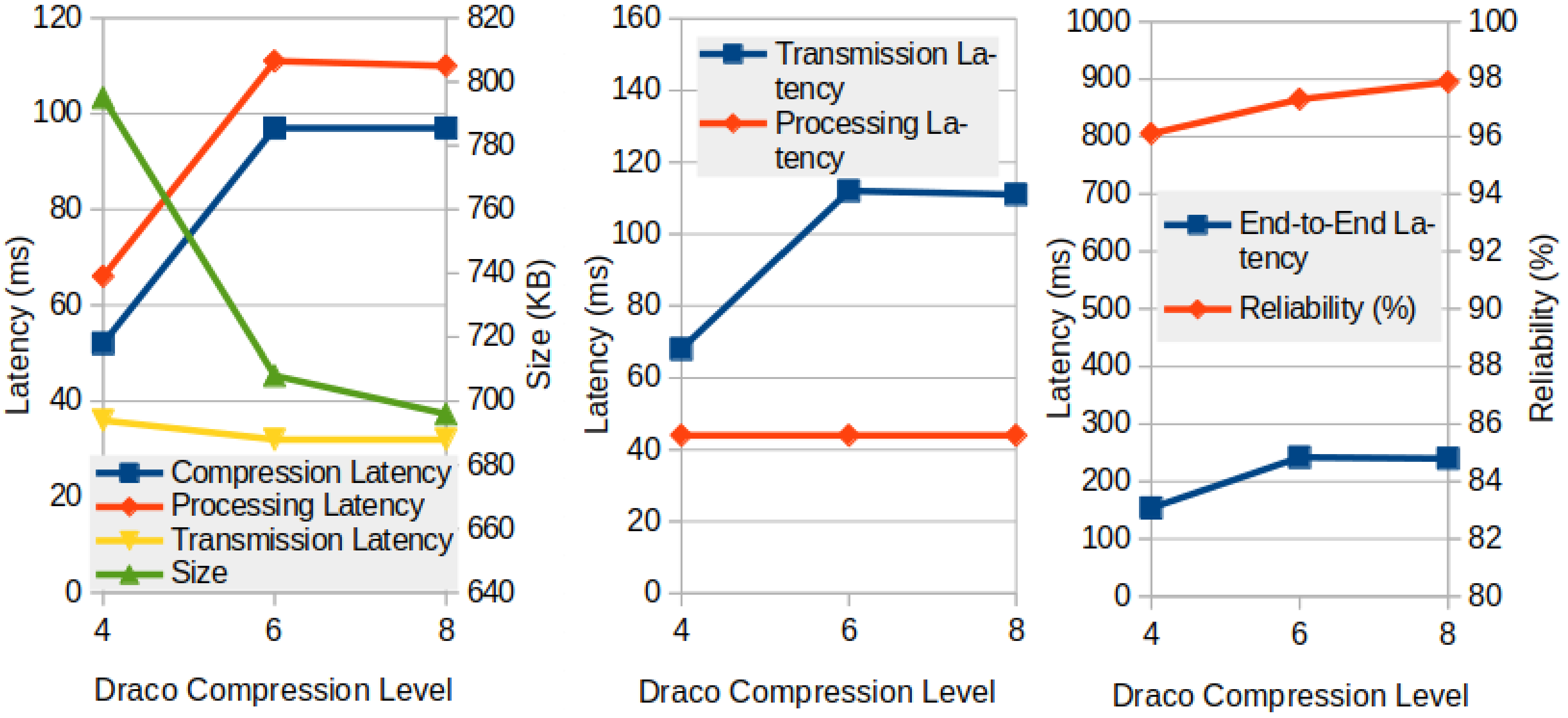
- Levels 4 to 8 are the most interesting
- Compression latency dominates – overall latency is very large

Drop color



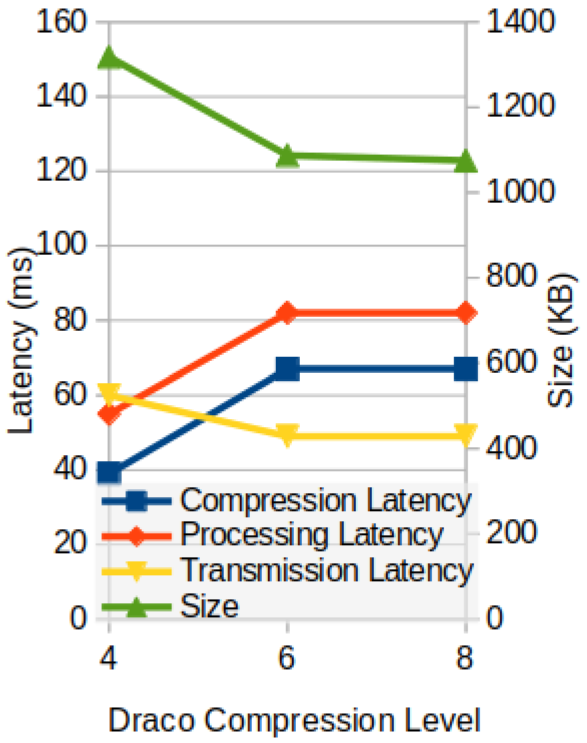
- Very big difference in processing and frame size
- Color compression is not very efficient in Draco

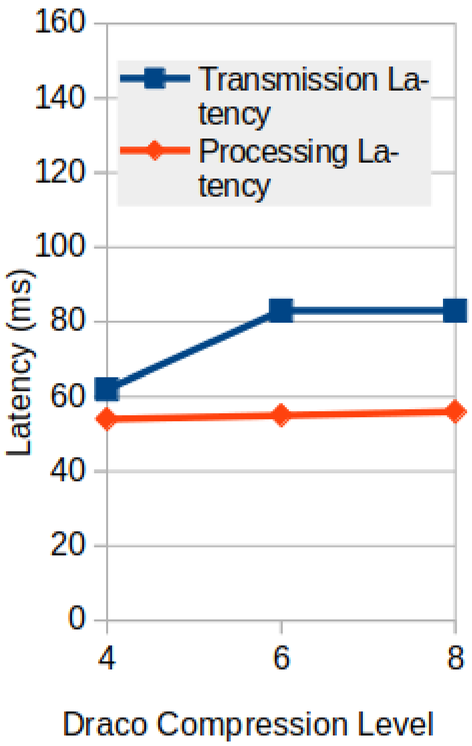
Drop 25% of points

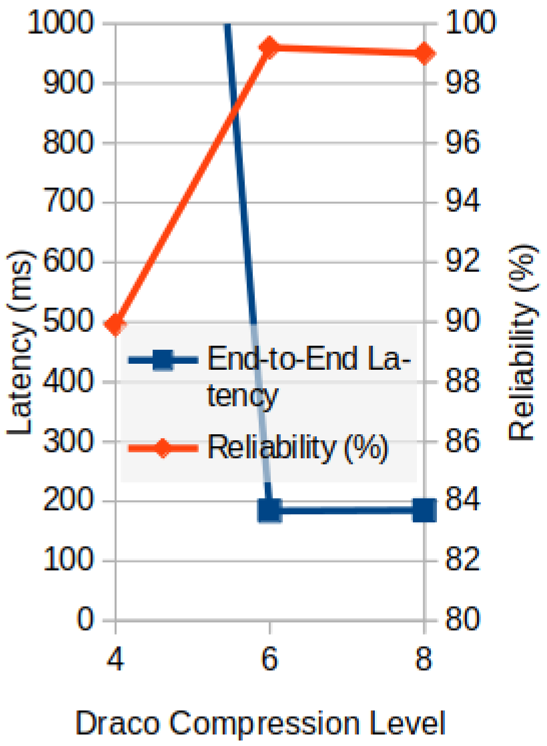


- Very effective in reducing frame size, but not latency
- Linear reduction if we drop 50% of points

Two compression threads

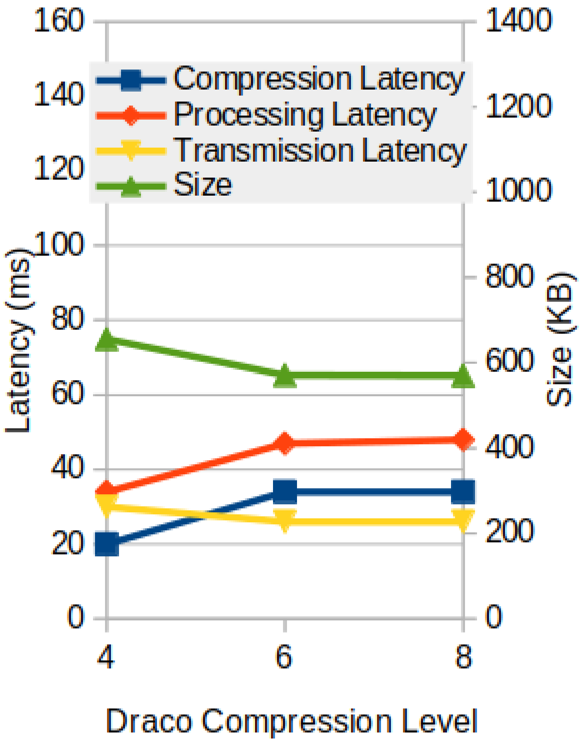


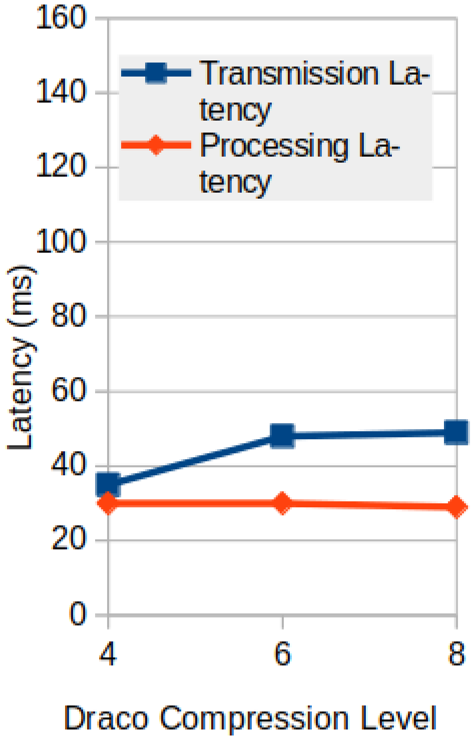


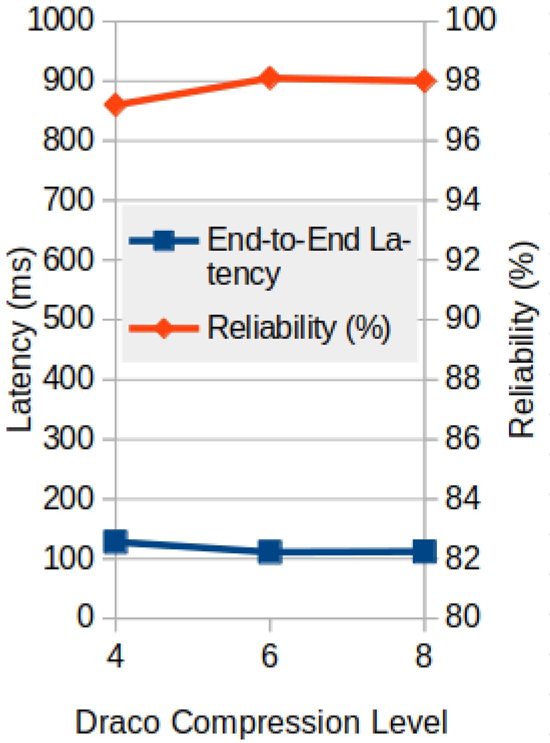


- Linear reduction in processing latency
- Compression efficiency reduced

Two threads and 50% fewer points

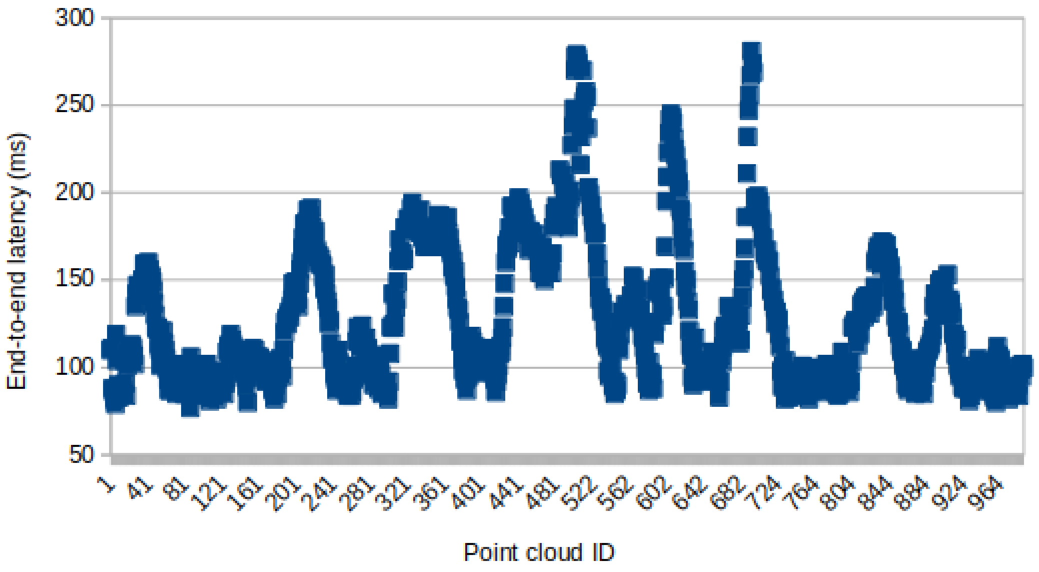






- Can actually reach 30 fps @ 20 MBps
- Latency around a bit more than 100 ms

Frame delay for last scenario



- At least 80 ms of latency
- High variance, around 40 ms
- Too close to channel capacity!

Conclusions & Future Work

What’s next?

- TENeMP project has concluded
  - PC streaming close to feasible
  - But quality needs to be sacrificed
  - More (and better) tricks are possible
- AViD-NMP project has started
  - Adds SFU/MCU for multiparty sessions
    - Compose a new scene
    - Reduce quality as needed
  - Render PC to 3D video
    - Project PC to 3D image
    - Exploit video compression

SPIRIT

6GXR

Thanks!

xgeorge@aueb.gr

EUROXR Winterthur 2025