Enabling Time-Critical Application over 5G by Managed Latency with L4S

Intel MeetUp Dec 8, 2022

Torbjörn Sölve, Manager Radio Access, DU Networks Ericsson
Dominik Schnieders, Head of Edge Computing, Deutsche Telekom

L4S - **L**ow **L**atency **L**ow **L**oss **S**calable throughput

Use cases, learnings and examples with application partners

Multiple latency-critical applications in all market domains

AGVs
(Offloading)

Tele-Operated Driving
(Offloading)

Cloud Gaming
(Offloading)

Multiplayer
(Synchronization)

Cloud AR
(Offloading)

LATENCY-CRITICAL USE CASES

Drones
(Offloading AI)

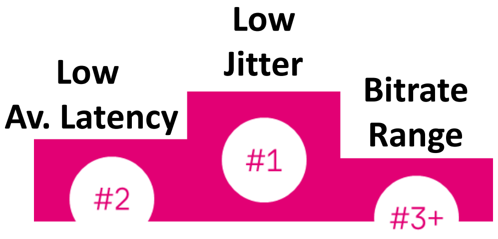
Automotive
(Synchronization)

Video Conferencing
(Offloading)

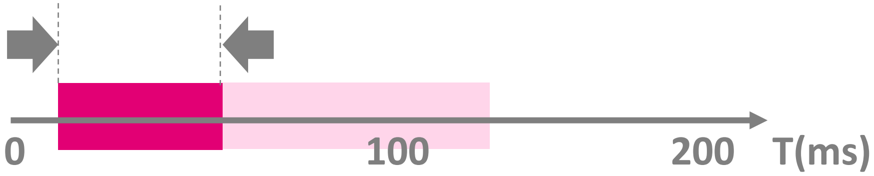
AR Occlusion
(Offloading)

Cloud VR
(Offloading)

Requirements of latency-critical applications to the operator networks

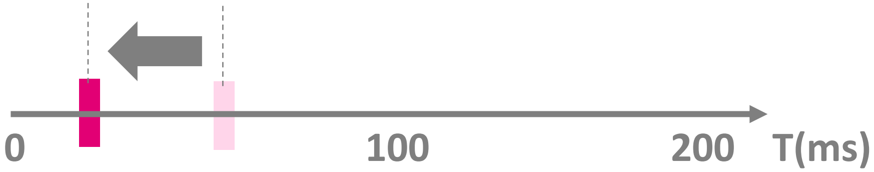


Variation of Latency / Jitter



- **Stability of latency most important** providing **low jitter** and **upper bounded latency** with high percentile.

Average Latency

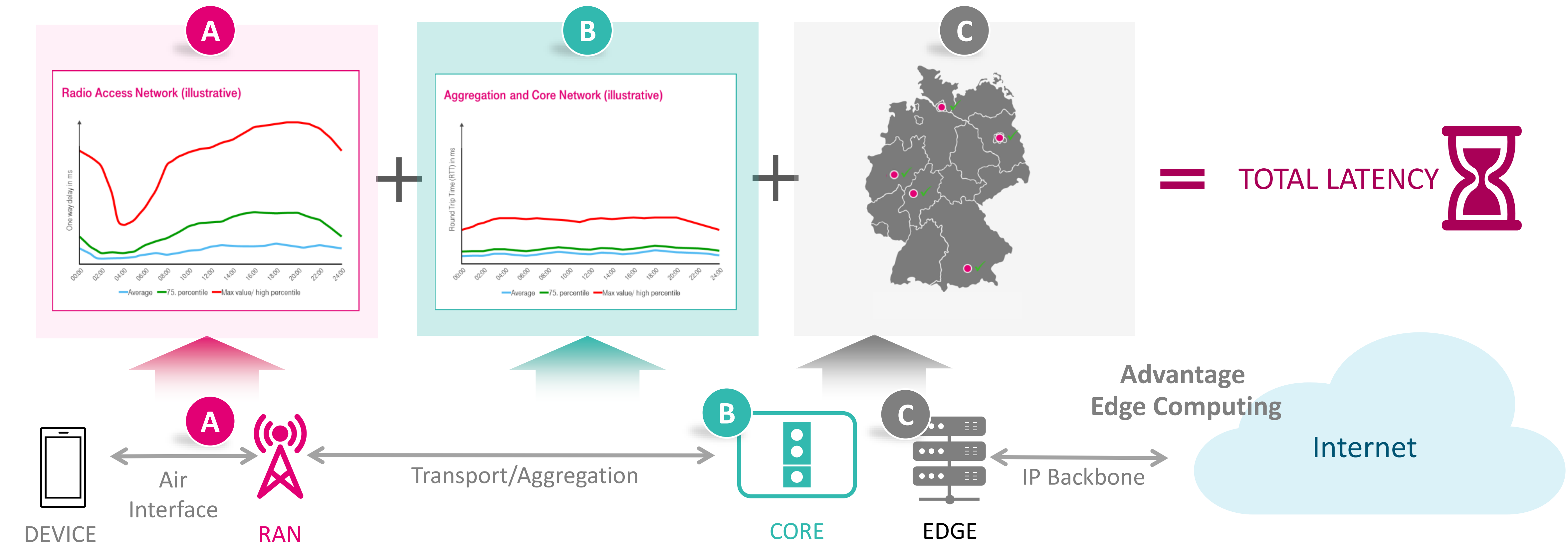


- **Lower average latency** provides **better average reaction time of application**.
- But most applications **don‘t require single digit average latency**.

Bitrate

- Almost all latency applications can **adapt their bitrate** over a significant range.
- The **higher the bitrate** the **better the quality**.

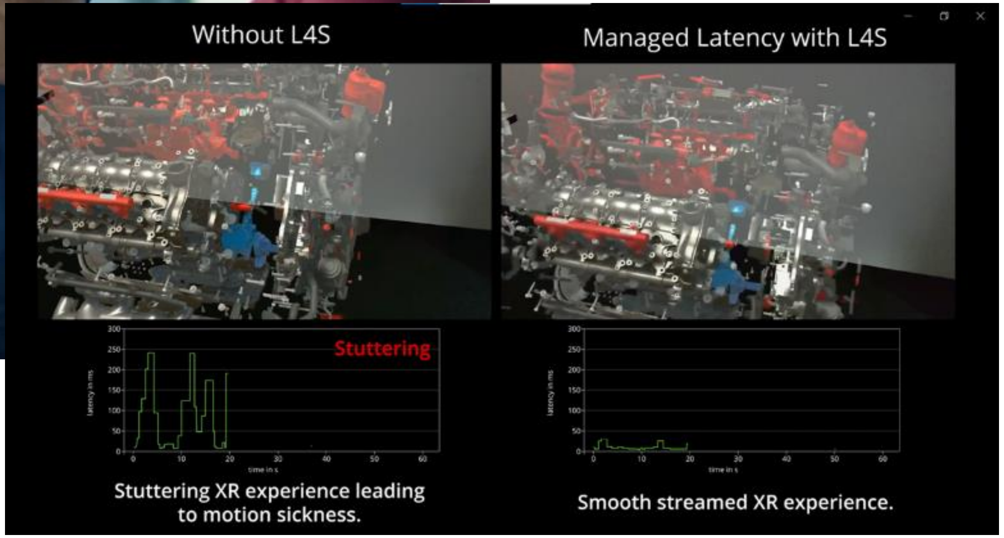
It’s important to optimize the entire latency chain of the network with Edge Computing, Radio Access Network and Transport / Core Network



Edge computing as Intel® Smart Edge and/or Intel® Smart Edge Open as well as improving the latency on the air interface (RAN) play the decisive role to unlock latency-critical applications

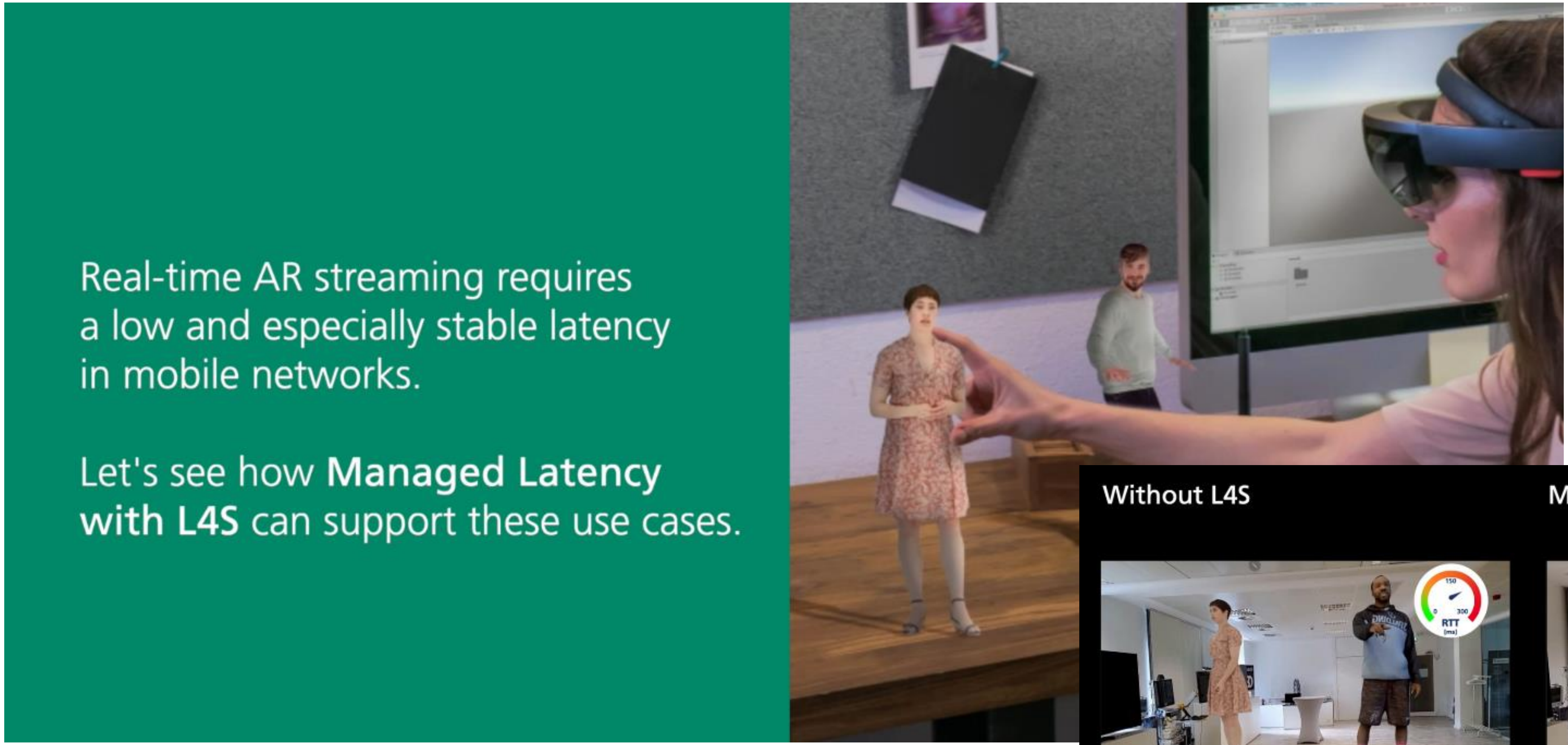
Managed Latency with L4S
AR Streaming for industry with Holo-Light

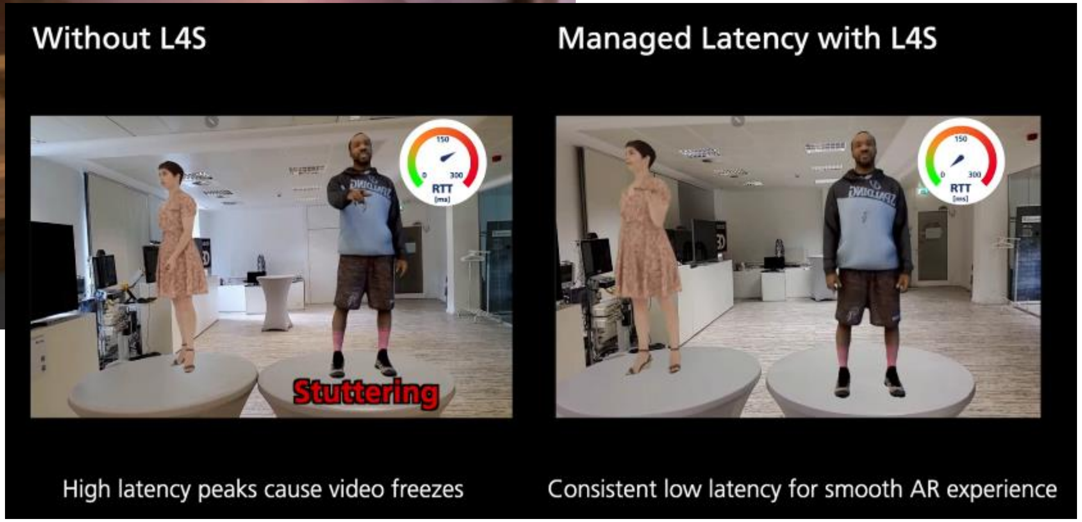
HOLO-LIGHT



Link to video: https://magentacloud.de/s/2fdLsYoHMNcdjiL

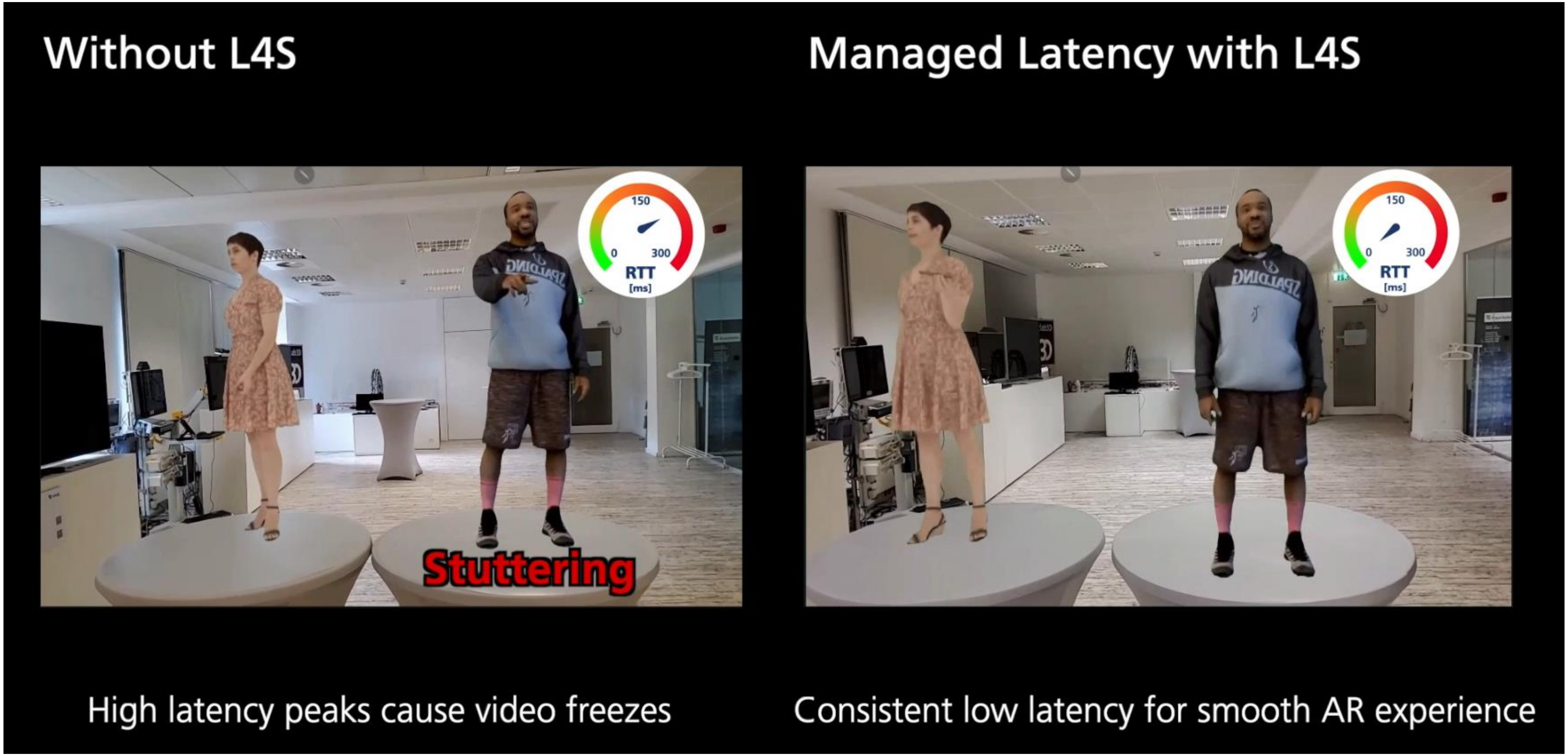
Managed Latency with L4S
Volumetric Video with Fraunhofer



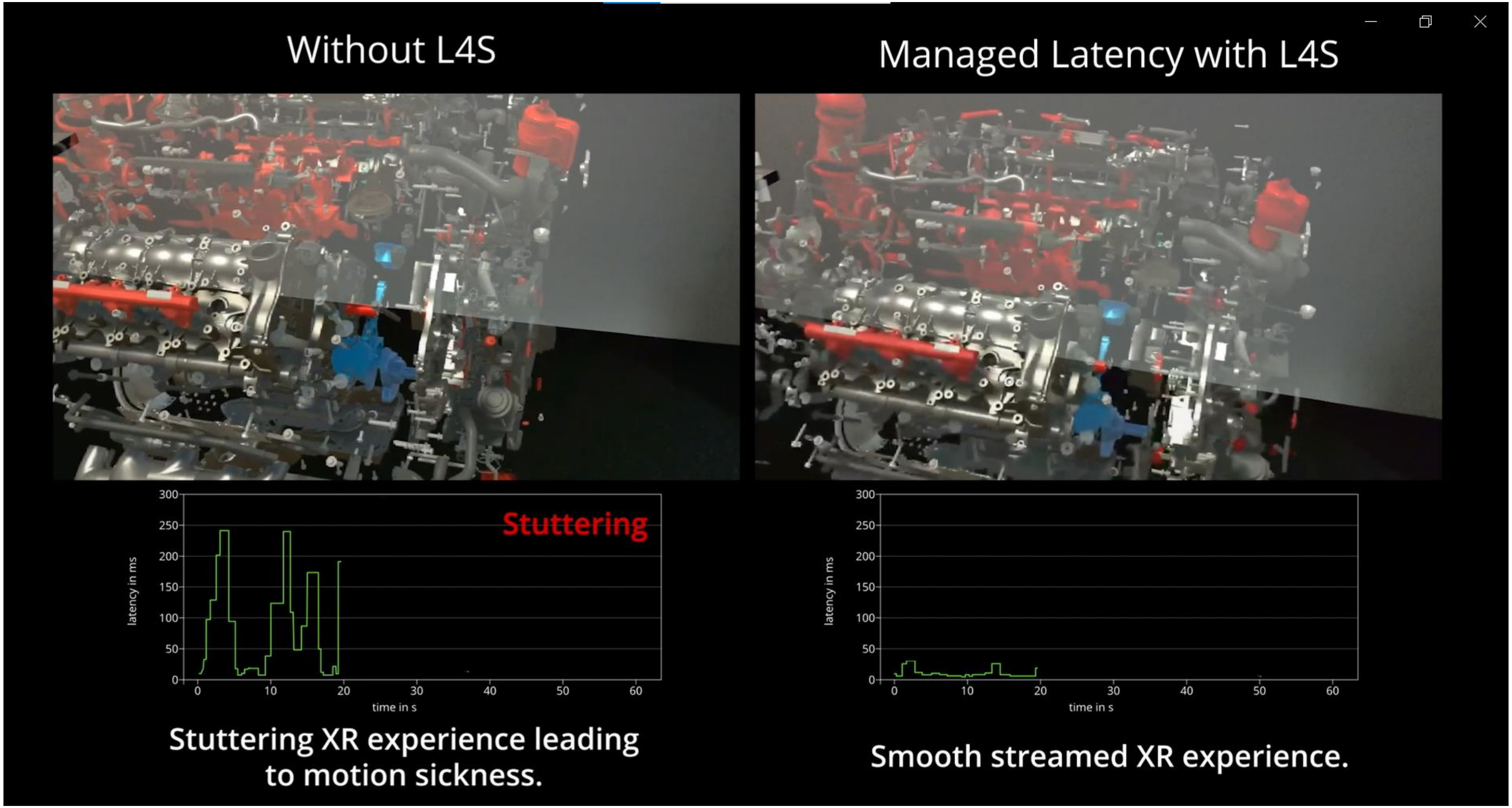


Link to video: https://magentacloud.de/s/BHprb5xxymxYa5A

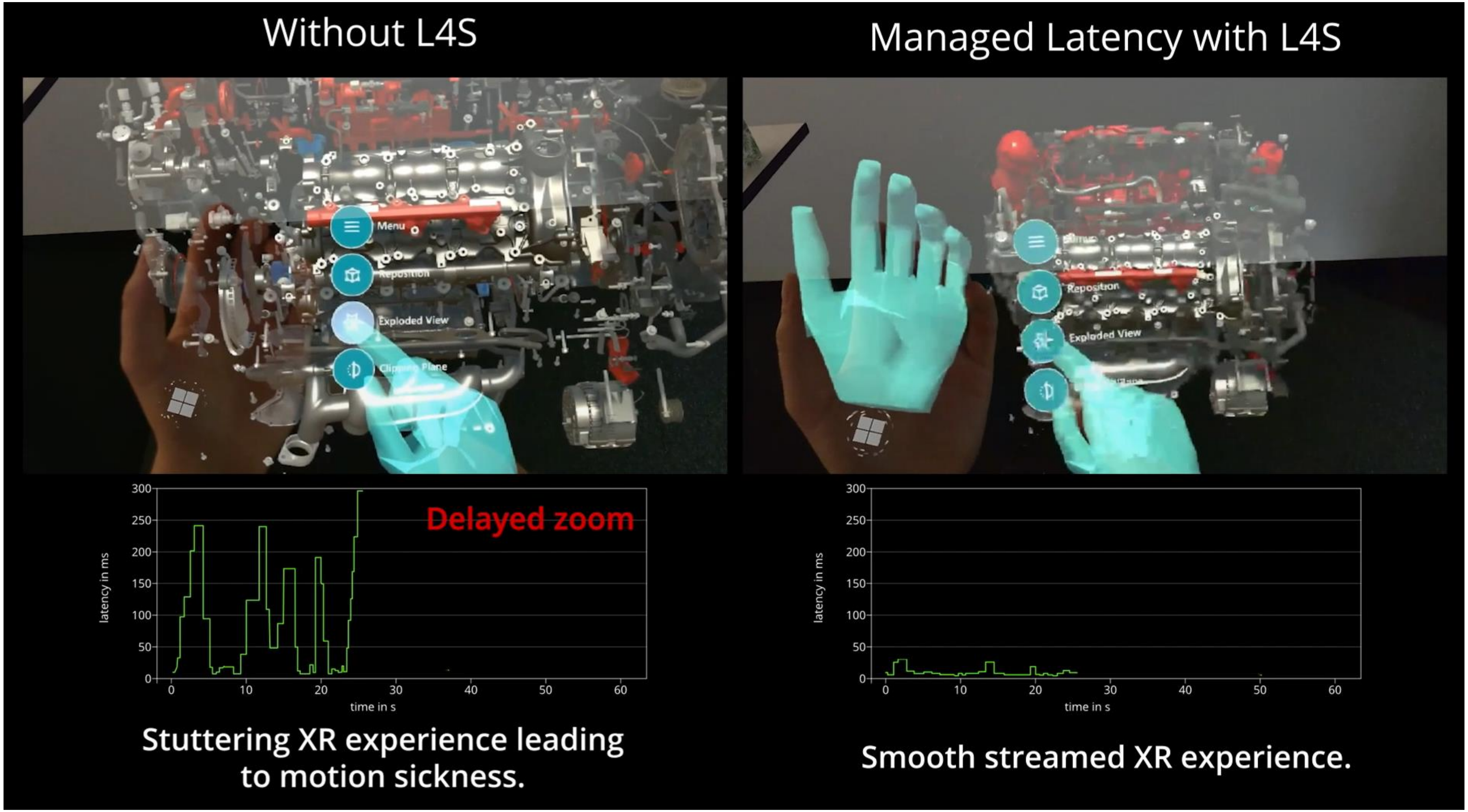
Benefit with L4S with use case examples



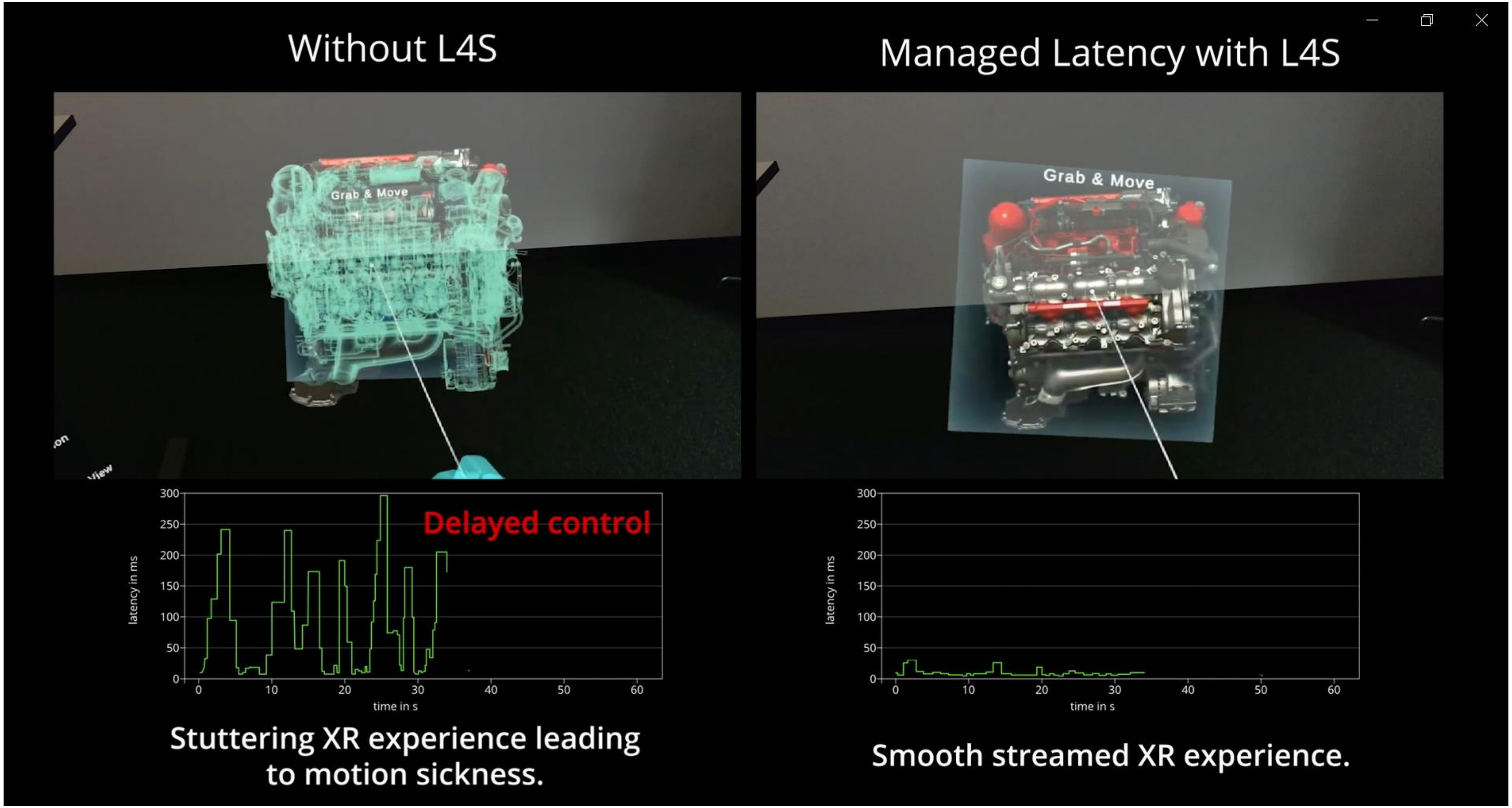
Benefit with L4S with use case examples



Benefit with L4S with use case examples



Benefit with L4S with use case examples



Benefit with L4S with use case examples

Without L4S

(active audio source)

High latency peaks cause video and audio freeze

Managed Latency with L4S

(muted audio source)

Consistent low latency for smooth AR experience

Deep-dive in L4S

Introducing Ericsson Time-Critical Communication for 5G networks

Time-Critical Communication toolbox utilizing a holistic and step-wise approach to remove delay, interrupts, and lags

A wide range of consistent low latency capabilities (50ms-1ms) for diverse time-critical use cases

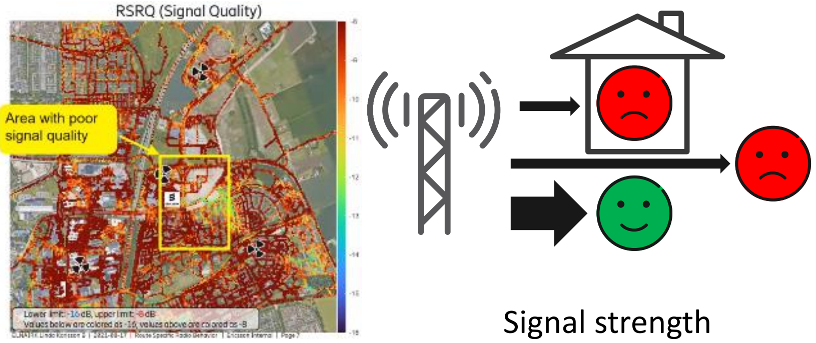
Varying guarantee levels (99,9-99,999% reliability), depending on the application need

Addressing major causes of latency:

- Congestion
- Radio environment
- Mobility
- Standards/Protocols
- Power Saving
- Network Topology

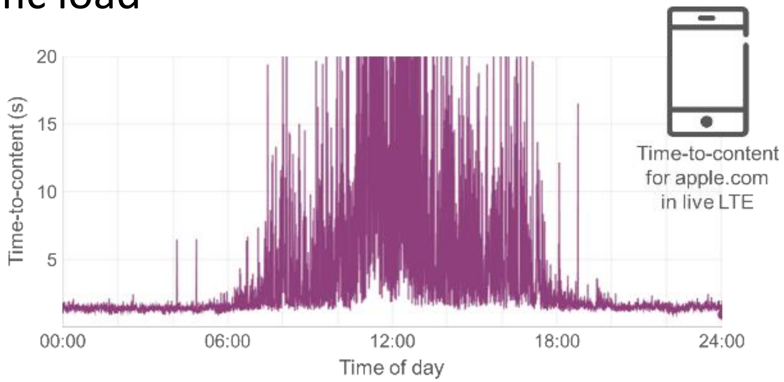
Problem statement from a radio perspective

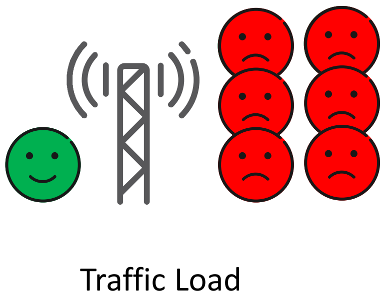
Signal strength/ Coverage



Can only be mitigated with network upgrades/investments

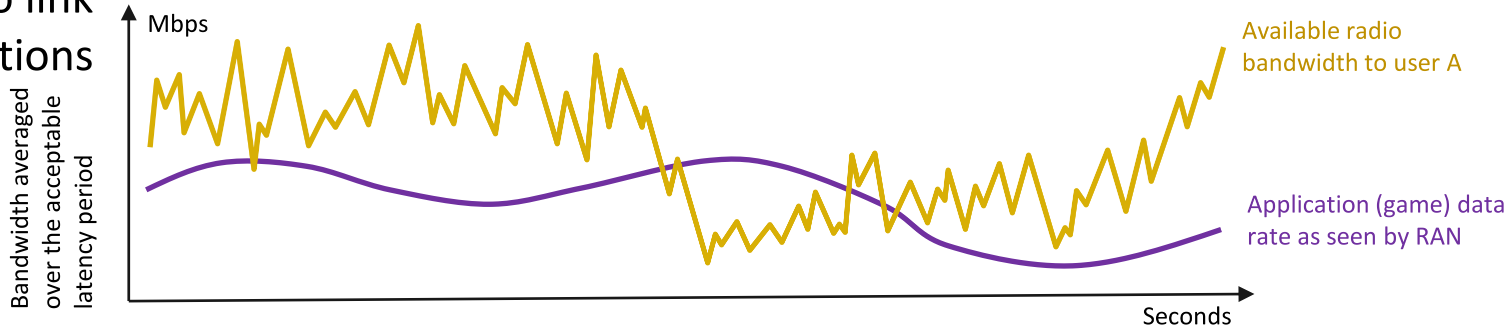
Traffic load





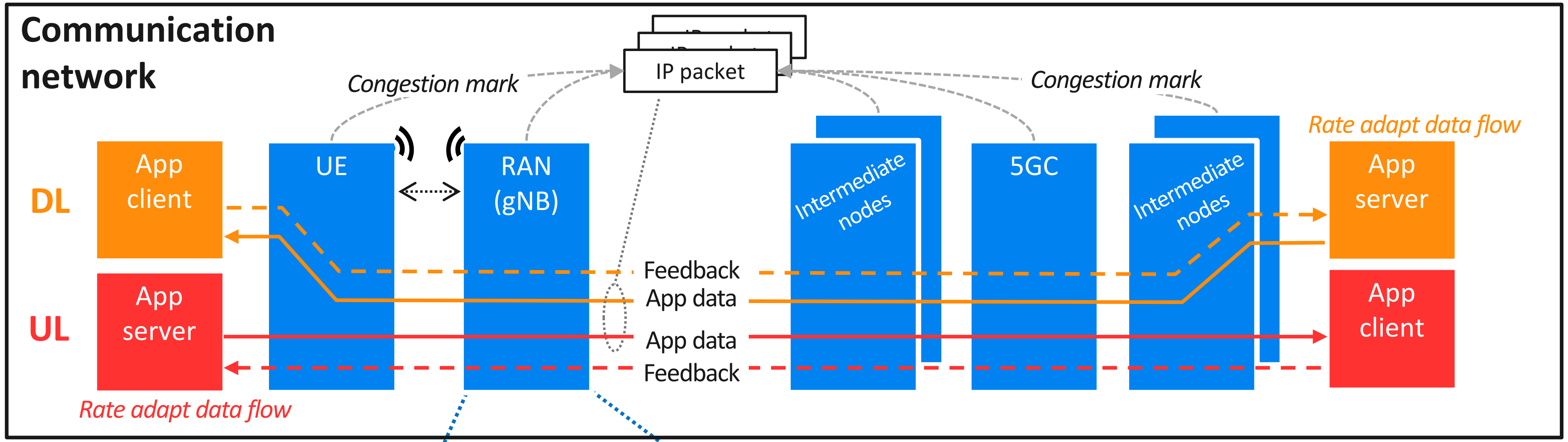
Can technically be mitigated with QoS/Prio and improved capacity. Only a certain part of traffic can be prioritized to get a meaningful improvement.

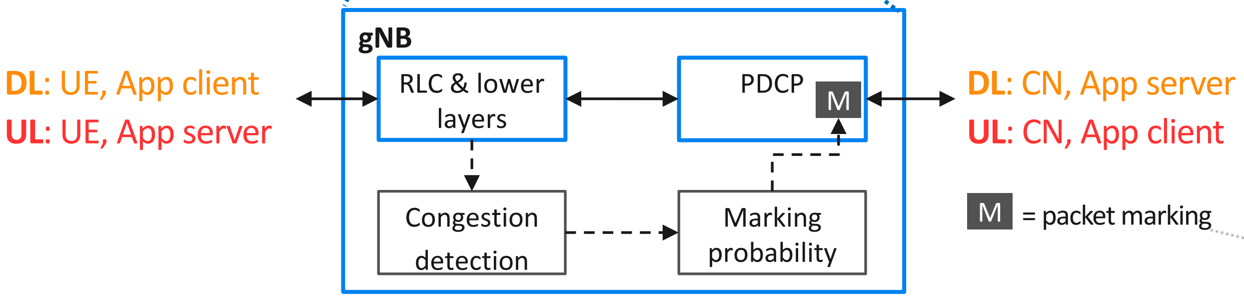
Radio link variations



This results in an individual-user-bandwidth that WILL vary over time and geography, and though some variations can be mitigated in RAN, bandwidth variations will be the norm.

L4S supported in 5G RAN, overview





L4S essentials:

- Marking of E2E application IP data flow
- Congestion experienced (CE) marking in the forward direction
- Scalable rate adaptation in server based on congestion level feedback from app client
- RAN support ➔ proactive measures
- Function allocation of RAN packet marking constrained to PDCP

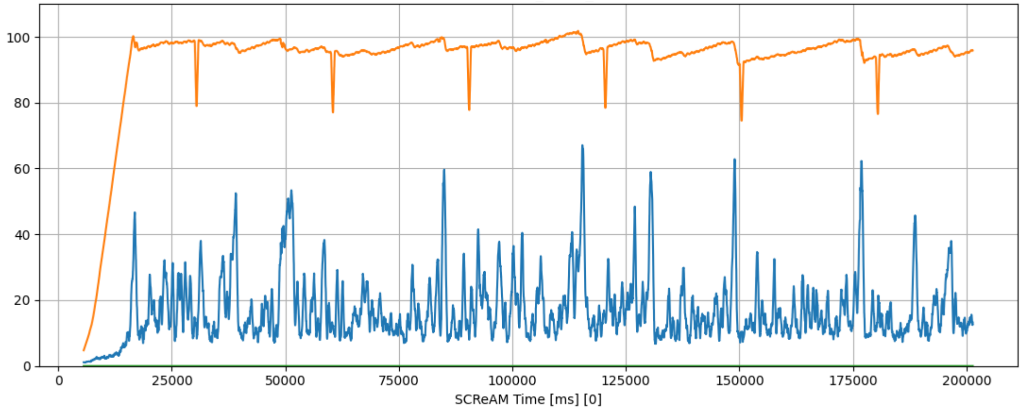
IP header, ECN bits

Non-ECT, 00	Not ECN-capable
ECT-1, 01	L4S-capable transport
ECT-0, 10	ECN-capable transport
CE, 11	Congestion experienced

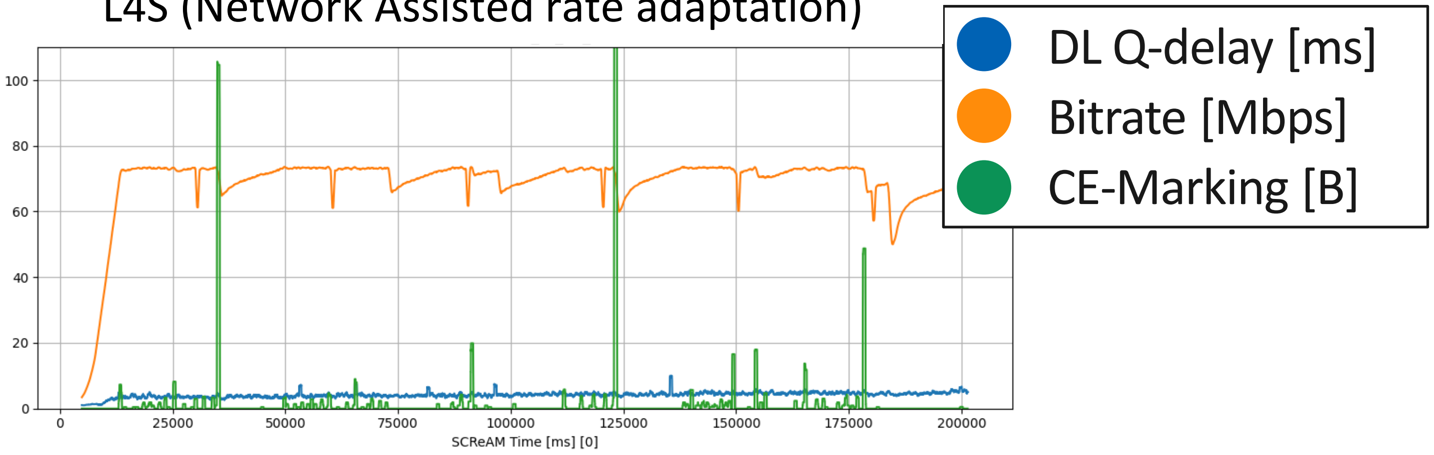
Results

Varying attenuation, fast fading and background traffic

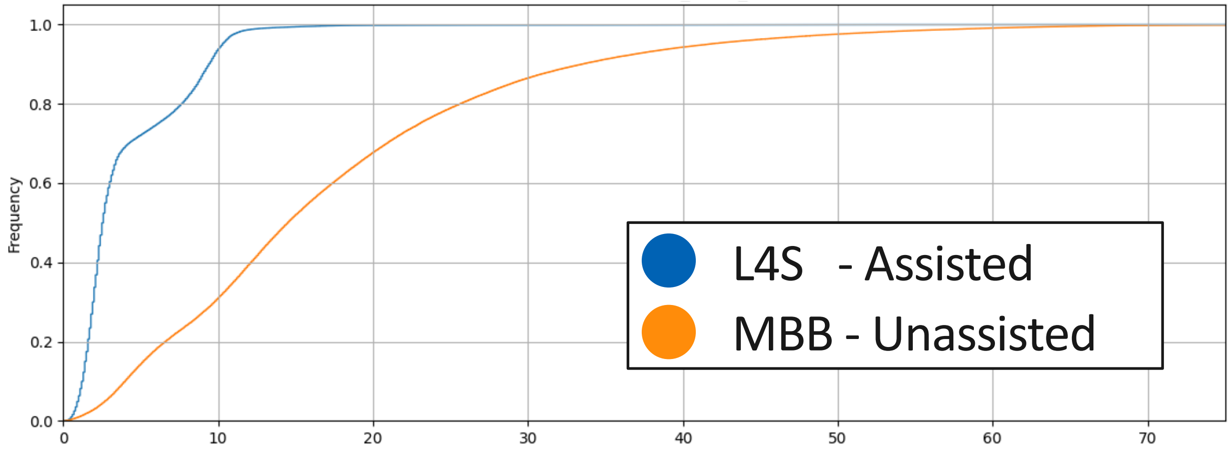
MBB (Unassisted rate adaptation)



L4S (Network Assisted rate adaptation)



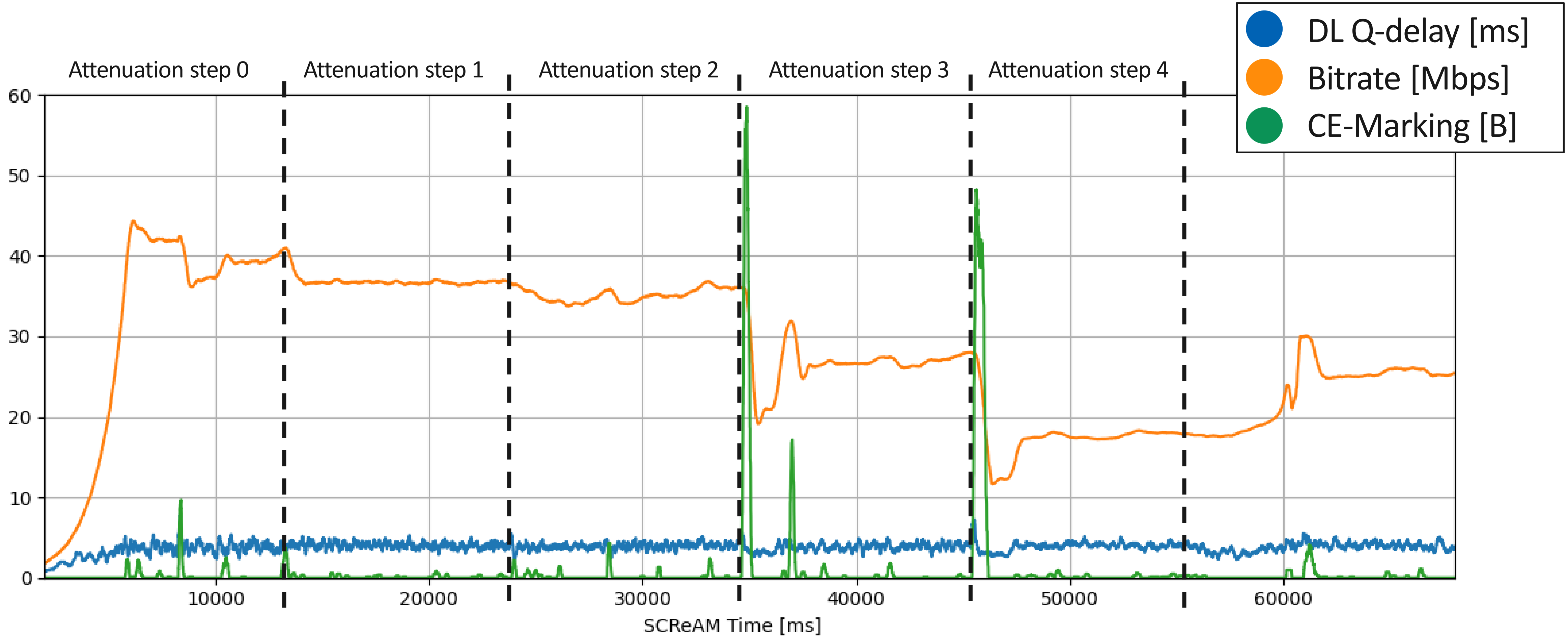
DL Queue Delay



- Large improvements in DL latency
- Maintained high throughput
- Very promising results in lab & field tests

Additional L4S lab results

Varying attenuation, fast fading and background traffic



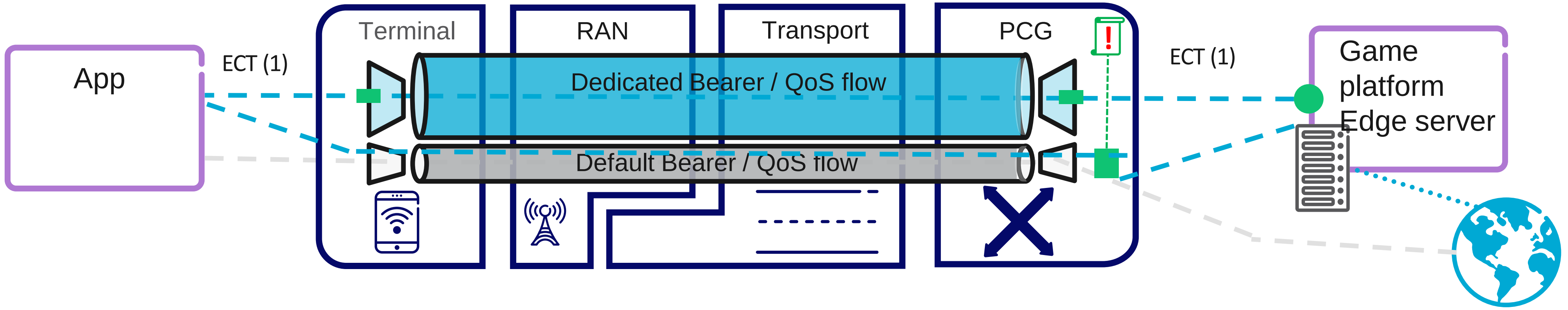
Dedicated QoS-Flow/bearer for Latency-critical traffic

Why a dedicated QoS-flow/bearer:

- Separate queue for latency-critical traffic
- Differentiation based on subscription
- Optional: provide QoS/priority
- Tune of functions for latency, not throughput

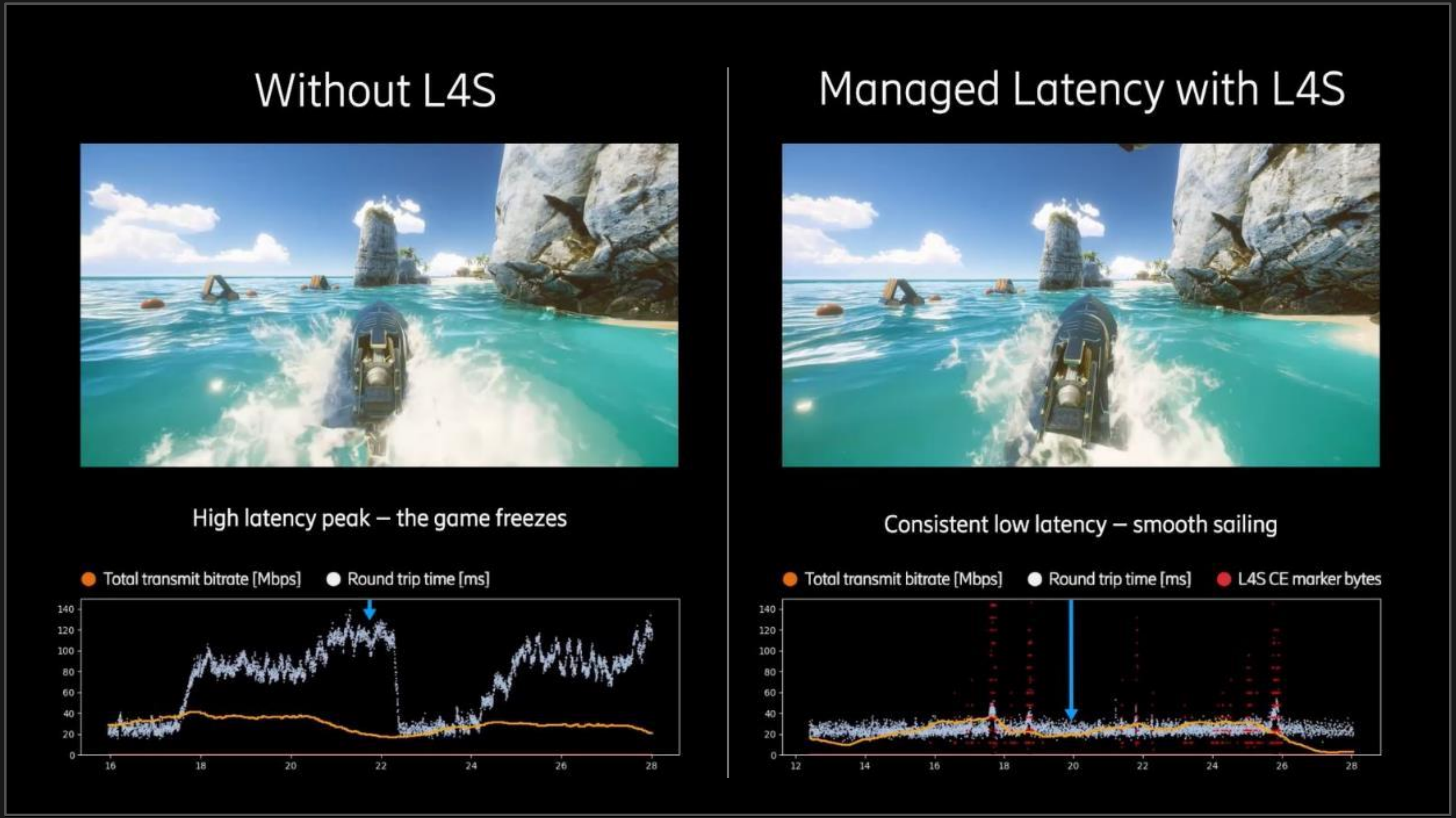
How to use dedicated QoS-flow/bearer:

- Traffic filters to map latency-critical traffic to QoS-flow
- Identify traffic by L4S capability (ECN bits)/IP address
- Subscription control via Packet Control Function
- Dedicated QoS-flow setup when traffic detected
- In special cases: may use Network API to request QoS-flow



Example: L4S-enabled Time-Critical Communication for cloud gaming

L4S feature for fast rate adaptation can significantly improve the performance of time-critical, high-rate applications on 5G networks



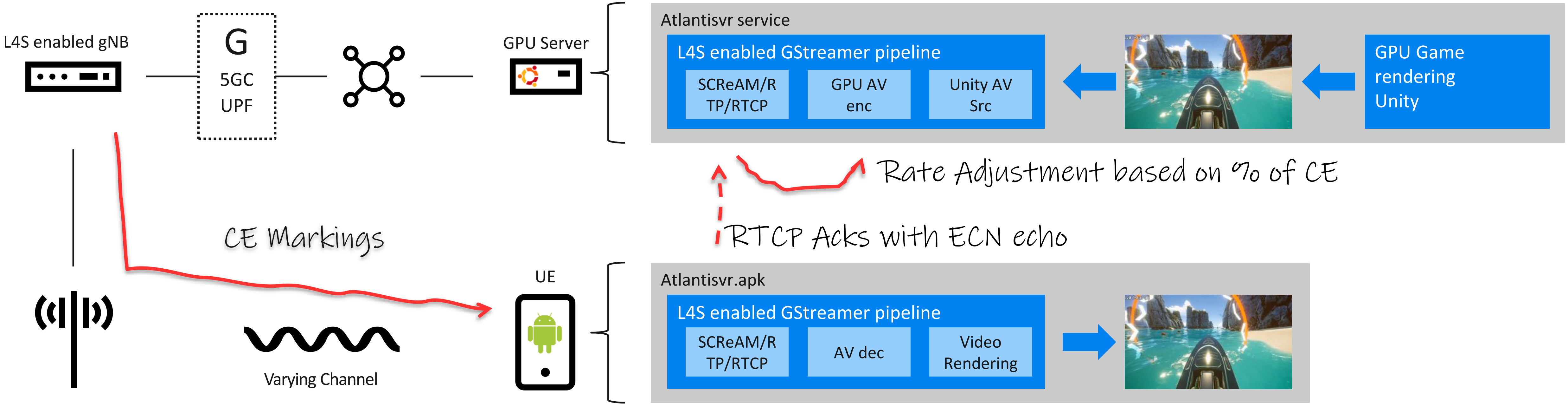
L4S: **L**ow-**L**atency, **L**ow **L**oss, **S**calable throughput

Info links:

- Ericsson and DT demo 5G low latency feature
- Time-Critical Communication makes each moment count - Ericsson
- White paper

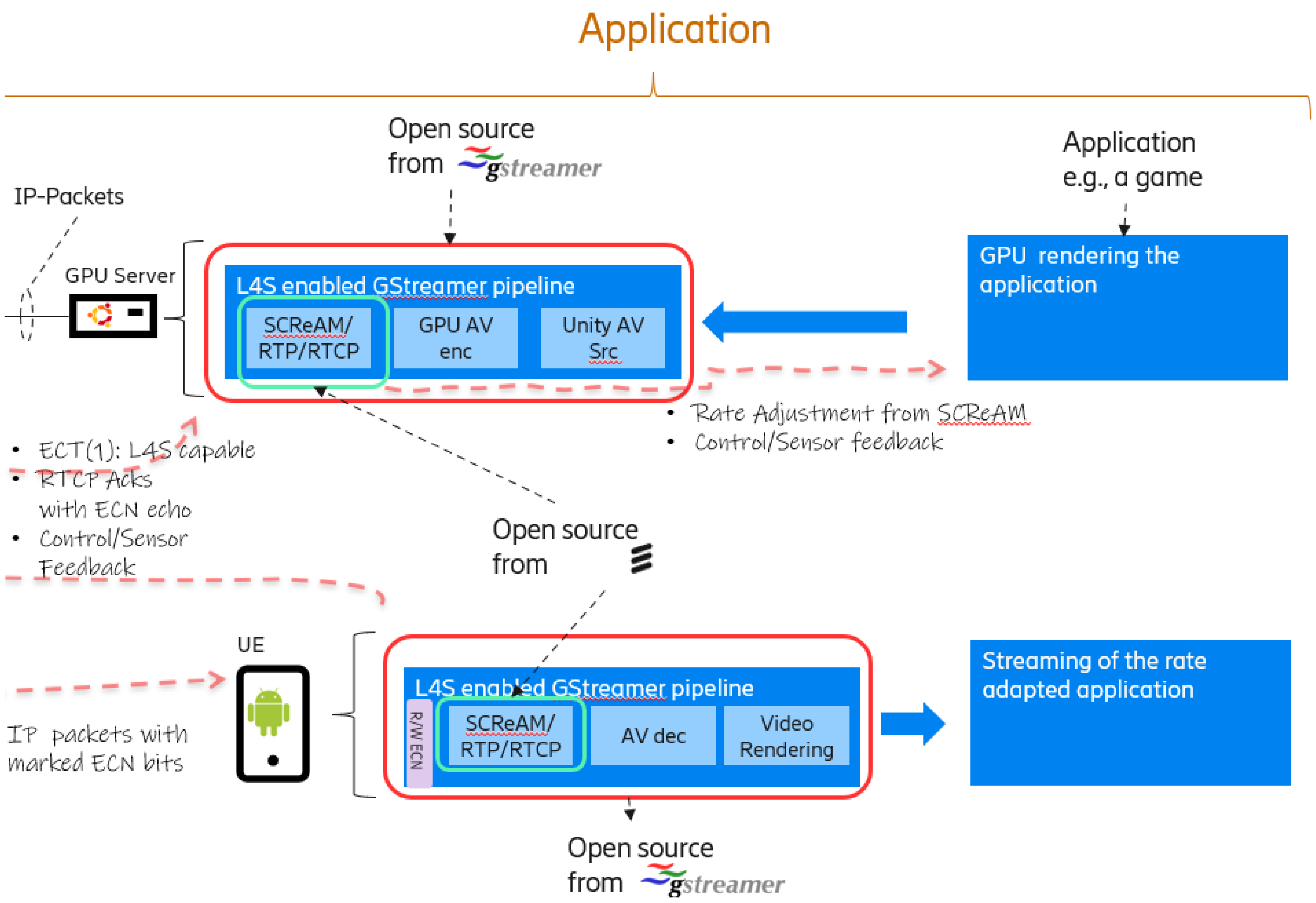
How to implement a rate adaptive application with L4S

Boat Attack Implementation Overview



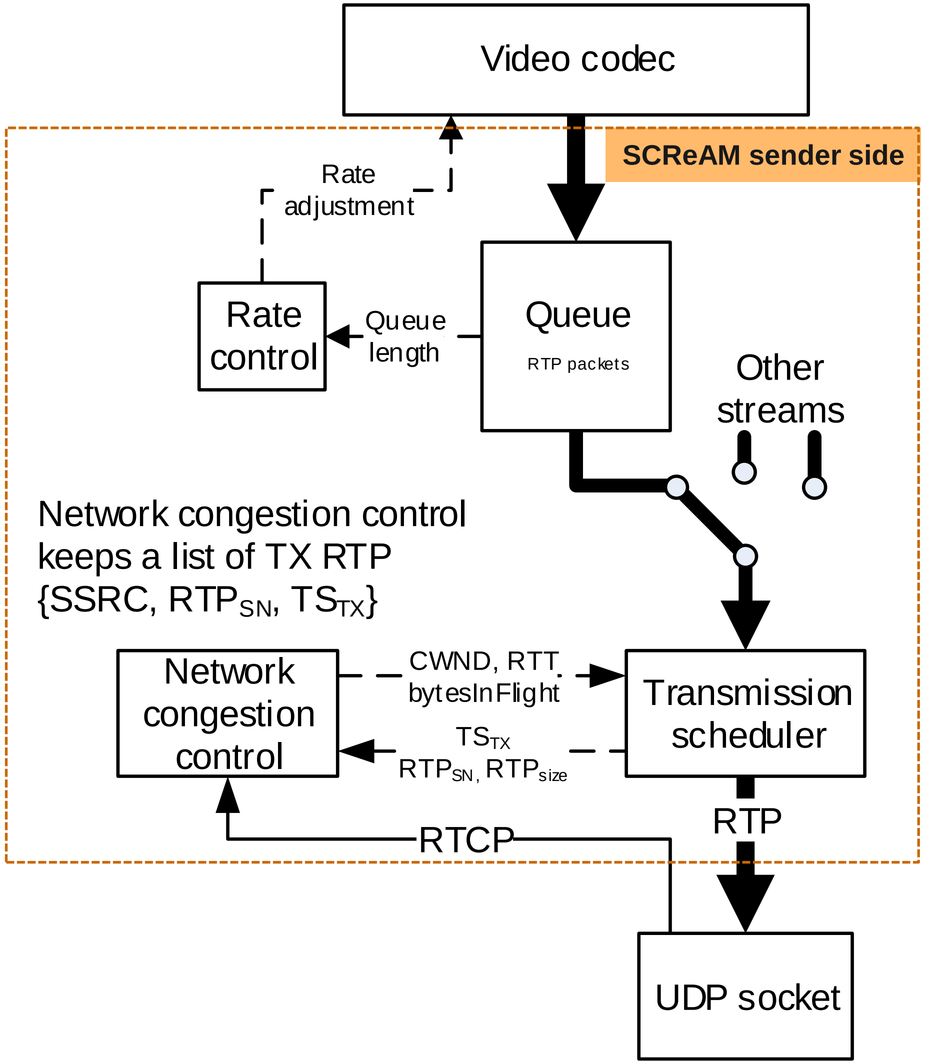
SCReAM
How and where ?

- The sender side
  - Congestion control = Keep network queue short
  - Rate control = Control bitrate of source(s)
- The receiver side
  - Generates feedback
- **Feedback** over RTCP (RFC8888)
  - Timestamp of received packets
  - ECN echo (for L4S)



SCReAM (RFC8298) in one page

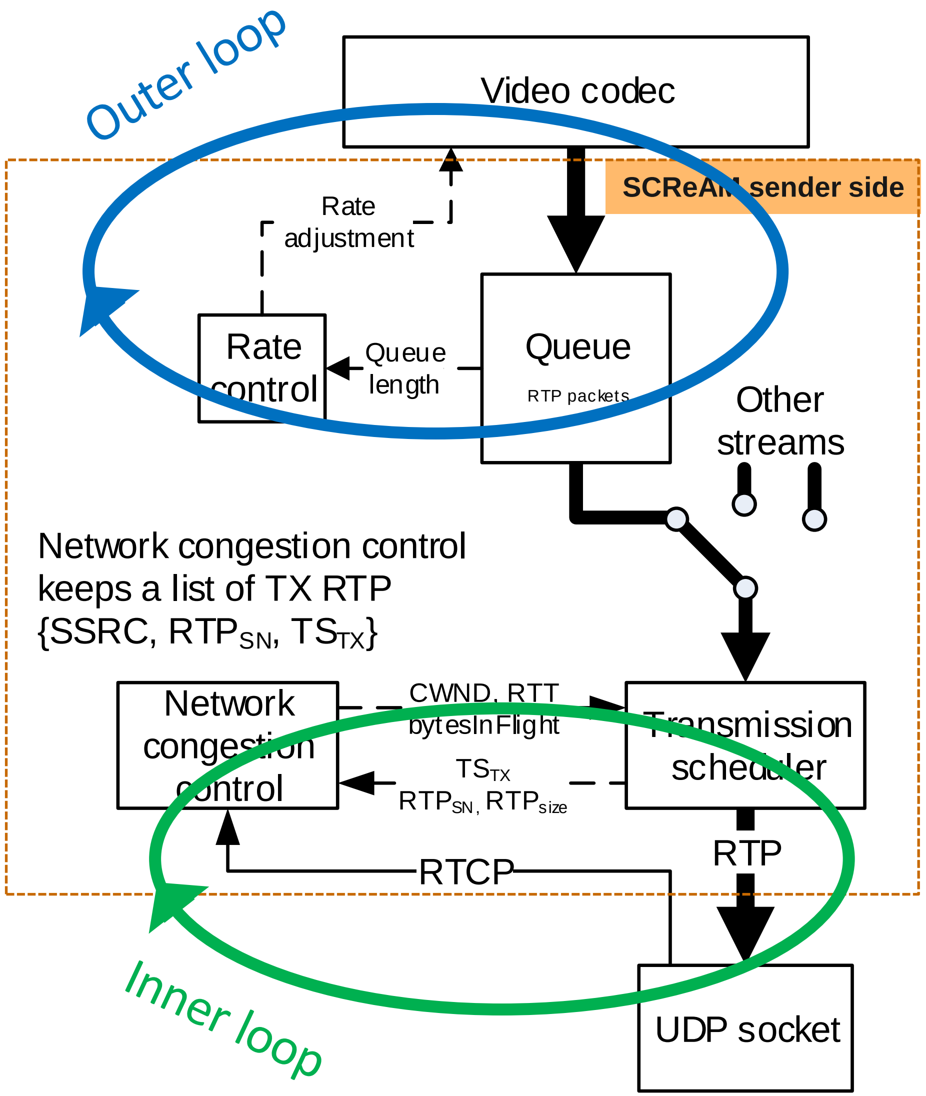
- SCReAM = **S**elf-**C**locked **R**ate **A**daptation for **M**ultimedia
  - Window based congestion control → like **TCP** but **without the retransmissions**
  - Algorithm reacts on delay, packet loss **and ECN/L4S**
  - RTP packets can be queued up already in sender
- Developed since 2014
  - Design goal : Good performance for wireless access (LTE, 5G)
  - Originally intended for WebRTC
  - Today : focus on high bitrate consumer and industrial applications
    - XR, cloud gaming, remote control
- Most RTP media can be congestion controlled
  - Video, Audio, Haptics, Motion-JPEG, Hologram, Smell....
- Multi-stream handling with prioritization
- Available as open source
  - Operating range : ~50kbps .. >100Mbps
  - https://github.com/EricssonResearch/scream (gstreamer plugin available)



ECN : Explicit Congestion Notification
L4S : Low Latency Low Loss Scalable throughput

Control Basic of SCReAM

- Outer loop
  - The target media bitrate is the “desired” rate by which to produce the media packets to the RTP Queue (keep the Queue low)
  - The target media bitrate governing the “media creation rate”
  - The target media bitrate increase opportunistic when not decreasing due to the rate adjustment
- Inner loop
  - The RTT & Congestion Window are the parameters governing the rate by which to take packets out of the RTP queue and send them to the client.
  - Depend on the radio channel quality



Links to the open source SW from Ericsson

- The link below is to the open source github that includes
  - The scream code with basic adaption to Gstreamer
  - Link to open source of SCReAM https://github.com/EricssonResearch/scream

- To perform the handling of the ECN bits in the IP-header by the Client, one recommendation is to explore how this is realized in the Linux based "SCReAM BW test tool".
  - perform the required reading of the ECN bits to be able to create the RTCP feedback to the server see scream/scream_receiver.cpp at master · EricssonResearch/scream (github.com)
  - set the ECN bit in the IP header to confirm the client is L4S capable see scream/scream_sender.cpp at master · EricssonResearch/scream (github.com)

  An adaption of the above functions to the relevant platform will then be needed.

Development and test tools

What is required to use L4S from a developer perspective?

01

Receive feedback from network according to L4S (IETF)

Read ECN bits from IP header via socket

ECN bits will provide you the congestion state of the network

02

Adapt bitrate

Adapt the bitrate / throughput of your application according to the feedback of the network / ECN bits

03

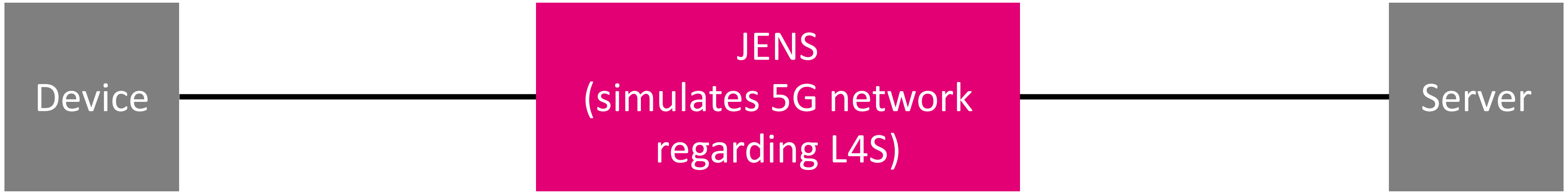
Networks support Managed Latency with L4S

Test networks available

Managed Latency with L4S

- Essential to provide a stable latency / low jitter
- Key enabler for latency-critical applications

L4S simulation tool „JENS“ (Jitter/Latency Network Simulator) for developers



- Tool for developers to develop and test their application with L4S at their home location
- Simulates 5G network behavior regarding L4S by shaping the bandwidth and applying ECN marking in case of congestion
- Acts like a Router between clients and servers (ethernet and wifi)
- Operates between OSI-Layer 2 and 4
- Uses Grafana to present KPIs for latency, bandwidth, link capacity, packet loss, etc.
- Provided via github as package

L4S simulation tool „JENS“ for developers

Testbed for managed Latency with L4S in Cologne

- Testbed for Managed Latency with L4S in Cologne
- Allows on-site testing with 5G network using L4S
- Tests feasible also for scenarios with fully loaded cell, cell edge conditions, attenuation, etc. to investigate the performance of L4S especially in difficult network conditions
- Testing supported by team of Deutsche Telekom

Do you have a **high-bitrate application** that requires a **low and stable latency over a mobile network?** Please contact us and we are looking forward to discuss with you the potential of L4S for your application and support you to try it out with our test tools and test environment.

Contact person: Dominik Schnieders, Deutsche Telekom, dominik.schnieders@telekom.de

THANK YOU

Contact person: Dominik Schnieders, Deutsche Telekom, dominik.schnieders@telekom.de