SCReAM EXPERIMENTS

REMOTE CONTROL OF VEHICLES OVER 4G/5G

Ericsson Research
Network Protocols & E2E Performance
Ingemar Johansson (ingemar.s.johansson@ericsson.com)

Robert Hedman, Olov Sehlin : Luleå University of Technology

THIS IS ABOUT TECHNOLOGY..

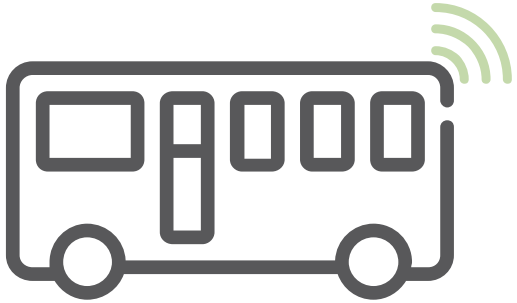
…to go from this...

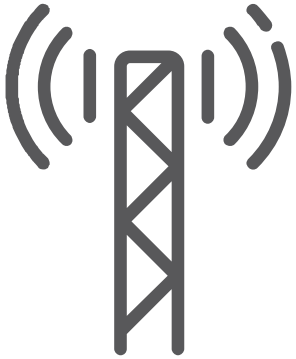
- Frequent packet drops → distorted video image
- Large delay variations → choppy video

...to this

- Reduced packet loss → good video image
- Reduced delay variations → stable video

PROBLEM



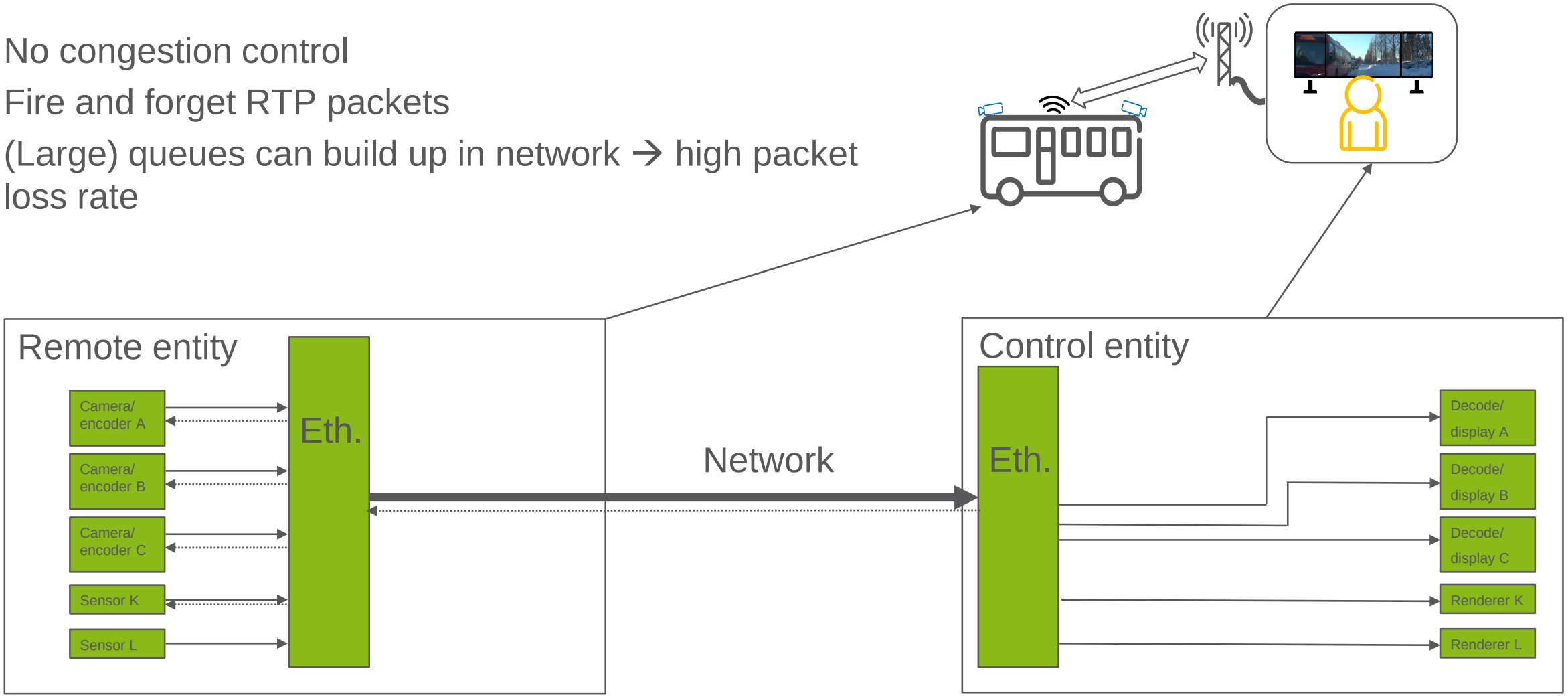


- Remote control applications generally require high quality video feedback
  - Multiple cameras needed for wide angle view and good ergonomics
  - High contrast, high resolution and high frame rate desired for reduced operator fatigue
  - Resulting peak bitrate with many cameras can be in excess of 20-30Mbps
- LTE/5G deployment can not always guarantee high UL bitrate
  - Insufficient coverage
  - High network load – multiple machines/vehicles, competing services

- A remote control solution with video must be rate adaptive
  - A basic requirement for worst case stability
- Various network support enhancements can improve performance further
  - Densification of network
  - QoS, higher service priority in congested cells
  - Explicit Congestion Notification (ECN)

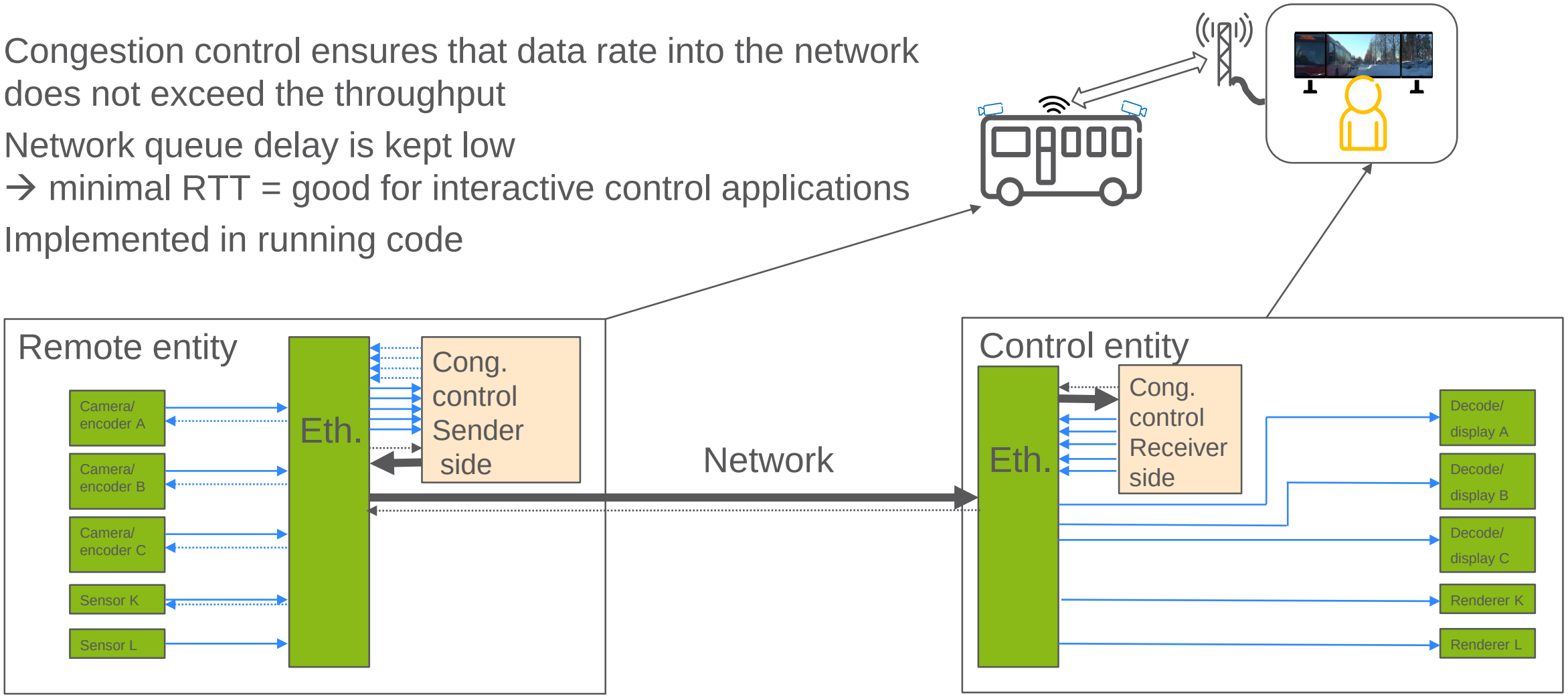
NO CONGESTION CONTROL

- No congestion control
- Fire and forget RTP packets
- (Large) queues can build up in network → high packet loss rate



CONGESTION CONTROL WHERE AND WHY?

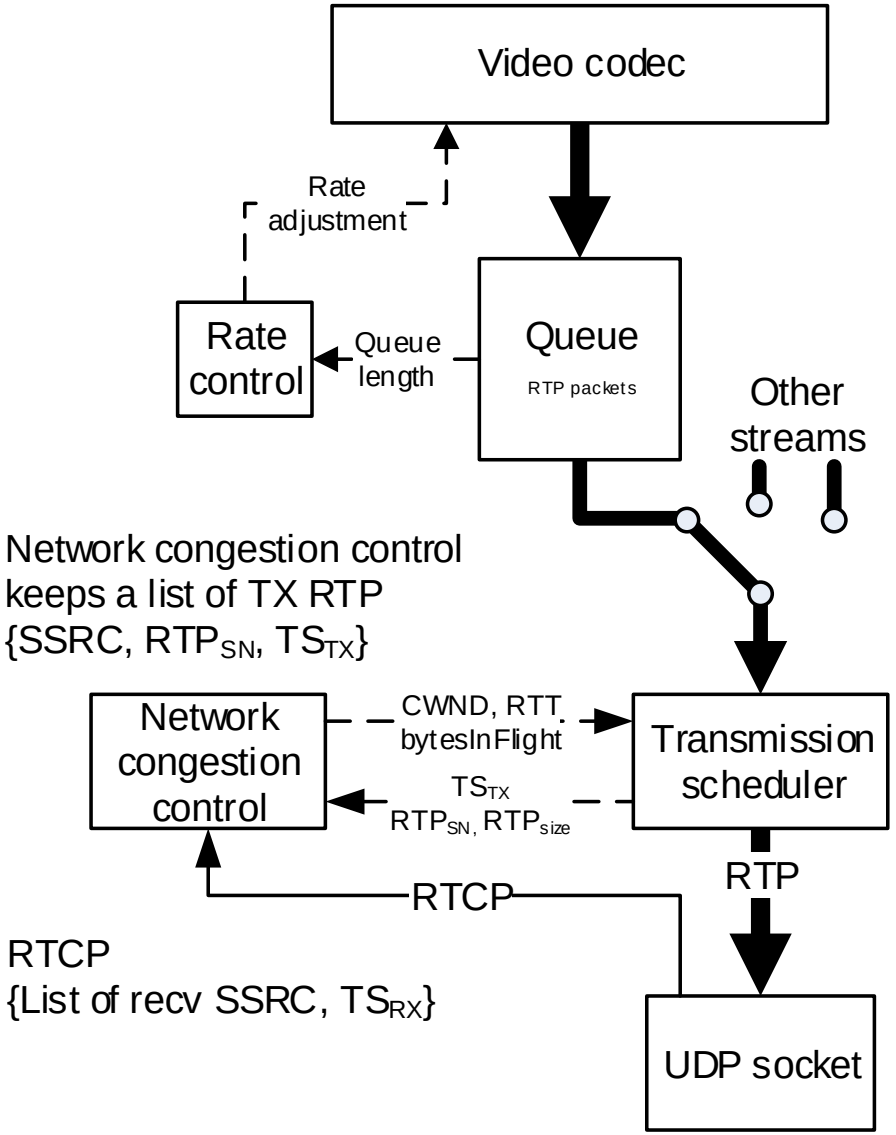
- Congestion control ensures that data rate into the network does not exceed the throughput
- Network queue delay is kept low
  → minimal RTT = good for interactive control applications
- Implemented in running code



SCReAM IN ONE PAGE

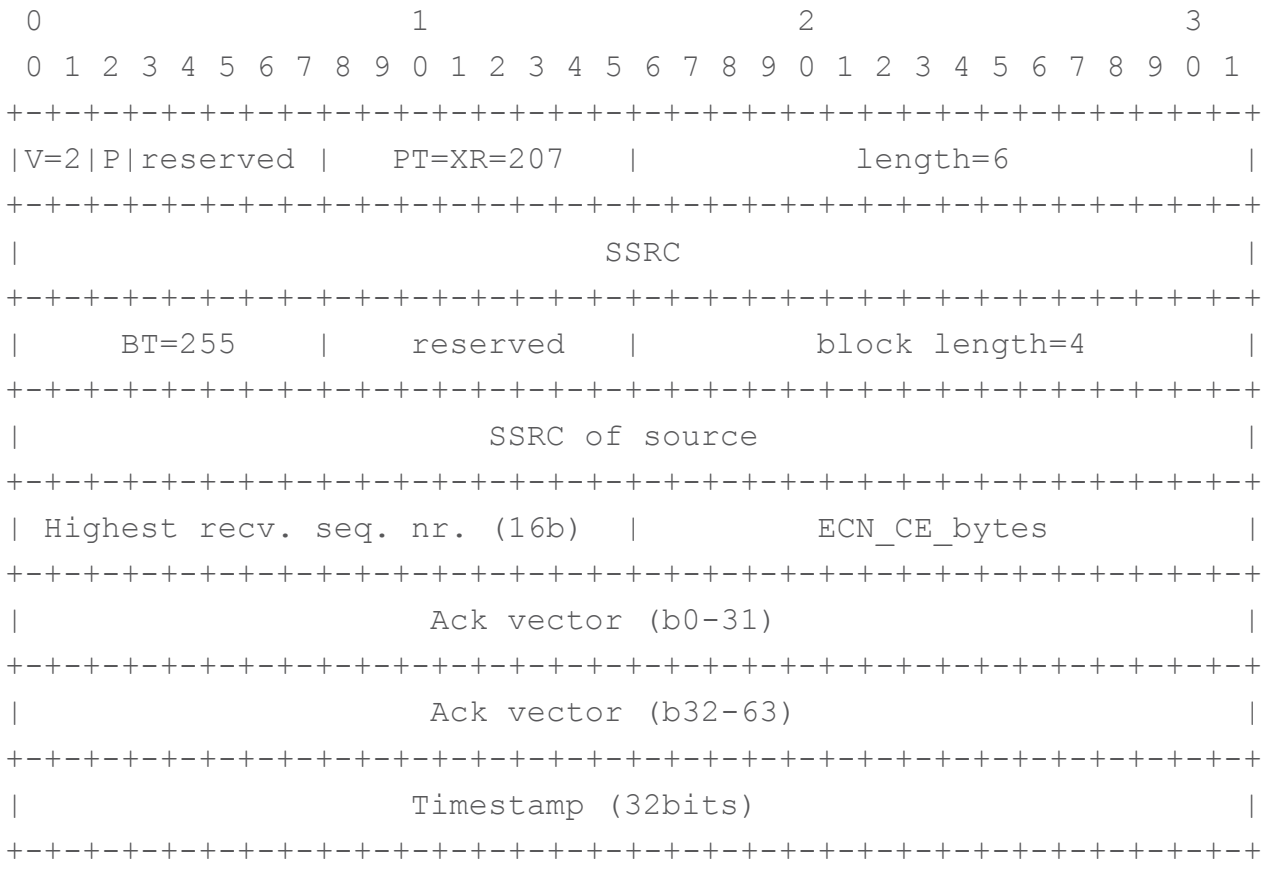
- SCReAM = **S**elf-**C**locked **R**ate **A**daptation for **M**ultimedia
  - Window based congestion control → like TCP but without the retransmissions
  - Algorithm reacts on packet loss as well as delay **and ECN**
  - RTP packets can be queued up already in sender
- Developed since 2014
  - Design goal : Good performance for wireless access (LTE, 5G)
  - In RFC Editors Queue!

- Most RTP media can be congestion controlled
  - Video, Audio, Haptics, Motion-JPEG
- Multi-stream handling with prioritization
- Available as open source
  - Operating range : ~50kbps .. 100Mbps
  - https://github.com/EricssonResearch/scream
  - Ongoing work :
    - **L4S** support
    - GStreamer plugin, student project



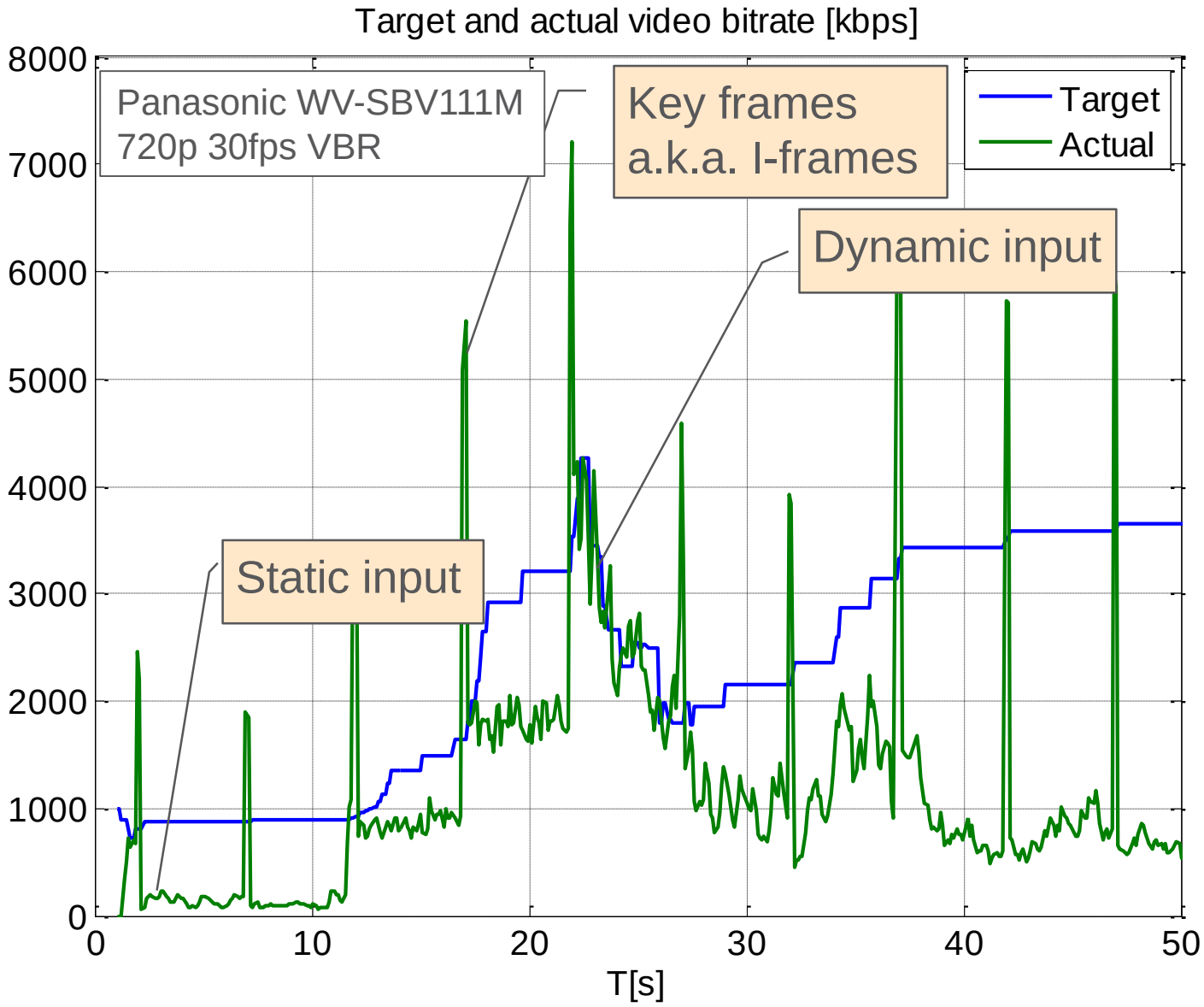
FEEDBACK

- Minimal size feedback message to make SCReAM fully functional
- RTCP XR, BT = 255 (experimental use)
- ECN block reports accumulated sizeof(RTP packets) ECN-CE marked
- Feedback overhead varies with media rate
  - 100kbps ~5%
  - 100Mbps ~2%
- Todo :
  - Replace with **draft-ietf-avtcore-cc-feedback-message**
  - Add support for bundling if more than one stream



VIDEO CODER PROPERTIES

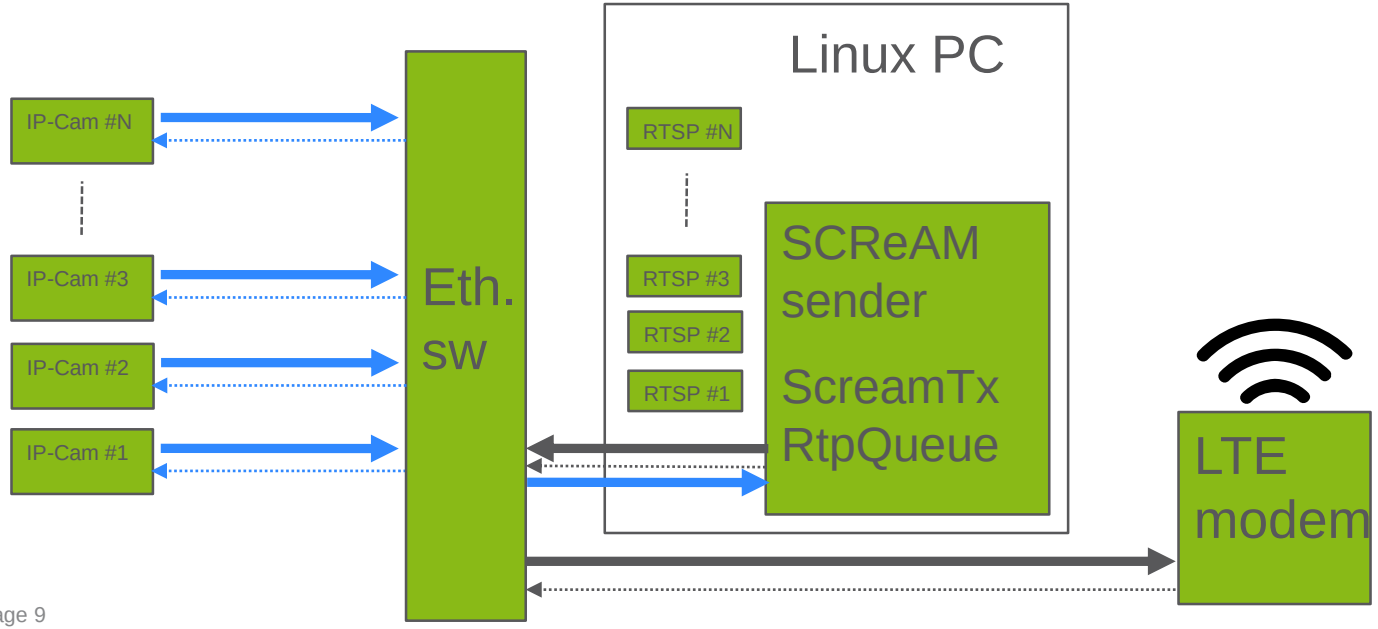
- Video coders are challenging to work with
  - Large bitrate variations
  - Keyframes..
  - Quantizers change on GOP boundaries
  - Don’t expect a constant bitrate from a commercially available video encoder!
- Limited tuning capabilities of HW video coders
- Large impact on design of congestion control



SENDER SIDE VIEW

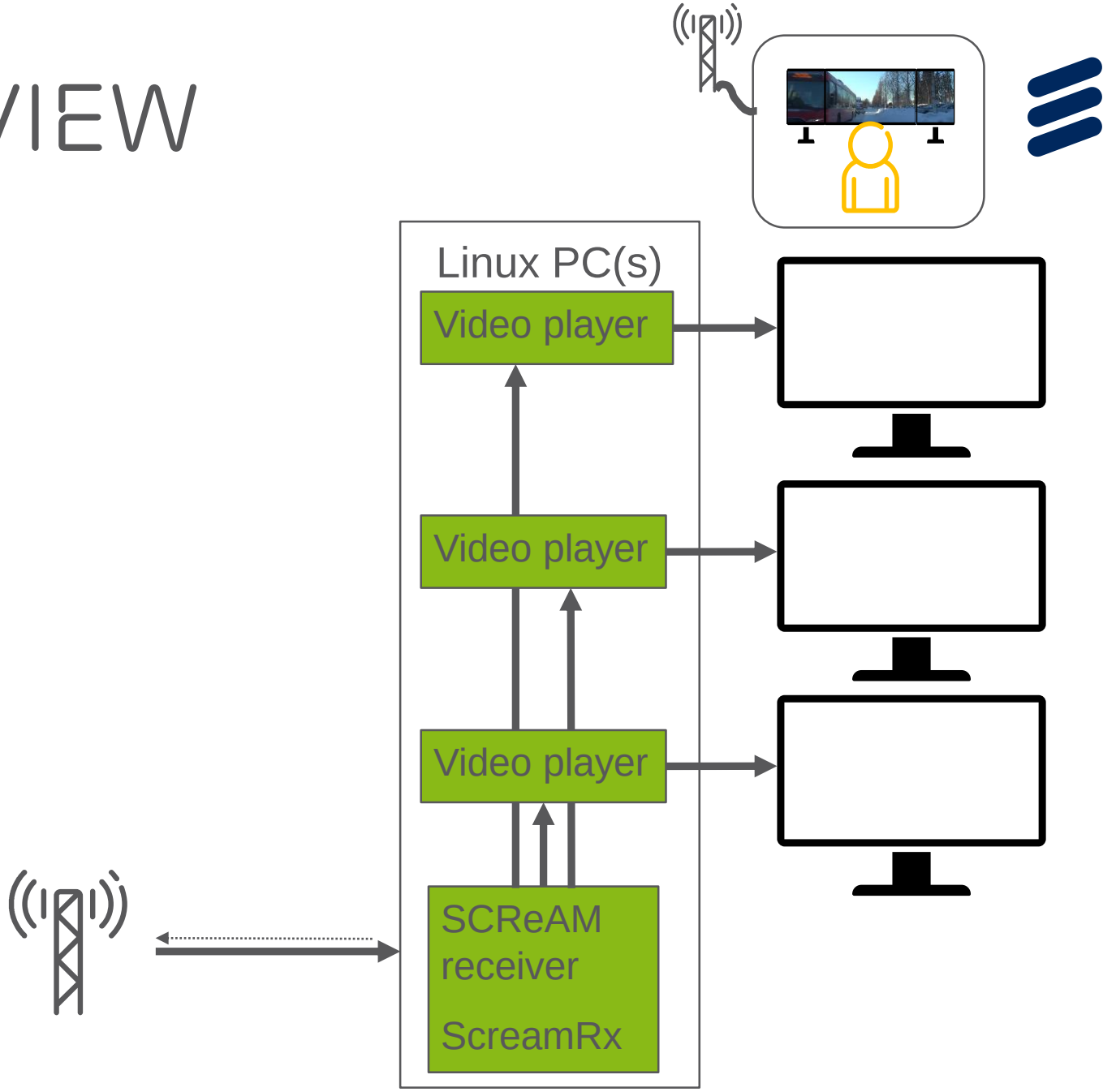
- Linux PC handles congestion control algorithm
  - Simple RTSP clients start RTSP streaming from each camera and makes RTP stream direct towards SCReAM sender.
    - For the cases that IP cams require RTSP control (e.g. Panasonic)
  - SCReAM sender handles congestion control, stream prioritization and rate control of IP cameras.

- PC can likely handle other tasks too as SCReAM congestion control algorithm has a modest complexity
  - Raspberry PI 3 → ca 5-10% @ 3Mbps



RECEIVER SIDE VIEW

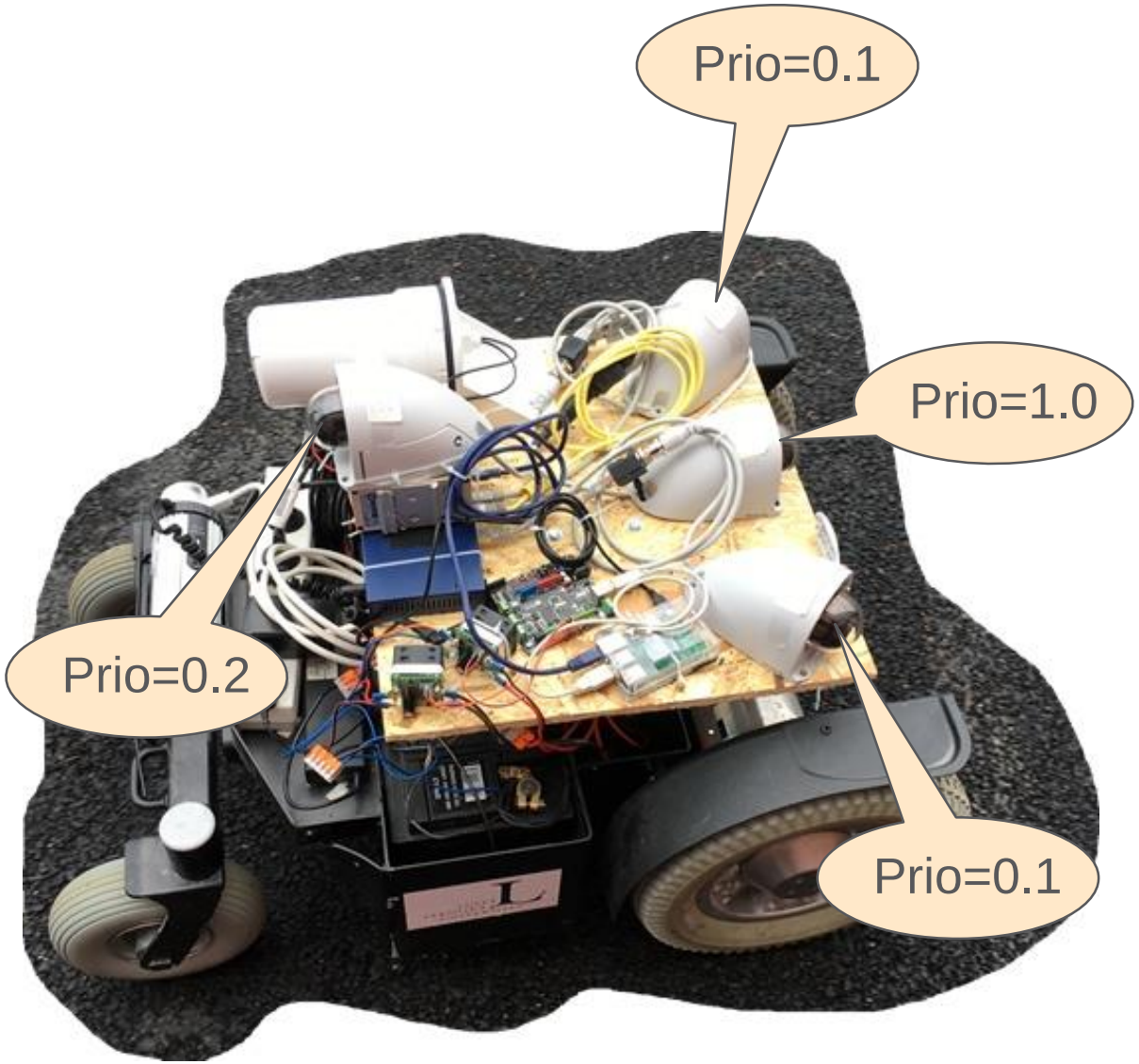
- Linux PC on receiver side
- Implements generation of congestion control feedback and forwards packets to Video player
- Low complexity



STREAM PRIORITIZATION

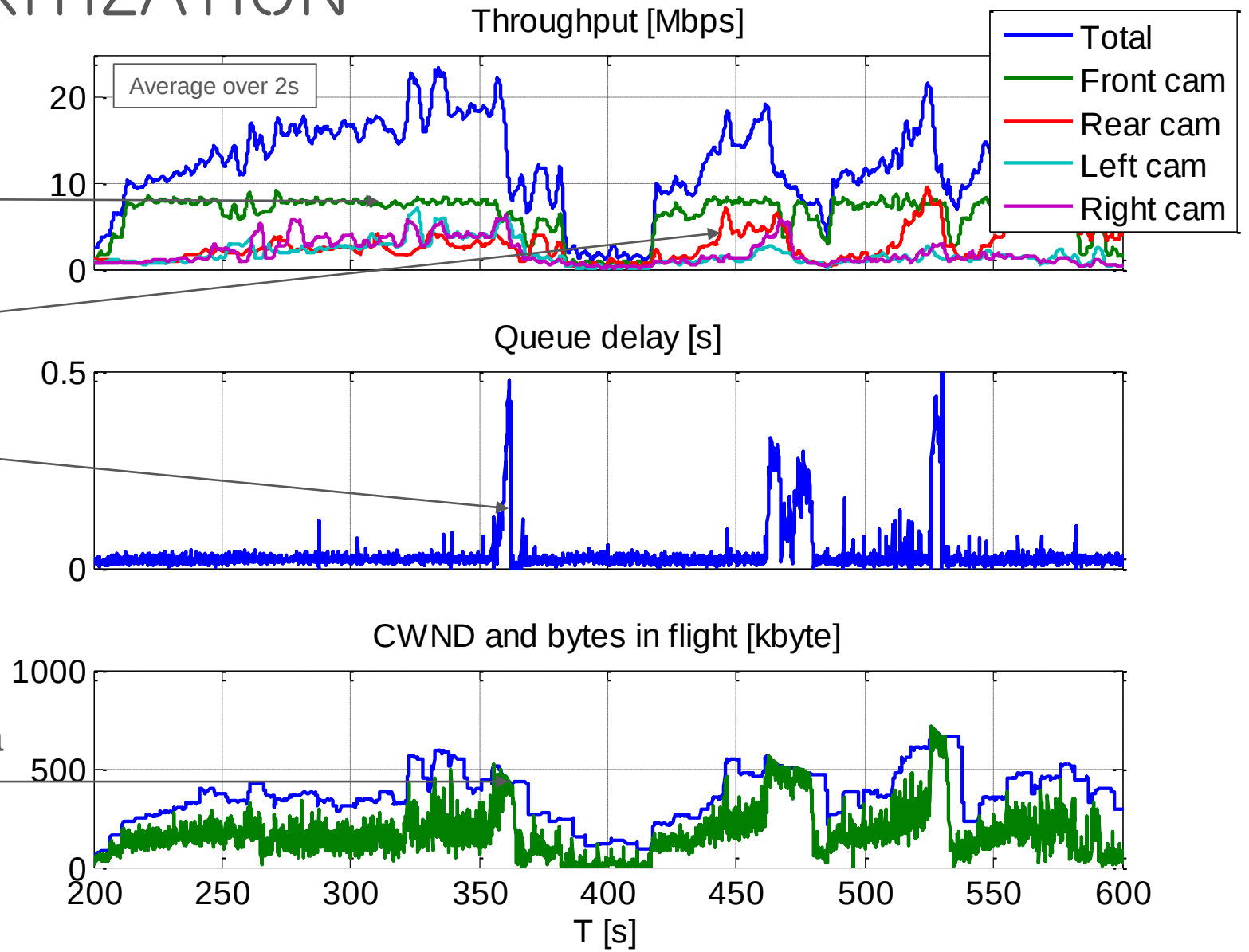
- Objective : Ensure that the most important media gets most of the resources when bandwidth is limited
- Weighted credit based scheduling with configurable weights in range]0.0 .. 1.0]
- Periodic rate adjustments
- Bandwidth allocation in theory

Camera	Bandwidth share [%]	
Front	**72**	[1.0/(1.0+0.2+0.1+0.1)]
Rear	**14**	[0.2/(1.0+0.2+0.1+0.1)]
Left	**7**	[0.1/(1.0+0.2+0.1+0.1)]
Right	**7**	[0.1/(1.0+0.2+0.1+0.1)]



LIVE EXAMPLE
WITH STREAM PRIORITIZATION

- The highest priority camera gets the highest bitrate
- Lower priority cameras catch up when throughput increases
- Occasional large delays when link throughput drops
- Congestion control is only delay based
  - Lack of proper AQM in LTE modem (tail drop queue) → packet loss based adaptation is disabled → Reaction to reduced throughput is a bit slow
  - ECN would be real good



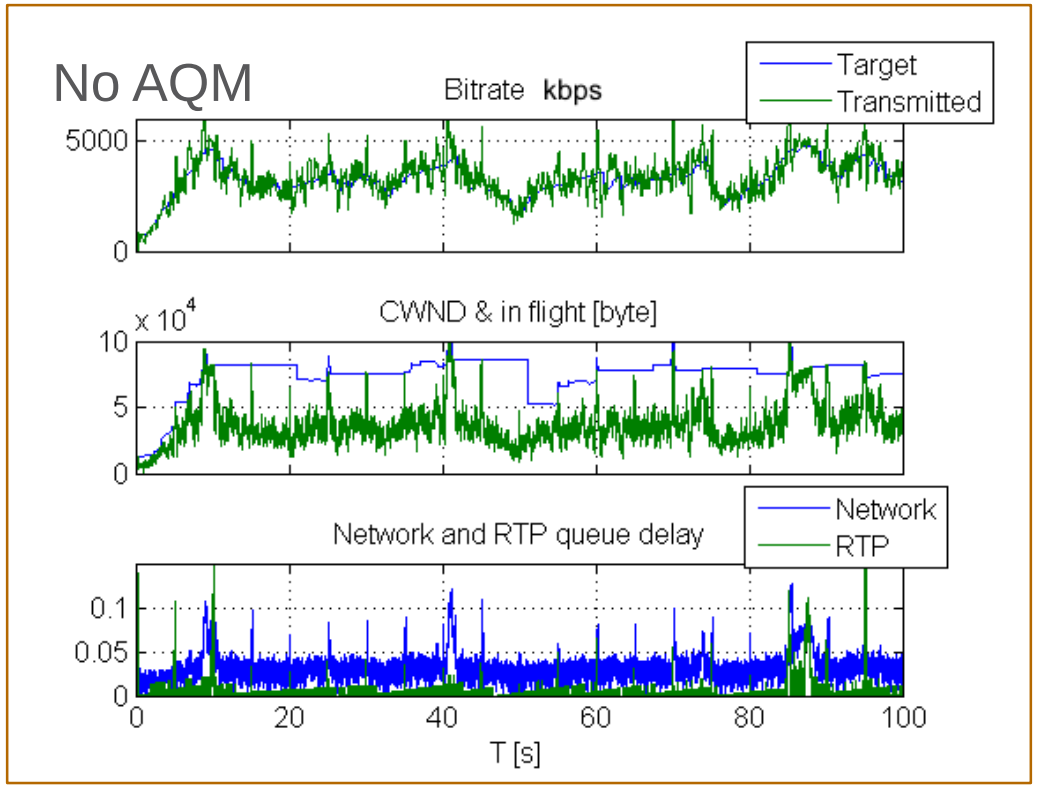
STREAM PRIORITIZATION INITIAL IMPRESSIONS

- Works quite OK!
- Improved stability for high priority media
  - Lower priority media (side and rear cameras) take the hit when throughput drops
- Gradual tunnel vision effect when throughput drops
  - Lower priority media degraded first
- Sudden high dynamic input in high priority media is better absorbed
  - Lower priority media is pushed back

- Room for improvement
  - Switch off some cameras in very problematic conditions
  - Verify and possibly improve run time priority switching

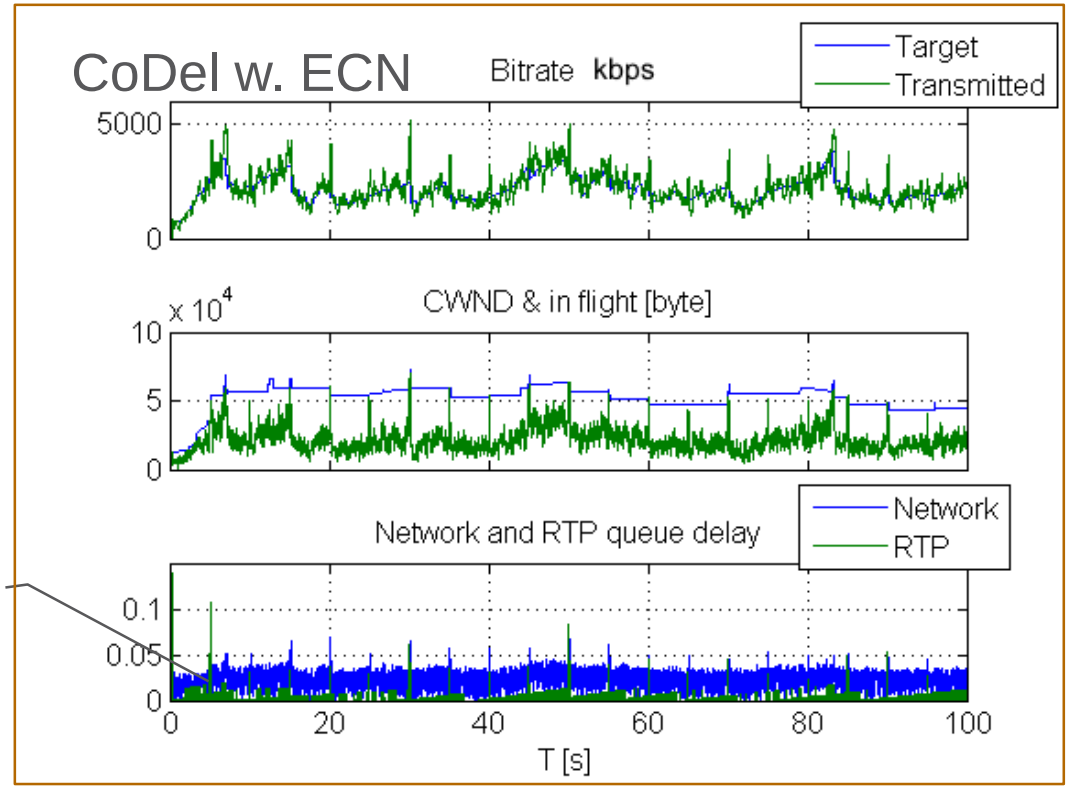
ECN SUPPORT

YES, ECN works !



ECN

Reduced e2e delay



CoDel without ECN
https://www.youtube.com/watch?v=J0po78q1QkU

CoDel with ECN
https://www.youtube.com/watch?v=qIe0ubw9jPw

Complex (video) sources → ECN improves performance

CONCLUSION

- SCReAM can provide high quality video feedback for a remote control applications by providing congestion control
  - Reduces impact of varying connectivity quality
  - Reduced impact of varying media rate
- Solution manages to control resulting video bitrate well and thus avoids excessive queue delay and loss in network
  - Configurable stream priority

- Rate adaptive video feedback and congestion control provide good quality!

Comments/questions ?

ingemar.s.johansson@ericsson.com

65.5263209N, 22.7964308E

EXTRA

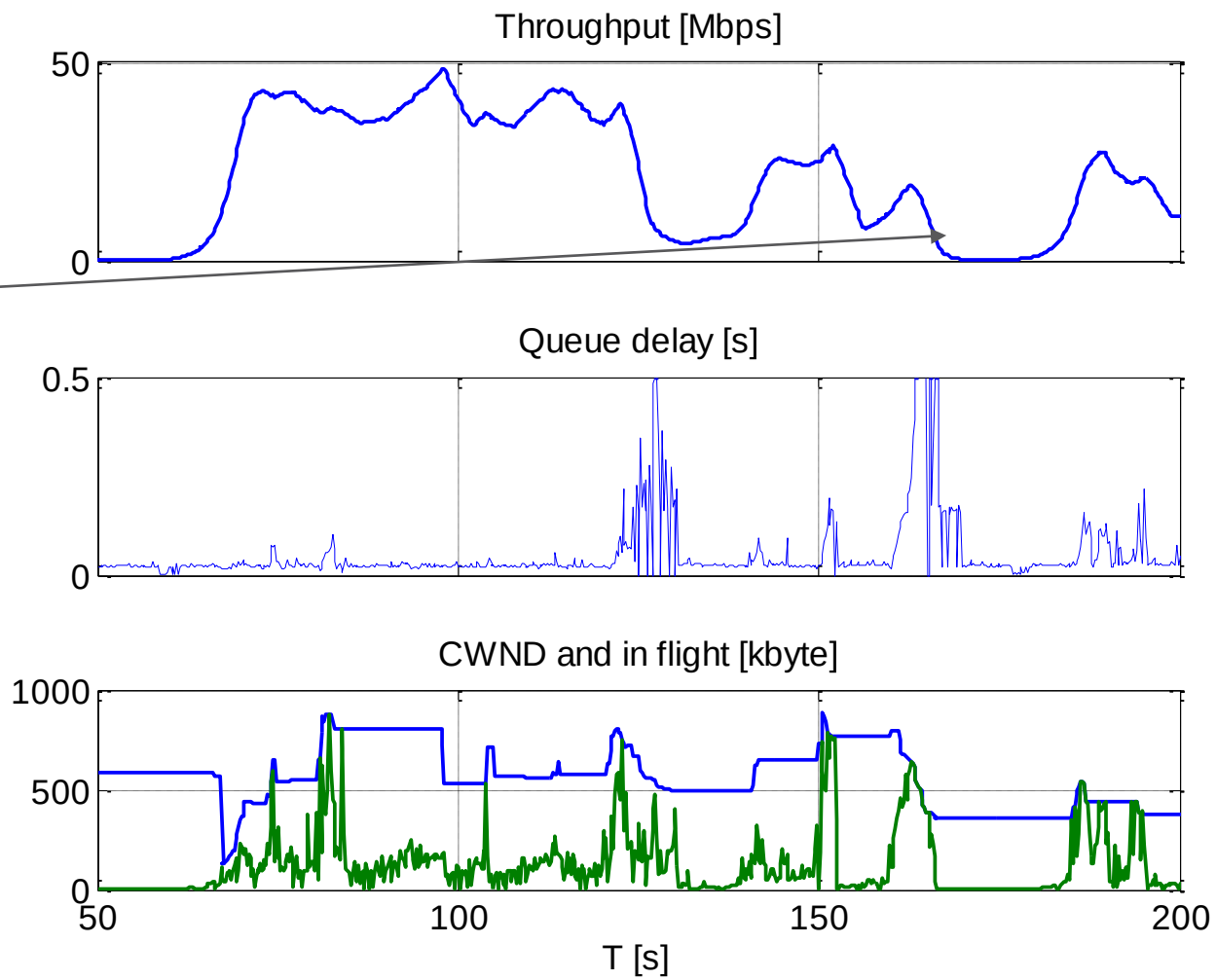
WiFi Network test

INTRO

- Highly non-scientific test with a bandwidth test application
  - Based on code from https://github.com/EricssonResearch/scream
- Test gear
  - ASUS RT-AC66U, 5GHz band
  - Two Lenovo Thinkpad E470, Ubuntu 17.10
- Test application implements a “fake” RTP packet generator
  - Mimics rate adaptive video encoder with 25fps, no key-frames
- Test range [0.1..100] Mbps
- Other traffic present in WiFi network
  - Teenage daughter…
  - Streaming video, Netflix
  - Snapchat

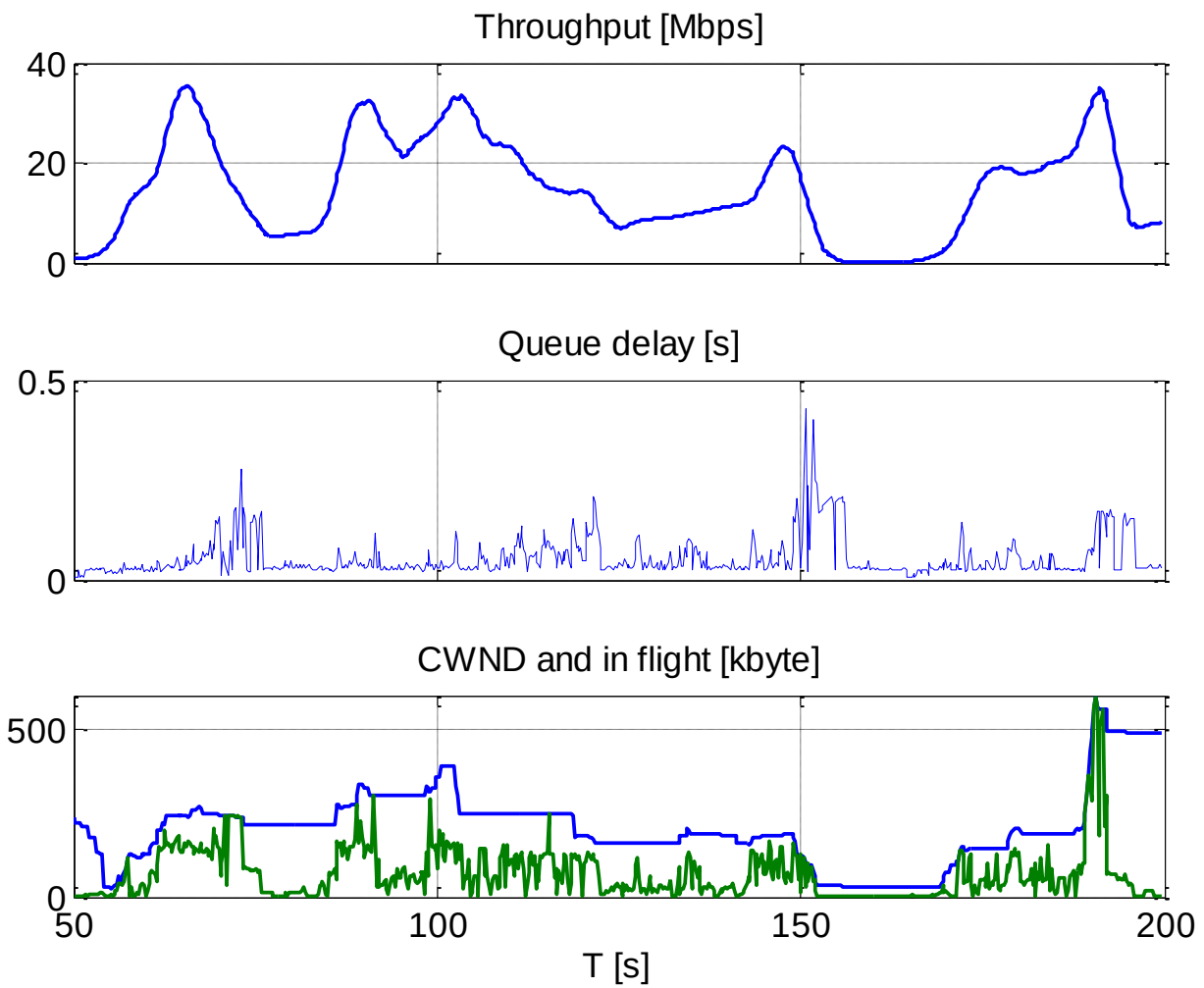
EXPERIMENT 1
SENDER CLOSE TO AP, RECEIVER MOVING

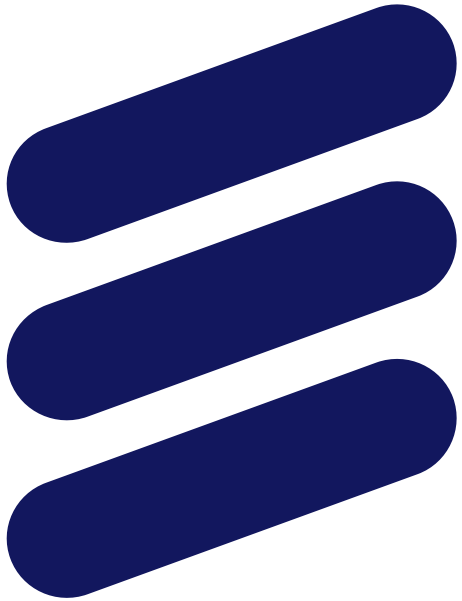
- Congestion control adapts to estimated congestion (queue delay)
- Bitrate sometimes forced to very low values



EXPERIMENT 2
RECEIVER CLOSE TO AP, SENDER MOVING

- Congestion control adapts to estimated congestion (queue delay)
- Bitrate sometimes forced to very low values





ERICSSON