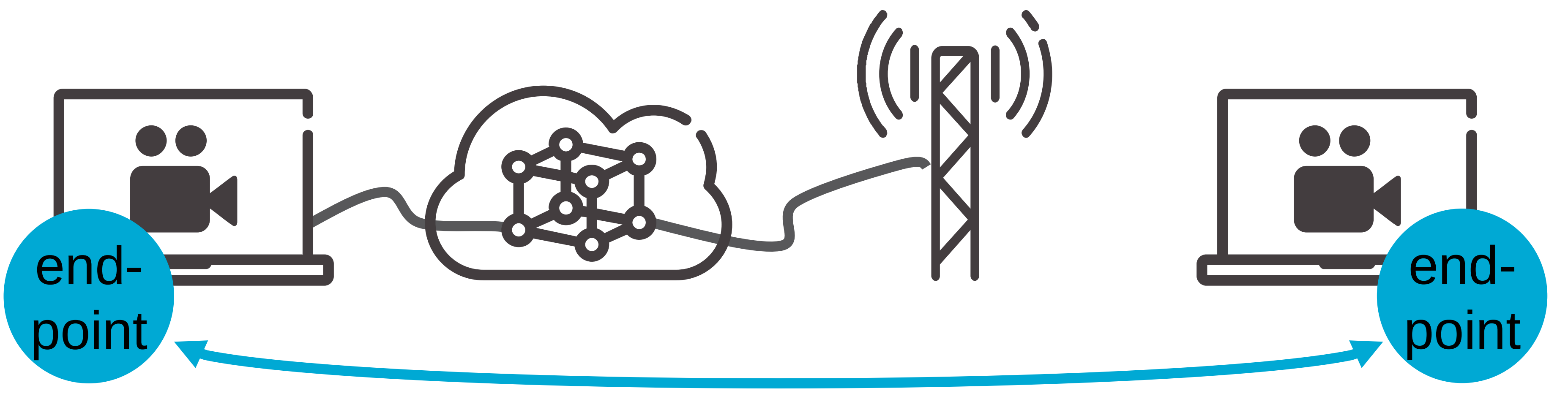
NETWORK ASSISTED RATE ADAPTATION FOR CONVERSATIONAL VIDEO OVER LTE

CSWS'14

{Ylva.Timner, Jonas.Pettersson, Hans.Hannu, Min.W.Wang, Ingemar.S.Johansson}@ericsson.com

ENDPOINT BASED RATE ADAPTATION

- Examples
  - Google Congestion Control (GCC)
  - Self-clocked rate adaptation (SCReAM)
- Bandwidth is estimated based on delay and/or packet loss
- Path capacity is probed

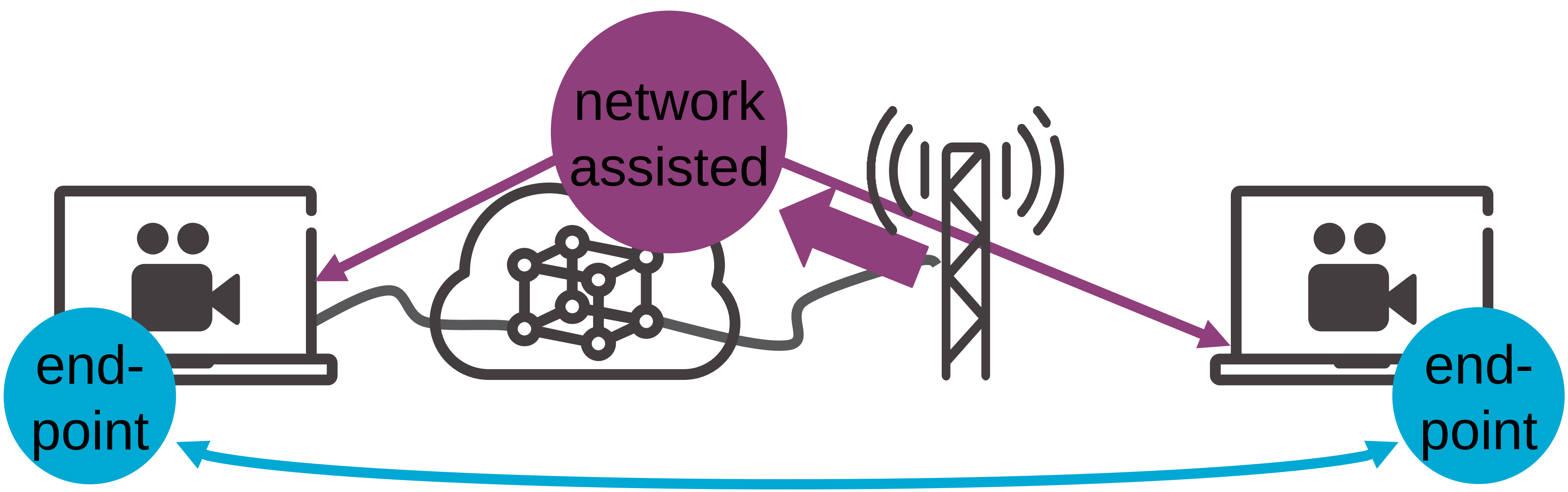


ENDPOINT BASED RATE ADAPTATION ISSUES

- Difficult to precisely determine congestion
  - Over-reaction to handover
  - Over-reaction to congestion
- Negative impact from other cross traffic
- Frequent (and large) bitrate changes
- Network service optimizations → congestion may become invisible to end users
- Fairness between users difficult to achieve

NETWORK ASSISTED RATE ADAPTATION

- Rate adaptation algorithm placed close to the air interface bottle neck
- A session controller is in control of the bitrates

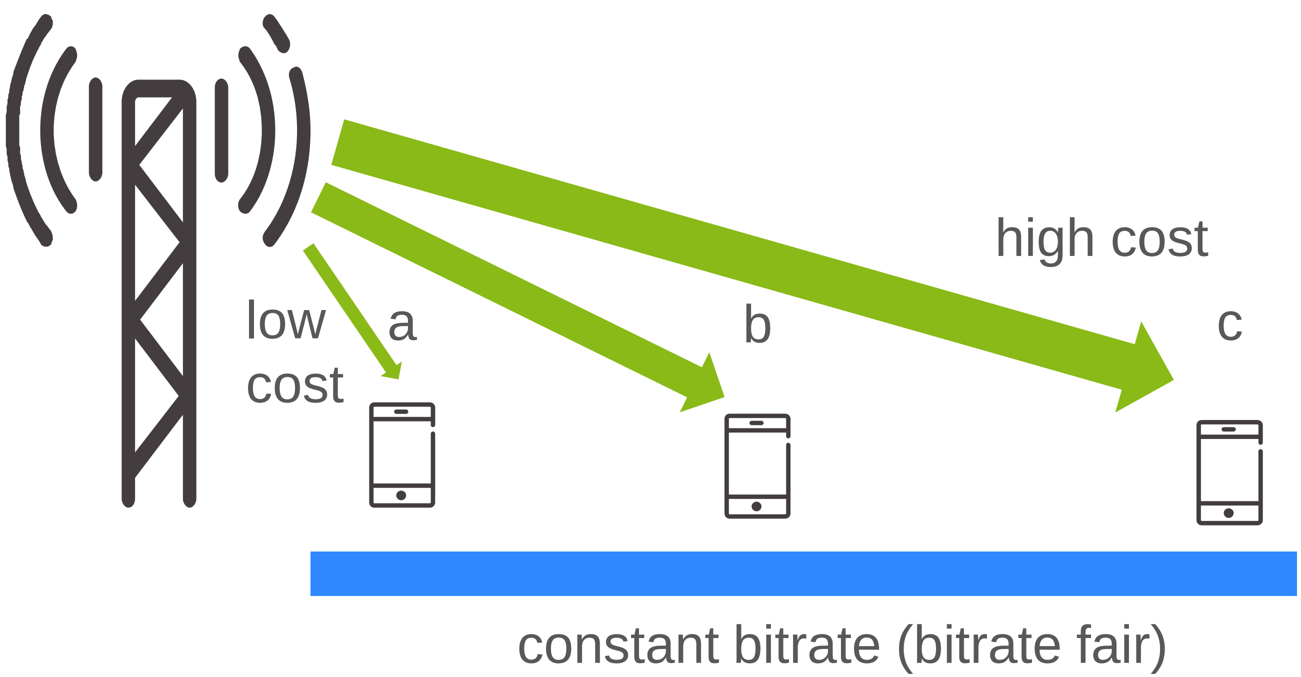


NETWORK ASSISTED RATE ADAPTATION

- Two algorithm alternatives examined:
  - Bitrate fair
  - Resource fair
- Network assisted rate values can be conveyed in RTP header extensions (RFC5285) or in RTCP application layer feedback messages (RFC4585) → framework can complement an endpoint based solution
- RRC signaling can be used to forward the bitrates to the session controller or directly to the terminals.

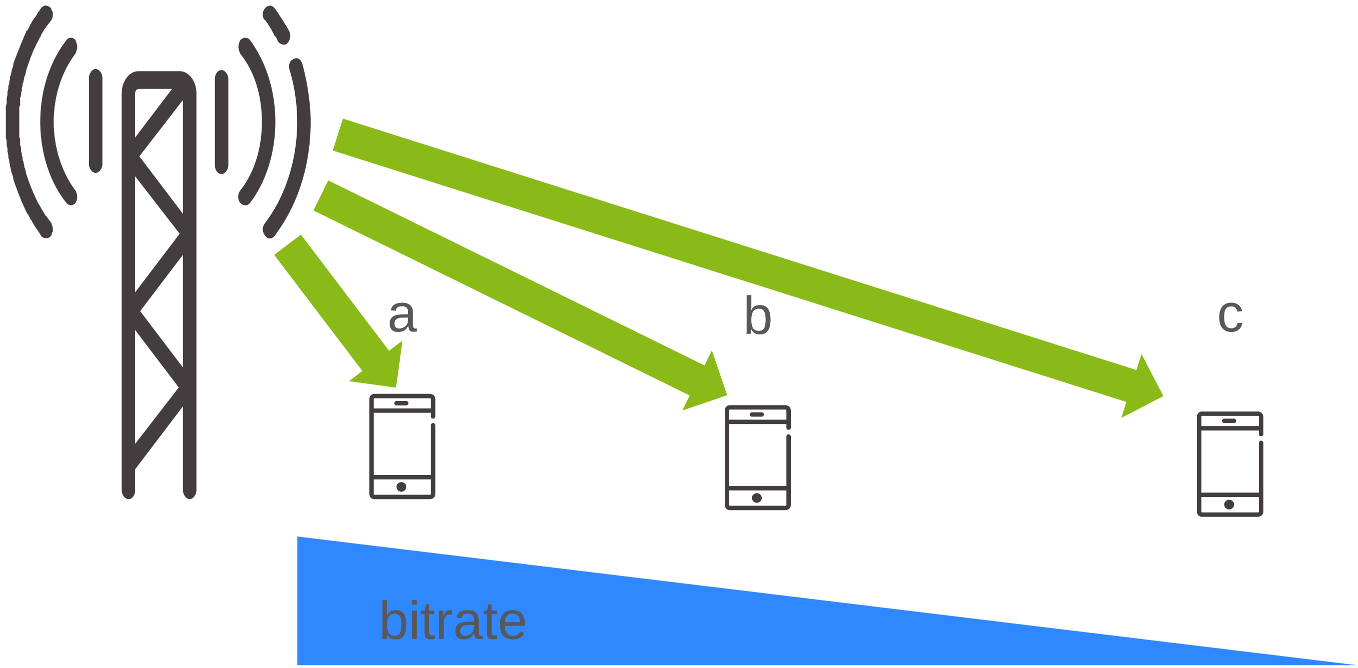
BITRATE FAIR

- All video users in a cell get the same rate
- A utilization target (e.g. 80%) controls the resource split between video and other users



RESOURCE FAIR (PROPORTIONALLY FAIR)

- All users (regardless of service) should get their fair share of the radio resources

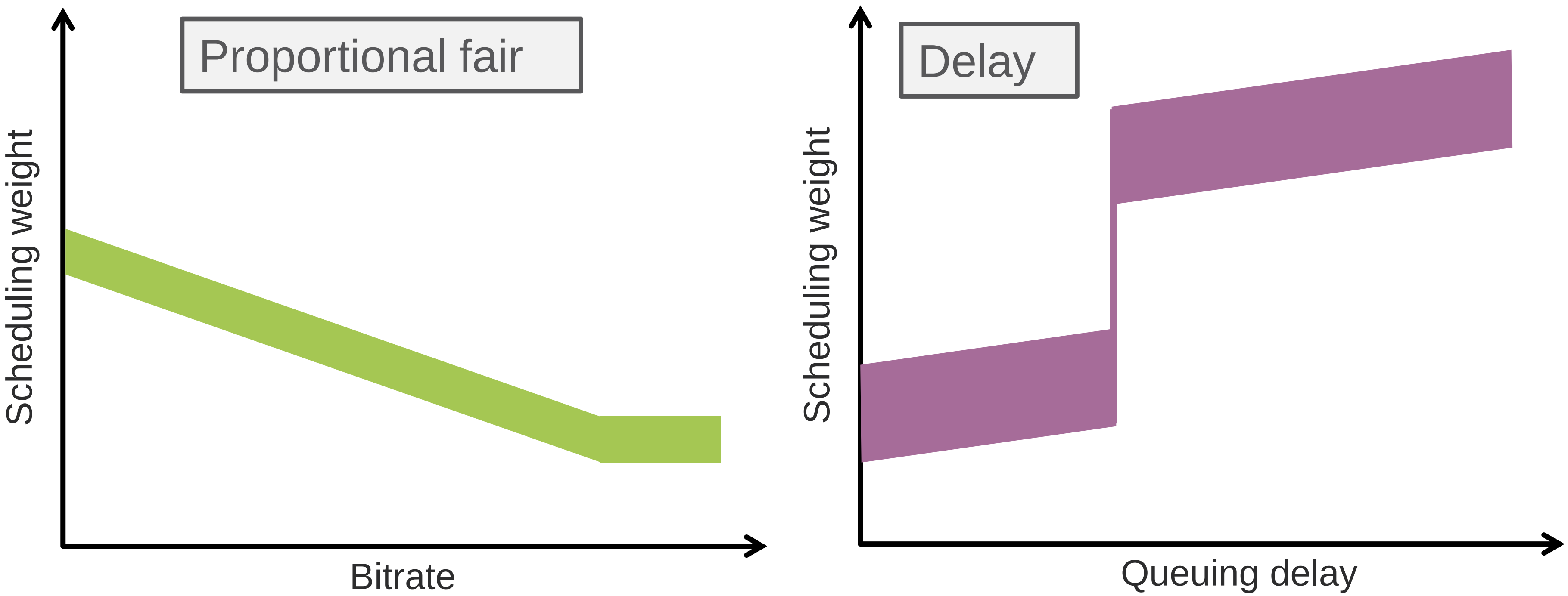


LTE SCHEDULING BASICS

- The scheduler allocates transmission resources to bearers (users)
- Scheduling interval: 1ms
- A scheduling weight controls the likelihood that a bearer gets transmission resources
- Different scheduling algorithms
  - Proportional fair
  - Delay (optimized queuing delay)

BEARER CONFIGURATIONS

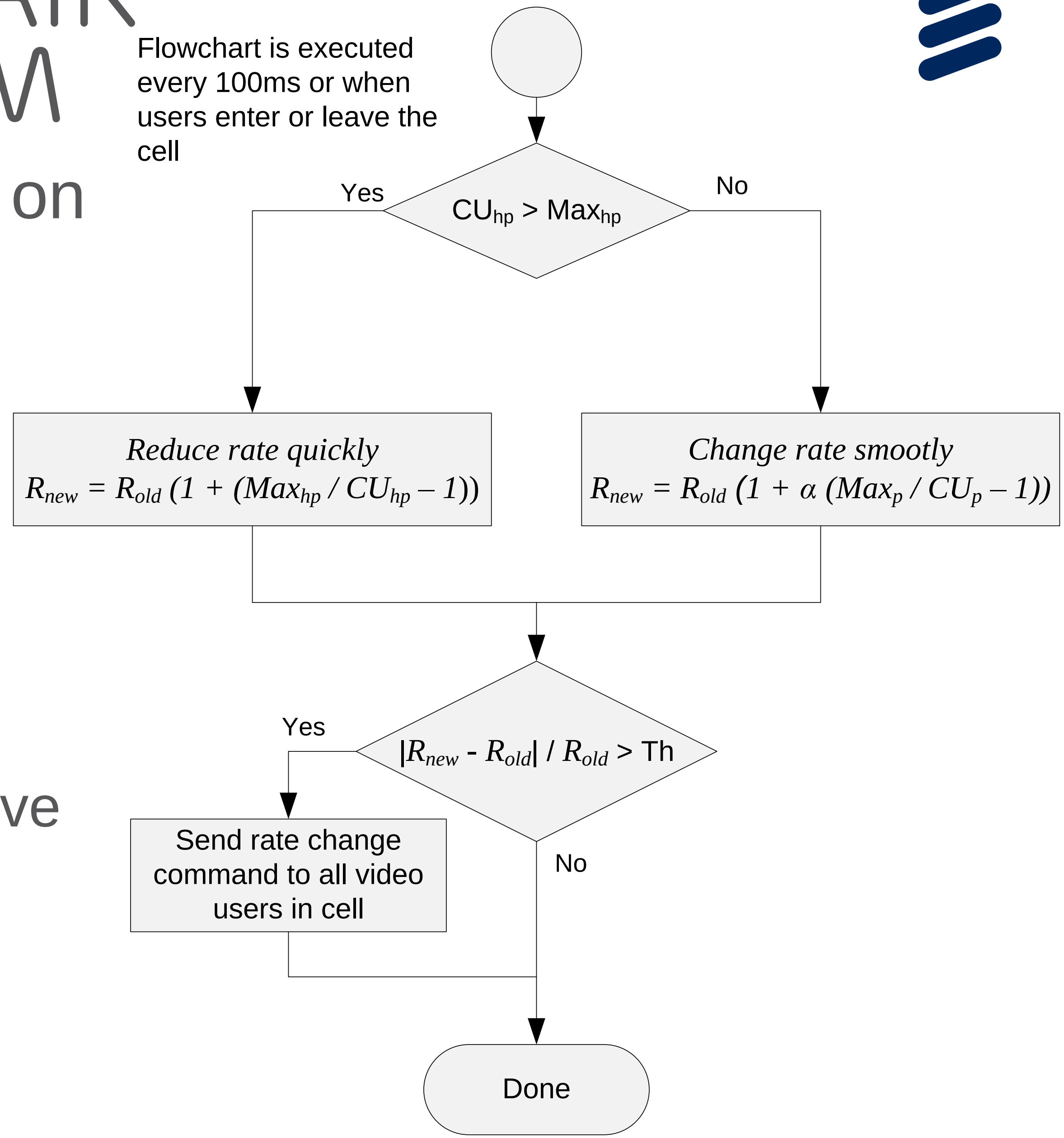
- FTP on “best effort” bearer:
  - Proportional fair
- Conversational bearer:
  - Delay scheduling



BITRATE FAIR ALGORITHM

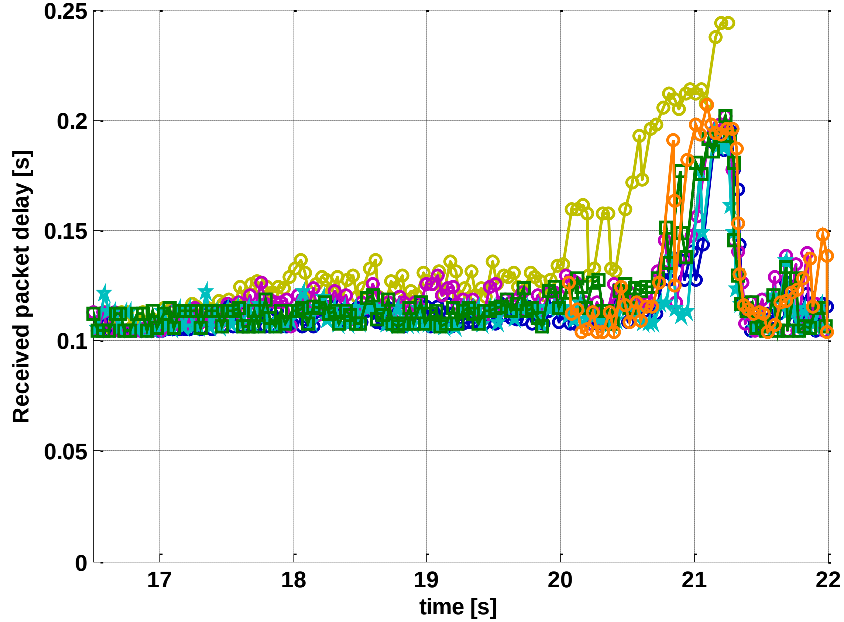
› Adaptation based on channel utilization measures:

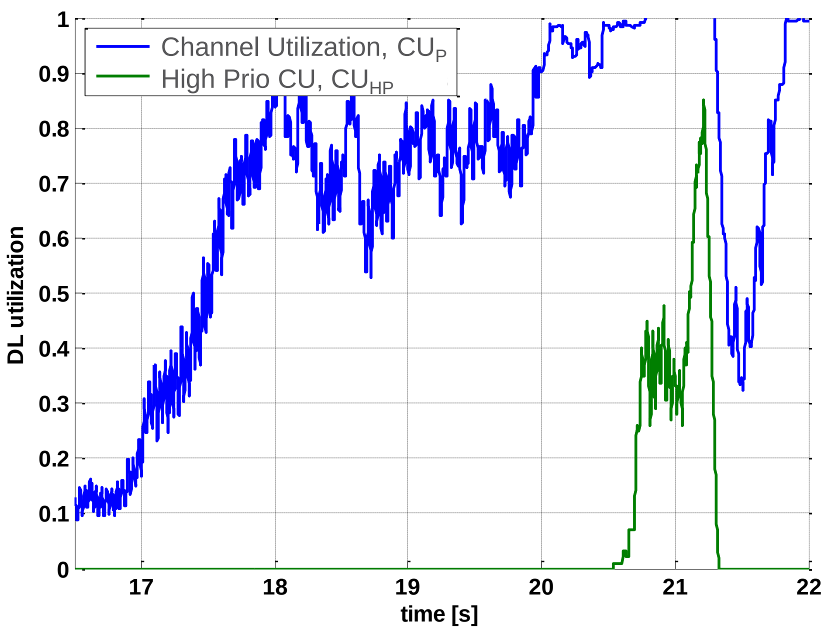
- CU_P : video traffic + tcp traffic.
- CU_{HP} : retransmissions + video packets above delay threshold

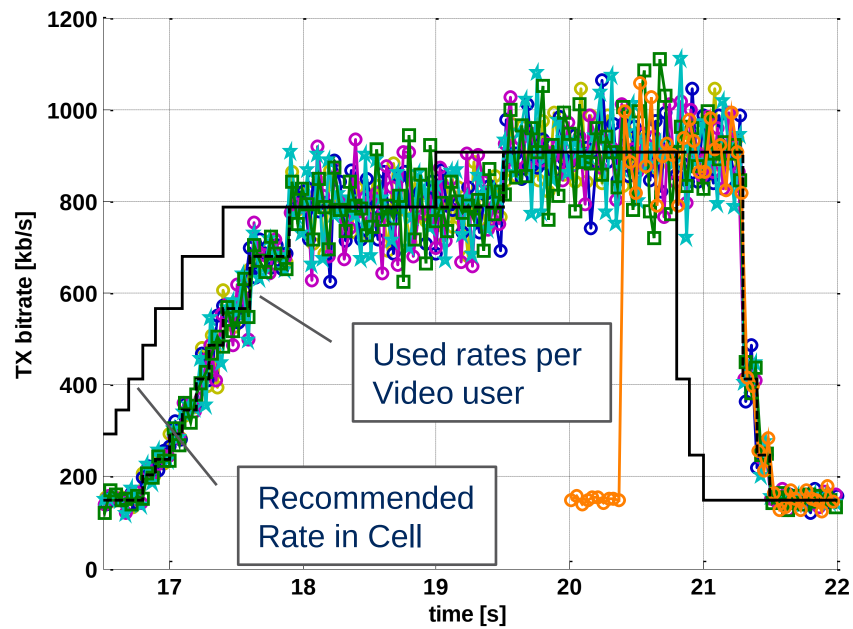


BITRATE FAIR EXAMPLE

- >18 s : low CU, ramp up rate.
- 18-20s – good utilization
- Delays are rising at 21s
- Detected by High CU_{HP}







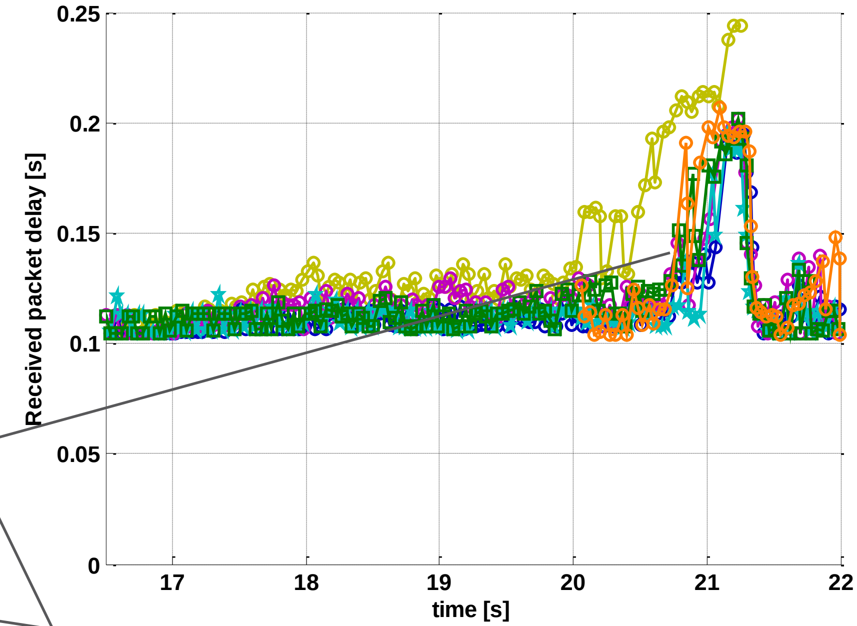
BITRATE FAIR EXAMPLE

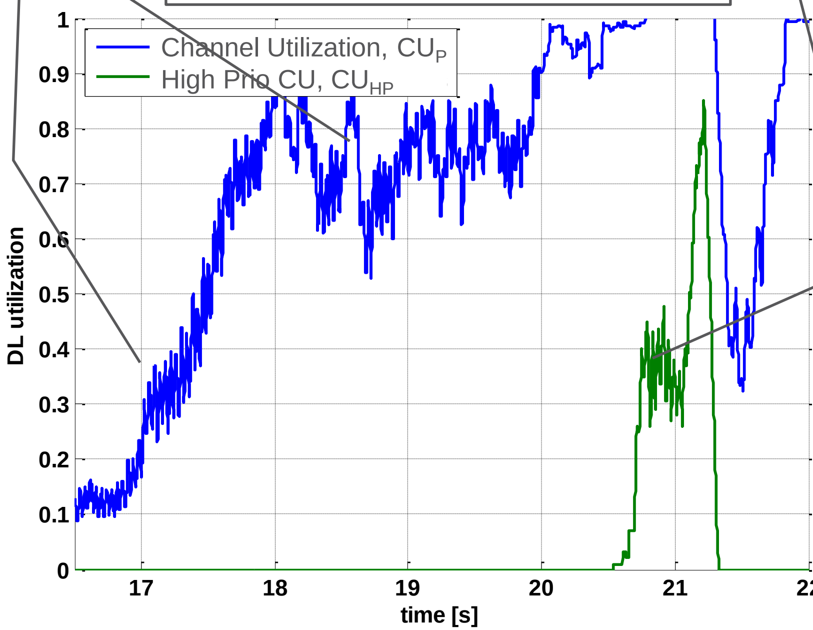
Low CU – Increase Rate

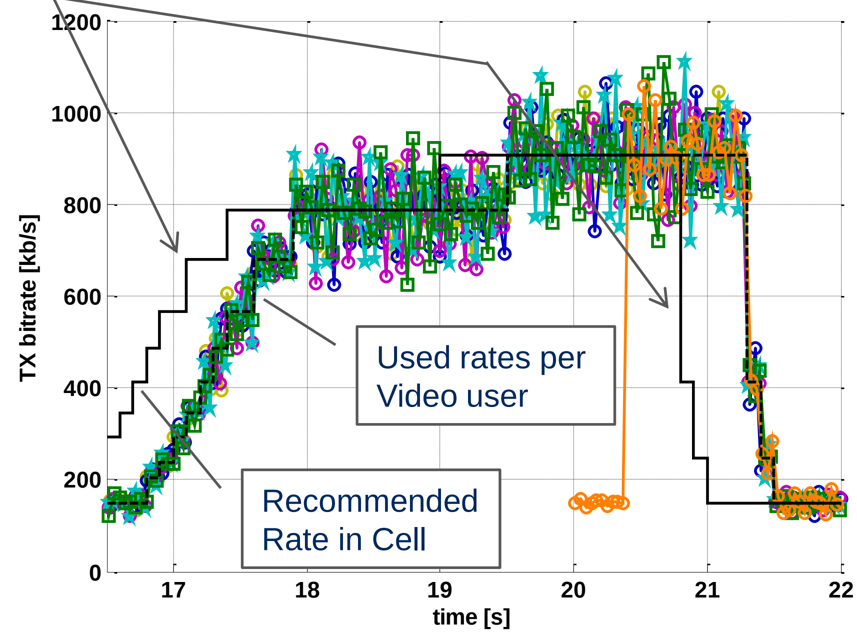
Good CU – Keep Rate

Delays rising, High CU_{HP}

– Decrease Rates fast







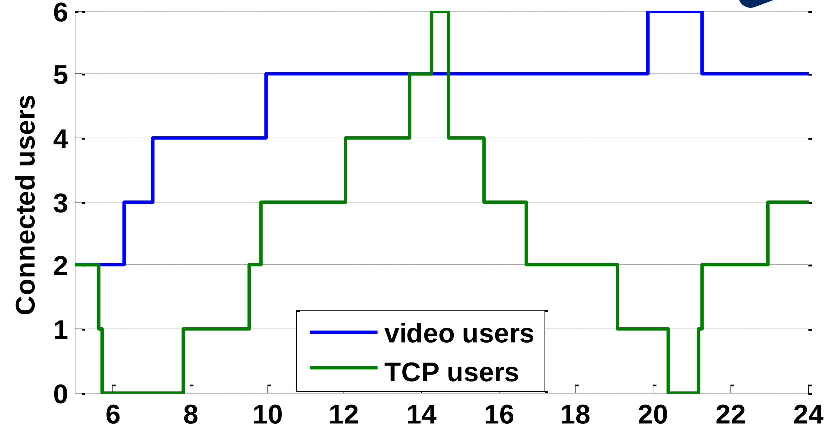
RESOURCE FAIR ALGORITHM

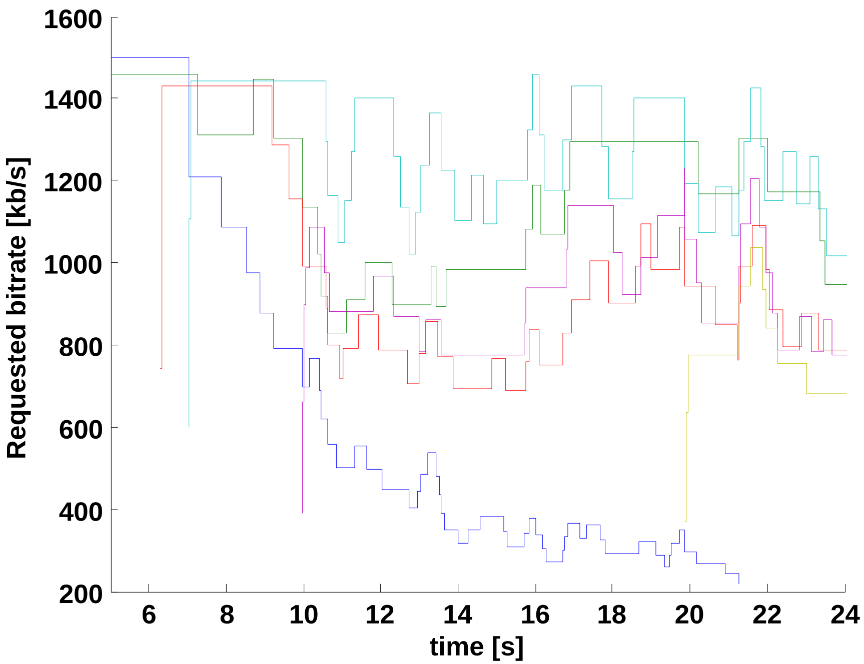
- Determine target utilization u^* (equal split):
- $u^* = \frac{1}{N_{video}+N_{other}}$
- Use measured channel quality (potential rate) to determine target rate, R, for each user i :
- $u_i = \frac{R_i}{Q_i} \quad \Rightarrow \quad R_i^* = u^* Q_i$

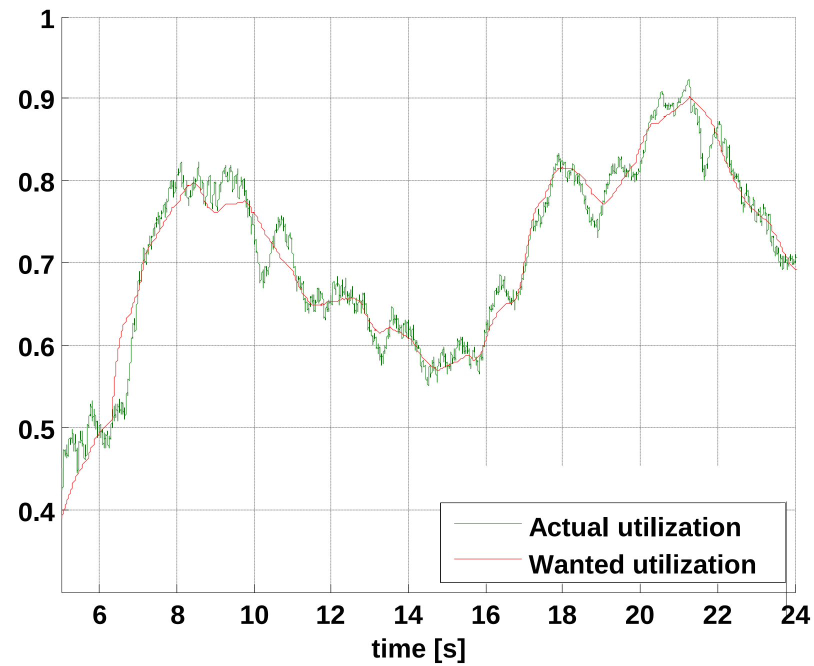
- A closed loop part complements with compensation for possible estimation errors

RESOURCE FAIR EXAMPLE

- Rate changes
- Ability to keep fairness target

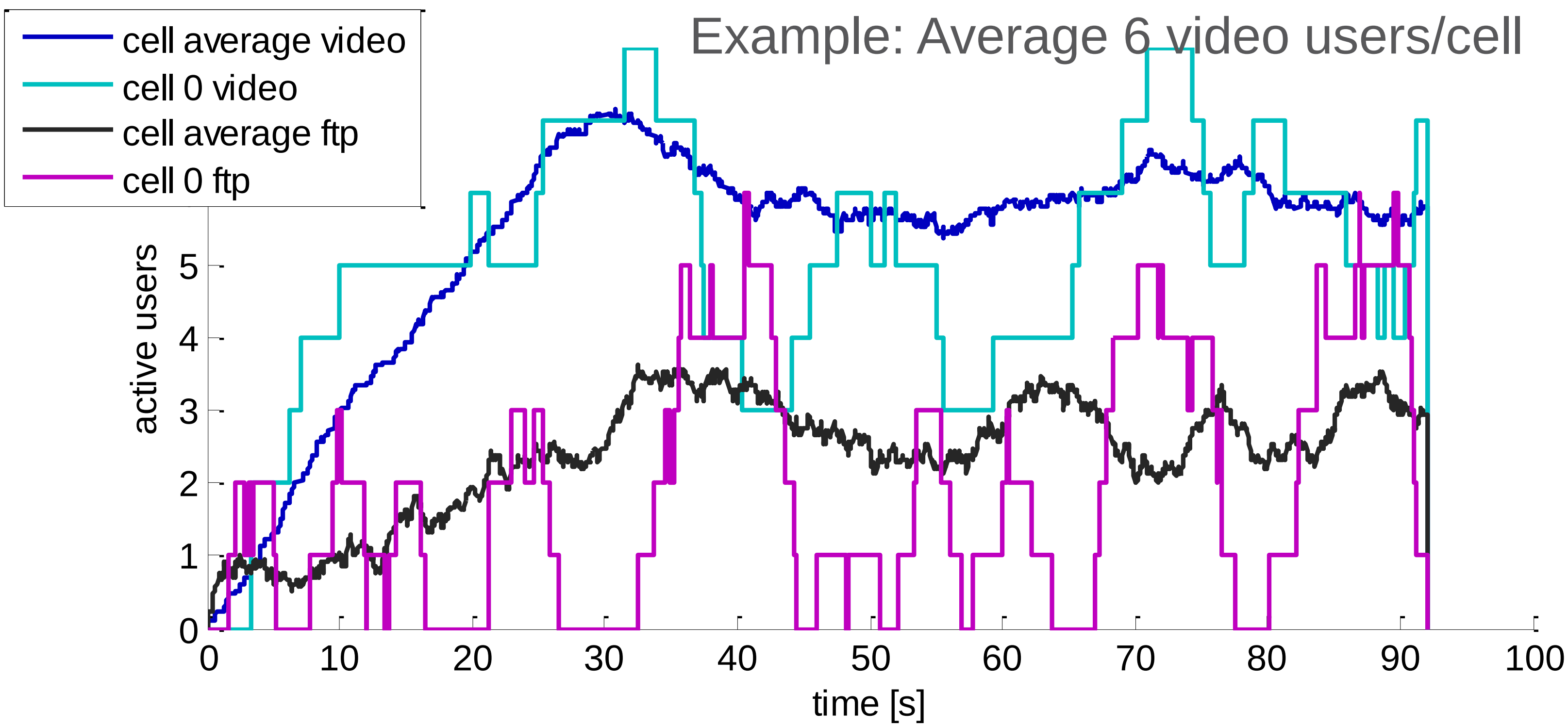






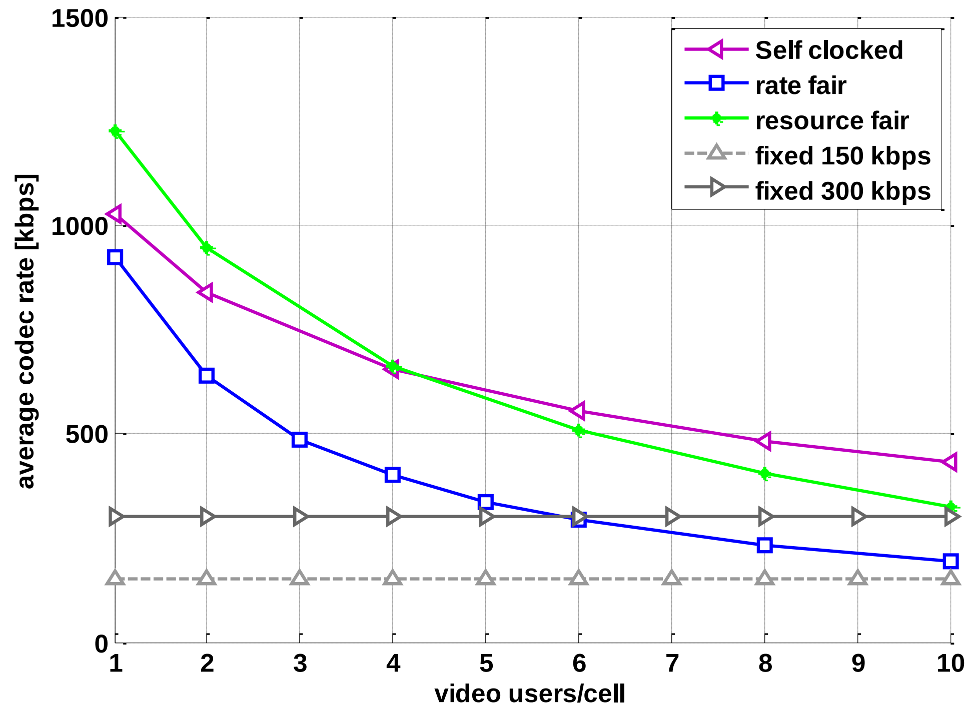
SIMULATION SCENARIO

- 21 cells, 3GPP Case 1, 5 MHz
- Conversational video: 150 kbps – 1500 kbps
- Other traffic: Small file download 2 Mbit/s/cell



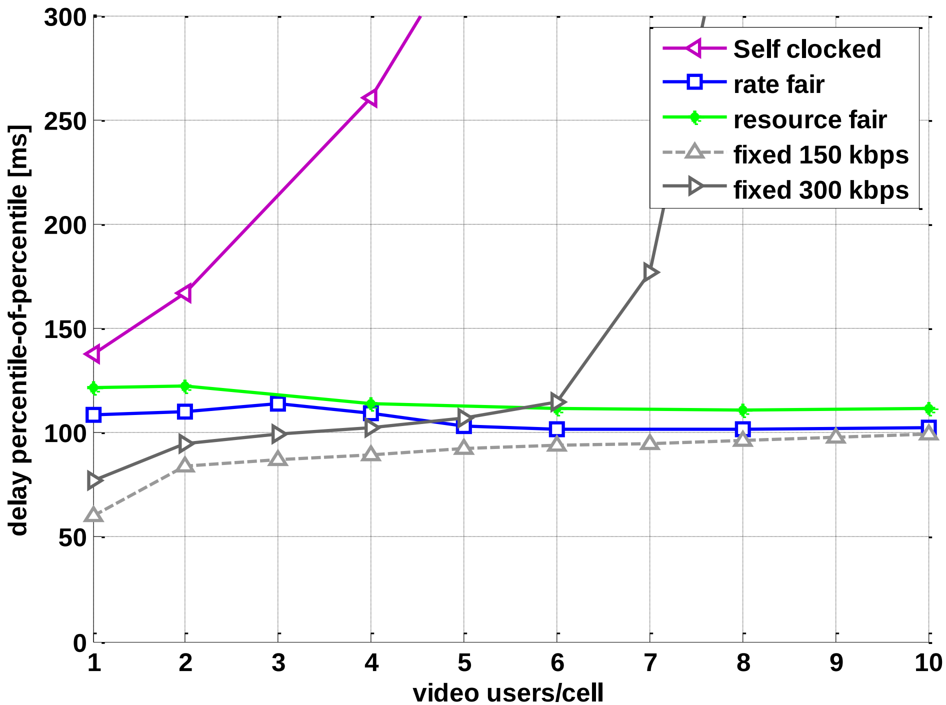
BITRATE

- Resource fair achieves the highest average bitrate among the network assisted rate adaptation algorithms
- …but is more conservative than endpoint solution at high load



LATENCY

- Both network assisted algorithms manages to ensure low latency even at high load levels
- Self-clocked algorithm works well only at lower load levels



CONCLUSION

- Flexible way to distribute system resources between conversational video and other best effort traffic
  - Fairness between users can be controlled
- The use of delay schedulers :
  - Gives very low latency regardless of load
  - Increased grace time upon congestion
- Precise discrimination between congestion and other non-congestion related impairments

ERICSSON

RESOURCE FAIR CLOSED LOOP PART

- Add a correction factor γ to the open loop:
- $R_i^* = u^* Q_i(1+\gamma)$
- Compute expected utilization:
- $U_{video}^* = \frac{N_{video}}{N_{video}+N_{other}}$ if unlimited rates, $\sum \frac{\text{limit}(R_i^*)}{Q_i(1+\gamma)}$ otherwise

- Adapt γ based on the relative error versus the measured utilization Ûvideo :
- $e = U_{video}^* - \widehat{U}_{video}$
- $\gamma_k = \gamma_{k-1} + K_p\left[\left(1+\frac{\Delta}{T_i}\right)e_k - e_{k-1}\right]$

ERICSSON