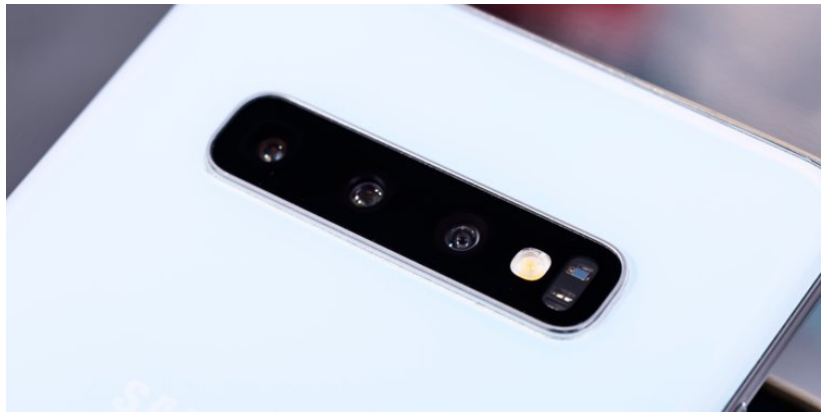
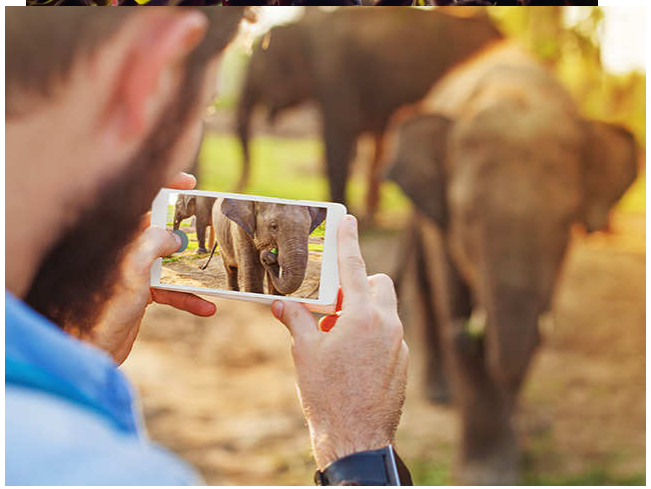
Edge-based Transcoding for Adaptive Live Video Streaming

Pradeep Dogga* Sandip Chakraborty† Subrata Mitra‡ Ravi Netravali*

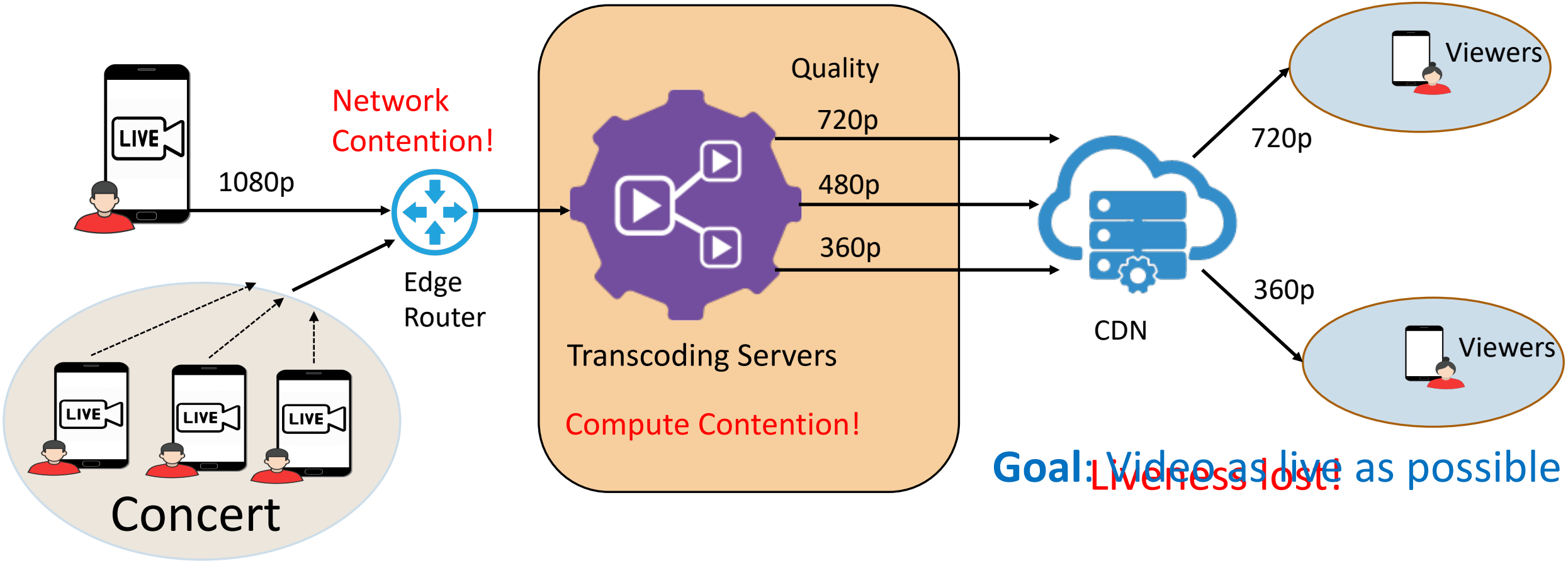
UCLA * † ‡

Increase in live video streaming



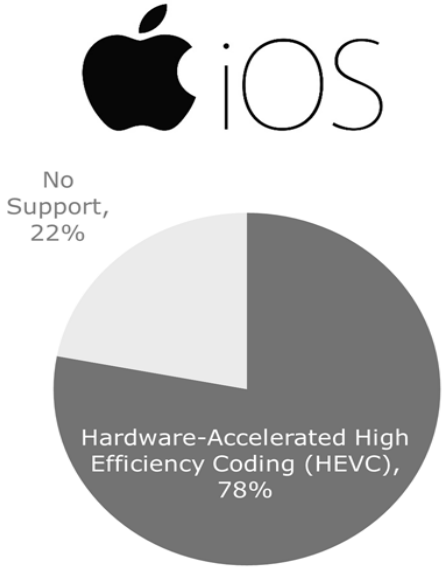


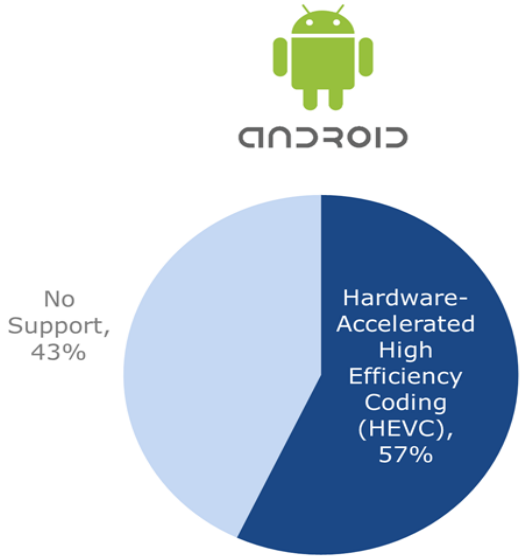
Live Video Streaming Architecture



Problem

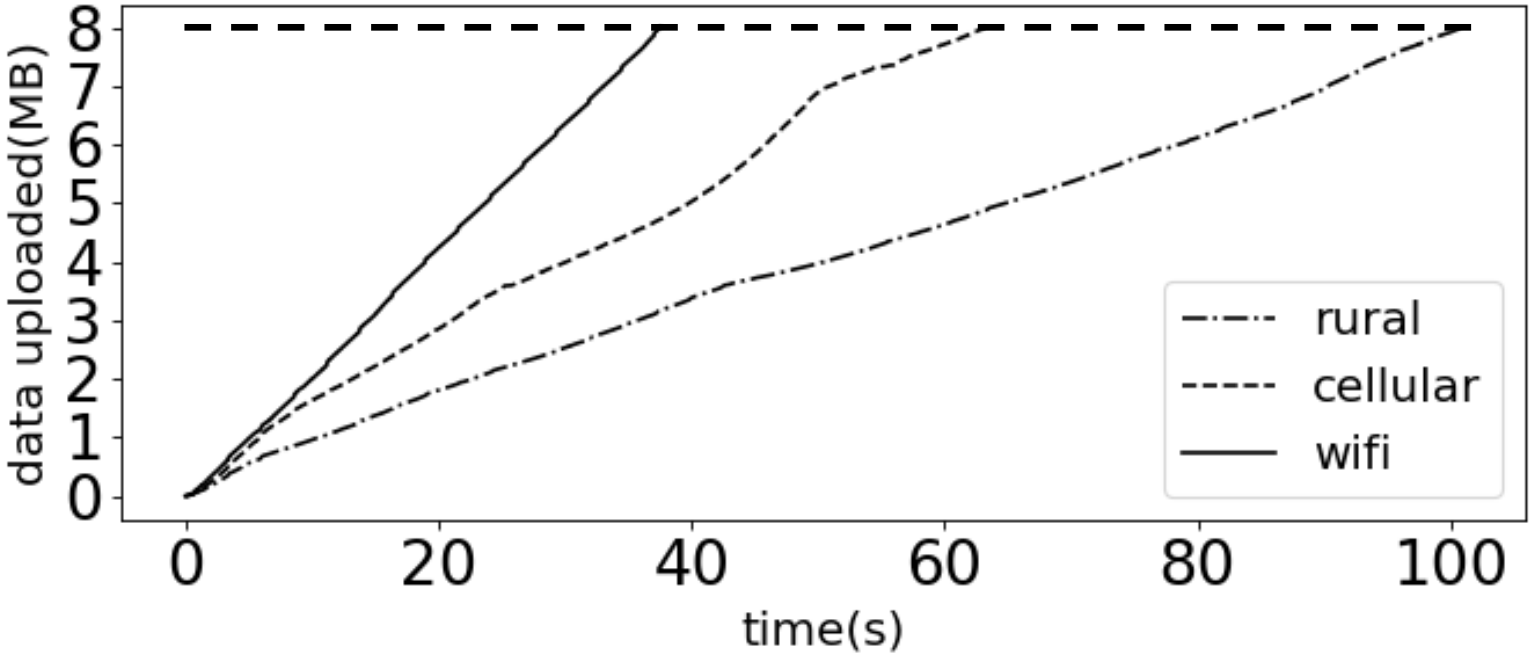
- Sharing infrastructure → Video liveness lost
- Key Question: Can we alleviate the problem by incorporating sources?
  - Can we use a P2P approach for transcoding and transmitting live video?





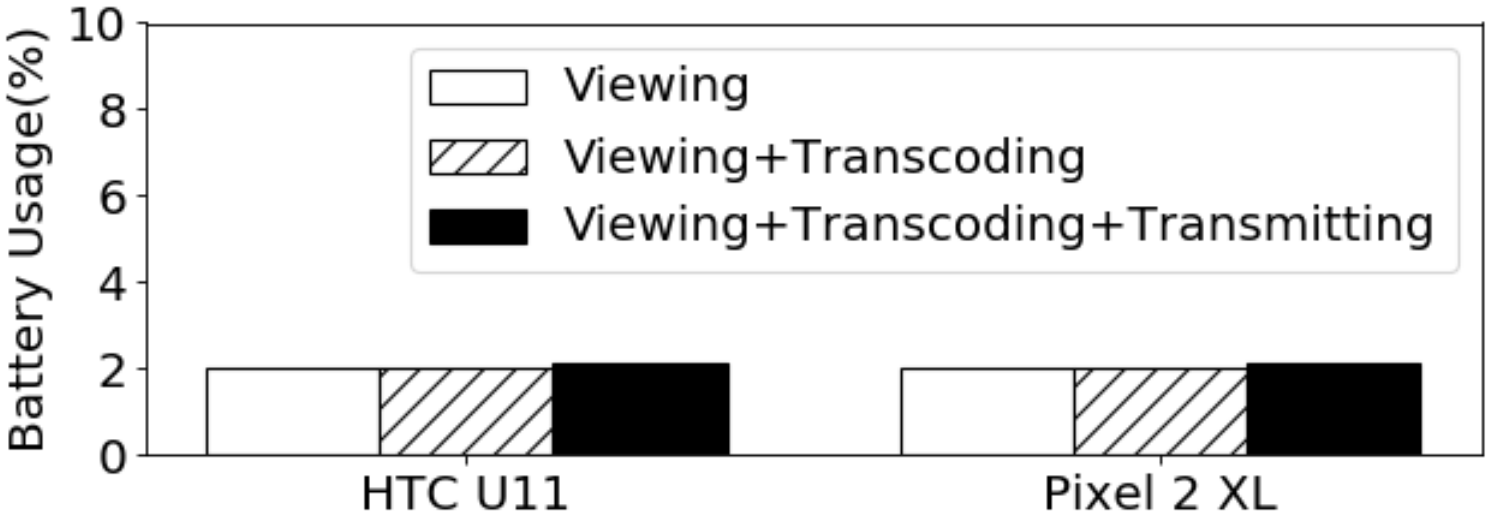
Reason 1

- Existing live video applications do not adapt to network conditions!



Reason 2

- Phones are energy constrained



Codecs are energy efficient!

Reason 3

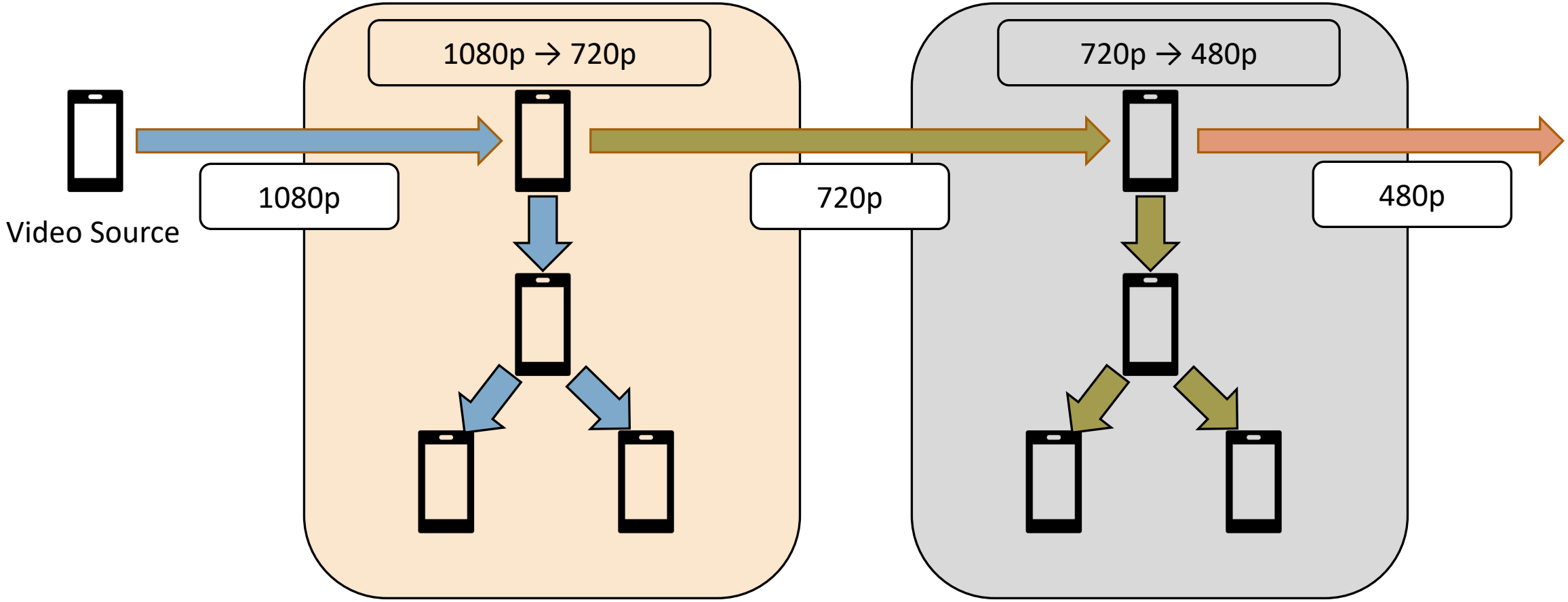
- Transcoding with smartphones should keep up with live video time limits

Device	1080p -> 720p	720p -> 360p	1080p -> 360p
HTC U11	15ms	7ms	9ms
Pixel 2 XL	8ms	6ms	9ms
PC (FFMPEG)	6ms	2ms	2ms

Time taken to transcode video per frame

Codecs are fast!

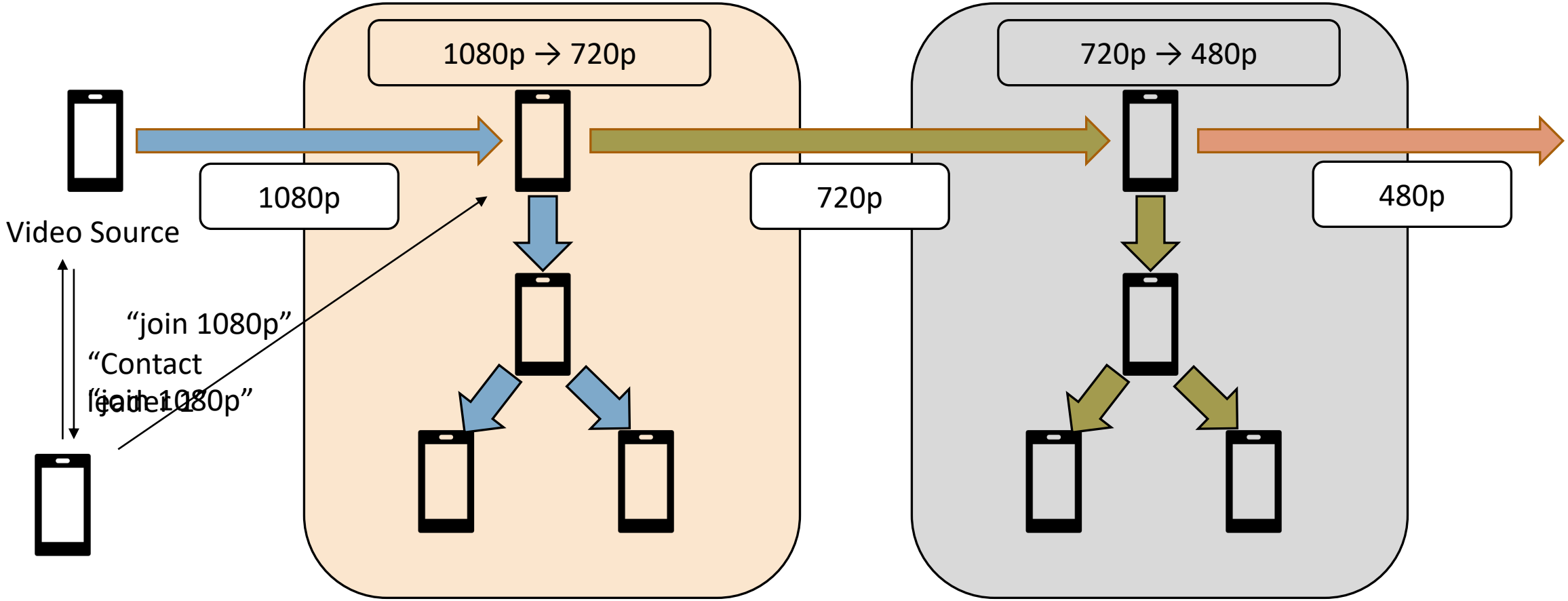
Peer to peer transcoding and streaming



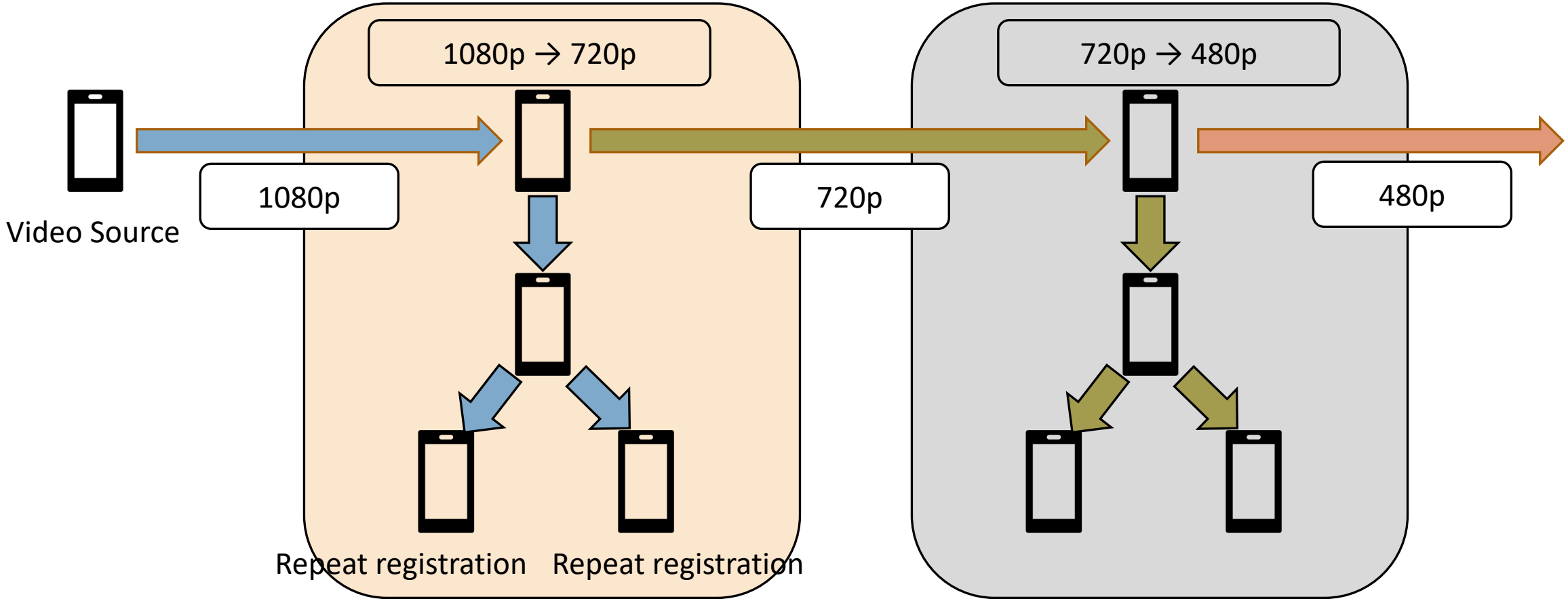
Balancing Trees

- Balancing trees when nodes leave or enter the system to minimize latency
- Restructuring trees for fairness in energy and available bandwidth consumption
- Can be modelled as an optimization problem

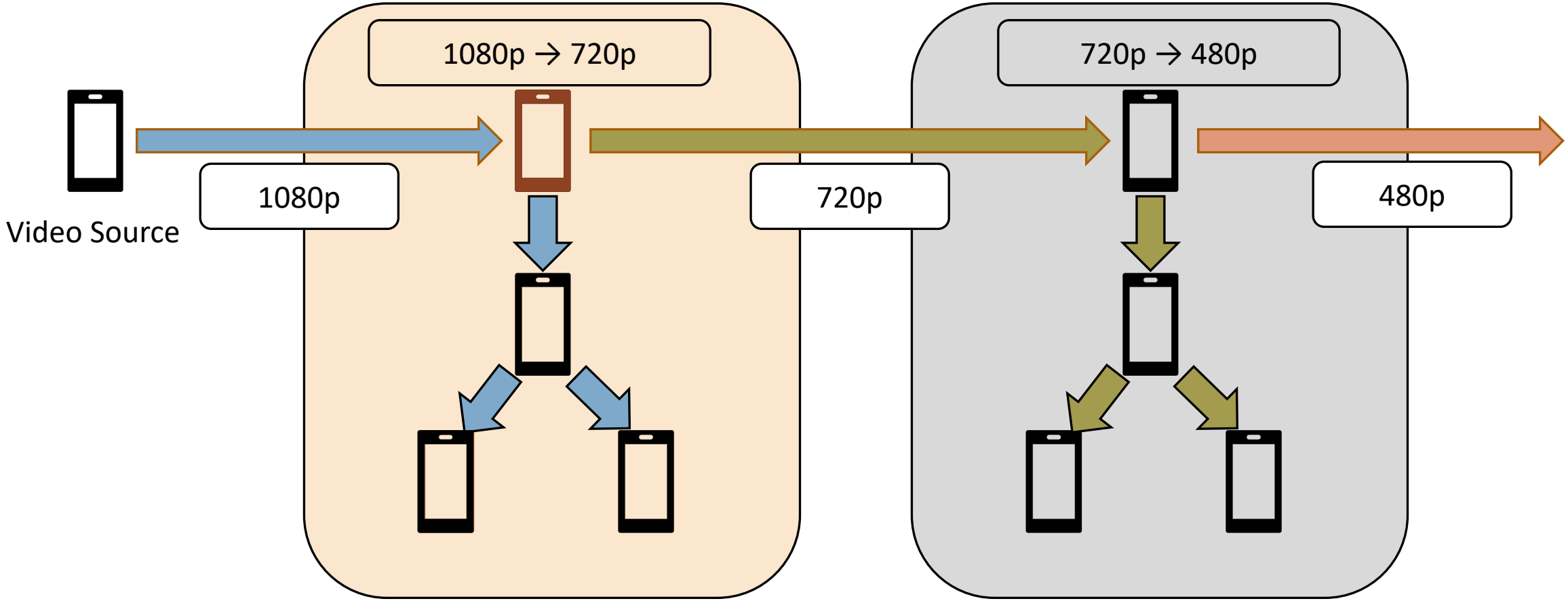
Details - Registration



Details – Failure/Departure of viewer



Details – Swapping transcoders



Open Issues

- **Peer Presence**: Incentives for transcoding and transmitting
- **Data Integrity**: Overheads in verification and attestation
- **Seeks and Delayed Viewing**: Mechanism to store stream intermittently
- **Viewer Flux**: Need decentralized heuristic based optimization to closely approximate to optimal distribution of viewers

Alternate use-cases

- Real-time Video Analytics can avoid server-side computation overheads
- Private Live Video can be enabled without additional infrastructure

Conclusion

- Significant opportunity to better use client devices in live streaming systems
  - Hardware accelerated codecs on phones are fast and energy efficient

Thank you!

dogga@cs.ucla.edu