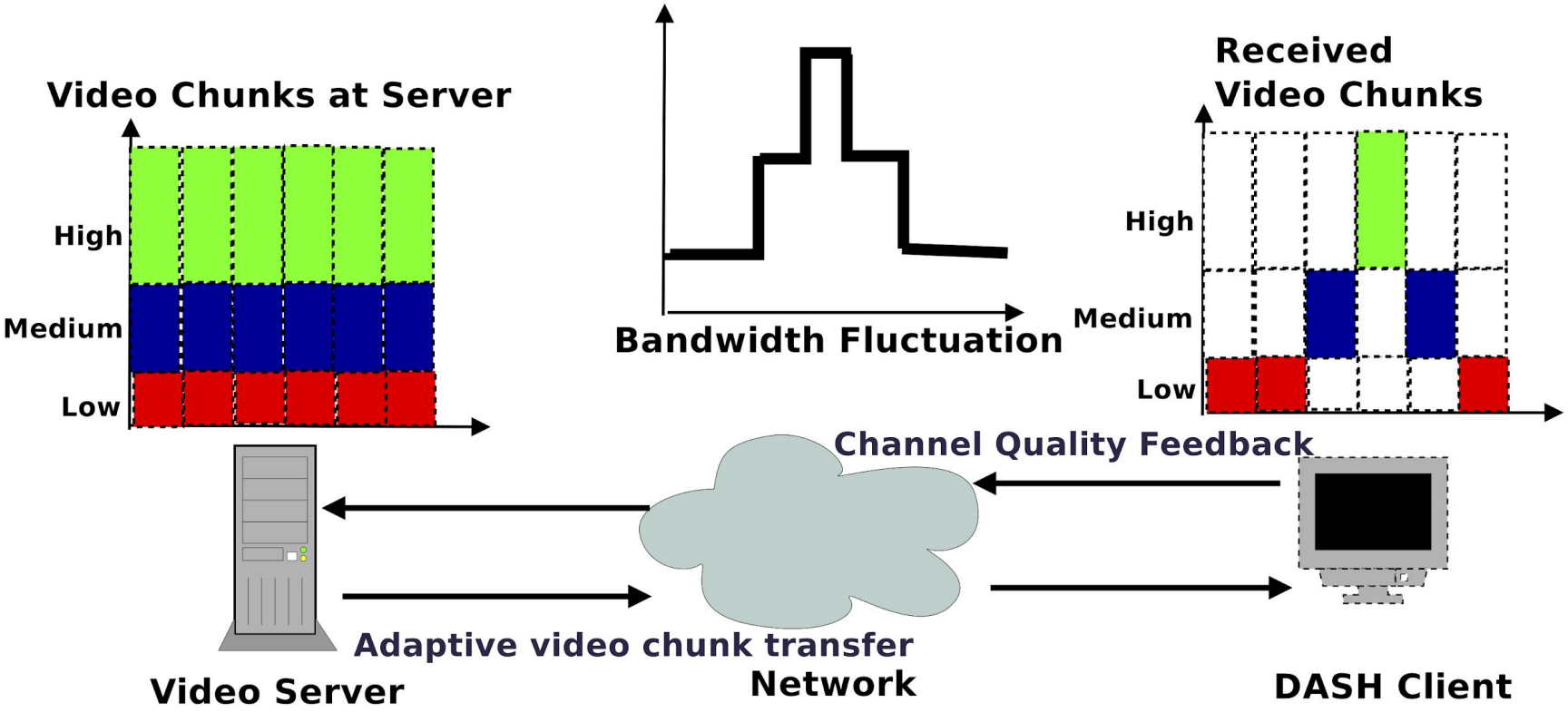
Federated Adaptive Bitrate Live Streaming over Locality Sensitive Playback Coalitions

Abhijit Mondal
Sandip Chakraborty
Indian Institute of Technology Kharagpur

NAI 2020

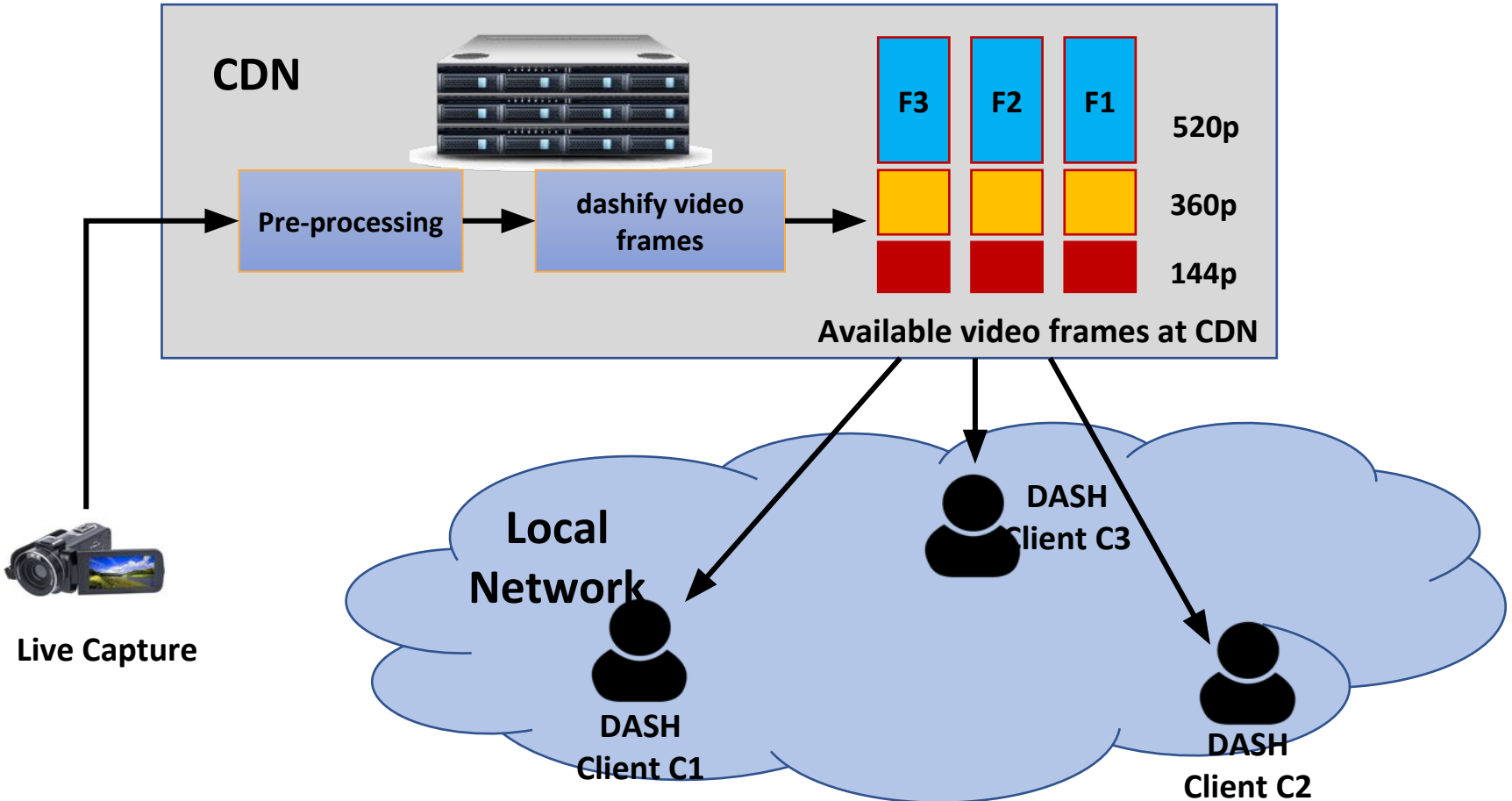
August 03, 2020

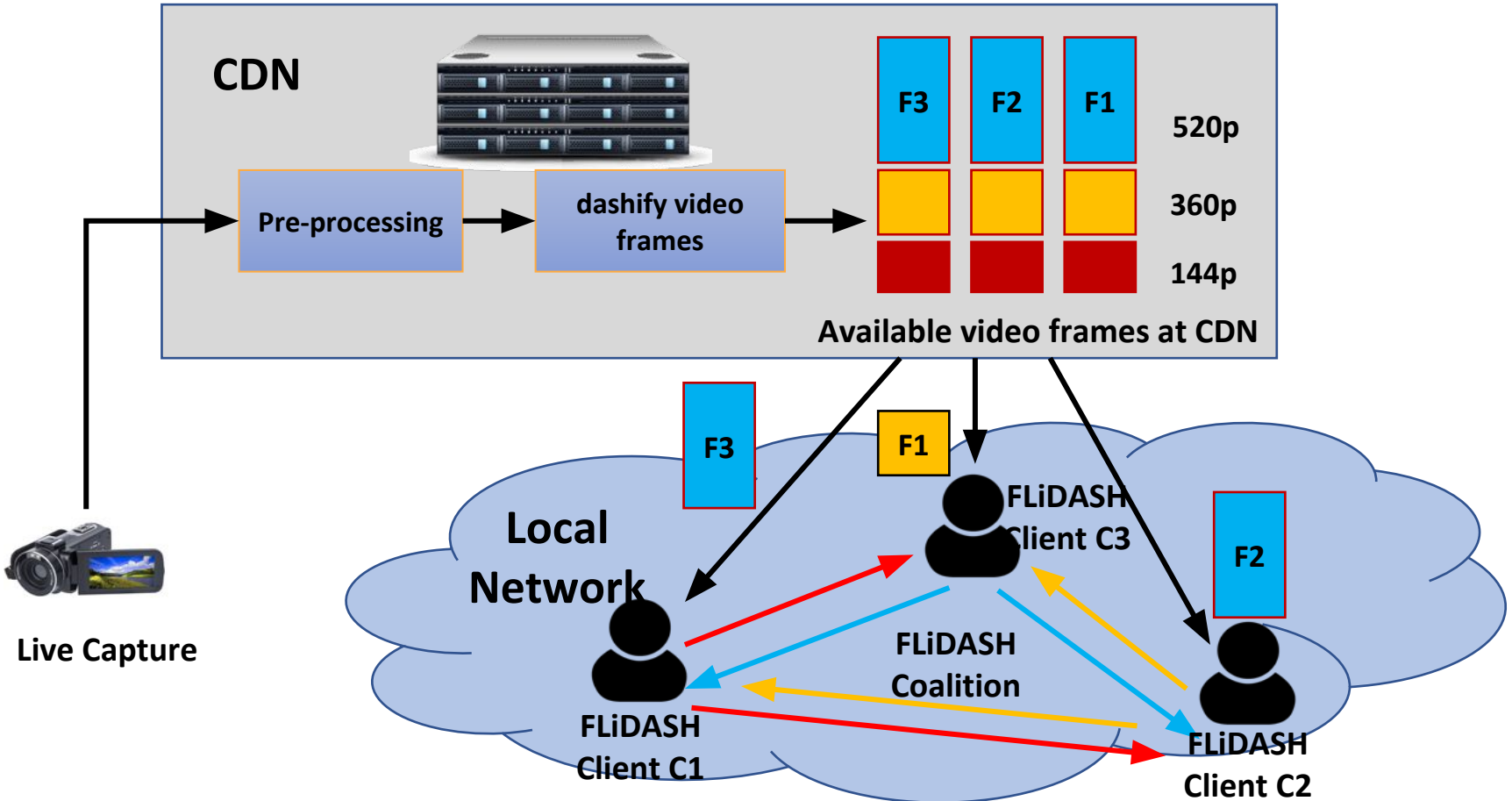
Dynamic Adaptive Streaming over HTTP(S)



Video Streaming

Video Streaming - Live


CDN
Pre-processing
dashify video frames
F3
F2
F1
520p
360p
144p
Available video frames at CDN
Live Capture
Local Network
DASH Client C1
DASH Client C3
DASH Client C2


CDN
Pre-processing
dashify video frames
F3
F2
F1
520p
360p
144p
Available video frames at CDN
Live Capture
F3
F1
F2
Local Network
FLiDASH Client C3
FLiDASH Coalition
FLiDASH Client C1
FLiDASH Client C2

Challenges

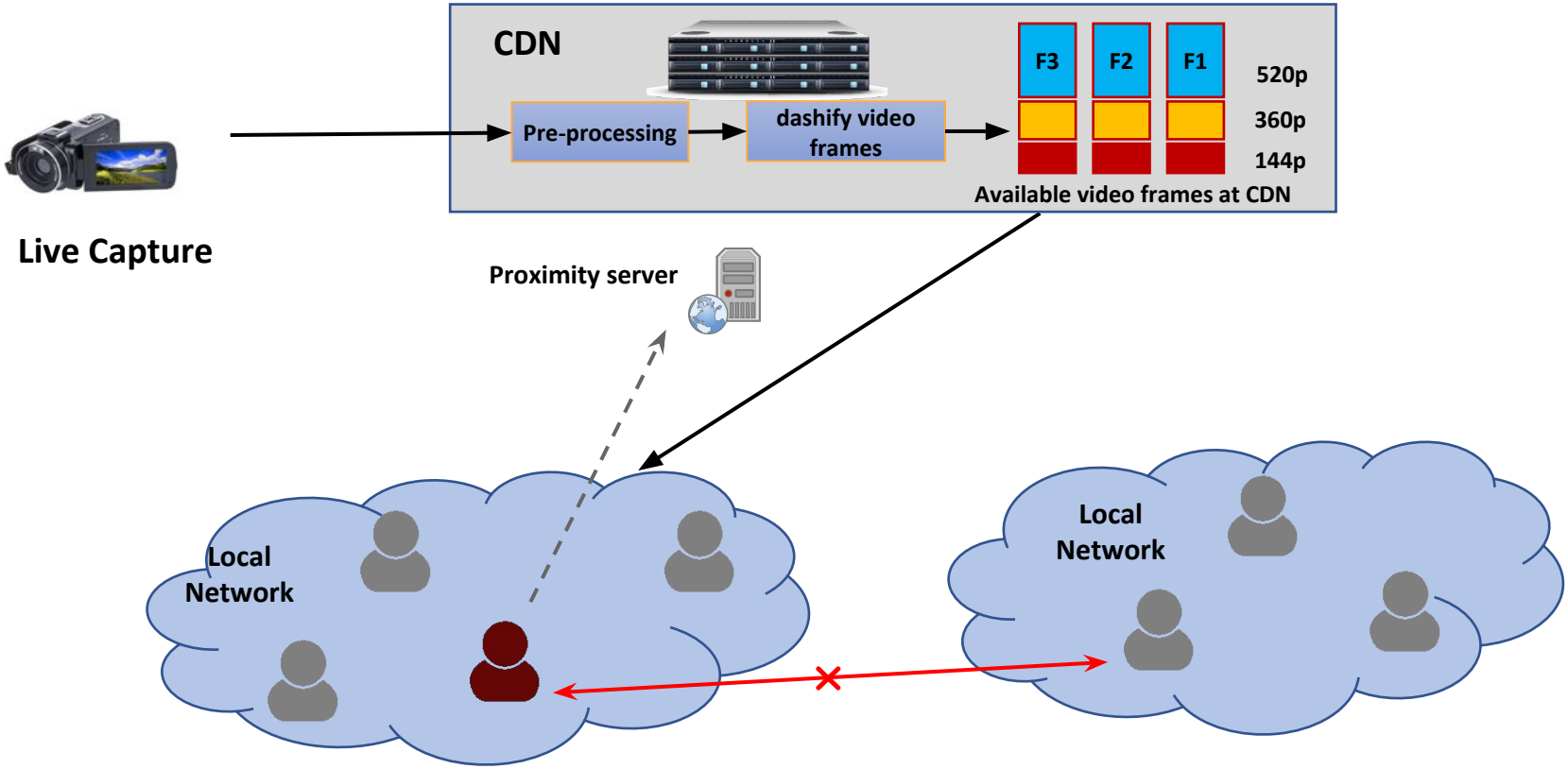
- Nearby player discovery
  - No Internet uses overhead
    - Share among players from same LAN
- A scheme to share video data
  - No free rider
  - Improve QoE
    - Form coalition
    - Share within the coalition only
    - Download and share policy

Nearby Player Discovery

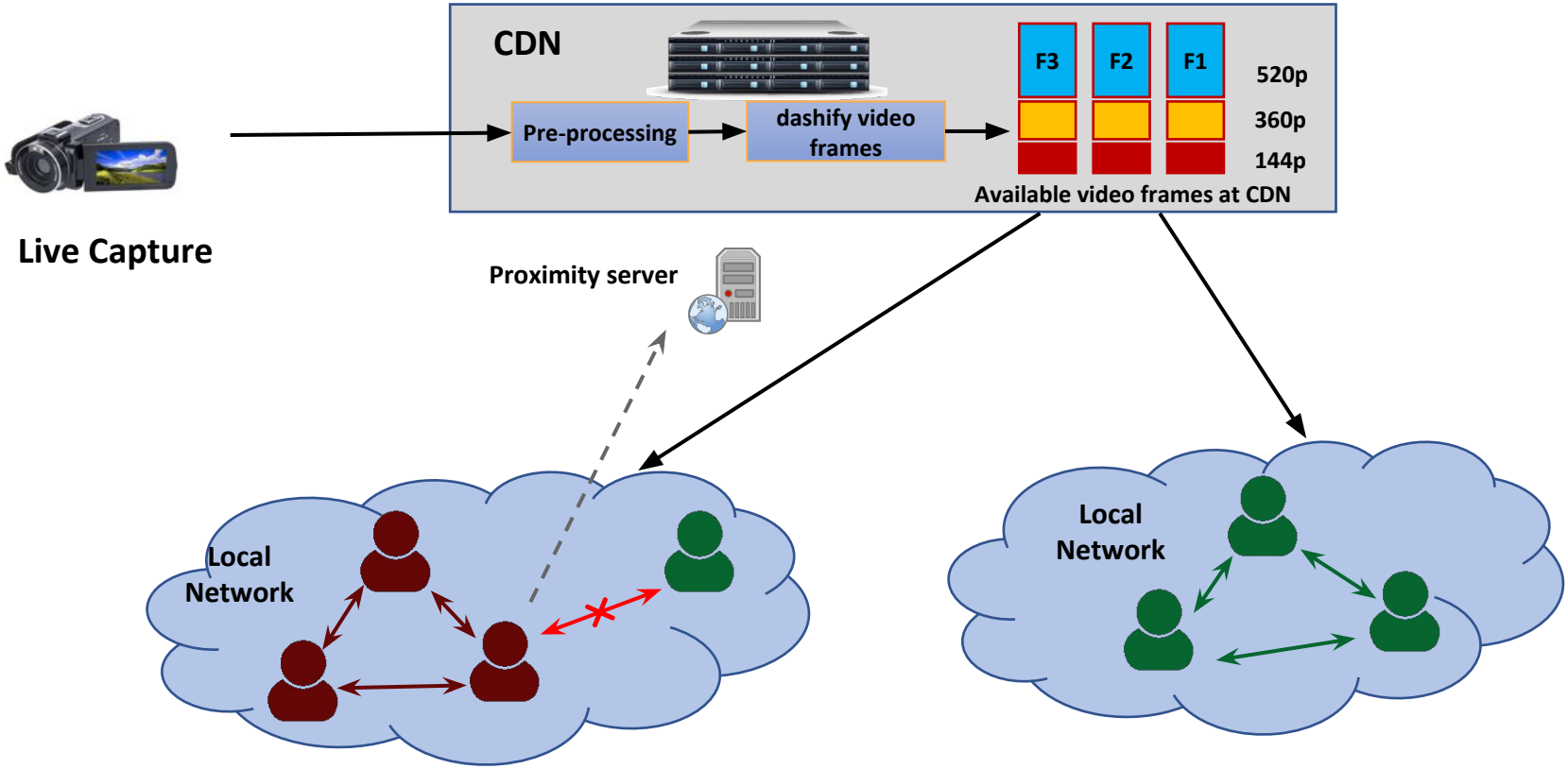
1. LAN broadcast
    a. Players are in same broadcast domain

2. Proximity server
    a. external server
        i. Provides potential neighbours
    b. ALTO server
        i. Determine feasibility of communication

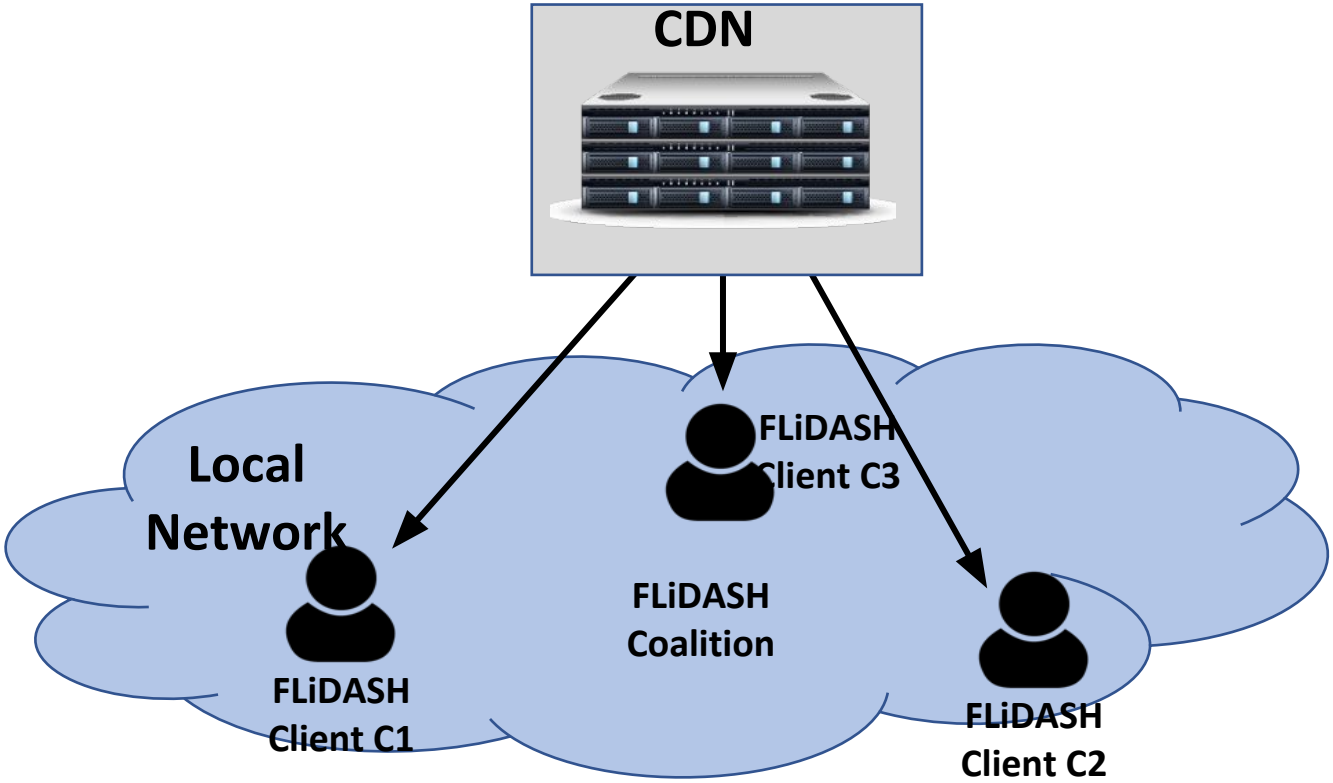
Streaming Coalition Formation



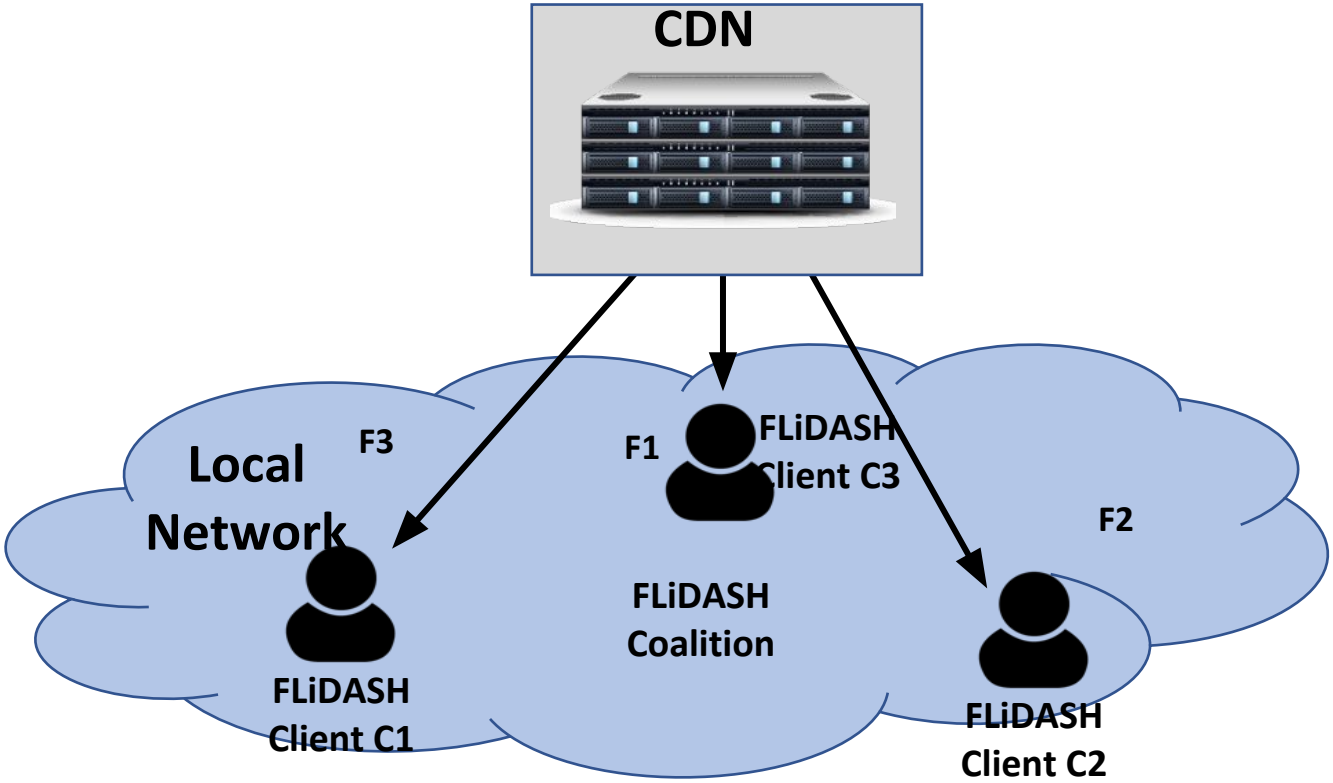
Streaming Coalition Formation



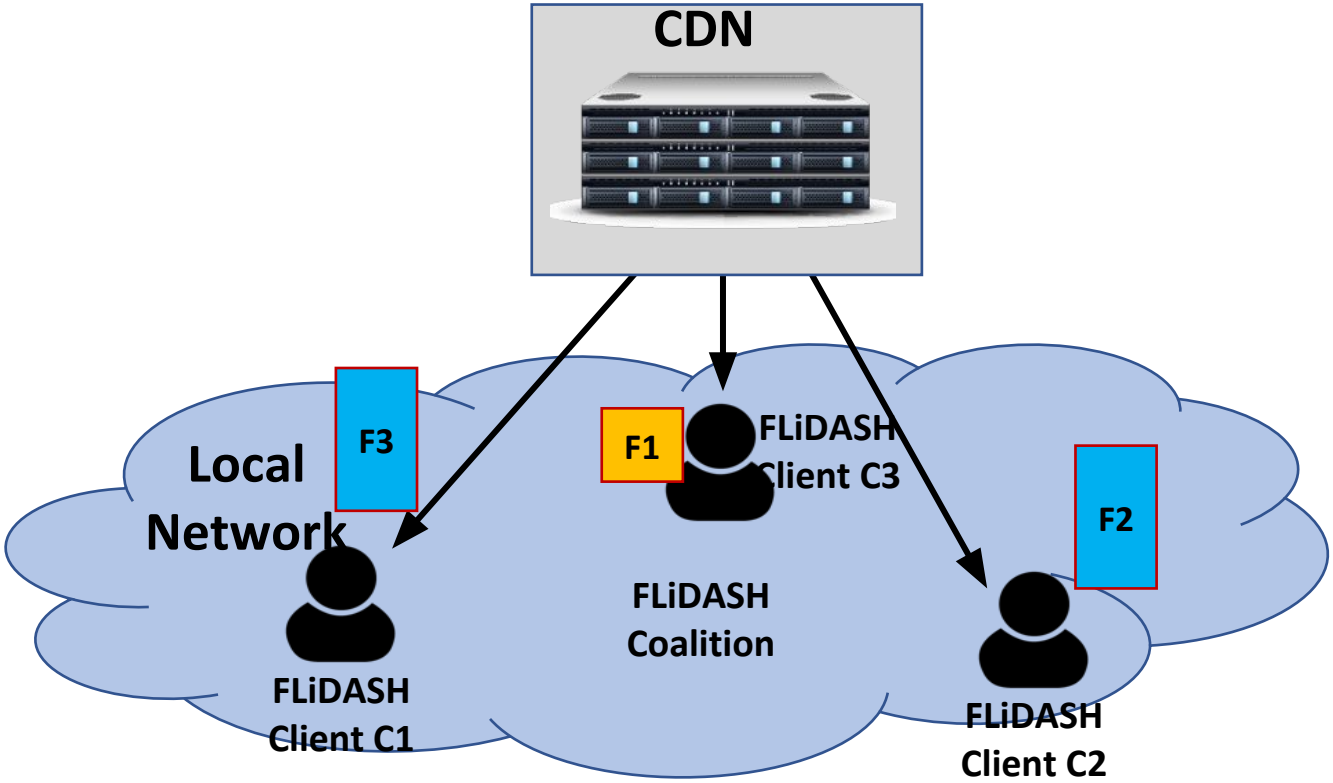
Download and Share Policy



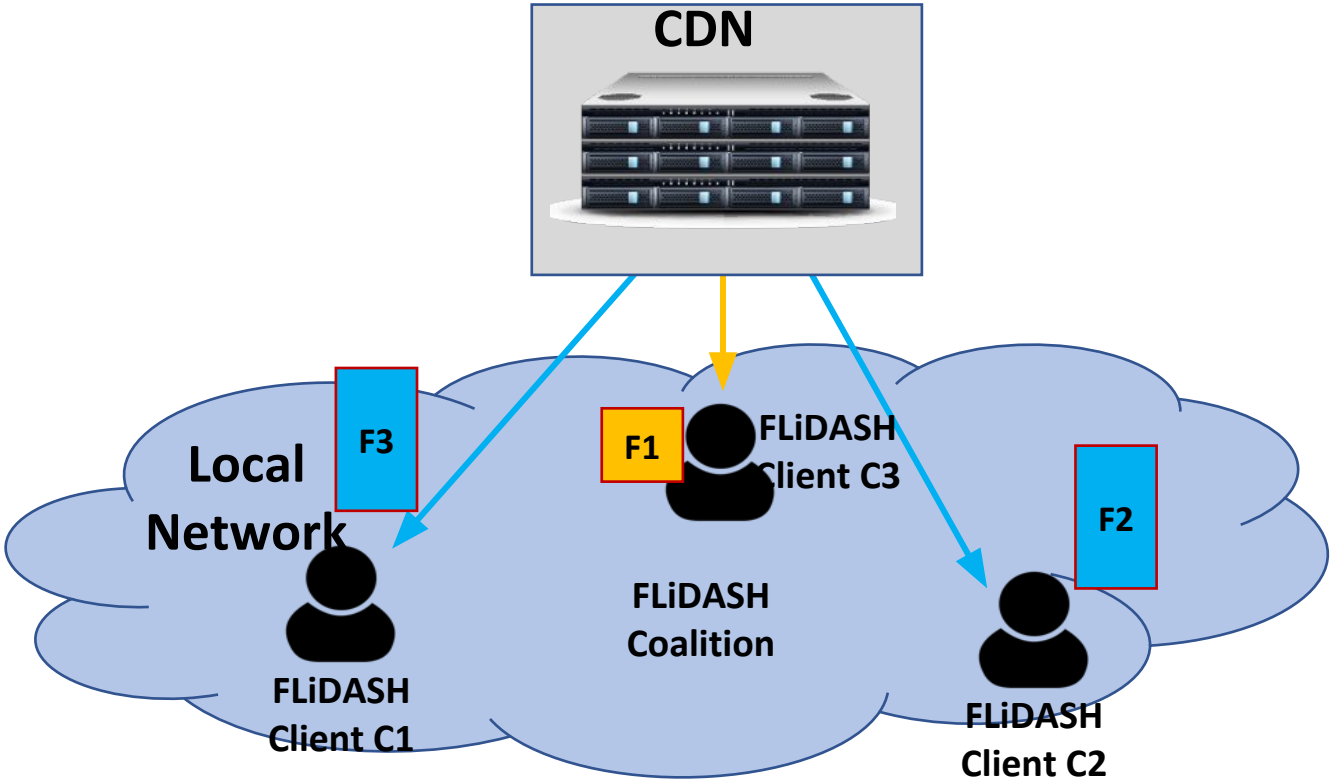
Download and Share Policy



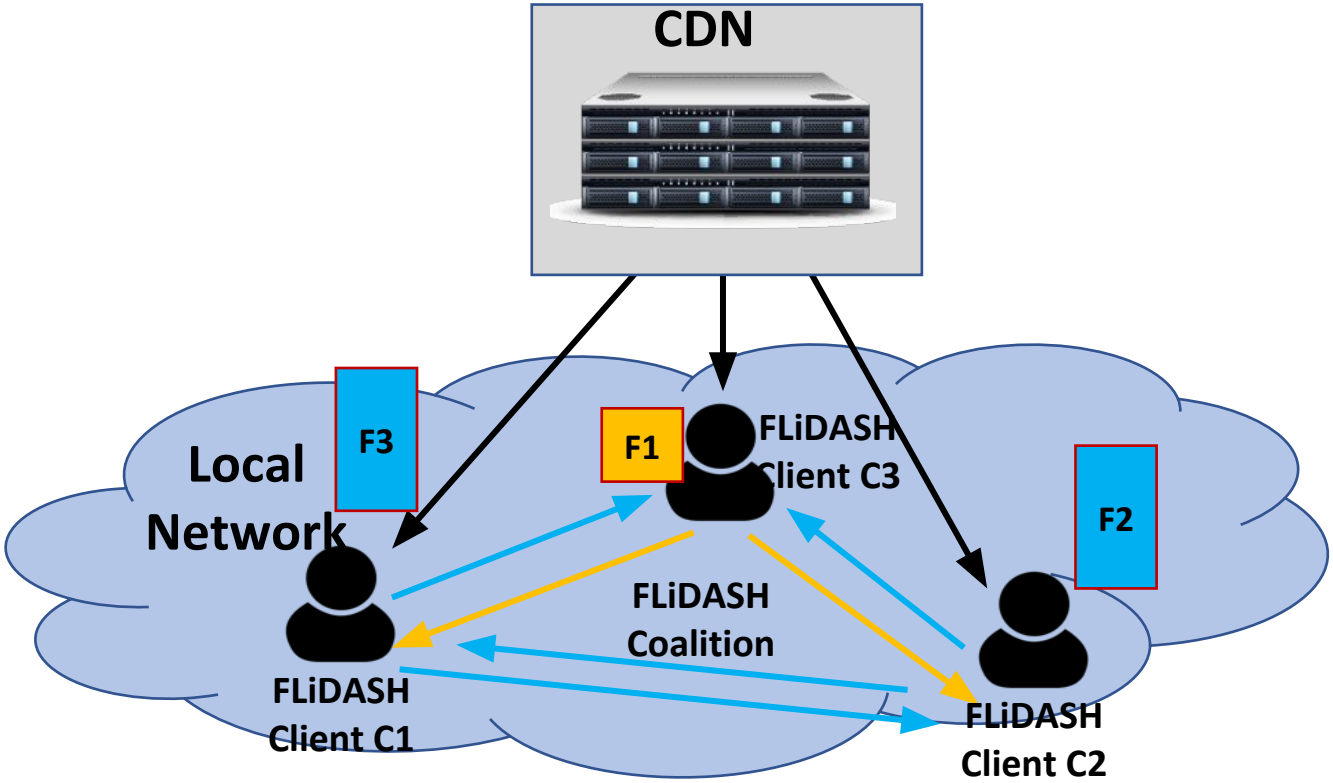
Download and Share Policy



Download and Share Policy



Download and Share Policy



Download and Share Policy

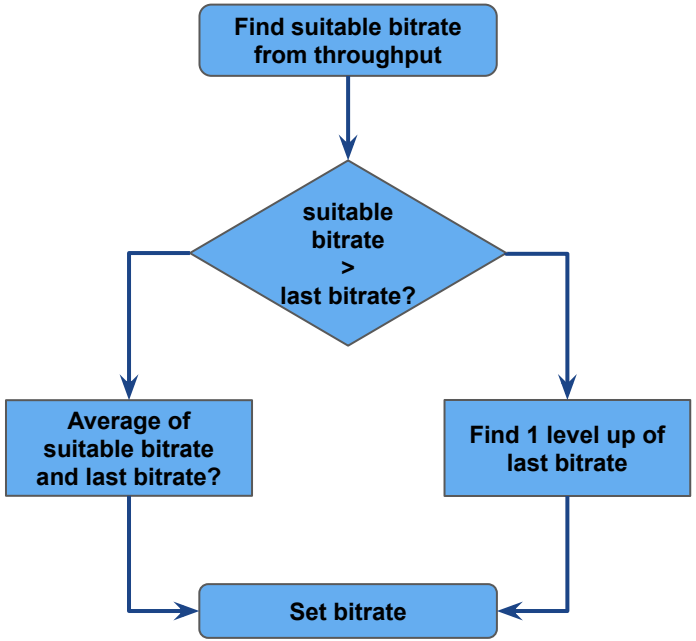
Downloader Selection

$$P_i = \underset{x \in G_p}{\operatorname{argmin}} \left(\mathcal{I}_x - \mathcal{D}q_x - \mathcal{D}l_x \right)$$

Downloader (P_i) selection

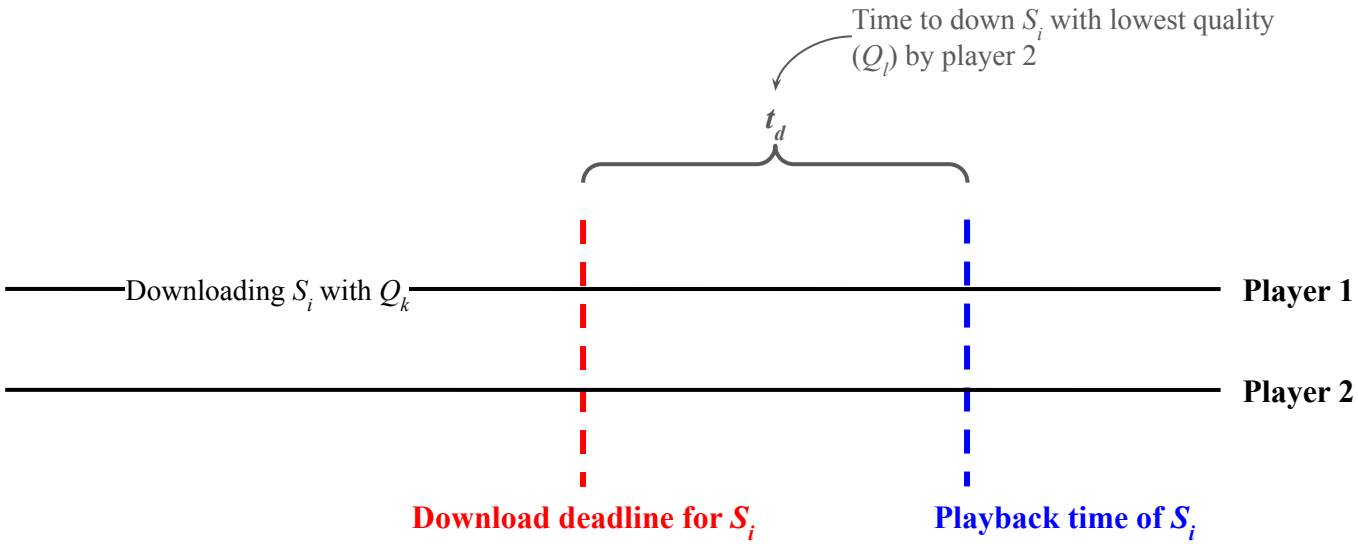
1. Maximum *idle* time (I_x)
2. Minimum pending downloads (Dq_x)
3. Minimum remaining download completion time (Dl_x)

Quality Selection Algorithm

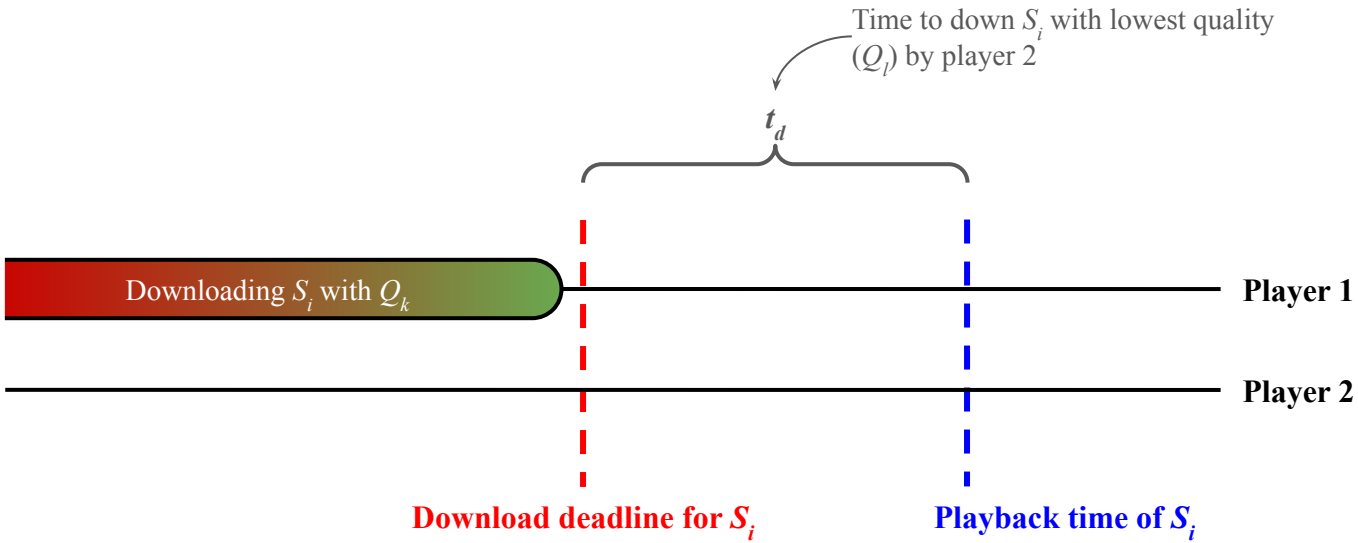


- Quality Selection
  - Increase quality **additively** (1 step at a time)
  - Decrease quality **multiplicatively**
- Advantages
  - No **sudden changes** in quality
  - Improves **QoE**
  - Low **stalls**

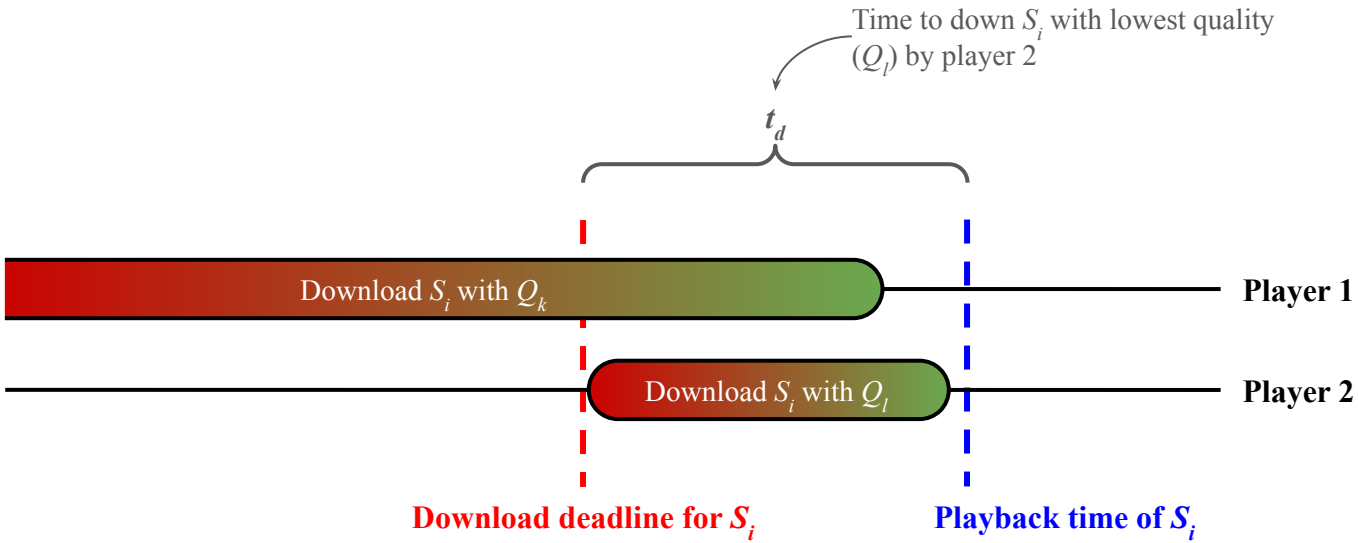
Forceful Self-Download



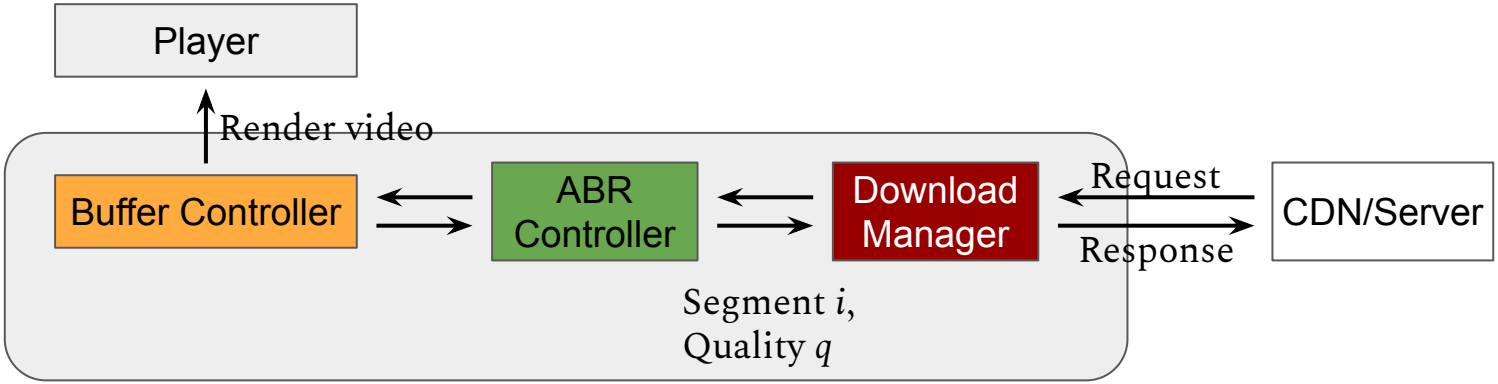
Forceful Self-Download



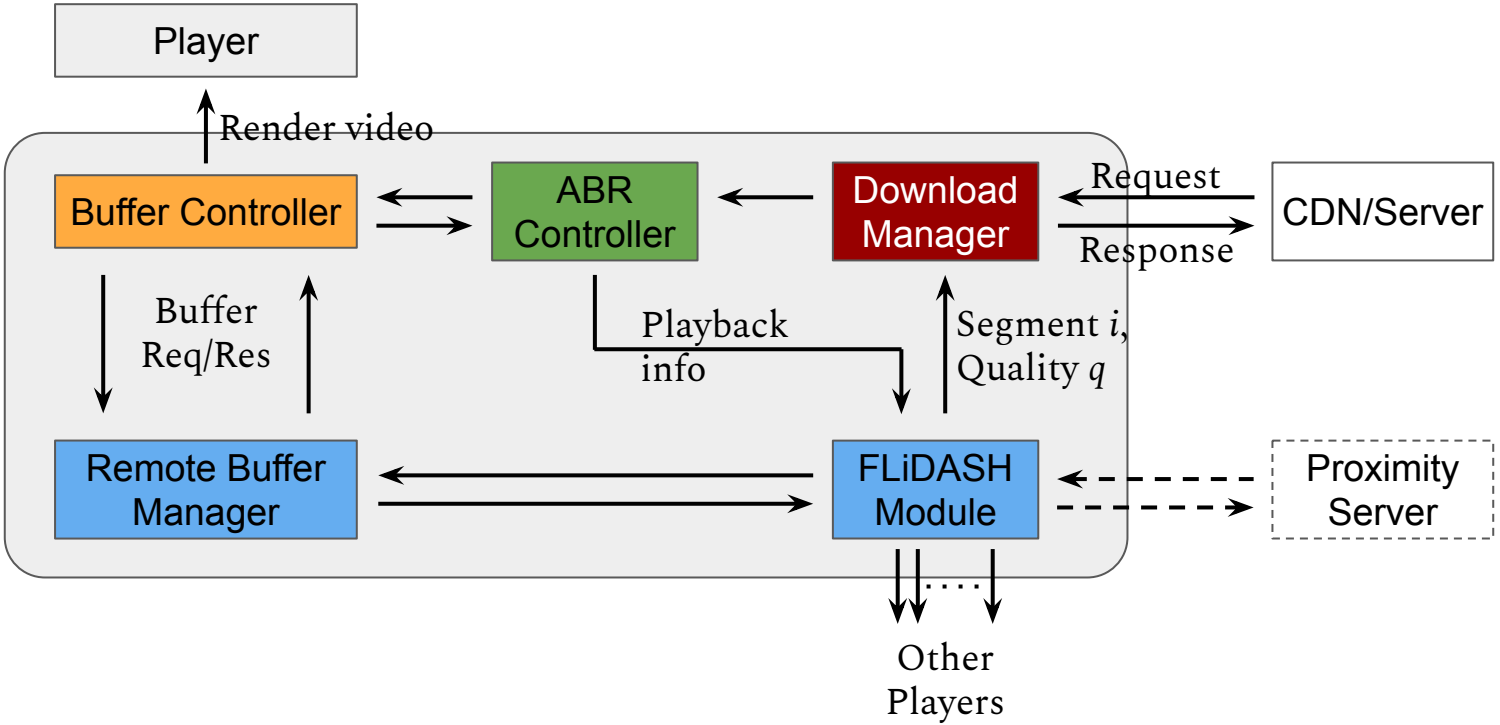
Forceful Self-Download



System Architecture



System Architecture



Experimental Environment

- We developed **event driven simulator** based of pensieve simulator
- Multiple player can be **simulated simultaneously**
- Players communication is emulated based on a **reference network**
- All the modules are **plugable**
- We used prewritten ABR module from **DASH.JS**
- **Mahimahi[1] network trace** are use to emulated link condition between CDN and player

[1]. R. Netravali, A. Sivaraman, S. Das, A. Goyal, K. Winstein, J. Mickens, and H. Balakrishnan, “Mahimahi: Accurate record-and-replay for HTTP,” in USENIX ATC, 2015, pp. 417–429.

Experimental Setup

- Compared ABRs like **BOLA**[2]**, MPC**[3] **and Pensieve**[4] and **distributed hash table (DHT)** based peer assisted streaming system[5]
- Live streaming emulated with **58 dashified videos** with total duration of **45 hours**
- Used **autonomous system data** from **SNAP** database as reference network[6]
- Broadband traces from **FCC**[7] and **3G/HSDPA** mobile dataset from Norway[8] used as trace to simulate link between player and CDN

[2] K. Spiteri, R. Sitaraman, and D. Sparacio, “From theory to practice: Improving bitrate adaptation in the DASH reference player,” in ACM MMSys, 2018, pp. 123–137.
[3] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli, “A Control-theoretic Approach for Dynamic Adaptive Video Streaming over HTTP,” in ACM SIGCOMM, New York, New York, USA, 2015.
[4] H. Mao, R. Netravali, and M. Alizadeh, “Neural Adaptive Video Streaming with Pensieve,” in ACM SIGCOMM, New York, USA, 2017.
[5] H. Shen, Z. Li, and J. Li, “A DHT-aided chunk-driven overlay for scalable and efficient peer-to-peer live streaming,” IEEE TPDS, vol. 24, no. 11, pp. 2125–2137, 2013.
[6] J. Leskovec, J. Kleinberg, and C. Faloutsos, “Graphs over time: Densification laws, shrinking diameters and possible explanations,” in ACM SIGKDD, 2005, pp. 177–187.
[7] F. C. Commission, “Raw Data - Measuring Broadband America - Eighth Report,” https://www.fcc.gov/reports-research/reports/measuring-broadband-america/raw-data-measuring-broadband-america-eighth, 2018, [Online; accessed 29-March-2019].
[8] H. Riiser, P. Vigmostad, C. Griwodz, and P. Halvorsen, “Commute path bandwidth traces from 3G networks: Analysis and applications,” in ACM MMSys, 2013, pp. 114–118.

QoE Measurement

$$QoE = \frac{\alpha}{N}\sum_{n=1}^{N}\mathcal{F}(\mathcal{Q}_n) - \frac{\beta}{N-1}\sum_{n=2}^{N}|\mathcal{F}(\mathcal{Q}_n) - \mathcal{F}(\mathcal{Q}_{n-1})| - \gamma\mathcal{T}_n$$

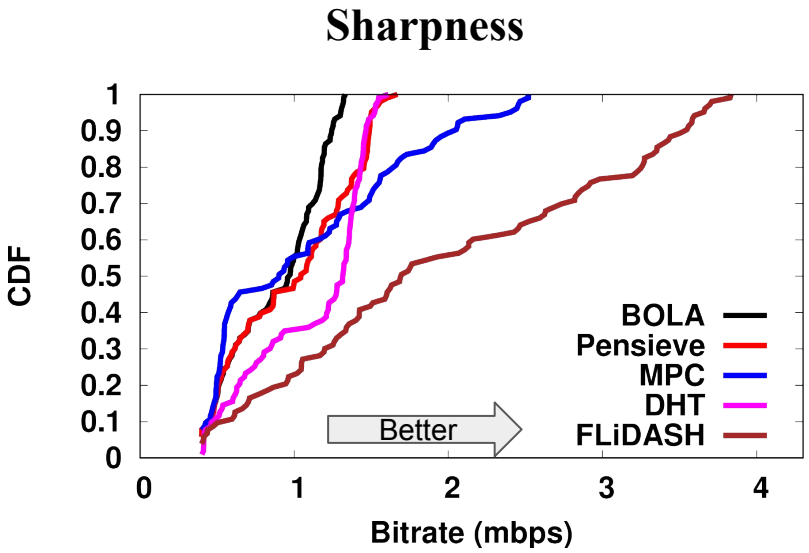
$\mathcal{Q}_n$: bitrate level of nth segment

$\mathcal{T}_n$: Overall stall

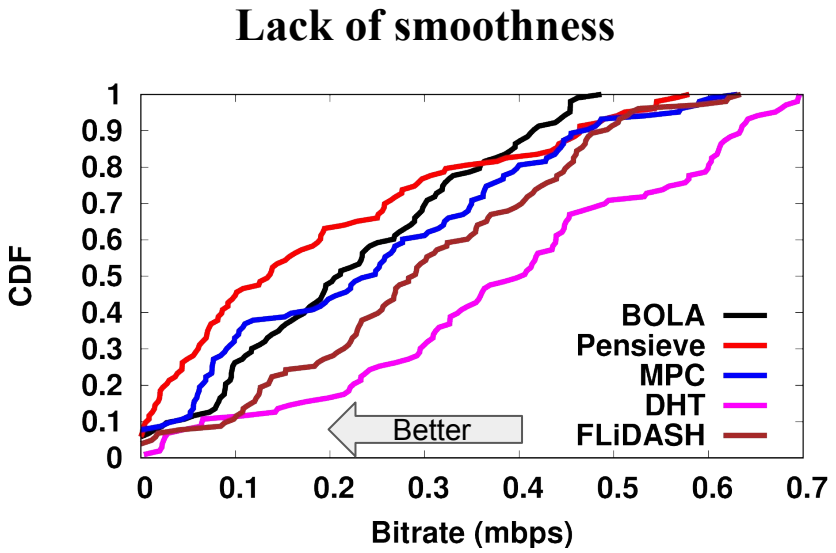
$\mathcal{F}(\cdot)$: Quality function. We use F(Qn) as the bitrate in mbps

α, β and γ are normalization coefficient
We use
$\alpha = \beta = 1$
$\gamma = 4.3$

Results: QoE Components

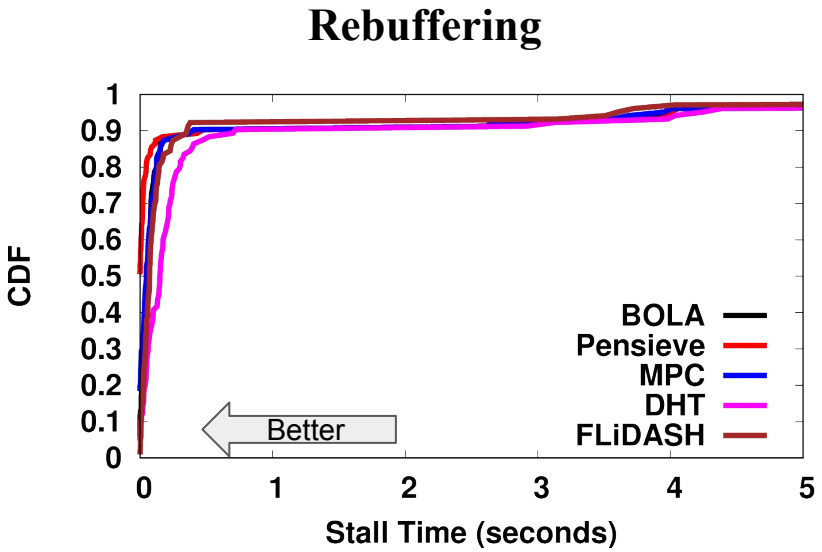


Average bitrate played by each player

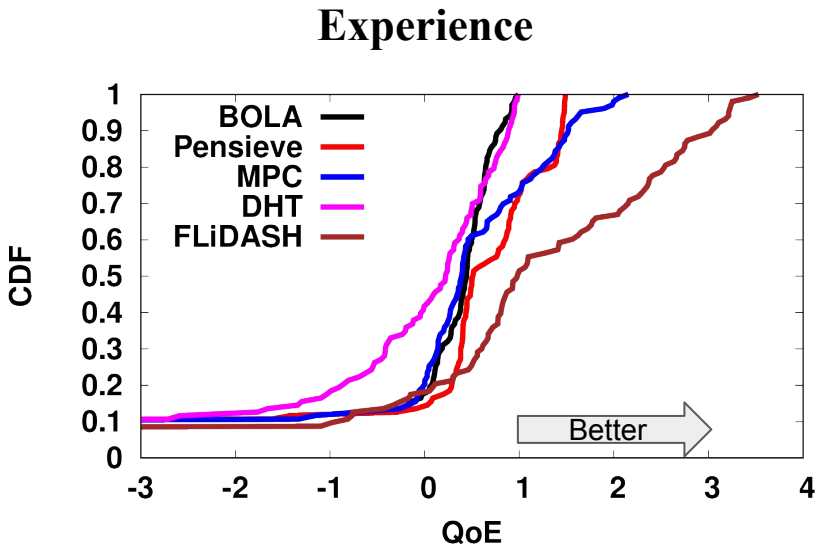


Average bitrate varied for each player

Results: QoE Components

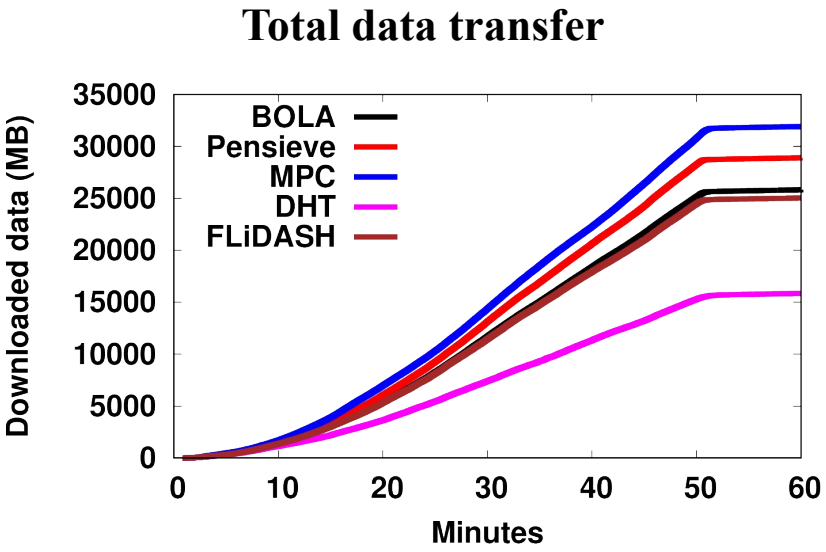


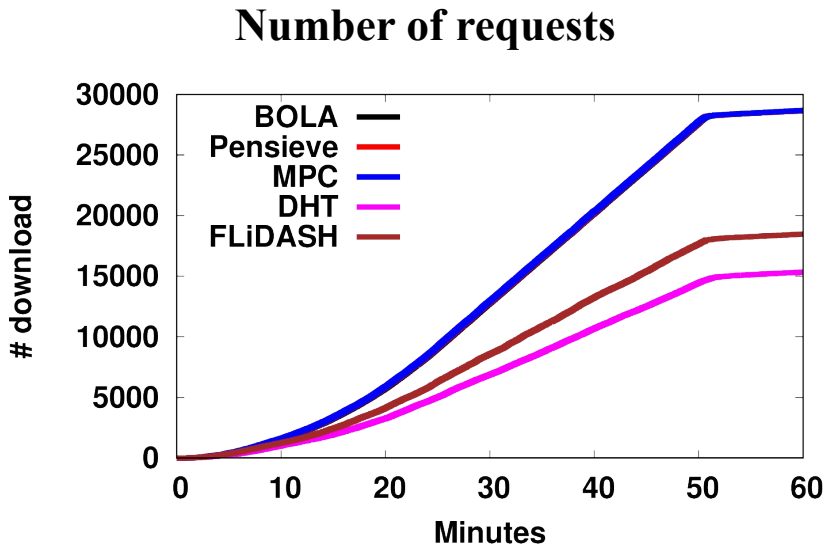
Average rebuffering time for each player (each segment)



Overall QoE achieved by each player

Results: Server Load





Data downloaded from (CDN) server

Evaluation: Benefit of Using FLiDASH

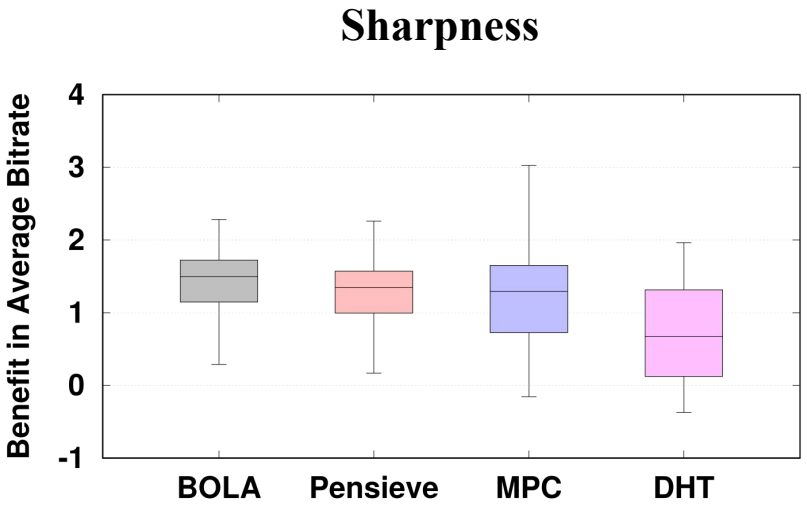
Compare the benefit of FLiDASH over existing systems

$$Ben(S) = \frac{\mathcal{G}_f - \mathcal{G}_o}{|\mathcal{G}_o|}$$

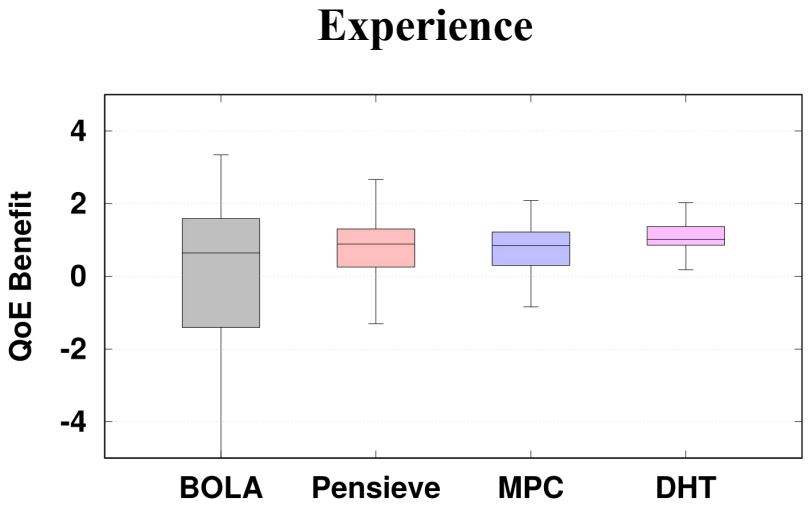
$\mathcal{G}_f$: Instant value of FLiDASH

$\mathcal{G}_o$: Instant value of system ABR

Results: Benefit of Using FLiDASH



Benefit measured in term of Average bitrate(Mbps)



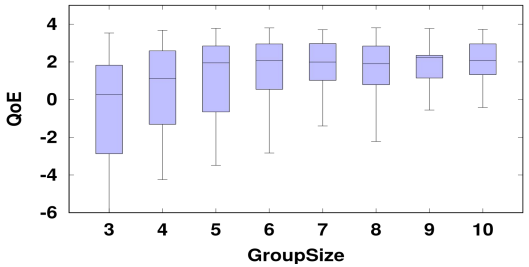
Benefit measured in terms of QoE

Evaluation: Effect of Coalition-Size

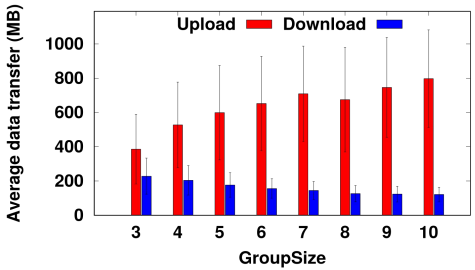
- We change the coalition-size from **3 to 10**
- Measure the **QoE, data transfer** by each players and **fairness** among players
- We measure fairness using **Jain fairness index**[9] on QoE

[9] R. Jain, A. Durresi, and G. Babic, “Throughput fairness index: An explanation,” in ATM Forum contribution, vol. 99, no. 45, 1999.

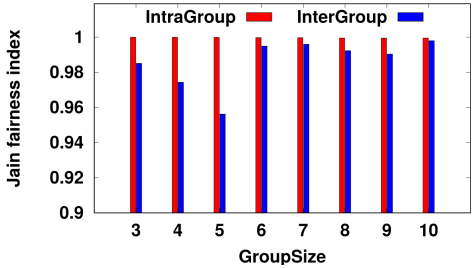
Results: Effect of Coalition Size



Overall QoE per player



Average data transfer per player

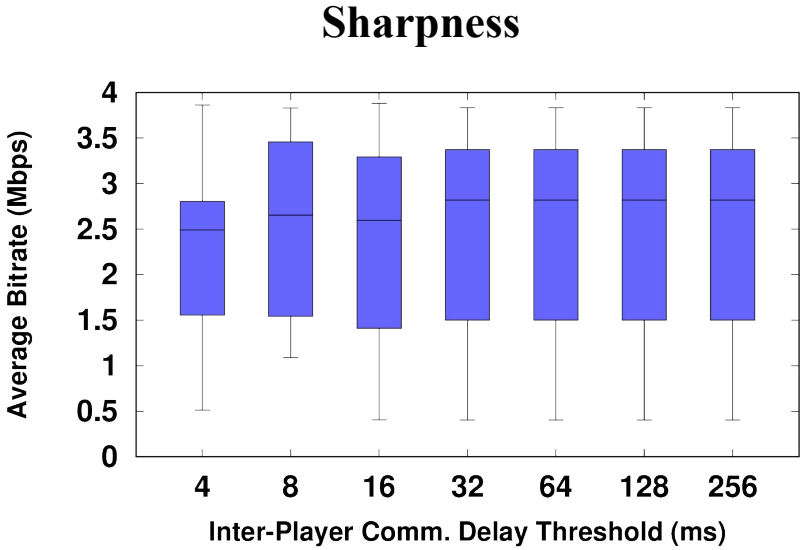


Jain fairness index in terms of QoE

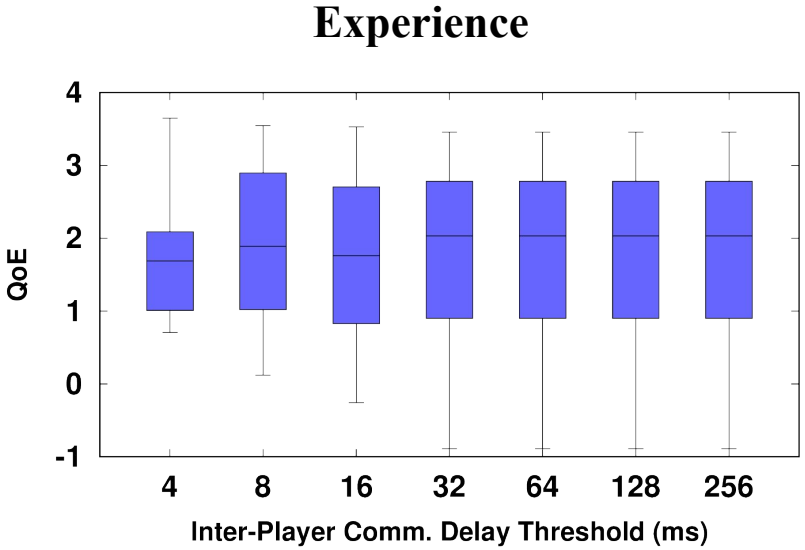
Evaluation: Comm. Delay Threshold (t_d)

- Change **communication delay threshold** from **4 ms to 256 ms**
- Measure the **average bitrate** and **QoE**

Results: Comm. Delay Threshold (t_d)



Average bitrate played by each player



Overall QoE achieved by each player

Conclusion

- A middlebox-free collaborative adaptive live streaming system
- Better overall QoE while reducing the network traffic
- A highly-scalable architecture for mass-scale live streaming
- Incurs low over-head to the backbone network

Thank You