FDCHAS

A Distributed Approach for Bitrate Selection in HTTP Adaptive Streaming

Alpen-Adria-Universität Klagenfurt, 13 December 2018.

Abdelhak Bentaleb, PhD Student (NUS), Intern (Bitmovin)

bentaleb@comp.nus.edu.sg

Authors

Abdelhak Bentaleb (NUS)
bentaleb@comp.nus.edu.sg

Ali C. Begen (OzU)
ali.begen@ozyegin.edu.tr

Saad Harous (UAEU)
harous@uaeu.ac.ae

Roger Zimmermann (NUS)
rogerz@ comp.nus.edu.sg

Paper (ACM MM 2018)

Session: System-1 (Video Analysis & Streaming) MM'18, October 22-26, 2018, Seoul, Republic of Korea

A Distributed Approach for Bitrate Selection in HTTP Adaptive Streaming

Abdelhak Bentaleb[⋆], Ali C. Begen[*], Saad Harous[+], and Roger Zimmermann[⋆]
[⋆]National University of Singapore, [*]Ozyegin University, [+]United Arab Emirates University
{bentaleb,rogerz}@comp.nus.edu.sg, ali.begen@ozyegin.edu.tr, harous@uaeu.ac.ae

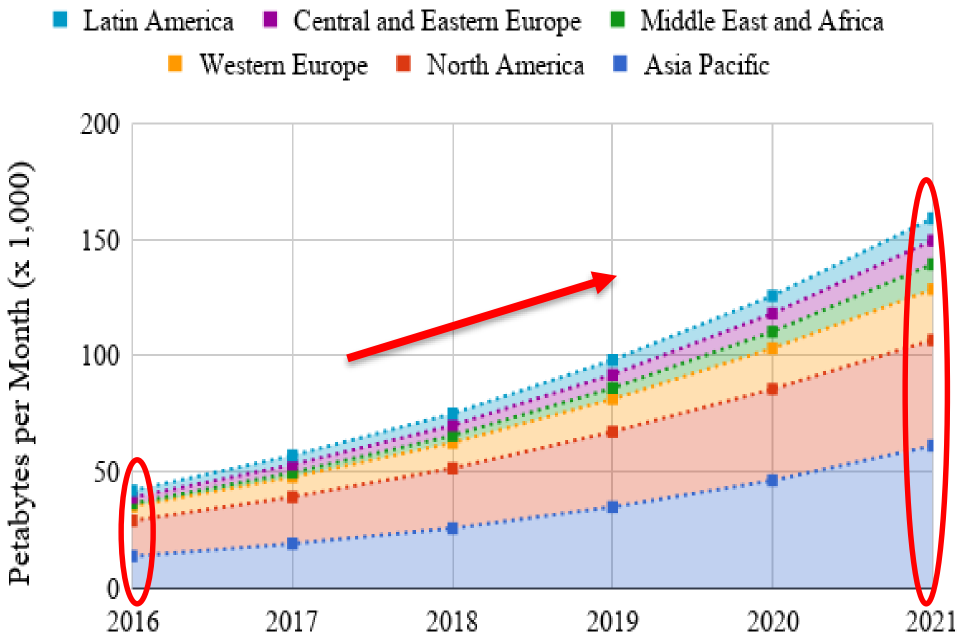
Outline

1. Context and Background
2. Literature Review
3. Problem Statement and Motivations
4. Proposed Solution
5. Evaluation and Experimental Results
6. Demo
7. Conclusions and Future Directions

Context and Background

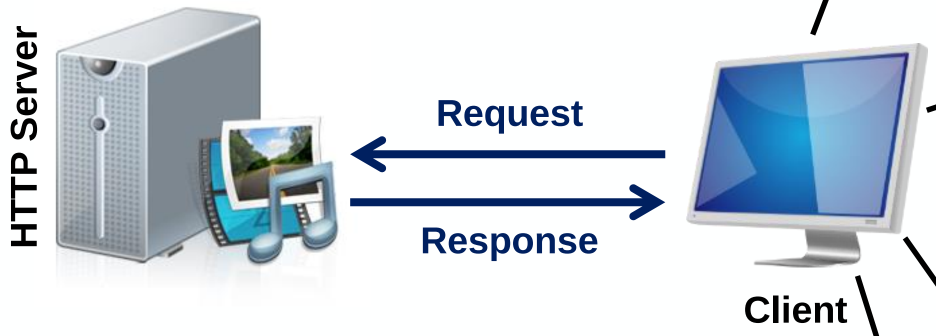
Video is Predominant on the Internet

- Cisco reported in their annual VNI:
  - In 2016, 67% of the global Internet traffic was video, with a projection to reach 80% by 2021
- Popular services:
  - YouTube (14.0%)
  - Netflix (34.9%)
  - Amazon Video (2.6%)
  - Hulu (1.4%)
- All these are delivered as OTT



HAS Working Principle

Smart and Selfish Clients



- Client fetches and parses the manifest
- Client uses the OS-provided HTTP stack (HTTP may run over TCP or QUIC)
- Client uses the required decryption tools for the protected content

Client monitors and measures

- Size of the playout buffer (both in bytes and seconds)
- Chunk download times and throughput
- Local resources (CPU, memory, window size, etc.)
- Dropped frames

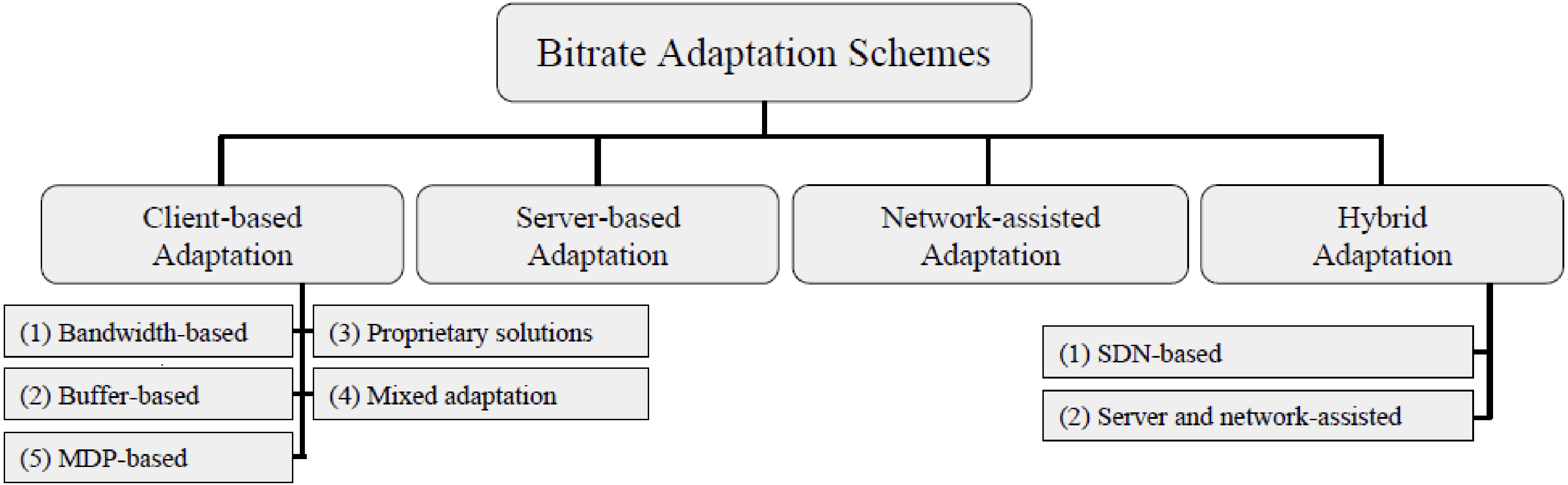
Client performs adaptation

Client measures and reports metrics for analytics

Literature Review

ABR Schemes Classification

The entity of the system where the logic is implemented



Reading: Abdelhak Bentaleb *et al* "A Survey on Bitrate Adaptation Schemes for Streaming Media over HTTP," IEEE COMST 2018

Problem Statement and Motivations

Does HTTP adaptive streaming simply work?

Mostly yes, when streaming clients compete with other types of traffic

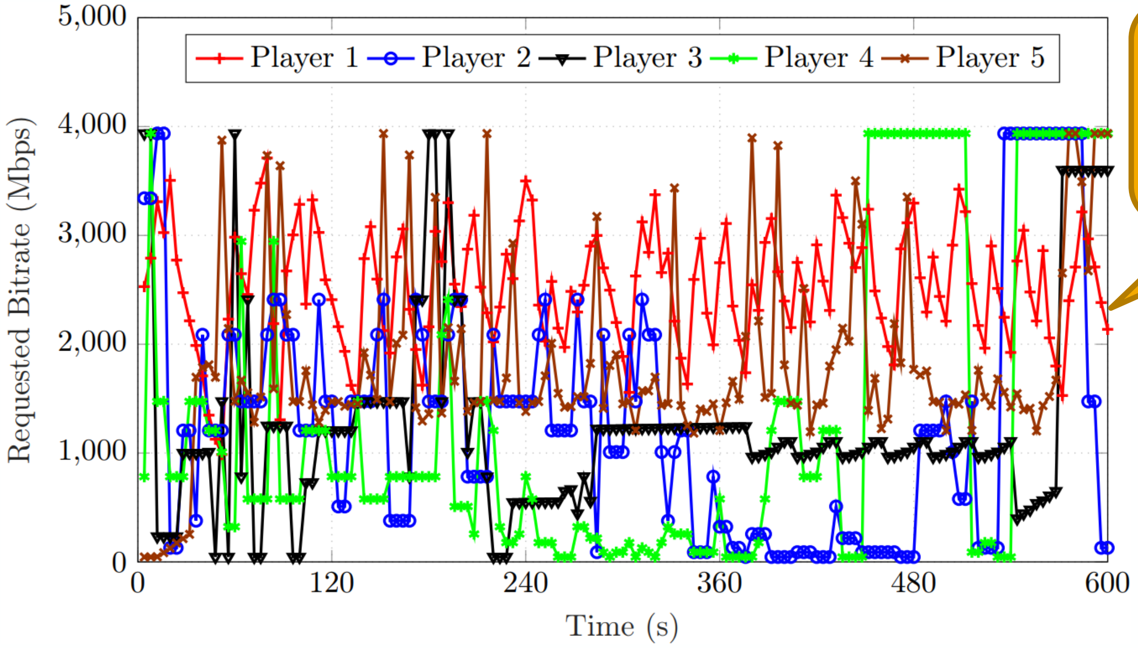
Not really, when streaming clients compete with each other

Streaming clients interact with each other forming an “accidental” distributed control-feedback system

- Multiple screens within a household
- ISP access/aggregation links
- Small cells in stadiums and malls

Selfishness Hurts Everyone

Five dash.js Streaming Clients Sharing a 25 Mbps Link



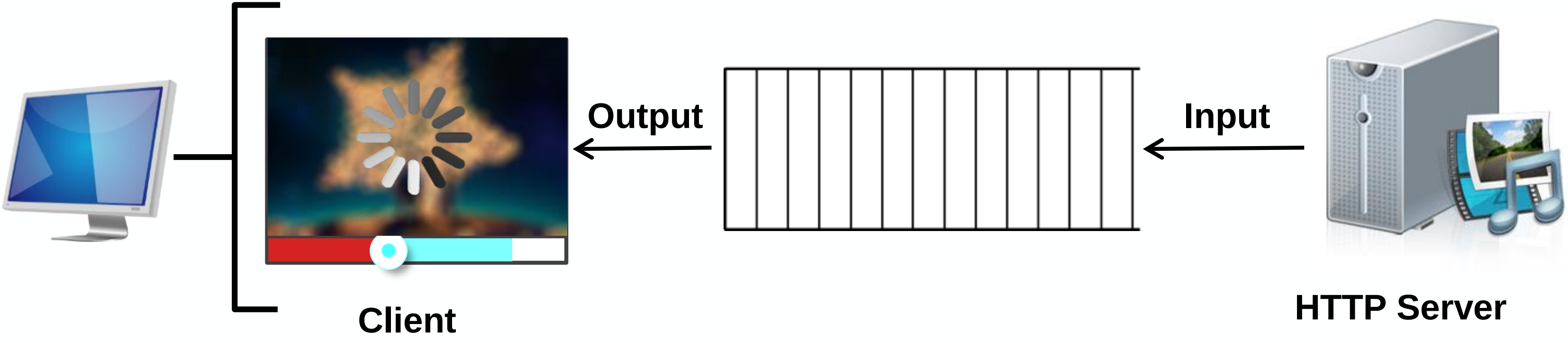
- Video instability
- QoE unfairness
- Underutilization or oversubscription

Five clients requesting animation videos with bitrate levels varying from 50 to 3,960 Kbps and competing for 25 Mbps shared bandwidth in a home network.

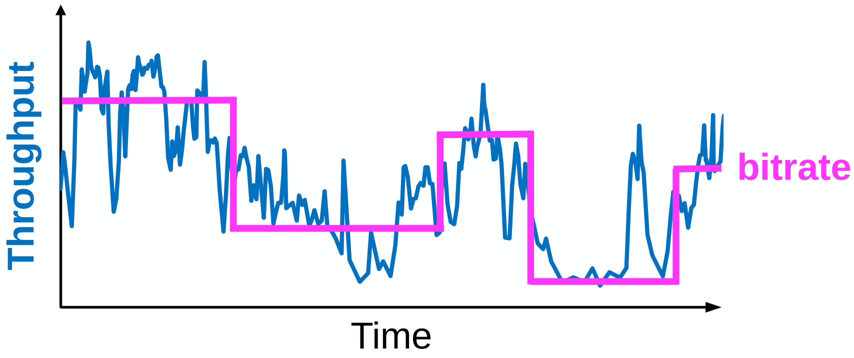
Reading– Tutorial 1 Delivering Traditional and Omnidirectional Media

Selecting an Appropriate Bitrate is Tricky

Bandwidth Overestimation > Wrong Bitrate Selection

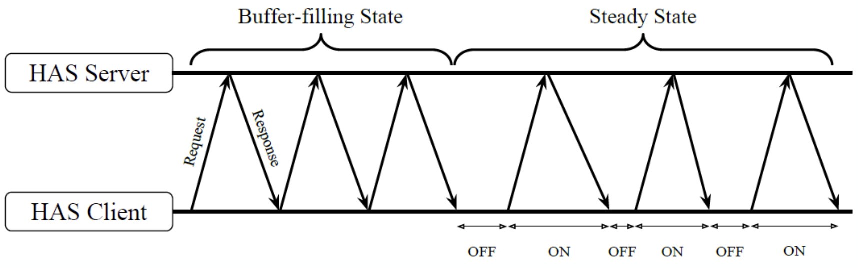


Adaptive Bitrate (ABR) Algorithms



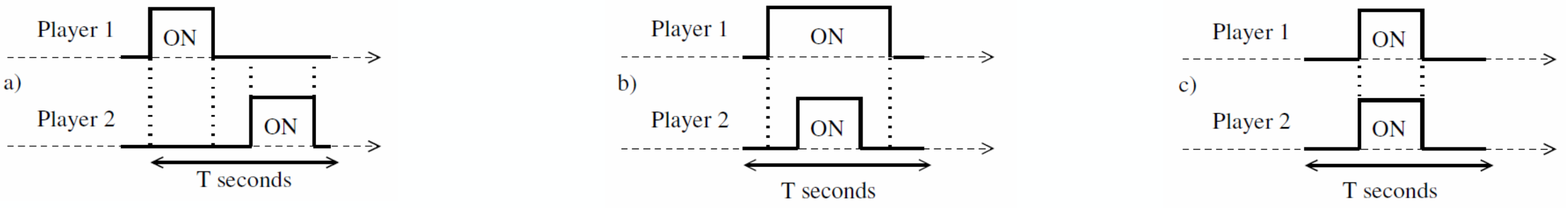
Understanding the Root Cause

Two Competing Clients



- Depending on the timing of the ON periods:
  - Unfairness, underutilization and/or instability may occur.
  - Clients may grossly overestimate their fair share of the available bandwidth.

Clients cannot figure out how much bandwidth to use until they use too much (Just like TCP)



Reading: Saamer Akhshabi *et al* "What happens when HTTP adaptive streaming players compete for bandwidth?," ACM NOSSDAV 2012

How to Solve the Issues?

Fix the clients and/or the transport

- Use a better adaptation algorithm like PANDA or BOLA
- Use machine learning or deep learning like Pensieve
- Improve the HTTP/TCP stack, try out the alternatives
- Adopt ideas from game/consensus theory (FDCHAS) NEW

Get support from the network

- QoS in the core/edge
- SDN

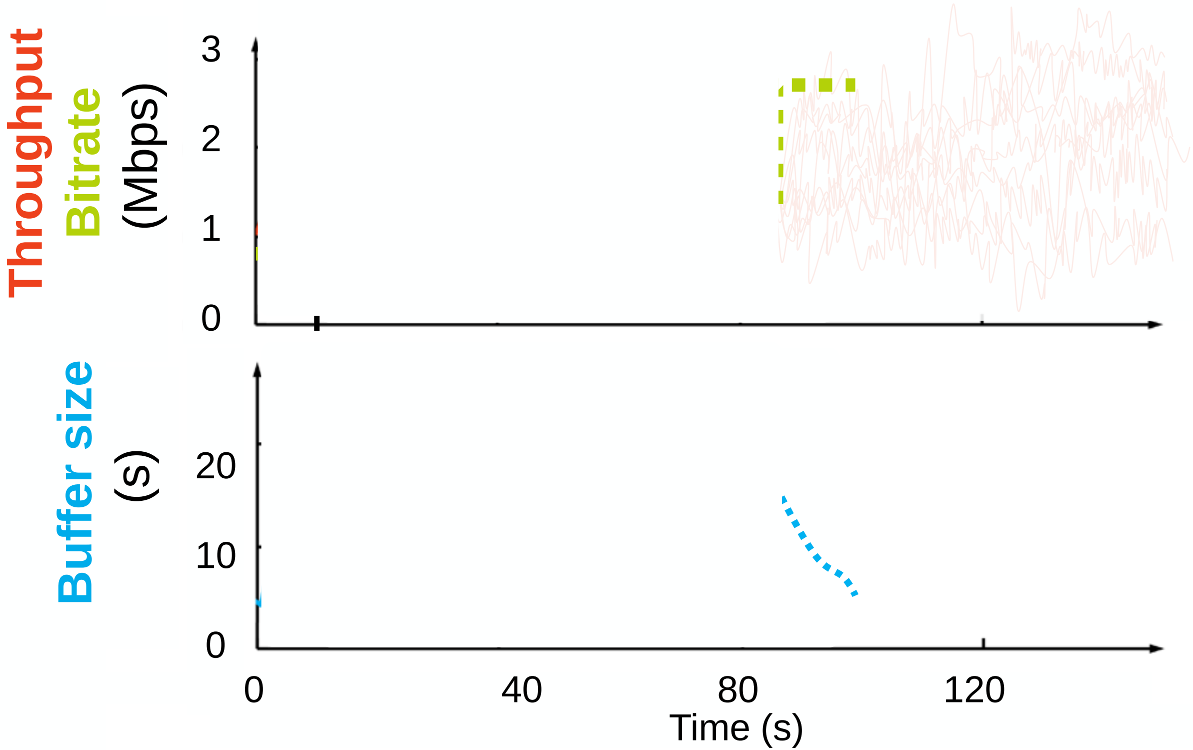
Enable a control plane

- Assist the clients and network elements thru metrics and analytics

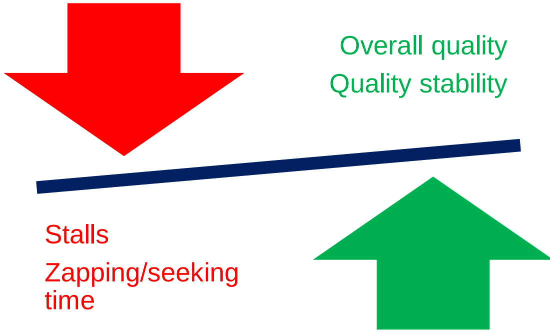
Proposed Solution

Our Solution (Fully Distributed Collaborative HAS)

Objectives

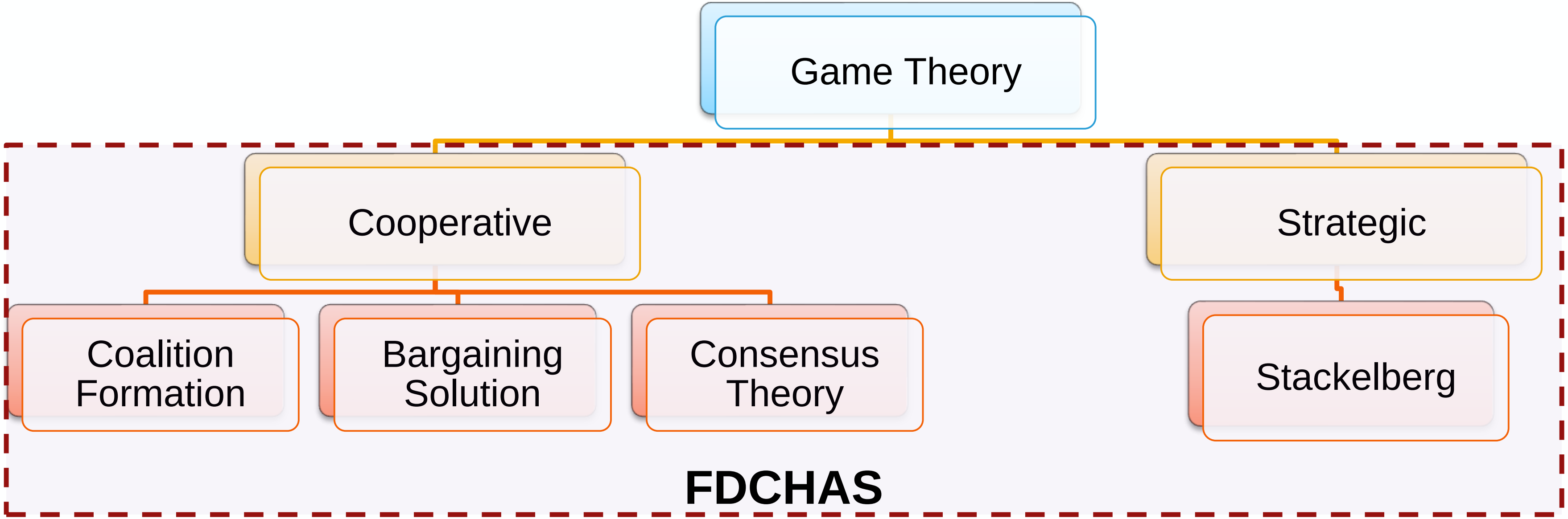


- Network throughput is variable & uncertain
- Conflicting QoE goals
  - Quality to be selected
  - Stalls
  - Quality stability
- FDCHAS Objectives:

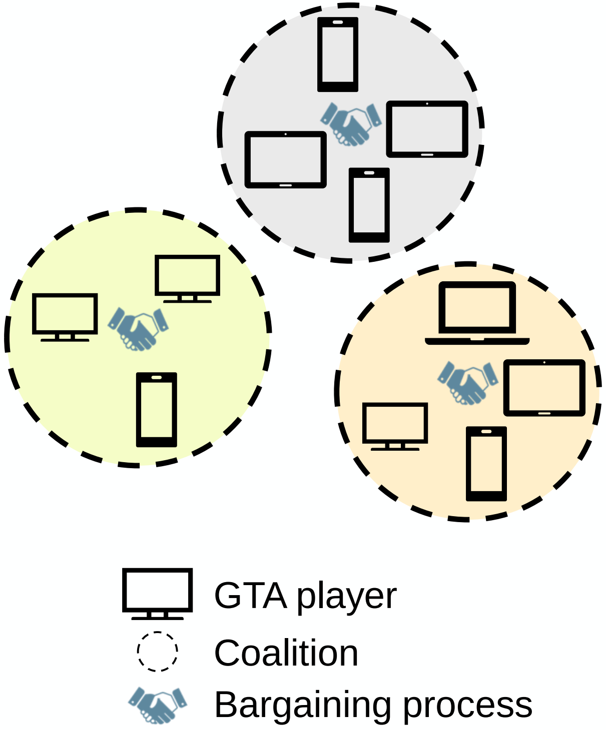


Let’s Adopt Ideas from Game/Consensus Theory

- A set of a mathematical formal analytical tools
- Model/analyze the complex interactions among several rational decision makers (players in this case) who can have common or conflicting objectives



FDCHAS Solution

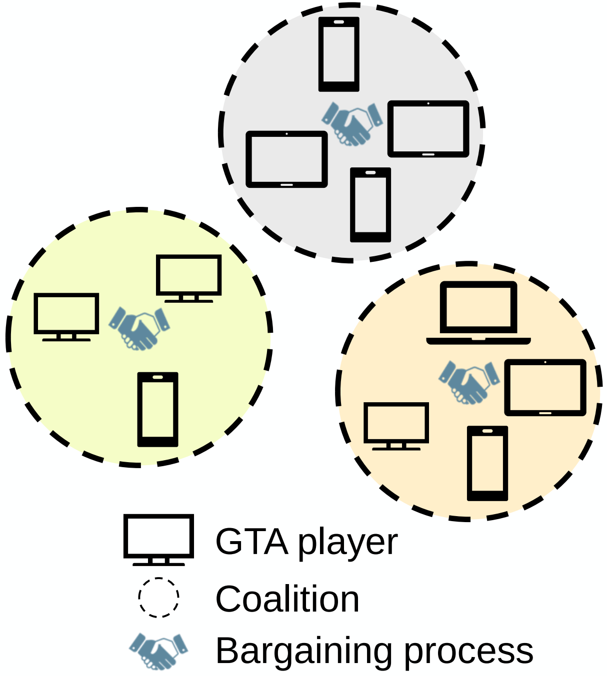


FDCHAS (2 Stages game):

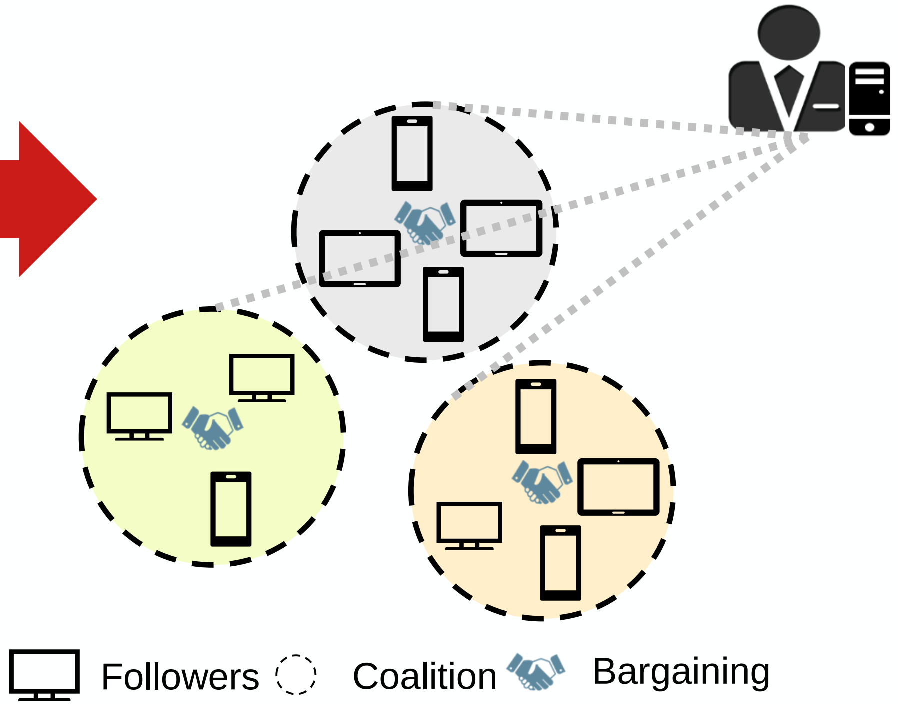
- Designed based on a cooperative game and non-cooperative games:
  - Cooperative game in the form of static formation-based coalitions.
  - Non-cooperative as Stackelberg game for resource allocation
- Formulates the ABR decision problem as a bargaining process and consensus mechanism.
- 1st stage: Outputs the optimal bitrate decision by reaching the Pareto optimal (PO) Nash bargaining solution (NBS).
- 2nd stage: Outputs the optimal network resource decision by reaching the Stackelberg equilibrium.

FDCHAS Solution

Stage 1: Coalitions and Bargaining games



Stage 2: Network Resource Allocation (Stackelberg-game)



FDCHAS Design

Stage 1a- Coalition Formation

- Structuring a set of players in cooperative groups
- Enables a strong level of cooperation among the players
- Model and analyze fairness, stability and efficiency

Stage 1b- Bargaining Solution

- Players that are interested in reaching an agreement among themselves over a common objective (how to reach this agreement and on the terms of the agreement)
- Concept solution: Nash Bargaining Solution (NBS)

Stage 1b- Consensus Theory

- Exploited together with NBS and coalition-formation game
- Relied on a strong coordination among the players

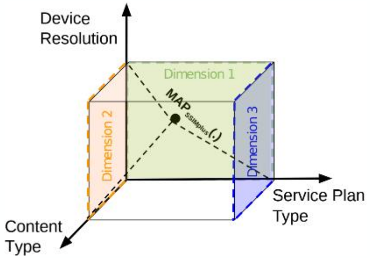
Stage 2- Network Resource Allocation

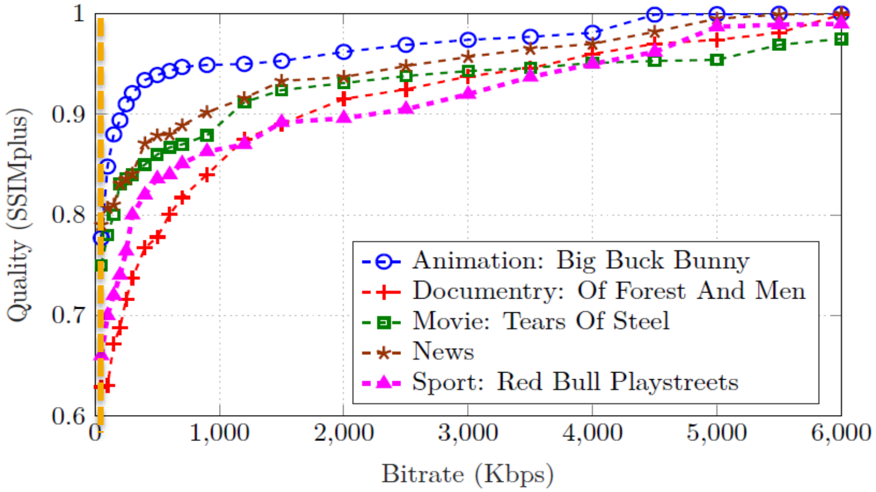
- Formulates the dynamic per-coalition network resource slicing, allocation and QoS provisioning as a Stackelberg strategic game
- The aggregation router or server (leader) holds a strong position and imposes a set of resource allocation rules upon all coalition members (followers)

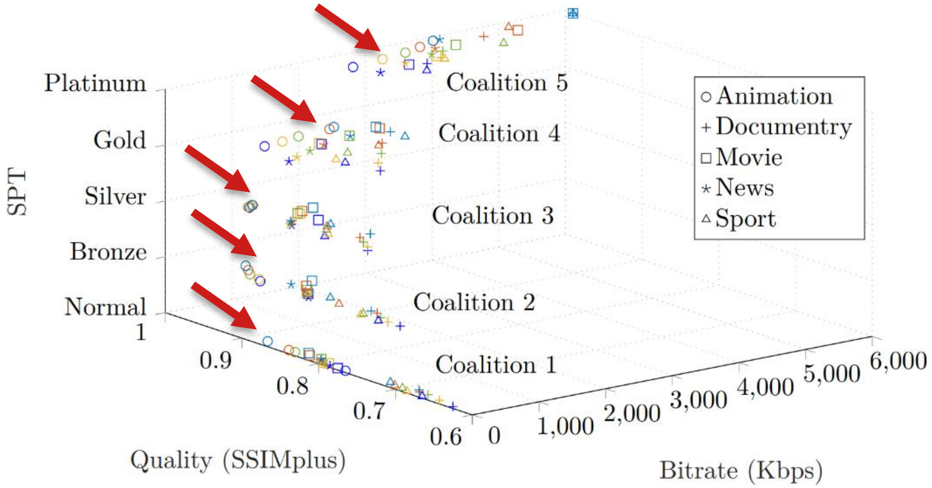
FDCHAS Design

Stage 1a: Coalition Formation (full reference quality measurements using SQM)

Feature	Description	Values
DR	Player device resolution	240p, 360p, 480p, 720p, 1080p
CT	Type of video requested	Animation, Documentary, Movie, News, Sport
SPT	ISP QoS business plan	Normal, Bronze, Silver, Golden, Platinum







GTA Design

Stage 1b: Bargaining and Consensus (ABR Decision Formulation)

- GT Strategy Calculator
  - Compute the GT strategy (i.e., the action-utility U(A) relationship set) which is defined over the strategy space S as follows:

Agreement point

$$\begin{cases} \mathcal{S}_{CL} = \{S_1, \ldots, S_{Cl_\mu}, \ldots, S_B | S_{Cl_\mu} \in \mathcal{U}_{CL}(\mathcal{A}_{CL})\}, \\ S_{Cl_\mu} = \{s_{p_1}, \ldots, s_{p_k}, \ldots, s_{p_{N_{Cl_\mu}}} | s_{p_k} \in U^v_{Cl_\mu}(A^v_{Cl_\mu})\}, \\ s_{p_k} = \{\boxed{u^v_{i,p_k}(a^v_{i,p_k})} | \forall i = \{1, 2, \ldots, K\}\}. \end{cases}$$

MAX

- GT (Dis)agreement Calculator
  - Compute the GT disagreement decision
    - i.e., bargaining outcome disagreements (Y)

Disagreement point

$$\begin{cases} \mathcal{Y}_{CL} = \{Y_1, \ldots, Y_{Cl_\mu}, \ldots, Y_B | Y_{Cl_\mu} \in \mathcal{U}^-{}_{CL}(\mathcal{A}^-{}_{CL})\}, \\ Y_{Cl_\mu} = \{y_{p_1}, \ldots, y_{p_k}, \ldots, y_{p_{N_{Cl_\mu}}} | y_{p_k} \in U^{-,v}_{Cl_\mu}(A^{-,v}_{Cl_\mu})\}, \\ y_{p_k} = \{\boxed{u^{-,v}_{i,p_k}(a^{-,v}_{i,p_k})} | \forall i = \{1, 2, \ldots, K\}\}, \end{cases}$$

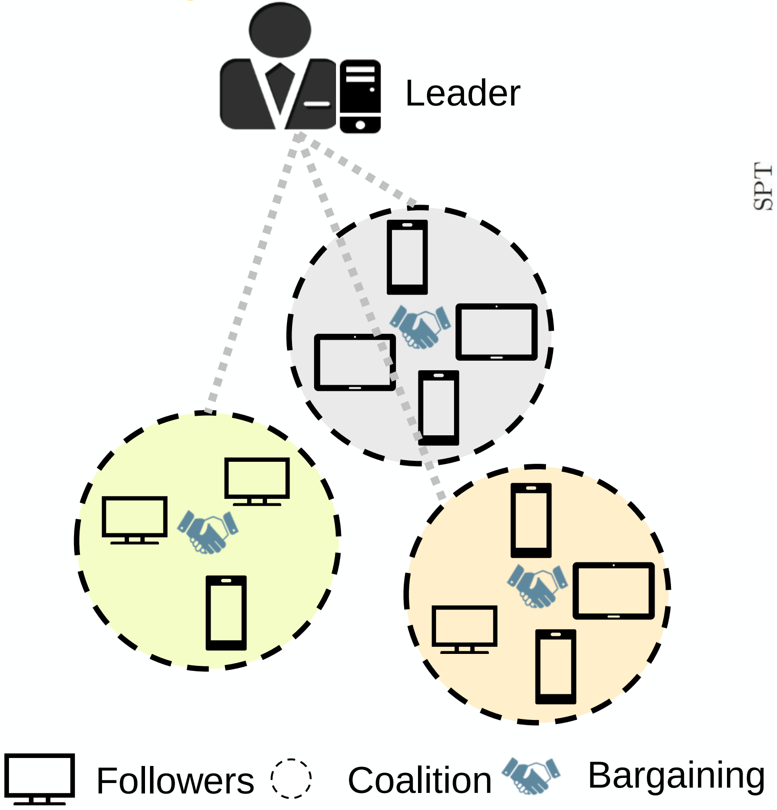
CL = $\{Cl_1,\ldots,Cl_B\}$ is the set of *B* coalitions
P = $\{p_1,\ldots,p_N\}$ is the set of *N* players, and N_{Clu} is number of player in cluster Cl_u
U is the set of possible utilities when actions from **A** are taken

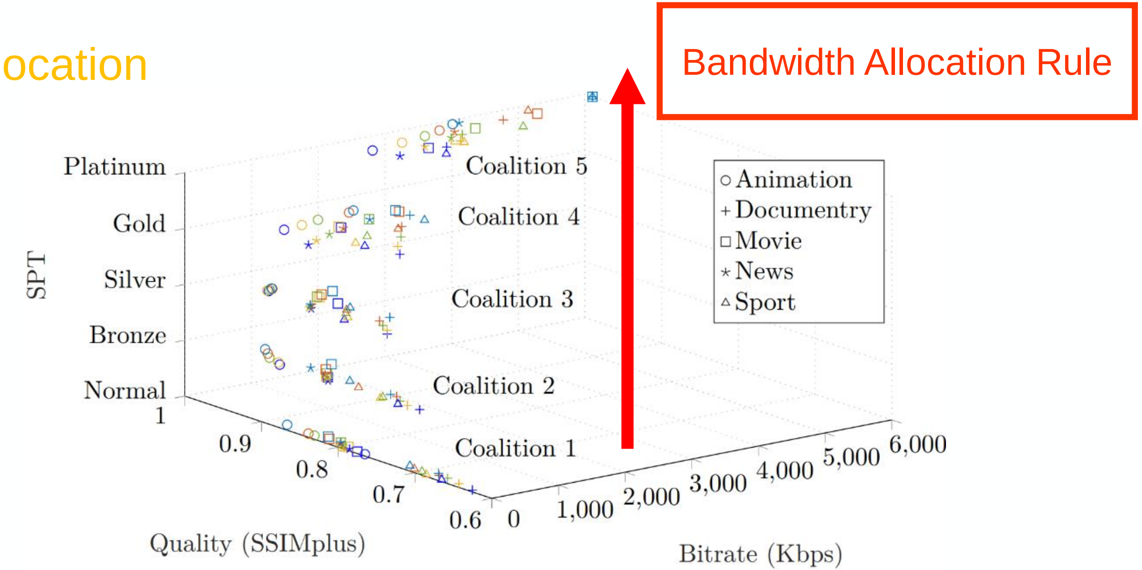
- u is the player utility when action a is taken
- *i* is the current step, *K* is total number of segments of video u

U(**A**) is the set of possible actions with their achievable utilities
U⁻(**A**⁻) is the set of pessimal actions with their resulting utilities
S and **Y** are defined by the function F which is defined over the space: F: (**A** ⟶ **U**) ∪ {**Y**}
PO (NBS): max. (S – Y)

FDCHAS Design

Stage 2: Network Resource Allocation





Inputs:

- The joint decision taken by each follower
- The coalition rule (SSIMplus mapping model)
- The preference relation (▷):
  - $CI_5 \triangleright CI_4 \triangleright CI_3 \triangleright CI_2 \triangleright CI_1$

Output:

- The best amount of bandwidth for CL that reaches the Stackelberg equilibrium.

FDCHAS Design

Objective Function

- FDCHAS rule for ABR selection is formulated as a network utility maximization
  - Strictly increasing concave
  - Solver: Dual decomposition + dynamic programming + fastMPC

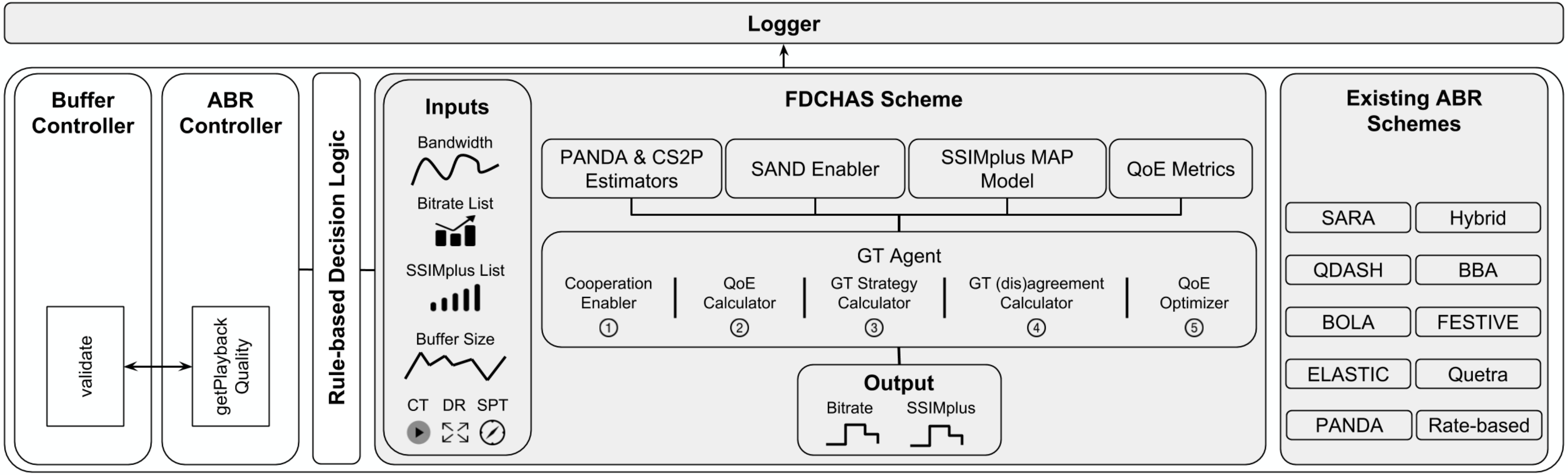
$$
\begin{cases}
\left.\begin{array}{l}
\text{find } a^{\star,v}_{i,p_k} \Leftrightarrow qt^{\star,v}_{i,p_k}(l^{\star,v}_{i,p_k}) \\
\arg \max\limits_{u^{v}_{p_k} \in U^{v}_{Cl_\mu}, a^{v}_{p_k} \in A^{v}_{Cl_\mu}} u_{i,p_k}
\end{array}\right\} \Leftrightarrow x^{\star}_{i,p_k} \in X^{\star}_{i,Cl_\mu} \\
\text{s.t. } buff^{min}_{p_k} \leq buff_{i,p_k} \leq buff^{max}_{p_k} & \text{C.1} \\
\quad \text{MAP}_{SSIMplus}(a^{\star,v}_{i,p_k}, \{CT^{u}_{i,p_k}, DR_{i,p_k}, SPT_{i,p_k}\}) & \text{C.2} \\
\quad x^{\star}_{i,p_k} = u^{\star,v}_{i,p_k}(a^{\star,v}_{i,p_k}) = \mathbb{F}(s_{i,p_k}, y_{i,p_k}) & \text{C.3} \\
\quad \phi^{x}(X^{\star}_{i,Cl_\mu}) = \mathcal{F}^{x}(X^{\star}_{i,Cl_\mu}) \approx 0 & \text{C.4} \\
\quad l^{\star,v}_{i,p_k} \leq bw^{e}_{i,p_k} \Leftrightarrow \tau^{max}(l^{\star,v}_{i,p_k}) \leq \tau, \forall l^{\star,v}_{i,p_k} \in L & \text{C.5} \\
\quad c^{e}_{i,p_k} \leq 1, \text{ where } c^{e}_{i,p_k} = (l^{\star,v}_{i,p_k} + bw^{e,bt}_{i,p_k})/bw^{e}_{i,p_k} & \text{C.6}
\end{cases}
$$

C.1 Maintain the current buffer occupancy between the min and max buffer thresholds
C.2 Satisfy the SSIMplus MAP model
C.3 Lead to a unique (PO) NBS
C.4 Local objectives and their potential functions should achieve approximate equality.
C.5 Chunk time constraint where max time needed to download the corresponding chunk ≦ chunk duration
C.6 Avoid network congestion situation

- Find the bargaining solution by solving the Problem(S,Y)
  - Bargaining Process and consensus decision problem is defined as a pair Problem(S,Y)
  - The bargaining solution is defined by a function F over the space F : (S, Y) → R^n
  - F specifies a unique Pareto optimal (PO) bargaining outcome (O* = F(S,Y)) for every Problem(S,Y) using Nash Bargaining Solution (NBS)
  - O* is the set that contains only the optimal bargaining outcomes

dash.js Integration

Source Code and Demo is Available at http://streaming.university/GTA/



Gray boxes indicate the components that have been developed or modified

Evaluation and Experimental Results

Setup for Performance Evaluation

VoD experimental setup:

	Parameters	Evaluated Values
HAS Player (FDCHAS.js)	$buff^{min}$	8 seconds
	$buff^{max}$	36 seconds
	u	Normalized QoE (1 to 5) [5]
	DR	240p, 360p, 480p, 720p, 1080p
	SPT	Normal, bronze, silver, gold, platinum
	$\omega_{1,2,3,4}$	0.25 for each
Manifest Files	CT	Five types of videos
	T	600 seconds
	K (Steps)	150 steps
	τ	4 seconds
	L (Actions)	20 bitrate levels (H.264) varying from 45 to 4000 Kbps
	QT	SSIMplus-based
Coalitions	Cl	Five coalitions $\{Cl_1, \ldots, Cl_5\}$
	φ fastMPC lookahead	Three steps
	α Bargaining power	$1/N_{CL_\mu}, \forall Cl_\mu \in CL$
Network Configuration	# of HAS players (N)	100
	Total bandwidth	170 Mbps
	Background traffic	rand(10..70) Mbps
PANDA BW Estimator	κ, ω, $\mathcal{B}$	0.14, 0.3, 0.2, respectively
N-QoE	$u \in [0.8, 1] \longrightarrow [4, 5], [0.6, 0.8] \longrightarrow [3, 4], [0.4, 0.6] \longrightarrow [2, 3], [0.2, 0.4] \longrightarrow [1, 2], [0, 0.2] \longrightarrow [0, 1]$	
Other HAS Schemes	As suggested in their respective papers	

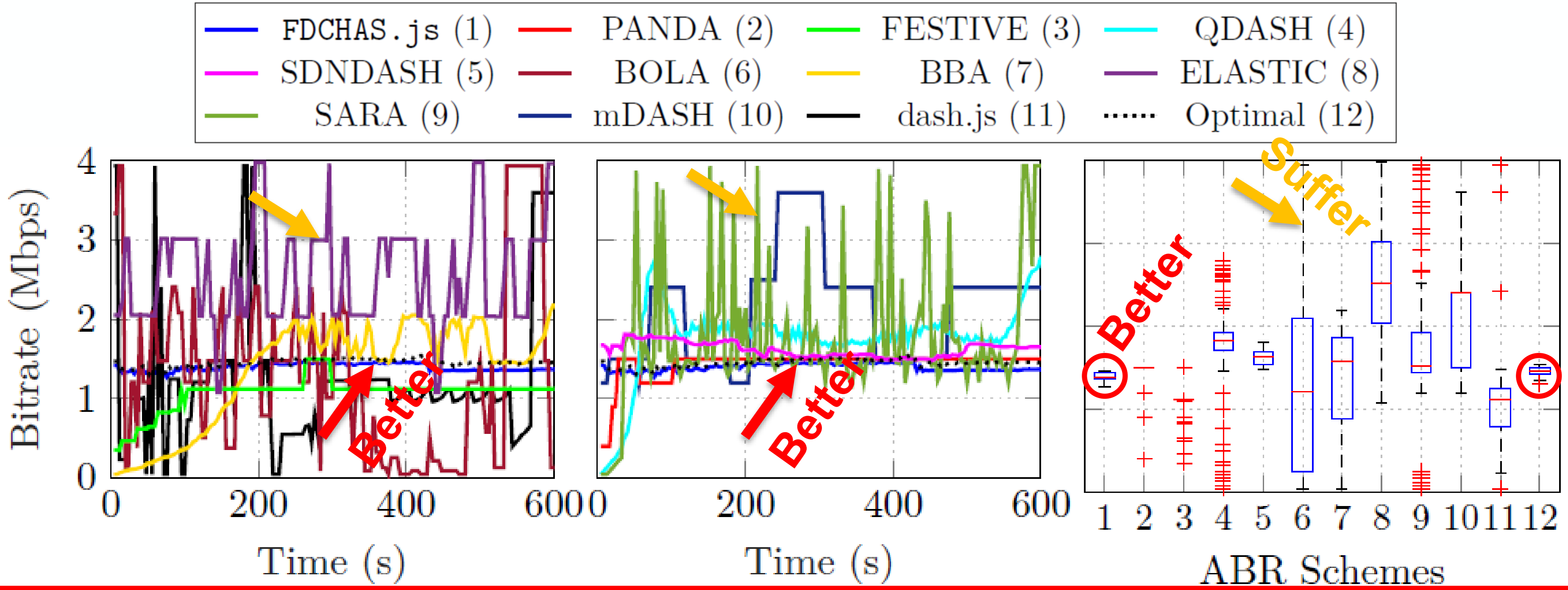
Setup for Performance Evaluation

QoE Calculator

$$\underbrace{\delta_1 \sum_{k=1}^{K} q_p^k(l_p^k)}_{\text{Avg. Quality}} - \underbrace{\delta_2 \sum_{k=1}^{K-1} \left| q_p^{k+1}(l_p^{k+1}) - q_p^k(l_p^k) \right|}_{\text{Avg. Quality Switch}} - \underbrace{\delta_3 SE_p^k}_{\text{Stalls}} - \underbrace{\delta_4 T_p^{sd}}_{\text{Startup Delay}}$$

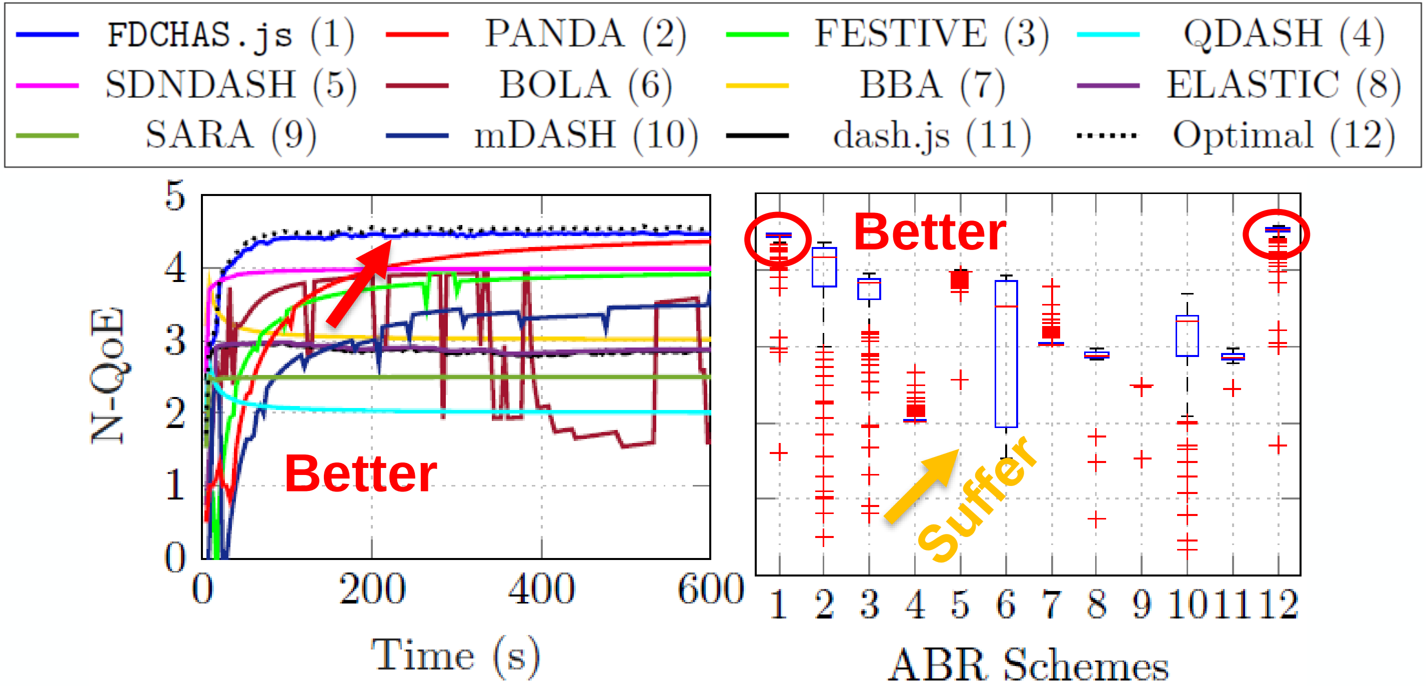
NUS National University of Singapore | School of Computing

Results: Video Stability (Avg. 100 players)



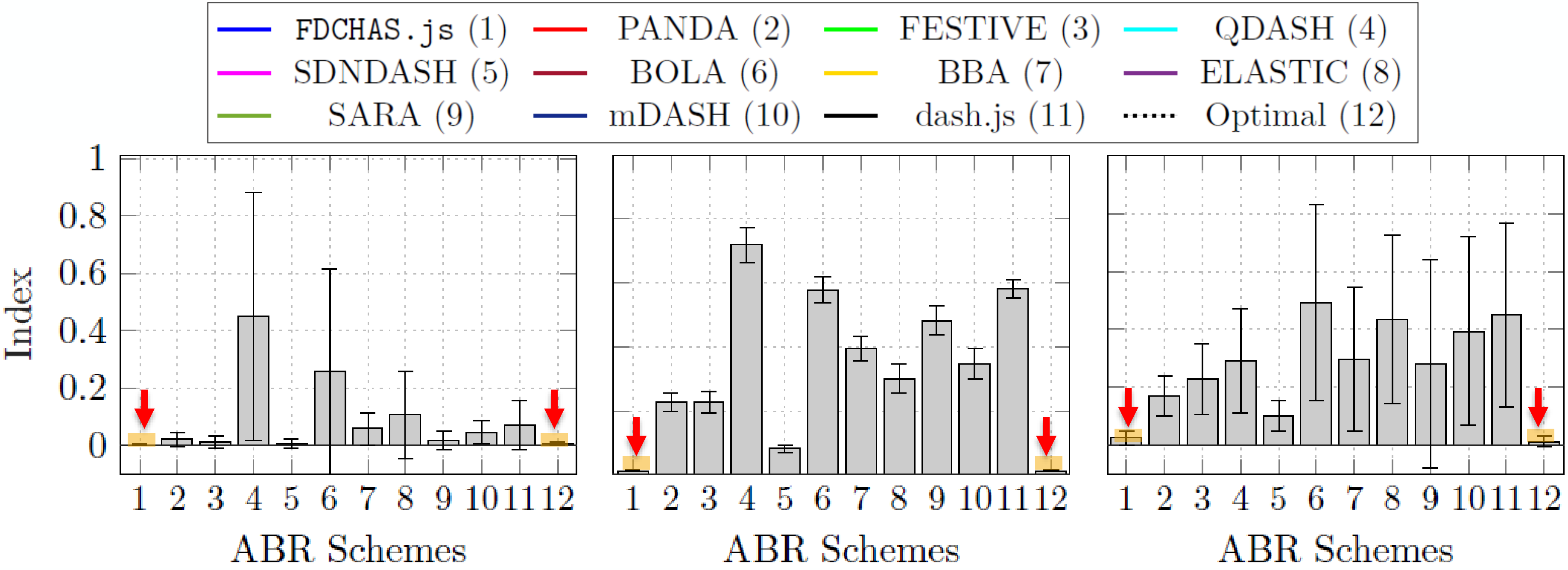
FDCHAS improves the video stability by 2-97% and is within 1-10% of the offline optimal

Results: QoE (Avg. 100 players)



FDCHAS improves the QoE by 8-41% (5-61 % fair) and is within 3-8% (2-9% fair) of the offline optimal

Results: Video Instability, Unfairness, Underutilization Indexes



FDCHAS improves the utilization by 10-44% and is within 1-5% of the offline optimal

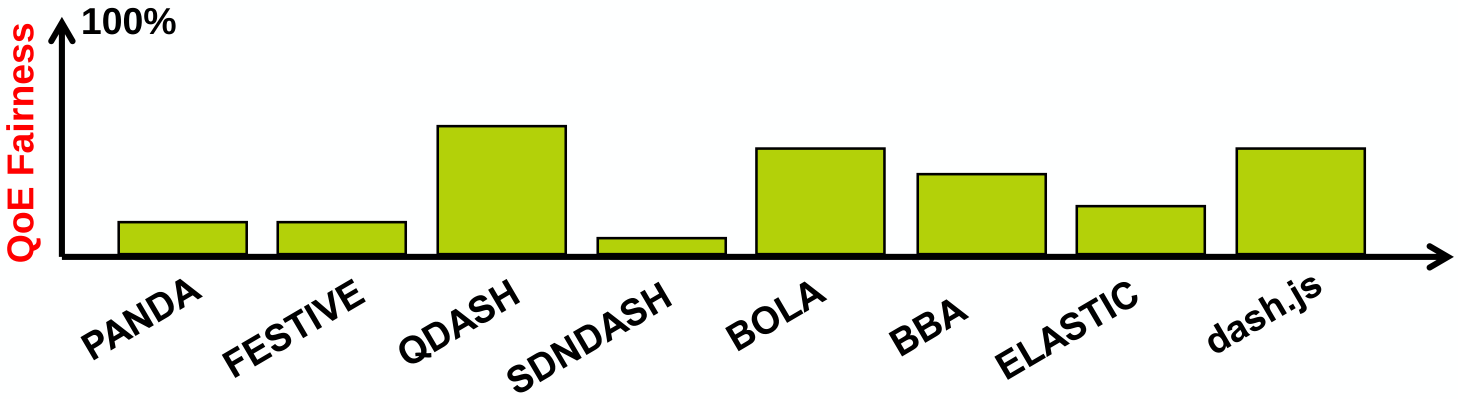
Reading: Zhi Li *et al* "Probe and Adapt: Rate Adaptation for HTTP Video Streaming At Scale?," IEEE JSAC 2014

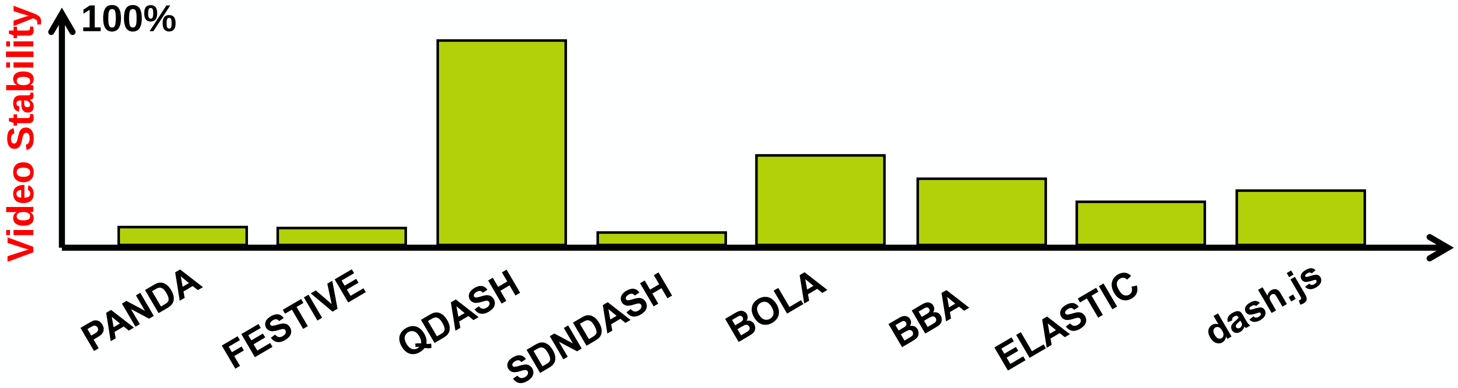
Results: Average total Results over 100 players

	AVG Bitrate Level (Kbps)	AVG Quality (SSIMplus)	AVG # of Oscillations & Stalls	AVG Utility & Conv-T	AVG Quality Variance	AVG Instability	AVG Unfairness	AVG Underutilization
FDCHAS.js	**1200 to 1500**	**0.885 to 0.918**	**3.6 & 1.6 (0.85 s)**	**4.33 & 14.38 s**	**0.033**	**0.009**	**0.013**	**0.033**
PANDA	400 to 1500	0.84 to 0.918	8 & 3 (7.66 s)	3.69 & 29 s	0.04	0.023	0.18	0.17
FESTIVE	350 to 1200	0.83 to 0.885	13 & 3 (11.33 s)	3.58 & 34.9 s	0.057	0.021	0.16	0.22
QDASH	45 to 2790	0.69 to 0.937	147 & 11 (11.6 s)	2.05 & 120 s	0.247	0.98	0.62	0.29
SDNDASH	1500 to 2100	0.918 to 0.94	7 & 3 (4.3 s)	3.96 & 23.2 s	0.022	0.014	0.056	0.135
BOLA	45 to 4000	0.69 to 0.97	66 & 9 (9.4 s)	3.19 & 84 s	0.28	0.44	0.51	0.54
BBA	45 to 2100	0.69 to 0.94	36 & 4 (8 s)	3.06 & 101 s	0.25	0.24	0.33	0.29
ELASTIC	1000 to 4000	0.889 to 0.97	23 & 5 (2.8 s)	3.41 & 38 s	0.081	0.12	0.22	0.36
dash.js	45 to 4000	0.69 to 0.97	23 & 8 (9.8 s)	2.88 & 125 s	0.28	0.153	0.53	0.47
Offline bound	1300 to 1500	0.893 to 0.91	2.2 & 1 (0.174 s)	4.25 & 10.14 s	0.017	0.008	0.011	0.018

FDCHAS Outperforms the Existing ABR Schemes

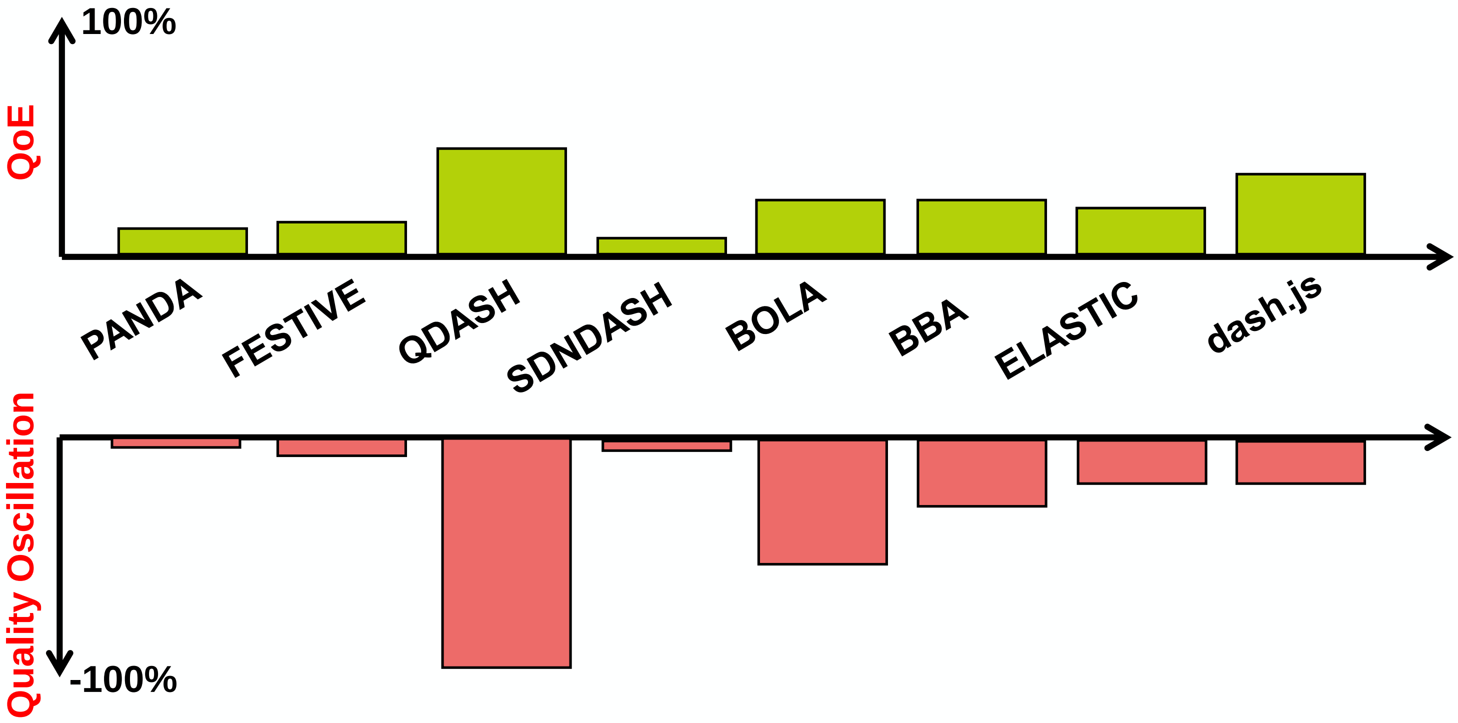
Avg. 100 Players





FDCHAS Outperforms the Existing ABR Schemes

Avg. 100 Players



FDCHAS Outperforms the Existing ABR Schemes

Avg. 100 Players

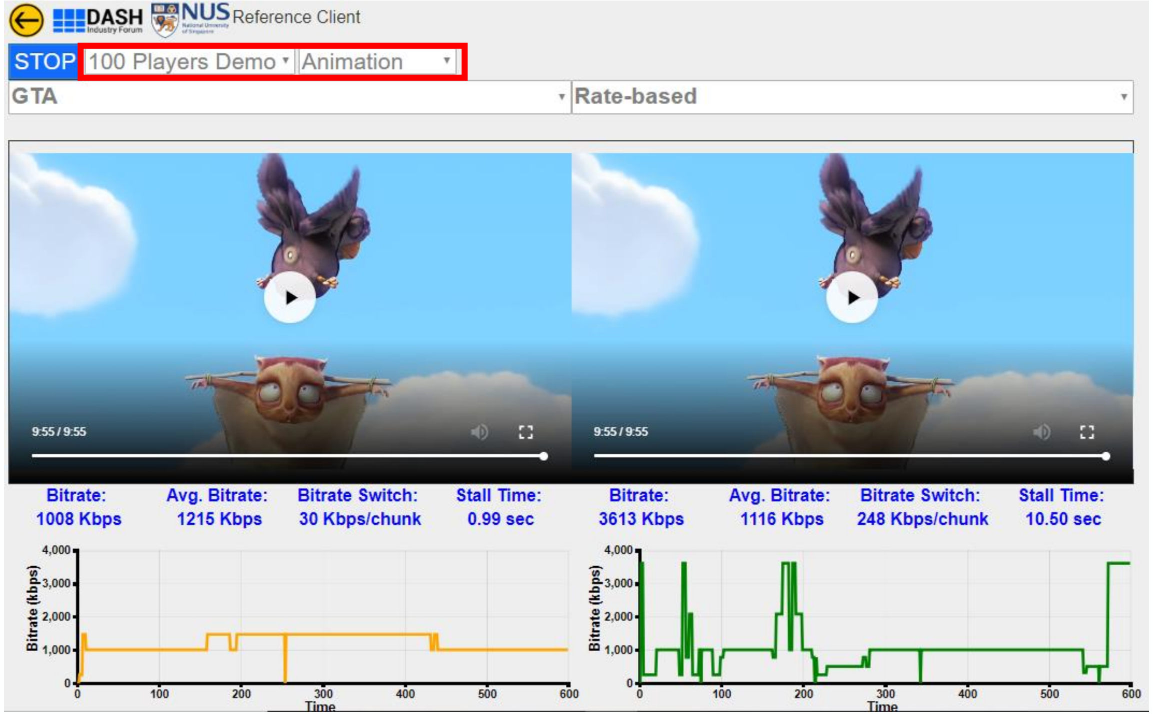
- Consecutive numbers represent the results
  - FDCHAS ensures a lower startup delay compared to the existing ABR schemes

	PANDA	FESTIVE	QDASH	SDNDASH	BOLA	BBA	ELASTIC	dash.js
QoE Fairness	17%	15%	61%	4.5%	50%	32%	21%	52%
Video Stability	1.5%	1.2%	97.1%	1%	43%	23%	11.5%	15%
QoE	12.8%	15%	41%	7.5%	23%	25.5%	18.5%	29%
Quality Oscillations	3%	6%	95%	2%	41.5%	21.5%	13%	13%

Improvement Minimization

FDCHAS Demo

http://streaming.university/GTA/offline/index.html



Conclusions and Future Work

Conclusions and Future Directions

- Leveraging a two-stage game, we designed FDCHAS, a fully distributed collaborative bitrate selection scheme for the HAS-based VoD services.
  - First stage: Group set of players in coalitions then formulate the ABR as consensus and bargaining problem.
  - Second stage: Formulate the network resource allocation as Stackelberg game.
- FDCHAS largely eliminates HAS scalability issues: Instability, Unfairness, and Under/over utilization.
- Results showed that the FDCHAS achieves high efficiency across all players, can be practically implemented, adheres to the spirit of distributed and client-driven HAS.
- FDCHAS significantly outperforming other state-of-the-art adaptation schemes.
- We plan to extend FDCHAS to support :
  - Networks with multiple shared bottleneck links and live streaming services.
  - Dynamic coalition formation, including the analysis of the deviating players, and its theoretical guarantees

Want to Play DASH? A Game Theoretic Approach for Adaptive Streaming over HTTP

Abdelhak Bentaleb
National University of Singapore
bentaleb@comp.nus.edu.sg

Ali C. Begen
Ozyegin University
ali.begen@ozyegin.edu.tr

Saad Harous
United Arab Emirates University
harous@uaeu.ac.ae

Roger Zimmermann
National University of Singapore
rogerz@comp.nus.edu.sg

GAME THEORY BASED BITRATE ADAPTATION FOR DASH.JS REFERENCE PLAYER

Abdelhak Bentaleb[⋆], Ali C. Begen[‡] and Roger Zimmermann[⋆]
[⋆]National University of Singapore, [‡]Ozyegin University
{bentaleb,rogerz}@comp.nus.edu.sg, ali.begen@ozyegin.edu.tr

Session: System-1 (Video Analysis & Streaming) MM'18, October 22-26, 2018, Seoul, Republic of Korea

A Distributed Approach for Bitrate Selection in HTTP Adaptive Streaming

Abdelhak Bentaleb[⋆], Ali C. Begen[*], Saad Harous[+], and Roger Zimmermann[⋆]
[⋆]National University of Singapore, [*]Ozyegin University, [+]United Arab Emirates University
{bentaleb,rogerz}@comp.nus.edu.sg, ali.begen@ozyegin.edu.tr, harous@uaeu.ac.ae

Thank You

https://www.comp.nus.edu.sg/~bentaleb/

http://streaming.university/