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

ABSTRACT

Past research has shown that concurrent HTTP adaptive streaming (HAS) players behave selfishly and the resulting competition for shared resources leads to underutilization or oversubscription of the network, presentation quality instability and unfairness among the players, all of which adversely impact the viewer experience. While coordination among the players, as opposed to all being selfish, has its merits and may alleviate some of these issues. A fully distributed architecture is still desirable in many deployments and better reflects the design spirit of HAS. In this study, we focus on and propose a distributed bitrate adaptation scheme for HAS that borrows ideas from consensus and game theory frameworks. Experimental results show that the proposed distributed approach provides significant improvements in terms of viewer experience, presentation quality stability, fairness and network utilization, without using any explicit communication between the players.

ACM Reference Format:
Abdelhak Bentaleb, Ali C. Begen, Saad Harous, and Roger Zimmermann. 2018. A Distributed Approach for Bitrate Selection in HTTP Adaptive Streaming. In *2018 ACM Multimedia Conference (MM '18), October 22-26, 2018, Seoul, Republic of Korea.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3240508.3240589

1 INTRODUCTION

HTTP adaptive streaming (HAS) players are designed to individually select the appropriate, usually the highest feasible, bitrate level at each decision epoch. However, this decentralized and strategically selfish behavior results in serious problems when multiple players compete for shared bandwidth [1, 5]. Past measurements [2] have shown that the standard on-off transmission pattern of an HAS adaptation process leads to bandwidth overestimation, which in turn results in suboptimal bitrate selections. Thus, players suffer from scalability issues [5] such as unstable presentation quality, quality unfairness and network underutilization or oversubscription, with an overall detrimental effect on viewers' satisfaction (QoE).

Consequently, efficient cooperation and coordination between HAS players, and effective network resource allocation and management mechanisms are needed to mediate between the growing demand from HAS players, their competition, and the limited network capacity. Our work is inspired by a recent *game theory* (GT) [11] framework with its *consensus* [20] variant and its application to emerging signal processing applications [3]. GT represents an attractive mathematical tool that has been used to address fundamental issues in network communications amid distributed networks, and in particular multimedia applications. GT models and analyzes the interactions between multiple purely decentralized and isolated decision makers (*i.e.*, HAS players in our case) in strategic or cooperative ways. In our context, we postulate that formulating a fully distributed decision problem based on GT is a good match with the high interdependence between the actions taken by a set of players and the shared resources.

We propose a two-stage, game theoretic bitrate adaptation scheme termed FDCHAS (Fully Distributed Collaborative HAS) for video-on-demand (VoD) services. The goal is to eliminate HAS scalability issues while maximizing players' utilities in a fully distributed manner without any explicit signaling between different HAS entities (*e.g.*, players, network devices, gateways, and servers). FDCHAS uses a GT framework with consensus mechanism [20] to formulate the strong coordination among the players belonging to the same coalition and to analyze the distributed and collaboration behaviors. These benefits incentivize the HAS players to use FDCHAS and join the game. Our solution integrates a cooperative game that is *coalition formation-based* at the first stage and a strategic game that is *Stackelberg-based* at the second stage. Both stages operate concurrently during a streaming session. In our system formulation, the main insights of using these particular game models are: they allow a robust cooperation among the players, and they reflect the design spirit of HAS given the competing individual and group interests inherent in having multiple players stream video over a shared network.

(1) First Stage (@players side): We use a static coalition formation-based game to construct a non-overlapping coalition virtual network topology. This formation incorporates the Structural Similarity plus (SSIMplus) perceptual quality [9, 22] that maps three distinctive features, including device resolution (DR), content type (CT) and subscription plan type (SPT), into one common embedding space. The grouping of players into a set of coalitions helps FDCHAS benefit from GT cooperation, and thus, our model can support large-scale network deployments. Thereafter, we formulate the per-coalition bitrate and quality decisions as a bargaining process and consensus problem, where the set of players in the same coalition should agree on similar decisions and reach a consensus considering other coalition members' decisions and varying network conditions.
(2) Second Stage (@network side): We formulate the per-coalition dynamic bandwidth slicing and allocation as a Stackelberg game, where an aggregation router (or the HAS server) represents the leader that incorporates the joint decisions of each coalition

MM '18, October 22–26, 2018, Seoul, Republic of Korea

ACM ISBN 978-1-4503-5665-7/18/10…$15.00
https://doi.org/10.1145/3240508.3240589

member for QoS provisioning while coalition members (HAS players) represent the followers.

We extensively evaluated FDCHAS on a broad set of trace-driven and real-world experiments. Results show that the proposed solution does well compared to the offline calculated optimal bound, and it significantly improves the viewer QoE, presentation quality stability, fairness and network resource utilization compared to existing well-known schemes [15]. The rest of the paper describes existing bitrate adaptation schemes in Section 2, presents FDCHAS in Section 3, highlights the design steps in Section 4, evaluates the performance in Section 5 and concludes in Section 6.

2 RELATED WORK

To date, several HAS bitrate adaptation schemes have been proposed [15]. In this section, we describe the most recent schemes that have focused on addressing HAS scalability issues. Bentaleb *et al.* [5] proposed an architecture for SDN-enabled HAS systems, where the goal was to improve every viewer's QoE through the use of a Software Defined Networking (SDN) [14] central coordinator. Similarly, Mok *et al.* [19] developed a network-assisted bitrate adaptation logic where a proxy between the players and a server was used to eliminate quality oscillations based on a gradual variation between the available representations using integrated intermediate levels. ELASTIC [8] is an adaptation scheme designed based on linearization feedback. It uses control theory to avoid HAS' on-off pattern, and thus tries to maximize bandwidth fairness among the players based on network loop feedback. Another interesting work is FESTIVE [13], which is comprised of a bandwidth estimator, bitrate selector, and buffer-based randomized scheduler. Its goal is to achieve a high level of fairness, efficiency and stability when multiple players compete in a shared network. Along the same lines, PANDA [18] was designed to avoid bandwidth overestimation caused by on-off patterns by proposing an accurate bandwidth estimation algorithm. Sobhani *et al.* [24] designed a fuzzy logic based bitrate adaptation solution that combines both the predicted throughput and buffer occupancy. In [12] and [25], buffer-based bitrate adaptation schemes were introduced. The former, named BBA, tries to improve the video quality while avoiding video stalls. The latter, termed BOLA, is an online control scheme that formulates the adaptation as a utility maximization problem. Yuan *et al.* [29] proposed a bitrate adaptation scheme that is based on non-cooperative GT. It aims to address QoE fairness when multiple users share a network. Although the proposed solution showed good performance, it generates extra overhead due to the Nash Equilibrium (re)compute. In general, the existing schemes do not provide any theoretical guarantees for the bitrate decision process. In contrast, we derive a GT-based bitrate selection rule with theoretical guarantees and proofs.

3 FDCHAS ARCHITECTURE

3.1 Overview

We propose FDCHAS, a fully distributed (*i.e.*, no explicit message exchanges are needed between the entities) bitrate adaptation scheme for HAS players competing for resources. More background information on the GT-based mathematical concepts that are used in the proposed approach can be found in [3, 11]. FDCHAS employs GT and consensus mathematical concepts to respect the decentralized nature of HAS and to provide a collaborative mechanism between the players. FDCHAS is applicable to any type of bottleneck link (*e.g.*, last-mile links in mobile and public WiFi networks). It consists of four main entities:

An **HAS Player** requests and downloads the manifest (the file that contains the segment addresses and perceptual quality information including an SSIMplus-based mapping (2)) and the segments of a given video content from the HAS server. The interactions between a player and a server are delivered via a local and an aggregation router. We use the DASH-IF reference player dash.js [7] as our HAS player. To support our architectural design, we extended dash.js by adding the following classes:

(1) The *FDCHAS Bitrate Adaptation Scheme* implements our bitrate adaptation logic. As shown in Table 1, our scheme requires local variables as input, and it computes the optimal bitrate level with its corresponding quality as the output (see Algorithm 1 in the supplementary document).
(2) The *File Logger* class records the player status including its buffer occupancy, QoE, bitrate and quality decisions, segment download times, stall events, bitrate and quality switches.

The **HAS Server** is an HTTP server that stores media segments and manifests. We complement the conventional manifests with perceptual quality measurements based on the SSIMplus index [9, 22], which were performed offline for each video (see Section 4.1). The SSIMplus-derived mappings based on our model (see Figure 1 in the supplementary document) are stored in an XML file. This information allows a player to dynamically identify its corresponding coalition, and players in the same coalition cooperate to achieve their objectives. The HAS server periodically reports the number of players in each coalition to the coalition members leveraging the HTTP header in a response.

A **Local Router** represents any smaller-grade router deployed in a specific area like a home network.

The **Aggregation Router** (*i.e.*, the edge router) represents the network entity that interconnects multiple local routers and aggregates their traffic. The FDCHAS-based network resource slicing and allocation algorithm (see Section 4.3) can be implemented in such an entity by the service provider. Our design also needs to obtain accurate available bandwidth estimations, and we use two algorithms, namely PANDA [18] and CS2P [26] for this purpose.

3.2 System Model and Problem Formulation

We consider an HAS delivery system with a set $P = \{p_1, \ldots, p_N\}$ of N HAS players, sharing a single bottleneck link with a fixed total capacity BW^{all} and unknown available network resources BW^e. There exist varying background traffic (BW^{bt}) and HAS (BW^{HAS}) bandwidth requirements. The bottleneck link has a limited available bandwidth with high variability, and it may not be able to satisfy the requirements of all players. Before starting any streaming session, every player $p_k \in P$ joins one of the coalitions $\forall Cl_\mu \in CL$ of B constructed coalitions using the Coalition Rule of (2), detailed in Section 4. Such a rule may be imposed by a service provider as a business or operational policy. Furthermore, a player $p_k \in Cl_\mu$ is physically connected to one of the local routers $r_j^{local} \in R^{local}$, with $j = \{1, \ldots, |R^{local}|\}$, which collectively connect to the same aggregation router $r^{agg} \in R^{agg}$. Player p_k sends requests via router

r_j^{local} and r^{agg} for a segmented video u from the set of content types CT, denoted as $CT^u_{p_k}$, that is stored on the HAS server I. Each video consists of K segments of a fixed duration τ (2–10 seconds) for a total of T seconds of media, and thus $\tau = \frac{T}{K}$. Every segment is encoded at various bitrate levels $l_\rho \in L$ and each level has an associated SSIMplus-based perceptual quality $qt_\rho \in QT$. At any decision epoch $i = \{1 \dots K\}$, each player selects a bitrate level with its corresponding quality. This is expressed as:

$$\begin{cases} L^{CT^u,DR}_{i,p_k} & = \{l^1_{i,p_k}, \dots, l^i_{i,p_k}, \dots, l^K_{i,p_k}\}, \\ QT^{CT^u,DR}_{i,p_k} & = \{qt^1_{i,p_k}, \dots, qt^i_{i,p_k}, \dots, qt^K_{i,p_k}\}, \end{cases} \tag{1}$$

where, $qt_{i,p_k}(l_{i,p_k})$ is the non-linear relationship between a specific bitrate level $l^\bullet_{p_k}$ and its corresponding quality $qt^\bullet_{p_k}$.

In general, our problem can be described as a four-tuple game $\mathcal{G}_{HAS-problem} = \langle \mathcal{E}, \mathcal{A}, \mathcal{S}, \mathcal{U} \rangle$, with environment $\mathcal{E}$, action space $\mathcal{A}$, strategy $\mathcal{S}$ and utility $\mathcal{U}$ as follows:

- $\mathcal{E} = \{E_1, \dots, E_K\}$ is the set of problem instances during a streaming session, with each element of environment $E_i = \{P_i, CL_i, I_i, R_i^{local}, R_i^{agg}\}$.
- $\mathcal{A} = \{A_1, \dots, A_K\}$ is the discrete and finite set of actions taken by the players from a list of bitrate levels and qualities of a specific video (see (1)), with $A_i = \{a_{p_1}, \dots, a_{p_k}, \dots, a_{p_N}\}$. We note that $\mathcal{A}_{CL} = \{A_{Cl_1}, \dots, A_{Cl_\mu}, \dots, A_{Cl_B}\}$ are the joint actions taken in each coalition, with $A_{Cl_\mu} = \{a_{p_1}, \dots, a_{p_{N_{Cl_\mu}}}\}$ equating the set of actions taken by the players that belong to the same coalition, with a total number of N_{Cl_μ}.
- $\mathcal{S} = \{S_1, \dots, S_K\}$ represents the set of heterogeneous strategies implemented by the players (*i.e.*, bitrate adaptation logic), with $S_i = \{s_{p_1}, \dots, s_{p_k}, \dots, s_{p_N}\}$. The players use their strategies to select a bitrate level with its corresponding quality. We note that $\mathcal{S}_{CL} = \{S_{Cl_1}, \dots, S_{Cl_\mu}, \dots, S_{Cl_B}\}$ are the joint strategies used in each coalition, with $S_{Cl_\mu} = \{s_{p_1}, \dots, s_{p_{N_{Cl_\mu}}}\}$ being the set of strategies that are used by the players of the same coalition.
- $\mathcal{U} = \{U_1, \dots, U_K\}$ denotes the set of utilities (*i.e.*, viewer QoE, see (15)) that are received when a set of actions $\mathcal{A}$ are taken by a set of player strategies $\mathcal{S}$ during a streaming session, where $U_i = \{u_{p_1}, \dots, u_{p_k}, \dots, u_{p_N}\}$. Note that $\mathcal{U}_{CL} = \{U_{Cl_1}, \dots, U_{Cl_\mu}, \dots, U_{Cl_B}\}$ are the joint[1] utilities in each coalition, with $U_{Cl_\mu} = \{u_{p_1}, \dots, u_{p_{N_{Cl_\mu}}}\}$ being the set of utilities received by the players of the same coalition.

3.3 Environment Variables

We define global and local environment variables with a mixture of network and player states as given in Table 1. In terms of coalitions, we apply an aggregation process (Agg(.)) to accumulate the local variables of the players belonging to a similar coalition Cl_μ.

Table 1: Environment variables.

Global Variables (of router R^{agg})	**Description**
$BW^{all}_{R^{agg}}$	Total bandwidth at the bottleneck link
$BW^{e}_{i,R^{agg}}$	Estimated available bandwidth
$BW^{HAS}_{i,R^{agg}}$	Throughput of the HAS traffic
$BW^{bt}_{i,R^{agg}}$	Throughput of the background traffic
$C_{i,R^{agg}}$	Network congestion level
Local Variables (of player p_k)	
$qt_{i,p_k}(l_{i,p_k})$	Bitrate with its quality decision in p_k
$buff_{i,p_k}$	Buffer occupancy size at p_k
$buff^{min}$	Min. threshold of buffer occupancy at p_k
$buff^{max}$	Max. threshold of buffer occupancy at p_k
$QoE_{i,p_k}(= u_{i,p_k})$	Viewer QoE at p_k
$CT^u_{p_k}, DR_{p_k}, SPT_{p_k}$	CT selected by p_k, DR of p_k, SPT for p_k
bw^e_{i,p_k}	Available bandwidth estimated by p_k
c^e_{i,p_k}	Congestion level estimated by p_k

4 FDCHAS DESIGN

We formally define the bitrate and quality decision problem when multiple HAS players compete for network resources as a parallel two-stage game, which is defined in the following subsections.

4.1 Stage 1.a – Coalition Formation

We are interested in a static coalition formation-based game with a nontransferable-utility (NTU) setting [3], which is dedicated to the grouping of a set of players into a few sets of cooperative coalitions. Coalition formation enables a strong level of cooperation among the players to strengthen their positions in a given game. To this end, we define a specific rule named *Coalition Rule* as follows:

$$\text{Coalition Rule} \equiv MAP_{SSIMplus}(DR_{p_k}, CT_{p_k}, SPT_{p_k}). \tag{2}$$

This rule employs the SSIMplus mapping function ($MAP_{SSIMplus}(.)$) used by each player to select an appropriate value from the common 3D embedding space (see Figure 1 in the supplementary document) that considers three features:

- The device resolution (DR) represents the display resolutions supported by the devices, *e.g.*, 240p, 360p, 480p, 720p and 1080p.
- The content type (CT) categorizes the videos stored on an HAS server into sports, documentary, news, animation and movie.
- The subscription plan type (SPT) represents the service plan levels that are offered by service providers, *e.g.*, normal, bronze, silver, gold and platinum.

Formally, our coalition formation-based game involves a set of N players $P = \{p_1, \dots, p_k \dots, p_N\}$ who are responsible to form a fixed number of B non-overlapping cooperative coalitions (*i.e.*, disjoint coalitions) based on the coalition rule (2), where $CL = \{Cl_1, \dots, Cl_\mu, \dots, Cl_B\}$ such that:

$$\forall \mu \neq \grave{\mu},\ Cl_\mu \cap Cl_{\grave{\mu}} = \varnothing, \quad \text{and} \quad \bigcup_{\mu=1}^{B} Cl_\mu = P. \tag{3}$$

Any coalition $CL \subseteq P$ represents a binding formation agreement (*i.e.*, a negotiation process outcome) between the players in CL to work as a single entity. This allows them to select joint decisions[2] that can maximize their individual utilities and reach a decision consensus. The binding formation agreement is defined by the coalition rule given in (2). Additionally, one fundamental concept in our modeling is the *coalition value* v [11] of the per-coalition joint actions (*i.e.*, bitrate and quality decisions vector) taken by the coalition members and resulting in specific values of utility. The coalition value of a specific $Cl_\mu \in CL$ is denoted as $v(Cl_\mu)$, which consists of actions taken by the coalition Cl_μ members. V is the set of all coalition values, *i.e.*, $V = \{v(Cl_1), v(Cl_2), \dots, v(Cl_\mu), \dots, v(Cl_B)\}$. As a consequence, our coalition formation-based game is defined as a triple $\langle P, V, CL \rangle$.

Definition 1. The coalition formation-based game $\langle P, V, CL \rangle$ with NTU is said to be non-superadditive if for any two disjoint coalitions $(Cl_\mu, Cl_{\grave{\mu}}) \in CL$, and $CL \subseteq P$, $v(Cl_\mu \cup Cl_{\grave{\mu}}) \geq v(Cl_\mu) + v(Cl_{\grave{\mu}})$, is not satisfied.

[1] We use the terms joint X, aggregation X and coalition X interchangeably.

[2] X and X^v are used interchangeably where $X^v \in v(.)$ is used after defining v, which is the coalition value or total payoff for the corresponding coalition.

We use the SSIMplus-based mapping model adopted from [4, 6]. This model (Figure 1 in the supplementary document) builds a fixed number of five static disjoint coalitions ($B = 5$) considering CT, DR and SPT features (2). The rationale behind this grouping model is to: (*i*) allow a strong cooperation and coordination between all coalition members, and thus benefiting from consensus and bargaining process capabilities to improve player utilities and avoid HAS scalability issues, and (*ii*) significantly reduce overhead in support of large-scale deployments. More details on how to construct this mapping model are given in the supplementary document.

Theorem 1. The proposed game $\langle P, V, CL \rangle$ is non-superadditive.

Proof. The proof is omitted here for brevity but included in the supplementary document (Appendix). ■

The stability of the created coalitions is proved using the *core* concept solution [3] that represents the set of optimal joint actions taken by every coalition, which ensures players' utility maximization and that none of the players has an incentive to leave the grand coalition and generate other coalitions.

Definition 2. For any coalition $Cl_\mu \in CL$ and $CL \subseteq P$, an action vector $A^v_{Cl_\mu} = \{a^v_{p_1}, \dots, a^v_{p_{N_{Cl_\mu}}}\} \in \mathcal{A}^v_{CL}$ is said to satisfy group rationality, if and only if $\sum_{p_k \in Cl_\mu} a^v_{p_k} = v(Cl_\mu)$. Thus, group rationality implies that for each coalition $Cl_\mu \in CL$, the coalition value $v(Cl_\mu)$ is distributed among its members.

Definition 3. An action vector $A^v_{Cl_\mu} \in \mathcal{A}^v_{CL}$ for any coalition $Cl_\mu \in CL$ and $CL \subseteq P$ is said to be individually rational if and only if player $p_k \in Cl_\mu$ can select the optimal action that maximizes its utility no less than acting alone, more specifically $a^{v,\star}_{p_k} \geq v(p_k), \forall p_k, k = \{1, 2, \dots, N\}$. Furthermore, if action vector $A^v_{Cl_\mu}$ satisfies group and individual rationalities, then it is called an imputation.

Definition 4. For any coalition $Cl_\mu \in CL$ and $CL \subseteq P$, an imputation $A^v_{Cl_\mu}$ is unstable if $v(Cl_\mu) > \sum_{p_k \in Cl_\mu} a^v_{p_k}$. The set C of coalitions CL is a stable imputation and it is called the *core*, where it is defined as follows:

$$C = \left\{A^v : \sum_{p_k \in P} a_{p_k} = v(P), \sum_{p_k \in Cl_\mu} a_{p_k} \geq v(Cl_\mu), \forall Cl_\mu \right\}. \quad (4)$$

Theorem 2. Our static coalition formation-based game $\langle P, V, CL \rangle$ satisfies the *core* definition.

Proof. The proof is similar to the one for Theorem 1. ■

Definition 5. A collection of coalitions in the grand coalition P, denoted by CL, is defined as $CL = \{Cl_1, Cl_2, Cl_3, Cl_4, Cl_5\}$ (see Figure 1 in the supplementary document) of mutually disjoint coalitions $\forall C_\mu \in CL$ and $CL \subseteq P$.

To ensure a preference level between the set of created coalitions CL in terms of decisions that the players have taken, utilities achieved, and the amount of bandwidth allocated, we introduce a *preference relation* (*i.e.*, comparison relation) that is characterized by the operator $\triangleright$ and defined as:

Definition 6. For any generated coalitions, a preference relation is defined to prefer one coalition over others and compare them. Let us consider two coalitions $(Cl_\mu, Cl_{\grave{\mu}}) \in CL$ and $CL \subseteq P$, then $Cl_\mu \triangleright Cl_{\grave{\mu}}$ implies that coalition Cl_μ is preferred over $Cl_{\grave{\mu}}$. Thus, our coalition rule results in five coalitions, where $Cl_5 \triangleright Cl_4 \triangleright Cl_3 \triangleright Cl_2 \triangleright Cl_1$.

Based on Definition 6, we further define two *order rules* [11], including the *utilitarian order* and the *Pareto order*, which are used to compare the set of created coalitions CL. Suppose coalition Cl_μ is preferred over coalition $Cl_{\grave{\mu}}$ ($Cl_\mu \triangleright Cl_{\grave{\mu}}$), and thus, the utilitarian order is defined as follows:

$$Cl_\mu \triangleright Cl_{\grave{\mu}} \iff \sum_{p_k \in Cl_\mu}^{k=[1\dots N_{Cl_\mu}]} v(Cl_\mu) > \sum_{p_{\grave{k}} \in Cl_{\grave{\mu}}}^{\grave{k}=[1\dots N_{Cl_{\grave{\mu}}}]} v(Cl_{\grave{\mu}}). \quad (5)$$

Furthermore, consider action vectors $A^v_{Cl_\mu}$ and $A^v_{Cl_{\grave{\mu}}}$ of Cl_μ and $Cl_{\grave{\mu}}$, respectively, where player $p_k \in Cl_\mu$ takes action $a^v_{p_k}$ and $p_{\grave{k}} \in Cl_{\grave{\mu}}$ takes action $a^v_{p_{\grave{k}}}$. The Pareto order is defined as:

$$Cl_\mu \triangleright Cl_{\grave{\mu}} \iff a^v_{p_k} \geq a^v_{p_{\grave{k}}}, \forall p_k \in Cl_\mu, \forall p_{\grave{k}} \in Cl_{\grave{\mu}}. \quad (6)$$

In (6), we require at least one strict inequality (>) for a player p_k for its actions taken during the streaming session. The Pareto order ensures that each player within a coalition improves its utility without hurting the utilities of other coalition members while considering network resource dynamics.

4.2 Stage 1.b – Bargaining and Consensus

After the creation of coalitions, we formulate the bitrate and quality decisions of the players as bargaining process and consensus problem, where the players that belong to similar coalitions form bargaining process agreements among themselves and reach an optimal decision consensus with their corresponding utilities (*i.e.*, bargaining outcome), respecting other coalition members' decisions and network resource variations. To achieve this, we use a generalized *Nash Bargaining Solution* (NBS) [11] with a weighted proportional fair resource sharing mechanism as a concept solution.

Formally, each coalition member $\forall p_k \in Cl_\mu, \forall Cl_\mu \in CL$ is trying to form a decision agreement and reach consensus over an outcome in space $\mathcal{A}_{CL}$ by selecting an action $a^v_{p_k} \in A^v_{Cl_\mu}$ that leads to utility $u^v_{p_k} \in U^v_{Cl_\mu}$. The utility function relationship f of player p_k, denoted by f_{p_k}, is defined over the space $(f_{CL} : \mathcal{A}_{CL} \rightarrow \mathcal{U}_{CL}) \cup \{\mathcal{Y}_{CL}\}$, with $(f_{Cl_\mu} : A^v_{Cl_\mu} \rightarrow U^v_{Cl_\mu}) \cup \{Y_{Cl_\mu}\}$ and $(f_{p_k} : a^v_{p_k} \rightarrow u^v_{p_k}) \cup \{y_{p_k}\}$, where $\mathcal{Y}_{CL}$ is the set of suboptimal decisions with their corresponding unsatisfactory utilities (*i.e.*, bargaining outcome disagreement). Let $\mathcal{U}_{CL}(\mathcal{A}_{CL})$ denote the utility function relationship that represents the set of possible actions with their achievable utilities for the set of coalitions CL. Thus, $U^v_{Cl_\mu}(A^v_{Cl_\mu})$ and $u^v_{p_k}(a^v_{p_k})$ represent a utility function relationship in terms of each coalition $Cl_\mu \in CL$ and one player p_k of its members, respectively. Based on this, the strategy space $\mathcal{S}_{CL}$ represents the action-to-utility relationship defined as:

$$\begin{cases} \mathcal{S}_{CL} = \{S_1, \dots, S_{Cl_\mu}, \dots, S_B | S_{Cl_\mu} \in \mathcal{U}_{CL}(\mathcal{A}_{CL})\}, \\ S_{Cl_\mu} = \{s_{p_1}, \dots, s_{p_k}, \dots, s_{p_{N_{Cl_\mu}}} | s_{p_k} \in U^v_{Cl_\mu}(A^v_{Cl_\mu})\}, \\ s_{p_k} = \{u^v_{i,p_k}(a^v_{i,p_k}) | \forall i = \{1, 2, \dots, K\}\}. \end{cases} \quad (7)$$

In addition, the outcome disagreement is defined as follows:

$$\begin{cases} \mathcal{Y}_{CL} = \{Y_1, \dots, Y_{Cl_\mu}, \dots, Y_B | Y_{Cl_\mu} \in \mathcal{U}^-{}_{CL}(\mathcal{A}^-{}_{CL})\}, \\ Y_{Cl_\mu} = \{y_{p_1}, \dots, y_{p_k}, \dots, y_{p_{N_{Cl_\mu}}} | y_{p_k} \in U^{-,v}_{Cl_\mu}(A^{-,v}_{Cl_\mu})\}, \\ y_{p_k} = \{u^{-,v}_{i,p_k}(a^{-,v}_{i,p_k}) | \forall i = \{1, 2, \dots, K\}\}, \end{cases} \quad (8)$$

where $\mathcal{Y}_{CL} \subset \mathcal{S}_{CL}, \mathcal{U}^-{}_{CL}(\mathcal{A}^-{}_{CL})$ is the outcome disagreement set of the grand coalition P, and $u^{-,v}_{i,p_k}(a^{-,v}_{i,p_k}) \in Y_{Cl_\mu}$ is also called the *disagreement point* of player $p_k \in Cl_\mu$.

In particular, we are interested in the bargaining solution, which is the function $\mathbb{F} : (\mathcal{S}_{CL}, \mathcal{Y}_{CL}) \to \mathbb{R}^n$ that gives a unique and fair Pareto optimal outcome for every bargaining problem $(\mathcal{S}_{CL}, \mathcal{Y}_{CL})$. Such a bargaining outcome should fulfill a set of axioms as below.

Definition 7. Any bargaining problem $(\mathcal{S}_{CL}, \mathcal{Y}_{CL})$, $\mathcal{X}^\star_{CL} = \mathbb{F}(\mathcal{S}_{CL}, \mathcal{Y}_{CL})$ is said to be NBS in $\mathcal{S}_{CL}$ for the disagreement points $\mathcal{Y}_{CL}$, if a set of the following axioms are satisfied, including:

- *Pareto efficiency*: $\mathcal{X}^\star_{CL} \geq \mathcal{Y}_{CL}, \forall p_k \in Cl_\mu, \forall Cl_\mu \in CL$ and $CL \subseteq P$, then $X^\star_{Cl_\mu} \geq Y_{Cl_\mu}$ with $x^\star_{p_k} \geq y_{p_k}$.
- *Feasibility*: $CL \subseteq P, \mathcal{X}^\star_{CL} \in \mathcal{S}_{CL}$.
- *Pareto optimality*: $\mathcal{X}^\star_{CL}$ is Pareto optimal.
- *Symmetry*: $\forall (p_k, p_{\grave{k}}) \in Cl_\mu, \forall Cl_\mu \in CL$, $(\mathcal{S}_{CL}, \mathcal{Y}_{CL})$ is invariant and symmetric around if and only if $s_{p_k} = s_{p_{\grave{k}}}$, and $y_{p_k} = y_{p_{\grave{k}}}$; then $\mathbb{F}(s_{p_k}, y_{p_k}) = \mathbb{F}(s_{p_{\grave{k}}}, y_{p_{\grave{k}}})$.
- *Independence of irrelevant alternatives*: Given two bargaining problems $(\mathcal{S}_{CL}, \mathcal{Y}_{CL})$ and $(\grave{\mathcal{S}}_{CL}, \mathcal{Y}_{CL})$ such that $\mathcal{X}^\star_{CL} \in \grave{\mathcal{S}}_{CL}, \grave{\mathcal{S}}_{CL} \subseteq \mathcal{S}_{CL}$, if $\mathcal{X}^\star_{CL} = \mathbb{F}(\mathcal{S}_{CL}, \mathcal{Y}_{CL})$, then $\mathcal{X}^\star_{CL} = \mathbb{F}(\grave{\mathcal{S}}_{CL}, \mathcal{Y}_{CL})$.
- *Invariance to equivalent utility representation*: For any linear scale transformation ψ, if the bargaining problem $(\mathcal{S}_{CL}, \mathcal{Y}_{CL})$ is transformed into another different bargaining problem $\psi(\mathcal{S}_{CL}, \mathcal{Y}_{CL}) = (\grave{\mathcal{S}}_{CL}, \grave{\mathcal{Y}}_{CL})$ with $\psi(\mathcal{S}_{CL}) = \grave{\mathcal{S}}_{CL}$ and $\psi(\mathcal{Y}_{CL}) = \grave{\mathcal{Y}}_{CL}$, then $\psi(\mathbb{F}(\mathcal{S}_{CL}, \mathcal{Y}_{CL})) = \mathbb{F}(\psi(\mathcal{S}_{CL}), \psi(\mathcal{Y}_{CL}))$.
- *Decision consensus*: For each coalition member $\forall p_k \in Cl_\mu, \forall Cl_\mu \in CL$, $x^\star_{p_1} \approx \ldots \approx x^\star_{p_k} \approx \ldots \approx x^\star_{p_{N_{Cl_\mu}}}$ (leading to $a^\star_{p_1} \approx \ldots \approx a^\star_{p_k} \approx \ldots \approx a^\star_{p_{N_{Cl_\mu}}}$). Thus, every coalition member seeks to come to an agreement and reach a consensus point in its decision.
- *Network and congestion*: For each step i, $C_{i,R^{agg}} = (BW^{HAS}_{i,R^{agg}} + BW^{bt}_{i,R^{agg}})/BW^{all}_{R^{agg}} \leq 1$.

Our NBS is a unique Pareto order bargaining solution if a set of axioms and the objective function $\mathbb{F}$ are satisfied, where:

$$\mathbb{F} : \begin{cases} \text{find } x^\star_{i,p_k} \Leftrightarrow \underset{k=[1\ldots N_{Cl_\mu}]}{u^{\star,v}_{i,p_k}(a^{\star,v}_{i,p_k})} \\ \arg \max\limits_{s_{i,p_k} \in S_{Cl_\mu}} \prod\limits_{p_k \in Cl_\mu} (s_{i,p_k} - y_{i,p_k})^{\alpha_{i,p_k}}, \\ \text{s.t. } \underset{k=[1\ldots N_{Cl_\mu}]}{s_{i,p_k}} \in S_{i,Cl_\mu}, S_{i,Cl_\mu} \in \mathcal{S}_{i,CL}, s_{i,p_k} \geq y_{i,p_k}, \\ \sum\limits_{p_k \in Cl_\mu} \alpha_{i,p_k} = 1, \alpha_{i,p_k} \in [0,1]. \end{cases} \tag{9}$$

Here, $\mathcal{S}_{CL}$ is a convex and compact set and $\mathbb{F}$ is strictly concave, and α_{p_k} is the *bargaining power* of player p_k, which is associated with every coalition member. The objective of the bargaining power is to assign a weight to the bargaining process, where the player with the highest α value receives the best final bargaining outcome.

Theorem 3. For each step i in our game, $\exists$ a unique Pareto optimal (PO) NBS (*i.e.*, a consensus point) that maximizes player utilities while avoiding HAS scalability issues.

Proof. The proof is omitted here for brevity but included in the supplementary document (Appendix). ■

To reach a unique (PO) NBS (*i.e.*, the global objective), each player p_k needs to know the *unknown* environment variable values of its coalition members $\forall p_{\grave{k}} \in Cl_\mu$-$\{p_k\}$, in particular, their optimal bargaining outcome $x^\star_{p_{\grave{k}}}$. At each step i, knowing these unknown variable values in a fully decentralized HAS system is a challenging task and usually requires a message exchange between coalition members or a centralized entity that is responsible for collecting player statuses to aggregate a global state. Nonetheless, the mentioned solutions are not practical in FDCHAS as they could introduce significant overhead. To achieve cooperation between the coalition members without incurring communication overhead cost, we design a novel *local control law policy* that combines potential [16] and pipelined consensus (*i.e.*, *memory dynamics*) [10] theories. This policy enables every coalition member to transform the global objective into a set of local objective functions $\mathcal{F}^x : \mathcal{X}^\star{}_{CL} \to \mathbb{R}, \forall p_k, \forall Cl_\mu \in CL$ with a minimization goal (*i.e.*, $\arg_{\min} \mathcal{F}^x$). Thus, they cooperatively fulfill the global objective.

Consider the global objective that is observed via a potential function ϕ such that $\phi^x : \mathcal{X}^\star{}_{CL} \to \mathbb{R}$. This function ensures that the local objective of each player correctly aligns with the global objective [16]. For each player $p_k \in Cl_\mu, \forall p_{\grave{k}} \in Cl_\mu$, let us denote by $X^\star_{Cl_\mu - \{p_k\}} \in \mathcal{X}^\star_{CL}$, $A_{Cl_\mu - \{p_k\}} \in \mathcal{A}_{CL}$ and $U_{Cl_\mu - \{p_k\}} \in \mathcal{U}_{CL}$ the joint bargaining outcome, action and utility of the coalition members without player p_k, respectively.

Definition 8. For each step i, for every player $p_k \in Cl_\mu, \forall Cl_\mu \in CL$ and $CL \subseteq P$, p_k's local objective function $\mathcal{F}^x : \mathcal{X}^\star{}_{CL} \to \mathbb{R}$, p_k finds the optimal bargaining outcome $x^\star_{i,p_k} \in X^\star_{i,Cl_\mu}$. Thus, the optimal action $a^{\star,v}_{i,p_k} \in A^v_{i,Cl_\mu}$ is selected, which leads to the optimal utility $u^{\star,v}_{i,p_k} \in U^v_{i,Cl_\mu}$. The function $\phi^x : \mathcal{X}^\star{}_{CL} \to \mathbb{R}$ is said to be a potential function if:

$$\begin{gathered} \mathcal{F}^x(\hat{\hat{x}}^\star_{i,p_k}, X^\star_{i,Cl_\mu - \{p_k\}}) - \mathcal{F}^x(\hat{x}^\star_{i,p_k}, X^\star_{i,Cl_\mu - \{p_k\}}) = \\ \phi^x(\hat{\hat{x}}^\star_{i,p_k}, X^\star_{i,Cl_\mu - \{p_k\}}) - \phi^x(\hat{x}^\star_{i,p_k}, X^\star_{i,Cl_\mu - \{p_k\}}), \\ \forall \hat{\hat{x}}^\star_{i,p_k}, \hat{x}^\star_{i,p_k} \in \mathcal{X}^\star{}_{CL}, \forall X^\star_{i,Cl_\mu - \{p_k\}} \in \mathcal{X}^\star{}_{CL - \{p_k\}}. \end{gathered} \tag{10}$$

Eqn. (10) shows that the potential function relies on a strong alignment assumption between the global objective and players' local objective functions. Thus, if any player unilaterally changes its decision, it requires an equal modification in both its local objective and potential functions. Hence, the consensus global objective function for every player $p_k \in Cl_\mu$, with $\forall p_{\grave{k}} \in Cl_\mu$ (other coalition members), $\forall Cl_\mu \in CL$ in terms of potential theory is defined by:

$$\phi^x(X^\star_{i,Cl_\mu}) = - \sum_{p_k \in Cl_\mu} \sum_{p_{\grave{k}} \in Cl_\mu - \{p_k\}} \frac{\| x^\star_{i,p_k} - x^\star_{i,p_{\grave{k}}} \|}{2}. \tag{11}$$

If (11) equals zero, the optimal consensus point is reached. The goal is to assign suitable local objective functions that are aligned with the global objective function $\forall p_k \in Cl_\mu, Cl_\mu \in CL$ such that $\mathcal{F}^x(X^\star_{i,Cl_\mu}) = \phi^x(X^\star_{i,Cl_\mu})$. The assignment of the local objective function of player p_k requires the capture of the decisions of all its coalition members. For this, we rewrite the local objective function as *wonderful life utility* (WLU) [28], where it will observe only the contribution margin to the potential function based on the accurate estimation e of the variables, such that

$$\mathcal{F}^x(x^\star_{i,p_k}, X^\star_{i,Cl_\mu - \{p_k\}}) = - \sum_{p_{\grave{k}} \in Cl_\mu - \{p_k\}} \| x^\star_{i,p_k} - x^{e,\star}_{i,p_{\grave{k}}} \|. \tag{12}$$

Since WLU can be assessed quite accurately, the estimated values of other coalition members by player p_k are almost equal to the original values that are computed by every other coalition member $x^{e,\star}_{i,p_{\grave{k}}} \triangleq x^\star_{i,p_{\grave{k}}}$, and thus $a^v_{i,p_{\grave{k}}} \triangleq a^{e,v}_{i,p_{\grave{k}}}$.

Theorem 4. At each step i, every coalition member $\forall p_k \in Cl_\mu$ with $\forall Cl_\mu \in CL$ satisfies its local objective function $\mathcal{F}^x$ locally without any explicit message exchange and reaches an optimal consensus point (*i.e.*, a unique (PO) NBS).

Proof. The proof is omitted here for brevity but included in the supplementary document (Appendix). ■

4.3 Stage 2 – Network Resource Allocation

The second stage of our game formulates the dynamic per-coalition network resource slicing, allocation and QoS provisioning as a Stackelberg strategic game [11]. The aggregation router holds a strong position and imposes a set of resource allocation rules upon all coalition members[3]. It is designated as the *leader*, while coalitions with their members require reacting to the allocation decisions and are called *followers*. Due to space limits, the second stage of the game along with its algorithm (see Algorithm 2 in the supplementary document) is explained at the conceptual level.

At each step i, the dynamic network resource slicing and allocation algorithm takes into consideration three main inputs, namely: (*i*) the joint decision taken by each coalition member, (*ii*) the coalition rule (2), and (*iii*) the preference relation ($\triangleright$) with its utilitarian and Pareto orders (Definition 6). The joint decision information could be piggybacked in the header of the HTTP responses carrying the segments, limiting the overhead without requiring an additional message. Given a set of coalitions $CL = \{Cl_1, \ldots, Cl_B\}$, with $B = 5$ and aggregation router R^{agg}, the leader allocates an amount of bandwidth $BW^{alloc,Cl_\mu}_{i,R^{agg}}$ (*i.e.*, a minimal per-coalition bandwidth slice guarantee) for every coalition $\forall Cl_\mu \in CL$ and $CL \subseteq P$ such that $\sum_{Cl_\mu \in CL} BW^{alloc,Cl_\mu}_{i,R^{agg}} < BW^{all}_{i,R^{agg}}$.

Definition 9. Given a Stackelberg strategic finite game that consists of set of coalitions CL with their players P and leader R^{agg}, the set $\mathcal{J}^{CL}_{i,R^{agg}} = \{J^{Cl_1}_{i,R^{agg}}, \ldots, J^{Cl_\mu}_{i,R^{agg}}, \ldots, J^{Cl_B}_{i,R^{agg}}\}$ is called the *reaction rule* and it is defined for each coalition $Cl_\mu \in CL$ at each step i as:

$$J^{Cl_\mu}_{i,R^{agg}} = \{A^{v}_{i,Cl_\mu} \in \mathcal{A}_{CL}, v_i(Cl_\mu) \le BW^{alloc,Cl_\mu}_{i,R^{agg}}, \\ \forall v_i(Cl_\mu) \in V, \forall p_k \in Cl_\mu, \forall Cl_\mu \in CL\}. \tag{13}$$

Based on Definition 9, we define the Stackelberg equilibrium as the optimal dynamic network resource allocation for each coalition as shown below.

Definition 10. At each step i, the set $\mathcal{J}^{CL}_{i,R^{agg}}$ of the allocated bandwidth for the set of coalitions CL reaches the Stackelberg equilibrium $\mathcal{J}^{\star,CL}_{i,R^{agg}}$ (*i.e.*, the optimal allocation decision) when the leader solves the optimization problem defined by the function $\mathcal{R}$ for any coalition $Cl_\mu \in CL$. $\mathcal{R}$ is defined as follows:

$$\mathcal{R}: \begin{cases} \text{find } J^{\star,Cl_\mu}_{i,R^{agg}} \Leftrightarrow BW^{\star,alloc,Cl_\mu}_{i,R^{agg}} \\ \arg \max\limits_{\forall U^{R^{agg}} \in \mathcal{U}_{R^{agg}}} U^{R^{agg}}_i, \\ \text{s.t. } v_i(Cl_\mu) \le BW^{alloc,Cl_\mu}_{i,R^{agg}}, \\ \quad \max\limits_{\forall U_{Cl_\mu} \in \mathcal{U}_{CL}} U_{i,Cl_\mu}, \forall Cl_\mu \in CL \text{ (see (17))}, \\ \quad BW^{HAS}_{i,R^{agg}} < BW^{all}_{i,R^{agg}} \text{ and } C_{i,R^{agg}} \le 1, \\ \quad Cl_\mu \triangleright Cl_{\grave{\mu}} \Leftrightarrow J^{\star,Cl_\mu}_{i,R^{agg}} \triangleright J^{\star,Cl_{\grave{\mu}}}_{i,R^{agg}}, \forall Cl_{\grave{\mu}} \in CL, \mu > \grave{\mu}, \end{cases} \tag{14}$$

where $\mathcal{U}_{R^{agg}} = \{U^{R^{agg}}_1, \ldots, U^{R^{agg}}_K\}$ represents the utility (in terms of profit) of the aggregation router R^{agg} at each step i.

4.4 Objective Function

We formulate the utility maximization problem of both the coalition members and the leader as a network utility maximization (NUM) problem [21] and show their relationship. During every step i, each coalition member and the aggregation router aim to optimize their utilities that are represented by the viewer QoE and profit, respectively. For this purpose, we denote by $\mathcal{U}_{i,CL}, U_{i,Cl_\mu}$ and u_{i,p_k} coalition CL's utilities, per-coalition player utilities and player p_k utility, respectively. The utility of the leader is defined as $U^{R^{agg}_i} \in \mathcal{U}_{R^{agg}}$. The defined utility functions are carefully designed to be strictly increasing concave functions, flexible enough to accommodate a set of dynamic constraints, and ensure a trade-off maximization between players' utilities and leader utility. This last property is beneficial for both as it ensures satisfactory viewer QoE while maintaining the profit of a service provider.

4.4.1 *Leader Utility Function.*

The leader utility function consists of profit captured by the amount of money that the leader manager can receive from its customers (HAS players) for its services, and the money paid for the network resources and use of the physical links. The leader concave objective function aims to ensure a trade-off between maximizing the leader utility while maintaining high utilities for all players in each coalition (see (14)).

4.4.2 *Player's Utility Function.*

The player utility function combines four main parameters that are considered rewards and penalties including the average segment perceptual quality (AvgSPQ), startup delay (SD), average quality switching (AvgQS) and stall events (SE). Thus, the utility of player p_k is defined as

$$u_{i,p_k} = QoE_{i,p_k} = \omega_1 \times AvgSPQ_{i,p_k} - \omega_2 \times AvgQS_{i,p_k} \\ - \omega_3 \times SE_{i,p_k} - \omega_4 \times SD_{i,p_k}. \tag{15}$$

For the non-negative weighting factors $\sum_{j=1}^{4} \omega_j = 1$, we performed many empirical tests for tuning by considering the recommendations of [13], and finally selected a value of 0.25 for each. The computation of each utility parameter is similar to the QoE model that is presented in [5]. Furthermore, we define the per-coalition utility, $\forall p_k \in Cl_\mu, \forall Cl_\mu \in CL$ as:

$$U_{i,Cl_\mu}(Agg(u_{i,p_k})) = AggQoE_{i,p_k} = \frac{1}{N_{Cl_\mu}} \sum_{p_k \in Cl_\mu} QoE_{i,p_k}. \tag{16}$$

Next, for each step i, player $p_k \in Cl_\mu$ solves the NUM-based concave objective function by finding the optimal action subject to a set of constraints (C.1–C.6 in (17) including buffer occupancy, CT-DR-SPT, consensus, potential function, network[4] and congestion level constraints), which are further explained in the supplementary document.

Eqn. (17) is our GT-based rule which is solved by combining a set of techniques accelerating the decision process while avoiding suboptimal results, including online decomposition methods (*i.e.*, dual decomposition and Lagrange duality) [21], dynamic programming and fast model predictive control (fastMPC) [27]. The basic idea of our optimization is to relax the original problem (17) by transferring the constraints to the objective in the form of a weighted sum (QoE function). When the problem

[3] We use per-coalition priority queues and meter tables for bandwidth slicing.

[4] $\tau^{max}(l_{p_k})$: max. downl. time required to fetch a segment encoded at bitrate l by p_k.

is relaxed, our optimization problem is decomposed into several subproblems and solved iteratively in a distributed manner. The computational complexity and overhead increases depending on the total number of players N and number of available bitrate levels with their corresponding qualities $QT(L)$. The overall complexity is independent of the number of segments K as the effect of frequent updates offsets the impact of a shorter horizon within each update ($\approx O(N \times |QT(L)|)$ iterations). The objective function is defined as:

$$\begin{cases} \left.\begin{aligned} &\text{find } a^{\star,v}_{i,p_k} \Leftrightarrow qt^{\star,v}_{i,p_k}(l^{\star,v}_{i,p_k}) \\ &\arg \max_{u^v_{p_k} \in U^v_{Cl_\mu},\, a^v_{p_k} \in A^v_{Cl_\mu}} u_{i,p_k} \end{aligned}\right\} \Leftrightarrow x^{\star}_{i,p_k} \in X^{\star}_{i,Cl_\mu} \\ \text{s.t. } buff^{min}_{p_k} \leq buff_{i,p_k} \leq buff^{max}_{p_k} & \textbf{C.1} \\ \quad \mathsf{MAP}_{SSIMplus}(a^{\star,v}_{i,p_k}, \{CT^u_{i,p_k}, DR_{i,p_k}, SPT_{i,p_k}\}) & \textbf{C.2} \\ \quad x^{\star}_{i,p_k} = u^{\star,v}_{i,p_k}(a^{\star,v}_{i,p_k}) = \mathbb{F}(s_{i,p_k}, y_{i,p_k}) & \textbf{C.3} \\ \quad \phi^x(X^{\star}_{i,Cl_\mu}) = \mathcal{F}^x(X^{\star}_{i,Cl_\mu}) \approx 0 & \textbf{C.4} \\ \quad l^{\star,v}_{i,p_k} \leq bw^e_{i,p_k} \Leftrightarrow \tau^{max}(l^{\star,v}_{i,p_k}) \leq \tau, \forall l^{\star,v}_{i,p_k} \in L & \textbf{C.5} \\ \quad c^e_{i,p_k} \leq 1, \text{ where } c^e_{i,p_k} = (l^{\star,v}_{i,p_k} + bw^{e,bt}_{i,p_k})/bw^e_{i,p_k} & \textbf{C.6} \end{cases} \tag{17}$$

5 PERFORMANCE EVALUATION

We implemented the FDCHAS scheme, named `FDCHAS.js` within the current stable release (v2.4.1) of the reference player dash.js [7] and made the FDCHAS player available on our demo Web site[5]. For testing we conducted a set of VoD experiments using different numbers of HAS players (*i.e.*, varying from 1 to 100), bandwidth variation profiles obtained from DASH-IF and the 3G/HSDPA [23], coalition rule features (DR, CT, and SPT) and test environments (*e.g.*, mobile/fixed player, and arbitrary player arrival/departure times). The test scenarios are described in Table 2.

Table 2: Test scenarios (fixed (F) ≡ one value, various (V) ≡ different possible values, dynamic (D) ≡ dynamic # of players).

Variables	Test 1	Test 2	Test 3	Test 4
CT	F	F	V	V
DR	V	V	V	V
SPT	F	V	V	F
# of players	F/D	F/D	F/D	F/D

Table 3: Parameters used in the experiments.

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

[5][Online]. Available: http://streaming.university/GTA/

Due to space limitations, we present one test scenario (Test 3) that represents a realistic multi-premise last-mile streaming environment. It consists of a fixed number of 100 heterogeneous players (20 per coalition) that compete for 170 Mbps of total bandwidth at a bottleneck link. `FDCHAS.js` is compared against eight well-known HAS adaptation schemes from the literature: PANDA [18], FESTIVE [13], QDASH [19], SDNDASH [5], BOLA [25], BBA [12], ELASTIC [8] and the original dash.js [7]. Furthermore, we derive an offline optimal bound that is computed using dynamic programming with complete future player and network information. We then compare the per-coalition efficiency of `FDCHAS.js` with available-bandwidth-based and buffer-based policies [7]. The results were compared using four HAS scalability metrics: presentation quality stability, fairness, bandwidth utilization and QoE. The HAS server stores five content types (animation, documentary, movie, news and sports) with different resolutions and bitrate levels [17] L = {45, 100, 150, 200, 250, 300, 400, 500, 600, 700, 900, 1200, 1500, 2000, 2100, 2400, 2900, 3300, 3600, 4000} Kbps. We also installed a *Dummynet*[6] traffic shaper to throttle the bandwidth and used *iperf* to generate dynamic background traffic[7] to emulate a typical, realistic multi-player shared bottleneck link scenario. In addition, a PANDA-based bandwidth estimator was used to predict the network resources. The values of the experimental parameters are listed in Table 3.

We conducted two experiments for each test scenario to evaluate the effectiveness of FDCHAS. First, we compare the average results over all coalition members against the offline optimal bound, the available-bandwidth-based and buffer-based heuristics of the original dash.js in terms of presentation quality, quality oscillations, video stalls, startup delay, utility (normalized QoE, see Table 3), and convergence time (Conv-T). Second, we average the results over all players (all coalition members N) and compare them with eight well-known adaptation schemes.

Tables 5 and 6 represent the per-coalition average results and their offline optimal bounds, respectively, while Table 4 highlights the overall average results over all players. Table metrics can be found in [5, 6, 25]. In Table 5, we observe that `FDCHAS.js` achieves better per-coalition results compared to the other schemes and closely tracks the offline optimal bounds given in Table 6. `FDCHAS.js` provides the highest and most stable per-coalition average representation perceptual quality that ranges from 0.885 to 0.918 with a maximum utility of 4.33 (*i.e.*, a higher average utility implies that there is no player deviating from the grand coalition as shown in Theorem 2). Oscillations are low at 3.6, buffer stalls at 1.6, startup delays at 3.93 s, and convergence time at 14.38 s for all coalitions. These results are unrivaled, but not unexpected since `FDCHAS.js` leverages game and consensus theories that allow a high level of collaboration between coalition members. Thus, only the optimal decisions are selected, aimed at maximizing viewer QoE and avoiding HAS scalability issues.

In Table 4, similarly excellent results are observed where `FDCHAS.js` achieves very close average results to the offline optimal bound and significantly outperforms PANDA, FESTIVE, QDASH, SDNDASH, BOLA, BBA, ELASTIC, and the original dash.js schemes,

[6][Online]. Available: http://info.iet.unipi.it/~luigi/dummynet/, https://iperf.fr/
[7]Our solutions consider the players who choose not to join the cooperation community (*i.e.*, other bitrate adaptation schemes) as background traffic in the model.

Table 4: Average total quality stability, fairness, utilization and QoE, Test 3.

	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

Table 5: Average presentation quality stability and QoE metrics, Test 3.

	FDCHAS.js				**Available-rate-based**				**Buffer-based**			
	AVG Quality (SSIMplus)	AVG # of Oscillations & Stalls	AVG Startup Delay	AVG Utility & Conv-T	AVG Quality (SSIMplus)	AVG # of Oscillations & Stalls	AVG Startup Delay	AVG Utility & Conv-T	AVG Quality (SSIMplus)	AVG # of Oscillations & Stalls	AVG Startup Delay	AVG Utility & Conv-T
Cl_1	0.812 to 0.827	3 & 2 (0.23 s)	2.45 s	3.82 & 12 s	0.692 to 0.909	62 & 21 (29.1 s)	11.3 s	2.7 & 109 s	0.692 to 0.918	18 & 10 (10.33 s)	4.9 s	3.11 & 34.5 s
Cl_2	0.848 to 0.86	4 & 1 (0.5 s)	4.2 s	4.14 & 15.2 s	0.692 to 0.95	42 & 25 (30 s)	9.7 s	2.9 & 85 s	0.78 to 0.942	33 & 17 (13.8 s)	8.56 s	3.33 & 25.6 s
Cl_3	0.89 to 0.914	3 & 1 (0.1 s)	4.25 s	4.41 & 12 s	0.692 to 0.961	55 & 16 (11.2 s)	7.6 s	3.66 & 54 s	0.81 to 0.95	19 & 9 (6.88 s)	11 s	3.2 & 59 s
Cl_4	0.918 to 0.944	5 & 2 (0.11 s)	5 s	4.55 & 21.7 s	0.692 to 0.987	61 & 30 (22.7 s)	12.1 s	3.07 & 91 s	0.87 to 0.983	24 & 7 (4.69 s)	7.7 s	3.89 & 23 s
Cl_5	0.957 to 0.972	3 & 2 (3.29 s)	3.77 s	4.75 & 11 s	0.692 to 0.99	53 & 19 (15.9 s)	10 s	3.8 & 44 s	0.89 to 0.99	17 & 11 (8.2 s)	5.1 s	3.97 & 18.55 s
Avg	0.885 to 0.91	3.6 & 1.6 (0.85 s)	3.93 s	4.33 & 14.38 s	0.692 to 0.959	54.6 & 22.2 (21.78 s)	10.14 s	3.22 & 76.6 s	0.808 to 0.956	22.2 & 10.8 (8.78 s)	7.45 s	3.5 & 32.13 s

Table 6: Per-coalition offline optimal bounds, Test 3.

	AVG Quality (SSIMplus)	AVG # of Oscillations & Stalls	AVG Startup Delay	AVG Utility & Conv-T
Cl_1	0.823 to 0.835	2 & 0 (0 s)	2.3 s	4 & 8.6 s
Cl_2	0.858 to 0.866	2 & 1 (0.2 s)	3.3 s	4.33 & 10.1 s
Cl_3	0.904 to 0.918	1 & 1 (0.1 s)	4 s	4.59 & 11.6 s
Cl_4	0.919 to 0.949	3 & 1 (0.07 s)	4.7 s	4.82 & 14.26 s
Cl_5	0.964 to 0.99	3 & 2 (0.5 s)	2.1 s	4.9 & 9.45 s
Avg	0.893 to 0.91	2.2 & 1 (0.174 s)	3.28 s	4.52 & 10.80 s

which are suffering from scalability issues and major fluctuations. `FDCHAS.js` significantly reduces quality instability, QoE unfairness (based on Jian's index) and network resource underutilization by 99.1%, 98.7%, and 96.7%, respectively, while topping the viewer QoE at an average of 4.33. Furthermore, `FDCHAS.js` selects high and stable bitrate levels that range from 1200 to 1500 Kbps while maintaining a consistent perceptual quality that ranges from 0.885 to 0.918 with a variance of 0.033, on average. It achieves infrequent quality oscillations (an average of 3.6), few stalls (an average of 1.6 and 0.85 s stall duration) and low startup delay (3.93 s). Similarly, `FDCHAS.js` quickly converges to the optimal solution at an average time of 14.38 s, and it ensures a full utilization of network resources with an average congestion level of 0.967 without any violations[8] (please refer to the network and congestion axiom in Section 4.2). Finally, `FDCHAS.js` is highly scalable due to the fact that it uses fully distributed coordination and collaboration between the `FDCHAS.js` players without any explicit message exchanges.

We analyzed the maximum computational complexity of the objective function (17) and found that it required approximately 2,000 iterations over a streaming session with $K = 150$ segments. Each iteration took 7.15 milliseconds and all 100 players updated their statuses in parallel. Hence, the computation to find the optimal decisions over this streaming session took approximately 14.3 seconds. The results compared to the existing techniques are summarized in Table 7.

`FDCHAS.js` achieves these outcomes because of several reasons. (*i*) Players form coalitions using the coalition rule in (2); thus, the HAS players use joint actions to gain mutual benefits (*i.e.*, utility maximization). Further, using a bargaining process and consensus together with potential and local functions allows the players to select the optimal actions and reach consensus decisions without any extra signaling overhead between them, and also without introducing any deviating players. (*ii*) The use of the fastMPC technique provides an accurate estimation of the network resource dynamics for a few decision steps in advance, which also helps eliminate suboptimal solutions, reduces the convergence time and startup delay. (*iii*) `FDCHAS.js` uses the PANDA algorithm to estimate the available bandwidth accurately, especially in cases of time intervals with long-term variations. (*iv*) The Stackelberg network resource allocation dynamically selects a suitable slice of bandwidth to accommodate every coalition member's decision while maximizing the service provider profits.

[8]When the sum of players' demands is greater than the available bandwidth, a *bandwidth violation* occurs, and network congestion grows ($C \approx 1$ is good with full utilization, while $C > 1$ is bad with oversubscription).

Table 7: Summary of results. Percentage improvements of FDCHAS over the existing techniques, at scale.

FDCHAS Improvement vs.	PANDA %	FESTIVE %	QDASH %	SDNDASH %	BOLA %	BBA %	ELASTIC %	dash.js %
Quality Stability	1.5	1.2	97.1	1	43	23	11.5	15
QoE Fairness	17	15	61	4.5	50	32	21	52
Network Utilization	14	19	26	10	51	26	33	44
Viewer 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
Startup Delay	32	12	91	6.9	32	59	15	61

6 CONCLUSIONS

Leveraging a two-stage game, we designed FDCHAS, a fully distributed collaborative bitrate selection scheme for the HAS-based VoD services. FDCHAS largely eliminates HAS scalability issues. We provided for both stages theoretical guarantees and experimental evaluations. Results show that the FDCHAS player (`FDCHAS.js`) achieves high efficiency across all players, can be practically implemented, adheres to the spirit of distributed and client-driven HAS, all while significantly outperforming other state-of-the-art adaptation schemes. As future work, we plan to extend FDCHAS to support (*i*) networks with multiple shared bottleneck links and live streaming services, and (*ii*) dynamic coalition formation, including the analysis of the deviating players, and its theoretical guarantees.

ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China under Grant No. 61472266, in part by the National University of Singapore (Suzhou) Research Institute, and in part by grant 31T102-UPAR-1-2017 from UAE University.

REFERENCES

[1] Saamer Akhshabi, Lakshmi Anantakrishnan, Ali C. Begen, and Constantine Dovrolis. 2012. What Happens when HTTP Adaptive Streaming Players Compete for Bandwidth?. In *Proceedings of the 22Nd International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV '12)*. ACM, New York, NY, USA, 9–14. https://doi.org/10.1145/2229087.2229092

[2] Saamer Akhshabi, Ali C. Begen, and Constantine Dovrolis. 2011. An Experimental Evaluation of Rate-adaptation Algorithms in Adaptive Streaming over HTTP. In *Proceedings of the Second Annual ACM Conference on Multimedia Systems (MMSys '11)*. ACM, New York, NY, USA, 157–168. https://doi.org/10.1145/1943552.1943574

[3] Giacomo Bacci, Samson Lasaulce, Walid Saad, and Luca Sanguinetti. 2016. Game Theory for Networks: A Tutorial on Game-Theoretic Tools for Emerging Signal Processing Applications. *IEEE Signal Processing Magazine* 33, 1 (Jan 2016), 94–119. https://doi.org/10.1109/MSP.2015.2451994

[4] Abdelhak Bentaleb, Ali C. Begen, Saad Harous, and Roger Zimmermann. 2018. Want to Play DASH?: A Game Theoretic Approach for Adaptive Streaming over HTTP. In *Proceedings of the 9th ACM Multimedia Systems Conference (MMSys '18)*. ACM, New York, NY, USA, 13–26. https://doi.org/10.1145/3204949.3204961

[5] Abdelhak Bentaleb, Ali C. Begen, and Roger Zimmermann. 2016. SDNDASH: improving QoE of HTTP adaptive streaming using software defined networking. In *Proceedings of the 2016 ACM on Multimedia Conference (MM '16)*. ACM, New York, NY, USA, 1296–1305. https://doi.org/10.1145/2964284.2964332

[6] Abdelhak Bentaleb, Ali C Begen, Roger Zimmermann, and Saad Harous. 2017. SDNHAS: An SDN-Enabled Architecture to Optimize QoE in HTTP Adaptive Streaming. *IEEE Transactions on Multimedia* 19, 10 (Oct 2017), 2136–2151. https://doi.org/10.1109/TMM.2017.2733344

[7] DASH Reference Player. 2017. [Online]. Available: https://goo.gl/o9oC4n. Online; accessed 15 July 2017.

[8] Luca De Cicco, Vito Caldaralo, Vittorio Palmisano, and Saverio Mascolo. 2013. ELASTIC: A Client-Side Controller for Dynamic Adaptive Streaming over HTTP (DASH). In *2013 20th International Packet Video Workshop*. 1–8. https://doi.org/10.1109/PV.2013.6691442

[9] Zhengfang Duanmu, Kai Zeng, Kede Ma, Abdul Rehman, and Zhou Wang. 2017. A Quality-of-Experience Index for Streaming Video. *IEEE Journal of Selected Topics in Signal Processing* 11, 1 (Feb 2017), 154–166. https://doi.org/10.1109/JSTSP.2016.2608329

[10] Golnaz Habibi, Zachary Kingston, Zijian Wang, Mac Schwager, and James McLurkin. 2015. Pipelined Consensus for Global State Estimation in Multi-agent Systems. In *ICAAMS*. 1315–1323.

[11] Zhu Han, Dusit Niyato, Walid Saad, Tamer Baar, and Are Hjrungnes. 2012. *Game Theory in Wireless and Communication Networks: Theory, Models, and Applications* (1st ed.). Cambridge University Press, New York, NY, USA.

[12] Te-Yuan Huang, Ramesh Johari, Nick McKeown, Matthew Trunnell, and Mark Watson. 2014. A Buffer-based Approach to Rate Adaptation: Evidence from a Large Video Streaming Service. In *Proceedings of the 2014 ACM Conference on SIGCOMM (SIGCOMM '14)*. ACM, New York, NY, USA, 187–198. https://doi.org/10.1145/2619239.2626296

[13] Junchen Jiang, Vyas Sekar, and Hui Zhang. 2012. Improving Fairness, Efficiency, and Stability in HTTP-based Adaptive Video Streaming with FESTIVE. In *Proceedings of the 8th International Conference on Emerging Networking Experiments and Technologies (CoNEXT '12)*. ACM, New York, NY, USA, 97–108. https://doi.org/10.1145/2413176.2413189

[14] Diego Kreutz, Fernando MV Ramos, Paulo Esteves Verissimo, Christian Esteve Rothenberg, Siamak Azodolmolky, and Steve Uhlig. 2015. Software-Defined Networking: A Comprehensive Survey. *Proc. IEEE* 103, 1 (Jan 2015), 14–76. https://doi.org/10.1109/JPROC.2014.2371999

[15] Jonathan Kua, Grenville Armitage, and Philip Branch. 2017. A Survey of Rate Adaptation Techniques for Dynamic Adaptive Streaming Over HTTP. *IEEE Communications Surveys Tutorials* 19, 3 (thirdquarter 2017), 1842–1866. https://doi.org/10.1109/COMST.2017.2685630

[16] Quang Duy Lã, Yong Huat Chew, and Boon-Hee Soong. 2016. *Potential Game Theory*. Springer.

[17] Stefan Lederer, Christopher Müller, and Christian Timmerer. 2012. Dynamic Adaptive Streaming over HTTP Dataset. In *Proceedings of the 3rd Multimedia Systems Conference (MMSys '12)*. ACM, New York, NY, USA, 89–94. https://doi.org/10.1145/2155555.2155570

[18] Zhi Li, Xiaoqing Zhu, Joshua Gahm, Rong Pan, Hao Hu, Ali C Begen, and David Oran. 2014. Probe and Adapt: Rate Adaptation for HTTP Video Streaming At Scale. *IEEE Journal on Selected Areas in Communications* 32, 4 (April 2014), 719–733. https://doi.org/10.1109/JSAC.2014.140405

[19] Ricky K. P. Mok, Xiapu Luo, Edmond W. W. Chan, and Rocky K. C. Chang. 2012. QDASH: A QoE-aware DASH System. In *Proceedings of the 3rd Multimedia Systems Conference (MMSys '12)*. ACM, New York, NY, USA, 11–22. https://doi.org/10.1145/2155555.2155558

[20] Reza Olfati-Saber, J Alex Fax, and Richard M Murray. 2007. Consensus and Cooperation in Networked Multi-Agent Systems. *Proc. IEEE* 95, 1 (Jan 2007), 215–233. https://doi.org/10.1109/JPROC.2006.887293

[21] Daniel Pérez Palomar and Mung Chiang. 2006. A Tutorial on Decomposition Methods for Network Utility Maximization. *IEEE Journal on Selected Areas in Communications* 24, 8 (Aug 2006), 1439–1451. https://doi.org/10.1109/JSAC.2006.879350

[22] Abdul Rehman, Kai Zeng, and Zhou Wang. 2015. Display Device-adapted Video Quality-of-Experience Assessment. In *SPIE/IS&T Electronic Imaging*. Int. Society for Optics and Photonics, 939406–939406.

[23] Haakon Riiser, Paul Vigmostad, Carsten Griwodz, and Pål Halvorsen. 2013. Commute Path Bandwidth Traces from 3G Networks: Analysis and Applications. In *Proceedings of the 4th ACM Multimedia Systems Conference (MMSys '13)*. ACM, New York, NY, USA, 114–118. https://doi.org/10.1145/2483977.2483991

[24] Ashkan Sobhani, Abdulsalam Yassine, and Shervin Shirmohammadi. 2017. A Video Bitrate Adaptation and Prediction Mechanism for HTTP Adaptive Streaming. *ACM Trans. Multimedia Comput. Commun. Appl.* 13, 2, Article 18 (March 2017), 25 pages. https://doi.org/10.1145/3052822

[25] Kevin Spiteri, Rahul Urgaonkar, and Ramesh K Sitaraman. 2016. BOLA: Near-optimal Bitrate Adaptation for Online Videos. In *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*. 1–9. https://doi.org/10.1109/INFOCOM.2016.7524428

[26] Yi Sun, Xiaoqi Yin, Junchen Jiang, Vyas Sekar, Fuyuan Lin, Nanshu Wang, Tao Liu, and Bruno Sinopoli. 2016. CS2P: Improving Video Bitrate Selection and Adaptation with Data-Driven Throughput Prediction. In *Proceedings of the 2016 ACM SIGCOMM Conference (SIGCOMM '16)*. ACM, New York, NY, USA, 272–285. https://doi.org/10.1145/2934872.2934898

[27] Yang Wang and Stephen Boyd. 2010. Fast Model Predictive Control Using Online Optimization. *IEEE Transactions on Control Systems Technology* 18, 2 (March 2010), 267–278. https://doi.org/10.1109/TCST.2009.2017934

[28] David H Wolpert and Kagan Tumer. 2001. Optimal Payoff Functions for Members of Collectives. *Advances in Complex Systems* 4, 02n03 (2001), 265–279.

[29] Hui Yuan, Huayong Fu, Ju Liu, Junhui Hou, and Sam Kwong. 2018. Non-Cooperative Game Theory Based Rate Adaptation for Dynamic Video Streaming over HTTP. *IEEE Transactions on Mobile Computing* (2018), 1–1. https://doi.org/10.1109/TMC.2018.2800749