Automated Bidding for Media Services at the Edge of a Content Delivery Network

Nicholas Mastronarde, Mihaela van der Schaar
Department of Electrical Engineering (EE), University of California Los Angeles (UCLA), Los Angeles, CA
{nhmastro, mihaela} @ ee.ucla.edu

***Abstract*—We investigate the problem of providing media services to multiple autonomous wireless users at the edge of a Content Delivery Network (CDN) in a setting where wireless resources are priced based on real-time market demands. Our focus is on the multimedia service resource negotiation process, which is performed prior to the actual media transmission. We adopt the Progressive Second Price (PSP) auction mechanism, which is used to determine the network resource allocation to the users and a corresponding tax for the consumed resources. Our interest in this negotiation mechanism lies in understanding a single user's (or *agent*'s) ability to learn to improve its bids over time in order to increase its own utility in the face of time-varying resource valuations and contention for resources with other users. We pay particular attention to the implementation complexity and the information requirements of the agent's deployed learning rule, and we quantify the impact of these factors on the rule's ultimate performance (i.e. the cumulative utility achieved over time) and efficiency (i.e. the utility gained per unit of computation). These factors are especially important in the mobile video streaming context, where limited resources must be efficiently utilized, and where communication and computation overheads can significantly impact the quality of service experienced by the user.**

***Index Terms*—Multimedia service middleware, Resource Negotiation, Session Negotiation, Multi-agent learning, Content Delivery Networks.**

I. Introduction

At the edge of content delivery networks, independent *media-service providers* [1] deliver network-content to multiple *autonomous mobile users* through their wireless radios. The number of users that access a single media-service provider may vary depending on the available infrastructure (e.g an 802.11-enabled access point versus a cellular network base-station). Regardless of the type of infrastructure-to-mobile wireless link, there are always limited network resources (i.e. bandwidth). Unfortunately, such bandwidth constraints conflict with the resource requirements of real-time, bandwidth intense, video streaming applications, which we will focus on in this paper. That being the case, resource allocation becomes of paramount importance in ensuring that multiple autonomous users fairly and optimally *share* the scarce wireless resources.

We frame the resource allocation problem as an auction in which wireless resources are priced based on real-time market demands [2][8]. In contrast to charging a flat-rate (as is currently done for most services provided over cellular networks and the internet), this auction based framework allows media-service providers and other owners of communication bandwidth to manage the network resources by charging users based on their bandwidth usage and the real-time market price of those resources at the time of their usage. This obviates the need for service providers to implement controversial bandwidth “throttling” algorithms, which slow traffic from big consumers of bandwidth (such as those participating in peer-to-peer video streaming and other multimedia services), because, during times of high congestion, users would have to pay a premium for the bandwidth required by these services.

By framing the network resource management problem as an auction, the media-service provider is no longer the sole decider of how resources are allocated among users. Individual users now have incentive to actively participate in the resource negotiating process because they can influence their resource allocations through their bids. We assume that a user’s mobile device automates the bidding process by acting as the user’s *agent*; that is, it submits bids to the media-service provider on the user’s behalf. Importantly, an agent’s bid impacts its resource allocation as well as the resource allocations of the other competing agents, which we will regard as *opponents* in the auction. This auction game is played repeatedly over time such that agents can request more or less resources based on the congestion that they experience and their time-varying resource valuations.

In order to improve their bids over time (i.e. increase their payoffs), we assume that agents receive (limited) feedback from the media-server about their opponents' bids. Using this information, agents can *learn to bid* so that their utility is improved over time. For example, the agent can analyze its opponents' previously submitted bid quantities and then submit a bid commensurate with the anticipated future congestion level and its current valuation of the resources.

In this paper, we adopt the Progressive Second Price (PSP) auction[1] mechanism [2] as the foundation of the decentralized media-service resource negotiation. In the early work on the PSP auction [2], the authors assume that (i) an agent may learn to bid based on complete knowledge of its opponents' bid profile in the previous time slot; (ii) bids are always submitted asynchronously, with only one agent submitting a new bid in each time slot; and, (iii) all agents deploy the same learning rule (i.e. "self-play" assumptions [4]). These assumptions are designed to study the equilibrium and convergence properties of the PSP auction, but they are not appropriate for the considered multimedia resource allocation scenario for two reasons. Firstly, each agent (i.e. mobile device) has limited (but different) computational capabilities, so it may be infeasible for it to process all of its opponents' bids in order to determine the bid that will maximize its payoffs within a tolerable delay. Secondly, due to the decentralized nature of the wireless resource allocation problem, and the possibility that there are many agents requesting

[1] However, it should be noted that techniques proposed in this paper could be deployed in conjunction with other auction mechanisms.

resources, possibly significant communication overheads are incurred as the media-server repeatedly updates agents with information about *all* of their opponents' bids. Hence, unlike in conventional game-theoretic solutions, we shift our attention away from equilibrium concepts in favor of *modeling, analyzing, and improving the dynamic behavior of interacting users* in dynamic settings, out of equilibrium, while also explicitly considering the information requirements and implementation constraints occurring when deploying multimedia applications.

Considering the range of communication and processing capabilities in existing mobile devices (e.g. smart phones, PDAs, laptops), it is clear that there cannot be a one-size-fits-all learning algorithm. In other words, learning algorithms with various informational and computational requirements are necessary so that a wide range of mobile devices can all participate in the auction game. Despite this need for complexity- and information-adaptive learning rules in real-life multi-agent systems, most literature on multi-agent learning in games only considers two categories of learning rules. These learning rules land on the extremes of both informational and computational requirements and have not been designed with any realistic application in mind (i.e. they are general and not application specific). For example, payoff-based learning rules (e.g. reinforcement learning [3]) are simple to implement, but they assume that no a priori information about the system is available. Meanwhile, probabilistic learning rules (e.g. fictitious play [4]) are informationally and computationally prohibitive, which precludes their applicability in the distributed mobile context.

Our contributions in this paper are as follows:

- We cast the media-service resource negotiation problem at the edge of the CDN as an auction game among multiple agents. In this auction scenario, the agents' resource valuations are defined based on operational rate-distortion models for the encoded video sequences. Hence, their valuations depend on such factors as the video source's characteristics (e.g. high-motion, detailed textures) and multimedia format (e.g. MPEG-2, H.264/AVC). Importantly, the agent's utility, which is a quasi-linear function of the resource valuation and cost, is aligned with the user's desire to achieve a high-quality video at a low cost.

- We investigate different levels of centralized coordination in the auction game. Under the proposed coordination policies, the media-server can poll a variable number of agents to submit a new bid in each time slot, thereby impacting the learning dynamics.
- We introduce learning rules that require an agent to acquire different levels of information from the media-service provider about its opponents, and we consider the computational requirements (in floating point operations) of the different learning rules. We also consider heterogeneous opponents, which all deploy different learning rules.
- Finally, we propose two evaluation metrics with which we can quantify (i) the value of the information required by an agent to deploy a particular learning rule (i.e. the cumulative impact of more or less information about its opponents' bids on its utility), (ii) the efficiency of the learning rule in terms of achieved utility per unit of computational complexity, (iii) the cumulative impact of different levels of centralized coordination on an agent's utility, and (iv) the utility impact when agents deviate from "self-play" assumptions, and employ heterogeneous learning rules with varying informational and computational requirements.

The remainder of this paper is organized as follows. In Section II, we describe the system setup and the PSP auction mechanism. We then formalize the agents' goals and the repeated media-service resource negotiation framework. In Section III, we describe several coordination policies that can be employed in this framework. In Section IV, we introduce learning rules that vary in computational and informational complexity and, we propose two metrics for evaluating and comparing these learning rules. In Section V, we present our experimental results and in Section VI we conclude the paper.

II. Media-Services at the Edge of the Content Delivery Network

This section presents our proposed media-service model for auctioning wireless resources at the edge of a CDN.

A. *System setup*

We consider a system for delivering multimedia content to users at the edge of a CDN. The system involves one or more content distributors within the CDN, which deliver on-demand video streams to users through an intermediate media-service provider.

To accommodate the various users, the media-service provider performs real-time transcoding to seamlessly convert the content distributor's video streams to a format compatible with the end-user's device [1] [6], and to meet the bandwidth constraints imposed by the user's resource allocation. In return, the users pay a tax to the media-service provider. The tax that each user pays increases with (i) the amount of bandwidth used by the media-service provider to stream the user's video, (ii) the current demand for the media-server's bandwidth, and (iii) the other users' valuations of the resources (as represented by their bids).

In this paper, we present a solution based on auction theory for deploying media-services in which multiple users must share a single media-service provider's bandwidth. We assume that there are M such autonomous users. The total network resources that must be shared among the users is R (kb/s).[2] The agents, in the context of our auction game, are the M mobile video decoders, which are indexed by $i \in \mathcal{I} = \{1, \ldots, M\}$ (one agent per user). These agents will play the

[2] The total network resources could also be expressed as a fraction of time (i.e. transmission opportunity duration) within each service interval. This time-allocation would then have to be converted to a rate allocation, which would depend on each user's channel conditions. For simplicity, however, we assume that the rate allocation is divided among the users.

auction game in every time slot t in order to compete for the available wireless resources on behalf of the users.

The flow diagram in Fig. 1 details the negotiation between the agents and the media-server, which auctions off the available bandwidth in each time slot. The figure illustrates the internal logic flow of the agents and the media-server, as well as the information that is exchanged between them at different stages of the negotiation. The notation in the figure will be introduced throughout the rest of this section. The shaded functional blocks are the most significant components of our proposed solution for media-service negotiation. We discuss them in their respective sections, as specified in the figure.

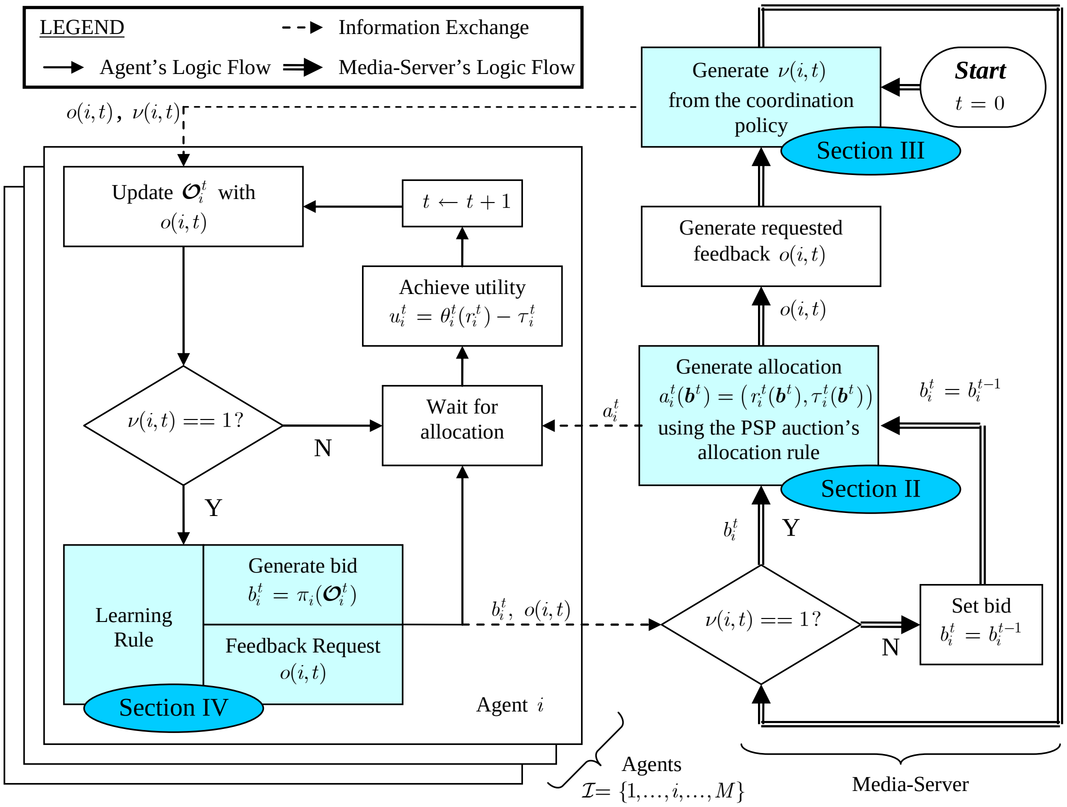
LEGEND
Information Exchange
Agent's Logic Flow
Media-Server's Logic Flow
$o(i,t)$, $\nu(i,t)$
Update $\mathcal{O}_i^t$ with $o(i,t)$
$t \leftarrow t+1$
Achieve utility $u_i^t = \theta_i^t(r_i^t) - \tau_i^t$
$\nu(i,t) == 1?$
N
Y
Wait for allocation
Learning Rule
Generate bid $b_i^t = \pi_i(\mathcal{O}_i^t)$
Feedback Request $o(i,t)$
Section IV
Agent i
b_i^t, $o(i,t)$
a_i^t
Generate $\nu(i,t)$ from the coordination policy
Section III
Start
$t = 0$
Generate requested feedback $o(i,t)$
$o(i,t)$
Generate allocation $a_i^t(\boldsymbol{b}^t) = (r_i^t(\boldsymbol{b}^t), \tau_i^t(\boldsymbol{b}^t))$ using the PSP auction's allocation rule
Section II
$b_i^t = b_i^{t-1}$
b_i^t
Y
$\nu(i,t) == 1?$
N
Set bid $b_i^t = b_i^{t-1}$
Agents
$\mathcal{I} = \{1,\ldots,i,\ldots,M\}$
Media-Server

Fig. 1. Detailed media-service resource negotiation diagram. Beginning at the block "Start $t = 0$" (at the upper right), the logic for the Media-Server and the Agents runs in parallel, with synchronization between the two wherever information exchange is required. The shaded functional blocks are further described in the corresponding labeled sections.

B. *Auction Mechanism for allocating resources*

In this paper, we consider a simple auction mechanism for the resource negotiation in each time slot $t \in \{0,1,2,\ldots\}$. This stage of the media-service resource negotiation is illustrated in Fig. 1 as the shaded functional block with the label “Section II”. Specifically, we deploy the Progressive Second Price (PSP) auction mechanism introduced by Lazar in [2]. The PSP mechanism is a generalization of the well-known Vickrey auction, which is used to allocate a *single non-divisible object* to one bidder [5].[3] The PSP, on the other hand, is used to divide variable-sized portions of a *divisible resource* among multiple bidders, making it more appropriate than the Vickrey auction [5] for allocating (infinitesimally divisible) bandwidth to multiple agents.

In the PSP mechanism, each agent submits a bid $b_i = (q_i, p_i) \in \mathcal{B}_i = [0,R) \times [0,\infty)$ [4], where $\mathcal{B}_i$ is a set of possible bid-actions. Each bid is a *quantity-price pair* (q_i, p_i), where q_i and p_i are the i th agent’s desired quantity of resources[5] (kb/s) and offered price per unit resource, respectively. A *bid profile* is a tuple of agent’s bids $\boldsymbol{b} = (b_1,\ldots,b_M) = (b_i, \boldsymbol{b}_{-i}) \in \mathcal{B}$, where $\boldsymbol{\mathcal{B}} = \prod_{i \in \mathcal{I}} \mathcal{B}_i$ and $\boldsymbol{b}_{-i} = (b_1,\ldots b_{i-1}, b_{i+1},\ldots,b_M) \in \boldsymbol{\mathcal{B}}_{-i}$ denotes the bids of the i th agent’s opponents. b_i is determined by the policy with which agent i plays the auction game. We discuss such policies later in Section IV. If an agent does not require resources, its bid becomes $b_i = \mathbf{0}$. Note that the agents in our game are actually the devices on which the users view their video streams. In other words, the device’s software generates bids on behalf of the user.

After each agent submits its bid b_i, the media-service provider performs the following two computations as part of the PSP auction’s *allocation rule*: (i) The resource assignment, and (ii) the payment computation based on the inconvenience a particular agent causes the other agents during the current time slot.

[3] In a Vickrey auction, the highest bidder wins the non-divisible resource, and must pay the price offered by the second highest bidder.

[4] For notational simplicity, we do not explicitly indicate the time-index in this section (i.e. we do not write b_i^t , c_i^t , etc.).

[5] To account for packet losses in the wireless channel, the agent should bid for the *throughput* that will achieve its desired *goodput*.

An allocation rule A maps a bid profile $\boldsymbol{b} \in \mathcal{B}$ to an allocation profile $\boldsymbol{a}(\boldsymbol{b}) \in \mathcal{B}$. The allocation to the ith user is denoted by the pair $a_i(\boldsymbol{b}) = (r_i(\boldsymbol{b}), \tau_i(\boldsymbol{b})) \in \mathcal{B}_i = [0,R) \times [0,\infty)$, where $r_i(\boldsymbol{b})$ is the resource allocation (kb/s) and $\tau_i(\boldsymbol{b})$ is the *total* tax paid for $r_i(\boldsymbol{b})$. The allocation rule is said to be *feasible* if it satisfies [2]:

$$\begin{aligned}&\sum_{i=1}^{M} r_i(\boldsymbol{b}) \leq R,\ \forall \boldsymbol{b} \in \boldsymbol{\mathcal{B}},\ \text{and}\\ &r_i(\boldsymbol{b}) \leq q_i,\ \forall i \in \mathcal{I}\\ &\tau_i(\boldsymbol{b}) \leq p_i q_i,\ \forall i \in \mathcal{I}\end{aligned} \tag{1}$$

The auctioneer (i.e. the media-service provider) allocates resources in order to maximize the total “social welfare,” i.e.

$$\boldsymbol{r}^{opt} = \arg\max_{\boldsymbol{r} \in [0,R)^M} \sum_{i=1}^{M} p_i r_i\ , \tag{2}$$

where $\boldsymbol{r}^{opt} = (r_1^{opt}(\boldsymbol{b}),\ldots,r_M^{opt}(\boldsymbol{b}))$ meets the feasibility conditions in (1). An allocation rule that allocates resources as in (2) was introduced by Lazar in [2]; we repeat it below for completeness.

Allocation Rule [2]**:** For a *unit* price $y \in [0,\infty)$, define

$$R_i(y, \boldsymbol{b}_{-i}) := \left[R - \sum_{k \in \mathcal{I}_{-i}, p_k \geq y} q_k \right]^+ , \tag{3}$$

where $[x]^+ = \max\{x, 0\}$. The PSP allocation $a_i(\boldsymbol{b}) = (r_i(\boldsymbol{b}), \tau_i(\boldsymbol{b}))$ to user i is defined as follows:

$$\begin{aligned}&r_i(\boldsymbol{b}) = \min\{q_i, R_i(p_i, \boldsymbol{b}_{-i})\},\\ &\tau_i(\boldsymbol{b}) = \sum_{k \in \mathcal{I}_{-i}} p_k \left[r_k(\boldsymbol{0}, \boldsymbol{b}_{-i}) - r_k(b_i, \boldsymbol{b}_{-i}) \right]\end{aligned}. \tag{4}$$

Equation (3) defines $R_i(y, \boldsymbol{b}_{-i})$, which represents the resources available to agent i if it bids at the unit price $p_i = y$. In (4), $r_i(\boldsymbol{b})$ denotes the resource allocation to agent i and the tax $\tau_i(\boldsymbol{b})$ represents the impact that agent i has on the users who are excluded by its presence. By construction, $\tau_i(\boldsymbol{b})$ is always non-negative.

C. Problem formulation for the agents

The value that agent i derives from a resource allocation r_i is denoted as $\theta_i(r_i)$. In our setting, θ_i is the video quality received by agent i; it is measured in terms of the peak-signal-to-noise ratio (PSNR in dB), which is a commonly used objective video quality metric[6]. To obtain an analytical expression for θ_i, we may adopt any video distortion-rate models in the literature. Importantly, any other resource valuation function could be deployed, as long as it is differentiable, monotonically non-decreasing with increased resources, and concave [2].

For a one stage auction game in which the agents submit the bid profile $\boldsymbol{b} = (b_i, \boldsymbol{b}_{-i})$, the utility[7] gained by agent i has the quasi-linear form

$$u_i(\boldsymbol{b}) = \theta_i(r_i(\boldsymbol{b})) - \tau_i(\boldsymbol{b}), \tag{5}$$

which is merely the value of the allocation less the cost to the agent. Note that $u_i(\boldsymbol{b}) \leq \theta_i(r_i(\boldsymbol{b}))$ because $\tau_i(\boldsymbol{b})$ is always non-negative.

At each stage of the auction game, the agents desire to maximize their own utilities. In other words, each agent i wants to determine the bid b_i^{opt} such that

$$b_i^{opt} = \arg\max_{b_i \in \mathcal{B}_i} u_i(b_i, \boldsymbol{b}_{-i}). \tag{6}$$

A solution to (6) (within a tolerance $\varepsilon > 0$) is given by Lazar in [2] (assuming that $\boldsymbol{b}_{-i}$ is known, and $p_i \neq p_k$, $\forall k \neq i$). We will discuss the intuition behind the solution later in Section IV.B.

In large-scale decentralized and competitive auction scenarios, (6) cannot be solved by each user because they do not necessarily know other user's valuations and they cannot determine a priori their opponents' bids $\boldsymbol{b}_{-i}$ (except under very specific circumstances, which we will describe in Section IV.B). Therefore, users must learn through repeated interaction to make

[6] We note that if users participating in the resource negotiation process request videos with different spatial and/or temporal resolutions, then the PSNR may not accurately reflect the relative differences in perceived video quality among the users. This can be remedied by performing separate auctions for groups of users who are streaming videos with the same spatial and temporal resolutions.

[7] Throughout this paper, we will use the terms "utility," "payoff," and "reward" interchangeably.

better bids, thereby increasing their own utilities. In the next subsection, we formally define the repeated negotiation procedure, which specifies how agents request media-service resources over time. Later, in Section IV, we describe how the agents learn to improve their bids over time.

D. Repeated media-service resource negotiation

In our setting, the media-service provider auctions the available resources per time slot to the users. The proposed media-service resource negotiation framework provides a way of describing the dynamic interactions among the agents.

Formally, we define the resource negotiation procedure at the edge of the CDN as a tuple $(\mathcal{I},\mathcal{B},\mathcal{U},\nu)$, where $\mathcal{I}$ is the set of M agents and $\mathcal{B}$ is the joint bid-action space as defined in Section II.B. $\mathcal{U}$ is a reward vector function defined as a mapping from the joint actions $\boldsymbol{b} \in \mathcal{B}$ to an M-dimensional real vector representing the rewards for the various tasks, i.e. $\mathcal{U}:\mathcal{B} \mapsto \mathbb{R}^M$. Lastly, $\nu : \mathcal{I} \times \{0,1,2,...\} \mapsto \{0,1\}$ is a *centralized coordination policy*. The media-service provider uses ν to determine which agents may submit *new* bids in each time slot t. Specifically, ν maps an agent index $i \in \mathcal{I}$ and a time slot $t \in \{0,1,2,...\}$ to a binary variable:

$$\nu(i,t) = \begin{cases} 1, \text{ if player } i \text{ can submit a new bid at time } t \\ 0, \text{ if player } i \text{ cannot submit a new bid at time } t \end{cases} \tag{7}$$

Importantly, if $\nu(i,t) = 0$, then the ith agent's bid is the same as in the previous time slot, as illustrated in Fig. 1. We also assume that $\nu(i,0) = 1$, $\forall i \in \mathcal{I}$, regardless of the deployed coordination policy.

In the game theory literature, a "repeated game" is a game with a coordination policy ν in which the agents simultaneously submit their actions at every time slot [7], i.e. $\nu(i,t) = 1$, $\forall (i,t) \in \mathcal{I} \times \{0,1,2,...\}$. In this paper, however, we want to investigate how different levels of coordination can impact each agent's ability to learn to improve its utility over time.

Accordingly, we generalize the repeated game concept to consider different levels of centralized coordination. In Section III, we describe several coordination policies in detail.

The action taken by task i during each time slot t (in which $\nu(i,t) = 1$) is to submit the bid vector $b_i^t = (q_i^t, p_i^t)$. We define the history of the game up to time slot t as

$$\mathcal{H}^t = \{\boldsymbol{b}^0, \boldsymbol{a}^0, \boldsymbol{u}^0, ..., \boldsymbol{b}^{t-1}, \boldsymbol{a}^{t-1}, \boldsymbol{u}^{t-1}\}, \tag{8}$$

where $\boldsymbol{b} = (b_1, ..., b_M) = [(q_1, p_1), ..., (q_M, p_M)]$ is the bid profile, $\boldsymbol{a} = (a_1, ..., a_M) = [(r_1, \tau_1), ..., (r_M, \tau_M)]$ is the allocation profile, and $\boldsymbol{u} = (u_1, ..., u_M)$ is the utility profile. This history summarizes the bids played, the resulting allocations, and the pay-offs received by each agent up to time slot t. During the repeated game, however, each agent i may not be able to observe the entire history $\mathcal{H}^t$, but instead may only observe a subset $\mathcal{O}_i^t \subseteq \mathcal{H}^t$. There are several reasons for this. First, the information exchange overhead may be too great to distribute all the information to every agent in each time slot; second, an agent's memory limitations may make it impossible for the agent to maintain all of the history; and, third, an agent may not have the computational capacity to use all of the information to their benefit, thereby making it useless to the agent.

We note that the observed history may be built up from time slot to time slot based on metadata information sent from the media-server to the agents. As illustrated in Fig. 1, when agent i submits a bid in time slot t, it also submits a corresponding *feedback request* metadata unit denoted by $o(i,t)$. This request depends on the deployed learning rule, and prompts the media-server to provide the agent with the corresponding information at the beginning of the following time slot. Subsequently, the agent updates its observed history $\mathcal{O}_i^t$ with $o(i,t)$.

Finally, we define a decision rule $\pi_i : \mathcal{O}_i^t \mapsto \mathcal{B}_i$ for agent i as a mapping from its observed history into a specific bid, i.e.

$$b_i^t = \pi_i(\mathcal{O}_i^t)\,. \tag{9}$$

We describe learning rules and decision rules further in Section IV.

III. Coordination Policies

Recall that $\nu : \mathcal{I} \times \{0,1,2,...\} \mapsto \{0,1\}$ is a coordination policy, which determines when different agents may submit *new* bids in each time slot $t \in \{0,1,2,\ldots\}$. In the following subsections, we describe three simple coordination policies. We note that the policies described in this subsection are only a subset of an infinite set of possible policies. Nevertheless, we have selected policies that reasonably span the possible levels of coordination, from low to high. This stage of the media-service resource negotiation is illustrated in Fig. 1 as the shaded functional block with the label “Section III”.

A. Repeated Game Coordination (RG)

As we mentioned in Section II.D, the RG policy $\nu^{(RG)}$ requires that agents simultaneously submit their bids at each time slot, i.e. $\nu^{(RG)}(i,t) = 1$, $\forall (i,t) \in \mathcal{I} \times \{0,1,2,...\}$. This policy imposes a low level of coordination because *every* agent can update its bid at the smallest granularity, i.e. in every time slot. The consequence of this, however, is that individual agents may not be able to adapt to maximize their utility because all of their opponents are always changing their bids, thereby making it difficult to accurately predict the optimal bid in each time slot. In the next subsection, we propose a coordination policy that addresses this problem.

B. Random-N Polling (RNP)

The RNP policy $\nu^{(RNP)}$ randomly and uniformly selects $N \le M$ agents that can submit new bids in each time slot, i.e. this policy ensures that $\sum_{k \in \mathcal{I}} \nu^{(RNP)}(k,t) = N$, $\forall t \in \{1,2,\ldots\}$. Under this policy, the other $M - N$ agents (i.e. agents $k \in \mathcal{I}$ for which $\nu^{(RNP)}(k,t) = 0$) do not submit a new bid in time slot t; instead, they repeat their bids from the previous time slot, i.e. $b_k^t = b_k^{t-1}$. If $N = M$, then the RNP policy is equivalent to RG coordination policy described in the previous subsection. Clearly, the RNP policy has a level of coordination that scales with N. In particular, higher values of N indicate lower levels of coordination. Importantly, $\nu^{(RNP)}(i,0) = 1,\ \forall i \in \mathcal{I}$, which is a requirement of all coordination policies.

The value of N defined by the coordination policy, in conjunction with the number of agents M, determine the probability with which an individual agent submits a new bid in time slot t. This *bid-submission frequency*, denoted by μ, is determined as,

$$\mu = \frac{N}{M}. \tag{10}$$

Note that μ is the same for every agent. As we will show in the experiments in Section V.B, the repetition of previous bids benefits agents. For example, if some of an agent's opponents must repeat their previous bids, then the agent's new bid is less susceptible to unexpected decisions by its opponents, and is therefore more likely to improve its utility in the current time slot. Clearly, there exists a tradeoff between bid-submission frequency and agents improving their utility. In particular, an individual agent wants to submit bids in every time slot in order to improve its utility, however, the more agents that submit a bid in a time slot, the less improvement any individual agent can make due to uncertainty in its opponents' bids. We explore this tradeoff in our experiments in Section V by using different values of N, which impose different bid-submission frequencies μ.

Due to the randomness built into this policy, there will be cases in short intervals of time in which some agents are allowed to submit bids more frequently than other agents. To ensure fairness in bid-submission frequency, we propose the policy in the following subsection.

C. Round-Robin Polling (RRP)

As in the RNP policy when $N = 1$, the RRP policy $\nu^{(RRP)}$ ensures that $\sum_{i \in \mathcal{I}} \nu^{(RRP)}(i,t) = 1$, $\forall t \in \{1, 2, \ldots\}$. The RRP policy, however, also requires that agents are polled in a (deterministic) round-robin fashion, i.e. $\nu^{(RRP)}(i, t + M) = \nu^{(RRP)}(i,t)$, $\forall t \in \{1, 2, \ldots\}$ and $\forall i \in \mathcal{I}$. Clearly, this policy imposes a high level of coordination. As with the other policies, $\nu^{(RRP)}(i, 0) = 1$, $\forall i \in \mathcal{I}$.

Fig. 2 summarizes the relative levels of coordination imposed by the proposed coordination policies.

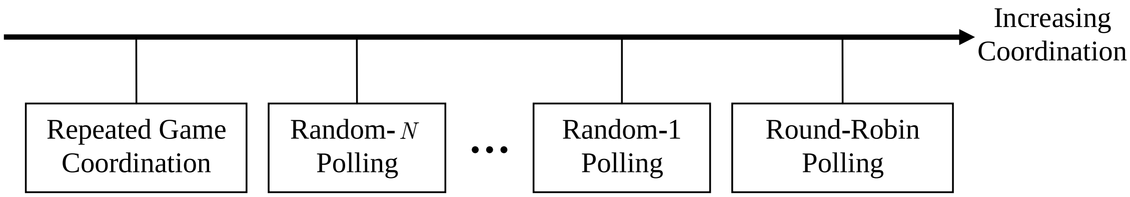

Fig. 2. Relative coordination levels for the proposed coordination policies.

IV. Learning to Bid

Since the utility derived by one task in each stage of the auction game depends on the bids of all other agents, an individual agent cannot in general solve the optimization problem in (6) aimed at maximizing its utility. As we mentioned before, the agents must learn to improve their bids based on their repeated interactions and their observed histories $\mathcal{O}_i^t$, $\forall i \in \mathcal{I}$. This stage of the media-service resource negotiation is illustrated in Fig. 1 as the shaded functional block with the label "Section IV".

We define a learning rule deployed by the ith agent as a tuple $L_i = (\mathcal{O}_i, \pi_i, \zeta_i, \nu)$, where $\mathcal{O}_i$ is the information required to implement the learning rule, π is the decision rule mapping the available information to a bid (i.e. $\pi_i : \mathcal{O}_i \mapsto \mathcal{B}_i$), and ζ_i is the computational burden associated with determining a bid using the decision rule π_i. ζ_i is measured in floating point operations (FLOPs). We also include the coordination policy ν in our definition of the learning rule because some rules may have to be paired with a particular coordination policy for them to perform well (e.g. the best-reply learning rule presented in Section IV.B).

In Section IV.A, we begin our discussion on learning by introducing metrics for evaluating the performance of different learning rules with different information, complexity, and coordination requirements. In Section IV.B-IV.D, we introduce three learning rules, which require varying levels of computation and information to determine the bid at each time slot.

A. Evaluating The Benefits of Learning

Our goal in this paper is to investigate the performance of learning rules that span the three dimensional space of information, complexity, and coordination. In this subsection, we introduce performance metrics for comparing different learning rules within this parameter space. Importantly, these metrics are evaluated *outside* of the actual media-service resource negotiation process. In other words, they are not used by the media-server or the agents to make decisions during the negotiation process. Rather, they are useful tools that enable a system designer to quantitatively analyze different configurations of the resource negotiation system (e.g. to investigate the tradeoffs between different coordination policies and learning rules for a particular media-service application).

A measure of a learning rule's performance in time slot t of the auction game is the utility that agent i receives for deploying it, i.e.

$$u_i^t(\pi_i(\boldsymbol{\mathcal{O}}_i^t),\boldsymbol{b}_{-i}^t) = \theta_i^t(r_i^t(\pi_i(\boldsymbol{\mathcal{O}}_i^t),\boldsymbol{b}_{-i}^t)) - \tau_i^t(\pi_i(\boldsymbol{\mathcal{O}}_i^t),\boldsymbol{b}_{-i}), \tag{11}$$

where $b_i^t = \pi_i(\boldsymbol{\mathcal{O}}_i^t)$ is the bid generated by the ith agent's decision rule in time slot t. Note that (11) is equivalent to (5). Based on this, a simple measure of the performance loss associated with a learning rule L_i in time slot t can be defined as the difference between the optimal achievable utility and the actual utility achieved, i.e.

$$V_i^t(L_i) := u_i^t(b_i^{opt}, \boldsymbol{b}_{-i}^t) - u_i^t(\pi_i(\boldsymbol{\mathcal{O}}_i^t), \boldsymbol{b}_{-i}^t)\,, \tag{12}$$

where b_i^{opt} is the solution to problem (6). The metric in (12) is inspired by theory for the "value of information" [9], which measures the change in the optimized performance index in control systems when certain information is known, relative to when it is not known. In our formulation, a lower value of (12) indicates better performance.

Since the auction game is played repeatedly over time slots $t \in \{0,1,2,\ldots\}$, the performance metric in (12) may not be representative of the long-term performance of one learning rule compared to the optimal achievable utility. To account for this, we extend (12) into a cumulative performance loss metric

$$J_i^t(L_i) := \sum_{s=0}^{t} V_i^s(L_i)\,, \tag{13}$$

which is the cumulative performance loss from time 0 through time t. This metric measures the absolute performance of a learning rule without regard for the implementation complexity.

In order to compare the computational overheads of different learning rules, we introduce the cumulative-utility-to-complexity ratio

$$G_i^t(L_i) = \left(\frac{\sum_{s=0}^{t} u_i^s(\pi_i(\boldsymbol{\mathcal{O}}_i^s), \boldsymbol{b}_{-i}^s)}{\sum_{s=0}^{t} \nu(i,s) \cdot \zeta_i^s} \right), \tag{14}$$

where ζ_i^t is the number of FLOPs required to compute $b_i^t = \pi_i(\boldsymbol{\mathcal{O}}_i)$ a single time, and the summation in the denominator is the total number of FLOPs performed by the agent from time 0 through time t. Hence, G_i^t is the average utility gained per unit of computational overhead. In this way, the cumulative-utility-to-complexity ratio represents the efficiency of the learning rule. Clearly, since (14) is a ratio, it can be high or low regardless of (13) being high or low. By considering both (13) and (14), however, we are able to get a complete picture of a learning rule in terms of both efficiency and absolute performance.

B. Best-reply Learning

In this subsection, we assume that the ith agent knows its opponents' bid profile $\boldsymbol{b}_{-i}^{t}$ in time slot t. Given this information, we aim to determine the ith agent's bid that maximizes its utility (i.e. solves problem (6)). It is important to note that in a decentralized and competitive scenario like the auction game, it is usually unreasonable to assume that agent i knows $\boldsymbol{b}_{-i}^{t}$ before it makes its bid b_i^t. This requirement can be met, however, under two strict conditions. First, the observed history must contain $\boldsymbol{b}_{-i}^{t-1}$ (i.e. $\boldsymbol{b}_{-i}^{t-1} \in \mathcal{O}_i^t$), therefore the feedback request will be $o(i,t-1) = \boldsymbol{b}_{-i}^{t-1}$. Second, the coordination policy must allow only agent i to submit a new bid in time slot t. Together, these two conditions dictate that $\sum_{k \in \mathcal{I}} \nu(k,t) = 1$, $\nu(i,t) = 1$, and $\boldsymbol{b}_{-i}^{t} = \boldsymbol{b}_{-i}^{t-1}$. Hence, the best-reply learning policy can be implemented if either the RRP or the RNP coordination policy (with $N = 1$) is deployed.

It can been shown that if its opponents' bid profile $\boldsymbol{b}_{-i}^{t}$ is known, then the ith agent's best-reply bid $b_i^{t,opt} = (q_i^{t,opt}, p_i^{t,opt}) = \pi_i^{opt}(\mathcal{O}_i^t)$ (i.e. the solution to (6)) has the optimal bid price

$$p_i^{t,opt} = \theta_i'(q_i^{t,opt}), \tag{15}$$

where $\theta_i'(q) = \dfrac{d}{dq}\theta_i(q)$ is the agent's *marginal valuation* of the quantity q [2]. The optimal bid quantity $q_i^{t,opt}$ can be determined as described in [2].

Not only does this learning rule require significant processing (as will be illustrated in Section V in Table II), but it also requires an agent to receive complete information from the media-server about its opponents, i.e. the bid profile $\boldsymbol{b}_{-i}$. To reduce the communication and complexity overheads, the opponents' bids can be clustered as described in the next subsection.

C. Clustered Best reply Learning

Thus far, we have assumed that the learning process is highly coordinated (i.e. one agent bids in each time slot) and that the agents know their opponents' bid profile a priori. In this subsection, however, we consider a more general scenario in which agents do not precisely know their opponents' bid profile, and any coordination policy can be deployed.

As we mentioned in the previous subsection, agents may not precisely know their opponents' bids, or may not be able to process all of the information about their bids, due to excessive communication or computational overheads. Since users' devices have different inherent constraints, some agents will be more capable of accurately modeling their opponents than others. To accommodate this, the media-server must be able to provide different levels of feedback to every agent, which they can use to improve their bids over time. To this end, we assume that agent i can query the media-server for coarser, or more detailed, information about its opponents through its feedback request metadata information $o(i,t)$. Specifically, the ith agent will request that the media-server clusters the agent's opponents into $H_i^t \in \{2,3,\ldots,H_i^{MAX}\}$ mutually exclusive and collectively exhaustive subsets of $\mathcal{I}_{-i}$. Here, $H_i^{MAX} \le |\mathcal{B}_{-i}|$ is an upper bound on the number of clusters, which depends on the computational capabilities of the user's device or the tolerable communication overheads (i.e. larger values of H_i^{MAX} require higher computational capacity and more information exchange between the media-server and the agent).

We denote the ith agent's clustered opponents as $\mathcal{I}_{-i(h)}^t \subseteq \mathcal{I}_{-i}$, $1 \le h \le H_i^t$. At the coarsest level, the media-server will cluster an agent's opponents into two mutually exclusive and collectively exhaustive subsets of $\mathcal{I}_{-i}$ during each time slot t. We design these coarse opponent clusters to match the form of the PSP auction in (4). Specifically, we define

$$\underline{\mathcal{I}}_{-i}^{t} \coloneqq \{k \in \mathcal{I}_{-i} \mid p_k^t < p_i^t\} \subseteq \mathcal{I}_{-i} \tag{16}$$

and

$$\bar{\mathcal{I}}_{-i}^{t} \coloneqq \{k \in \mathcal{I}_{-i} \mid p_k^t \geq p_i^t\} \subseteq \mathcal{I}_{-i} \,. \tag{17}$$

$\underline{\mathcal{I}}_{-i}^{t}$ and $\bar{\mathcal{I}}_{-i}^{t}$ are sets of opponents whose bid prices are lower and higher than the ith agent's bid price in time slot t, respectively. These two coarse clusters may also be written as $\mathcal{I}_{-i(1)}^{t} = \underline{\mathcal{I}}_{-i}^{t}$ and $\mathcal{I}_{-i(2)}^{t} = \bar{\mathcal{I}}_{-i}^{t}$ (with $H_i^t = 2$), however, we will use the under- and over-bar notation to emphasize the form of these clusters. Importantly, the opponents in both of these sets impact the ith agent's utility. This is because, based on the PSP auction in (4), the opponents in $\underline{\mathcal{I}}_{-i}^{t}$ determine the tax that agent i must pay and the opponents in $\bar{\mathcal{I}}_{-i}^{t}$ determine its resource allocation.

If the agent requests $H_i^t > 2$ clusters, then the subsets of $\mathcal{I}_{-i}$ must be constructed to satisfy the following two conditions: first, agents in cluster $\mathcal{I}_{-i(h)}^{t}$ offer a lower bid-price than agents in cluster $\mathcal{I}_{-i(h+1)}^{t}$; second, cluster $\mathcal{I}_{-i(h)}^{t}$ is either a subset of $\underline{\mathcal{I}}_{-i}^{t}$ or a subset of $\bar{\mathcal{I}}_{-i}^{t}$. In this way, the agents' opponents are still divided into clusters that impact its resource allocation or its tax.

Instead of considering each individual opponent as in subsection IV.B, agent i sees each cluster as an opponent. Accordingly, each cluster $\mathcal{I}_{-i(h)}^{t} \subseteq \mathcal{I}_{-i}$, $1 \leq h \leq H_i^t$, is associated with a single bid $b_{-i(h)}^t = (q_{-i(h)}^t, p_{-i(h)}^t)$, where the subscript $-i(h)$ indicates cluster h of the ith agent's opponents. The bid quantity associated with cluster $\mathcal{I}_{-i(h)}^{t}$ is defined as

$$q_{i(h)}^t \coloneqq \sum_{k \in \mathcal{I}_{-i(h)}^{t}} q_k^t \,, \tag{18}$$

and the associated bid price is defined as

$$p_{i(h)}^t \coloneqq \frac{1}{\left|\mathcal{I}_{-i(h)}^{t}\right|} \sum_{k \in \mathcal{I}_{-i(h)}^{t}} p_k^t \,. \tag{19}$$

When agent i clusters its opponents, its observed history becomes $\mathcal{O}_i^t = \{b_{-i(1)}^{t-1}, \ldots, b_{-i(H_i^{t-1})}^{t-1}\}$, which is a record of every clusters' bids in the previous time slot. It follows that agent i can determine its *clustered best reply* bid $\hat{b}_i^{t,opt}$ using (6) by treating each cluster as a single user. We use the hat notation to indicate that the clustered best reply is an approximation of the best reply $b_i^{t,opt}$.

In our experiments in Section V, we use the evaluation metrics introduced in Section IV.A to investigate the tradeoffs between computational complexity and achieved utility when an agent clusters its opponents at different granularities, or does not cluster its opponents at all. Additionally, in Section V in Table II, we illustrate the number of FLOPs required for a player to calculate its best-reply bid against different numbers of opponents, using different sized clusters.

D. A greedy learning solution

In the previous subsections, we have discussed learning solutions that require an agent to collect information about its opponents' bids. In this subsection, we use a greedy solution that does not require an agent to observe its opponents' bids. This greedy solution serves as a performance baseline. We expect that the learning algorithms that make bid decisions based on more information will perform better than this naïve solution. We note that any coordination policy can be deployed if agents implement this learning solution.

Recall that each user $i \in \mathcal{I}$ has valuation θ_i that is differentiable, monotonically non-decreasing with increased resources, and concave. The form of the users' valuations naturally leads to a greedy learning solution in the repeated auction game. The essence of this solution is that users will start by asking for some maximum quantity of resources at their marginal valuation, which at their maximum quantity is at its minimum. After observing their own allocations and payoffs, each user will decrease their requested quantity and increase their offered price so that they can achieve a higher utility. This solution does not require the agents to observe their opponents bids, therefore the agents' feedback request will be $o(i,t) = \varnothing$. Accordingly, the ith agent's observed history at time slot t only includes its bid $b_i^{t-1} = (q_i^{t-1}, p_i^{t-1})$, allocation $a_i^{t-1} = \left(r_i^{t-1}, \tau_i^{t-1}\right)$, and valuation θ_i^{t-1} from time slot $t-1$, i.e.

$$\mathcal{O}_i^t = \{b_i^{t-1}, a_i^{t-1}, \theta_i^{t-1}(r_i^{t-1})\}.$$

The following steps comprise the greedy learning algorithm:

Greedy Learning Algorithm

1. **Initial Bid Request:** At time slot $t = 0$, each user $i \in \mathcal{I}$ submits its initial bid request $b_i^0 = (q_i^0, p_i^0)$. At this initial stage, each user requests a maximum quantity of resources q_i^{MAX} for a minimum price $p_i^{MIN} = \theta_i'(q_i^{MAX})$. User i then receives its first allocation $a_i^0 = \left(r_i^0, \tau_i^0\right)$.

2. **Bid Quantity and Bid Price Update:** At time slot $t \in \{1, 2, \ldots\}$ each user updates its bid based on its observed history $\mathcal{O}_i^t$. In particular:

   a. If $r_i^{t-1} < q_i^{t-1}$, then $q_i^t = q_i^{t-1} - \Delta q_i^{t-1}$ and $p_i^t = \theta_i'(q_i^{t-1} - \Delta q_i^{t-1})$. Here, $\Delta q_i^{t-1} > 0$ is a (possibly time-varying) step-size for the bid quantity.

   b. Otherwise, if $r_i^{t-1} = q_i^{t-1}$, then $q_i^t = q_i^{t-1}$ and $p_i^t = p_i^{t-1}$ with probability γ_i and $q_i^t = q_i^{t-1} + \Delta q_i^{t-1}$ and $p_i^t = \theta_i'(q_i^{t-1} + \Delta q_i^{t-1})$ with probability $1 - \gamma_i$. In other words, if agent i receives its desired resources in time slot $t-1$, then it prefers to bid the same quantity in time slot t with probability γ_i. With probability $1 - \gamma_i$, however, agent i would like to try to increase its resource allocation in case extra resources become available (e.g. due to other users leaving the auction game or asking for less resources).

3. **Repeat:** Go to step 2.

A time-varying bid-quantity step size can be set such that each step results in an approximately equivalent PSNR drop (or increase). To do this, we simply let $\Delta q_i^t = \Delta\theta_i \,/\, \theta_i'(q_i^{t-1})$, where $\Delta\theta_i$ is the constant PSNR step size. In this way, the agent will quickly back-off its bid-quantity when its marginal valuation is small, but will slowly change its bid-quantity when its marginal valuation is large.

Fig. 3 summarizes the relative information requirements and implementation complexity levels for the proposed learning rules and Table I summarizes their properties and requirements. In Section V in Table II, we illustrate the number of floating-point operations (FLOPs) required for a player to calculate its greedy bid against different numbers of opponents.

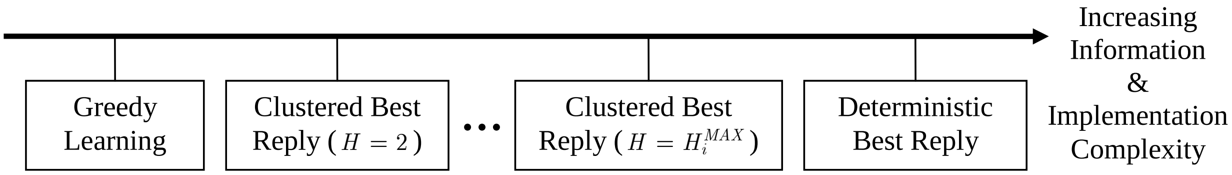
Increasing Information & Implementation Complexity
Greedy Learning
Clustered Best Reply ($H = 2$)
…
Clustered Best Reply ($H = H_i^{MAX}$)
Deterministic Best Reply

Fig. 3. Relative information requirements and implementation complexity levels for the proposed learning rules.

TABLE I. SUMMARY OF THE PROPOSED LEARNING RULES.

Learning Solution	**Required Information**	**Complexity**	**Coordination Policy**	**Decision Rule**
Best Reply	$o(i,t)=\{\boldsymbol{b}_{-i}^{t-1}\} \subseteq \boldsymbol{\mathcal{O}}_i^t$	High	Random-1 Polling Round-Robin Polling	Eq. (6)
Clustered Best Reply	$o(i,t)=\{b_{-i(1)}^{t-1},\ldots,b_{-i(H_i^{t-1})}^{t-1}\} \subseteq \boldsymbol{\mathcal{O}}_i^t$	Low to High	Any	Eq. (6) (with clustered opponents)
Greedy	$o(i,t)=\varnothing$, $\{b_i^{t-1}, a_i^{t-1}, \theta_i^{t-1}(r_i^{t-1})\} \subseteq \boldsymbol{\mathcal{O}}_i^t$	Low	Any	Greedy learning algorithm

V. EXPERIMENTS

A. *Learning rule properties*

To better understand the properties of the learning rules, we consider two scenarios in which $M = 25$ agents negotiate for media-service resources at the edge of the CDN. In both scenarios, we assume that agent $i = 5$ deploys one of the learning rules introduced in Section IV. We are interested in how the 5th agent's performance is impacted by its choice of learning rule. We assume that the round-robin coordination policy from Section III.C is employed by the media-server. Therefore, agent $i = 5$ is polled to submit one bid every M time slots.

1) Scenario 1: Two users stop and resume streaming session

In the first scenario, we assume that the 5th agent's opponents $\mathcal{I}_{-5}$ maintain a constant bid-profile throughout the duration of the streaming session, except for during time slots $t \in \{150,151,\ldots,399\}$ and $t \in \{200,201,\ldots,399\}$ when agents 4 and 3, respectively, submit no bids at all. This simulates the effect of agents stopping their streams, which increases the wireless resources that are available to the other users, and correspondingly decreases the cost of the resources. In response to these changes, the 5th agent can adapt its bid (i.e. increase its bid-quantity and decrease its bid-price) in order to increase its resource allocation and, consequently, its utility. At $t = 400$, agents 3 and 4 resume their video streams by submitting the same constant bids as in earlier time slots. This increases the network congestion, thereby raising the price to the users and forcing the 5th agent to adapt its bid (i.e. decrease its bid-quantity and correspondingly increase its bid-price) in order to maximize its utility given the current level of congestion.

Table II illustrates the average number of FLOPs required for agent $i = 5$ to determine its bid using different learning rules against different numbers of opponents. The data in this table confirms our intuition about the relative complexity of the learning rules in Section IV; clearly, the greedy learning algorithm is the least complex and the best reply is the most complex. The greedy learning algorithm and the clustered best-reply learning have complexity that is approximately invariant with the number of users. Additionally, the implementation complexity of the clustered best-reply increases approximately linearly with the number of clusters. A consequence of these predictable complexity levels is that an agent could dynamically select different learning algorithms depending on its instantaneously available computational resources and delay tolerance. We note that some of the entries in Table II are empty because the number of clusters exceeds the number of opponents with non-zero bids during some of the time slots (for example, when two users are inactive in the 5 user scenario, there are not enough opponents with non-zero bids to divide them into $H = 4$ clusters).

TABLE II. LEARNING RULE IMPLEMENTATION COMPLEXITY IN FLOPS.

Number of Users	Learning Rule (L)				
	Greedy	Cluster (H = 2)	Cluster (H=4)	Cluster (H=8)	Best Reply
5	3.80	37.69	-	-	57.83
10	4.66	38.14	76.14	-	154.04
25	3.85	38.35	76.35	152.35	439.25

Fig. 4 illustrates the 5th agent's utility over the course of the streaming session described above. We note that the PSNR (in dB) is at least as large as the utility shown in the figure (because the utility is the PSNR less the non-negative cost). Fig. 4(b) illustrates the utility achieved when the agent deploys the best-reply and the greedy learning algorithms. The "optimal" utility shown in the figure is the solution to (6). We note that, since all of its opponents have constant bids, agent $i = 5$ almost always achieves the optimal utility when deploying the best-reply learning algorithm. There are only a few occasions when it does not achieve the optimal utility: first, when it has no information about its opponents (i.e. in time slot $t = 0$); and, second, during time slots after other agents enter or leave the streaming session, but before agent $i = 5$ is polled to submit a new bid (e.g. $t = 151$ and $t = 201$).

Fig. 4(b) also illustrates the utility achieved by the agent when it deploys the greedy learning algorithm. As expected, this algorithm is slow to adapt to changes because it does not consider any feedback about the other agents. No matter how quickly resources become available, the greedy algorithm's adaptation rate is limited by the quantity step-size defined in Section IV.D. If this step size is set too large, then the agent's utility will oscillate, however, if it is set too small (as it is here), then the utility may be far from optimal.

Fig. 4(a) compares the utility achieved when agent $i = 5$ deploys the clustered best-reply algorithm with $H_5 = 2$ and $H_5 = 8$. Surprisingly, even when 24 opponents are clustered into two groups, this learning algorithm performs almost as well as the best-reply algorithm without clustering. When the number of clusters is increased to eight (incurring a computational burden that is four times greater than with two clusters according to Table II), the utility is increased marginally. We note that the clustered best-reply performs nearly as well as the best-reply algorithm even though it is less complex. It does this by exchanging instantaneous implementation complexity for time-complexity. In other words, by considering opponent clusters, the agent saves in computations per time slot, but requires more time slots to achieve the optimal utility.

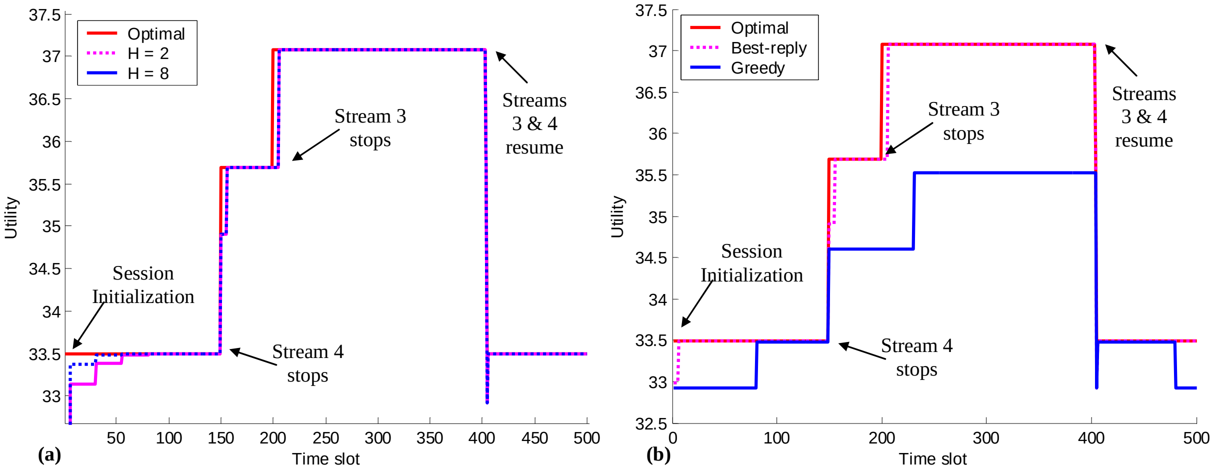

Fig. 4. Utility over time for stream 5 under various learning rules (500 time slots): (a) Clustered best-reply learning with $H_5 = 2$ and $H_5 = 8$, and (b) best-reply and greedy learning. At times $t = 150$ and $t = 200$ video stream 4 and video stream 3 stop, respectively. At time $t = 400$ video streams 3 and 4 resume.

Now that we have discussed the constituent components of the evaluation metrics proposed in IV.A (i.e. complexity in FLOPs and utility), we are in a position to better understand the performance tradeoffs associated with the various learning rules. Table III provides the 5th agent's cumulative performance loss metric $J_5^t(L_5)$ (see (13)) and cumulative-utility-to-complexity ratio metric $G_5^t(L_5)$ (see (14)) at times $t \in \{149, 199, 399, 499\}$. The first three times are chosen because they occur just before agents enter or leave the streaming session. The final time is chosen to capture the average results over the entire streaming session.

TABLE III. LEARNING RULES EVALUATION

Time Slot (t)	Evaluation Metric	Learning Rule (L)				
		Greedy	Cluster (H = 2)	Cluster (H=4)	Cluster (H=8)	Best Reply
149	$J^t(L)$	47.53	70.86	65.56	62.16	2.57
	$G^t(L)$	141.24	20.93	10.63	5.36	1.81
199	$J^t(L)$	102.21	75.56	70.26	66.86	7.27
	$G^t(L)$	180.35	21.54	10.90	5.48	1.87
399	$J^t(L)$	440.79	83.88	78.57	75.18	15.59
	$G^t(L)$	264.42	22.9	11.53	5.78	2.03
499	$J^t(L)$	460.07	84.46	79.15	75.76	16.17
	$G^t(L)$	244.24	22.78	11.44	5.73	1.99

It is clear from Table III that only the greedy learning algorithm is more efficient (i.e. has a greater cumulative-utility-to-complexity ratio) than the clustered best-reply learning algorithm with $H = 2$. The cumulative performance of the clustered case, however, is much better (recall that lower values of $G^t(L)$ indicate a smaller deviation from the optimal achievable utility). Comparing this clustered case to the best-reply, it is clear that the latter is very inefficient. We can also observe that with only two clusters, we only lose approximately 19 utility points more than the best-reply algorithm over the 500 time slots. This result indicates that dividing 24 opponents into two clusters provides a very good tradeoff between complexity and performance.

2) Scenario 2: Many users randomly stop and resume streaming session

Now, we consider a second scenario in which more of the $M = 25$ users stop and resume sessions over a 1000 time slot simulation (i.e. $t \in \{1, 2, \ldots, 1000\}$). In this scenario, we assume that the length of each video streaming session is geometrically distributed with a mean of 1000 time steps. Once a session ends, we assume that the time until the session resumes is geometrically distributed with a mean of 500 time steps. Fig. 5(a) illustrates the number of active users in each time slot. Fig. 5(b) and (c) show the utility achieved in each time slot by the 5th user when it deploys various learning strategies. We note that the "no learning" strategy in Fig. 5(b) is a strategy where the 5th user submits the same bid in every time slot.

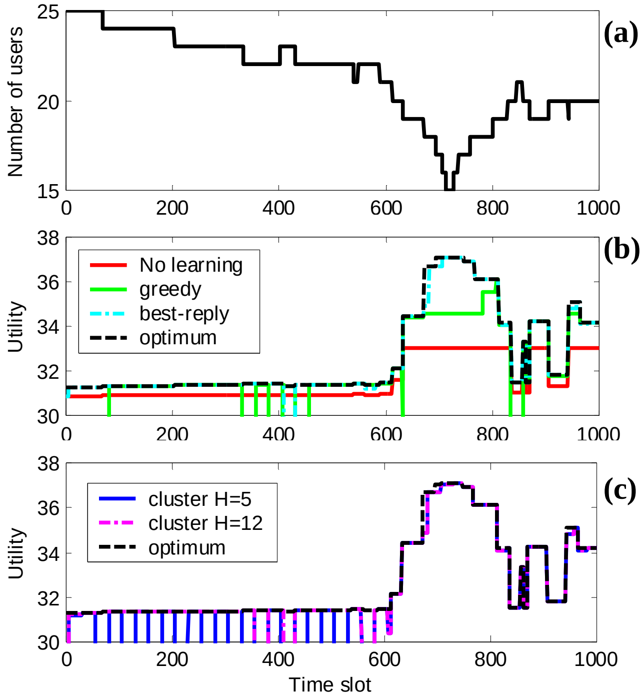

Fig. 5. Utility over time for stream 5 under various learning rules (1000 time slots): (a) Number of active users. (b) No learning (i.e. constant bids), greedy learning, and best-reply learning, (c) clustered best-reply learning with $H = 5$ and $H = 12$ clusters.

From Fig. 5, we observe that there is a lot of congestion in the first 600 time slots (i.e. more than 20 of 25 possible users are active). As a result, there are many time slots during which the 5th agent's utility is zero. Importantly, this does not mean that its video playback freezes. Instead, it means that its pre-decoding buffer will drain faster than it fills [10]. As long as this buffer is not empty, the agent will be able to continue playing back its streamed video [10]. In order to illustrate this, we show in Fig. 6 the number of kilobits received by the 5th agent over time when it deploys different learning policies and its corresponding average PSNR. We make the following observations:

- At time slot $t = 400$, the best-reply policy receives less kilobits than in the optimum case. This is because the best-reply learning algorithm is only guaranteed to be optimal if the agent's opponents maintain the same bid as they did in the previous time slot. As we can observe in Fig. 5(a), however, a new agent starts streaming video at this time slot and therefore the opponents' bid profile is not the same as it was in the previous time slot.
- At time slot $t = 600$, the clustered best-reply learning policy (with $H = 5$) starts to improve dramatically. This is because the number of active users rapidly declines after this time slot (see Fig. 5(a)), which allows the user to improve its bids despite only having coarse grained information about its opponents' past bids.

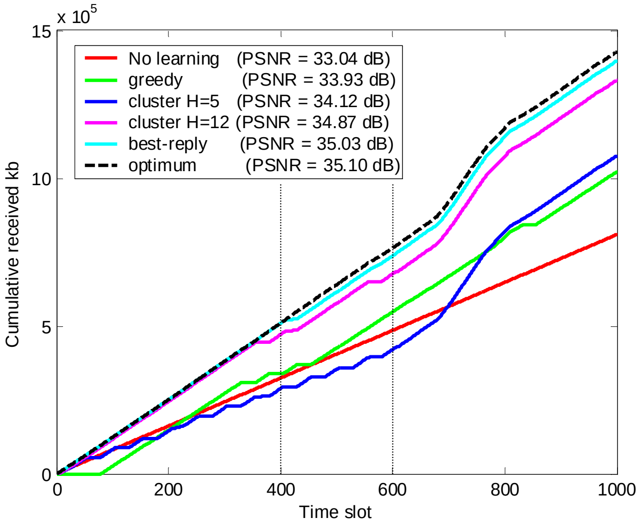

Fig. 6. Kilobits received for stream $i = 5$ over time for different learning policies. The average PSNR for each streaming session is shown in the legend.

3) Learning rule properties summary

If very few agents are learning, then the more complex learning algorithms, which take into account more information from the previous stage of bidding, perform better (in terms of the cumulative performance metric) than less complex algorithms, which take into account less information from the previous stage. In the next subsection, we will see that this is not the case in very dynamic settings, where many agents are learning.

B. Coordination policy comparison

In this subsection, we investigate the impact of the coordination policy on the 5th agent's utility when it is negotiating for media-service resources against heterogeneous opponents in two scenarios, which are distinct from the two scenarios in the previous subsection. We first describe the two scenarios and then make our observations below.

1) Scenario 1: Coordinating 10 heterogeneous users

In the first scenario, we let there be a total of $M = 10$ agents. Three of the 5th agent's opponents deploy the greedy learning algorithm; two deploy the best-reply algorithm; and, four deploy the clustered best-response algorithm (with $H = 2, 4, 6 \text{ and } 7$ clusters).

In Fig. 7 (top left), we compare the 5th agent's cumulative performance loss metric evaluated at $t = 499$ (i.e. $J_5^{499}(L_5)$) when it deploys the best-reply, greedy, and clustered best-reply ($H_5 = 2$) learning algorithms under five different coordination policies introduced in Section IV. Specifically, we consider the Round-robin, Random-2, Random-5, Random-8, and Repeated coordination policies. Fig. 7 (bottom left) illustrates the 5th agent's cumulative-utility-to-complexity ratio evaluated at $t = 499$ (i.e. $G_5^{499}(L_5)$) under the same settings.

2) Scenario 2: Coordinating 100 heterogeneous users

In the second scenario, we assume that the 5th agent's 99 opponents deploy the following learning policies: sixteen users deploy the greedy learning policy, eight users deploy the best-reply learning policy, twenty-three users deploy the clustered learning policy with $H = 5$, twelve users deploy the clustered learning policy with $H = 12$, twenty users deploy the clustered learning policy with H = 19, and twenty users deploy constant bids. Because there are many more users than in the previous scenario, agents that deploy the clustered learning policy must consider more clusters to achieve good performance and the coordination policy must poll more users at a time in order to maintain a reasonable bid-frequency for each user.

3) Observations

The top two plots in Fig. 7 (especially the top left plot) illustrate that the best-reply learning algorithm's performance and, to a lesser extent, the clustered best-reply's performance depends on the choice of coordination policy. In particular, the algorithm's performance depends on the agent's bid-frequency μ and the number of agents that simultaneously submit bids, which are both determined by the coordination policy.

On the one hand, if many agents bid simultaneously (e.g. Repeated coordination or Random-N polling with large N), then a user deploying the best-reply learning algorithm will not significantly improve its performance because its opponents' bids vary too frequently and too unpredictably for it to reasonably predict its optimal bid in each time slot; on the other hand, if agents bid too infrequently (e.g. Round-robin polling or Random-N polling with small N), then the user deploying the best-reply learning algorithm will be quickly outbid by the other users, and its performance will degrade. Hence, it is important for a coordination policy to be in place that strikes a good balance between the bid-submission frequency and the number of agents that simultaneously submit bids. From Fig. 7 (top left) and Fig. 7 (top right), we observe that the Random-2 and Random-19 coordination policies, respectively, yield the best performance for the 5th agent when it uses the best-reply learning algorithm (i.e. the minimum cumulative performance loss). These policies both have an expected bid-submission frequency of approximately $\bar{\mu} = 1/5$.

In both the 10 and 100 user scenarios, the best-reply learning algorithm performs particularly poorly when the Round-robin polling policy is used. This is precisely because of the low bid-submission frequency. To see this, recall that the best-reply learning algorithm is optimal in the current stage if the opponents' bids stay the same as they were in the previous stage. Hence, when the agent submits a new bid under the Round-robin policy, that bid is optimal. However, the other agents each get to outbid the agent before it gets the opportunity to submit another bid, which severely degrades its overall performance.

Fig. 7 (top left) shows that the greedy learning algorithm performs the best against heterogeneous opponents (as indicated by the smallest cumulative performance loss values). Although this algorithm is at a disadvantage when the opponents' bids never change (as in the previous subsection), it performs better when the opponents' bids change frequently. This is because the greedy algorithm's bid decision is less sensitive to the agent's opponents' previous bids, which may not accurately predict their current bids.

An interesting result in Fig. 7 is that the more complex learning algorithms, which take into account more information from the previous stage of bidding (e.g. best-reply and random-N with large N), actually perform worse in very dynamic settings than less complex algorithms (e.g. greedy and random-N with small N), which take into account less information from the previous stage. As we have noted, this is because the opponents' bids are unpredictable in very dynamic settings, thereby making it impossible for the 5th agent to reasonably predict its optimal bid.

Fig. 7 (bottom left) and Fig. 7 (bottom right) illustrate the cumulative-utility-to-complexity ratio in the two scenarios. We can see that, not only does the 5th agent's utility decrease as the bid-submission frequency increases, but its gain per unit of computation decreases dramatically. In other words, agent 5 expends significantly more processing time (and energy) for a worse utility.

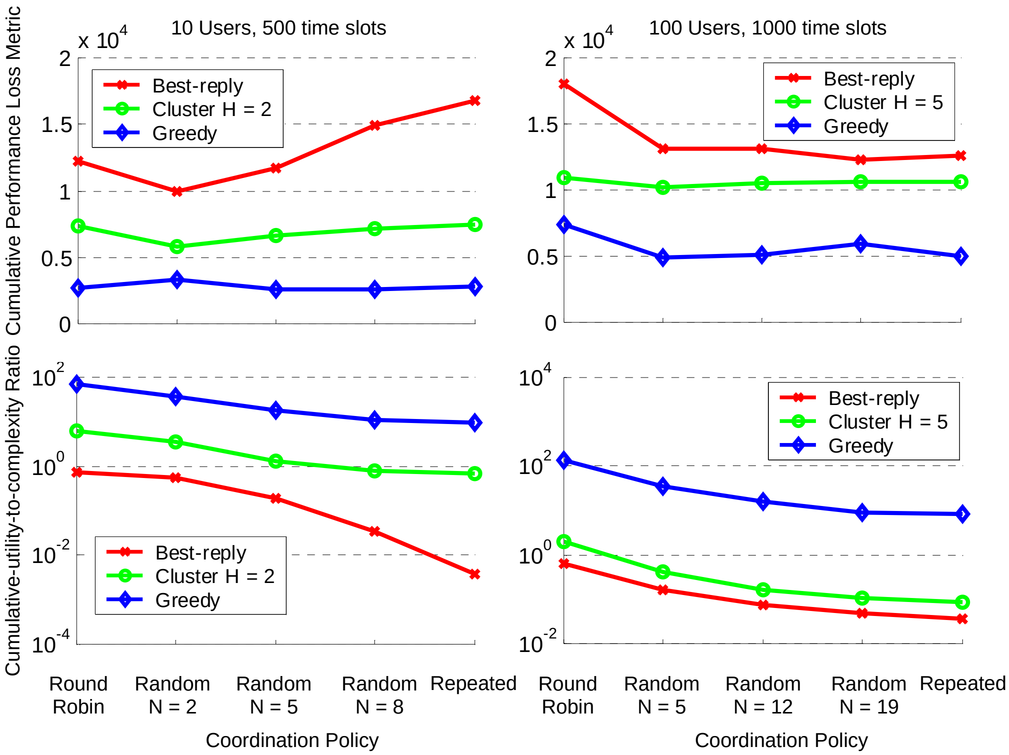

Fig. 7. Impact of the coordination policy on one agent's performance against heterogeneous opponents. (Top Row) Cumulative performance loss metric for the 10 user scenario and a 100 user scenario. (Bottom Row) Cumulative-utility-to-complexity ratio metric for the same scenarios, with the y-axis in the logscale.

VI. Conclusion

In this paper, we propose a solution for negotiating the network resources necessary for mobile multimedia streaming services over wireless networks. The focus of this paper is on understanding a single agent's (i.e. mobile multimedia device) ability to learn to improve its bids over time given its limited computational capabilities and (possibly) limited feedback about its environment (i.e. the bids of its opponents). Our experimental results show that, in a static environment, an agent can learn to make near optimal bids with limited information, and with very little computational overhead. However, in a more dynamic environment, with many agents simultaneously submitting bids, any single agent's ability to bid well depends heavily on the degree of imposed centralized coordination. In very dynamic scenarios, our results show that a greedy bidding approach performs better over time than more complex approaches that consider other agent’s bids. We note that, although we deployed the Progressive Second Price (PSP) auction, other auction mechanisms could be used in this framework.

REFERENCES

[1] S. Wee, W. Tan, J. Apostolopoulos, “Infrastructure-Based Streaming Media Overlay Networks", Multimedia over IP and Wireless Networks: Compression, Networking, and Systems, eds. P. A. Chou and M. van der Schaar, Academic Press, 2007.

[2] A. A. Lazar, N. Semret, “Design and analysis of the progressive second price auction for network bandwidth sharing,” *Telecommunication Systems – Special issue on Network Economics*, 1999.

[3] H. P. Young, Strategic learning and its limits, Princeton, NJ: Princeton University Press, 1998.

[4] S. Hart, “Adaptive heuristics,” Econometrica, vol. 73, no. 5, pp. 1401-1430, Sept, 2005.

[5] M. Jackson, “Mechanism Theory,” *Encyclopedia of Life Support Systems*, 2003.

[6] A. Vetro and C. Timmerer, “Digital Item Adaptation: Overview of Standardization and Research Activities,” to appear in IEEE Trans. on Multimedia, vol. 7, no. 3, Jun 2005.

[7] D. Fudenberg and J. Tirole, *Game Theory*. Cambridge, MA: MIT Press, 1991.

[8] P. Maillé, B. Tuffin, “Multi-bid versus progressive second price auctions in a stochastic environment,” in *Proc. IEEE INFOCOM*, Mar. 2004.

[9] R. Nadiminti, T. Mukhopadhyay, C. H. Kriebel, “Risk aversion and the value of information,” *Decision Support Systems*, pp. 241-254, Mar. 1996.

[10] P. A. Chou, "Streaming media on demand and live broadcast," in Multimedia over IP and Wireless Networks: Compression, Networking, and Systems, eds. P. A. Chou and M. van der Schaar, Academic Press, 2007.