Motion-Prediction-based Wireless Scheduling for Multi-User Panoramic Video Streaming

Jiangong Chen[*] Xudong Qin[*] Guangyu Zhu[†] Bo Ji[§] Bin Li[*]
[*]Department of ECBE, University of Rhode Island, Kingston, Rhode Island, USA
[†]Department of CSS, University of Rhode Island, Kingston, Rhode Island, USA
[§]Department of CS, Virginia Tech, Blacksburg, VA, USA

***Abstract*—Multi-user panoramic video streaming demands $4 \sim 6\times$ bandwidth of a regular video with the same resolution, which poses a significant challenge on the wireless scheduling design to achieve desired performance. On the other hand, recent studies reveal that one can effectively predict the user's Field-of-View (FoV) and thus simply deliver the corresponding portion instead of the entire scenes. Motivated by this important fact, we aim to employ autoregressive process for motion prediction and analytically characterize the user's successful viewing probability as a function of the delivered portion. Then, we consider the problem of wireless scheduling design with the goal of maximizing application-level throughput (i.e., average rate for successfully viewing the desired content) and service regularity performance (i.e., how often each user gets successful views) subject to the minimum required service rate and wireless interference constraints. As such, we incorporate users' successful viewing probabilities into our scheduling design and develop a scheduling algorithm that not only asymptotically achieves the optimal application-level throughput but also provides service regularity guarantees. Finally, we perform simulations to demonstrate the efficiency of our proposed algorithm using a real dataset of users' head motion.**

I. Introduction

The fast growth of wireless Head-Mounted Displays (HMDs) (such as Oculus Go and Google Daydream) spurs the multi-user panoramic video streaming application that can provide an immersive experience for a group of users, which is quite attractive in education, virtual museum touring, entertainment, just to name a few. In order to provide the best immersive experience, it requires providing high throughput (i.e., the average rate of successful views) and seamless experience (i.e., regular service) to each user. This is extremely challenging since each panoramic video delivery typically consumes $4 \sim 6\times$ bandwidth of a regular video with the same resolution (e.g., [1], [2]). This, together with the wireless interference, poses a significant challenge on the scheduling design that determines which and when users should be allowed to transmit for the multi-user panoramic video streaming application.

Fortunately, each user may only need to see as low as 20% of 360° scenes without affecting her/his visual perception, depending on her/his perspective. Imagine that a child is watching a panoramic roller coaster video, likely, he/she expects to see scenes in front of himself/herself only. Thus, if one can accurately predict a user's immediate motion, it suffices to deliver just about 20% of panoramic images, which dramatically reduces the consumption of precious wireless bandwidth. As such, it is important to actively exploit each user's motion prediction and incorporate it into the design of wireless scheduling algorithms.

Recent work (e.g., [1]–[3]) has developed efficient motion prediction algorithms and incorporated them into the panoramic video delivery in single-user or multicast wireless systems. However, these results are not directly applicable to the setting of multi-user panoramic video streaming in the presence of wireless interference, due to which only a subset of users can be scheduled to transmit at each time. While some more recent work (e.g., [4], [5]) leveraged motion prediction in the multi-user scheduling design, they provided neither performance guarantee nor regular services, where the service regularity metric is extremely important for panoramic video streaming.

As such, in this paper, we use the autoregressive process to predict each user's motion and analytically characterize the successful viewing probability as the function of the delivered portion surrounding the predicted viewport of the user. We are interested in whether each user can view his/her desired content instead of receiving as high as possible raw service rates. To differentiate it from the traditional concept of network-level throughput, we call it application-level throughput, which measures the average number of times a user successfully views his/her desired content. Indeed, in our context, getting high network-level throughput is not equivalent to the application-level throughput, as the user would not watch the content outside of the FoV even if it is delivered. For example, under the autoregressive process motion prediction model, delivering a half-sphere scene is sufficient to guarantee that the user successfully views the desired content and thus has the same application-level throughput as that of delivering the whole panoramic scene, but its network-level throughput is just half of that of delivering the whole panoramic scene.

In this paper, we are interested in maximizing application-level throughput (i.e., rate of average successful views) while meeting the minimum network-level throughput requirement as well as providing regular service guarantees for each user. In order to maximize application-level throughput, we formulate a stochastic network optimization problem that includes the successful viewing probability in the objective function, which

This research has been supported in part by NSF grants: CNS-1717108, CNS-1815563, CNS-1942383, and CNS-1651947.

is non-convex. To address the service regularity performance, we introduce Time-Since-Last-Service (TSLS) counter (see [6]) for each user to keep track of the elapsed time since the last time the user received the service. We nicely incorporate it into our scheduling design by using the stochastic network optimization framework (see [7] for an overview) while using a non-standard Lyapunov function. The main contributions of this paper are listed as follows:

- We analytically characterize the successful viewing probability as the function of the delivered portion of the panoramic scenes under the autoregressive process motion prediction model.
- We propose a concept of application-level throughput and formulate the multi-user scheduling for panoramic video streaming as a stochastic network optimization problem, where the objective is to maximize the application-level throughput subject to the minimum required network-level throughput and wireless interference constraints.
- We develop a motion prediction based scheduling algorithm that explicitly incorporates the motion prediction into the scheduling decision, and show that not only does it asymptotically optimize the application-level throughput, but it also provides regular service guarantees.
- We use the real dataset of users' head movement (see [1]) and evaluate the efficiency of our proposed algorithm via simulations.

The remainder of this paper is organized as follows: Section II reviews related work. Section III introduces system model and problem formulation. Section IV provides a motivating example and illustrates the impact of the scheduling design on both application-level throughput and service regularity performance. Section V introduces our motion prediction based scheduling algorithm and studies its performance. Section VI presents simulation results using the real dataset of users' head movement, and Section VII concludes this paper.

II. Related Work

In this section, we overview two main areas that are closely related to our work: panoramic video streaming and wireless scheduling design.

(a) Panoramic video streaming: Panoramic video streaming consumes a much larger bandwidth than the traditional video counterpart with the same resolution, which prohibits it from wide adoption especially via wireless. One major approach is to explore each user's motion prediction and incorporate it into the wireless transmission algorithm design. This lies in the fact that a user can only see as low as 20% of the 360° scenes and it is sufficient to deliver such a portion only if a user's motion can be accurately predicted. However, it cannot be avoided to introduce prediction error and thus it usually delivers a larger portion of the panoramic scenes to overcome the prediction error. Recent work (e.g., [1]–[3], [8]) has explored this idea and successfully incorporated it into the algorithm design. Another interesting line of research formulated the problem of adaptive rate selection in the panoramic video streaming as a Markov Decision Process with the goal of optimizing Quality of Experience (QoE) (e.g., [9]–[11]). Then, they used the reinforcement learning approach to effectively implement the adaptive rate selection algorithm and obtained the desired performance. However, all these works focused on a single-user case and neither provided any performance guarantees nor considered the case with multiple users, where the efficient scheduling design is required to manage the wireless interference. While some recent work (e.g., [4], [5]) explored the adaptive rate selection in the presence of multiple users, they did not analytically guarantee the system performance. Moreover, they mainly focused on the throughput or delay performance and did not provide any regular service performance guarantee, which is extremely important for panoramic video streaming.

(b) Wireless Scheduling Design: The design of wireless scheduling is to determine which and when users are allowed to transmit in the presence of wireless interference, which has been a central topic in wireless networks. The most related group of work to our setting concerns the efficient wireless scheduling design with various quality-of-service (QoS) requirements, such as throughput, delay, and service regularity. For example, some works focused on the wireless scheduling design with throughput performance guarantee for real-time traffic (e.g. [12]–[16]). Some other works focused on reducing the delay performance (e.g., [17]–[19]) and providing service regularity guarantees (e.g., [6]). However, all these scheduling designs aimed to optimize the network-level performance instead of application-level performance, which is crucial for panoramic video streaming. Moreover, these works did not explore any learning component, which is necessary for each user's motion prediction to enable multi-user panoramic video streaming.

III. System Model

We consider a system with N users, each of which downloads its panoramic video from a wireless access point (AP). We assume that each panoramic video consists of a series of chunks, where each chunk contains a series of 360° scenes with the same duration. We note that each user can only see a portion (around $20\% - 25\%$) of a chunk, known as *Field of View (FoV)*, and thus it is sufficient to deliver such a portion of the chunk if the head motion prediction is 100% accurate. However, it is unavoidable to incur prediction errors and thus we should always deliver a portion larger than the FoV to tolerate an imperfect motion prediction. To facilitate our mathematical modeling and algorithm developments, we unify the units for video chuck size and wireless transmission rate and assume that the system operates in a time-slotted manner. Let $S_n[t]$ be the allocated transmission rate for user n in time slot t, where $S_n[t] \in \mathcal{R} \triangleq \{0, R_1, R_2, \ldots, R_M\}$, where $0 < R_1 < R_2 \cdots < R_M \triangleq 1$ and R_1 corresponds to the rate to deliver the FoV of a video chunk and R_M is the rate to deliver a whole chunk. This is motivated by the fact that each chunk is partitioned into a finite number of tiles with the same duration and only a subset of tiles can be selected for transmission (e.g., [2], [20]). In each time slot, we will choose a set of tiles around the center of the predicted FoV based on the allocated transmission rate.

Due to the wireless interference constraints, only a subset of users can be scheduled to download their corresponding panoramic videos simultaneously in each time slot. Hence, in each time slot t, the AP needs to determine a set of users that it wants to deliver and their corresponding transmission rates $S_n[t], \forall n = 1, 2, \dots, N$. We call $\mathbf{S}[t] \triangleq (S_n[t])_{n=1}^N$ the *feasible rate vector*, which depends on the specific wireless interference constraints. We consider the case with block channel fading, where there are a finite number of global channel states and the global channel state is independently and identically distributed (*i.i.d.*) over time. Let $\mathcal{C}$ be the set of global channel states and $C[t] \in \mathcal{C}$ denotes the global channel state in time slot t. Let $\phi_c \triangleq \Pr\{C[t] = c\}$ denote the probability that the channel state is c in time slot t. We use $\mathcal{S}^{(c)}$ to denote the set of all feasible rate vectors when the channel state is c.

Let $I_n(S_n[t]) = 1$ denote that user n successfully views its desired content, i.e., the delivered content completely covers the FoV in time slot t when the transmitted rate is $S_n[t]$ and $I_n(S_n[t]) = 0$ otherwise. We use $\delta_n(S_n[t]) \triangleq \Pr\{I_n(S_n[t]) = 1\}$ to denote the *successful viewing probability* for user n in time slot t given its transmission rate $S_n[t]$. It is easy to see that $\delta_n(S_n[t])$ is a non-decreasing function with respect to $S_n[t]$. This is because a larger transmission rate $S_n[t]$ corresponds to delivering a larger portion of the video chunk that is around the center of the predicted FoV and thus can overcome a larger prediction error, which in turn yields a higher successful viewing probability.

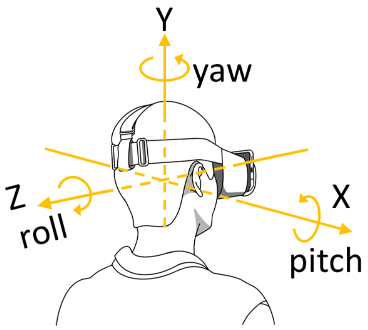


Fig. 1: Rotation coordinates.

In order to calculate the successful viewing probability for each user, we introduce 3-D rotation angles to capture a user's head motion. As shown in Fig. 1, a user could rotate his/her head in three axes: *pitch*, *yaw*, and *roll*. Let $X_n[t], Y_n[t]$ and $Z_n[t]$ be the rotating angles of the center of user n's FoV in pitch, yaw, and roll directions in time slot t, respectively. Since users rarely rotate head along the *roll* axis while watching panoramic videos, we focus on *pitch* and *yaw* axes as in [2], i.e., $(X_n[t], Y_n[t]), \forall n, t \geq 0$. Since the correlation between $X_n[t]$ and $Y_n[t]$ is much smaller than their individual autocorrelations (see [1]), we predict them separately based on the Autoregressive Process (AR) model (see [21]). While there are many machine learning-based prediction algorithms explored in existing works (e.g., [1]), we adopt the AR model here since it makes online real-time predictions and can quickly adapt to changing panoramic video contents and wireless environment.

We assume that the prediction errors of both pitch and yaw angles of user n follow normal distribution with standard deviation σ_n^X and σ_n^Y, respectively. This is motivated by the fact that under the AR model, the distribution of the prediction error converges to the normal distribution as the number of data samples goes to infinity (see [22, Theorem 8.2.1]). In Appendix A, we show that the successful viewing probability of user n can be expressed as follows:

$$\delta_n(S_n[t]) = \text{erf}^2\left(\frac{\gamma_n(S_n[t])}{\sqrt{2}}\right), \tag{1}$$

where $\text{erf}(x) \triangleq \frac{2}{\sqrt{\pi}}\int_0^x e^{-y^2}dy$ is the error function and $\gamma_n(S_n[t])$ is the number of standard deviations of the prediction error, when rate $S_n[t]$ is used. Here, $\gamma_n(S_n[t])$ follows from the basic geometry calculations and is available in Appendix A. Fig. 2 shows the successful viewing probability with respect to the allocated transmission rate, where we use the data traces of four different users watching the same panoramic video (see [1]) and obtain their standard deviations of prediction errors of both pitch and yaw angles under the AR model. We can observe from Fig. 2 that a larger standard deviation of the angle prediction error requires a larger allocated transmission rate to keep the same successful viewing probability.

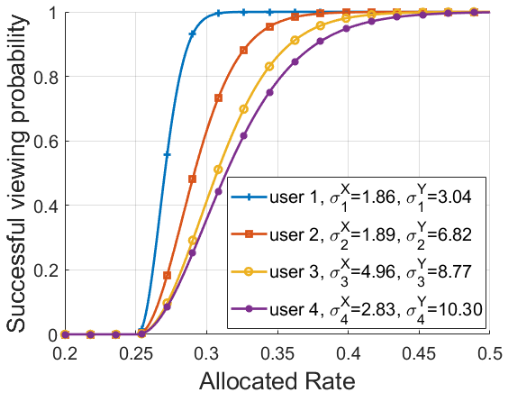


Fig. 2: Successful viewing probability.

In this paper, we would like to develop a scheduling algorithm to optimize both *application-level throughput* (defined as average total successful viewing probability) and *service regularity* (defined as the variance of the time between two consecutive successful views for each user) performance. This is motivated by the fact that each user would like to regularly and frequently view the desired panoramic scenes. In particular, our first goal is to maximize the application-level throughput subject to the constraint that the average allocated transmission rate should not be less than some minimum rate as well as wireless interference constraints, i.e.,

$$\max_{(S_n[t])_{n=1}^N} \lim_{L\to\infty} \frac{1}{L}\sum_{t=0}^{L-1}\sum_{n=1}^{N} w_n \mathbb{E}\left[\delta_n(S_n[t])\right] \tag{2}$$

$$\text{s.t. } (S_n[t])_{n=1}^N \in \mathcal{S}^{(C[t])}, \forall t \geq 0, \tag{3}$$

$$\lim_{L\to\infty} \frac{1}{L}\sum_{t=0}^{L-1} \mathbb{E}[S_n[t]] \geq r_n, \forall n, \tag{4}$$

where the objective function is the weighted sum of the application-level throughput, $w_n > 0$ is the weight of user n, and $r_n > 0$ is the minimum required allocated transmission rate for user n on average. Different from the traditional network optimization problem, we are interested in the average successful viewing probability or application-level throughput in each time slot instead of the average throughput. Even with the same average allocated transmission rate, the application-level throughput performance is different as shown in our motivating example in the next section.

To capture the service regularity performance, we introduce $g_n[m]$ to denote the time duration between the $(m+1)^{th}$ and m^{th} successful views of the user n. Noting the non-Markovian property of $g_n[m]$, similar to [6], we introduce a Time-Since-Last-Service (TSLS) counter $T_n[t]$ for each user n, which increases by one if user n does not see the desired content and reset to 0 otherwise. In particular, the evolution of $T_n[t]$ can be precisely described as follows:

$$T_n[t+1] \triangleq \begin{cases} 0, & \text{if } I_n(S_n[t]) = 1; \\ T_n[t]+1, & \text{otherwise.} \end{cases} \tag{5}$$

It has been shown in [6] that minimizing the normalized variance of $g_n[m]$ is equivalent to minimizing the expected $T_n[t]$. As such, our second goal is to keep the following quantity as small as possible:

$$\lim_{L\to\infty} \frac{1}{L}\sum_{t=0}^{L-1} \mathbb{E}\left[T_n[t]\right].$$

Next, we will first study a motivating example to illustrate the possibility of improving both application-level throughput and service regularity performance simultaneously by carefully designing a scheduling algorithm, and then accomplish our dual objective by developing a parameterized wireless scheduling algorithm.

IV. A Motivating Example

In this section, we provide an example to illustrate the impact of the scheduling design on both application-level throughput and service regularity performance in multi-user panoramic video streaming. We consider $N = 4$ users, where the total service rates of all users are at most one in each time slot. Since Round-Robin is known to provide good service regularity performance, we consider two different Round-Robin (RR) scheduling algorithms: (i) RR I that provides each user with the rate of one in turn; (ii) RR II that serves the first two users with the rate of 0.5 in even time slots and the other two users in odd time slots. Table I-(a) provides the allocated service rate for each user under these two different versions of the Round-Robin algorithm. Here, we assume that $\delta_n(0.5) = \delta_n(1) = 1$, $\forall n$, which is true for data traces in [1] under the AR prediction model.

$S_n[t]$ time user	0	1	2	3	…
User 1	1, 0.5	0, 0	0, 0.5	0, 0	…
User 2	0, 0.5	1, 0	0, 0.5	0, 0	…
User 3	0, 0	0, 0.5	1, 0	0, 0.5	…
User 4	0, 0	0, 0.5	0, 0	1, 0.5	…

(a) Service rate of each user in each time slot.

$\delta_n(S_n[t])$ time user	0	1	2	3	…
User 1	1, 1	0, 0	0, 1	0, 0	…
User 2	0, 1	1, 0	0, 1	0, 0	…
User 3	0, 0	0, 1	1, 0	0, 1	…
User 4	0, 0	0, 1	0, 0	1, 1	…

(b) Application-level throughput of each user in each time slot.

TABLE I: Service rate and application-level throughput under two different versions of RR algorithms: the results under RR I and RR II are colored by blue and red, respectively.

While these two Round-Robin algorithms yield the same average service rate of 0.25 for each user, i.e., $\lim_{L\to\infty}\frac{1}{L}\sum_{t=0}^{L-1} S_n[t] = 0.25$, $\forall n = 1,2,3,4$, they result in different application-level throughput and service regularity performance. Indeed, $S_n[t] = 0.5$ is large enough to tolerate prediction error and thus yields the successful delivery of desired content for user n in time slot t, which can be demonstrated in our simulations using the collected users' head motion data (cf. Fig. 2). As such, for each user n, both $S_n[t] = 0.5$ and $S_n[t] = 1$ can result in a successful delivery. Table I-(b) shows the successful delivery of each user in each time slot. Hence, the application-level throughput is 1 and 2 under the first and second version of the Round-Robin algorithm, respectively.

Fig. 3 shows the evolution of the TSLS counter of the first user under both versions of the Round-Robin algorithm. We can easily compute that the average TSLS for each user is 1.5 and 0.5 under the first and second versions of the Round-Robin algorithm, respectively. To summarize, we can see that the application-level throughput and average TSLS under the second version of the Round-Robin algorithm are twice and three times better than that under its first version.

The above example demonstrates the significant impact of the scheduling design on both application-level throughput and service regularity performance. In cases where the allocated transmission rate can be scheduled from a large discrete space, more complex scheduling decisions should be explored. In the next section, we will develop an efficient scheduling algorithm that yields both good application-level throughput and service regularity performance.

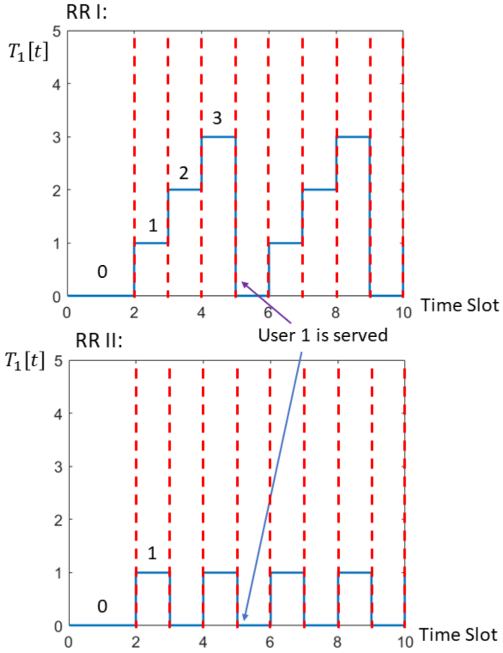


Fig. 3: Example of TSLS dynamics of user 1

V. Algorithm Design and Analysis

In this section, we will develop a scheduling algorithm and show that it achieves asymptotically-optimal application-level throughput and provides service regularity guarantees.

We use the stochastic network optimization framework (e.g., [7]) to introduce a virtual queue for each user that measures the degree of violation of the average service rate constraint. Specifically, we use $Q_n[t]$ to denote the virtual queue length for user n in time slot t. The amount of traffic entering the virtual queue n in time slot t is r_n, while the amount of service for virtual queue n in time slot t is $S_n[t]$. Then, the evolution of the virtual queue n can be described as follows:

$$Q_n\left[t+1\right] \triangleq \left(Q_n\left[t\right]+r_n-S_n[t]\right)^+, \forall n, \forall t, \tag{6}$$

where $(x)^+ = \max\{x, 0\}$. We say that virtual queue n is *mean rate stable* (see [7]) if $\lim_{t\to\infty}\frac{\mathbb{E}[Q_n[t]]}{t} = 0$. If the virtual queue n is mean rate stable, then the average service rate of user n is at least r_n (see [7, Theorem 2.5]). The following algorithm is derived by minimizing the difference between the drift of the Lyapunov function

$$V[t] = \frac{1}{2}\sum_{n=1}^{N} Q_n^2[t] + \eta\sum_{n=1}^{N} T_n[t]$$

and the application-level throughput $K\sum_{n=1}^{N} w_n\mathbb{E}[\delta_n(S_n[t])]$ in time slot t, where η and K are controlled positive real numbers, i.e.,

$$V[t+1] - V[t] - K\sum_{n=1}^{N} w_n\mathbb{E}[\delta_n(S_n[t])].$$

Different from selecting a quadratic Lyapunov function in the classical stochastic network optimization framework, we choose the sum of quadratic virtual queue function and the linear TSLS function as our Lyapunov function. This is because we aim to keep both virtual queue lengths and TSLS counters as small as possible, yielding the mean rate stability and desired service regularity performance. Our scheduling algorithm is described as follows:

Algorithm 1 Motion Prediction based Scheduling (MPS)

In each time slot t, given channel state $C[t] = c$.

AR-based Motion Prediction: Each user n predicts its pitch and yaw angles $\widehat{X}_n[t]$ and $\widehat{Y}_n[t]$ based on the previous W slots' pitch samples $(X_n[t-1], X_n[t-2], \cdots, X_n[t-W])$ and yaw samples $(Y_n[t-1], Y_n[t-2], \cdots, Y_n[t-W])$ using the AR model, i.e.,

$$\widehat{X}_n[t] = -\sum_{k=1}^{W} a_n[k]X_n[t-k]$$

$$\text{and} \quad \widehat{Y}_n[t] = -\sum_{k=1}^{W} b_n[k]Y_n[t-k],$$

where $a_n[1], a_n[2], \cdots, a_n[W]$ and $b_n[1], b_n[2], \cdots, b_n[W]$ are the prediction coefficients that are estimated by using the standard Yule-Walker equation (see [21]).

Wireless Scheduling:select the schedule $\mathbf{S}^*[t]$ satisfying

$$\mathbf{S}^*[t] \in \arg\max_{\mathbf{S}\in\mathcal{S}^{(c)}} \sum_{n=1}^{N}(S_n[t]Q_n[t] + (\eta T_n[t] + Kw_n)\delta_n(S_n[t])),$$

where η and K are some positive numbers, and $\delta_n(S_n[t])$ is calculated based on the sample variances $(\widehat{\sigma}_n^X[t])^2$ and $(\widehat{\sigma}_n^Y[t])^2$ of prediction errors of pitch and yaw angles.

Our algorithm has two major components: i) AR-based motion prediction, and ii) wireless scheduling. In each time slot t, we use the AR model for the pitch and yaw angle prediction and updates the prediction coefficients based on the Yule-Walker equation (see [21]). Besides, we obtain the sample variance of prediction errors of pitch and yaw angles until time t, and then use it to calculate the successful viewing probability, which is critical for the wireless scheduling design. We incorporate the instantaneous application-level throughput, TSLS counter, and virtual queues into the scheduling design with the algorithmic parameters η and K balancing their weights. When the virtual queue length of a user is large, it means that the user has not received a sufficient amount of service rates, which enforces it to be scheduled. Similarly, if a user has not been served for a long time, the TSLS counter will linearly increase and thus the user will get a high priority to get served. Also, the user with a larger weight on the application-level throughput should always have a high priority to be scheduled to achieve a large weighted sum of application-level throughput.

Moreover, when $\eta = 0$, our algorithm coincides with the traditional "drift-plus-penalty" method for classical utility maximization problems (e.g., [23]). The larger the value of

η, the more emphasis on the TSLS counter and thus keeps services more regular. When $K = 0$, the goal is to keep service as regular as possible while meeting the minimum rate requirement and the resulting algorithm is similar to the Regular Service Guarantee Algorithm in [6]. The larger the value of K, the larger the weight put on the instantaneous throughput and thus leads to the larger application-level throughput. However, similar to the well-known MaxWeight scheduling algorithms (e.g., [24]), our proposed MPS algorithm also has a high computational complexity (which could be exponential) in general.

Next, we show that our proposed MPS Algorithm asymptotically optimizes the application-level throughput and provides service regularity guarantees while meeting the minimum service rate requirement.

Theorem 1. *Under the MPS Algorithm, all virtual queues are mean rate stable, which implies that the average service rate of each user is at least r_n. In addition, the weighted sum of mean TSLS counters and application-level throughput can be respectively bounded from above as follows:*

$$\lim_{L\to\infty} \frac{1}{L} \sum_{t=0}^{L-1} \sum_{n=1}^{N} U_n^* \cdot \mathbb{E}[T_n[t]] \le \frac{B(\eta) + KNw_{\max}}{\eta}$$

$$\textit{and} \lim_{L\to\infty} \frac{1}{L} \sum_{t=0}^{L-1} \sum_{n=1}^{N} \mathbb{E}[w_n \delta_n(S_n[t])] \ge U^* - \frac{B(\eta)}{K}$$

where $B(\eta) \triangleq \sum_{n=1}^{N} (r_n^2 + R_M^2)/2 + \eta N$, U_n^* is the successful viewing probability of user n when optimal weighted sum of application-level throughput is achieved (i.e., $U^* \triangleq \sum_{n=1}^{N} w_n U_n^*$ is the optimal value of the optimization problem (2)-(4)*).*

Proof. Select the Lyapunov function

$$V[t] = \frac{1}{2} \sum_{n=1}^{N} Q_n^2[t] + \eta \sum_{n=1}^{N} T_n[t].$$

and consider its conditional Lyapunov drift given the current system state $\mathbf{H}[t] \triangleq (Q_n[t], T_n[t])_{n=1}^{N}$.

$$\begin{aligned}
&\Delta V[t] \triangleq \mathbb{E}\left[V[t+1] - V[t] | \mathbf{H}[t]\right] \\
=&\mathbb{E}\Bigg[\frac{1}{2} \sum_{n=1}^{N} (Q_n^2[t+1] - Q_n^2[t]) \\
&\quad + \eta \sum_{n=1}^{N} (T_n[t+1] - T_n[t]) \Big| \mathbf{H}[t]\Bigg] \\
\overset{(a)}{\le}& \frac{1}{2} \sum_{n=1}^{N} \mathbb{E}\left[(Q_n[t] + r_n - S_n^*[t])^2 - Q_n^2[t] \big| \mathbf{H}[t]\right] \\
&\quad + \eta \sum_{n=1}^{N} \mathbb{E}\left[(T_n[t] + 1)(1 - I_n(S_n^*[t])) - T_n[t] | \mathbf{H}[t]\right] \\
\overset{(b)}{\le}& \sum_{n=1}^{N} \mathbb{E}\left[Q_n[t](r_n - S_n^*[t]) - \eta T_n[t] I_n(S_n^*[t]) | \mathbf{H}[t]\right] + B(\eta),
\end{aligned}$$

where step (a) follows from the dynamics of $Q_n[t]$ (cf. (6)) and $T_n[t+1]$ (cf. (5)); (b) is true for

$$B(\eta) \triangleq \sum_{n=1}^{N} (r_n^2 + R_M^2)/2 + \eta N.$$

By subtracting $K \sum_{n=1}^{N} w_n \mathbb{E}[\delta_n(S_n^*[t]) | \mathbf{H}[t]]$ on both sides of the above Lyapunov drift $\Delta V[t]$, we have

$$\begin{aligned}
&\Delta V[t] - K \sum_{n=1}^{N} w_n \mathbb{E}[\delta_n(S_n^*[t]) | \mathbf{H}[t]] \\
\le& \sum_{n=1}^{N} Q_n[t] r_n - \sum_{n=1}^{N} Q_n[t] \mathbb{E}\left[S_n^*[t] | \mathbf{H}[t]\right] + B(\eta) \\
&- \sum_{n=1}^{N} (\eta T_n[t] + K w_n) \mathbb{E}[\delta_n(S_n^*[t]) | \mathbf{H}[t]] \\
\le& \sum_{n=1}^{N} Q_n[t] r_n - \sum_{n=1}^{N} Q_n[t] \mathbb{E}\left[\widehat{S}_n[t] \Big| \mathbf{H}[t]\right] + B(\eta) \\
&- \sum_{n=1}^{N} (\eta T_n[t] + K w_n) \mathbb{E}[\delta_n(\widehat{S}_n[t]) | \mathbf{H}[t]], \qquad (7)
\end{aligned}$$

where the last step follows from the definition of our proposed MPS Algorithm.

We note that there exists a randomized stationary schedule $(\widehat{S}_n[t])_{n=1}^{N}$ such that

$$\mathbb{E}[\widehat{S}_n[t]] \ge r_n, \forall n, t, \qquad (8)$$

$$U_n^* = \mathbb{E}[\delta_n(\widehat{S}_n[t])] \qquad (9)$$

$$U^* = \sum_{n=1}^{N} w_n U_n^*, \qquad (10)$$

where U^* is the optimal value of the optimization problem (2)-(4). Indeed, let $p_{n,m}^{(c)} \triangleq \Pr\{\widehat{S}_n[t] = R_m\}$ be the probability of user n selecting rate R_m when the global channel state is in c. Then our optimization problem (2)-(4) can be written as follows:

$$\begin{aligned}
\max &\sum_{c\in\mathcal{C}} \phi_c \sum_{n=1}^{N} w_n \sum_{m=1}^{M} p_{n,m}^{(c)} \delta_n(R_m) \\
\text{s.t. } &\sum_{c\in\mathcal{C}} \phi_c \sum_{m=1}^{M} p_{n,m}^{(c)} R_m \ge r_n, \forall n, \\
&(\sum_{m=1}^{M} p_{n,m}^{(c)} R_m)_{n=1}^{N} \in \mathbf{CH}(\mathcal{S}^{(c)}),
\end{aligned}$$

where we recall that $\mathbf{CH}(\mathcal{A})$ is the convex hull of the set $\mathcal{A}$. This is a standard convex optimization problem and thus has an optimal solution.

By using the property of the stationary randomized schedule

$\widehat{\mathbf{S}}[t]$ (cf. (8)-(10)), inequality (7) becomes

$$\begin{aligned}&\Delta V[t] - K\sum_{n=1}^{N} w_n \mathbb{E}[\delta_n(S_n^*[t])|\mathbf{H}[t]]\\ \le& B(\eta) - \sum_{n=1}^{N}(\eta T_n[t] + Kw_n)U_n^*\\ \le& -\eta\sum_{n=1}^{N} U_n^* T_n[t] + B(\eta) - KU^*.\end{aligned}$$

Taking the expectation on both sides, we have

$$\begin{aligned}&\mathbb{E}[V[t]] - \mathbb{E}[V[t]] - K\sum_{n=1}^{N} w_n \mathbb{E}[\delta_n(S_n^*[t])]\\ \le& B(\eta) - \eta\sum_{n=1}^{N} U_n^* \mathbb{E}[T_n[t]] - KU^*,\end{aligned} \tag{11}$$

holding for all $t \ge 0$.

By summing both sides of (11) over $t \in \{0, 1, \cdots, L-1\}$ and dividing by L, we have

$$\begin{aligned}&\frac{1}{L}(\mathbb{E}[V(L)] - \mathbb{E}[V(0)]) - K\frac{1}{L}\sum_{t=0}^{L-1}\sum_{n=1}^{N} w_n \mathbb{E}[\delta_n(S_n^*[t])]\\ \le& B(\eta) - \eta\frac{1}{L}\sum_{t=0}^{L-1}\sum_{n=1}^{N} U_n^* \mathbb{E}[T_n[t]] - KU^*.\end{aligned} \tag{12}$$

Hence, we have

$$\begin{aligned}&\eta\frac{1}{L}\sum_{t=0}^{L-1}\sum_{n=1}^{N} U_n^* \mathbb{E}[T_n[t]]\\ \le& B(\eta) + K\frac{1}{L}\sum_{t=0}^{L-1}\sum_{n=1}^{N} \mathbb{E}[w_n\delta_n(S_n^*[t])] + \frac{1}{L}\mathbb{E}[V(0)].\end{aligned}$$

By taking the limit as $L \to \infty$, we have

$$\lim_{L\to\infty}\frac{1}{L}\sum_{t=0}^{L-1}\sum_{n=1}^{N} U_n^* \mathbb{E}[T_n[t]] \le \frac{B(\eta) + KNw_{\max}}{\eta}.$$

In addition, from (12) we have

$$K\frac{1}{L}\sum_{t=0}^{L-1}\sum_{n=1}^{N} \mathbb{E}[w_n\delta_n(S_n[t])] \ge \quad KU^* - B(\eta) - \frac{1}{L}\mathbb{E}[V(0)].$$

By taking the limit as $L \to \infty$, we have

$$\lim_{L\to\infty}\frac{1}{L}\sum_{t=0}^{L-1}\sum_{n=1}^{N} w_n \mathbb{E}[\delta_n(S_n^*[t])] \ge U^* - \frac{B(\eta)}{K}.$$

Finally, we will show that all virtual queues are mean rate stable. From (12), we have

$$\mathbb{E}\left[V[L]\right] - \mathbb{E}\left[V[0]\right] \le L(B(\eta) + KNw_{\max}), \forall t \ge 0.$$

Using the fact that $V[L] \ge \sum_{n=1}^{N} Q_n^2[L]/2$ yields

$$\frac{1}{2}\sum_{n=1}^{N}\mathbb{E}\left[Q_n^2[L]\right] \le L(B(\eta) + KNw_{\max}) + \mathbb{E}\left[V[0]\right].$$

Therefore, for each $n \in \{1, \ldots, N\}$, we have

$$\mathbb{E}\left[Q_n^2[L]\right] \le 2\left(L(B(\eta) + KNw_{\max}) + \mathbb{E}\left[V[0]\right]\right)$$

However, because the variance of $|Q_n[L]|$ cannot be negative, we have $\mathbb{E}\left[Q_n^2[L]\right] \ge \left(\mathbb{E}\left[Q_n[L]\right]\right)^2$. Thus, we have

$$\mathbb{E}\left[Q_n[L]\right] \le \sqrt{2\left(L(B(\eta) + KNw_{\max}) + \mathbb{E}\left[V[0]\right]\right)}$$

By dividing by L and taking a limit as $L \to \infty$, we have

$$\lim_{L\to\infty}\frac{\mathbb{E}\left[Q_n[L]\right]}{L} \le 0$$

Since $Q_n[L] \ge 0$, we have $\lim_{L\to\infty}\mathbb{E}\left[Q_n[L]\right]/L = 0$, which implies that virtual queue n is mean rate stable. □

The above theorem reveals the tradeoff between the weighted sum of application-level throughput and service regularity performance. Indeed, as the parameter K increases, the application-level throughput improves, while the upper bound on the weighted sum of mean TSLS counters increases (i.e., the service regularity performance deteriorates). Besides, when η increases, the service regularity performance improves but is at the cost of reduced application-level throughput.

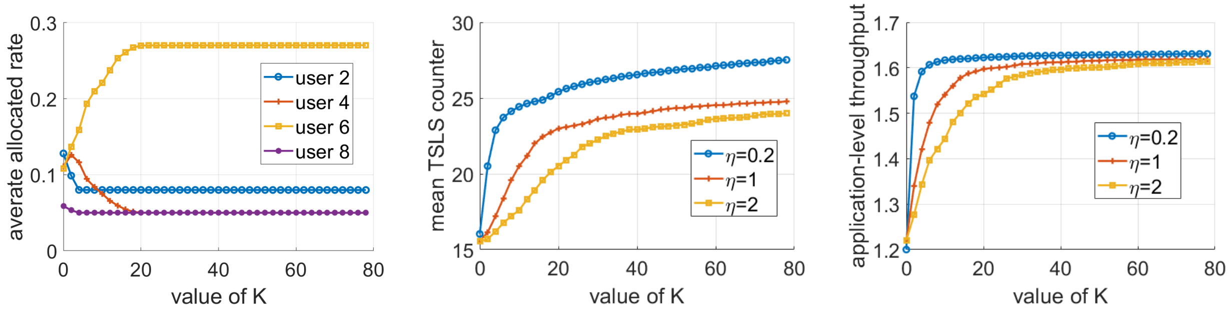
VI. Simulations

In this section, we perform simulations to evaluate the efficiency of our proposed MPS algorithm. We consider $N = 8$ users. Each user experiences i.i.d. ON-OFF channel fading over time with probability p_n that its channel is ON in each time slot. We assume that at most two users can be scheduled in each time slot and the total rate of all scheduled users is no more than 1. Each user n has a minimum required service rate r_n and weight w_n on the application-level throughput. The allocated transmission rate can be selected from the set $\mathcal{R} = \{0, 0.3, 0.4, 0.5, 0.7, 1\}$. The detailed simulation parameters are available in TABLE II. In addition, we use synthetic head motion data generated from the dataset in [1] for each user.

	user 1	user 2	user 3	user 4
Required rate r_n	0.1	0.08	0.11	0.05
Weight w_n	0.2	0.1	1.0	0.8
Fading prob. p_n	0.8	0.9	0.7	0.9
	user 5	user 6	user 7	user 8
Required rate r_n	0.18	0.06	0.16	0.05
Weight w_n	0.9	1.2	0.3	0.2
Fading prob. p_n	0.8	0.9	0.7	0.8

TABLE II: Simulation setup.

Fig. 4a shows the average allocated rates of four different users with respect to parameter K when $\eta = 1$. We can



(a) Average rate for each user: $\eta = 1$

(b) Average TSLS

(c) Application-level throughput

Fig. 4: Performance of the MPS algorithm.

observe from Fig. 4a that our proposed MPS algorithm guarantees the minimum service rate required by each user. Fig. 4b and Fig. 4c show an impact of the parameter K on the performance of our proposed MPS algorithm. We can observe from Fig. 4b and Fig. 4c that for each fixed value of η, as the parameter K increases, both mean TSLS and application-level throughput increases. The reason is that the larger the K, the more emphasis on the application-level throughput and the lower priority on the TSLS, resulting in the application-level throughput improvement and service regularity deterioration. This also matches our derived bounds in Theorem 1 that the upper bound on the average TSLS linearly increases with the parameter K and the lower bound on the application-level throughput also increases as K increases. Besides, as η becomes larger, both mean TSLS and application-level throughput become smaller. The reason lies in the fact that a large η gives a high priority on the TSLS counter and enforces to provide more regular service, but it is at the cost of reducing the application-level throughput. This again matches our derived bounds in Theorem 1 that both the upper bound on the average TSLS and the lower bound on the application-level throughput decrease as the parameter η increases.

VII. Conclusion

In this work, we studied the problem of wireless scheduling design for multi-user panoramic video streaming to optimize application-level throughput and service regularity performance. We used the autoregressive process to predict the user's motion and analytically characterized the successful viewing probability as the function of the delivered portion. We used the Time-Since-Last-Service counter to account for the service regularity performance and developed a motion prediction based scheduling algorithm by integrating it into the stochastic network optimization framework. We proved that our proposed algorithm can provide desired application-level throughput and service regularity guarantees. Finally, we demonstrated the efficiency of our proposed algorithm through simulations with real datasets.

Appendix A
Successful viewing probability

In this section, we will characterize the successful viewing probability $\delta_n(S_n[t]; \sigma_n^X, \sigma_n^Y)$ by assuming that the prediction errors of the pitch and yaw angles follow Gaussian distribution with zero mean and standard deviation σ_n^X and σ_n^Y, respectively. To that end, we first need to know whether the n^{th} user's motion prediction is successful, i.e., the delivered portion completely covers the actual FoV.

As shown in Fig. 5, assume that a user is at the center denoted by the point O and the predicted center of the FoV is O'. The delivered content could be seen as a spherical crown centered by O' in different sizes. Notice that $\square ABCD$ in Fig. 6 lies on the cross section of the sphere whose radius is $O'F$ such that $\theta_0/2$ coincides with the beam angle $\angle FOO'$ of the spherical crown whose size is equal to the FoV as shown in Fig. 5a and Fig. 5b. Recall that α_0 and β_0 are horizontal and vertical angles corresponding to the pitch and yaw axis, respectively. Then, by simple geometry calculation, we obtain the beam angle θ_0 as follows:

$$\theta_0 = \text{diag}(\alpha_0, \beta_0),$$

where $\text{diag}(\cdot)$ refers to the diagonal angle, which is defined as:

$$\text{diag}(\alpha, \beta) \triangleq 2 \arccos \left[\frac{1}{\sqrt{1 + \tan^2(\frac{\alpha}{2}) + \tan^2(\frac{\beta}{2})}} \right], \quad (13)$$

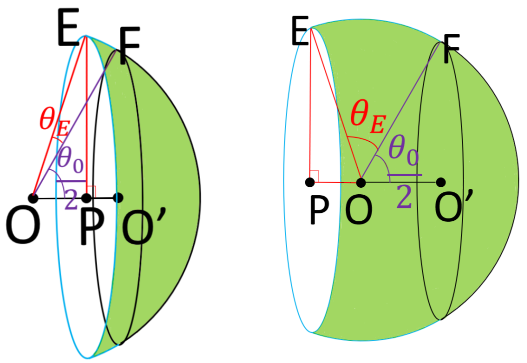
where $\alpha \leq \pi$ and $\beta \leq \pi$. Indeed, considering the right triangles $\triangle OO'Q$ and $\triangle OO'R$ in Fig. 6, we have

$$O'Q = OO' \tan(\alpha_0/2), O'R = OO' \tan(\beta_0/2). \quad (14)$$

In the right triangle $\triangle AQO'$, we have $AO' = \sqrt{AQ^2 + O'Q^2}$. Assume the radius of the sphere is 1, i.e., $AO = 1$. We have $AO'^2 + OO'^2 = AO^2 = 1$. Combining the above equations, we have

$$OO' = \frac{1}{\sqrt{1 + \tan^2(\alpha_0/2) + \tan^2(\beta_0/2)}}.$$

This, together with the fact that $\cos(\theta_0/2) = OO'$, implies (13).



(a) $\theta_0 + 2\theta_E \leq \pi$.

(b) $\theta_0 + 2\theta_E \geq \pi$.

Fig. 5: The delivered content.

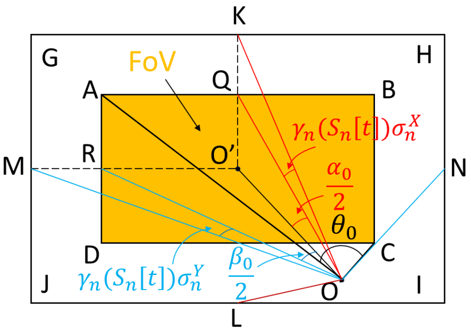


Fig. 6: FoV and various angles.

Since α_0 and β_0 are only determined by the model of the HMD, θ_0 is a constant. Let θ_E be the transmission margin $\angle FOE$ outside the predicted FoV, i.e., the extra portion delivered to overcome the prediction error. Apparently, θ_E can be determined by the allocated transmission rate $S_n[t]$ as follows:

$$\theta_0 + 2\theta_E = 2\arccos(1 - 2S_n[t]). \tag{15}$$

Indeed, the height h of a spherical crown is proportional to its surface area, while the surface area ratio is equal to the allocated rate $S_n[t]$. Thus, $h/2R = S_n[t]/1$. Recall that the radius of the sphere is assumed to 1, i.e., $R = 1$. Then we have $h = 2S_n[t]$. In Fig. 5a, $h = R - OP$, in Fig. 5b, $h = R + OP$. Yet in both cases, $h = R - \cos(\theta_E + \theta_0/2)$. This, together with $R = 1$ and $h = 2S_n[t]$, implies (15).

In this work, we use the autoregressive process to predict the user's orientation in pitch and yaw axes. Let γ be the number of standard deviations. Let $\alpha_n(\gamma) \triangleq \alpha_0 + 2\gamma\sigma_n^X$ and $\beta_n(\gamma) \triangleq \beta_0 + 2\gamma\sigma_n^Y$ be the vertical angle (e.g., $\angle KOL$ in Fig. 6) and horizontal angle (e.g., $\angle MON$ in Fig. 6) of the delivered portion, respectively. Let $Ang(\gamma)$ be the diagonal angle of the delivered portion and can be calculated depending on the values of $\alpha_n(\gamma)$ and $\beta_n(\gamma)$.

- If both α_n and β_n are smaller than π, as shown in Fig. 5a, we have
$$Ang(\gamma) = \text{diag}(\alpha_n(\gamma), \beta_n(\gamma)).$$
- If one of $\alpha_n(\gamma)$ and $\beta_n(\gamma)$ is smaller than π, then in order to guarantee the continuity of $Ang(\gamma)$, we define $Ang(\gamma) = \pi$.
- If both $\alpha_n(\gamma)$ and $\beta_n(\gamma)$ are greater than π, as shown in Fig. 5b, we have
$$Ang(\gamma) = 2\pi - \text{diag}\big(2\pi - \alpha_n(\gamma), 2\pi - \beta_n(\gamma)\big).$$

Note that $Ang(\gamma)$ is equal to $\theta_0 + 2\theta_E$ and thus given $S_n[t]$, $\gamma(S_n[t])$ can be calculated as follows.

$$Ang(\gamma(S_n[t])) = 2\arccos(1 - 2S_n[t]). \tag{16}$$

Let $\widehat{X}_n[t]$ and $\widehat{Y}_n[t]$ are the predicted angles in the pitch and yaw axis, respectively, under the Autoregressive Process. Hence, given $S_n[t]$, there is a successful view only when both events $\mathcal{A}_X \triangleq \{\widehat{X}_n[t] - \gamma(S_n[t])\sigma_n^X < X_n[t] < \widehat{X}_n[t] + \gamma(S_n[t])\sigma_n^X\}$ and $\mathcal{A}_Y \triangleq \{\widehat{Y}_n[t] - \gamma(S_n[t])\sigma_n^Y < Y_n[t] < \widehat{Y}_n[t] + \gamma(S_n[t])\sigma_n^Y\}$ happen and thus the successful viewing probability $\delta_n(S_n[t])$ can be calculated as follows:

$$\begin{aligned}\delta_n(S_n[t]) &= \Pr\{\mathcal{A}_X \cap \mathcal{A}_Y\}\\ &= \Pr\{\mathcal{A}_X\}\Pr\{\mathcal{A}_Y\}\\ &= \text{erf}^2\left(\frac{\gamma_n(S_n[t])}{\sqrt{2}}\right).\end{aligned}$$

References

[1] Y. Bao, H. Wu, T. Zhang, A. A. Ramli, and X. Liu, "Shooting a moving target: Motion-prediction-based transmission for 360-degree videos," in *2016 IEEE International Conference on Big Data (Big Data)*. IEEE, 2016, pp. 1161–1170.

[2] F. Qian, B. Han, Q. Xiao, and V. Gopalakrishnan, "Flare: Practical viewport-adaptive 360-degree video streaming for mobile devices," in *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, 2018, pp. 99–114.

[3] M. Hosseini and V. Swaminathan, "Adaptive 360 vr video streaming: Divide and conquer," in *2016 IEEE International Symposium on Multimedia (ISM)*. IEEE, 2016, pp. 107–110.

[4] C. Perfecto, M. S. Elbamby, J. Del Ser, and M. Bennis, "Taming the latency in multi-user vr 360°: A qoe-aware deep learning-aided multicast framework," *IEEE Transactions on Communications*, vol. 68, no. 4, pp. 2491–2508, 2020.

[5] J. Chakareski, "Viewport-adaptive scalable multi-user virtual reality mobile-edge streaming," *IEEE Transactions on Image Processing*, vol. 29, pp. 6330–6342, 2020.

[6] B. Li, R. Li, and A. Eryilmaz, "Throughput-optimal scheduling design with regular service guarantees in wireless networks," *IEEE/ACM Transactions on Networking*, vol. 23, no. 5, pp. 1542–1552, 2014.

[7] M. J. Neely, "Stochastic network optimization with application to communication and queueing systems," *Synthesis Lectures on Communication Networks*, vol. 3, no. 1, pp. 1–211, 2010.

[8] J. Chen, B. Li, and R. Srikant, "Thompson-sampling-based wireless transmission for panoramic video streaming," in *2020 18th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOPT)*. IEEE, 2020, pp. 1–3.

[9] M. Xu, Y. Song, J. Wang, M. Qiao, L. Huo, and Z. Wang, "Predicting head movement in panoramic video: A deep reinforcement learning approach," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 11, pp. 2693–2708, 2018.

[10] N. Kan, J. Zou, K. Tang, C. Li, N. Liu, and H. Xiong, "Deep reinforcement learning-based rate adaptation for adaptive 360-degree video streaming," in *ICASSP 2019-2019 IEEE International Conference*

on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2019, pp. 4030–4034.

[11] Y. Zhang, P. Zhao, K. Bian, Y. Liu, L. Song, and X. Li, "Drl360: 360-degree video streaming with deep reinforcement learning," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2019, pp. 1252–1260.

[12] I.-H. Hou, V. Borkar, and P. Kumar, *A theory of QoS for wireless*. IEEE, 2009.

[13] I.-H. Hou and P. Kumar, "Admission control and scheduling for qos guarantees for variable-bit-rate applications on wireless channels," in *Proceedings of the tenth ACM international symposium on Mobile ad hoc networking and computing*, 2009, pp. 175–184.

[14] I.-H. Hou, "Scheduling heterogeneous real-time traffic over fading wireless channels," *IEEE/ACM Transactions on Networking*, vol. 22, no. 5, pp. 1631–1644, 2013.

[15] J. J. Jaramillo and R. Srikant, "Optimal scheduling for fair resource allocation in ad hoc networks with elastic and inelastic traffic," in *2010 Proceedings IEEE INFOCOM*. IEEE, 2010, pp. 1–9.

[16] B. Li and A. Eryilmaz, "Optimal distributed scheduling under time-varying conditions: A fast-csma algorithm with applications," *IEEE Transactions on Wireless Communications*, vol. 12, no. 7, pp. 3278–3288, 2013.

[17] L. Bui, R. Srikant, and A. Stolyar, "Novel architectures and algorithms for delay reduction in back-pressure scheduling and routing," in *IEEE INFOCOM 2009*. IEEE, 2009, pp. 2936–2940.

[18] L. Ying, S. Shakkottai, A. Reddy, and S. Liu, "On combining shortest-path and back-pressure routing over multihop wireless networks," *IEEE/ACM Transactions on Networking*, vol. 19, no. 3, pp. 841–854, 2010.

[19] H. Xiong, R. Li, A. Eryilmaz, and E. Ekici, "Delay-aware cross-layer design for network utility maximization in multi-hop networks," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 5, pp. 951–959, 2011.

[20] G. Papaioannou and I. Koutsopoulos, "Tile-based caching optimization for 360 videos," in *Proceedings of the Twentieth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, 2019, pp. 171–180.

[21] S. M. Kay, *Fundamentals of statistical signal processing*. Prentice Hall PTR, 1993.

[22] W. A. Fuller, *Introduction to statistical time series*. John Wiley & Sons, 2009, vol. 428.

[23] M. J. Neely, E. Modiano, and C.-P. Li, "Fairness and optimal stochastic control for heterogeneous networks," *IEEE/ACM Transactions On Networking*, vol. 16, no. 2, pp. 396–409, 2008.

[24] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," in *29th IEEE Conference on Decision and Control*. IEEE, 1990, pp. 2130–2132.