Motion-Prediction-based Wireless Scheduling for Interactive Panoramic Scene Delivery

Jiangong Chen, *Student Member, IEEE,* Xudong Qin, *Student Member, IEEE,* Guangyu Zhu, Bo Ji, *Senior Member, IEEE,* and Bin Li, *Senior Member, IEEE*

Abstract—Mobile Virtual Reality (VR) and panoramic video streaming rely on interactive panoramic scene delivery to provide desirable user experiences. However, it is pretty challenging to support multiple users via the wireless network since a panoramic scene typically consumes $4 \sim 6\times$ bandwidth compared with a regular video with the same resolution. Motivated by the fact that users only perceive the Field-of-View (FoV), we employ the autoregressive process to predict the user's motion and stream only part of the panoramic content. Notably, we analytically characterize the effect of the delivered portion on the user's successful viewing probability. Then, we formulate an optimization problem to maximize the application-level throughput (which measures the average rate for successful viewing the desired content instead of raw network throughput) while providing a regular service. In addition, we impose three main constraints to our problem: minimum required service rate, maximum allowable energy consumption, and wireless interference. We then propose a novel scheduling algorithm that incorporates users' successful viewing probabilities and asymptotically maximizes application-level throughput while providing service regularity guarantees. We conduct real-trace simulations to evaluate the efficiency of our algorithm.

Index Terms—Panoramic video streaming, Virtual Reality, wireless scheduling, stochastic network optimization

1 Introduction

The development of network technology and emerging wireless Head-Mounted Displays (HMDs) (such as Oculus Quest and HTC VIVE) make Virtual Reality (VR) and panoramic video streaming more and more popular. The immersive experience provided by panoramic scenes is quite attractive in education/training, social networking, and entertainment, to name a few. For instance, students can visit museums or landscapes by wearing a VR headset in the classroom as long as we have the virtual model for that place (e.g., [1]). To provide the best immersive experience, i.e., the users hardly differentiate between virtual scenes and the real world, the system must provide high throughput and seamless experience (i.e., regular service) to each user. However, both VR and panoramic video streaming rely on panoramic scene delivery, which typically consumes $4 \sim 6\times$ bandwidth compared with regular 2-dimensional (2D) images with the same resolution (see, e.g., [2], [3]). Besides, unlike traditional video streaming, the interactivity requirement on panoramic scene delivery makes it impossible to prefetch and cache the panoramic content for several seconds, further complicating the problem by adding a harsh delivery time restriction for each panoramic image. Moreover, under wireless interference, only a subset of users can be allowed to transmit the content simultaneously, which poses a significant challenge to the scheduling design.

- *Jiangong Chen, Xudong Qin, and Bin Li are with the Department of Electrical Engineering, The Pennsylvania State University, State College, PA 16801, USA. E-mail: {jiangong, xfq5024, binli}@psu.edu; Guangyu Zhu is with the Department of Computer Science and Statistics, University of Rhode Island, Kingston, RI 02881, USA. E-mail: guangyuzhu@uri.edu; Bo Ji is with the Department of Computer Science, Virginia Tech, Blacksburg, VA 24061, USA. E-mail: boji@vt.edu.*
- *This work has been supported in part by NSF under the grants CNS-2152658 and CNS-2152610.*

Fortunately, the Field of View (FoV) is sufficient for the user's visual perception, which only occupies about 20% of the whole panoramic scene. For instance, when a user is playing a collaborative VR drawing game, she will likely focus on the view in front of herself for a short time. This observation makes it possible to save about 80% of the consumed bandwidth under the wireless network if we could perfectly predict a user's motion. However, the prediction always incurs an error. As such, the central question is how to incorporate the error-prone motion prediction into the scheduling algorithm design.

Motion prediction-based approach has been explored by many recent works (e.g., [2], [4], [5], [6]). They utilized various learning mechanisms to improve the prediction accuracy and successfully incorporated them into the algorithm design. However, they did not consider the multi-user setting with wireless interference, where only a subset of users can be scheduled to transmit at each time. Although some other works (e.g., [7], [8]) made some progress on the multi-user scheduling design, they haven't explicitly considered the server regularity, which measures how often the user sees the desired content. Such a metric is important since irregular video streaming services easily cause dizziness.

In this paper, we predict each user's motion by using the autoregressive process and incorporate it into the scheduling design by analytically characterizing the successful viewing probability as a function of the delivered portion surrounding the predicted FoV of the user. In accordance with the panoramic scene delivery setup, we focus on whether each user can view their desired content instead of receiving raw service rates as high as possible. Particularly, we use network-level throughput to denote the raw service rates and application-level throughput to denote the average number of times a user successfully views her desired content. Note that a higher network-level through-

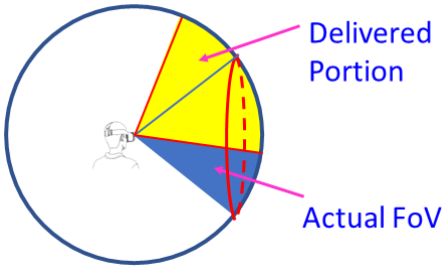


Fig. 1: Example of mismatch between delivered portion and the user's FoV.

put does not necessarily result in higher application-level throughput. For example, our experimental results (using the dataset from [2]) show that most users successfully view the desired content when delivering a half-sphere scene since it is sufficient to cover the actual FoV perceived by the users. As a result, delivering the whole panoramic scene has a nearly identical application-level throughput as delivering half of the scene but yields twice the network-level throughput.

Evaluating Quality of Experience (QoE) can often pose challenges in certain scenarios. Generally, a QoE model seeks to deduce the perceived quality a user experiences during interactions with a specific application. Traditional QoE modeling methods hinge on manually engineered features, including standard quality metrics such as image quality and events of video rebuffering. However, since our problem formulation does not encompass these particular parameters—image quality and rebuffering—we are necessitated to identify an alternative metric to gauge the user's QoE. In the context of panoramic content delivery, it's imperative to recognize that the area dispatched to the user might not precisely coincide with the user's Field of View (FoV), as depicted in Fig. 1. This mismatch can result in parts of the user's FoV being rendered as a blank, black space, undeniably diminishing the overall experience. Consequently, we've opted to use application-level throughput as an indicative measure for QoE. This metric gauges the likelihood that the content delivered fully encompasses the user's FoV. While it might seem logical to calculate the proportion of the covered area relative to the FoV—especially when the FoV is only partially covered—we, for the sake of modeling simplicity, have restricted our measure to a binary assessment: determining if the user's FoV is entirely covered or not.

Although we focused on pursuing as high application-level throughput as possible instead of network-level throughput, we need to clarify that a minimal network-level throughput is required to support our objective. For instance, if the network-level throughput is even lower than the required bandwidth to deliver 20% of panoramic images, we cannot reach any application-level throughput. Therefore, we need to set a threshold to limit the minimum average allocated transmission rate. Moreover, the threshold varies between users since they may experience diverse panoramic contents and thus have various network-level throughput requirements. Considering that each downloading of the panoramic content will consume a certain amount of energy on the client-side, we also set a maximum allowable energy consumption to avoid overheating the client device. Note that different devices naturally have different energy efficiency, and thus the energy threshold is also user-specific. Meanwhile, we will never have infinite network resources in practical scenarios, where many factors like wireless interference will always limit available network resources. Considering all those constraints, we require a well-designed wireless scheduling algorithm to maximize application-level throughput.

In this paper, we aim to maximize application-level throughput (i.e., rate of average successful views) while meeting both network-level throughput and energy consumption requirements as well as providing regular service guarantees for each user. Specifically, we propose a non-convex objective function that incorporates the successful viewing probability. We introduce Time-Since-Last-Service (TSLS) counter (see [9]) to track the service regularity. We adopt the stochastic network optimization framework (see [10]) and design a non-standard Lyapunov function to incorporate the TSLS counter into the scheduling design. The main contributions of this paper are listed as follows:

- We analytically characterize the successful viewing probability as a function of the delivered portion of the panoramic scenes by assuming that the prediction error follows a Gaussian distribution.
- We propose a novel concept of application-level throughput under the panoramic scene delivery setup and show that it is different from the traditional network-level throughput by a motivating example.
- We formulate the multi-user scheduling for interactive panoramic scene delivery as a stochastic network optimization problem. We aim to maximize the application-level throughput while satisfying the minimum required network-level throughput, maximum allowable energy consumption, and wireless interference constraints.
- We develop a motion-prediction-based scheduling algorithm that explicitly incorporates the motion prediction into the scheduling decision and shows that it asymptotically optimizes the application-level throughput and provides regular service guarantees.
- We conduct real-trace simulations to demonstrate the superior performance of our algorithm compared with the heuristic baseline. We consider users' motion traces in different applications including the panoramic scene gallery (see [2]) and VR touring (see [11]).

While this paper is built upon our prior work [12], which was published in INFOCOM 2021, we have the following new contributions: (1) we consider the energy consumption limitation of mobile users; (2) we extend the user motion model and accounted for both virtual reality and panoramic videos; (3) we add a trace-based simulation for VR touring application; (4) we propose a heuristic baseline approach and demonstrate the superiority of our algorithm; (5) we adopt the pick and compare method to significantly reduce the computational complexity of the proposed algorithm.

The remainder of this paper is organized as follows: Section 2 introduces the system model and problem formulation. Section 3 provides a motivating example and illustrates the impact of the scheduling design on both application-

level throughput and service regularity performance. Section 4 introduces our motion-prediction-based scheduling algorithm and studies its performance. Section 5 presents simulation results using the real datasets of users' motion traces. Section 6 shows the derivation of the successful viewing probability. Section 7 provides the detailed proof of our algorithm. Section 8 reviews related work, and Section 9 concludes this paper.

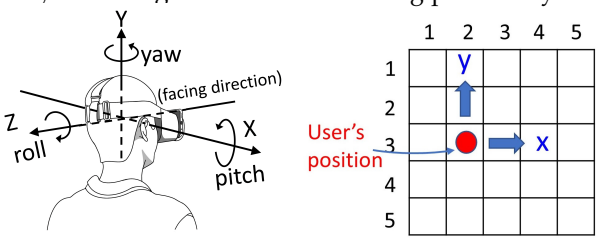
2 SYSTEM MODEL

We consider a system with N users, each of which downloads its desired panoramic content from a wireless access point (AP). We assume the system operates in a slotted manner. Each user requests a series of panoramic scenes, and we need to deliver each panoramic scene within a fixed time interval. We note that only a part of panoramic content (around $20\% - 25\%$) can be seen by each user, known as *Field of View (FoV)*. Thus, if we have a perfect prediction of the user's motion, we can save up to 80% of the bandwidth by delivering only the FoV. However, the prediction is error-prone such that we need to deliver a larger portion than the FoV to tolerate the prediction error. To facilitate our mathematical modeling and algorithm developments, we unify the panoramic scene size and wireless transmission rate units 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 $S_n[t] \in \mathcal{R} \triangleq \{0, R_1, R_2, \ldots, R_M\}$, $0 < R_1 < R_2 \cdots < R_M \triangleq 1$, R_1 is the minimum required rate that delivers only the FoV, and R_M refers to the required rate when delivering a whole scene. The discrete manner lies in the fact that we can split each panoramic scene into a finite number of tiles with the same resolution and transmit a subset of tiles to users (e.g., [3], [13]). We will choose a set of tiles around the predicted viewport (center of the FoV) in each time slot based on the allocated transmission rate.

In order to streamline the modeling process, this study limits its examination to setups in which the scheduling rate is proportionate to the percentage of transmitted panoramic content. However, it's important to acknowledge that in real-world scenarios, where factors such as the tiling scheme, the projection from panoramic content to regular images, and video encoding are taken into account, the problem grows in complexity. To encode a panoramic video, an initial step involves the reprojection of panoramic content onto a rectangular texture. Typically, the projection across the panoramic content is non-uniform. This implies that identical-sized areas within the panoramic view can yield a varied number of pixels when mapped onto the rectangular texture. Subsequent to this reprojection, the texture undergoes encoding through conventional video encoding mechanisms, such as H264. It's pivotal to acknowledge that the compression rates are content-dependent. An all-black image, for instance, may exhibit a vastly superior compression ratio compared to a texture teeming with intricate objects. Consequently, forging a precise correspondence between the size of the delivered area and the scheduling rate accounting for the video encoding becomes an endeavor demanding profound expertise and thorough investigation. As it extends beyond the scope of our current research, we have identified it as a substantial avenue for future exploration.

The wireless interference further complicates the problem by limiting the number of active users in each time slot, i.e., the simultaneous downloading of the panoramic content. Therefore, the AP is expected to deliver content to a subset of users and allocate the proper transmission rates $S_n[t], \forall n = 1, 2, \ldots, N$. 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 panoramic scene 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.



(a) Rotation coordinates. (b) VR grid world.

Fig. 2: User movement.

In order to calculate the successful viewing probability for each user, we discuss two different scenarios: panoramic scene gallery with three *Degree of Freedoms* (DoF) on head rotation and VR touring with six DoFs on both head rotation and translation in the virtual world. We first introduce 3-D rotation angles to capture a user's head motion. As shown in Fig. 2a, a user could rotate 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 scenes, we focus on *pitch* and *yaw* axes as in [3], 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 [2]), we predict them separately based on the autoregressive model (see [14]). While there are many machine learning-based prediction algorithms explored in existing works (e.g., [2]), we adopt the autoregressive model here since it makes online real-time predictions and can quickly adapt to changing

Symbol	Description	Symbol	Description
N	number of users	n	index of user
L	length of time horizon	t	index of time slot
$\mathcal{R}$	set of transmission rates	$\mathbf{S}[t]$	feasible rate vector in time slot t
$C[t]$	global channel state in time slot t	$S_n[t]$	allocated transmission rate for user n
$\mathcal{C}$	set of global channel states	ϕ_c	probability that the channel state is c
$\mathcal{S}^{(c)}$	set of all feasible rate vectors when the channel state is c	$I_n(S_n[t])$	indicator function of whether user n successfully views its desired content
$\delta_n(S_n[t])$	successful viewing probability for user n in time slot t given its transmission rate $S_n[t]$	r_n	minimum required allocated transmission rate for user n on average
$\mathbb{1}_{\{\cdot\}}$	indicator function	w_n	weight of user n in the objective function
e_n	energy constraint for user n	$T_n[t]$	TSLS counter for user n in time slot t
$Q_n^{(S)}[t]$	length of the virtual queue for user n that measures the degree of violation of the average service rate	$Q_n^{(E)}[t]$	length of the virtual queue for user n that measures the degree of violation of the average energy consumption constraint
$V[t]$	Lyapunov function	K	hyperparameters for the utility function
η	hyperparameters for the regular service	U^*	optimal value of the optimization problem
U_n^*	successful viewing probability of user n under optimal policy		

TABLE 1: Notations for system model.

panoramic contents and wireless environment. On the other hand, when a user is experiencing an interactive VR tour, besides the head rotation, she is also able to move freely in a virtual world. As shown in Fig. 2b, we assume the user translates in a 2D grid world where each grid corresponds to a unique panoramic scene. Similar to the head rotation, we independently predict the movement in those two axes using the autoregressive model.

We assume that the prediction errors of both pitch and yaw angles of user n follow the normal distribution with standard deviation σ_n^X and σ_n^Y, respectively. This is motivated by the fact that under the autoregressive model, the distribution of the prediction error converges to the normal distribution as the number of data samples goes to infinity (see [15, Theorem 8.2.1]). In contrast, we characterize the successful prediction probability on the position as a constant since it is independent of the scheduling rate $S_n[t]$. As we know the prediction results of previous slots for both rotation and translation, it is convenient to estimate the prediction probability on position by calculating the running average during the experiments. In Section 6, we show that the successful viewing probability of user n can be expressed as follows:

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

where δ_n^p is the probability that the prediction on the position is successful, $\text{erf}(x) \triangleq \frac{2}{\sqrt{\pi}}\int_0^x e^{-y^2}dy$ 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 Section 6. Fig. 3 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 [2]) and obtain their standard deviations of prediction errors of both pitch and yaw angles under the autoregressive model. We can observe from Fig. 3 that a larger standard deviation of the angle prediction error requires a larger allocated transmission rate to keep the same successful viewing probability.

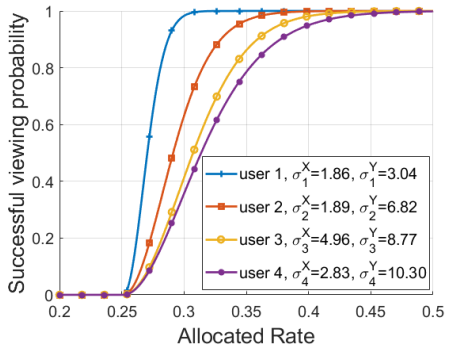


Fig. 3: Successful viewing probability.

In this paper, we would like to develop a scheduling algorithm to optimize both *application-level throughput* (defined as the weighted average of the 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 and the average energy consumption should not exceed a specific threshold as well as wireless interference constraints, i.e.,

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

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

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

$$\lim_{L\to\infty} \frac{1}{L}\sum_{t=0}^{L-1} \mathbb{E}[\mathbb{1}_{\{S_n[t]>0\}}] \leq e_n, \forall n, \tag{5}$$

where the objective function is the weighted sum of the

application-level throughput, $w_n > 0$ is the weight of user n, $r_n > 0$ is the minimum required allocated transmission rate for user n on average, e_n refers to the average number of transmission times for user n, and $\mathbb{1}_{\{\cdot\}}$ is an indicator function. We consider a simplified energy model, where each download of the panoramic content will consume a certain amount of energy to activate the wireless transmission modules. Therefore, the constraint on the transmission times corresponds to the energy consumption constraint. 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 [9], 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:

$$T_n[t+1] \triangleq \begin{cases} 0, & \text{if } I_n(S_n[t]) = 1; \\ T_n[t]+1, & \text{otherwise.} \end{cases} \tag{6}$$

It has been shown in [9] 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:

$$\lim_{L\to\infty} \frac{1}{L} \sum_{t=0}^{L-1} \mathbb{E}\left[T_n[t]\right]. \tag{7}$$

To facilitate the readers, we list the notations used in our system model in TABLE 1. Next, we will 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.

3 A Motivating Example

In this section, we provide numerical examples of multi-user interactive panoramic scene delivery which motivates our design. We observe a significantly different performance on application-level throughput and service regularity with different scheduling algorithms. We consider $N = 4$ users and limit the total service rates by 1 in each time slot, where all users will share the total rates and have the same weight 1. We employ two different Round-Robin (RR) scheduling algorithms and list the allocated service rate for each user in Table 2-(a). Notably, we use a blue font to denote the first RR algorithm that serves each user with the rate of one in turn. The red font refers to the second RR algorithm that provides the first two users with the rate of 0.5 in even time slots and the other two users in odd time slots.

Based on the given rules, we can calculate the average service rate by $\lim_{L\to\infty} \frac{1}{L}\sum_{t=0}^{L-1} S_n[t]$, $\forall n = 1,2,3,4$. Unsurprisingly, both RR algorithms result in the save average rate of 0.25. However, they yield different application-level throughput performance. As shown in Fig. 3, $S_n[t] = 0.5$ is sufficient for most of users to see the desired content with traces in [2]. As such, we let $\delta_n(0.5) = \delta_n(1) = 1$, $\forall n$. Then the viewing results of each user in each time slot are shown in Table 2-(b). The viewing result is 1 if the user successfully sees her desired content, 0, otherwise. Hence, the second RR algorithm yields the application-level throughput of 2, while the first RR algorithm yields only 1.

$S_n[t]$ 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.

result 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) Viewing results of each user in each time slot.

TABLE 2: Service rate and viewing results under two different versions of RR algorithms: the results are colored by blue and red, respectively.

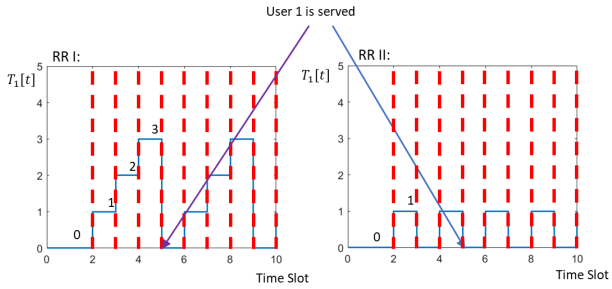


Fig. 4: Example of TSLS dynamics of user 1

Fig. 4 shows the evolution of user 1's TSLS counter under two RR algorithms, while other users will have similar patterns. We can calculate that the first and the second RR algorithms have the average TSLS for each user of 1.5 and 0.5, respectively. To summarize, the second RR algorithm is twice better than the first version in the application-level throughput and three times better in average TSLS.

The above example demonstrates the significant impact of the scheduling design on both application-level throughput and service regularity performance. We should explore more complex scheduling decisions once we can select the transmission rates from a discrete space. In the next section, we will develop an efficient scheduling algorithm that yields good application-level throughput and service regularity performance under various constraints mentioned before.

4 Algorithm Design and Analysis

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

We use the stochastic network optimization framework (e.g., [10]) to introduce two virtual queues for each user that measures the degree of violation of the average service rate constraint and the average energy consumption constraint, respectively. We use $Q_n^{(S)}[t]$ and $Q_n^{(E)}[t]$ to denote the length of two virtual queues for user n in time slot t. Particularly, the amount of traffic entering the first virtual queue n in time slot t is r_n, while the amount of service for the first virtual queue n in time slot t is $S_n[t]$. Meanwhile, $\mathbb{1}_{\{S_n[t]>0\}}$ is the amount of traffic entering the second virtual queue for user n in time slot t, while e_n is the amount of service for the second virtual queue. Then, the evolution of those two virtual queues for user n can be described as follows:

$$Q_n^{(S)}[t+1] \triangleq \left(Q_n^{(S)}[t] + r_n - S_n[t]\right)^+, \forall n, \forall t, \quad (8)$$

$$Q_n^{(E)}[t+1] \triangleq \left(Q_n^{(E)}[t] + \mathbb{1}_{\{S_n[t]>0\}} - e_n\right)^+, \forall n, \forall t. \quad (9)$$

where $(x)^+ = \max\{x, 0\}$. We say that virtual queue n for service rate is *mean rate stable* (see [10]) if $\lim_{t\to\infty} \frac{\mathbb{E}[Q_n^{(S)}[t]]}{t} = 0$. If the virtual queue n for service rate is mean rate stable, then the average service rate of user n is at least r_n (see [10, Theorem 2.5]). Similarly, the virtual queue n for energy consumption is mean rate stable if $\lim_{t\to\infty} \frac{\mathbb{E}[Q_n^{(E)}[t]]}{t} = 0$. Then, the average energy consumption of user n is not greater than e_n. The following algorithm is derived by minimizing the drift of the Lyapunov function

$$V[t] = \frac{1}{2}\sum_{n=1}^{N} Q_n^{(S)2}[t] + \frac{1}{2}\sum_{n=1}^{N} Q_n^{(E)2}[t] + \eta \sum_{n=1}^{N} T_n[t] \quad (10)$$

minus 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.,

$$V[t+1] - V[t] - K\sum_{n=1}^{N} w_n \mathbb{E}[\delta_n(S_n[t])]. \quad (11)$$

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:

Our algorithm has two major components: i) AR-based motion prediction, and ii) wireless scheduling. In each time slot t, we use the autoregressive model for the pitch and yaw angle prediction and updates the prediction coefficients based on the Yule-Walker equation (see [14]). 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., [16]). 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 [9]. 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., [17]), our proposed MPS algorithm also has a high computational complexity (which could be exponential) in general. To reduce the computational complexity, we adopt an evolutionary randomized algorithm [18], named the pick and compare method, which is also evaluated in Section 5.

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

Algorithm 1 Motion-Prediction-based Scheduling (MPS)

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

Autoregressive process-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 autoregressive model:

$$\widehat{X}_n[t] = -\sum_{k=1}^{W} a_n[k] X_n[t-k] \quad (12)$$

$$and \quad \widehat{Y}_n[t] = -\sum_{k=1}^{W} b_n[k] Y_n[t-k], \quad (13)$$

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 [14]). Similar approach is used to predict the position for VR applications.

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

$$\mathbf{S}^*[t] \in \arg\max_{\mathbf{S}\in\mathcal{S}^{(c)}} \sum_{n=1}^{N} \Big(S_n[t] Q_n^{(S)}[t] - \mathbb{1}_{\{S_n[t]>0\}} Q_n^{(E)}[t] + (\eta T_n[t] + K w_n)\delta_n(S_n[t]) \Big), \quad (14)$$

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

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:*

$$\lim_{L\to\infty}\frac{1}{L}\sum_{t=0}^{L-1}\sum_{n=1}^{N}U_n^*\cdot\mathbb{E}[T_n[t]]\leq\frac{B(\eta)+KNw_{\max}}{\eta},\quad(15)$$

$$\lim_{L\to\infty}\frac{1}{L}\sum_{t=0}^{L-1}\sum_{n=1}^{N}\mathbb{E}[w_n\delta_n(S_n[t])]\geq U^*-\frac{B(\eta)}{K},\quad(16)$$

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., $U^*\triangleq\sum_{n=1}^{N}w_nU_n^*$ is the optimal value of the optimization problem (2)-(4)*).*

Proof. The proof follows from the stochastic network optimization framework and can be found in Section 7. □

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.

5 Simulations

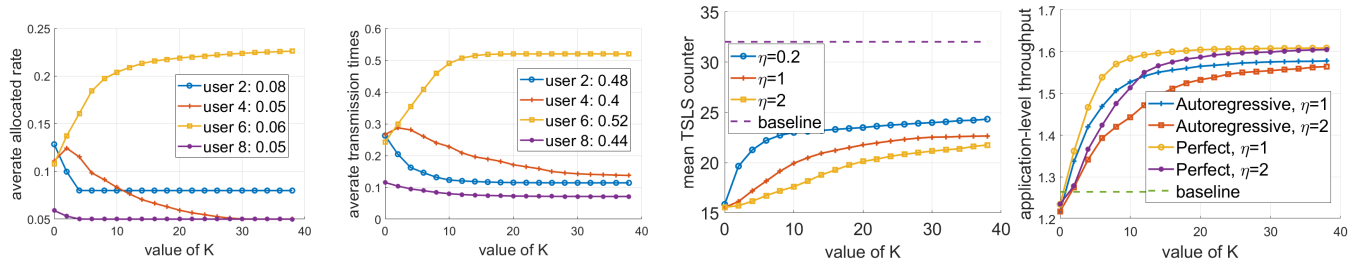
In this section, we perform simulations to evaluate the efficiency of our proposed MPS algorithm. We consider $N=8$ users. Each user experiences i.i.d. ON-OFF channel fading over time with probability p_n that its channel is ON in each time slot. We assume that at most two users can be scheduled in each 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. We also list the energy limitation e_n for each user as in (5). The allocated transmission rate can be selected from the set $\mathcal{R}=\{0,0.3,0.4,0.5,0.7,1\}$. We consider two different simulation setups for the panoramic scene gallery and VR touring. In both simulation setups, the autoregressive model is used to predict the user's motion. Specifically, we predict both translation and rotation in VR touring, while we only predict the head rotation in panoramic scene gallery. The detailed simulation parameters are available in TABLE 3. Note that we use the same hyperparameters for two different setups to simulate the identical environments. In addition, we use synthetic head motion data generated from the dataset in [2] (panoramic scene gallery) and [11] (VR touring) for each user.

To demonstrate the superiority of our proposed algorithm, we introduce a heuristic algorithm as the baseline and evaluate its performance. The baseline algorithm proceeds in three steps: first, the energy constraint is applied to filter the available users in each time slot, whereby those who violate the constraint are set inactive; second, if any available users violate the rate requirement constraint, we designate the users who do not violate the constraint as inactive; finally, we select the available users with the highest utility function under the capacity region. While this algorithm ensures compliance with both constraints, it may not achieve optimal application-level throughput and service regularity.

	user 1	**user 2**	**user 3**	**user 4**
Required rate r_n	0.1	0.08	0.11	0.05
Energy lim. e_n	0.64	0.48	0.46	0.4
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
Energy lim. e_n	0.45	0.52	0.54	0.44
Weight w_n	0.9	1.2	0.3	0.2
Fading prob. p_n	0.8	0.9	0.7	0.8

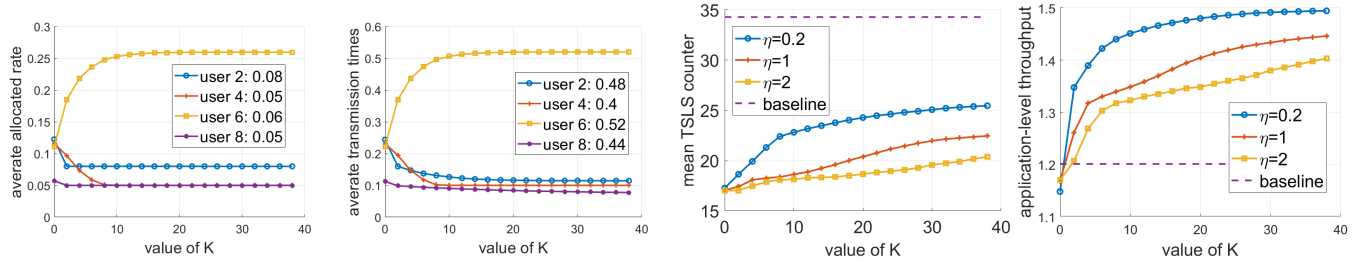
TABLE 3: Simulation setup.

The simulation results for panoramic scene gallery and VR touring are shown in Fig. 5 and Fig. 6, respectively. Fig. 5a shows the average allocated rates of four different users with respect to parameter K when $\eta=1$. We can observe from Fig. 5a that our proposed MPS algorithm guarantees the minimum service rate required by each user. Also, Fig. 5b shows the average transmission times corresponding to Fig. 5a, while each user strictly satisfies the energy limitation constraint. Fig. 5c and Fig. 5d show an impact of the parameter K on the performance of our proposed MPS algorithm. We can observe from Fig. 5c and Fig. 5d 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. To evaluate the effect of motion prediction, we also conduct the simulation with perfect prediction. In such a case, $\delta_n(S_n[t])=1$ when $S_n[t]\geq R_1$. Recall that R_1 corresponds to the rate to deliver the FoV of a panoramic scene. $\delta_n(S_n[t])=0$, otherwise. As shown in Fig. 5d, we have a larger average application-level throughput under the same η compared with the results using autoregressive model since we have a larger $\delta_n(S_n[t])$ given $S_n[t]$. Furthermore, our algorithm outperforms baseline algorithms in terms of service regularity (measured by TSLS counter) across all hyperparameters, with a minimum improvement of 24%. In addition, our algorithm achieves at least 14%



(a) Average rate for each user: $\eta = 1$ (b) Average transmission times for each user: $\eta = 1$ (c) Average TSLS (d) Application-level throughput (v.s. perfect prediction)

Fig. 5: Performance of the MPS algorithm (panoramic scene gallery). Note that the numbers in the legends of Fig. 5a and Fig. 5b correspond to the required rate r_n and energy limitation e_n in TABLE 3, respectively.

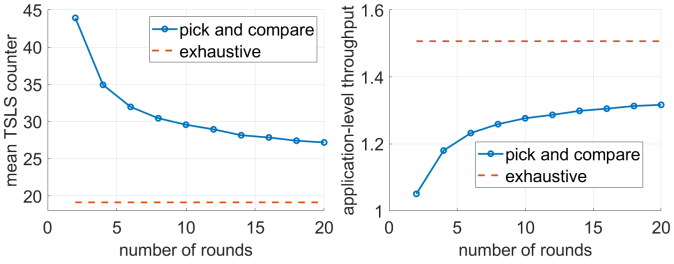


(a) Average rate for each user: $\eta = 0.2$ (b) Average transmission times for each user: $\eta = 0.2$ (c) Average TSLS (d) Application-level throughput

Fig. 6: Performance of the MPS algorithm (VR touring). Note that the numbers in the legends of Fig. 6a and Fig. 6b correspond to the required rate r_n and energy limitation e_n in TABLE 3, respectively.

higher application-level throughput when K is greater than 10. We can observe a similar phenomenon in the VR touring dataset in Fig. 6. Since it is harder to predict the movement on both translation and rotation, we will get a lower successful viewing probability $\delta_n(S_n[t])$ given $S_n[t]$, resulting in different performance metrics.

As previously discussed, our proposed MPS algorithm has a high computational complexity, which can be exponential in some cases. To address this issue, we have adopted an evolutionary randomized algorithm, called the pick and compare algorithm. We randomly select users and allocate rates in each time slot, calculating the weight using our MPS algorithm. Next, we compare the weight of the new allocation with the previous one. If the new allocation has a greater weight, we adopt it; otherwise, we continue to use the previous allocation. If the previous users are not available in the current time slot, we adopt the new allocation. This process is repeated for a specific number of rounds in each time slot to yield the final results. It is clear that with more repetitive rounds, the performance of the pick and compare algorithm becomes closer to the original exhaustive approach, but this results in higher computational complexity. We evaluate the performance of this approach using varying numbers of iteration rounds. As shown in Fig. 7, the service regularity and application-level throughput both converge to the exhaustive approach as the number of rounds increases. Moreover, the pick and compare method reduces the potentially exponential computation complexity of the original MPS algorithm to linear complexity. For example, in our simulation setup, we need to iterate over 912 steps with the exhaustive approach in each time slot, while the number of iterations ranges from only 2 to 20 in the pick and compare method. We present the results for the hyperparameters $\eta = 1$ and $K = 8$ under the panoramic scene gallery setup, as the other parameters and VR touring have produced similar results.



(a) Average TSLS. (b) Throughput.

Fig. 7: Performance of the pick and compare method.

6 Proof of Successful Viewing Probability

In this section, we characterize the successful viewing probability $\delta_n(S_n[t])$ for each user given the standard deviation of the prediction error. We assume that the prediction on orientation and position are independent. Therefore, the successful viewing probability is simply the product of the successful prediction probability on orientation and position, i.e., $\delta_n(S_n[t]) = \delta_n^o(S_n[t]) \cdot \delta_n^p$. We first calculate the successful probability of the orientation prediction $\delta_n^o(S_n[t]; \sigma_n^X, \sigma_n^Y)$ 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. 8, 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. 9 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. 8a and Fig. 8b. 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:

$$\theta_0 = \text{diag}(\alpha_0, \beta_0), \tag{17}$$

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

$$\text{diag}(\alpha, \beta) \triangleq 2\arccos\left[\frac{1}{\sqrt{1+\tan^2(\frac{\alpha}{2})+\tan^2(\frac{\beta}{2})}}\right], \tag{18}$$

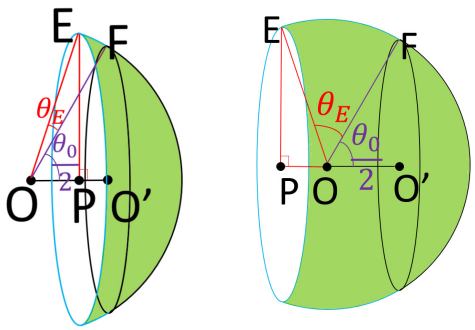
where $\alpha \leq \pi$ and $\beta \leq \pi$. Indeed, considering the right triangles $\triangle OO'Q$ and $\triangle OO'R$ in Fig. 9, we have

$$O'Q = OO'\tan(\alpha_0/2), O'R = OO'\tan(\beta_0/2). \tag{19}$$

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

$$OO' = \frac{1}{\sqrt{1+\tan^2(\alpha_0/2)+\tan^2(\beta_0/2)}}. \tag{20}$$

This, together with $\cos(\theta_0/2) = OO'$, implies (18).



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

Fig. 8: The delivered content.

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 $S_n[t]$:

$$\theta_0 + 2\theta_E = 2\arccos(1 - 2S_n[t]). \tag{21}$$

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. 8a, $h = R - OP$, in Fig. 8b, $h = R + OP$. Yet in both cases, $h = R - \cos(\theta_E + \theta_0/2)$. This, together with $R = 1$ and $h = 2S_n[t]$, implies (21).

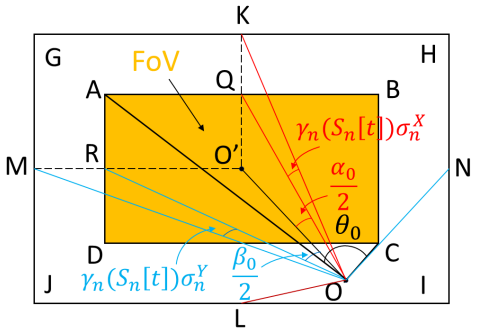


Fig. 9: FoV and various angles.

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. 9) and horizontal angle (e.g., $\angle MON$ in Fig. 9) 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. 8a, 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. 8b, 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.

$$Ang(\gamma(S_n[t])) = 2\arccos(1 - 2S_n[t]). \tag{22}$$

Let $\widehat{X}_n[t]$ and $\widehat{Y}_n[t]$ be the predicted angles in the pitch and yaw axis, respectively, under the autoregressive process. Hence, given $S_n[t]$, the FoV of the user can be covered 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 orientation prediction successful probability $\delta_n(S_n[t])$ can be calculated:

$$\begin{aligned}\delta_n^o(S_n[t]) &= \Pr\{\mathcal{A}_X \cap \mathcal{A}_Y\} \\ &= \Pr\{\mathcal{A}_X\}\Pr\{\mathcal{A}_Y\} \\ &= \text{erf}^2\left(\frac{\gamma_n(S_n[t])}{\sqrt{2}}\right).\end{aligned} \tag{23}$$

Since the position prediction probability is independent of the scheduling rate, we characterize it as a constant δ_n^p. Although we cannot know the exact value of the constant in prior, it can be estimated during the experiments using approaches like running average. As for those scenarios with only 3DoF traces, we set $\delta_n^p = 1$.

In a word, we can calculate the successful viewing probability $\delta_n(S_n[t])$ as follows:

$$\delta_n(S_n[t]) = \delta_n^o(S_n[t]) \cdot \delta_n^p = \delta_n^p \text{erf}^2\left(\frac{\gamma_n(S_n[t])}{\sqrt{2}}\right) \tag{24}$$

7 Proof of application-level throughput optimality

We select the Lyapunov function

$$V[t]=\frac{1}{2}\sum_{n=1}^{N}\left(Q_n^{(S)}[t]\right)^2+\frac{1}{2}\sum_{n=1}^{N}\left(Q_n^{(E)}[t]\right)^2+\eta\sum_{n=1}^{N}T_n[t]. \tag{25}$$

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

$$\begin{aligned}
&\Delta V[t]\triangleq\mathbb{E}\left[V[t+1]-V[t]|\mathbf{H}[t]\right]\\
=&\mathbb{E}\Big[\frac{1}{2}\sum_{n=1}^{N}\left((Q_n^{(S)}[t+1])^2-(Q_n^{(S)}[t])^2\right)\\
&+\frac{1}{2}\sum_{n=1}^{N}((Q_n^{(E)}[t+1])^2-(Q_n^{(E)}[t])^2)\\
&+\eta\sum_{n=1}^{N}(T_n[t+1]-T_n[t])\Big|\mathbf{H}[t]\Big]\\
\overset{(a)}{\leq}&\frac{1}{2}\sum_{n=1}^{N}\mathbb{E}\left[(Q_n^{(S)}[t]+r_n-S_n^*[t])^2-(Q_n^{(S)}[t])^2\Big|\mathbf{H}[t]\right]\\
&+\frac{1}{2}\sum_{n=1}^{N}\mathbb{E}\left[(Q_n^{(E)}[t]+\mathbb{1}_{\{S_n^*[t]>0\}}-e_n)^2-(Q_n^{(E)}[t])^2\Big|\mathbf{H}[t]\right]\\
&+\eta\sum_{n=1}^{N}\mathbb{E}\left[(T_n[t]+1)(1-I_n(S_n^*[t]))-T_n[t]|\mathbf{H}[t]\right]\\
\overset{(b)}{\leq}&\sum_{n=1}^{N}\mathbb{E}\Big[Q_n^{(S)}[t](r_n-S_n^*[t])+Q_n^{(E)}[t](\mathbb{1}_{\{S_n^*[t]>0\}}-e_n)\\
&-\eta T_n[t]I_n(S_n^*[t])\Big|\mathbf{H}[t]\Big]+B(\eta),
\end{aligned}\tag{26}$$

where step (a) follows from the dynamics of $Q_n^{(S)}[t]$ (cf. (8)), $Q_n^{(E)}[t]$ (cf. (9)), and $T_n[t+1]$ (cf. (6)); (b) is true for

$$B(\eta)\triangleq\sum_{n=1}^{N}\left(r_n^2+e_n^2+R_M^2+1\right)/2+\eta N. \tag{27}$$

Recall that R_M is the maximum transmission rate, i.e., $S_n^*[t]\leq R_M$.

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

$$\begin{aligned}
&\Delta V[t]-K\sum_{n=1}^{N}w_n\mathbb{E}[\delta_n(S_n^*[t])|\mathbf{H}[t]]\\
\leq&\sum_{n=1}^{N}Q_n^{(S)}[t]r_n-\sum_{n=1}^{N}Q_n^{(E)}[t]e_n+B(\eta)\\
&-\sum_{n=1}^{N}Q_n^{(S)}[t]\mathbb{E}\left[S_n^*[t]|\mathbf{H}[t]\right]\\
&+\sum_{n=1}^{N}Q_n^{(E)}[t]\mathbb{E}\left[\mathbb{1}_{\{S_n^*[t]>0\}}\Big|\mathbf{H}[t]\right]\\
&-\sum_{n=1}^{N}(\eta T_n[t]+Kw_n)\mathbb{E}[\delta_n(S_n^*[t])|\mathbf{H}[t]]\\
\leq&\sum_{n=1}^{N}Q_n^{(S)}[t]r_n-\sum_{n=1}^{N}Q_n^{(E)}[t]e_n+B(\eta)\\
&-\sum_{n=1}^{N}Q_n^{(S)}[t]\mathbb{E}\left[\widehat{S}_n[t]\Big|\mathbf{H}[t]\right]\\
&+\sum_{n=1}^{N}Q_n^{(E)}[t]\mathbb{E}\left[\mathbb{1}_{\{\widehat{S}_n[t]>0\}}\Big|\mathbf{H}[t]\right]\\
&-\sum_{n=1}^{N}(\eta T_n[t]+Kw_n)\mathbb{E}[\delta_n(\widehat{S}_n[t])|\mathbf{H}[t]],
\end{aligned}\tag{28}$$

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

$$\mathbb{E}[\widehat{S}_n[t]]\geq r_n,\forall n,t, \tag{29}$$

$$\mathbb{E}[\mathbb{1}_{\{\widehat{S}_n[t]>0\}}]\leq e_n,\forall n,t, \tag{30}$$

$$U_n^*=\mathbb{E}[\delta_n(\widehat{S}_n[t])] \tag{31}$$

$$U^*=\sum_{n=1}^{N}w_nU_n^*, \tag{32}$$

where U^* is the optimal value of the optimization problem (2)-(4). 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:

$$\max\sum_{c\in\mathcal{C}}\phi_c\sum_{n=1}^{N}w_n\sum_{m=1}^{M}p_{n,m}^{(c)}\delta_n(R_m) \tag{33}$$

$$\text{s.t. }\sum_{c\in\mathcal{C}}\phi_c\sum_{m=1}^{M}p_{n,m}^{(c)}R_m\geq r_n,\forall n, \tag{34}$$

$$\sum_{c\in\mathcal{C}}\phi_c\sum_{m=1}^{M}p_{n,m}^{(c)}\mathbb{1}_{\{R_m>0\}}\leq e_n,\forall n, \tag{35}$$

$$(\sum_{m=1}^{M}p_{n,m}^{(c)}R_m)_{n=1}^N\in\mathbf{CH}(\mathcal{S}^{(c)}), \tag{36}$$

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

By using the property of the stationary randomized schedule $\widehat{\mathbf{S}}[t]$ (cf. (29)-(32)), inequality (28) becomes

$$\begin{aligned}
&\Delta V[t]-K\sum_{n=1}^{N}w_n\mathbb{E}[\delta_n(S_n^*[t])|\mathbf{H}[t]]\\
\leq&B(\eta)-\sum_{n=1}^{N}(\eta T_n[t]+Kw_n)U_n^*\\
\leq&-\eta\sum_{n=1}^{N}U_n^*T_n[t]+B(\eta)-KU^*.
\end{aligned}\tag{37}$$

Taking the expectation on both sides, we have

$$\begin{aligned}
&\mathbb{E}[V[t+1]]-\mathbb{E}[V[t]]-K\sum_{n=1}^{N}w_n\mathbb{E}[\delta_n(S_n^*[t])]\\
\leq&B(\eta)-\eta\sum_{n=1}^{N}U_n^*\mathbb{E}[T_n[t]]-KU^*,
\end{aligned}\tag{38}$$

holding for all $t\geq 0$.

By summing both sides of (38) over $t\in\{0,1,\cdots,L-1\}$

and dividing by L, we have

$$\frac{1}{L}(\mathbb{E}[V(L)] - \mathbb{E}[V(0)]) - K\frac{1}{L}\sum_{t=0}^{L-1}\sum_{n=1}^{N} w_n \mathbb{E}[\delta_n(S_n^*[t])]$$
$$\leq B(\eta) - \eta\frac{1}{L}\sum_{t=0}^{L-1}\sum_{n=1}^{N} U_n^* \mathbb{E}[T_n[t]] - KU^*. \quad (39)$$

Hence, we have

$$\eta\frac{1}{L}\sum_{t=0}^{L-1}\sum_{n=1}^{N} U_n^* \mathbb{E}[T_n[t]]$$
$$\leq B(\eta) + K\frac{1}{L}\sum_{t=0}^{L-1}\sum_{n=1}^{N} \mathbb{E}[w_n \delta_n(S_n^*[t])] + \frac{1}{L}\mathbb{E}[V(0)]. \quad (40)$$

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

$$\lim_{L\to\infty} \frac{1}{L}\sum_{t=0}^{L-1}\sum_{n=1}^{N} U_n^* \mathbb{E}[T_n[t]] \leq \frac{B(\eta) + KNw_{\max}}{\eta}. \quad (41)$$

In addition, from (39) we have

$$K\frac{1}{L}\sum_{t=0}^{L-1}\sum_{n=1}^{N} \mathbb{E}[w_n \delta_n(S_n[t])] \geq \quad KU^* - B(\eta) - \frac{1}{L}\mathbb{E}[V(0)]. \quad (42)$$

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

$$\lim_{L\to\infty} \frac{1}{L}\sum_{t=0}^{L-1}\sum_{n=1}^{N} w_n \mathbb{E}[\delta_n(S_n^*[t])] \geq U^* - \frac{B(\eta)}{K}. \quad (43)$$

Finally, we will show that all virtual queues are mean rate stable. From (39), we have

$$\mathbb{E}[V[L]] - \mathbb{E}[V[0]] \leq L(B(\eta) + KNw_{\max}), \forall t \geq 0. \quad (44)$$

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

$$\frac{1}{2}\sum_{n=1}^{N} \mathbb{E}\left[(Q_n^{(S)}[L])^2\right] \leq L(B(\eta) + KNw_{\max}) + \mathbb{E}[V[0]]. \quad (45)$$

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

$$\mathbb{E}\left[(Q_n^{(S)}[L])^2\right] \leq 2\left(L(B(\eta) + KNw_{\max}) + \mathbb{E}[V[0]]\right) \quad (46)$$

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

$$\mathbb{E}\left[Q_n^{(S)}[L]\right] \leq \sqrt{2\left(L(B(\eta) + KNw_{\max}) + \mathbb{E}[V[0]]\right)} \quad (47)$$

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

$$\lim_{L\to\infty} \frac{\mathbb{E}\left[Q_n^{(S)}[L]\right]}{L} \leq 0 \quad (48)$$

Since $Q_n^{(S)}[L] \geq 0$, we have $\lim_{L\to\infty} \mathbb{E}\left[Q_n^{(S)}[L]\right]/L = 0$, which implies that virtual queue n for service rate is mean rate stable. Similarly, we can get that the virtual queue n for energy consumption is also mean rate stable.

8 Related Work

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

(a) Panoramic video streaming: Researchers have put a lot of effort into reducing the consumed bandwidth for panoramic video streaming. One widely accepted approach is to leverage motion prediction and stream part of the panoramic content. To cope with the imperfect prediction, researchers usually deliver a larger portion than the FoV. Recent work (e.g., [2], [4], [5]) has explored this idea and successfully incorporated it into the algorithm design. They utilized various learning mechanisms to improve the prediction accuracy further. Particularly, [5] proposed a novel multi-armed bandit formulation to dynamically determine the delivered portion of the whole panoramic scene. In a more practical scenario, some work (e.g., [3], [19], [20]) split the whole panoramic scene into tiles, prioritized the tiles by the possibility that they will be requested by the users, and streamed only partial of the tiles with higher priority. Other works (e.g., [21], [22], [23], [24]) further improve the quality of experience for the user by proposing innovative algorithms and integrating various system optimizations. Moreover, they conduct extensive evaluations to demonstrate robust performance under heterogeneous network conditions. However, it is worth noting that they predominantly focus on single-user scenarios and do not offer theoretical guarantees regarding their methods. In contrast, our paper addresses a distinct avenue of research, tackling multi-user scenarios with an emphasis on providing theoretical guarantees. We believe this sets our work apart and contributes uniquely to the field.

There are also some papers (e.g., [25], [26]) leveraging either the unique properties of the panoramic video streaming or the client-side computation resources. [25] observed that there are some unique quality-determining factors in 360-degree videos, such as moving speed of a user's viewport, luminance variance, and the difference of depth-of-field. Then, the authors designed a novel tiling scheme to cope with such factors related to users' sensitivity. [26] utilized additional client-side computation to reduce bandwidth requirement. Specifically, the client runs a deep learning model to recover and enhance the quality of the video received from the server which is significantly compressed.

(b) Virtual Reality: Unlike panoramic video streaming, virtual reality (VR) introduces three more degrees of freedom for the user's action. Besides the rotation of the head, users are also able to move in a virtual world, which further complicates the problem. On the other hand, rendering high-quality VR images requires extremely high computation capability, which cannot be supported by even the latest model of mobile devices. Recent work has developed mobile VR systems which wirelessly communicate with a server and proposed multiple methods to meet the extremely high bandwidth requirement. One of the intuitive ways is to increase the bandwidth by modern wireless technologies like mmWave (e.g., [27], [28], [29]). Since the highly directional property of mmWave contradicts the movement nature of VR users, they either designed an adaptive antenna mirror to reflect the signals or utilized a link adaptation algorithm

with relatively lower frequency.

Beyond the scope of a single-user VR system, some recent work (e.g., [11], [30], [31]) focused on providing a simultaneous VR viewing experience among multiple users and guaranteed good quality of experience for all of them. They observed unique properties in a muitl-user setup like redundant VR frames with similar contents and designed novel systems with heuristic approaches to be implemented on commercial devices. There are also some researchers (e.g., [32], [33]) interested in the interaction between multiple users and came up with ideas to improve the synchronization performance among users. However, most of them designed heuristic approaches and did not provide the theoretical analysis of the system performance.

(c) Wireless Scheduling Design: The scheduling design is essentially an important topic under wireless networks since only a subset of the users are available for simultaneous data transmission in the presence of wireless interference. Existing works have concentrated on efficient scheduling design considering various quality-of-service (QoS) components, including throughput, delay, and service regularity. For instance, some works aimed to design a novel scheduling framework that provides a throughput performance guarantee for real-time traffic (e.g. [34], [35]). While other works put lots of effort into reducing the delay (e.g., [36], [37], [38]) and providing service regularity guarantees (e.g., [9], [39]). However, none of them considered the application-level throughput, which is essential for the panoramic scene delivery.

9 Conclusion and Future Work

In this work, we studied the wireless scheduling design for multi-user interactive panoramic scene delivery with the goal of maximizing application-level throughput while guaranteeing service regularity performance. Notably, the problem is constrained by wireless interference, minimum service rate, and maximum allowable energy consumption. We analytically characterized the successful viewing probability as the function of the delivered portion. We used the Time-Since-Last-Service counter to capture the service regularity performance and incorporated it into the stochastic network optimization framework to develop a motion-prediction-based scheduling algorithm. We proved that our proposed algorithm asymptotically maximizes application-level throughput and provides service regularity guarantees. Finally, we conducted trace-based simulations using panoramic scene gallery (3 DoF) and VR touring (6 DoF) datasets to demonstrate the efficiency of our algorithm. It is worth noting that our study only examined the influence of different portions of panoramic content on the transmission rate. In practice, there may be multiple available bitrates for the same panoramic content, which would further complicate this problem and is left for future research.

References

[1] "Google Earth VR." [Online]. Available: https://arvr.google.com/earth/

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] H. Gupta, J. Chen, B. Li, and S. R., "Online learning-based rate selection for wireless interactive panoramic scene delivery," in *IEEE INFOCOM 2022-IEEE Conference on Computer Communications*. IEEE, 2022.

[7] 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.

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

[9] 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.

[10] 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.

[11] X. Liu, C. Vlachou, M. Yang, F. Qian, L. Zhou, C. Wang, L. Zhu, K.-H. Kim, G. Parmer, Q. Chen *et al.*, "Firefly: Untethered multi-user {VR} for commodity mobile devices," in *2020 {USENIX} Annual Technical Conference ({USENIX}{ATC} 20)*, 2020, pp. 943–957.

[12] J. Chen, X. Qin, G. Zhu, B. Ji, and B. Li, "Motion-prediction-based wireless scheduling for multi-user panoramic video streaming," in *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*. IEEE, 2021, pp. 1–10.

[13] 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.

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

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

[16] 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.

[17] 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.

[18] L. Tassiulas, "Linear complexity algorithms for maximum throughput in radio networks and input queued switches," in *Proceedings. IEEE INFOCOM'98, the Conference on Computer Communications. Seventeenth Annual Joint Conference of the IEEE Computer and Communications Societies. Gateway to the 21st Century (Cat. No. 98*, vol. 2. IEEE, 1998, pp. 533–539.

[19] M. Xiao, C. Zhou, V. Swaminathan, Y. Liu, and S. Chen, "Bas-360: Exploring spatial and temporal adaptability in 360-degree videos over http/2," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 2018, pp. 953–961.

[20] C. Zhou, M. Xiao, and Y. Liu, "Clustile: Toward minimizing bandwidth in 360-degree video streaming," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 2018, pp. 962–970.

[21] L. Sun, F. Duanmu, Y. Liu, Y. Wang, Y. Ye, H. Shi, and D. Dai, "Multi-path multi-tier 360-degree video streaming in 5g networks," in *Proceedings of the 9th ACM multimedia systems conference*, 2018, pp. 162–173.

[22] L. Xie, Z. Xu, Y. Ban, X. Zhang, and Z. Guo, "360probdash: Improving qoe of 360 video streaming using tile-based http adaptive streaming," in *Proceedings of the 25th ACM international conference on Multimedia*, 2017, pp. 315–323.

[23] C. Ozcinar, A. De Abreu, and A. Smolic, "Viewport-aware adaptive 360 video streaming using tiles for virtual reality," in *2017*

IEEE International Conference on Image Processing (ICIP). IEEE, 2017, pp. 2174–2178.

[24] S. Ji, S. Lee, G. Park, and H. Song, "Head movement-aware mpeg-dash srd-based 360° video vr streaming system over wireless network," in *2022 IEEE 23rd International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. IEEE, 2022, pp. 281–287.

[25] Y. Guan, C. Zheng, X. Zhang, Z. Guo, and J. Jiang, "Pano: Optimizing 360 video streaming with a better understanding of quality perception," in *Proceedings of the ACM Special Interest Group on Data Communication*, 2019, pp. 394–407.

[26] M. Dasari, A. Bhattacharya, S. Vargas, P. Sahu, A. Balasubramanian, and S. R. Das, "Streaming 360-degree videos using super-resolution," in *IEEE INFOCOM*, 2020.

[27] O. Abari, D. Bharadia, A. Duffield, and D. Katabi, "Enabling high-quality untethered virtual reality," in *14th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 17)*, 2017, pp. 531–544.

[28] Y. Liu, J. Liu, A. Argyriou, and S. Ci, "Mec-assisted panoramic vr video streaming over millimeter wave mobile networks," *IEEE Transactions on Multimedia*, vol. 21, no. 5, pp. 1302–1316, 2018.

[29] Y. Liu, S. Zhang, M. Gowda, and S. Nelakuditi, "Leveraging the properties of mmwave signals for 3d finger motion tracking for interactive iot applications," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 6, no. 3, pp. 1–28, 2022.

[30] Y. Li and W. Gao, "Muvr: Supporting multi-user mobile virtual reality with resource constrained edge cloud," in *2018 IEEE/ACM Symposium on Edge Computing (SEC)*. IEEE, 2018, pp. 1–16.

[31] J. Chen, F. Qian, and B. Li, "Enhancing quality of experience for collaborative virtual reality with commodity mobile devices," in *2022 IEEE 42nd International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2022, pp. 1018–1028.

[32] ——, "An interactive and immersive remote education platform based on commodity devices," in *IEEE INFOCOM 2021-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2021, pp. 1–2.

[33] Z. Dong, J. Chen, and B. Li, "Collaborative mixed-reality-based firefighter training," in *IEEE INFOCOM 2023-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2023, pp. 1–2.

[34] N. Lu, B. Ji, and B. Li, "Age-based scheduling: Improving data freshness for wireless real-time traffic," in *Proceedings of the eighteenth acm international symposium on mobile ad hoc networking and computing*, 2018, pp. 191–200.

[35] T. Zhang, T. Gong, S. Han, Q. Deng, and X. S. Hu, "Distributed dynamic packet scheduling framework for handling disturbances in real-time wireless networks," *IEEE Transactions on Mobile Computing*, vol. 18, no. 11, pp. 2502–2517, 2018.

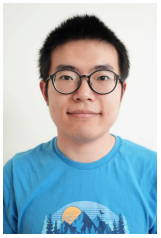
[36] S. Yang, F. Li, S. Trajanovski, X. Chen, Y. Wang, and X. Fu, "Delay-aware virtual network function placement and routing in edge clouds," *IEEE Transactions on Mobile Computing*, 2019.

[37] M. Masoudi and C. Cavdar, "Device vs edge computing for mobile services: Delay-aware decision making to minimize power consumption," *IEEE Transactions on Mobile Computing*, 2020.

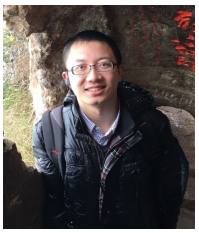
[38] J. Li, W. Liang, Y. Li, Z. Xu, X. Jia, and S. Guo, "Throughput maximization of delay-aware dnn inference in edge computing by exploring dnn model partitioning and inference parallelism," *IEEE Transactions on Mobile Computing*, 2021.

[39] B. Li, R. Li, and A. Eryilmaz, "Heavy-traffic-optimal scheduling with regular service guarantees in wireless networks," in *Proceedings of the fourteenth ACM international symposium on Mobile ad hoc networking and computing*, 2013, pp. 79–88.

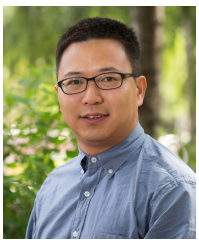
Jiangong Chen (Student Member, IEEE) received the B.S. degree in automation from Shandong University, China, in 2019, and the M.S. degree in electrical engineering from the University of Rhode Island in 2021. He is currently working towards a Ph.D. degree in the Department of Electrical Engineering at The Pennsylvania State University. His research interests include system developments and algorithm design in networking for virtual/augmented reality. He was a recipient of the IEEE INFOCOM 2023 Best Demo Award and REUNS 2022 Best Poster Presentation Award.



Xudong Qin received the B.S. degree in automation from Shandong University, Ji'nan, China, in 2015, the M.S. degree in control theory and engineering from Northeastern University, Shenyang, China, in 2018, and the Ph.D. degree in electrical engineering at The Pennsylvania State University, Pennsylvania, USA in 2023. His research interests include mobile edge/cloud computing and wireless scheduling design, and he is currently working as an electrical engineer at Marsh Bellofram.



Guangyu Zhu received his B.S. degree in Mathematics from Zhejiang University, Hangzhou, China, in 2011, and his MSc and Ph.D. degree in Statistics from The University of Florida in 2016. Dr. Zhu is an Assistant Professor in the Department of Computer Science and Statistics at at the University of Rhode Island. His research focuses on the intersection of machine learning and Statistics, methods and algorithms for analyzing high-dimensional data.



Bo Ji (S'11-M'12-SM'18) received his B.E. and M.E. degrees in Information Science and Electronic Engineering from Zhejiang University, Hangzhou, China, in 2004 and 2006, respectively, and his Ph.D. degree in Electrical and Computer Engineering from The Ohio State University, Columbus, OH, USA, in 2012. Dr. Ji is an Associate Professor of Computer Science and a College of Engineering Faculty Fellow at Virginia Tech, Blacksburg, VA, USA. Prior to joining Virginia Tech, he was an Associate/Assistant Professor in the Department of Computer and Information Sciences at Temple University from July 2014 to July 2020. He was also a Senior Member of the Technical Staff with AT&T Labs, San Ramon, CA, from January 2013 to June 2014. His research interests lie broadly at the intersections of networking, machine learning, security & privacy, and mixed reality.

Bin Li (S'11-M'16-SM'20) received his B.S. degree in Electronic and Information Engineering in 2005, M.S. degree in Communication and Information Engineering in 2008, both from Xiamen University, China, and Ph.D. degree in Electrical and Computer Engineering from The Ohio State University in 2014. He is currently an associate professor in the Department of Electrical Engineering at the Pennsylvania State University, University Park, PA, USA. Prior to joining Penn State, he was a Postdoctoral Researcher in the Coordinated Science Lab at the University of Illinois at Urbana-Champaign (UIUC) from June 2014 to August 2016, and an Assistant Professor in the Department of Electrical, Computer and Biomedical Engineering at the University of Rhode Island from August 2016 to July 2021. His research focuses on the intersection of networking, machine learning, and system developments, and their applications in networking for virtual/augmented reality, mobile edge computing, mobile crowd-learning, and Internet-of-Things.