
Mingzhe Chen*, Walid Saad†, and Changchuan Yin*

*Beijing Key Laboratory of Network System Architecture and Convergence,
Beijing University of Posts and Telecommunications, Beijing, China 100876, Emails: chenmingzhe@bupt.edu.cn, ccyin@ieee.org.
†Wireless@VT, Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA, USA, Email: walids@vt.edu.

Abstract—In this paper, the problem of resource management is studied for a network of wireless virtual reality (VR) users communicating over small cell networks (SCNs). In order to capture the VR users’ quality-of-service (QoS), a novel VR model, based on multi-attribute utility theory, is proposed. This model jointly accounts for VR metrics such as tracking accuracy, processing delay, and transmission delay. In this model, the small base stations (SBSs) act as the VR control centers that collect the tracking information from VR users over the cellular uplink. Once this information is collected, the SBSs will then send the three-dimensional images and accompanying surrounding audio to the VR users over the downlink. Therefore, the resource allocation problem in VR wireless networks must jointly consider both the uplink and downlink. This problem is then formulated as a noncooperative game and a distributed algorithm based on the machine learning framework of echo state networks (ESNs) is proposed to find the solution of this game. The proposed ESN algorithm enables the SBSs to predict the VR QoS of each SBS and guarantees the convergence to a mixed-strategy Nash equilibrium. Simulation results show that the proposed algorithm yields significant gains, in terms of total utility value of VR QoS, that reach up to 22% compared to Q-learning. The results also show that the proposed algorithm has a faster convergence time than Q-learning and can guarantee low delays for VR services.

I. INTRODUCTION

Recent advances in storage, computing, big data analytics, and artificial intelligence will realize the much sought after vision of immersive technologies and user-centric applications such as virtual reality (VR) [1]. VR enables the users to experience and interact with virtual environments through a first-person view. For instance, individuals can use a VR device to walk around in a fully immersive world and travel to any destination, within the confines of their own home. However, if VR devices such as HTC Vive rely on wired connections to a VR control center (e.g., a computer), for processing the information (e.g., 3D image generation), then the users will be significantly restricted in the type of actions that they can take and VR applications that they can experience. In particular, when using wired networking solutions, VR users only can use the VR applications within a very restricted area (limited by the wires) and, hence, they will not have the opportunity to use VR services anytime, anywhere. In order to enable pervasive and truly immersive VR applications, one can deploy wireless VR systems [2] that use reliable wireless connections, such as those provided by cellular networks. In particular, VR systems can use the wireless connectivity of emerging small cell networks (SCNs) [3] in which small cell base stations (SBSs) can act as the VR control centers that directly connect to the VR devices over wireless links and, consequently, the SBSs will collect the

tracking information from the VR devices and send the VR images to the VR devices over wireless cellular links. However, operating VR devices over wireless cellular networks such as SCNs faces many challenges [2] that include tracking accuracy, low delay, high data rate, and effective image compression.

The existing literature has studied a number of problems related to VR such as in [1], [2], and [4]–[7]. The authors in [1] and [2] discussed current and future trends of VR systems. However, the works in [1] and [2] are restricted to preliminary surveys that do not provide any mathematically rigorous modeling of VR over wireless networks. In [4], the authors propose a streaming scheme that delivers only the visible portion of a 360° video based on head movement prediction. In [5], an algorithm for generating high-quality stereo panoramas is proposed. The authors in [6] proposed a real-time solution that uses a single commodity RGB-D camera to track hand manipulation. However, existing works such as in [1], [2], and [4]–[6] focus on VR systems that are deployed over wired networks and, as such, these works do not capture any challenges of deploying VR over cellular networks. Moreover, most of these existing works [1], [2], and [4]–[6] only focus on the improvement of one VR quality-of-service (QoS) metric such as tracking or 3D image generation. Indeed, this prior art does not develop any VR-specific model that can capture all factors of VR QoS and, hence, it falls short in addressing the challenges of optimizing VR QoS for wireless users.

Some recent works such as [8]–[12] have studied a number of ideas related to ultra-reliable and low-delay communication in wireless networks which can be relevant for VR applications. However, the works in [8]–[12] focus on conventional machine type devices that transmit small data packets (e.g., sensors) and, hence, their results may not scale well to a VR network which typically requires ultra-reliable, low-delay, and high data rate transmission. Moreover, most of these existing works [8]–[12] that only focus on the transmission delay in wireless networks do not consider the processing delay for VR image construction.

The main contribution of this paper is a novel framework for enabling wireless cellular networks to integrate VR applications and services. To our best knowledge, this is the first work that develops a comprehensive framework for analyzing the performance of VR services over cellular networks. This paper provides the following key contributions:

- We propose a novel VR model based on the tools of multi-attribute utility theory [13], to jointly capture the tracking accuracy, transmission delay and processing delay thus en-

This work was supported in part by the National Natural Science Foundation of China under Grants 61671086 and 61629101, in part by BUPT-SICE Excellent Graduate Students Innovation Fund, and in part by the National Science Foundation under Grants CNS-1460316 and CNS-1513697.

1Here, tracking pertains to the fact that the immersive VR applications must continuously collect a very accurate localization of each user including the positions, orientation, and eye movement (i.e., gaze tracking).
For the considered VR applications over wireless, we analyze resource (subcarrier) allocation jointly over the uplink and downlink. We formulate the problem as a noncooperative game in which the players are the SBSs. Each player seeks to find an optimal spectrum allocation scheme to optimize a utility function that captures the VR QoS.

To solve this VR resource management game, we propose a learning algorithm based on echo state networks (ESNs) [14] and [15] that can predict the value of VR QoS that results from resource allocation and, hence, can reach a mixed-strategy Nash equilibrium (NE).

Extensive simulations are used to assess the performance of the proposed framework. Simulation results show that the proposed algorithm can yield 22% gain in terms of total utility value of VR QoS compared to Q-learning.

The rest of this paper is organized as follows. The system model and problem formulation are presented in Section II. The ESN-based resource allocation algorithm is proposed in Section III. In Section IV, numerical simulation results are presented and analyzed. Finally, conclusions are drawn in Section V.

II. SYSTEM MODEL AND PROBLEM FORMULATION

Consider the downlink of a wireless SCNs servicing a set \( \mathcal{U} \) of \( V \) wireless VR users via a set \( \mathcal{B} \) of \( B \) SBSs. Here, we focus on entertainment VR application such as watching immersive videos and playing immersive games [4]. In contrast to traditional mobile gaming and video streaming, VR technology allows the users to be immersed in a virtual environment during which they can experience a 3D and high-resolution 360° vision with 3D surround stereo. Compared to a classical 120° image, a 360° panoramic image enables the VR users to experience a surrounded vision without any dead spots. Moreover, immersive VR will provide a 360° panoramic image for each eye of a VR user. In this case, a VR image needs more pixels than a traditional two-dimensional image, and, hence, VR transmission will require more stringent requirements in terms of data rate (over 100 Mbps) [2], delay (less than 20 ms), and reliability than traditional multimedia services.

In our model, the SBSs adopt an orthogonal frequency division multiple access (OFDMA) technique and transmit over a set of \( S^u \) of \( S^d \) uplink orthogonal subcarriers and a set of \( S^d \) of \( S^d \) downlink orthogonal subcarriers, as shown in Fig. 1. The uplink subcarriers are used to transmit the data that is collected by the VR sensors placed at a VR user’s headset or near the VR user while the downlink subcarriers are used to transmit the image displayed on each user’s VR device. We define the coverage of each SBS as a circular area of radius \( r_B \). We assume that each SBS only allocates subcarriers to the users located in its coverage range and each user will associate with its nearest SBS. We also assume that all of the subcarriers of each SBS will be allocated to the associated users.

A. VR Model

In a VR model, we need to capture the VR transmission requirements such as high data rate, low delay, and accurate tracking. We consider delay and tracking accuracy as the main VR QoS metrics of interest. Based on the accurate localization of each user, the SBS can build the immersive and virtual environment for each user. Among the components of the VR QoS, a delay metric can be defined to capture two key VR service requirements: high data rate and low delay. Next, we will explicitly discuss the components of the VR QoS metric that will be accounted for.

1) Tracking Model: VR tracking will consist of the position tracking and orientation tracking [4]. VR tracking directly affects the construction of users’ virtual environment. This is due to the fact that the SBSs need to use the user’s localization information to construct the virtual environment. Hereafter, we use the term “localization information” to refer to the information related to the user’s location and orientation. We use the localization information of each user as the primary component of tracking [2]. The tracking vector of each user \( i \) can be given by \( \chi_i = \left[ x_i, y_i, z_i, \varphi_i, \psi_i, \eta_i \right] \), where the vector \( [x_i, y_i, z_i] \) represents the position of each VR user while the vector \( [\varphi_i, \psi_i, \eta_i] \) represents the orientation of each user. Here, we need to note that the position and orientation of each user are determined by the SBS via the information collected by the sensors. The tracking accuracy of each VR user, \( K_i(s_{ij}^u) \), is:

\[
K_i(s_{ij}^u) = 1 - \frac{\| \chi_i(s_{ij}^u) - \chi_i^R \|}{\max_{s_{ij}} \| \chi_i(s_{ij}^u) - \chi_i^R \|},
\]

where \( \chi_i(s_{ij}^u) \) is the localization determined by SBS \( j \) while \( \chi_i^R \) is the localization which can be obtained from the process of force feedback. Force feedback represents the feedback that the users send to the SBSs whenever those users are not satisfied with the displayed VR image. Both \( \chi_i(s_{ij}^u) \) and \( \chi_i^R \) are transmitted via wireless links and, hence, the SBSs cannot receive the real localization information of each user. Thus, we use the deviation between the localization determined by each SBS and the localization provided by user’s force feedback to evaluate the tracking accuracy. The value of \( K_i(s_{ij}^u) \) depends on the uplink rate of each user. The increase of the uplink rate of each user enables the SBS to obtain more tracking information for a given user, and, hence, the SBSs can use more tracking information to determine the user’s localization more accurately.

2) Delay: Next, we define the delay component that consists of the transmission delay and processing delay. The transmission delay of each user \( i \) can be given by:

\[
D_{ij}^T(s_{ij}^d) = \frac{L}{c_{ij}(s_{ij}^d)},
\]

where \( c_{ij}(s_{ij}^d) \) is the capacity of subcarrier \( s_{ij}^d \).
where \( L \) is the size of VR image that each SBS needs to transmit to the associated users, \( c_{ij} (s_{ij}^d) = \sum_{k=1}^{s_{ij}} s_{ij,k} \cdot \log_2 (1 + \gamma_{ij,k}) \).

Here, \( s_{ij}^d = [s_{ij,1}, \ldots, s_{ij,s_{ij}}] \) is the vector of subcarriers that SBS \( j \) allocates to user \( i \) with \( s_{ij,k} \in (1, 0). \) \( s_{ij,k} = 1 \) indicates that subcarrier \( k \) is allocated to user \( i. \) \( \gamma_{ij,k}^0 = \frac{p_{ij}}{\sigma^2} + \frac{1}{\sum_{i \in \mathcal{B}_j, i \neq j} p_{ik} h_{ik}^0} \) is the signal-to-interference-plus-noise ratio (SINR) between user \( i \) and SBS \( j \) over subcarrier \( k = B_k \) being the set of the SBSs that use downlink subcarrier \( k. \) \( B \) is the bandwidth of each subcarrier, \( P_B \) is the transmit power of SBS \( j \) which is assumed to be equal for all SBSs, \( \sigma^2 \) is the variance of the Gaussian noise, and \( h_{ij,k}^0 = g_{ij}^0 \cdot p_{ij}^\beta \) is the path loss between user \( i \) and SBS \( j \) over subcarrier with \( g_{ij}^0 \) being the Rayleigh fading parameter, \( p_{ij} \) the distance between user \( i \) and SBS \( j \) and \( \beta \) the path loss exponent. In the VR QoS, the \textit{processing delay} primarily stems from the tracking accuracy. Due to possibly inaccurate tracking, the SBS needs to adjust the virtual environment of each user. In this case, the SBS needs to spend additional time slots to reconstruct the virtual environment and transmit it to the user. While no existing work has quantified this processing delay, we propose the following function:

\[
D_i^p (s_{ij}^u) = v \log_2 (1 + K_i (s_{ij}^u)) \tag{3}
\]

where \( v \) is a scaling parameter that captures the relationship between the processing delay \( D_i^p \) and the tracking accuracy \( K_i (s_{ij}^u). \) To our best knowledge, there is no existing work that considers the relationship between the tracking prediction and delay. From (3), we can see that inaccurate tracking will directly lead to an increased processing delay. That is due to the fact that each SBS must use additional time slots to re-construct and transmit the virtual environment to each VR user. However, as the inaccuracy increases, the processing delay will saturate at a given maximum value. This maximum value represents the time needed to retransmit the entire VR image to the user again. Hence, we propose (3) to capture this relationship. The total delay of each user \( i \) can hence be given by \( D_i (s_{ij}^d, s_{ij}^u) = D_i^p (s_{ij}^u) + D_i^t (s_{ij}^d). \)

\section*{B. Utility Function Model}

Next, we introduce a method based on the framework \textit{multiattribute utility theory} [13] to construct an appropriate utility function that can effectively capture the delay and tracking of VR QoS. In order to construct the total utility function, we first define a conditional utility function for delay of VR QoS [13]. Here, the total utility function indicates that both tracking and delay will contribute to the utility value while, in the conditional utility function of delay, only delay contributes to the utility function and the tracking value is given. The conditional utility function can be used to formulate the total utility function which is similar to the formulation of a joint probability function that uses conditional probability function. In particular, if one cannot derive the total utility function directly, the conditional utility function can be used to formulate it. In our model, the total utility function of user \( i \) is given by \( U_i (D_i (s_{ij}^d, s_{ij}^u), K_i (s_{ij}^u)). \) This utility jointly considers the delay and tracking of VR QoS. The conditional utility function of delay, \( U_i (D_i (s_{ij}^d, s_{ij}^u) | K_i (s_{ij}^u)) \), represents the total utility function given a certain value of tracking accuracy. In our case, the tracking and delay are not independent and, hence, it is hard to formulate the total utility function directly. Therefore, we first formulate the conditional utility function of delay. As shown in [13], a suitable definition for a conditional utility of delay for user \( i \) can be given by:

\[
U_i (D_i (s_{ij}^d, s_{ij}^u) | K_i (s_{ij}^u)) = \frac{D_{max} (s_{ij}^d) - D_i (s_{ij}^d, s_{ij}^u)}{D_{max} (s_{ij}^d) - \gamma_D}, \quad D_i (s_{ij}^d, s_{ij}^u) < \gamma_D,
\]

where \( \gamma_D \) is the maximal tolerable delay for each VR user (maximum supported by the VR system being used) and \( D_{max} (s_{ij}^d) = \max \{ D_i (s_{ij}^d, s_{ij}^u) \} \) is the maximum delay of VR user \( i \) given \( s_{ij}^d. \) Here, \( U_i (D_{max} (s_{ij}^d) | K_i (s_{ij}^u)) = 0 \) and \( U_i (\gamma_D | K_i (s_{ij}^u)) = 1. \) Since delay and tracking are both dominant components, we can construct the total utility function, \( U_i (D_i (s_{ij}^d, s_{ij}^u), K_i (s_{ij}^u)) \), that jointly considers the delay and tracking based on [13]. Here, a \textit{dominant component} represents the component that will minimize the total utility function regardless of the value of other components. For VR QoS, delay and tracking are both dominant components. For example, the VR QoS will be minimized when the value of delay function is at a minimum regardless of the value of tracking accuracy. Dominant components such as delay and tracking will simplify the formulation of the total utility function [13]. Therefore, the total utility function of tracking and delay is [13]:

\[
U_i (D_i (s_{ij}^d, s_{ij}^u), K_i (s_{ij}^u)) = U_i (D_i (s_{ij}^d, s_{ij}^u) | K_i (s_{ij}^u)) \cdot U_i (K_i (s_{ij}^u)).
\]

\[
U_i (D_i (s_{ij}^d, s_{ij}^u), K_i (s_{ij}^u)) \equiv (1 - \frac{\| X_i (s_{ij}^u) - X^R_i \|}{\max_{s_{ij}} \| X_i (s_{ij}) - X^R_i \|}) \cdot \frac{D_{max} (s_{ij}^d) - D_i (s_{ij}^d, s_{ij}^u)}{D_{max} (s_{ij}^d) - \gamma_D}. \tag{5}
\]

where (a) is obtained from the fact \( U_i (K_i (s_{ij}^u)) = K_i (s_{ij}^u) \) and substituting (1) and (4) into (5). From (5), we can see that, the subcarriers allocated to user \( i \) for data transmission, \( s_{ij}^d \), and the subcarriers allocated to user \( i \) for obtaining the tracking information, \( s_{ij}^u \), jointly determine the value of the total utility function. Moreover, this total utility function can assign a unique value to each tracking and delay components of the VR QoS.

\section*{C. Problem Formulation}

Given this system model, our goal is to develop an effective resource allocation scheme that allocates subcarriers in a way to maximize the VR QoS of all users. However, the maximization problem depends jointly on both the downlink subcarriers allocation and the uplink subcarriers allocation. Moreover, the VR QoS of each SBS depends not only on its own choice of the subcarriers allocation scheme but also on the remaining SBSs’ schemes. In this regard, we formulate a noncooperative game [16] given by \( \mathcal{G} = \{ \mathcal{B}, \{ A_j \}_{j \in \mathcal{B}}, \{ u_j \}_{j \in \mathcal{B}} \} \) with the SBSs being the players. Each player \( j \) has a set \( A_j = \{ a_{j1}, \ldots, a_{j|A_j|} \} \) of \( A_j \) actions. In this game, each action of SBS \( j, a_j = (d_{ij}, s_{ij}) \) consists of:

(i) \textbf{downlink subcarriers allocation vector,} \( d_j = [s_{ij,1}, \ldots, s_{ij,s_{ij}}] \) and

\[ \sum_{k=1}^{s_{ij}} s_{ij,k} = 1. \] Here, \( s_{ij}^d = [s_{ij,1}, \ldots, s_{ij,s_{ij}}] \) represents the downlink subcarrier that SBS \( j \) allocates to user \( i \) and
where \( a_j \in A_j \) is an action of SBS \( j \) and \( a_{-j} \) denotes the action profile of all SBSs other than SBS \( j \). Let \( \pi_{j,a_i} = \lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} \{ a_i = a_j \} \) be the probability of SBS \( j \) using action \( a_{ij} \) and, hence, \( \pi_j = [\pi_{j,a_1}, \ldots, \pi_{j,a_j}, a_{-j}] \) is a probability distribution of SBS \( j \). We assume that the VR transmission is analyzed during a period that consists of \( T \) time slots. Therefore, the average value of the utility function is:

\[
\bar{u}_j (a_j, a_{-j}) = \lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} u_j (a_j, a_{-j}) = \sum_{a \in A} (u_j (a_j, a_{-j}) \prod_{j \in \mathcal{B}} \pi_j, a_j).
\]

Given the proposed model, our goal is to solve the proposed resource allocation game. A suitable solution for the studied game is the concept of the mixed-strategy Nash equilibrium, formally defined as follows [17]:

**Definition 1.** (mixed-strategy Nash Equilibrium): A mixed-strategy profile \( \pi^* = (\pi_1^*, \ldots, \pi_B^*) \) is a mixed-strategy Nash equilibrium if, \( \forall j \in \mathcal{B} \) and \( \pi_j \), we have:

\[
\bar{u}_j (\pi_j, \pi_{-j}^*) \geq \bar{u}_j (\pi_j^*, \pi_{-j}^*) ,
\]

where \( \bar{u}_j (\pi_n, \pi_{-n}) = \sum_{a \in A} u_j (a) \prod_{\pi_{j,a}} \pi_{j,a} \) is the expected utility of SBS \( j \) selecting the mixed strategy \( \pi_j \).

For our game, the mixed-strategy NE for the SBSs represents a solution of the game at which each SBS \( j \) can maximize the average VR QoS for its associated users, given the actions of its opponents.

### III. Echo State Networks for Self-Organizing Resource Allocation

To solve the VR game and find its NE, a learning algorithm based on the powerful framework of echo state networks (ESN) [18]. The proposed ESN-based learning algorithm enables each SBS to predict the value of VR QoS that results from each action and, hence, can reach a mixed-strategy NE without having to traverse all actions. Moreover, the proposed algorithm can store the past ESN information and, hence, can find an optimal convergence path from the initial state to a mixed-strategy NE. Next, we first introduce the components of an ESN-based learning algorithm. Then, we define the update process that the ESN-based algorithm uses to find the mixed-strategy NE.

### A. ESN Components

An ESN-based learning algorithm consists of five components: a) agents, b) inputs, c) ESN model, d) actions, and e) output. The specific components of the proposed ESN-based learning approach are thus defined as follows:

- **Agent:** The agents in our ESN are the SBSs in the set \( \mathcal{B} \).
- **Actions:** Each action of SBS \( j \), \( a_j \), jointly considers the uplink and downlink subcarriers, which is specified as follows:

\[
a_j = (d_j, v_j) = [s^d_{ij,1}, s^d_{ij,2}, \ldots, s^d_{ij,B_k}, s^u_{ij,1}, \ldots, s^u_{ij,B_k}]^T.
\]

In order to guarantee that any action always has a non-zero probability to be chosen, the \( \varepsilon \)-greedy exploration [19] is adopted in the proposed algorithm. This mechanism is responsible for selecting the actions that each SBS will perform during the learning process while harmonizing the tradeoff between exploitation and exploration. Therefore, the probability with which SBS \( j \) chooses action \( i \) will be given by:

\[
Pr (a_j) = \begin{cases} 
1 - \varepsilon + \frac{\varepsilon}{|A_j|}, & \text{arg max } \hat{u}_{\tau,j}(a_j), \\
\frac{\varepsilon}{|A_j|}, & \text{otherwise,}
\end{cases}
\]

where \( \hat{u}_{\tau,j}(a_j) = \sum_{a_{ij} \in A_j} u_j (a_j, a_{-j}) \pi_{-j,a_{-j}} \) is the expected utility of an SBS \( j \) with respect to the actions of its opponents, \( A_{-j} = \prod_{k \neq j, k \in \mathcal{B}} A_k \) is the set of actions other than SBS \( j \) and \( \pi_{-j,a_{-j}} = \sum_{a_{ij} \in A_j} \pi_j (a_j, a_{-j}) \) is the marginal probability distribution over the action set of SBS \( j \). From (10), we can see that each SBS will assign the highest probability, \( 1 - \varepsilon + \frac{\varepsilon}{|A_j|} \), to the action that results in the maximum utility value, \( \hat{u}_{\tau,j} \). For other actions, the SBS will assign the probability \( \frac{\varepsilon}{|A_j|} \). Thus, as each SBS maximizes the utility \( \hat{u}_{\tau,j} \), \( \bar{u}_j \) is maximized.

- **Input:** The input to the ESN-based learning algorithm is a vector \( x_{\tau,j} = [x_1, \ldots, x_B] \) where \( x_j \) represents the index of the probability distribution that SBS \( j \) uses at time \( \tau \). The vector \( x_{\tau,j} \) is then used to estimate the value of the utility value \( \bar{u}_j \) that captures the average VR QoS of SBS \( j \), \( y_{\tau,j} \).

- **ESN Model:** An ESN model for each SBS \( j \) must be defined. This model is a learning architecture that can find the relationship between the input \( x_{\tau,j} \) and output \( y_{\tau,j} \), thus building the function between the SBS’s probability distribution and the conditional utility value. Mathematically, the ESN model consists of the output weight matrix \( W_j^m \in \mathbb{R}^{W_n \times B} \) and the dynamic reservoir containing the input weight matrix \( W_j^i \in \mathbb{R}^{W_n \times N_w} \), and the recurrent matrix \( W_j = \begin{bmatrix} w_{11} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & w_{N_w, N_w} \end{bmatrix} \) with \( N_w \) being the number of the dynamic reservoir units. Here, the dynamic reservoir is used to store historical ESN information that includes input, reservoir state and output. Note that the historical ESN information can be used to find a fast convergence process from the initial state to the mixed-strategy NE. Here, the number of actions for each SBS determines the output weight matrix and recurrent matrix of each ESN.

- **Output:** The output of the ESN-based learning algorithm at time \( \tau \) is a vector of utility values \( y_{\tau,j} = [y_{\tau,j,1}, y_{\tau,j,2}, \ldots, y_{\tau,j,|A_j|}] \). Here, \( y_{\tau,j} \) represents the estimated value of utility \( \hat{u}_{\tau,j}(a_j) \) due to action \( i \) of SBS \( j \) and, hence, the ESN output, \( y_{\tau,j} \), will converge to the utility \( \hat{u}_j = [\hat{u}_j (a_{j1}), \ldots, \hat{u}_j (a_{j|A_j|})] \).
B. ESN-Based Learning Algorithm for Subcarriers Allocation

The proposed learning algorithm can find an optimal convergence path from initial state to a mixed-strategy NE. In particular, the proposed algorithm enables each SBS to reach a mixed-strategy NE traversing minimum number of strategies after training. In order to find the optimal convergence path, the proposed algorithm needs to store the past ESN information that consists of input, reservoir states, and output. The past ESN information from time 0 up until time \( \tau \) is stored by the dynamic reservoir state \( \mu_{\tau,j} \). The dynamic reservoir state of each SBS \( j \) at time \( \tau \) is given by:

\[
\mu_{\tau,j} = f(W_j \mu_{\tau-1,j} + W^{in}_j x_{\tau,j}),
\]

where \( f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \) is the tanh function. From (11), we can see that the dynamic reservoir state consists of the past dynamic reservoir states and the mixed strategy at time \( \tau \). The dynamic reservoir state actually stores the mixed strategy from time 0 to time \( \tau \). Based on the dynamic reservoir state, the proposed ESN algorithm will combine with the output weight matrix to estimate the value of conditional utility value. The estimation of the conditional utility value can be given by:

\[
y_{\tau,j} = W^{out}_j \mu_{\tau,j},
\]

where \( W^{out}_j \) is the output weight matrix at time slot \( \tau \). To enable the ESN to use reservoir state \( \mu_{\tau,j} \) to predict the conditional utility value, \( \hat{u}_{\tau,j} \), due to action \( a_{ji} \), we must train the output weight matrix \( W^{out}_j \) using a linear gradient descent approach, which is:

\[
W^{out}_{\tau+1,j} = W^{out}_{\tau,j} + \lambda \left( \hat{u}_{\tau,j} - y_{\tau,j} \left( x_{\tau,j}, a_{ji} \right) \right) \mu_{\tau,j}^T,
\]

where \( W^{out}_{\tau,j} \) is row \( i \) of \( W^{out}_j \), \( \lambda \) is the learning rate, and \( \hat{u}_{\tau,j} \) is the actual utility value. Here, \( \hat{u}_{\tau,j} \) is estimated by the utility value resulting from the actions performed by each SBS during each time slot \( \tau \). Based on the above formulations, the distributed ESN-based learning algorithm performed by every SBS \( j \) is summarized in Table I.

C. Convergence of the ESN-Based Learning Algorithm

Now, we prove the convergence of the proposed ESN-based learning algorithm. Then, we prove that the proposed algorithm reaches a mixed-strategy NE.

**Theorem 1.** The proposed ESN-based learning algorithm converges to the utility value, \( \bar{u}_j \), if any following conditions is satisfied:

\[
\begin{align*}
&\lambda < 1, \\
&\min \lambda \left( e^{-\lambda t} + \epsilon \right) \mu_{\tau,j} > 0,
\end{align*}
\]

where \( \lambda \) is a constant and \( \min \lambda \left( e^{-\lambda t} + \epsilon \right) \mu_{\tau,j} \geq 2 \), where \( W^{in}_{ji} \) represents the row \( i \) of \( W^{in}_j \). If \( \lambda \) satisfies the Robbins-Monro conditions [18] \( \lambda(t) > 0, \sum_{t=0}^{\infty} \lambda(t) = +\infty, \sum_{t=0}^{\infty} \lambda^2(t) < +\infty \).

**Proof.** See Appendix.

IV. Simulation Results

For our simulations, we consider an SCN deployed within a circular area with radius \( r = 100 \) m. \( U = 25 \) users and \( B = 4 \) SBSs are uniformly distributed in this SCN area. The Oculus VR device is considered in our simulations and, hence, the number of pixels for a panoramic image is \( 1920 \times 1080 \) and each pixel is stored in 32 bits. The flashed rate which represents the update rate of a VR image, is 60 images per second and the factor of compression is 300 [2]. Since two panoramic images consist of one VR image (one panoramic image per eye), the rate requirement of VR transmission will be 25.32 Mbit/s.

See Appendix.
Fig. 2. Total VR QoS utility of each user vs. the bandwidth of downlink and uplink subcarriers and tracking accuracy. Here, total VR QoS utility refers to (5). The tracking accuracy in Fig. 2(b) is more accurate than the tracking accuracy in Fig. 2(a).

In Fig. 2, different colors indicate different total VR QoS utilities. From Fig. 2, we can see that when the bandwidth of downlink (uplink) subcarrier is 0, the total VR QoS utility is 0 regardless of the bandwidth of uplink (downlink) subcarrier. This is due to the fact that the VR QoS depends on both delay and tracking. This corresponds to a scenario in which SBS \( j \) has enough downlink bandwidth to send a VR image to the user while the tracking information is inaccurate. In this case, SBS \( j \) cannot construct the accurate VR image due to the inaccuracy of user’s localization and, hence, the VR QoS of this user will be 0. From Figs. 2(a) and 2(b), we can see that, as the tracking accuracy increases, the uplink subcarrier bandwidth that each user needs to maximize the VR QoS decreases. This stems from the fact that tracking accuracy affects the processing delay.

In Fig. 3, we show how the average delay for each servicing user varies as the number of SBSs increases. From Fig. 3, we can see that, as the number of SBSs increases, the transmission delay for each serviced user increases then decreases. This is due to the fact that as the number of SBSs increases, the number of users located in each SBS’s coverage decreases and, hence, the average delay increases. However, as the number of SBSs keeps increasing, the average delay decreases. This stems from the fact that the interference from the SBSs to the users increases as the number of SBSs continues to increase. Fig. 3 also shows that the proposed algorithm achieves up to 18.6% gain in terms of average delay compared to the Q-learning algorithm for the case with 6 SBSs. In Fig. 3, we can also see that the proposed ESN-based learning algorithm enables the wireless VR transmission to meet typical delay requirement of VR applications. These gains stem from the fact that the proposed algorithm use the past ESN information to find a better solution for the proposed game.

Fig. 4 shows how the VR QoS for all users changes as the number of SBSs varies. From Fig 4, we can see that average total utility value (at the mixed-strategy NE) of all considered algorithms increases as the number of SBSs increases. This is due to the fact that, as the number of SBSs increases, the number of users located within the coverage of each SBS increases and the distances from the SBSs to their associated users decrease. Fig. 4 shows that the proposed algorithm can yield up to of 16.1% gain in terms of the average of total VR QoS utility compared to the Q-learning for the case with 5 SBSs. Clearly, this gain is due to the fact that the proposed ESN algorithm can store the past ESN information and use this information to build the relationship between the input and output. In this case, the proposed learning algorithm can predict the output (conditional utility value) and, hence, find a better solution for allocating resources.

In Fig. 5, we show how the number of iterations needed till convergence for both the proposed ESN-based learning approach and Q-learning. In this figure, we can see that, as time elapses, the total VR QoS utilities for both the proposed algorithm and Q-learning increase until convergence to their final values. Fig. 5 also shows that the proposed algorithm needs 20 iterations to reach convergence while Q-learning needs 25 iterations to reach convergence. Hence, the proposed algorithm needs 25 iterations to reach convergence while Q-learning needs 25 iterations to reach convergence. Hence, the proposed algorithm achieves 25% gain in terms of the number of the iterations needed to reach convergence compared to Q-learning. This is due to the fact that the ESN in the proposed algorithm can store the SBSs’ action strategies and its corresponding total utility values.

V. CONCLUSION

In this paper, we have developed a novel multi-attribute utility theory based VR model that can capture the tracking and delay
components of VR QoS. Based on this model, we have proposed a novel resource allocation framework for optimizing the VR QoS for all users. We have formulated the problem as a noncooperative game between the SBSs that seeks to maximize the average VR QoS utilities for all users. To solve this problem, we have developed a novel algorithm based on the machine learning tools of echo state networks. The proposed algorithm enables each SBS to decide on its actions autonomously according to the users’ and networks’ network states. Moreover, the proposed learning algorithm only needs to update the mixed strategy during the training process and, hence, can quickly converge to a mixed-strategy NE. Simulation results have shown that the proposed VR model can capture the VR QoS in wireless networks. The results have also shown that the proposed approach yields significant performance gains in terms of total VR QoS utilities for all users compared to conventional approaches.

APPENDIX

In order to prove Theorem 1, we first need to prove that the ESN-based learning algorithm converges to a constant value. Here, we do not know the exact value to which the proposed algorithm converges. Our goal is to show that the proposed algorithm cannot diverge. Then, we derive the exact value to which the ESN algorithm converges. For i), based on [18, Theorem 8], the conditions of convergence for an ESN are: a) The ESN is k-step unambiguous and b) The ESN-based learning process is k order Markov decision process (MDP). Here, the definition of k-step unambiguous can be given as follows:

Definition 2. Given an ESN with initial state \( \mu_{0,j} \), we assume that the input sequence \( x_{0,j}, \ldots, x_{\tau,j} \) results in an internal state \( \mu_{\tau,j} \), and the input sequence \( x_{0,j}′, \ldots, x_{\tau,j}′ \) results in an internal state \( \mu_{\tau,j}′ \). If \( \mu_{\tau,j} = \mu_{\tau,j}′ \) implies that \( x_{\tau-r,i,j} = x_{\tau-r,i,j}′ \), for all \( i = 0, \ldots, \tau \), then the ESN is k-step unambiguous. Here, \( \mu_{\tau,j} - \mu_{\tau,j}′ = W_j (\mu_{\tau-j} - \mu_{\tau-j}′) + W_{1j} (x_{\tau-j} - x_{\tau-j}′) \) can be rewritten as:

\[
\begin{bmatrix}
\mu_{\tau-j} - \mu_{\tau-j}′
\end{bmatrix} = \begin{bmatrix}
w_{1j} & W_{1j} & \cdots & W_{1j}\
\end{bmatrix} \begin{bmatrix}
x_{\tau-j} - x_{\tau-j}′
\end{bmatrix},
\]

where \( \mu_{\tau-j} = \text{element } k \) of \( \mu_{\tau-j} \), and \( \mu_{\tau-j}′ = \text{element } k \) of \( \mu_{\tau-j}′ \). Since the tank function in (11) ranges from -1 to 1, the maximum value of \( \left( \mu_{\tau-j} - \mu_{\tau-j}′ \right) \) is 2. As \( w_{kk} \in (-1,1) \), \( k = 1, \ldots, N_w, \max_k w_{kk} \left( \mu_{\tau-j} - \mu_{\tau-j}′ \right) < 2 \). In this case, if \( W_{jk} (x_{\tau-j} - x_{\tau-j}′) \geq 2 \), then \( \mu_{\tau-j} - \mu_{\tau-j}′ \neq 0 \). Therefore, if \( \mu_{\tau-j} - \mu_{\tau-j}′ = 0 \), then \( \mu_{\tau-j} = \mu_{\tau-j}′ \). In this case, an ESN is k-step unambiguous when \( W_{jk} (x_{\tau-j} - x_{\tau-j}′) \geq 2 \). Since the dynamic reservoir can only store limited ESN information [15], the dynamic reservoir state \( \mu_{\tau,j} \) only depends on the finite past reservoir states, i.e., \( \mu_{\tau-j} \). Moreover, the number of reservoir states and actions in the proposed algorithm is finite. Therefore, the proposed ESN-based algorithm is a k order MDP and, hence, condition 2) is satisfied. For case 2), if the learning rate of the proposed algorithm satisfies Robbins-Monro conditions and the proposed algorithm is a k order MDP, the proposed algorithm will satisfy the conditions in [20, Theorem 1] and, hence, converges to a region. For both cases i) and ii), based on [20, Theorem 1], the proposed ESN-based learning algorithm will converge to the utility value, \( \hat{u}_j \). This completes the proof.

REFERENCES