Echo State Transfer Learning for Data Correlation Aware Resource Allocation in Wireless Virtual Reality

Mingzhe Chen*, Walid Saad†, Changchuan Yin*, and Mérouane Debbah‡
*Beijing Laboratory of Advanced Information Network,
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.
‡Mathematical and Algorithmic Sciences Lab, Huawei France R & D, Paris, France,
Email: merouane.debbah@huawei.com.

***Abstract*—In this paper, the problem of data correlation aware resource management is studied for a network of wireless virtual reality (VR) users communicating over cloud-based small cell networks (SCNs). In the studied model, the small base stations (SBSs) with limited computation resource act as VR control centers that collect the tracking information from VR users over the cellular uplink and send them to the VR users over the downlink. In such a setting, VR users may send or request the correlated or similar data (panoramic images and tracking data). This potential spatial data correlation can be factored into the resource allocation problem to reduce the traffic load in both uplink and downlink. This VR resource allocation problem is formulated as a noncooperative game that allows jointly optimizing the computation and spectrum resources, while being cognizant of the data correlation. To solve this game, a transfer learning algorithm based on the machine learning framework of echo state networks (ESNs) is proposed. Unlike conventional reinforcement learning algorithms that must be executed each time the environment changes, the proposed algorithm can intelligently transfer information on the learned utility, across time, to rapidly adapt to environmental dynamics due to factors such as changes in the users' content or data correlation. Simulation results show that the proposed algorithm achieves up to 16.7% and 18.2% gains in terms of delay compared to the Q-learning with data correlation and Q-learning without data correlation. The results also show that the proposed algorithm has a faster convergence time than Q-learning and can guarantee low delays.**

I. Introduction

Virtual reality (VR) can enable users to virtually hike the Grand Canyon or make a secret mission as a video game hero without leaving their room. However, due to the wired connections of conventional VR devices, the users are significantly restricted in the type of actions that they can take and VR applications that they can experience. To enable pervasive and truly immersive VR applications, VR systems can be operated using wireless networking technologies [1]. However, operating VR devices over wireless cellular systems such as small cell networks (SCNs) faces many challenges [1] that include tracking accuracy, extremely low delay, and effective image compression.

The existing literature has studied a number of problems related to wireless VR such as in [1]–[4]. The authors in [1] exposed the future challenges of VR systems over a wireless network. However, this work is restricted to preliminary surveys that do not provide any technical solutions for optimizing wireless VR. In [2], a channel access scheme for wireless multi-user VR system is proposed. The authors in [3] proposed an alternate current magnetic field-based tracking system to track the position and orientation of a VR user's head. However, existing works such as in [2] and [3] only focus on the improvement of one VR quality-of-service (QoS) metric such as tracking or delay. Indeed, this prior art does not develop any VR-specific model that can capture all factors of VR QoS (jointly consider uplink and downlink) and, hence, these works fall short in addressing the challenges of optimizing VR QoS for wireless users. In [4], we proposed a wireless VR model that captures the tracking accuracy, processing delay, and transmission delay and proposed a machine learning based algorithm to solve the resource allocation problem. However, this work is only focused on spectrum allocation that ignores the data correlation over the data transmission of VR users. Indeed, the sensors placed at a VR user can collect the tracking data of other users and, hence, the tracking data of VR users may have some correlation. Moreover, when the VR users are watching a football game with different perspective, the cloud only needs to transmit one 360° image to the SBS, then the SBS can rotate the image and transmit it to different users. In this case, the use of data correlation to reduce the traffic load in data transmission can improve the transmission delay.

The main contribution of this paper is to introduce a novel framework for enabling VR applications over wireless cellular networks. To the best of our knowledge, *this is the first work that jointly considers the data correlation, spectrum resource allocation, and computation resource allocation for VR over cellular networks.* Hence, our key contributions include:

- We propose a novel VR model to jointly capture the downlink and uplink transmission delay, backhaul transmission delay, and computation time thus effectively quantifying the VR delay for all users in a wireless VR network.
- For the considered VR applications over wireless, we analyze resource blocks allocation *jointly* over, the uplink and downlink and the computation resource allocation via the uplink. We formulate the problem as a noncooperative game in which the players are the small base stations (SBSs). Each player seeks to find an optimal resource allocation scheme to optimize a utility function that captures the VR delay.
- To solve this game, we propose a transfer learning algorithm based on echo state networks (ESNs) [5] to find the Nash equilibrium of the game. The proposed algorithm can intelligently transfer information on the learned utility across time, and, hence, allow adaptation to environmental dynamics due to factors such as changes in the users' data correlation.
- Simulation results show that the proposed algorithm can, respectively, yield 16.7% and 18.2% gains in terms of delay compared to Q-learning with data correlation and Q-learning without data correlation.

II. System Model and Problem Formulation

Consider the downlink and uplink transmission of a cloud-based SCN servicing a set $\mathcal{U}$ of U wireless VR users and a set $\mathcal{B}$ of B SBSs. Here, the downlink is used to transmit the VR images displayed on each user's VR device while the uplink is used to transmit the tracking information that is used to determine each VR user's location and orientation. The SBSs are connected to a cloud via *capacity-constrained backhaul* links and the SBSs serve their users using the cellular band. Here, V_F represents the maximum backhaul transmission rate for all users. Here, we focus on entertainment VR applications such as watching immersive videos and playing immersive games.

In our model, the SBSs adopt an orthogonal frequency division multiple access (OFDMA) technique and transmit over a set of $\mathcal{V}$ of V uplink resource blocks and a set of $\mathcal{S}$ of S downlink resource blocks. The coverage of each SBS is a circular area with radius r and each SBS only allocates resource blocks to the users located in its coverage range. We also assume that the resource blocks of each SBS will all be allocated to the associated users.

A. Data Correlation Model

1) Downlink Data Correlation Model: In VR wireless networks, multiple VR users may play the same immersive game with different locations and orientations. In this case, the cloud can exploit the data correlation between the users that are playing the same immersive game to reduce the traffic load of backhaul links. For example, when the users are watching the same immersive sports game, the cloud can extract the difference between the VR images of these users and will only need to transmit the data that is unique to each user to an SBS. However, when the VR users are playing different immersive games, the data correlation between the users is low and, hence, the cloud needs to transmit entire VR images to the associated VR users. In order to define the data correlation of VR images, we first assume that the number of pixels that user i needs to construct the VR images is N_i and the number of different pixels between any pair of users i and k is N_{ik}. Here, N_{ik} is calculated by the cloud using image processing methods such as motion search [6]. Then, the data correlation between user i and user k can be defined as follows:

$$\phi_{ik} = \frac{N_{ik}}{N_i + N_k}, \tag{1}$$

where N_k is the number of pixels that user k needs to construct the VR images during a period. Indeed, (1) captures the difference between the images of users i and j. From (1), we can see that when user i and user k are associated with the same SBS, the cloud only needs to transmit $N_i + N_j - (N_i + N_j)\,\phi_{ij}$ pixels to that SBS.

2) Uplink Data Correlation Model: In the uplink, the users must transmit the tracking information to the SBSs. The tracking information is collected by the sensors placed at a VR user's headset or near the VR user. It has been shown that for most data-gathering applications, the data source can be modeled as a Gaussian field [7]. The uplink data is collected by the sensors and, hence, the uplink data can be assumed to follow the Gaussian distribution. We can assume that the tracking data, X_i, collected by each VR user i is a Gaussian random variable with mean μ_i and variance σ_i^2. In wireless VR, observations from proximal VR devices are often correlated due to the dense deployment density. Hence, we consider the power exponential model [8] to capture the spatial correlation of VR tracking data. Here, the covariance σ_{ij} between user i and user j separated by distance d_{ij} is:

$$\sigma_{ij} = \operatorname{cov}(X_i, X_j) = \sigma_i \sigma_j e^{-d_{ij}^{\alpha}/\kappa}, \tag{2}$$

where α and κ capture the significance of distance variation on data correlation.

B. Delay Model

In an SCN, the VR images are transmitted from the cloud to the SBSs then to the users. The tracking information is transmitted from the users to the SBSs and processed at each corresponding SBS. In this case, the backhaul links are only used for VR image transmission and the transmission rate of each VR image from the cloud to the SBS can be given as $V_{Fi} = \frac{V_F}{U}$. Here, we assume that the backhaul transmission rate of each user is equal and we do not consider the optimization of the backhaul transmission. In a VR model, we need to capture the VR transmission requirements such as high data rate, low delay, and accurate tracking and, hence, we consider the transmission delay as the main VR QoS metric of interest. The downlink rate of user i associated with SBS j is:

$$c_{ij}(\boldsymbol{s}_{ij}) = \sum_{k=1}^{S} s_{ij,k} B \log_2(1 + \gamma_{ij,k}), \tag{3}$$

where $\boldsymbol{s}_{ij} = [s_{ij,1}, \ldots, s_{ij,S}]$ is the vector of resource blocks that SBS j allocates to user i with $s_{ij,k} \in \{1,0\}$. Here, $s_{ij,k} = 1$ indicates that resource block k is allocated to user i. $\gamma_{ij,k} = \frac{P_B h_{ij}^k}{N_0^2 + \sum\limits_{l \in \mathcal{R}^k, l \neq j} P_B h_{il}^k}$ is the signal-to-interference-plus-noise ratio (SINR) between user i and SBS j over resource block k. Here, $\mathcal{R}^k$ represents the set of the SBSs that use downlink resource block 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, N_0^2 is the variance of the Gaussian noise and $h_{ij}^k = g_{ij}^k p_{ij}^{-\beta}$ is the path loss between user i and SBS j over resource block with g_{ij}^k is the Rayleigh fading parameter, d_{ij} is the distance between user i and SBS j, and β is the path loss exponent. Based on (1) and (3), the downlink transmission delay at time slot t is:

$$D_{ij}\left(L_i\left(\phi_i^{\max}\right), \boldsymbol{s}_{ij}\right) = \frac{L_i\left(\phi_i^{\max}\right)}{c_{ij}(\boldsymbol{s}_{ij})} + \frac{L_i\left(\phi_i^{\max}\right)}{V_{FU}}, \tag{4}$$

where $L_i\left(\phi_i^{\max}\right)$ is the data that user i needs to construct a VR image during a period and $\phi_i^{\max} = \max\limits_{k \in \mathcal{U}_j, k \neq i}(\phi_{ik})$ is the maximum downlink data correlation between user i and other users associated with SBS j. Finding the maximum data correlation allows minimizing the downlink transmission data transmitted in the downlink and that will be used construct a VR image. Here, the first term is the transmission time from SBS j to user i and the second term is the transmission time from the cloud to SBS j. We assume that P_U is the transmit power of each user which is assumed to be equal for all users. The bandwidth of each uplink resource block is also B. In this case, the uplink rate of each user i associated with SBS j is:

$$c_{ij}(\boldsymbol{v}_{ij}) = \sum_{k=1}^{V} v_{ij,k} B \log_2\left(1 + \gamma_{ij,k}^{\mathrm{u}}\right), \tag{5}$$

where $\boldsymbol{v}_{ij} = [v_{ij,1}, \ldots, v_{ij,V}]$ is the vector of resource blocks that SBS j allocates to user i with $v_{ij,k} \in \{1,0\}$. $\gamma_{ij,k}^{\mathrm{u}} = \frac{P_U h_{ij}^k}{\sigma^2 + \sum\limits_{l \in \mathcal{U}^k, l \neq j} P_U h_{il}^k}$ is the SINR between user i and SBS j over resource block k with $\mathcal{U}^k$ represents the set of users that use uplink resource blocks k. In this case, the uplink transmission delay can be given by $\frac{K_i(\sigma_i^{\max})}{c_{ij}(\boldsymbol{v}_{ij})}$ where K_i is the data that needs to be transmitted and $\sigma_i^{\max} = \max\limits_{k \in \mathcal{U}_j, k \neq i} (\sigma_{ik})$ is the maximum uplink data correlation between user i and other SBS j's associated users. Similarly, finding the maximum data correlation allows minimizing the uplink transmission data that SBS j uses to determine user i's location and orientation.

In the uplink, the tracking information can be directly processed by the SBSs that have limited computation power. Here, the computation resource of each SBS, c, represents its ability to compute the tracking data. Each SBS j will allocate the total computation power to the associated users and, hence, m_{ij} is used to represent the computation power that SBS j allocates to user i with $\sum_{i \in \mathcal{U}_j} m_{ij} = m$. $\mathcal{U}_j$ represents the set of the users associated with SBS j. The computation time of SBS j that processes the tracking data collected by user i is $\frac{K_i(\sigma_i^{\max})}{m_{ij}}$ and the total uplink delay can be given by:

$$D_{ij}^{\mathrm{u}} \left(K_i(\sigma_i^{\max}), \boldsymbol{v}_{ij}, m_{ij} \right) = \frac{K_i(\sigma_i^{\max})}{c_{ij}\left(\boldsymbol{v}_{ij}\right)} + \frac{K_i(\sigma_i^{\max})}{m_{ij}}, \tag{6}$$

where the first term is the transmission time from user i to SBS j and the second term is the computation time for user i' data. Here, the computation time depends on the computation resource that SBS j allocates to each user that will affect the uplink delay.

C. Utility Function Model

In order to jointly consider the transmission delay in both uplink and downlink, we introduce a method based on the framework of multi-attribute utility theory [9] to construct an appropriate utility function to capture transmission delay in both uplink and downlink. We first introduce the utility functions of transmission delay in uplink and downlink, separately. Then, we formulate the utility function based on [9].

The utility function of downlink transmission delay is constructed based on the normalization of downlink transmission delay, which can be given by:

$$\begin{aligned} &\bar{D}_{ij} \left(L_i \left(\phi_i^{\max} \right), \boldsymbol{s}_{ij} \right) = \\ &\begin{cases} \frac{D_{ij,\max} - D_{ij}\left(L_i\left(\phi_i^{\max}\right), \boldsymbol{s}_{ij}\right)}{D_{ij,\max} - \gamma_D}, & D_{ij}\left(L_i\left(\phi_i^{\max}\right), \boldsymbol{s}_{ij}\right) \geq \gamma_D, \\ 1, & D_{ij}\left(L_i\left(\phi_i^{\max}\right), \boldsymbol{s}_{ij}\right) < \gamma_D, \end{cases} \end{aligned} \tag{7}$$

where γ_D is the maximal tolerable delay for each VR user (maximum supported by the VR system being used) and $D_{ij,\max} = \max\limits_{\boldsymbol{s}_{ij}} \left(D_{ij} \left(L_i \left(0 \right), \boldsymbol{s}_{ij} \right) \right)$ is the maximal transmission delay. From (7), we can see that, when the downlink transmission delay is smaller than γ_D, the utility value will remain at 1. This is due to the fact when the delay meets the system requirement, the network will encourage the SBSs to reallocate the resource blocks to other users. The utility function for the uplink transmission is:

$$\begin{aligned} &\bar{D}_{ij}^{\mathrm{u}} \left(K_i \left(\sigma_i^{\max} \right), \boldsymbol{v}_{ij}, m_{ij} \right) = \\ &\begin{cases} \frac{D_{ij,\max}^{\mathrm{u}} - D_{ij}^{\mathrm{u}}\left(K_i\left(\sigma_i^{\max}\right), \boldsymbol{v}_{ij}, m_{ij}\right)}{D_{ij,\max}^{\mathrm{u}} - \gamma_D}, & D_{ij}^{\mathrm{u}}(K_i(\sigma_i^{\max}), \boldsymbol{v}_{ij}, m_{ij}) \geq \gamma_D^{\mathrm{u}}, \\ 1, & D_{ij}^{\mathrm{u}}(K_i(\sigma_i^{\max}), \boldsymbol{v}_{ij}, m_{ij}) < \gamma_D^{\mathrm{u}}, \end{cases} \end{aligned} \tag{8}$$

where γ_D^{u} is the maximal tolerable delay for the VR tracking information transmission and $D_{ij,\max}^{\mathrm{u}} = \max\limits_{\boldsymbol{v}_{ij}, m_{ij}} \left(D_{ij}^{\mathrm{u}} \left(K_i \left(0 \right), \boldsymbol{v}_{ij}, m_{ij} \right) \right)$ is the maximal uplink delay. Based on (7) and (8), the total utility function that captures both downlink and uplink delay for user i associated with SBS j is:

$$\begin{aligned} &U_{ij} \left(\boldsymbol{s}_{ij}, \boldsymbol{v}_{ij}, m_{ij} \right) = \\ &\qquad \bar{D}_{ij} \left(L_i \left(\phi_i^{\max} \right), \boldsymbol{s}_{ij} \right) \bar{D}_{ij}^{\mathrm{u}} \left(K_i \left(\sigma_i^{\max} \right), \boldsymbol{v}_{ij}, m_{ij} \right). \end{aligned} \tag{9}$$

Here, $L_i \left(\phi_i^{\max} \right)$ and $K_i \left(\sigma_i^{\max} \right)$ are determined by the user association scheme. In order to capture the gain that stems from the allocation of the resource blocks and the computational capabilities, we state the following result:

Theorem 1. The utility gain of user i's delay due to an increase in the amount of allocated resource blocks and computational resources is:

i) The gain that stems from an increase in the allocated uplink resource blocks, ΔU_{ij}, is given by:

$$\Delta U_{ij} = \begin{cases} f_{\bar{D}_{ij}^{\mathrm{u}}} \left(\frac{1}{c_{ij}\left(\boldsymbol{v}_{ij}\right)} \right), & c_{ij}\left(\boldsymbol{\Delta v}_{ij}\right) \gg c_{ij}\left(\boldsymbol{v}_{ij}\right), \\ f_{\bar{D}_{ij}^{\mathrm{u}}} \left(\frac{c_{ij}\left(\boldsymbol{\Delta v}_{ij}\right)}{c_{ij}\left(\boldsymbol{v}_{ij}\right)^2} \right), & c_{ij}\left(\boldsymbol{\Delta v}_{ij}\right) \ll c_{ij}\left(\boldsymbol{v}_{ij}\right), \\ f_{\bar{D}_{ij}^{\mathrm{u}}} \left(\frac{c_{ij}\left(\boldsymbol{\Delta v}_{ij}\right)}{c_{ij}\left(\boldsymbol{v}_{ij}\right)^2 + c_{ij}\left(\boldsymbol{v}_{ij}\right) c_{ij}\left(\boldsymbol{\Delta v}_{ij}\right)} \right), & \text{else}, \end{cases} \tag{10}$$

where $f_{\bar{D}_{ij}^{\mathrm{u}}}\left(x\right) = \bar{D}_{ij} \left(L_i \left(\phi_i^{\max} \right), \boldsymbol{s}_{ij} \right) \left(\frac{K_i(\sigma_i^{\max}) x}{D_{ij,\max}^{\mathrm{u}} - \gamma_D^{\mathrm{u}}} \right)$.

ii) The gain that stems from the increase in the number of downlink resource blocks allocated to user i, ΔU_{ij}, is:

$$\Delta U_{ij} = \begin{cases} f_{\bar{D}_{ij}} \left(\frac{1}{c_{ij}\left(\boldsymbol{s}_{ij}\right)} \right), & c_{ij}\left(\boldsymbol{\Delta s}_{ij}\right) \gg c_{ij}\left(\boldsymbol{s}_{ij}\right), \\ f_{\bar{D}_{ij}} \left(\frac{c_{ij}\left(\boldsymbol{\Delta s}_{ij}\right)}{c_{ij}\left(\boldsymbol{s}_{ij}\right)^2} \right), & c_{ij}\left(\boldsymbol{\Delta s}_{ij}\right) \ll c_{ij}\left(\boldsymbol{s}_{ij}\right), \\ f_{\bar{D}_{ij}} \left(\frac{c_{ij}\left(\boldsymbol{\Delta s}_{ij}\right)}{c_{ij}\left(\boldsymbol{s}_{ij}\right)^2 + c_{ij}\left(\boldsymbol{s}_{ij}\right) c_{ij}\left(\boldsymbol{\Delta s}_{ij}\right)} \right), & \text{else}, \end{cases} \tag{11}$$

where $f_{\bar{D}_{ij}}(x) = \bar{D}_{ij}^{\mathrm{u}}(K_i\left(\sigma_i^{\max}\right), \boldsymbol{v}_{ij}, m_{ij}) \left(\frac{L_i(\phi_i^{\max}) x}{D_{ij,\max} - \gamma_D} \right)$.

iii) The gain that stems from the increase in the amount of computation resources, Δm, allocated to user i, ΔU_{ij}, is:

$$\Delta U_{ij} = \bar{D}_{ij}(L_i\left(\phi_i^{\max}\right), \boldsymbol{s}_{ij}) \left(\frac{K_i(\sigma_i^{\max}) \Delta m}{\left(D_{ij,\max}^{\mathrm{u}} - \gamma_D^{\mathrm{u}}\right)\left(m_{ij}(m_{ij} + \Delta m)\right)} \right). \tag{12}$$

Proof. For i), The gain that stems from an increase in the allocated uplink resource blocks, ΔU_{ij}, can be given by:

$$\begin{aligned} \Delta U_{ij} &= U_{ij} \left(\boldsymbol{s}_{ij}, \boldsymbol{v}_{ij} + \boldsymbol{\Delta v}_{ij}, m_{ij} \right) - U_{ij} \left(\boldsymbol{s}_{ij}, \boldsymbol{v}_{ij}, m_{ij} \right) \\ &= \bar{D}_{ij} \left(L_i \left(\phi_i^{\max} \right), \boldsymbol{s}_{ij} \right) \bar{D}_{ij}^{\mathrm{u}} \left(K_i \left(\sigma_i^{\max} \right), \boldsymbol{v}_{ij}, m_{ij} \right) \\ &\quad - \bar{D}_{ij} \left(L_i \left(\phi_i^{\max} \right), \boldsymbol{s}_{ij} \right) \bar{D}_{ij}^{\mathrm{u}} \left(K_i \left(\sigma_i^{\max} \right), \boldsymbol{v}_{ij} + \boldsymbol{\Delta v}_{ij}, m_{ij} \right). \end{aligned} \tag{13}$$

Submitting (8) and (6) into (13), (13) can be re written as follows:

$$\begin{aligned} \Delta U_{ij} &= \bar{D}_{ij} \left(L_i \left(\phi_i^{\max} \right), \boldsymbol{s}_{ij} \right) \left(\frac{ \frac{K_i(\sigma_i^{\max})}{c_{ji}(\boldsymbol{v}_{ij})} - \frac{K_i(\sigma_i^{\max})}{c_{ji}(\boldsymbol{v}_{ij} + \boldsymbol{\Delta v}_{ij})} }{D_{ij,\max}^{\mathrm{u}} - \gamma_D^{\mathrm{u}}} \right) \\ &= \bar{D}_{ij} \left(L_i \left(\phi_i^{\max} \right), \boldsymbol{s}_{ij} \right) \left(\frac{ \frac{K_i(\sigma_i^{\max}) c_{ij}(\boldsymbol{\Delta v}_{ij})}{c_{ij}(\boldsymbol{v}_{ij})^2 + c_{ij}(\boldsymbol{v}_{ij}) c_{ij}(\boldsymbol{\Delta v}_{ij})} }{D_{ij,\max}^{\mathrm{u}} - \gamma_D^{\mathrm{u}}} \right). \end{aligned} \tag{14}$$

Here, when $c_{ij}(\boldsymbol{\Delta v}_{ij}) \gg c_{ij}(\boldsymbol{v}_{ij})$, $\frac{c_{ij}(\boldsymbol{\Delta v}_{ij})}{c_{ij}(\boldsymbol{v}_{ij})^2 + c_{ij}(\boldsymbol{v}_{ij}) c_{ij}(\boldsymbol{\Delta v}_{ij})} \approx \frac{1}{c_{ij}(\boldsymbol{v}_{ij})}$, and, consequently, $\Delta U_{ij} = \frac{\bar{D}_{ij}(L_i(\phi_i^{\max}), \boldsymbol{s}_{ij}) K_i(\sigma_i^{\max})}{\left(D_{ij,\max} - \gamma_D^{\mathrm{u}}\right) c_{ij}(\boldsymbol{v}_{ij})}$. Moreover, as $c_{ij}\left(\boldsymbol{\Delta v}_{ij}\right) \ll c_{ij}\left(\boldsymbol{v}_{ij}\right)$, $\frac{c_{ij}(\boldsymbol{\Delta v}_{ij})}{c_{ij}(\boldsymbol{v}_{ij})^2 + c_{ij}(\boldsymbol{v}_{ij}) c_{ij}(\boldsymbol{\Delta v}_{ij})} \approx \frac{c_{ij}(\boldsymbol{\Delta v}_{ij})}{c_{ij}(\boldsymbol{v}_{ij})^2}$ and, consequently, $\Delta U_{ij} = \frac{\bar{D}_{ij}(L_i(\phi_i^{\max}), \boldsymbol{s}_{ij}) K_i(\sigma_i^{\max}) c_{ij}(\boldsymbol{\Delta v}_{ij})}{\left(D_{ij,\max}(\boldsymbol{v}_{ij}) - \gamma_D^{\mathrm{u}}\right) c_{ij}(\boldsymbol{v}_{ij})^2}$. For any other cases, $\Delta U_{ij} = \frac{\bar{D}_{ij}(L_i(\phi_i^{\max}), \boldsymbol{s}_{ij}) K_i(\sigma_i^{\max})}{\left(D_{ij,\max} - \gamma_D^{\mathrm{u}}\right)} \times \frac{c_{ij}(\boldsymbol{\Delta v}_{ij})}{c_{ij}(\boldsymbol{v}_{ij})^2 + c_{ij}(\boldsymbol{v}_{ij}) c_{ij}(\boldsymbol{\Delta v}_{ij})}$

Cases ii) and iii) can be proved using similar method as case i). This completes the proof. □

From Theorem 1, we can see that the allocation of spectrum and computation resource jointly determines the delay utility. Indeed, Theorem 1 provides guidance for the SBSs when they select actions in the learning algorithm that is proposed in Section III.

D. Problem Formulation

Given the defined system model, our goal is to develop an effective resource allocation scheme that allocates resource blocks and computation power to maximize the utility functions of all users. However, the maximization problem depends not only on the resource blocks allocation and computation resource allocation but also on the user associations. Moreover, the utility value of each SBS depends not only on its own choice of resource allocation scheme but also on the remaining SBSs' schemes. In addition, the data correlation among the users varies as the period changes, which will affect the resource allocation and user association. In this case, we first formulate a noncooperative game $\mathcal{G} = \left[\mathcal{R}, \{\mathcal{A}_j\}_{j\in\mathcal{R}}, \{U_j\}_{j\in\mathcal{R}}\right]$. In this game, the players are the SBSs, $\mathcal{A}_j$ represents the action set of each SBS j, and U_j is the utility function of each SBS j. Here, an action of SBS j, $\boldsymbol{a}_j$, consists of: (i) downlink resource allocation vector $\boldsymbol{s}_j = \left[\boldsymbol{s}_{1j}, \boldsymbol{s}_{2j}, \ldots, \boldsymbol{s}_{\mathcal{U}_j j}\right]$, (ii) uplink resource allocation vector $\boldsymbol{v}_j = \left[\boldsymbol{v}_{1j}, \boldsymbol{v}_{2j}, \ldots, \boldsymbol{v}_{\mathcal{U}_j j}\right]$, and (iii) computation resource allocation vector $\boldsymbol{m}_j = \left[m_{1j}, m_{2j}, \ldots, m_{\mathcal{U}_j j}\right]$. Here, $m_{ij} \in \mathcal{M}, i \in \mathcal{U}_j$ where $\mathcal{M} = \left\{\frac{c}{M}, \frac{2c}{M}, ..., c\right\}$ is a finite set of M level fractions of SBS j's total computation resource m_j. We assume that each SBS j adopts one action at each time slot t. Then, the utility function of each SBS j can be given by:

$$u_j\left(\boldsymbol{a}_j, \boldsymbol{a}_{-j}\right) = \frac{1}{T}\sum_{t=1}^{T}\sum_{i\in\mathcal{U}_j} U_{ij,t}\left(\boldsymbol{s}_{ij}, \boldsymbol{v}_{ij}, \boldsymbol{c}_{ij}\right), \tag{15}$$

where $\boldsymbol{a}_j \in \mathcal{A}_j$ is an action of SBS j and $\boldsymbol{a}_{-j}$ denotes the action profile of all SBSs other than SBS j. Indeed, (15) captures the average utility value of each SBS j. Let $\pi_{j,\boldsymbol{a}_{ij}} = \frac{1}{T}\sum_{t=1}^{T}\mathbb{1}_{\{\boldsymbol{a}_{j,t}=\boldsymbol{a}_{ij}\}} = \Pr\left(\boldsymbol{a}_{j,t} = \boldsymbol{a}_{ij}\right)$ be the probability of SBS j using action $\boldsymbol{a}_{ij}$. Here, $\boldsymbol{a}_{j,t}$ represents the action that SBS j uses at time t and $\boldsymbol{a}_{j,t} = \boldsymbol{a}_{ij}$ denotes that SBS j adopts action $\boldsymbol{a}_{ij}$ at time t. $\boldsymbol{\pi}_j = \left[\pi_{j,\boldsymbol{a}_{1j}}, \ldots, \pi_{j,\boldsymbol{a}_{|\mathcal{A}_j|j}}\right]$ is the action selection strategy of SBS j with $|\mathcal{A}_j|$ being the number of actions of SBS j. Based on the definition of the strategy, the utility function in (15) is given by:

$$u_j\left(\boldsymbol{a}_j, \boldsymbol{a}_{-j}\right) = \frac{1}{T}\sum_{t=1}^{T} U_{j,t}\left(\boldsymbol{a}_j, \boldsymbol{a}_{-j}\right) = \sum_{\boldsymbol{a}\in\mathcal{A}}\left(U_j\left(\boldsymbol{a}_j, \boldsymbol{a}_{-j}\right)\prod_{j\in\mathcal{B}}\pi_{j,\boldsymbol{a}_j}\right), \tag{16}$$

where $\boldsymbol{a} \in \mathcal{A}$ with $\mathcal{A}$ being the action set of all SBSs.

Given the proposed model, our goal is to solve the proposed resource allocation game. A solution for this game is the mixed-strategy Nash equilibrium (NE), formally defined as follows [11]: A mixed strategy profile $\boldsymbol{\pi}^* = \left(\boldsymbol{\pi}_1^*, \ldots, \boldsymbol{\pi}_B^*\right) = \left(\boldsymbol{\pi}_j^*, \boldsymbol{\pi}_{-j}^*\right)$ is a *mixed-strategy Nash equilibrium* if, $\forall j \in \mathcal{R}$ and $\boldsymbol{\pi}_j$, we have:

$$u_j\left(\boldsymbol{\pi}_j^*, \boldsymbol{\pi}_{-j}^*\right) \geq u_j\left(\boldsymbol{\pi}_j, \boldsymbol{\pi}_{-j}^*\right), \tag{17}$$

where $u_j\left(\boldsymbol{\pi}_n, \boldsymbol{\pi}_{-n}\right) = \sum_{\boldsymbol{a}\in\mathcal{A}} U_j\left(\boldsymbol{a}_j, \boldsymbol{a}_{-j}\right)\prod_{j\in\mathcal{B}}\pi_{j,\boldsymbol{a}_j}$ is the expected utility of SBS j when it selects the mixed strategy $\boldsymbol{\pi}_j$.For our game, the mixed-strategy NE for the SBSs represents a solution of the game at which each SBS j can minimize the delay for its associated users, given the actions of its opponents.

III. Echo State Networks for Self-Organizing Resource Allocation

Next, we introduce a transfer reinforcement learning (RL) algorithm that can be used to find an NE of the VR game. To satisfy the delay requirement for the VR transmission, we propose a transfer RL algorithm based on the neural networks framework of echo state networks (ESN) [16]. Traditional RL algorithms such as Q-learning typically rely on a Q-table to record the utility value. However, as the number of players and actions increases, the number of utility values that the Q-table needs to include will increase exponentially and, hence, the Q-table may not be able to record all of the needed utility values. However, the proposed algorithm uses a utility function approximation method to record the utility value and, hence, it can be used for large networks and large utility spaces. Moreover, a dynamic network in which the users' computation resource and data correlation may change across the time, traditional RL algorithms need to be executed each time the network changes. However, the proposed ESN transfer RL algorithm can find the relationship of the utility functions when the environment changes. After learning this relationship, the proposed algorithm can use the historic learning result to find a mixed strategy NE.

The proposed transfer RL algorithm consists of two components: (i) ESN-based RL algorithm and (ii) ESN-based transfer learning algorithm. The ESN-based RL algorithm is based on our work in [4], and, thus, here, we just introduce the ESN-based transfer learning algorithm.

We first assume that, before the users' state information changes, the strategy, action, and utility of each SBS j are $\boldsymbol{\pi}_j$, $\boldsymbol{a}_j$ and $\hat{u}_j\left(\boldsymbol{a}_j, \boldsymbol{a}_{-j}\right)$, while the strategy, action, and utility of SBS j, after the users' state information changes, are $\boldsymbol{\pi}_j$, $\boldsymbol{a}_j$, and $\hat{u}'_j\left(\boldsymbol{a}_j, \boldsymbol{a}_{-j}\right)$. Since the number of users associated with SBS j is unchanged, the sets of action and strategy of SBS j will not change when the users' state information changes. In this case, the proposed ESN-based transfer learning algorithm is used to find the relationship between $\hat{u}_j\left(\boldsymbol{a}_j, \boldsymbol{a}_{-j}\right)$ and $\hat{u}'_j\left(\boldsymbol{a}_j, \boldsymbol{a}_{-j}\right)$ when SBS j only knows $\hat{u}_j\left(\boldsymbol{a}_j, \boldsymbol{a}_{-j}\right)$. This means that the proposed algorithm can transfer the information from the already learned utility $\hat{u}_j\left(\boldsymbol{a}_j, \boldsymbol{a}_{-j}\right)$ to the new utility $\hat{u}'_j\left(\boldsymbol{a}_j, \boldsymbol{a}_{-j}\right)$ that must be learned. The ESN-based transfer learning algorithm of each SBS j consists of three components: (a) input, (b) output, and (c) ESN model, which are given by:

- *Input:* The ESN-based transfer learning algorithm takes the strategies of the SBSs and the action of SBS j uses at time t as input which is given by $\boldsymbol{x}'_{t,j} = \left[\pi_1, \cdots, \pi_B, \boldsymbol{a}_{j,t}\right]^{\mathrm{T}}$.
- *Output:* The output of the ESN-based transfer learning algorithm at time t is the deviation of the utility values when the users' information changes $y'_{j,t} = \hat{u}'_j\left(\boldsymbol{a}_{j,t}\right) - \hat{u}_j\left(\boldsymbol{a}_{j,t}\right)$.
- *ESN Model:* An ESN model is used to find the relationship between the input $\boldsymbol{x}'_{t,j}$ and output $y'_{t,j}$. The ESN model consists of the output weight matrix $\boldsymbol{W}_j^{'\text{out}} \in \mathbb{R}^{1\times N_w}$ and the dynamic reservoir containing the input weight matrix $\boldsymbol{W}_j^{'\text{in}} \in \mathbb{R}^{N_w\times B+1}$, and the recurrent matrix $\boldsymbol{W}'_j \in \mathbb{R}^{N_w\times N_w}$ with N_w being the number of the dynamic reservoir units. Here, the dynamic reservoir is used to store historic ESN information that includes

TABLE I
ESN-BASED LEARNING ALGORITHM FOR RESOURCE ALLOCATION

Inputs: $\boldsymbol{x}_{j,t}$ and $\boldsymbol{x}'_{j,t}$

Initialize: $\boldsymbol{W}_j^{\mathrm{in}}$, $\boldsymbol{W}_j$, $\boldsymbol{W}_j^{\mathrm{out}}$, $\boldsymbol{W}_j^{'\mathrm{in}}$, $\boldsymbol{W}'_j$, $\boldsymbol{W}_j^{'\mathrm{out}}$, $\boldsymbol{y}_j = 0$, and $y'_j = 0$.

for each time t **do**.

(a) Estimate the value of the utility function $\hat{u}_{j,t}$ based on (19).

if $t == 1$

(b) Set the mixed strategy $\boldsymbol{\pi}_{j,t}$ uniformly.

else

(c) Set the mixed strategy $\boldsymbol{\pi}_{j,t}$ based on the ε-greedy exploration.

end if

(d) Broadcast the index of the mixed strategy to other SBSs.

(e) Receive the index of the mixed strategy as input $\boldsymbol{x}_{j,t}$.

(f) Perform an action based on the mixed strategy.

(g) Use the index of the mixed strategies and action as input $\boldsymbol{x}'_{j,t}$.

(h) Estimate the value of the difference of utility function $y'_{j,t}$.

(i) Update the dynamic reservoir state $\boldsymbol{\mu}_{j,t}$.

(j) Update the output weight matrix $\boldsymbol{W}_j^{\mathrm{out}}$ based on $y'_{j,t}$.

end for

TABLE II
SYSTEM PARAMETERS

Parameter	Value	Parameter	Value
F	1000	P_B	20 dBm
B	2 MHz	S, V	5, 5
N_w	1000	σ^2	-95 dBm
N_v	6	λ, λ'	0.03, 0.3
m	5	r_B	30 m
α	2	V_F	100 Gbit/s

input, reservoir state, and output. This information is used to build the relationship between the input and output. The update process of the dynamic reservoir will be given by:

$$\boldsymbol{\mu}'_{j,t} = f\left(\boldsymbol{W}'_j \boldsymbol{\mu}'_{j,t-1} + \boldsymbol{W}_j^{'\mathrm{in}} \boldsymbol{x}'_{j,t}\right). \tag{18}$$

where $f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ is the tanh function. Based on the dynamic reservoir state, the ESN-based transfer learning algorithm will combine with the output weight matrix to approximate the deviation of the utility value, which can be given by:

$$y'_{j,t} = \boldsymbol{W}_{j,t}^{'\mathrm{out}} \boldsymbol{\mu}'_{j,t}, \tag{19}$$

where $\boldsymbol{W}_{j,t}^{'\mathrm{out}}$ is the output weight matrix at time slot t.

$$\boldsymbol{W}_{j,t+1}^{'\mathrm{out}} = \boldsymbol{W}_{j,t}^{'\mathrm{out}} + \lambda' \left(\hat{u}'_j\left(\boldsymbol{a}_{j,t}\right) - \hat{u}_j\left(\boldsymbol{a}_{j,t}\right) - y'_{j,t}\right) \boldsymbol{\mu}'^{\mathrm{T}}_{j,t}, \tag{20}$$

where λ' is the learning rate, and $\hat{u}'_{j,t}$ is the actual deviation between two utility values. In this case, the ESN-based transfer learning algorithm can find the relationship between the utility functions when the users' state information changes and, hence, reduce the iterations of the RL algorithm to learn the new utility values. The proposed, distributed ESN-based learning algorithm performed by each SBS j is summarized in Table I. The proposed algorithm is guaranteed to converge to an NE and this convergence follows from [4].

IV. SIMULATION RESULTS

For our simulations, we consider a cloud-based 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 rate requirement of VR transmission is 25.32 Mbit/s [4]. The detailed parameters are listed in Table III. For comparison purposes, we use ESN algorithm and a baseline Q-learning algorithm in [4].

Fig. 1 shows how the average delay per user changes with the number of SBSs varies. Fig. 1 shows that, as the number of SBSs increases, the average delay of all algorithms decreases, then increases. 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 decreases. However, as the number of SBSs keeps increasing, the interference will also increase. Fig. 1 also shows that our algorithm achieves up to 16.7% and 18.2% gains in terms of average delay compared to the Q-learning with data correlation and Q-learning without data correlation for 6 SBSs. This is due to the fact that our algorithm can transfer information across time. From Fig. 1, we can also see that the deviation between Q-learning algorithms decreases as the number of SBSs changes. This implies that as the number of SBSs increases, the number of users associated with each SBS decreases and, hence, the data correlation of users decreases. Fig. 1 also shows that the delay gain of the proposed algorithm is small compared with ESN algorithm. However, the proposed algorithm can converge much faster as shown in Fig. 2.

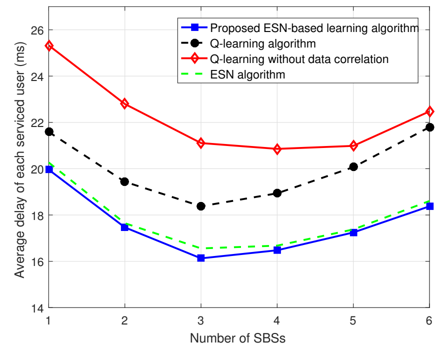

Fig. 1. Average delay of each user vs. number of SBSs.

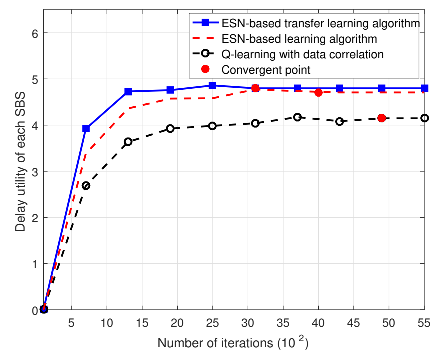

Fig. 2. Convergence of the proposed algorithm and Q-learning.

Fig. 2 shows the number of iterations needed till convergence for the proposed approach, ESN algorithm, and Q-learning with data correlation when the users' information changes. In this figure, we can see that, as time elapses, the delay utilities for all considered algorithms increase until convergence to their final values. Fig. 2 also shows that the proposed algorithm achieves, respectively, 22.5% and 36% gains in terms of the number of the iterations needed to reach convergence compared to ESN algorithm and Q-learning. This implies that the proposed algorithm can apply the already learned utility value to the new utility value that must be learned as the users' information changes.

V. CONCLUSION

In this paper, we have proposed a novel resource allocation framework for optimizing delay for wireless VR services with data correlation. We have formulated the problem as a noncooperative game and we have proposed a novel transfer learning algorithm based on echo state networks to solve the game. The proposed learning algorithm can use the existing learning result to directly find the optimal resource allocation when the users' state information changes and, hence, can quickly converge to a mixed-strategy NE. Simulation results have shown that the proposed algorithm has a faster convergence time than Q-learning and guarantees low delays for VR services.

REFERENCES

[1] E. Baştuğ, M. Bennis, M. Médard, and M. Debbah, "Towards interconnected virtual reality: Opportunities, challenges and enablers," *arXiv preprint arXiv:1611.05356*, 2016.

[2] J. Ahn, Y. Yong Kim, and R. Y. Kim, "Delay oriented VR mode WLAN for efficient wireless multi-user virtual reality device," in *Proc. of IEEE International Conference on Consumer Electronics*, Las Vegas, NV, USA, March 2017.

[3] M. Singh and B. Jung, "High-definition wireless personal area tracking using AC magnetic field for virtual reality," in *Proc. of IEEE Virtual Reality*, Los Angeles, California, USA, March 2017.

[4] M. Chen, W. Saad, and C. Yin, "Virtual reality over wireless networks: Quality-of-service model and learning-based resource management," *available online: arxiv.org/abs/1703.04209*, Mar. 2017.

[5] H. Jaeger, "Short term memory in echo state networks," in *GMD Report*, 2001.

[6] J. F. Yang, S. C. Chang, and C. Y. Chen, "Computation reduction for motion search in low rate video coders," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 10, pp. 948–951, Oct. 2002.

[7] N. Cressie and C. K. Wikle, *Statistics for spatio-temporal data*, John Wiley & Sons, 2015.

[8] M. C. Vuran, O. B. Akan, and I. F. Akyildiz, "Spatio-temporal correlation: theory and applications for wireless sensor networks," *Computer Networks*, vol. 45, no. 3, pp. 245–259, June 2004.

[9] A. E. Abbas, "Constructing multiattribute utility functions for decision analysis," *INFORMS Tutorials in Operations Research*, pp. 62–98, Oct. 2010.

[10] M. Chen, W. Saad, C. Yin, and M. Debbah, "Echo state transfer learning for data correlation aware joint computation and resource allocation in wireless virtual reality," *available online: http://resume.walid-saad.com/pdf/extendedasilomarpaper.pdf*, May 2017.

[11] Z. Han, D. Niyato, W. Saad, T. Basar, and A. HjÃ¸rungnes, *Game Theory in Wireless and Communication Networks: Theory, Models, and Applications*, Cambridge University Press, 2012.