Human-in-the-Loop Wireless Communications: Machine Learning and Brain-Aware Resource Management

Ali Taleb Zadeh Kasgari, Walid Saad, Merouane Debbah

Human-in-the-Loop Wireless Communications: Machine Learning and Brain-Aware Resource Management

Ali Taleb Zadeh Kasgari[1], Walid Saad[1], and Mérouane Debbah[2]

[1] Wireless@VT, Electrical and Computer Engineering Department, Virginia Tech, VA, USA, Emails:{alitk,walids}@vt.edu.

[2] Mathematical and Algorithmic Sciences Lab, Huawei France R&D, Paris, France, and CentraleSupelec, Universite Paris-Saclay, Gif-sur-Yvette, France, Email: merouane.debbah@huawei.com.

Abstract

Human-centric applications such as virtual reality and immersive gaming will be central to the future wireless networks. Common features of such services include: a) their dependence on the human user's behavior and state, and b) their need for more network resources compared to conventional cellular applications. To successfully deploy such applications over wireless and cellular systems, the network must be made cognizant of not only the quality-of-service (QoS) needs of the applications, but also of the perceptions of the *human users* on this QoS. In this paper, by explicitly modeling the limitations of the human brain, a concrete measure for the delay perception of human users in a wireless network is introduced. Then, a novel learning method, called probability distribution identification, is proposed to find a probabilistic model for this delay perception based on the brain features of a human user. The proposed learning method uses both supervised and unsupervised learning techniques to build a Gaussian mixture model of the human brain features. Given a model for the delay perception of the human brain, a novel brain-aware resource management algorithm based on Lyapunov optimization is proposed for allocating radio resources to human users while minimizing the transmit power and taking into account the reliability of both machine type devices and human users. The proposed algorithm is shown to have a low complexity. Moreover, a closed-form relationship between the reliability measure and wireless physical layer metrics of the network is derived. Simulation results using real data from actual human users show that a brain-aware approach can yield savings of up to 78% in power compared to the system

A preliminary version of this work appeared in the proceedings of the 51th Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA [1].

This research was supported by the U.S. National Science Foundation under Grants CNS-1460316 and IIS-1633363.

that only considers QoS metrics. The results also show that, compared with QoS-aware, brain-unaware systems, the brain-aware approach can save substantially more power in low-latency systems.

I. Introduction

The next generation of wireless services is expected to be highly human centric. Examples include virtual reality and interactive/immersive gaming [2], [3]. In order to cope with the quality-of-service (QoS) needs of such human-centric applications, in terms of data rate and ultra-low latency, wireless networks will have to allocate and exploit substantially more radio resources by leveraging heterogeneous spectrum bands across low and high frequencies [4]. However, even though allocating heterogeneous spectrum resources can potentially increase the raw QoS, given the human-centric nature of such emerging applications, their users may not be able to perceive the improved QoS, due to the cognitive limitations of the human brain [5]. Indeed, many empirical studies (anecdotal and otherwise) have shown that the limitations on the human brain can be translated into a limitation on how wireless users translate QoS into actual quality-of-experience (QoE) [6]–[8]. For example, the human brain may not be able to perceive any difference between videos transmitted with different QoS (e.g., rates or delays) [8], [9]. Hence, in order to deploy these services over wireless networks, such as 5G cellular systems, there is a need to enable the system to be strongly cognizant of the human user in the loop. In particular, to deliver such immersive, human-centric services, the network must tailor the usage and optimization of wireless resources to the intrinsic features of its human users such as their behavior and brain processing limitations. By doing so, the network can potentially save resources, accommodate more users, and provide a more realistic QoE to its users.

Developing resource management mechanisms that can cater to intrinsic needs of wireless users and their context (e.g., device features or social metrics) has recently been studied in [4], [10]–[17]. In [10], a context-aware scheduling algorithm for 5G systems is proposed. This algorithm exploits the context information of user equipments (UEs), such as battery level, to save energy in the system while satisfying the QoS requirements of users. The authors in [11], proposed a user-centric resource allocation framework for ultra-dense heterogeneous networks. Context-aware resource allocation for heterogeneous cellular networks is also studied in [4], [12], and [13]. In [4], a novel approach to context-aware resource allocation in small cell networks is introduced. Both wireless physical layer metrics and the social ties of human users are exploited in [4] to allocate wireless resource blocks. Proactive caching using context information from social networks is studied in [14]. In this work, it is shown that such a socially-aware caching technique reduces the peak traffic in 5G networks. Other context-aware resource allocation algorithms are also studied in [15]–[17]. However, despite this surge in literature on context-aware networking [4], [10]–[17], this prior art is still reliant on device-level features and is agnostic to the human users and their features (e.g., brain limitation or behavior) and, hence, they can potentially waste network resources as they can still allocate more resources to human users that cannot perceive the associated QoS gains, due to cognitive brain limitations.

A general framework for modeling the intelligence of communication systems which serve humans is proposed in [18]. The author defines intelligence in terms of predicting and serving human demands in advance. However, the work in [18] does not account for the cognitive limitations of a human brain. Moreover, demand prediction, as done in [18], will not be sufficient to capture the full spectrum of the human user limitations and behavior. By being aware of brain limitations of each user, the network can provide a unique experience for each user and optimize its performance. For example, an increase in the delay of a wireless system may have different effects on the QoE perceived by different human users. In particular, such different delay perceptions can potentially be exploited by the cellular network to minimize power consumption and reduce the amount of wasted resources. To our best knowledge, no existing work has studied the impact of such disparate brain delay perceptions on wireless resource allocation.

The main contribution of the paper is, thus, a *novel brain-aware learning and resource management framework* that explicitly factors in the brain state of human users during resource allocation in a cellular network which has both human and machine type devices. In particular, we formulate the brain-aware resource allocation problem using a joint learning and optimization framework. First, we propose a novel learning algorithm to identify the delay perceptions of a human brain. This learning algorithm employs both supervised and unsupervised learning to identify the brain limitations and also creates a statistical model for these limitations based on Gaussian mixture models. Then, using Lyapunov optimization, we address the resource allocation problem with time varying QoS requirements that captures the learned delay perception. Using this approach, the network can allocate radio resources to human users while considering the reliability of both machine type devices and human users. We then identify a closed-form relationship between system reliability and wireless physical layer metrics and derive a closed-form expression for the reliability as a function of the human brain's delay perception. Simulation results using real data show that the proposed brain-aware approach can substantially save power in the network while preserving the reliability of the users, particularly in low latency applications. In particular, the results show that the proposed brain-aware approach can yield power savings of up to 78% compared to a conventional, brain-unaware system.

The rest of the paper is organized as follows. Section II introduces the system model. Sections III and IV present the proposed learning algorithm and resource allocation framework, respectively. Section V presents the simulation results and conclusions are drawn in Section VI.

II. System Model and Problem Formulation

Consider the downlink of a cellular network with humans-in-the-loop having a single base station (BS) serving a set $\mathcal{H}$ of N human users with their UEs and a set $\mathcal{M}$ of M machine type devices (MTDs). Each UE or MTD can have a different application with different QoS requirements such as sending a command to an actuator (for an MTD) or playing a 3D interactive game (for a UE). We consider a time-slotted system with each slot duration being equal to the LTE transmission time interval (TTI). We define $\mathcal{K}$ as the set of K resource blocks (RBs). In our model, the packets associated with user $i \in \mathcal{H} \cup \mathcal{M}$ arrive at the BS according to independent Poisson processes with rate $a_i(t)$. The lengths $l_i, \forall i \in \mathcal{H} \cup \mathcal{M}$ of the packets follow an exponential distribution. Hence, each user's buffer at the BS will follow an M/M/1 queuing model. The total queuing and transmission delay of each user i is $D_i(t) = q_i(t) + \frac{l_i}{r_i(t)}$. The data rate for each user is given by:

$$r_i(t) = B \sum_{j=1}^{K} \rho_{ij}(t) \log_2 \left(1 + \frac{p_{ij}(t) h_{ij}(t)}{\sigma^2}\right), \tag{1}$$

where $p_{ij}(t)$ is the transmit power between the BS and user i over RB j at time t and $h_{ij}(t)$ is the time-varying Rayleigh fading channel gain. In (1) $\rho_{ij}(t) = 1$ if RB j is allocated to user i at time slot t, and $\rho_{ij}(t) = 0$, otherwise. B is the bandwidth of each RB.

The BS seeks to allocate RBs and power to the users according to their delay needs and their channel state. The delay that MTD i can tolerate is β_i^m, i.e., $D_i^{\max}(\beta_i^m) = \beta_m$. β_i^m is known to the system. The delay perception of the brain of a human user who is using a given UE is captured by $\beta_i(t)$. $\beta_i(t)$ essentially represents a delay perception threshold for human user i at time t. If we decrease the delay below the threshold $\beta_i(t)$, the human user will not be able to discern the difference. This delay perception is determined by the capabilities of the human brain. *By explicitly accounting for the cognitive limitations of the human brain, the BS can better allocate resources to the users that need it, when they can actually use it*. This is in contrast to conventional brain-agnostic networks [4], [18] in which resources may be wasted, as they are allocated only based on application QoS without being aware on whether the human user can indeed process the actual application's QoS target.

We pose this resource allocation problem as a power minimization problem that is subject to a brain-aware QoS constraint on the latency:

$$\min_{\boldsymbol{\rho}(t),\boldsymbol{P}(t)} \quad \sum_{j\in\mathcal{K}}\Big[\sum_{i\in\mathcal{H}}\bar{P}_i^j+\sum_{i\in\mathcal{M}}\bar{P}_i^j\Big], \tag{2a}$$

$$\text{s.t.} \quad \Pr\big\{D_i(t)\geq D_i^{\max}\big(\beta_i(t)\big)\big\}\leq \epsilon_i\big(\beta_i(t)\big), \qquad \forall i\in\mathcal{H}\cup\mathcal{M}, \tag{2b}$$

$$p_{ij}(t)\geq 0,\ \ \rho_{ij}(t)\in\{0,1\} \qquad \forall i\in\mathcal{H}\cup\mathcal{M}, j\in\mathcal{K}, \tag{2c}$$

$$\sum_{i\in\mathcal{H}\cup\mathcal{M}}\rho_{ij}(t)=1, \qquad \forall j\in\mathcal{K}, \tag{2d}$$

where $\boldsymbol{\rho}(t)$ is an $(M+N)\times K$ matrix having each element $\rho_{ij}(t)$. $\boldsymbol{P}(t)$ is an $(M+N)\times K$ matrix with each element $p_{i,j}(t)$, representing the instantaneous power allocated to user i on RB j. The term $\bar{P}_i^j=\lim_{t\rightarrow\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\rho_{ij}p_{ij}(\tau)$ is the time average of the power allocated to user i on RB j. $D_i^{\max}\big(\beta_i(t)\big)$ is the maximum tolerable delay, and $1-\epsilon_i\big(\beta_i(t)\big)$ is the reliability of user i. We define *reliability* as the proportion of time during which the delay of a given user does not exceed a threshold. For notational convenience, hereinafter, we use the terms $D_i^{\max}\left(\beta_i(t)\right)$ and $D_i^{\max}$ interchangeably. The key difference between our problem formulation and conventional RB allocation problems is seen in the QoS delay requirement in (2b). In (2b), the network explicitly accounts for the human brain's (and the MTDs') delay needs. By taking into account the features of the brain of the human UEs, the network can avoid wasting resources. This waste of resources can stem from allocating more power to a UE, solely based on the application QoS, while ignoring how the brain of the human carrying the UE perceives this QoS. Clearly, ignoring this human perception can lead to inefficient resource management.

For finding $\beta_i(t)$, we propose a machine learning algorithm to identify the human brain delay perception. Each human user has p features, (e.g., age, occupation, location) assumed to be known to the BS. This time-varying feature vector is denoted by $\boldsymbol{x}_i(t)\in\mathbb{R}^p$. We develop a learning algorithm to build a model that maps these features to $\beta_i(t)$ for each user. We then show that being aware of $\beta_i(t)$ can help the resource allocation algorithm to save a significant amount of resources for low-latency systems. Here, we assume that BS has the access to the user features $\boldsymbol{x}_i(t)$. In practice, the BS can collect such data whenever a given user registers in the network or by using the sensors of a user's mobile device.

III. Machine Learning for Probability Distribution Identification

To find the mapping $\beta_i(t) = f(\boldsymbol{x}_i(t))$ between human features $\boldsymbol{x}_i(t)$ and the delay perception of the brain, we use supervised learning [19], [20]. Since reliability is a key factor in a communication system, we need a supervised learning algorithm that not only predicts $\beta_i(t)$ as function of $\boldsymbol{x}_i(t)$, but also gives a measure of reliability for this prediction. Hence, we cannot rely on conventional supervised learning methods, such as neural networks [20]. Here, reliability of predictions is defined as the probability that the prediction of $\beta_i(t)$ lies within a certain range of the true values for $\beta_i(t)$.

To this end, we propose a novel supervised learning mechanism dubbed as *probability distribution identification (PDI) method* that can find the mapping between the features of a human user and the human brain's perception on delay, as captured by $\beta_i(t)$, while quantifying the reliability of this mapping. We will use this reliability to determine the overall reliability of our system. As discussed in [21], the delay perception of a human brain typically follows a multi-modal distribution. As a result, we design the proposed PDI approach to capture such a model and find the different modes of a human brain. Then, using the distribution of the brain delay, the PDI approach can find the *effective delay* of the human brain. This effective delay determines relationship between $\beta_i(t)$ and $\boldsymbol{x}_i(t)$ along with its reliability.

Consider a dataset $\{\boldsymbol{x}_1(t), \cdots, \boldsymbol{x}_n(t)\}$, where $\boldsymbol{x}_i(t) \in \mathbb{R}^p$ is one sample data vector. The elements of $\boldsymbol{x}_i(t)$ are features which can be both categorical (such as gender) and numerical (such as age). For each input vector $\boldsymbol{x}_i(t)$, we have a corresponding output value of delay perception $\beta_i(t)$. This data can be collected using experiments or surveys such as those in [22]. Since we can remove the data's time dependency of the data using time-series techniques, hereinafter, we use $\boldsymbol{x}$ instead of $\boldsymbol{x}(t)$. This dataset can be represented by a matrix $\boldsymbol{X} \in \mathbb{R}^{n\times p}$, where $\boldsymbol{x}_i^T$ is row i of $\boldsymbol{X}$. Using PDI, we first create an $n \times (p+1)$ dataset matrix $\boldsymbol{W}$:

$$\boldsymbol{W} = [\boldsymbol{X} \| \boldsymbol{\beta}] = \begin{bmatrix} \boldsymbol{w}_1^T \\ \vdots \\ \boldsymbol{w}_n^T \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}_1^T & \beta_1(t) \\ \vdots & \vdots \\ \boldsymbol{x}_n^T & \beta_n(t) \end{bmatrix}, \tag{3}$$

where $\boldsymbol{w}_i \in \mathbb{R}^{p+1}$ is a vector of the delay perception $\beta_i(t)$ and p other correlated features of the human brain. Then, we fit a Gaussian mixture model (GMM) of m modes to our dataset using the expectation-maximization (EM) algorithm [23]. In other words, we adopt an EM method for learning the joint probability distribution using the joint dataset $\boldsymbol{W}$, i.e., $p(\boldsymbol{x}, \beta_i(t))$. After finding the probability distribution $p(\boldsymbol{x}, \beta_i(t))$, we are able to cluster the data samples and find m modes in the data. Then, each data vector $\boldsymbol{x}_i$ is labeled based on its cluster so that each $\boldsymbol{x}_i,\ i = 1, \cdots, n$ has a label in the cluster set $\mathcal{C} = \{1, \cdots, m\}$. Subsequently, our dataset is labeled using its cluster number. In the next step, we train a classifier such that it finds a mapping between the input data $\boldsymbol{x}_i$ and its cluster number. *These cluster numbers will correspond to the modes of the human brain that determine its effective delay perception.* When this proposed learning approach is deployed in a wireless network, each user will be classified after connecting to the BS, and its brain mode will be identified. Then, each user's mode can be used to derive a probabilistic model of its delay perception. Before delving into the PDI method for finding the brain's effective delay, we describe the Gaussian mixture model that we use for the human brain.

A multi-modal stochastic model is assumed for the brain features $\boldsymbol{w}_i$ for user i. The proposed distribution for $\boldsymbol{w}_i$ is given by [19]:

$$p(\boldsymbol{w}_i) = \sum_{z} p(z) p(\boldsymbol{w}_i|z) = \sum_{k=1}^{L} \pi_k f(\boldsymbol{w}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \tag{4}$$

where $f(\boldsymbol{w}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ is the probability density function for a multivariate normal distribution with mean vector $\boldsymbol{\mu}_k$ and covariance matrix $\boldsymbol{\Sigma}_k$. $\boldsymbol{\Sigma}_k$ and $\boldsymbol{\mu}_k$ represent the covariance matrix and mean vector for mode k of the human brain, respectively. π_k is the mixing weight of mode $z = z_k$. $p(\boldsymbol{w})$ is the distribution of data points $\boldsymbol{w}_i$. L is total number of modes in the GMM. The human brain will be in mode k with probability π_k, and its features are generated using a multivariate normal distribution with mean and covariance $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$, respectively. The posterior probability, i.e. *responsibility*, for mode k will be:

$$r(z_k) = \frac{\pi_k f(\boldsymbol{w}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{L} f(\boldsymbol{w}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}. \tag{5}$$

This responsibility can be used for clustering the data as well. After fitting the GMM on the dataset, we can find the mode with highest responsibility for each data point and assign the data to this mode. The EM algorithm is used to find these parameters based on a real-time human brain behavior [23]. The log likelihood function for our dataset can be written as:

$$\ln \mathcal{L}(\boldsymbol{\Sigma}, \mu, \pi|\boldsymbol{w}) = \ln p(\boldsymbol{w}|\boldsymbol{\Sigma}, \mu, \pi) = \sum_{i} \ln \sum_{k=1}^{L} \pi_k f(\boldsymbol{w}_i|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k). \tag{6}$$

The likelihood function in (6) has singularities and, hence, it is infeasible to find parameters π_k, $\boldsymbol{\Sigma}_k$, and $\boldsymbol{\mu}_k$. The EM algorithm is proposed in [24] to maximize the likelihood function for a

Gaussian mixture model. We first initialize $\boldsymbol{\Sigma}_k$, $\boldsymbol{\mu}_k$, and π_k randomly. Next, we find the responsibility for each mode using (5). Then, we reestimate parameters using current responsibilities. Finally, the likelihood in (6) is maximized with respect to $\boldsymbol{\Sigma}_k$, $\boldsymbol{\mu}_k$,and π_k.

In the next step, we use the derived joint distribution of human features and its associated delay perception to derive a probabilistic model for the human insensitivity of the delay. In other words, we find the relationship between a certain level of the delay $D_i^{\min}$ and the probability $\Pr(\beta_i(t) > D_i^{\min})$. In the proposed PDI, along with the supervised learning component previously explained, we also propose to use an unsupervised learning step to measure the reliability of our predictions. In the unsupervised learning step, the data will be labeled based on the GMM as follows. For each feature vector $\boldsymbol{w}_i$, the responsibilities $r(z_k) = p(z_k = 1|\boldsymbol{w}_i)$ are found using the EM algorithm. Then, the most probable mode is assigned as the label of this data, i.e.,

$$c(\boldsymbol{w}_i) = \arg\max_k \, p(z_k = 1|\boldsymbol{w}_i). \tag{7}$$

The output of the unsupervised learning step, c_i, is used for training the supervised learning model. We will form an output vector $\boldsymbol{y}$ which is defined as $\boldsymbol{y} = \begin{bmatrix} c(\boldsymbol{w}_1) & \cdots & c(\boldsymbol{w}_n) \end{bmatrix}^T$. Then, during the supervised learning step, we train a classifier so that it can find the mode using the human features $\boldsymbol{x}$ as input. Given the data matrix $\boldsymbol{X}$ and the output vector $c(w_i)$, this supervised learning builds us a model f such that $c_i = f(\boldsymbol{x_i})$,i.e.,

$$f = \arg\min_{\hat{f}} \sum_{i=1}^{n} \xi\left(c(w_i), \hat{f}(x_i)\right), \tag{8}$$

where $\xi(.)$ is a 0-1 loss function. f is approximated using a set of points $(\boldsymbol{x}_i, c_i)$ and determines the relationship between the features of a user and its cluster. After approximating f, given each human user's feature vector $\boldsymbol{x}_i$, we find the modes c_i using model f. Finally, we bounds $D_i^{\max}(\beta_i(t))$ based on its features $\boldsymbol{x}_i$. Now that the system can identify the human users' modes, we need to find a relationship between a human user's mode and the probabilistic model of its delay perception by defining the concept of effective delay.

Definition 1. Given the statistical model for human delay perception, $D_i^{\min}(\epsilon')$ is the *effective delay* for human user i that satisfies:

$$\Pr\left\{\beta_i(t) < D_i^{\min}(\epsilon')\right\} < \epsilon'. \tag{9}$$

To find the effective delay for human user i, we first find the probability that the delay perception of human user i is less than a threshold $D_i^{\min}(\epsilon')$. In other words, we want to find the

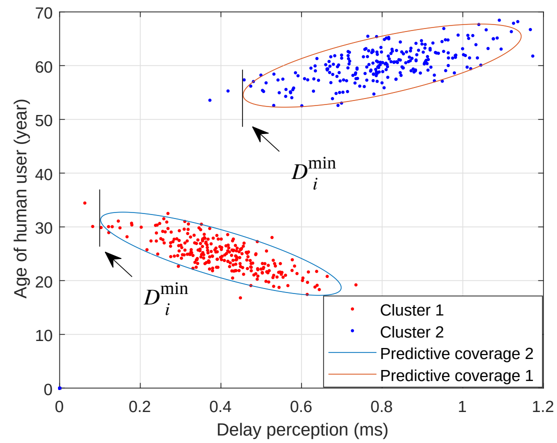

Figure 1: Finding $D_i^{\min}$ using a GMM model for two different clusters.

relation between ϵ' and $D_i^{\min}(\epsilon')$ in (9). The concept of effective delay is defined using the fact that delays less than $D_i^{\min}(\epsilon')$ cannot be sensed by a human with $(1-\epsilon')$ certainty. The relation between ϵ' and $D_i^{\min}(\epsilon')$ in (9) is found in Theorem 1. For notational simplicity, hereinafter, we use $D_i^{\min}$ instead of $D_i^{\min}(\epsilon')$.

Theorem 1. The delay perception of the identified brain mode k user i is bounded such that

$$\Pr\left\{|\beta_i - \mu_K^D| < \sqrt{Q_{p+1}(\gamma)\boldsymbol{e}_{p+1}^T\boldsymbol{\Sigma}_k\boldsymbol{e}_{p+1}}\right\} > \gamma, \tag{10}$$

where $\boldsymbol{\Sigma}_k$ and μ_k represent, respectively, the covariance matrix and the mean vector of the identified brain mode k. Also, $Q_p(\gamma)$ is the quantile function of chi-square distribution with p degrees of freedom, and is defined as

$$Q_{p+1}(\gamma) = \inf\left\{x \in \mathbb{R} | \gamma \leq \int_0^x \chi_{p+1}^2(u)du\right\}, \tag{11}$$

and $\boldsymbol{e}_j$ is a unit vector in $\mathbb{R}^{p+1}$, whose jth element is 1 and all other elements are zero. p is number of features used for learning. $\chi_{p+1}^2(x)$ is the probability density function of a chi-square random variable with $p+1$ degrees of freedom. μ_K^D is the element $p+1$ in vector $\boldsymbol{\mu}_k$.

Proof: See Appendix A. ■

As seen from Theorem 1, in addition to the delay perception element μ_K^D, the only other parameter that affects the delay is $\boldsymbol{e}_{p+1}^T\boldsymbol{\Sigma}_k\boldsymbol{e}_{p+1}$, which is the $(p+1)$th diagonal element of the covariance matrix $\boldsymbol{\Sigma}_k$. Note that, we did not assume that matrix $\boldsymbol{\Sigma}_k$ is diagonal. Fig.1 shows the relationship between D_i^{min} and GMM. From Fig.1, we can see that, after finding the GMM for the dataset, one can find the predictive coverage of each Gaussian distribution. Using this

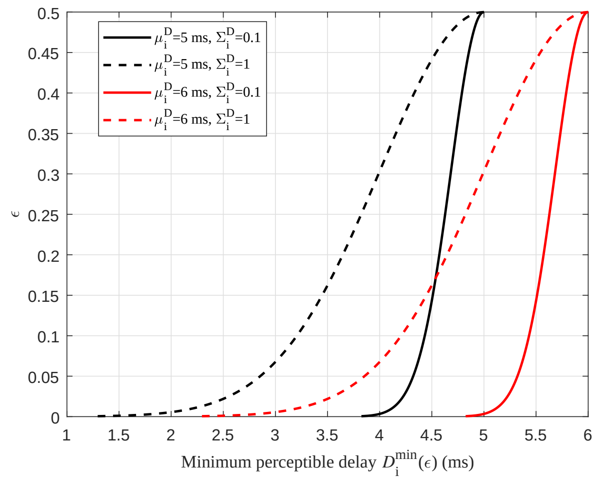

Figure 2: Relationship between ϵ and $D_i^{\min}(\epsilon)$ for different values of μ_K^D and Σ_K^D.

predictive coverage, we can determine the probability with which $\beta_i(t)$ for a user i will be higher than a threshold $D_i^{\min}$.

In order to find the effective delay for human user i, we first find the probability with which the delay perception for human user i will be less than a threshold $D_i^{\min}$. In other words, we will find the relationship between ϵ and $D_i^{\min}$ in (9) using the following corollary that follows directly from Theorem 1.

Corollary 1. As a direct result of Theorem 1, we can reduce (10) to

$$\Pr\left\{\beta_i(t) < \mu_K^D - \sqrt{Q(\gamma)\boldsymbol{e}_{p+1}^T \boldsymbol{\Sigma}_k \boldsymbol{e}_{p+1}}\right\} < \frac{1-\gamma}{2}. \tag{12}$$

Therefore, we find $D_i^{\min}(\epsilon)$ and ϵ in (9) as

$$D_i^{\min}(\epsilon) = \mu_K^D - \sqrt{Q(1-2\epsilon)\boldsymbol{e}_{p+1}^T \boldsymbol{\Sigma}_k \boldsymbol{e}_{p+1}}. \tag{13}$$

Since $Q(\gamma)$ can only be calculated numerically, a closed-form relationship cannot be found between $D_i^{\min}(\epsilon)$ and ϵ. However, we can numerically analyze this relationship, as shown in Fig. 2. From Fig. 2, we can first observe that $D_i^{\min}(\epsilon)$ is an increasing function. *This means that the probability of the human brain noticing QoS differences for low delays will be much smaller than for higher delays*, which is an intuitive fact. Furthermore, it can be inferred that, if the delay perception for a group of human users within a cluster is diverse, then the system's confidence on the delay perception of this group of humans will decrease, i.e., the estimation of the delay perception of this group of human users will be less reliable. Next, we determine constraint (2b) using $D_i^{\min}(\epsilon)$.

As stated before, some delays are not perceptible to human users. To capture this feature, we find $D_i^{\max}(\beta_i(t))$ and $\epsilon(\beta_i(t))$ in problem (2) using $D_i^{\min}(\epsilon)$. Recall that $D_i^{\max}(\beta_i(t))$ is a parameter that will be used by the resource allocation system to represent the maximum tolerable delay for the reliable communication of user i with $1-\epsilon(\beta_i(t))$ being the reliability of user i. There are three possible cases for $D_i^{\max}$ based on $D_i^{\min}$ of a human user i:

1) $D_i^{\max} > D_i^{\min}$: In this case, the system will not be reliable even if we satisfy $\Pr(D > D_i^{\max}) < \epsilon$. The reason is that the human user has a delay perception of less than the maximum delay $D_i^{\max}$ and hence, the system is not reliable.

2) $D_i^{\max} < D_i^{\min}$: In this case, if the system is able to satisfy $\Pr(D > D_i^{\max}) < \epsilon$, then the system will be reliable, because user i cannot sense delays less than $D_i^{\min}$ and its service delay will not exceed $D_i^{\max}$.

3) $D_i^{\max} = D_i^{\min}$: If this equality holds, the system will be reliable and it will also have prevented a waste of resources. If any given user cannot perceive delays less than $D_i^{\min}$, then it is not effective to allocate more resources to this user.

S represents the event where the system delay meets the cognitive perceptions of the human user, which is the desired result (case 2) and case 3)). In other words, if event S happens, the system is reliable. Also, we assume that the events E_1 and E_2 are defined as $D < D_i^{\max}$ and $\beta_i(t) > D_i^{\min}$, respectively. We know that for case 1, event $E_1 \cap E_2$ is a subset of event S, and in case 2, event S is a subset of event $E_1 \cap E_2$. Similarly, in case 3, event $E_1 \cap E_2$ is same as event S. Since the probability of $E_1 \cap E_2$ can be computed, if we set $D_i^{\min}$ to $D_i^{\max}$ (case 3), we will be able to find S based on the system parameters and design the resource allocation system in a way that each user is satisfied. We know that:

$$\Pr(E_1 \cap E_2) = 1 - \Pr\Big((D > D_i^{\max}) \cup (\beta_i(t) < D_i^{\min})\Big) \tag{14}$$

$$= 1 - \Big(\Pr(D > D_i^{\max}) + \Pr(\beta_i(t) < D_i^{\min}) - \Pr(D > D_i^{\max})\Pr(\beta_i(t) < D_i^{\min})\Big). \tag{15}$$

(14) follows from De Morgan's law, and (15) is true since D and $\beta_i(t)$ are two independent random variables. Therefore, if $D_i^{\min} = D_i^{\max}$ for user i and $\epsilon\epsilon'$ is small, we can see that

$$1 - \Big(\Pr(D > D_i^{\max}) + \Pr(\beta_i(t) < D_i^{\min})\Big) \geq 1 - (\epsilon + \epsilon'), \tag{16}$$

and, hence,

$$\Pr(S) = \Pr(E) > 1 - (\epsilon + \epsilon'), \tag{17}$$

where Pr(S) is the reliability of the system defined in (2). Subsequently, as we design the system, we consider the reliability as a predetermined target design parameter for the system. Using this parameter, we can set ϵ and ϵ'. Given ϵ' and numerical function $D_i^{\min}(\epsilon')$ derived in (12), $D_i^{\min}$ can be determined. Now, given $\epsilon(\beta_i(t))$ and $D_i^{\max}(\beta_i(t))$, we can fully characterize problem (2).

IV. Brain-Aware Resource Management

To solve problem (2), we propose a novel brain-aware resource management framework that takes into account the time-varying wireless channel and the time-varying brain-aware delay constraint (2b). In this section, we transform this constraint into a mathematically tractable form. First, in the next lemma, the relation between the packet length distribution and the service time distribution for a packet is shown.

Lemma 1. If a fixed rate r_i is allocated to a user and the packet lengths follow an exponential distribution with parameter χ, then, the distribution of the service time s will also be exponential with parameter χr_i.

Proof: The CDF of the exponential distribution is $F_{\psi}(l) = \Pr(l < \psi) = 1 - e^{-\chi\psi}$. Hence,

$$F_S(s) = \Pr(s < S) = \Pr(\frac{l}{r_i} < S) = \Pr(l < r_i S) = F_{r_i S}(s) = 1 - e^{-\chi r_i S}. \tag{18}$$

This means that the PDF for the service time is $f_S(s) = e^{-\chi r_i s}$. ■

Therefore, without loss of generality, we assume that the service time of each packet is exponential with parameter r_i, which is the same as the rate allocated to the user.

Here we assume that for any given user, the packets arrive with the rate $a_i(\tau)$, and the user data rate is $r_i(\tau)$ in slot $\tau = 1, \cdots, t$. In the next theorem, We derive the probability with which the delay of a given user i exceeds a threshold $D_i^{\max}$.

Theorem 2. Assume that user i has a time varying rate r_i^{τ} at time slot τ. If the duration of each time slot is long enough for the queue to reach its steady state, i.e.,

$$\frac{1}{r_i(\tau) - a_i(\tau)} << \delta\tau, \tag{19}$$

then, the probability that the delay exceeds a threshold is

$$\Pr(D > D_i^{\max}) = \lim_{t\to\infty} \frac{1}{t} \sum_{\tau=1}^{t} e^{-\left(r_i(\tau) - a_i(\tau)\right) D_i^{\max}}, \tag{20}$$

under the condition that $r_i(\tau) > a_i(\tau)$ for all $\tau > 0$.

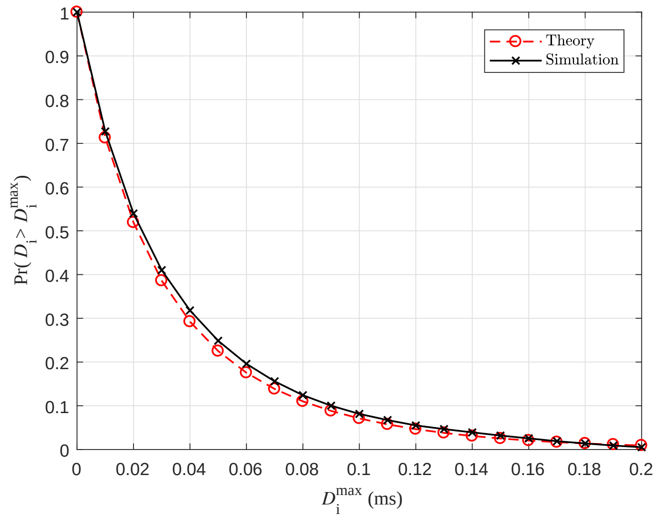

Figure 3: Comparison between simulation results and the result of Theorem 2.

Proof: See Appendix B. ■

Theorem 2 shows that constraint (2b) is satisfied if the network can satisfy the following condition

$$\lim_{t\to\infty} \frac{1}{t}\sum_{\tau=1}^{t} e^{-\left(r_i(\tau)-a_i(\tau)\right)D_i^{\max}} < \epsilon. \tag{21}$$

Fig. 3 shows the relationship between the theoretical result from Theorem 2 and the simulation results. It shows that the simulation and analytical results are a near-perfect match. The maximum error between the analytical and simulation results is only 0.0146.

A. Optimal Resource Allocation with Guaranteed Reliability

Constraint (21) is analogous to the drift-plus-penalty method in the Lyapunov optimization framework [25] and, hence, we will use this framework to solve (2). The problem has a time-varying nature since the human brain conditions and needs will change from time to time. The users' processing state $\beta(t)$ is also a function of time, and accordingly, the latency needs in (2b) will be time-varying. Therefore, we need to solve the optimization problem (2) during each time slot efficiently. Here, we propose an algorithm with a low computational complexity for solving problem (2). The drift-plus-penalty approach is used to stabilize a queue network while minimizing time average of a penalty function. In order for constraint (2b) to be satisfied in all time slots, we need to make sure that (20) will always be smaller than ϵ. For this reason, we define a virtual queue:

$$F_i(t+1) = \max\{F_i(t) + e^{-\left(r_i(t+1)-a_i(t+1)\right)D_i^{\max}} - \epsilon\}. \tag{22}$$

We can see that $e^{-\left(r_i(t+1)-a_i(t+1)\right)D_i^{\max}} - \epsilon < F_i(t+1) - F_i(t)$. Consequently, we obtain:

$$\sum_{\tau=1}^{t} e^{-\left(r_i(t+1)-a_i(t+1)\right)D_i^{\max}} - \epsilon t < F_i(t) - F_i(0). \tag{23}$$

If $F_i(0)$ is bounded, we have:

$$\lim_{t\to\infty} \frac{1}{t} \sum_{\tau=1}^{t} e^{-\left(r_i(t+1)-a_i(t+1)\right)D_i^{\max}} - \epsilon < \lim_{t\to\infty} \frac{F_i(t)}{t}. \tag{24}$$

If the queue $F_i(t)$ is mean-rate stable, that is, $\lim_{t\to\infty} \frac{F_i(t)}{t} = 0$, then we have:

$$\lim_{t\to\infty} \frac{1}{t} \sum_{\tau=1}^{t} e^{-\left(r_i(t+1)-a_i(t+1)\right)D_i^{\max}} < \epsilon. \tag{25}$$

The Lyapunov function is defined for all the queues in the base station as

$$Y(t) = \frac{1}{2} \sum_{i\in\mathcal{M}\cup H} F_i(t)^2. \tag{26}$$

Then, we can find the drift function $\Delta t = Y(t+1) - Y(t)$ as:

$$Y(t+1) = \frac{1}{2} \sum_{i\in\mathcal{M}\cup\mathcal{H}} F_i(t+1)^2 \leq \frac{1}{2} \sum_{i\in\mathcal{M}\cup\mathcal{H}} F_i(t)^2 + \frac{1}{2} \sum_{i\in\mathcal{M}\cup\mathcal{H}} y(t)^2 + \sum_{i\in\mathcal{M}\cup\mathcal{H}} y(t)F_i(t), \tag{27}$$

where

$$y_i(t) = e^{-\left(r_i(t+1)-a_i(t+1)\right)D_i^{\max}} - \epsilon. \tag{28}$$

Thus,

$$\Delta t \leq \frac{1}{2} \sum_{i\in\mathcal{M}\cup\mathcal{H}} y(t)^2 + \sum_{i\in\mathcal{M}\cup\mathcal{H}} y(t)F_i(t). \tag{29}$$

We can form the drift-plus-penalty by adding $V\sum_{i,j} p_{ij}(t)$ to both sides of inequality (29), where $\sum_{i,j} p_{ij}$ is the total power of the BS which we want to minimize, and V is a parameter that determines how important minimizing the objective function (2a) is in comparison with satisfying (2b). We can balance the tradeoff between power and delay. The drift-plus-penalty inequality is

$$\Delta t + V\sum_{i,j} p_{ij} \leq \frac{1}{2} \sum_{i\in\mathcal{M}\cup\mathcal{H}} y(t)^2 + V\sum_{i,j} p_{ij}(t) + \sum_{i\in\mathcal{M}\cup\mathcal{H}} y(t)F_i(t). \tag{30}$$

Given that we assumed $r_i(t) > a_i(t)$ for all t, we know that $|y_i(t)| < 1 \, \forall t, i \in \mathcal{M} \cup \mathcal{H}$, and hence, we can rewrite (31) as

$$\Delta t + V \sum_{i,j} p_{ij} \leq B + V \sum_{i,j} p_{ij}(t) + \sum_{i \in \mathcal{M} \cup \mathcal{H}} y(t) F_i(t), \tag{31}$$

where B is the upper bound of $\frac{1}{2} \sum_{i \in \mathcal{M} \cup \mathcal{H}} y(t)^2$, and is equal to $\frac{|\mathcal{H}|+|\mathcal{M}|}{2}$. $|\mathcal{A}|$ is the cardinality of set $\mathcal{A}$. Using the drift-plus-penalty algorithm [26], we know that, by minimizing the right hand side of equation (31), queue $F_i(t)$ will be mean-rate stable, and hence, the condition $y_i(t) < 0$ will be satisfied. As a result, constraint (2b) will also be satisfied. Furthermore, we know that by minimizing the right hand side of (31), cost function (2a) is also minimized, owing to the fact that (2a) is defined as a penalty function. By minimizing the right hand side of (31), our optimization problem can be converted to the following time-varying problem:

$$\min_{\boldsymbol{\rho}(t), \boldsymbol{P}(t)} \quad V \sum_{i,j} p_{ij}(t) + \sum_{i \in \mathcal{M} \cup \mathcal{H}} y_i(t) F_i(t), \tag{32a}$$

$$\text{s.t.} \quad r_i(t) > a_i(t), \qquad \forall i \in \mathcal{H} \cup \mathcal{M} \tag{32b}$$

$$p_{ij}(t) \geq 0, \quad \rho_{ij}(t) \in \{0,1\}, \qquad \forall i \in \mathcal{H} \cup \mathcal{M}, j \in \mathcal{K}, \tag{32c}$$

$$\sum_{i \in \mathcal{H} \cup \mathcal{M}} \rho_{ij}(t) = 1, \qquad \forall j \in \mathcal{K}. \tag{32d}$$

As discussed earlier, the cost function in (32a) is equivalent to (2a) and (2b) in the original optimization problem. Learning the effective delay of each human user using our proposed PDI method determines the parameters $y_i(t)$ and $F_i(t)$ in the problem (32a). However, in order to satisfy (2b), we need to also satisfy (32b). The reason for adding (32b) is that if this constraint is not satisfied in any time slot, the queue length will approach infinity. Constraints (32c) and (32d) are feasibility conditions and remain the same as (2). Hence, by solving (32) in each time slot, the original problem (2) will be solved.

Nonetheless, problem (2a) is not a convex optimization problem, due to the fact that it is a mixed integer problem and its complexity increases exponentially with the number of users. Since (2a) needs to be solved at each time slot, this exponential order of complexity makes the implementation infeasible. Consequently, we should use a dual decomposition method to break down optimization problem (32) to smaller subproblems, and find the optimal solution to (32) using a low complexity method.

It is rather challenging to solve (32) using a dual decomposition method, as the structure of $y_i(t)$ makes it infeasible to decompose the objective function for each resource block. In order to overcome this challenge, we convert (32) to a decomposable form. Then, we will show that this converted problem is equivalent to (32).

For this purpose, the Lagrangian for problem (32) is written as

$$V\sum_{i,j} p_{ij}(t) + \sum_{i\in\mathcal{M}\cup\mathcal{H}} y_i(t)F_i(t) + \sum_{i\in\mathcal{M}\cup\mathcal{H}} \lambda_i\big(a_i(t) - r_i(t)\big), \tag{33}$$

where λ_i is the Lagrange multiplier. As we know, $y_i(t) = e^{-\left(r_i(t+1)-a_i(t+1)\right)D_i^{\max}} - \epsilon$. Therefore, the only decision variables are allocation of resource blocks to the users and allocating power to each resource block. Although $F_i(t)$ is a function of $y_i(t)$, it is not a decision variable and is treated as a constant. Hence, (33) can be rewritten as

$$V\sum_{i,j} p_{ij}(t) + \sum_{i\in\mathcal{M}\cup\mathcal{H}} e^{-\left(r_i(t+1)-a_i(t+1)\right)D_i^{\max}} F_i(t) + \sum_{i\in\mathcal{M}\cup\mathcal{H}} \lambda_i\big(a_i(t) - r_i(t)\big). \tag{34}$$

The main optimization problem consists of two components. First, minimizing the total power of the BS with weight V, and second, minimizing the summation $\sum_{i\in\mathcal{M}\cup\mathcal{H}} e^{-r_i(t+1)}$ which has a weight $F_i(t)e^{-a_i(t+1)D_i^{\max}}$ for each user i.

As we can see, (34) is not decomposable for each resource block. Here we will have an approximation of (32) and then propose an algorithm to solve this approximation efficiently. In this C-additive approximation, $\sum_{i\in\mathcal{M}\cup\mathcal{H}} e^{-\left(r_i(t+1)-a_i(t+1)\right)D_i^{\max}} F_i(t)$ in (34) is substituted with its linear approximation of exponential term e^{-x} at $x = 0$.

$$\sum_{i\in\mathcal{M}\cup\mathcal{H}} -\big(r_i(t+1) - a_i(t+1)\big)F_i(t). \tag{35}$$

In the original problem, if $y_i(t)$ starts to become greater than zero for user i, then $F_i(t)$ will increase and it will give more weight to the term $e^{-\left(r_i(t+1)-a_i(t+1)\right)D_i^{\max}}$. As a result, the algorithm allocates more resources to user i such that it minimizes $e^{-\left(r_i(t+1)-a_i(t+1)\right)D_i^{\max}}$ for user i, and accordingly, $y_i(t)$ decreases. Hence, $F_i(t)e^{-\left(r_i(t+1)-a_i(t+1)\right)D_i^{\max}}$ plays the role of feedback in the system. As we can see from (35), this approximation will not change this feedback mechanism and plays the same role in the system. Therefore, we can write

$$\begin{aligned}&\min_{\boldsymbol{P},\rho}\big\{V\sum_{i,j} p_{ij}(t) + \sum_{i\in\mathcal{M}\cup\mathcal{H}} -\big(r_i(t+1) - a_i(t+1)\big)D_i^{\max}F_i(t)\big\}\\ &< C + \min_{\boldsymbol{P},\rho}\big\{V\sum_{i,j} p_{ij}(t) + \sum_{i\in\mathcal{M}\cup\mathcal{H}} e^{-\left(r_i(t+1)-a_i(t+1)\right)D_i^{\max}} F_i(t)\big\}.\end{aligned} \tag{36}$$

Using this C-additive approximation, it can be easily proved that all terms are mean-rate stable. Hence, (2b) in the original problem is satisfied [25]. Finally, problem (2) can be presented as:

$$\begin{aligned}
\min_{\boldsymbol{\rho}(t),\boldsymbol{P}(t)} \quad & V\sum_{i,j} p_{ij}(t) - \sum_{i\in\mathcal{M}\cup\mathcal{H}} \big(r_i(t+1)-a_i(t+1)\big) D_i^{\max} F_i(t), \\
\text{s.t.} \quad & r_i(t) > a_i(t), && \text{(37a)}\\
& P_i^n(t) \geq 0, \quad \forall i \in \mathcal{H}\cup\mathcal{M}, n\in\mathcal{N}, && \text{(37b)}\\
& \rho_i^n(t) \in \{0,1\}, \quad \forall i \in \mathcal{H}\cup\mathcal{M}, n\in\mathcal{N}, && \text{(37c)}\\
& \sum_{i\in\mathcal{H}\cup\mathcal{M}} \rho_{ij}(t) = 1, \quad \forall j \in \mathcal{K}. && \text{(37d)}
\end{aligned}$$

In order to solve this problem, we can decompose it into K subproblems. Since these subproblems are coupled through constraint (37d), we use the dual decomposition method for solving (37) [27]. First, the Lagrangian is written for problem (37), and in the second step, it is decomposed for each resource block. After that, the resource block allocation and the power of each RB are found in terms of the Lagrange multiplier λ. Finally, λ is calculated using an ellipsoid method.

The Lagrangian for problem (37) is

$$\begin{aligned}
\mathcal{L}(\boldsymbol{P},\rho,\lambda) &= V\sum_{i,j} p_{i,j}(t) + \sum_{i\in\mathcal{M}\cup\mathcal{H}} -\big(r_i(t)-a_i(t)\big) D_i^{\max} F_i(t) - \lambda_i\big(r_i(t)-a_i(t)\big) \\
&= V\sum_{i,j} p_{i,j}(t) - \sum_{i\in\mathcal{M}\cup\mathcal{H}} \big(\lambda_i + D_i^{\max} F_i(t)\big)\big(r_i(t)-a_i(t)\big). \qquad (38)
\end{aligned}$$

One major difference between our problem and conventional power minimization problems is that there is an additional term $D_i^{\max} F_i(t)$ added to the Lagrange multiplier (the shadow price). In this problem, $D_i^{\max} F_i(t)$ plays the role of a bias term. Therefore, a new hypothetical Lagrange multiplier λ_i' is assumed and defined as

$$\lambda_i' = \lambda_i + D_i^{\max} F_i(t). \qquad (39)$$

This means that adding constraint (2b) to the problem instead of constraint (37a) increases the shadow price by a factor of $D_i^{\max} F_i(t)$. Increasing the shadow price for a constraint makes it looser. As a result, in many time slots, constraint (37a) will not be a tight constraint and the Lagrange multiplier will be set to $\lambda_i = \left[\lambda_i' - D_i^{\max} F_i(t)\right]^+$. the Lagrange dual function is

$$g(\lambda) = \min_{\boldsymbol{\rho}(t),\boldsymbol{P}(t)} \quad \mathcal{L}(\boldsymbol{P},\boldsymbol{\rho},\lambda). \qquad (40)$$

The minimization problem (40) can be decomposed to K subproblems. $g'_j(\lambda)$ can be written as

$$g'_j(\lambda) = \min_{\boldsymbol{P}(t)} \quad V \sum_i p_{i,j} - \sum_{i \in \mathcal{M} \cup \mathcal{H}} \big(\lambda_i + D_i^{\max} F_i(t)\big)\big(W \log_2(1 + K \frac{h_{i,j} p_{i,j}}{\sigma^2})\big), \tag{41}$$

where $\mathcal{D}$ is a set of feasible p_{ij}s in which for RB j, there is only one i that $p_{ij} \neq 0$. Hence, $g(\lambda)$ is

$$g(\lambda) = \sum_j g'_j(\lambda) + \sum_{i \in \mathcal{M} \cup \mathcal{H}} \big(\lambda_i + D_i^{\max} F_i(t)\big)\big(a_i(t)\big). \tag{42}$$

If λ is fixed, $g'_j(\lambda)$ is a convex function of $\boldsymbol{P}$. Therefore, $\boldsymbol{P}$ is found by taking a derivate with respect to p_{ij} and setting it to zero. This results in

$$p_{ij} = \Big[\frac{\big(\lambda_i + D_i^{\max} F_i(t)\big)W}{V \log_2} - \frac{\sigma^2}{K h_{ij}}\Big]^+. \tag{43}$$

The optimal RB allocation for RB j is $k(j)$, and can be written as

$$k(j) = \underset{i}{\text{argmin}} V \sum_i p_{i,j} - \sum_{i \in \mathcal{M} \cup \mathcal{H}} \big(\lambda_i + D_i^{\max} F_i(t)\big)\big(W \log_2(1 + K \frac{h_{i,j} p_{i,j}}{\sigma^2})\big), \tag{44}$$

$$g'_j(\lambda) = \min_i V \sum_i p_{i,j} - \sum_{i \in \mathcal{M} \cup \mathcal{H}} \big(\lambda_i + D_i^{\max} F_i(t)\big)\big(W \log_2(1 + K \frac{h_{i,j} p_{i,j}}{\sigma^2})\big). \tag{45}$$

Thus, ρ^*_{ij} and p^*_{ij} will be given by:

$$\rho^*_{ij} = \begin{cases} 1, & i = k(j), \\ 0, & \text{otherwise.} \end{cases} \qquad p^*_{ij} = \begin{cases} p_{ij}, & i = k(j), \\ 0, & \text{otherwise.} \end{cases} \tag{46}$$

Hence, the optimal rate becomes $r^*_i = \sum_j W \log_2(1 + K \frac{h_{i,j} p^*_{i,j}}{\sigma^2})$. The only parameter that affects this joint RB and power allocation is λ. As the number of RBs increases, the duality gap in this problem approaches zero [27]. We know that the optimal value is found by maximization of $g(\lambda)$ with respect to λ. In order to find λ, we use the ellipsoid method [28], and to do so, we have to find the sub-gradient for the dual objective $g(\lambda)$. The following theorem will show that the subgradient for (38) is a vector with elements $d_i = a_i - r_i$.

Theorem 3. The subgradient of the dual optimization problem with dual objective defined in (42), is the vector $\boldsymbol{d}$ whose elements d_i, $\forall i \in \mathcal{H} \cup \mathcal{M}$ are given by:

$$d_i = \begin{cases} a_i - r^*_i, & a_i \geq r^*_i, \\ 0, & a_i < r^*_i. \end{cases} \tag{47}$$

Proof: Since

$$g(\boldsymbol{\lambda}) = \min_{\boldsymbol{P},\rho} \mathcal{L}(\boldsymbol{P}, \boldsymbol{\rho}, \boldsymbol{\lambda}) = \mathcal{L}(\boldsymbol{P}^*, \rho^*, \boldsymbol{\lambda}), \tag{48}$$

we have:

$$\begin{aligned} g(\boldsymbol{\delta}) \leq& \mathcal{L}(\boldsymbol{P}^*, \rho^*, \boldsymbol{\delta}) \\ =& V \sum_{i,j} p^*_{i,j}(t) - \sum_{i \in \mathcal{M} \cup \mathcal{H}} \big(\delta_i + D_i^{\max} F_i(t)\big)\big(r_i^*(t) - a_i(t)\big) \\ =& V \sum_{i,j} p^*_{i,j}(t) - \sum_{i \in \mathcal{M} \cup \mathcal{H}} \big(\lambda_i + D_i^{\max} F_i(t)\big)\big(r_i^*(t) - a_i(t)\big) \\ &+ (\lambda_i - \delta_i)\big(r_i^*(t) - a_i(t)\big) = g(\boldsymbol{\lambda}) + (\boldsymbol{\lambda} - \boldsymbol{\delta})^T \boldsymbol{d}', \end{aligned} \tag{49}$$

where

$$\boldsymbol{d}' = \begin{bmatrix} r_1^* - a_1 & \cdots & r_{N+M}^* - a_{N+M} \end{bmatrix}^T. \tag{50}$$

However, because of the term $D_i^{\max} F_i(t)$, when $\lambda = 0$ and $a_i < r_i^*$, then the direction of $\boldsymbol{d}'$ is infeasible. Using the projected subgradient method [29], we can transform this infeasible direction to a feasible one. The update rule for projected subgradient is:

$$\boldsymbol{\lambda}^{(k+1)} = \Pi(\boldsymbol{\lambda}^{(k)} - \alpha_k d'_k) \tag{51}$$

where α_k is the step size and Π is the Euclidan projection on the feasible set. Since the feasible set is $\lambda_i > 0$, we can see that

$$\Pi(\boldsymbol{\lambda}^{(k)} - \alpha_k d'_k) = \boldsymbol{\lambda}^{(k)} - \alpha_k d_k, \tag{52}$$

where:

$$d_i = \begin{cases} d'_i, & d'_i \geq 0 \\ 0, & d'_i < 0 \end{cases} = \begin{cases} a_i - r_i^*, & a_i \geq r_i^*, \\ 0, & a_i < r_i^*. \end{cases} \tag{53}$$

■

B. Complexity Analysis

Next, we find the complexity of our algorithm which needs to be run in each iteration. There are K RBs in our problem, for each of which (44) needs to be evaluated for $M + N$ users. It takes $\mathcal{O}\big((M+N)K\big)$ times to solve a primal problem. Subsequently, the dual problem will be solved, which gives us the optimal value of λ in an $M + N$ dimensional space and has a complexity of $\mathcal{O}\big((M+N)^2\big)$. Therefore, the overall complexity should be $\mathcal{O}\big((M+N)^3 K\big)$. However, as mentioned before, adding $D_i^{\max} F_i(t)$ to the Lagrange multiplier sets a major part of it to zero, and as a result, the order of complexity will decrease to $\mathcal{O}\big((M+N)K\big)$.

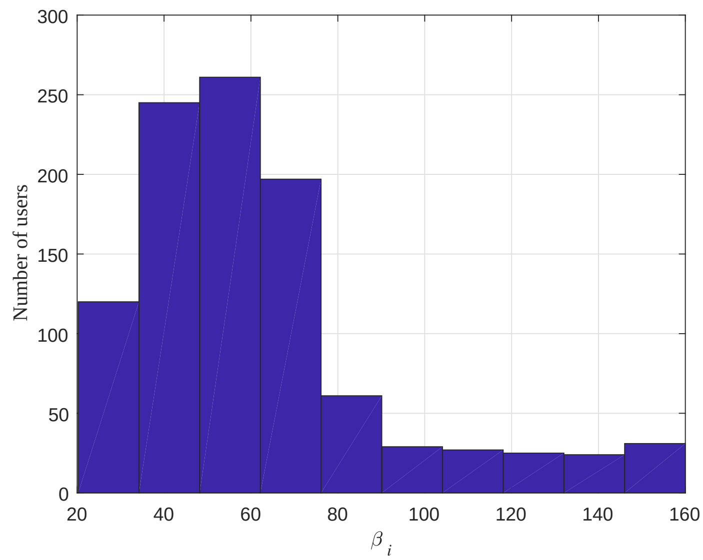

Figure 4: Distribution of β_i for the 1000 users in dataset.

V. Simulation Results and Analysis

For our simulations, we consider the dataset in [22] to model the delay perception of a human user. In [22], the authors conducted human subject studies using 30 human users, where each subject is asked to rate the quality of 5 movies while the delay and packet loss in the system is being increased. We used the average score of each human user to estimate their delay perception. We also used a variation of the bootstrap method [30] to increase the number of data points to 1000. We can see the histogram of the delay perception for these 1000 data points in Fig. 4.

Also, since the dataset has no features for each user, we attributed three random continuous features to each user. Hence, each user is associated with a vector $\boldsymbol{w} \in \mathbb{R}^4$.

We consider a network with a bandwidth of 10 MHz, $a_i(t) = 1$ Mbps, $\sigma^2 = -173.9$ dBm, and $\epsilon = 0.05$. We use a circular cell with the cell radius of 1.5 km. We set the path loss exponent to 3 (urban area) and the carrier frequency to 900 MHz. The packet length is an exponential random variable with an average size of 10 kbits. We use 5 MTD and 5 UE in the system and we set $D_i^{\max}$ to 20 ms for them, unless otherwise mentioned. For the brain aware users, we arbitrarily select 5 UE in the system out of all data points.

Fig. 5 shows the within cluster point scatter for the EM algorithm in our dataset. This *within cluster point scatter* for a clustering C is defined as [30]:

$$W(C) = \frac{1}{2} \sum_{k=1}^{n} \sum_{c(i)=k} \sum_{c(i')=k} d(x_i, x_{i'}), \tag{54}$$

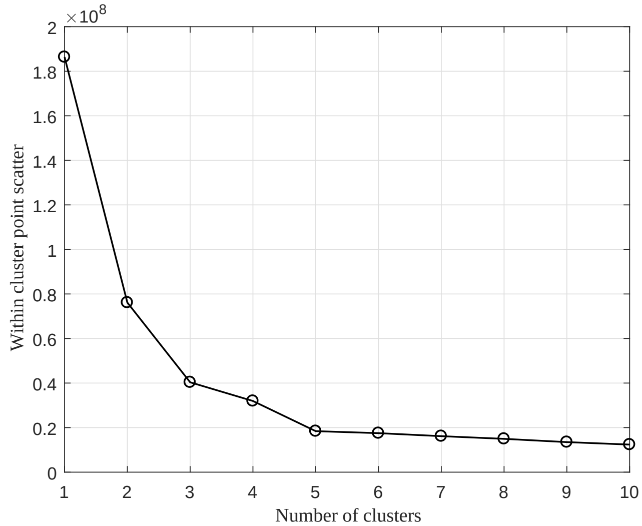

Figure 5: Within point scatter for the EM clustering method on the datasest.

where d is an arbitrary distance metric. In essence, the within cluster point scatter is a loss function that allows the determination of hyper-parameters in the clustering algorithm. The hyper-parameter that we seek to find here is the number of clusters in the dataset. As we can see from Fig. 5, after the number of clusters reaches 5, increasing the number of clusters does not decrease the within cluster point scatter substantially. Hence, the optimal number of clusters with is 5.

Fig. 6 shows the total BS power resulting from the proposed brain-aware case and from a brain-unaware case in which UEs have a fixed constraint (2b) with $D_i^{\max}$ between 10 ms to 60 ms. Here, the total power is the objective of main optimization problem (2). Fig. 6 shows that, as the latency increases, the total power decreases, because it is easier to satisfy constraint (2b) at higher latencies. Also, at higher delays, being brain-aware will no longer yield substantial gains, since $\beta_i(t)$ and $D_i^{\max}$ become close to each other and learning $\beta_i(t)$ cannot save resources for the system. In contrast, in Fig. 6, we can see that for stringent low-latency requirements, the proposed brain-aware approach yields significant gains in terms of saving power. In particular, for 10 ms delay in (2b), Fig. 6 shows that the BS in brain-unaware approach uses 44 % more power compared to the brain-aware case. These results stem from the fact that a brain-aware approach can minimize waste of resources and provide service to the users more precisely based on their real brain processing power. Fig. 7 shows average BS power for different number of MTDs. As we can see from Fig. 7, the brain-aware approach will always outperform the brain-unaware approach as the number of MTD increases. For the case of 30 MTD user, the BS in brain-unaware approach uses 16 % more power compared to the brain-aware case. This is due

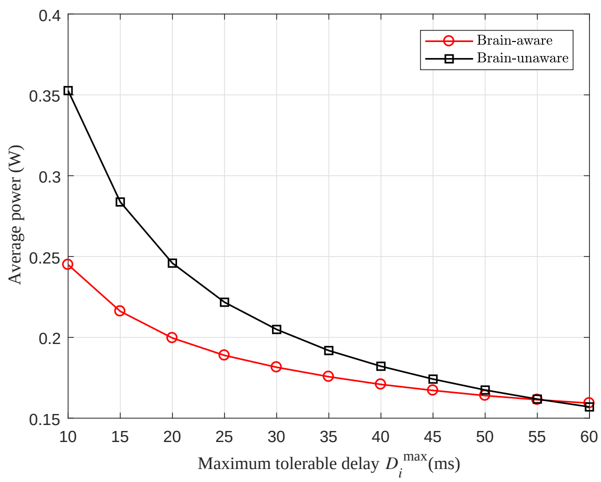

Figure 6: Average power usage of the system as function of different latency requirements for the users.

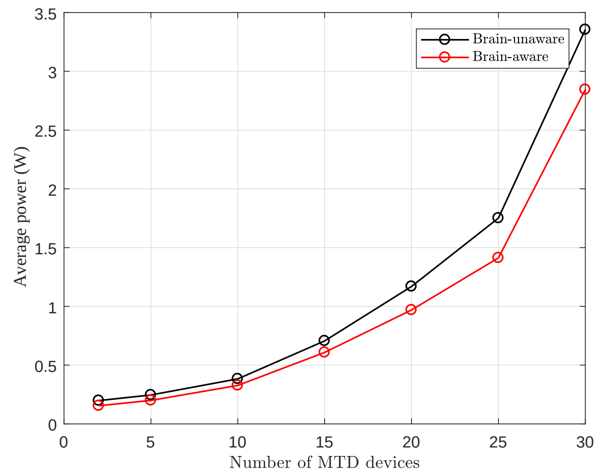

Figure 7: Average power usage of the system for different number of MTDs and 5 UEs.

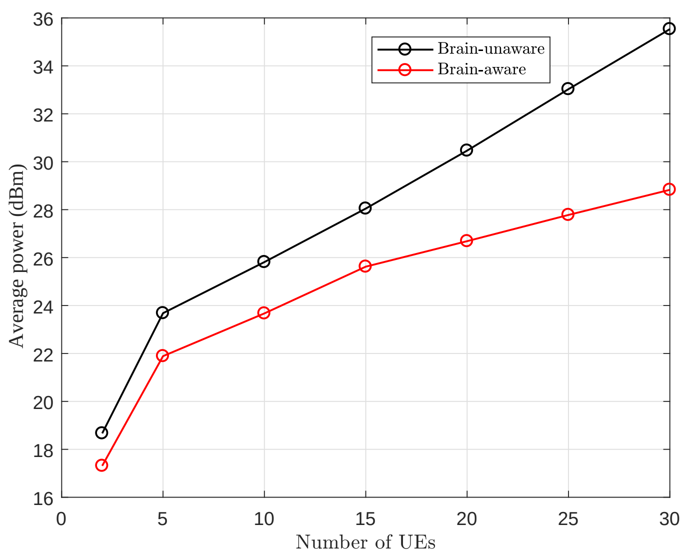

Figure 8: Average power usage of the system for different number of UEs.

to fact that brain-aware approach can allocate resources more efficiently in case of a shortage in resources.

In Fig. 8, we show the average power usage of the system when the number of UEs increases from 2 to 30 with $D_i^{\max}$ set to 20 ms. As the number of users increases, the average power consumption of the system will also increase. This is due to the fact that increasing the number of users will decrease the bandwidth per user. Since the delay and rate requirements of each

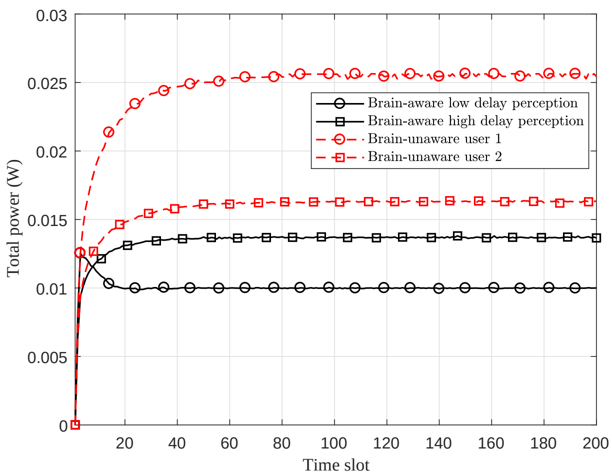

Figure 9: Transmit power for 4 different users. The delay perception of two of the users is learned.

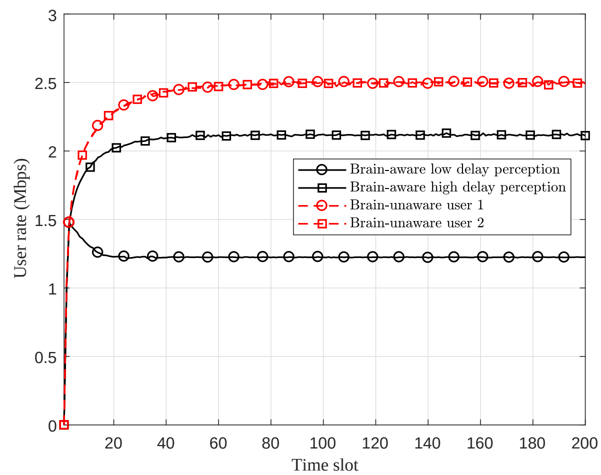

Figure 10: Transmission rate for four different users. The delay perception of two of the users is learned.

user are still unchanged, the system needs to use more power to compensate for the bandwidth deficiency. From Fig. 8, we can see that, in the case of 30 users, the brain-aware system is able to save 6.7 dB (78%) on average in the BS power. The brain-aware system can allocate resources based on each user's actual requirement instead of the predefined metrics and this leads to this significant saving in the power consumption of the BS.

In Fig. 9, Fig. 10, and Fig. 11, we consider the case of 7 UEs and 5 MTDs. Two UEs are chosen as brain-aware users and their delay perception is learned by the PDI method. One of the brain-aware UEs has a delay perception of $\beta_i = 133.73$ ms, and the other one has β_i equal to 26.8 ms. The system does not learn the delay perception of the 5 remaining UEs and, hence, it allocates resources to them by using a predefined delay requirement (brain-unaware users).

As we can see in Fig. 9, the power consumption of the first two brain-aware users will be less than that of the brain-unaware users. Moreover, the power consumption for a user with higher delay perception will be less than that of a user with lower delay perception. This shows that the system can successfully allocate resources according to the delay perception of the users. Furthermore, the power consumption related to each user with predetermined delay requirements is different, due to their different channel gains. However, as we will see later, the system is robust to such differences and can guarantee the reliability and rate requirements for users having different channel gains.

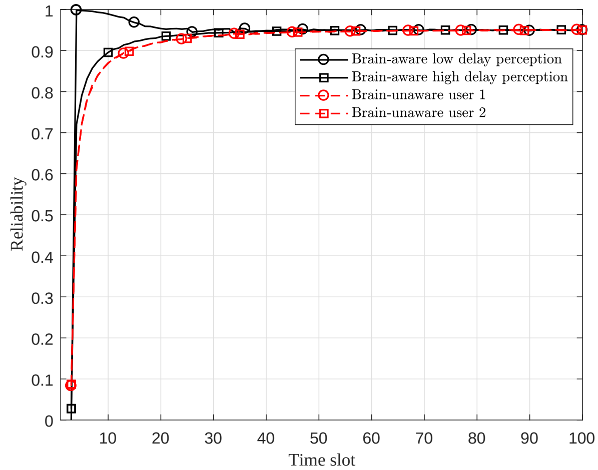

Figure 11: Reliability for four different users. The delay perception of two of the users is learned.

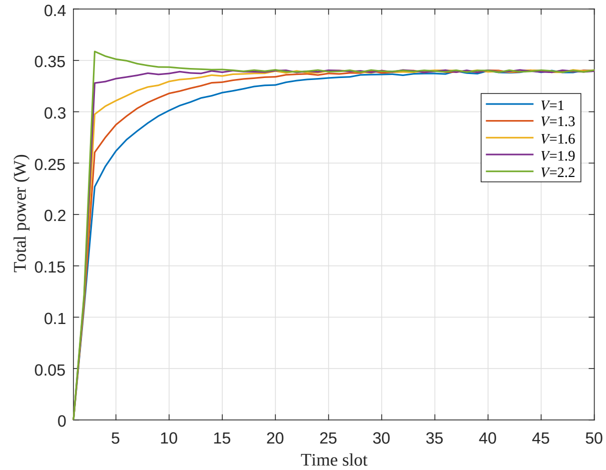

Figure 12: Effect of parameter V on the convergence of the resource allocation algorithm.

In Fig. 10, we show the transmission rate for four different users. We can see that the rate for brain-unaware users with predetermined delay will converge to 2.5 Mbps. This rate will ensure the reliability for these users. However, the rate of the users with learned delay perception will converge to a smaller rate. This is due to the fact that these users' actual requirements are known to the system, and the system uses this knowledge to avoid unnecessarily wasting resources. However, as we will see next, this rate reduction does not change the reliability for these users.

Fig. 11 shows the reliability for the four aforementioned users. As we can see, the reliability of all the users will converge to 95 %, which is the target reliability value for the users. We can see that the system is able to ensure reliability for the users with identified delay perceptions as well as the users with predefined delay requirements. However, the system used 45% less power for those users for which the delay perception is learned.

Finally, Fig. 12 investigates the effect of parameter V for the system with 5 MTDs and 5 UEs. We can see that, as V increases from 1 to 1.9, the convergence time decreases from 40 iterations to 15 iterations. Nevertheless, increasing V will make the algorithm unstable, and as we can see, increasing it to 2.2 will create an overshoot which is 11% higher than the final value. Hence, parameter V, if adjusted correctly, can create a balance between stability and convergence rate of our algorithm.

VI. Conclusion

In this paper, we have introduced and formulated the notion of delay perception of a human brain, in wireless networks with humans-in-the-loop. Using this notion, we have defined the concept of effective delay of human brain. To quantify this effective delay, we have proposed a novel learning method, named PDI, which consists of an unsupervised and supervised learning part. We have then shown that PDI can predict the effective delay for the human users and find the reliability of this prediction. Then, we have derived a closed-form relationship between the reliability measure and wireless physical layer metrics. Next, using this relationship and the PDI method, we have proposed a novel approach based on Lyapunov optimization for allocating radio resources to human users while considering the reliability of both machine type devices and human users. Our results have shown that the proposed brain-aware approach can save a significant amount of power in the system, particularly for low-latency applications and congested networks. To our best knowledge, this is the first study on the effect of human brain limitations in wireless network design.

Appendix A
Proof of Theorem 1

We assume that a single brain mode is dominant for each user at each time. We index this single mode as k. For each user i with this dominant mode, $\boldsymbol{w}_i = [w_1, \cdots, w_n]$ has the following probability density function:

$$f(\boldsymbol{w}_i) = |2\pi\boldsymbol{\Sigma}_k|^{-\frac{1}{2}} \exp\Big[-\frac{1}{2}(\boldsymbol{w}_i - \boldsymbol{\mu}_k)^T\boldsymbol{\Sigma}_k^{-1}(\boldsymbol{w}_i - \boldsymbol{\mu}_k)\Big]. \tag{55}$$

We want to find the smallest region $\mathcal{D}$ in $\mathbb{R}^{p+1}$, in which the delay perception lies with probability γ, i.e.,

$$\int \cdots \int_{\mathcal{D}} f(w_1, w_2, \ldots, w_n)\, dw_1 \cdots dw_n = \gamma. \tag{56}$$

$\mathcal{D}$ is not a unique region. However, the objective is to find the smallest region. To this end, we need to find the region where $f(w_1, w_2, \ldots, w_n)$ has the greatest value, i.e., if

$$\int \cdots \int_{\mathcal{D}_1} f(w_1, w_2, \ldots, w_n)\, dw_1 \cdots dw_n = \int \cdots \int_{\mathcal{D}_2} f(y_1, y_2, \ldots, y_n)\, dy_1 \cdots dy_n, \tag{57}$$

and also

$$f(y_1, y_2, \ldots, y_n) \leq f(w_1, w_2, \ldots, w_n) \qquad \forall \boldsymbol{y} \in \mathcal{D}_2,\ \forall \boldsymbol{w}_i \in \mathcal{D}_1, \tag{58}$$

then

$$\int \cdots \int_{\mathcal{D}_1} dw_1 \cdots dw_n \leq \int \cdots \int_{\mathcal{D}_2} dy_1 \cdots dy_n, \tag{59}$$

which implies that the volume of the region $\mathcal{D}_1$ is smaller than the volume of $\mathcal{D}_2$. Hence, if we find the region $\mathcal{D}$ for which (56) holds, and, using (58), show that all other regions for which (56) holds have greater volumes, then, we would have found the smallest region $\mathcal{D}$, in which the human behavior will stay with the probability γ.

Since $\boldsymbol{w}_i$ is distributed according to a multivariate Gaussian, we can find the region where it has the highest probability density, i.e., $\{\boldsymbol{w}_i | f(\boldsymbol{w}_i) > C_1\}$. This region can be written as:

$$\left\{\boldsymbol{w}_i \middle| |2\pi\boldsymbol{\Sigma}_k|^{-\frac{1}{2}} \exp\left[-\frac{1}{2}(\boldsymbol{w}_i - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\boldsymbol{w}_i - \boldsymbol{\mu}_k)\right] > C_1\right\}, \tag{60}$$

which is equivalent to

$$\mathcal{D} = \left\{\boldsymbol{w}_i \middle| (\boldsymbol{w}_i - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\boldsymbol{w}_i - \boldsymbol{\mu}_k) < C_2\right\}, \tag{61}$$

where C_2 is a positive constant and equals $-\ln|2\pi\boldsymbol{\Sigma}_k|^{\frac{1}{2}} C_1$. Since $\boldsymbol{\Sigma}_k$ is a positive definite matrix, (61) is the inner volume of an ellipsoid in a p dimensional space.

We now conjecture that this ellipsoid $\mathcal{D}$ is the smallest region, in which the delay perception lies with probability γ, i.e., the probability of $\boldsymbol{w}_i$ being in this region is γ. We use a proof by contradiction to show this. Consider that there exists any other space $\mathcal{E}$ which is smaller than $\mathcal{D}$, and the probability of $\boldsymbol{w}_i$ being in this region is γ. We can partition $\mathcal{E}$ into two parts $\mathcal{A} = \mathcal{E} \cap \mathcal{D}$ and $\mathcal{E}_2 = \mathcal{E} \cap \mathcal{D}'$, where $\mathcal{D}'$ is the complement of the set $\mathcal{D}$. We also define $\mathcal{D}_2 = \mathcal{D} \cap \mathcal{E}'$. We know that

$$\int_{\mathcal{D}} f(\boldsymbol{w}_i) d\boldsymbol{w}_i = \int_{\mathcal{A}} f(\boldsymbol{w}_i) d\boldsymbol{w}_i + \int_{\mathcal{D}_2} f(\boldsymbol{w}_i) d\boldsymbol{w}_i \tag{62}$$

$$= \int_{\mathcal{E}} f(\boldsymbol{w}_i) d\boldsymbol{w}_i = \int_{\mathcal{A}} f(\boldsymbol{w}_i) d\boldsymbol{w}_i + \int_{\mathcal{E}_2} f(\boldsymbol{w}_i) d\boldsymbol{w}_i \tag{63}$$

$$= \gamma. \tag{64}$$

Hence, $\int_{\mathcal{D}_2} f(\boldsymbol{w}_i) d\boldsymbol{w}_i = \int_{\mathcal{E}_2} f(\boldsymbol{w}_i) d\boldsymbol{w}_i$. Since

$$f(\boldsymbol{w}_i) < C_1 \leq f(\boldsymbol{y}) \qquad \forall \boldsymbol{w}_i \in \mathcal{E}_2, \boldsymbol{y} \in \mathcal{D}_2, \tag{65}$$

using (57) and (58) we have

$$\int_{\mathcal{E}_2} d\boldsymbol{w}_i < \int_{\mathcal{D}_2} d\boldsymbol{w}_i. \tag{66}$$

This means that the set $\mathcal{E}$ has a bigger volume than $\mathcal{D}$, which is a contradiction to our first assumption. This proves that region $\mathcal{D}$ is the smallest region in $\mathbb{R}^{p+1}$ that has the probability γ.

Next, we find the relation between C_2 and γ. γ can be defined as $\int_{\mathcal{D}} f(\boldsymbol{w}_i)d\boldsymbol{w}_i$ and can be calculated using chi-square distribution [31]. The region $\mathcal{D}$ can be written as

$$D = \big\{\boldsymbol{w}_i | (\boldsymbol{w}_i - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1} (\boldsymbol{w}_i - \boldsymbol{\mu}_k) \leq Q_p(\gamma)\big\}, \tag{67}$$

where $Q_p(\gamma)$ is the quantile function of the chi-square distribution with p degrees of freedom. It is defined as

$$Q_p(\gamma) = \inf \Big\{ x \in \mathbb{R} | Pc \leq \int_0^x \chi_p^2(u)du \Big\}. \tag{68}$$

Having defined the confidence region based on γ, we now must find the edges of this ellipsoid. We know that the center of this ellipsoid is μ_k. We need to solve the following optimization problem:

$$\min_{\boldsymbol{w}_i} \text{ or } \max_{\boldsymbol{w}_i} e_i^T \boldsymbol{w}_i, \qquad \text{s.t. } \boldsymbol{w}_i \in \mathcal{D}, \tag{69}$$

where $\boldsymbol{e}_i$ is a unit vector in $\mathbb{R}^p$, having 1 in its ith element and zero otherwise. Using KKT conditions for solving the above problem, we have:

$$e_i + \lambda \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{w}_i - \boldsymbol{\mu}_k) = 0, \tag{70a}$$

$$(\boldsymbol{w}_i - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{w}_i - \boldsymbol{\mu}_k) \leq Q_p(\gamma), \tag{70b}$$

$$\lambda\Big((\boldsymbol{w}_i - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{w}_i - \boldsymbol{\mu}_k) - Q_p(\gamma)\Big) = 0, \quad \lambda \geq 0. \tag{70c}$$

The inequality in (70b) is tight. With some algebraic manipulation, we have $\boldsymbol{w}_i - \boldsymbol{\mu}_k(i) = \frac{1}{\lambda}\boldsymbol{\Sigma}\boldsymbol{e}_i$, and so, $\frac{1}{\lambda^2}\mathbf{e}_i^T \boldsymbol{\Sigma}_k \boldsymbol{\Sigma}_k^{-1} \boldsymbol{\Sigma}_k \mathbf{e}_i = Q(\gamma)$. Therefore,

$$\boldsymbol{w}_i = \pm\sqrt{\frac{Q(\gamma)}{\boldsymbol{e}_i^T \, \boldsymbol{\Sigma}_k \, \boldsymbol{e}_i}} \boldsymbol{\Sigma}_k \boldsymbol{e}_i + \boldsymbol{\mu}_k, \tag{71}$$

$$\lambda = \pm\sqrt{\frac{\mathbf{e}_i^T \boldsymbol{\Sigma}_k \mathbf{e}_i}{Q(\gamma)}}, \tag{72}$$

$$\mathbf{e}_i^T \boldsymbol{w}_i = \pm\sqrt{Q(\gamma)\mathbf{e}_i^T \boldsymbol{\Sigma}_k \mathbf{e}_i} + \mu_k(i). \tag{73}$$

If λ is positive, we can find the maximum which is $+\sqrt{Q(\gamma)\mathbf{e}_i^T \boldsymbol{\Sigma}_k \mathbf{e}_i} + \boldsymbol{\mu}_k(i)$, and if λ is negative, we can find the minimum which is $-\sqrt{Q(\gamma)\mathbf{e}_i^T \boldsymbol{\Sigma}_k \mathbf{e}_i} + \boldsymbol{\mu}_k(i)$. Here, $\boldsymbol{\mu}_k(i)$ is the ith element of $\boldsymbol{\mu}_k$. If we set $i = p+1$, then the delay perception of user i is in the following range:

$$-\sqrt{Q(\gamma)\boldsymbol{e}_{p+1}^T \boldsymbol{\Sigma}_k \boldsymbol{e}_{p+1}} < \beta_i - \mu_k^D < \sqrt{Q(\gamma)\boldsymbol{e}_{p+1}^T \boldsymbol{\Sigma}_k \boldsymbol{e}_{p+1}}, \tag{74}$$

at least with probability γ. Hence, Theorem 1 is proved.

APPENDIX B
PROOF OF THEOREM 2

Since the queuing delay is much smaller than the duration of each time slot, we can assume that each packet arriving at a specific time slot will be served at the same time slot. For analyzing the packet delay, we consider a packet that just arrives in the system in time slot τ_k, and find $\text{Pr}(D > D_i^{\max})$ for this packet. When this packet arrives, there are m packets in the system. From lemma 1, we know that the serving time will be an exponential random variable. Since the exponential distribution is memoryless, there is no distinction between a packet already in service and the other packets. Therefore, the waiting time for the packet that has just arrived is the summation of m exponential distributions. Also, the transmission delay for this packet will be another exponential random variable. Hence, the delay of a packet which arrives at time slot τ_k while there are m packets in the system can be written as:

$$d(\tau_k, m) = t_s + t_1(\tau_k) + t_2(\tau_k) + \cdots + t_{m-1}(\tau_k) + t_c(\tau_k), \tag{75}$$

where $t_i(\tau_k)$ is the service time for packet i in the queue, and $t_c(\tau_k)$ is the service time for packet already in service. Also, t_s is the service time for the packet that has just arrived. we seek to find $\text{Pr}(d(\tau_k, m) > D_i^{\max})$ which can be written as

$$\begin{aligned}\text{Pr}\big(d(\tau_k, m) > D_i^{\max}\big) &= \sum_{m,k} \text{Pr}(D > D_i^{\max}|m, \tau_k)\text{Pr}(m, \tau_k) \\ &= \sum_{m,k} \text{Pr}(D > D_i^{\max}|m, \tau_k)\text{Pr}(m|\tau_k)\text{Pr}(\tau_k).\end{aligned} \tag{76}$$

The probability that there are m users in an M/M/1 queue at time slot τ_k, i.e. $\text{Pr}(m|\tau_k)$, can be written as (see [32]):

$$\text{Pr}(m|\tau_k) = \left(\frac{a_i(\tau_k)}{r_i(\tau_k)}\right)^m \left(1 - \frac{a_i(\tau_k)}{r_i(\tau_k)}\right). \tag{77}$$

Since we assumed the time slots have equal lengths, the packets arrive at each time slot with equal probability of $\text{Pr}(\tau_k) = \frac{1}{t}$, where t is the total number of time slots.

The sum of $m + 1$ identically independent exponential random variables with the mean $\frac{1}{r_i(\tau_k)}$ is a gamma random variable. Consequently, if the users arrive at time slot τ_k while there are m users in the system at the time of arrival, the distribution of delay is

$$f_D(\phi|m, \tau_k) = \frac{r_i(\tau_k)^{m+1}}{\Gamma(m+1)} \phi^m e^{-r_i(\tau_k)\phi}. \tag{78}$$

As a result, we can write the probability of delay exceeding a threshold $D_i^{\max}$ as

$$\Pr(D > D_i^{\max}) = \int_{D_i^{\max}}^{\infty} \sum_{m,k} f_D(\phi|m,\tau_k)\Pr(m|\tau_k)\Pr(\tau_k)d\phi \tag{79}$$

$$= \int_{D_i^{\max}}^{\infty} \frac{1}{t} \sum_{m,k} \frac{r_i(\tau_k)^{m+1}}{m!} \phi^m e^{-r_i(\tau_k)\phi} (\frac{a_i(\tau_k)}{r_i(\tau_k)})^m (1 - \frac{a_i(\tau_k)}{r_i(\tau_k)}) d\phi \tag{80}$$

$$= \frac{1}{t} \sum_{k=1}^{t} \int_{D_i^{\max}}^{\infty} (r_i(\tau_k) - a_i(\tau_k)) e^{-r_i(\tau_k)\phi} \sum_{m=0}^{\infty} \frac{\big(\phi a_i(\tau_k)\big)^m}{m!} d\phi \tag{81}$$

$$= \frac{1}{t} \sum_{k=1}^{t} \int_{D_i^{\max}}^{\infty} (r_i(\tau_k) - a_i(\tau_k)) e^{-\big(r_i(\tau_k) - a_i(\tau_k)\big)\phi} d\phi \tag{82}$$

$$= \frac{1}{t} \sum_{k=1}^{t} e^{-\big(r_i(\tau_k) - a_i(\tau_k)\big) D_i^{\max}}, \tag{83}$$

which proves the theorem.

References

[1] A. T. Z. Kasgari, W. Saad, and M. Debbah, “Brain-aware wireless networks: Learning and resource management,” in *Proc. 51th Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA*, Nov 2017.

[2] E. Gobbetti and R. Scateni, “Virtual reality: Past, present, and future,” *Virtual environments in clinical psychology and neuroscience: Methods and techniques in advanced patient-therapist interaction*, Nov 1998.

[3] M. Chen, W. Saad, and C. Yin, “Virtual reality over wireless networks: Quality-of-service model and learning-based resource management,” *arXiv preprint arXiv:1703.04209*, Mar 2017.

[4] O. Semiari, W. Saad, S. Valentin, M. Bennis, and H. V. Poor, “Context-aware small cell networks: How social metrics improve wireless resource allocation,” *IEEE Transactions on Wireless Communications*, vol. 14, no. 11, pp. 5927–5940, Nov 2015.

[5] H. Intraub, “Rapid conceptual identification of sequentially presented pictures,” *Journal of Experimental Psychology: Human Perception and Performance*, vol. 7, no. 3, pp. 604–610, 1981.

[6] K. ur Rehman Laghari, R. Gupta, S. Arndt, J. N. Antons, R. Schleicher, S. Möller, and T. H. Falk, “Neurophysiological experimental facility for quality of experience (QoE) assessment,” in *Proc. of IFIP/IEEE International Symposium on Integrated Network Management (IM 2013), Ghent, Belgium*, May 2013, pp. 1300–1305.

[7] I. Wechsung and K. De Moor, “Quality of experience versus user experience,” in *Quality of Experience: Advanced Concepts, Applications and Methods*, S. Möller and A. Raake, Eds. Springer International Publishing, 2014, pp. 35–54.

[8] T. Zhao, Q. Liu, and C. W. Chen, “Qoe in video transmission: A user experience-driven strategy,” *IEEE Communications Surveys & Tutorials*, vol. 19, no. 1, pp. 285–302, First quarter 2017.

[9] Y. Chen, K. Wu, and Q. Zhang, “From QoS to QoE: A tutorial on video quality assessment,” *IEEE Communications Surveys & Tutorials*, vol. 17, no. 2, pp. 1126–1165, Second quarter 2015.

[10] M. Alam, D. Yang, K. Huq, F. Saghezchi, S. Mumtaz, and J. Rodriguez, “Towards 5G: Context aware resource allocation for energy saving,” *Jour. of Signal Proc. Systems*, vol. 83, no. 2, pp. 279–291, May 2016.

[11] Y. Lin, R. Zhang, C. Li, L. Yang, and L. Hanzo, “Graph-based joint user-centric overlapped clustering and resource allocation in ultra dense networks,” *to appear in IEEE Transactions on Vehicular Technology*, 2017.

[12] J. Zhao, Y. Liu, K. K. Chai, M. Elkashlan, and Y. Chen, "Matching with peer effects for context-aware resource allocation in D2D communications," *IEEE Communications Letters*, vol. 21, no. 4, pp. 837–840, April 2017.

[13] M. Zalghout, S. Abdul-Nabi, A. Khalil, M. Helard, and M. Crussiere, "Optimizing context-aware resource and network assignment in heterogeneous wireless networks," in *Proc. of IEEE Wireless Communications and Networking Conference (WCNC), San Francisco, CA, USA*, March 2017, pp. 1–6.

[14] E. Bastug, M. Bennis, and M. Debbah, "Living on the edge: The role of proactive caching in 5G wireless networks," *IEEE Communications Magazine*, vol. 52, no. 8, pp. 82–89, Aug 2014.

[15] P. Makris, D. N. Skoutas, and C. Skianis, "A survey on context-aware mobile and wireless networking: On networking and computing environments' integration," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 1, pp. 362–386, First 2013.

[16] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context aware computing for the Internet of Things: A survey," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 414–454, First quarter 2014.

[17] M. Proebster, M. Kaschub, and S. Valentin, "Context-aware resource allocation to improve the quality of service of heterogeneous traffic," in *Proc. of IEEE International Conference on Communications*, Kyoto, Japan, Jul. 2011.

[18] L. Huang, "System intelligence: Model, bounds and algorithms," in *Proc. of the 17th ACM International Symposium on Mobile Ad Hoc Networking and Computing*. New York, NY, USA: ACM, 2016, pp. 171–180.

[19] C. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer Information Science and Statistics, 2007.

[20] M. Chen, U. Challita, W. Saad, C. Yin, and M. Debbah, "Machine learning for wireless networks with artificial intelligence: A tutorial on neural networks," *arXiv preprint arXiv:1710.02913*, Oct. 2017.

[21] S. Petkoski, A. Spiegler, T. Proix, and V. Jirsa, "Effects of multimodal distribution of delays in brain network dynamics," *BMC Neuroscience*, vol. 16, no. Suppl 1, p. P109, 2015.

[22] Y. Yang, L. T. Park, N. B. Mandayam, I. Seskar, A. L. Glass, and N. Sinha, "Prospect pricing in cognitive radio networks," *IEEE Transactions on Cognitive Communications and Networking*, vol. 1, no. 1, pp. 56–70, Oct. 2015.

[23] T. K. Moon, "The expectation-maximization algorithm," *IEEE Signal Processing Magazine*, vol. 13, no. 6, pp. 47–60, Nov 1996.

[24] G. Mclachlan and T. Krishnan, "The EM algorithm and extensions," vol. 382, 03 1998.

[25] 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.

[26] 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, April 2008.

[27] K. Seong, M. Mohseni, and J. M. Cioffi, "Optimal resource allocation for ofdma downlink systems," in *Proc. of IEEE International Symposium on Information Theory, Seattle, WA, USA*, July 2006, pp. 1394–1398.

[28] W. Yu and R. Lui, "Dual methods for nonconvex spectrum optimization of multicarrier systems," *IEEE Transactions on Communications*, vol. 54, no. 7, pp. 1310–1322, July 2006.

[29] S. Boyd and A. Mutapcic, "Subgradient methods," *Lecture notes of EE364b, Stanford University, Winter Quarter*, vol. 2007, 2006.

[30] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning*. Springer series in statistics New York, 2001, vol. 1.

[31] J. Berger, "A robust generalized bayes estimator and confidence region for a multivariate normal mean," *The Annals of Statistics*, pp. 716–761, 1980.

[32] A. Papoulis and S. U. Pillai, *Probability, random variables, and stochastic processes*. Tata McGraw-Hill Education, 2002.