Reinforcement Learning With Network-Assisted Feedback for Heterogeneous RAT Selection

Duong D. Nguyen, Hung X. Nguyen, *Member, IEEE*, and Langford B. White, *Senior Member, IEEE*

***Abstract*— Future wireless networks (e.g., 5G) will consist of multiple radio access technologies (RATs). In these networks, deciding which RAT users should connect to is not a trivial problem. Current fully distributed algorithms although guaranteeing convergence to equilibrium states, are often slow, require high exploration times and may converge to undesirable equilibria. To overcome these limitations, this paper develops a network feedback framework that uses limited network-assisted information to improve efficiency of distributed algorithms for RAT selection problem. We prove theoretically that a fully distributed algorithm developed within this framework is guaranteed to converge to a set of correlated equilibria. Our framework guarantees convergence in self-play even when only a single user applies the algorithm. Simulation results demonstrate that our solution: 1) is highly efficient with fast convergence time and low signaling overheads while achieving competitive, if not better, performance both in fairness and utility, as well as achieving lower per-user switchings than state-of-the-art algorithms; and 2) can flexibly support a wide range of network-assisted feedback. The simulations demonstrate the effectiveness of our solution in a heterogeneous environment, where users may potentially apply a number of different RAT selection procedures.**

***Index Terms*— RAT selection, heterogeneous wireless networks, reinforcement learning, network feedback, game theory, correlated equilibrium.**

I. Introduction

TO COPE with the exponential growth of mobile traffic, network operators are continuously looking for ways to leverage spectrum across available radio access technologies (RATs) [1]. Multiple wireless network architectures (e.g., LTE, UMTS, WiFi, femto, etc) are being deployed concurrently in the current and next generation wireless networks [2]. At the same time, mobile devices are increasingly equipped with multiple RATs that can connect to and choose among the different base stations (BSs) with different access technologies. Deciding which technology, and which individual BS supporting that technology mobile users should connect to, is known as the RAT selection problem [3], and is a topic of much current research in LTE and 5G [4].

RAT selection is often addressed in the literature by using either a network-centric or a user-centric approach. In a network-centric approach [5]–[7], a centralised controller assigns BSs to users in a service area. This approach is suitable in a software defined networking environment where a controller has a complete logical view of the network [7]. It, however, requires collaboration between all wireless networks and users – exchanging significant communication overheads. When the networks are run by competing operators, such close collaboration may not at all be possible. A user-centric approach can overcome this problem by implementing the network-selection algorithms at the user side [8]–[16]. When intelligence is pushed to the network edge, rational users select their RAT in order to selfishly maximize their utility. However, as users have no information on BS load conditions, their decisions may be in no user's long-term interest, causing performance degradation and sometimes oscillation or instability. To guarantee convergence, most existing distributed RAT selection algorithms [8]–[14] require that all users know the selection histories of other users, and are able to determine their own throughputs given other users' choices. This assumption implies that each user knows the instantaneous rates of the other users. The guaranteed convergence therefore comes at the cost of increased complexity, signalling and communication load.

To reduce the communication overheads, a fully distributed algorithm such as a reinforcement learning (RL) based algorithm [15]–[17] can be used. This algorithm does not require the users to know anything about other users. Indeed, users do not even need to know that they are parts of a RAT selection "game". Each user learns about the game by observing only its own achieved payoffs. Over time, using only this information, a user can rationally choose the best course of actions to maximize its utility. Under mild conditions of finite payoffs and of unchanged network conditions, the RL-based regret minimisation algorithm in [17] is guaranteed to converge to a stable set of equilibria. We refer to the algorithm [17] as *Hart's RL-based algorithm* throughout this paper. Despite this attractive property, Hart's RL-based algorithm suffer from problems of slow convergence, and of convergence to socially sub-optimal equilibria, making them unsuitable for RAT selection in real networks where the environment can change quickly [16].

One of the promising ideas to overcome the shortcomings of the Hart's RL-based algorithm is to use external information to aid users in their estimation of the game [18]. In engineering

Manuscript received September 21, 2016; revised February 1, 2017, May 17, 2017, and June 8, 2017; accepted June 8, 2017. Date of publication June 27, 2017; date of current version September 8, 2017. This work was supported by the Australian Research Council Linkage under Grant LP140100489. The associate editor coordinating the review of this paper and approving it for publication was T. Melodia. *(Corresponding author: Duong D. Nguyen.)*

D. D. Nguyen and L. B. White are with the School of Electrical and Electronic Engineering, The University of Adelaide, Adelaide, SA 5005, Australia (e-mail: duong.nguyen@adelaide.edu.au; lang.white@adelaide.edu.au).

H. X. Nguyen is with the Teletraffic Research Centre, The University of Adelaide, Adelaide, SA 5005, Australia (e-mail: hung.nguyen@adelaide.edu.au).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TWC.2017.2718526

systems such as wireless networks, such external information is often readily available at the network BSs. We therefore propose to use such information to improve distributed RAT selections. A real challenge is to design a method that guarantees fast convergence and good performance, while signalling and processing burden remains acceptable. To achieve this balance, in our solution users select their RAT depending on their individual observations, as well as feedback provided by the network. By tuning the network information, operators can also influence user decisions to achieve their objectives and avoid undesirable network states.

Our main contributions in this paper are as follows:

1) *A Network Feedback Model:* We develop a network feedback model that uses network-assisted information to improve the performance of the Hart's RL-based algorithm in [17] for RAT selection. We show that our framework can be applied to multiple types of feedback. To our best knowledge, this is the first work that introduces network-assisted information in a RL-based algorithm for distributed RAT selection. Our framework accommodates a heterogeneous environment, where not all users have the same learning strategy and utility function. In practice, different users pursue different objectives and thus may use different learning strategies or utility functions. Our solution guarantees no-regret payoff in the long run for any user adopting it, irrespective of the behaviour of other users. Using our self-learning technique, any independent user can individually interface with networks to obtain the desired feedback and implement a no-regret based strategy. This adaptive scheme does not require any modification of the current mobile network standards and can be easily implemented in software running on a end-user device.
2) *A Novel Fully Distributed RAT Selection Algorithm:* Using our framework, we develop a fully distributed algorithm which computes a correlated equilibrium solution. If all the users follow our algorithm, the empirical distribution of joint actions is guaranteed to converge to a set of correlated equilibria (CE), which are generalised Nash equilibria (NE).
3) *Comprehensive Practicality Study:* We perform extensive simulations with realistic network scenarios to evaluate our algorithm. Simulations demonstrate that our solution is highly efficient with fast convergence and low overheads. Our solution achieves competitive, if not better performance, both in fairness and utility, as well as per-user RAT switching, compared to state-of-the-art algorithms. A thorough evaluation of adaptive RAT selection algorithms including the one presented in this paper is provided in [19].

The rest of this paper is organised as follows. In Section II, we discuss the related work. In Section III, we present our game model. We formally propose our reinforcement learning with network-assisted feedback in Sections IV. The evaluation is presented in Section V. Finally, we conclude the paper in Section VI.

II. Related Work

This section discusses the major differences between our solution and the most recent distributed RAT selection schemes.

A. Game Theory Applications in RAT Selection

Game theory is a mathematical tool to model the interaction of decision makers with conflicting interests, and has been widely used to both design, and to study the dynamics of network selection problems in wireless networks (for a survey refer to [20]). Most related works formulate the problems as non-cooperative games and propose iterative procedures that converge to NE [8], [9], [16]. Unfortunately, most algorithms that aim to reach NE do not always guarantee convergence [21]. Substantial modifications of Nash-based algorithms are often required to achieve guaranteed behaviours for RAT selection games [8]–[14]. A hysteresis mechanism, where a user changes its RAT only if its expected throughput is higher than a threshold or if a network controller allows the move [9], is used in [8]–[10] to guarantee convergence to NE. Authors in [11]–[14] propose a network-assisted scheme, where additional knowledge of the network conditions is broadcast to all users, to aid them in their decisions.

Only a number of previous works [14], [18] consider the situation where players achieve co-ordination between their strategies, either directly or indirectly, in order to get better payoffs at the correlated equilibria. A CE is a generalised Nash equilibrium where each player chooses their actions based on their common knowledge of the game's history [22]. By allowing the players to coordinate their actions, a CE can provide a balance between the non-cooperative solution (where all the players work independently but may yield poor performance) and the fully cooperative solution (which requires coordination between players but can be highly efficient). In fact, the set of CE is more natural than the set of NE in decentralised adaptive learning environments since the common history observed by all players can serve as a natural coordination mechanism [17].

Several distributed algorithms can be used to achieve convergence to stable CE in a RAT game, including regret matching in [23] and its fully distributed variant – a reinforcement learning based regret minimisation algorithm in [17]. In Hart's RL-based algorithm, a user learns to make optimal decisions directly from its own past rewards without requiring any extra information. Contrary to the uncertainty of algorithms that aim to achieve convergence to NE, the Hart's RL-based algorithm in [17] converge to the set of CE almost surely. The main drawback of the Hart's RL-based algorithm in [17] is that although guaranteeing convergence to the CE set, it often requires long convergence time and can converge to a sub-optimal equilibrium. By this, we mean an outcome that yields lower payoffs, unfair resource allocation, or inefficient utilisation of available resources [18].

There are several possible approaches to theoretically analyse the convergence of RL-based algorithms. A method based on direct analysis was developed in [17]. Majority of subsequent proofs have been based on the stochastic approximation technique (i.e. averaging theory), such as the one used in [16]. More recently, Benam *et al.* [24] use the theory of

differential inclusion (DI) to prove the convergence of adaptive procedures used in game theory. The proofs in this paper are an application of [24] to RAT selection games. The use of DI technique yields a considerably simpler and shorter proof as compared to the classical approach in [17].

The DI based stochastic approximation method is a generalisation of ordinary differential equation approach used in standard adaptive systems. DI is particularly suitable to study the asymptotic trajectory of the iterative process in game-theoretic learning where the information available to a player is inaccurate or missing. It provides a rich set of theoretical tools that allows us to study the convergence behaviour of multiple game settings including games with imperfect rewards that must be estimated from noisy observations, and when the strategies of the other plays are unknown. DI has been used in [25]–[27] for RL-based algorithms but to the best of our knowledge, this is the first work that this method is applied in RL procedure in which the "external" information is incorporated in the decision rule.

B. Using External Feedback to Improve RAT Selection

There have been several RAT selection algorithms proposed that use some form of network feedback [8]–[14]. In all of these approaches, the network runs a centralised algorithm to determine the controllable parameters (such as users' instantaneous rate [8]–[10], network suggestions [9], [11], traffic loads [12], quality of services [13] and offered bandwidths and costs [14]) for each user. Each BS then broadcasts these parameters to all the users in their coverage area, including those that are not actively served by it. The high amount of information exchange, excessive signalling and communication load all contribute to make these approaches unattractive in practice.

Several attempts have been made to ensure that the signalling overheads among BSs and users is kept at the minimum level by using RL-based algorithms [15], [16]. Two problems with these approaches are slow and arbitrary convergence [18]. Another major issue is that a very high number of RAT switching per-user is required due to the lack of information on global network load conditions. This is because each user must try many different actions in order to develop an understanding of the global structure of the RAT "game".

Our solution in this paper follows the regret-based principles with significant modifications to accelerate convergence speed, reduce exploration times and avoid undesirable equilibria. We show in this paper, using extensive simulations, that:

1) The overall signalling overheads of our algorithm are significantly less than those in [8]–[16], which are the state-of-the-art RAT selection algorithms.
2) Our algorithm has a fast convergence rate with a small number of per-user RAT switchings, whilst achieving competitive performance both in fairness and utility.
3) Lastly, our algorithm is one of the few algorithms of which we are aware, that can flexibly support a wide range of feedback, which can be defined according to the network operators' policies. Existing algorithms [8]–[13] do not inherently support objective functions that are not directly related to throughput, and may require significant modifications to incorporate other objective functions. This will be described in detail in Section V.B.

III. Game Model

A. Heterogeneous Network Throughput Model

We consider a heterogeneous wireless network (HWN) consisting of M base stations (BSs) and N end-user equipments (users). We use BS to denote any network node that connects directly to users such as a base station in WCMDA/LTE network or an access point in WiFi. In this paper, we are primarily interested in user downlink throughput as the utility and use the same models as in [8]–[10] for different RATs. We divide the throughput models into two subclasses.

1) Class-1 (Proportional-Fair Model): Under this class, each user obtains a different user-specific throughput which is a function of its instantaneous physical (PHY) rate and the number of users sharing the same BS. The throughput of a user (i.e., user A) choosing BS k is

$$\bar{U}_A^k = \frac{R_A^k}{n^k}, \tag{1}$$

where R_A^k is the PHY rate of user A on BS k and n^k is the number of users on k. This class is suitable to model time/bandwidth-fair access technologies such as 3G/4G cellular networks.

2) Class-2 (Throughput-Fair Model): Under this class, all users connected to the same BS will have the same per-user throughput. The throughput of a user (i.e., user A) connected to BS k can be expressed as

$$\bar{U}_A^k = \left(\sum_{a=1}^{n^k} \frac{1}{R_a^k} \right)^{-1}. \tag{2}$$

This class is suitable for throughput-fair access technologies such as WiFi.

3) Realistic Throughput Model: Most existing works assume that the user knows its actual throughput in (1) and (2). By actual throughput, we mean the long-term average throughput that a user indeed experiences on a wireless network. In reality, the actual throughput of each user is influenced by not only the link quality (i.e., the signal to noise ratio) but also many other factors such as traffic load and interference from the surrounding environment. Therefore, in practice, the user only knows its sampled throughput, not the actual value. The sampled value can be modelled as a random variable where the actual throughput given in (1) or (2) is the mean, which is computed at the network side. At any one time, depending on the number of users per base station, the distribution of traffic load and sampling technique, instantaneous throughput observed by the user may vary from the mean.

Assumption 1: To model the real user observed throughput, we follow the most recently proposed instantaneous throughput model in [10], where the user observed throughput is assumed to follow a Gaussian distribution. Other distribution could be used but is outside the scope of this paper.

Under the Gaussian assumption, the mean is equal to the actual throughput and the standard deviation is equal to the

TABLE I
SUMMARY OF MAIN NOTATIONS USED IN THIS PAPER

Symbol	Semantics
N	Number of users
M	Number of base stations
n^k	Number of users on base station k
R_A^k	Physical (PHY) rate of user A to BS k
$\bar{U}_A^k$	The actual throughput of user A choosing BS k
U_A^k	The instantaneous throughput of user A choosing BS k
$s = (i, \ell)$	The action taken by all players, where i is the action of player A and ℓ is the actions of the others
$U(s)$	The payoff achieved by player A when the overall action taken by all players is s
$z = (x, y)$	The probability of the action taken by all players, where x is the probability of action of player A and y is the probability of action of all other players except player A
Y_τ^k	The network-assisted feedback that BS k sends to user A at time τ
$\bar{U}_\tau^k$	The BS computed throughput that BS k sends to user A at time τ
n_τ^k	The number of users on BS k at time τ
$B_t(j, k)$	The user estimated regret in average payoff of player A up to time t for not playing k in stead of j
$Y_t(j, k)$	The network measured regret in average payoff for player A up to time t for not playing k in stead of j
$p_t(k)$	The probability of choosing BS k at time t by player A
$\bar{z}_t(s)$	The empirical distribution of join action s of all players until time t

product of the noise value e and the actual throughput [10]. Thus, instantaneous throughput rate of a user A choosing BS k is a Gaussian random variable:

$$U_A^k \sim \mathsf{N}(\bar{U}_A^k, \sigma^2),$$

where $\sigma = e \times \bar{U}_A^k$ and $0 < e < 1$. In our game theoretic solution, the network provides every user with the actual throughput $\bar{U}_A^k$ calculated at the BS (BS computed throughput) rather than the randomly fluctuating rates U_A^k observed by the user (user observed throughput).

B. RAT Selection Model

In the following, we adopt the notation of [24]. We model the RAT selection as a repeated game where the players (mobile users) aim to maximize their long-run average payoffs (throughput). We consider a game with N players denoted by the set $\mathcal{N} = \{1, \ldots, N\}$ for some (finite) integer $N \geq 2$. Each player a has its set of finite actions $\mathcal{S}_a$ (set of available BSs) and we denote by $\mathcal{S} = \mathcal{S}_1 \times \cdots \times \mathcal{S}_N$, the set of all strategies for all players, i.e. the Cartesian product of all players' possible actions. We view the game from the point of view of player A – a randomly selected player among the set of all players. Let $\mathcal{I} = \mathcal{S}_A$ denote the set of actions of player A and $\mathcal{L} = \mathcal{S} \backslash \mathcal{S}_A$ the set of actions of all other players. Denote by X, the set of all probability mass functions (pmf) on $\mathcal{I}$ and Y the set of pmf on $\mathcal{L}$. Let Z denote the set of pmf on $\mathcal{S}$, then $X \times Y$ is a subset of Z comprised of all pmf of the form $z = (x, y)$ where $x \in X$ and $y \in Y$, i.e. all pmf of the probability of the action of player A and the actions of all other players taken together. The main notations that we use in this paper are summarised in Table I.

Let $U_A : \mathcal{S} \to \mathbb{R}$ denote the payoff achieved by player A when the overall action taken by all players is $s \in \mathcal{S}$. We represent a strategy in the form $s = (i, \ell)$ where i is the action of player A and ℓ is the action of all other players. We will consider the general formulation of the game where users apply mixed strategies over the possible selection set $\mathcal{S}$. Under randomised actions with overall probability (pmf) $z \in Z$, the payoff obtained by player A is defined as

$$U_A(z) = \sum\nolimits_{s \in \mathcal{S}} z(s) U_A(s).$$

The RAT selection game then can be denoted by $\mathcal{G} = (\mathcal{N}, (\mathcal{S}_A)_{A \in \mathcal{N}}, (U_A)_{A \in \mathcal{N}})$. In our game model, each player A knows only its set of actions $(\mathcal{S}_A)$ and its stream of payoffs (U_A) received in the past. Players are not aware of other players' actions and payoffs. Instead, players can observe the number of other players choosing the same action after each action, as explained later in Section IV.A. In this paper, we are interested in a popular notion of rationality that generalises the Nash equilibrium, known as a correlated equilibrium. CE is an optimality concept introduced by Aumann [22] and is proven to exist for any finite games with bounded payoffs [28]. It is relevant to probabilistic games, namely where strategies are determined probabilistically, and is a precise statement of rationality in this setting [22].

Definition 1: A probability distribution ψ defined on $\mathcal{S}$ is said to be a correlated equilibrium for the game $\mathcal{G}$ if for every player $A \in \mathcal{N}$, and for every pair of action $j, k \in \mathcal{I}$, it holds that[1]

$$\sum\nolimits_{s \in \mathcal{S}: i = j} \psi(s)(U_A(k, \ell) - U_A(j, \ell)) \leq 0, \tag{3}$$

CE models possible correlation or co-ordination between players' actions compared to the usual strategic equilibrium of Nash, where all players act independently. A CE results if each player does not benefit from choosing any other probability distribution over its actions, provided that all the other players do likewise. When each player chooses their action independently of the other players, or without any implicit co-ordination mechanism, a CE is also a NE.

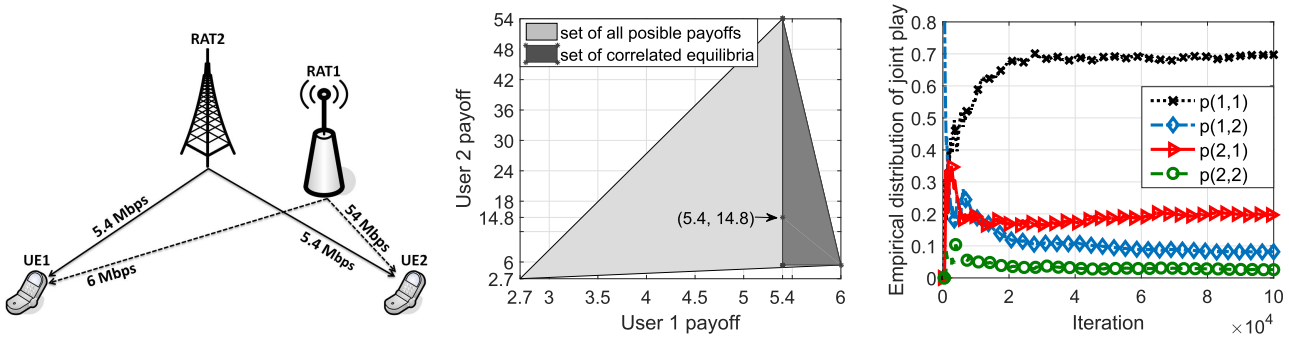
C. Computing the Correlated Equilibria

A fully distributed algorithm that can be used to reach the CE solution is the RL-based regret minimisation procedure in [17]. The key idea of this method is to adjust the player's action probability proportional to the "regrets" for not having played other actions. Specifically, for any two actions $j \neq k \in \mathcal{I}$ at any time t, the regret of player A for not playing k is

$$C_t(j, k) = \frac{1}{t} \sum_{\tau \leq t: i_\tau = j} U(k, \ell_\tau) - \frac{1}{t} \sum_{\tau \leq t: i_\tau = j} U(j, \ell_\tau), \tag{4}$$

where i_τ denotes the action taken by player A at time τ (i.e. $i_\tau = j$ means player A selects BS j at time τ) and ℓ_τ denotes the actions of the others at time τ. This is the change in the average payoff that player A would observed if playing k instead of j every time it played j in the past. Note that the notations should have the subscript A to indicate that it refers

[1] We write $\sum_{s \in \mathcal{S}: i = j}$ for the sum over all s in $\mathcal{S}$ whose i equals j. Similar notations are used elsewhere in the paper.



(a) (b) (c)

Fig. 1. (a) An example of RAT selection in a mixed 4G/WiFi network. (b) The set of correlated strategies and CE in payoff space. (c) The empirical distribution of join play by Hart's RL-based algorithm.

to player A. Since we view the game from player A's point of view, we drop this subscript to keep the notation simple (thus, we write C_t and U in stead of $C_{A,t}$ and U_A, and so on). Similar notations are used in the rest of the paper. Since player A only has access to the payoffs corresponding to actions it actually took, it cannot compute the first term. Thus, the regret in (4) needs to be replaced by an estimate that can be computed on the basis of the available information, via

$$B_t(j,k) = \frac{1}{t}\sum_{\tau\le t: i_\tau = k}\frac{p_\tau(j)}{p_\tau(k)}U(k,\ell_\tau) - \frac{1}{t}\sum_{\tau\le t: i_\tau = j} U(j,\ell_\tau), \quad (5)$$

where p_τ denotes the play probabilities of player A at time τ (i.e., $p_\tau(k)$ is the probability of choosing k at time τ). This approximate regret measures the historical difference of the average payoff over the periods when k was used and the periods when j was used [17].

If $i_t = j$ is the action chosen by player A at time t, then the probability distribution that player A chooses an action at time $t+1$ is defined recursively as [17][2]

$$p_{t+1}(k) = \begin{cases} (1-\delta_t)\min\left\{\dfrac{B_t^+(j,k)}{\mu}, \dfrac{1}{m}\right\} + \dfrac{\delta_t}{m} & \text{if } k \neq j, \\ 1 - \sum_{k'\neq j} p_{t+1}(k') & \text{if } k = j, \end{cases} \quad (6)$$

with the initial action probabilities at $t = 1$ uniformly distributed over the set of possible actions; $\mu > 2mG$ is a constant with m being the cardinality of the set $\mathcal{I}$ and G being an upper bound on $|U(s)|$ for all $s \in \mathcal{S}$; $\delta_t = \delta/t^\gamma$, $0 < \delta < 1$ and $0 \le \gamma < 1/4$.

It is proven in [17] that if *every* player chooses their actions according to (6), then the empirical distribution of joint actions s of all players until time t, which is given by[3] $\bar{z}_t(s) = \frac{1}{t}\sum_{\tau=1}^{t}\mathbb{1}_{\{s_\tau = s\}}$, converges almost surely as $t \to \infty$ to the set of CE of the game $\mathcal{G}$. Note that this does not imply convergence to a specific point on the CE set, but that the solution approaches the CE set.

[2]We use the notation $x^+ := \max(x, 0)$ for a real number x throughout this paper (e.g. $B_t^+(j,k) = \max(B_t(j,k), 0)$). The definition is extended to real vectors and matrices elementwise.

[3]Where $\mathbb{1}(.)$ denotes the indicator function.

TABLE II
PAYOFF MATRIX FOR THE RAT SELECTION GAME

	$s_2 = 1$	$s_2 = 2$
$s_1 = 1$	(5.4, 5.4)	(6.0, 5.4)
$s_1 = 2$	(5.4, 54)	(2.7, 2.7)

D. Example of RAT Selection Game

We use the example in Fig. 1(a) to illustrate the concepts introduced so far in this paper. In this example, there are two users and two RATs: WiFi (RAT1) and 4G (RAT2). User 2 is at the cell-center of RAT1 and has a good PHY rate of 54Mbps. User 1 is at the cell-edge location of RAT1 and so obtains a lower PHY rate of 6Mbps. Both users are located at similar distances from RAT2 and thus have the same PHY rate of 5.4Mbps. These PHY rates are also their obtained throughputs when connected alone to these RATs.

The set of actions is denoted by $\mathcal{S} = \{(j,k) : j,k = 1,2\}$ where $s = (j,k)$ means that user 1 chooses RAT j and user 2 chooses RAT k. The payoff functions are the throughput obtained for each user. When both users connect to the WiFi access point, under Class-1 throughput model, they receive a very low throughput of $(1/6 + 1/54)^{-1} = 5.4$ Mbps as given by equation (2) for their WiFi connections. The 4G BS are assumed to use the time-fair protocol (Class-2 throughput model) which allows each user has the same time duration to access to the network. When both users select RAT2, they receive the throughputs that are equal to half of their physical rates for their 4G connection. Using equation (1), the throughput payoff is $5.4/2 = 2.7$ Mbps for each user. We summarise this game in Table II.

Suppose we have a probability distribution p on $\mathcal{S}$ with $p(j,k)$ denoting the joint probability that player 1 chooses RAT j and player 2 chooses RAT k, for $j,k = 1,2$. Substituting the payoffs from table II, equation (3) yields the four linear inequalities

$$p(1,1)\{5.4-5.4\} + p(1,2)\{2.7-6.0\} \le 0 \Leftrightarrow p(1,2) \ge 0,$$
$$p(2,1)\{5.4-5.4\} + p(2,2)\{6.0-2.7\} \le 0 \Leftrightarrow p(2,2) \le 0,$$
$$p(1,1)\{5.4-5.4\} + p(2,1)\{2.7-54\} \le 0 \Leftrightarrow p(2,1) \ge 0,$$
$$p(1,2)\{5.4-5.4\} + p(2,2)\{54-2.7\} \le 0 \Leftrightarrow p(2,2) \le 0.$$

We also have the four inequalities $p(j,k) \ge 0$ for $j,k = 1,2$ and the equality $p(1,1) + p(1,2) + p(2,1) +$

$p(2,2) = 1$ which define a pmf. Then, a CE is a quadruple $(p(1,1),\ p(1,2),\ p(2,1),\ p(2,2))$ that satisfies:

$$\begin{cases} p(2,2) = 0, \\ p(1,1),\ p(1,2),\ p(2,1) \geq 0, \\ p(1,1) + p(1,2) + p(2,1) = 1. \end{cases}$$

Therefore, any solutions of the form $p(1,1) + p(1,2) + p(2,1) = 1$ will be in the set of CE. The corner points $p(1,1) = 1,\ p(1,2) = 1$ and $p(2,1) = 1$ are pure NE whilst the other solutions are mixed NE. Payoff pairs in these pure NE are, respectively, (5.4, 5.4), (6, 5.4) and (5.4, 54). Fig. 1(b) shows the set of all payoff allocations under correlated strategies and under correlated equilibria. The set of correlated strategies (light gray) is the set of all possible combination of players' pure strategies; and the set of CE (dark gray), which is a super set of the NE set, is the triangle with these three NE as vertices.

1) Limitations of the Hart's RL-Based Algorithm in [17] for RAT Selection: We implemented the Hart's RL-based algorithm in [17] and applied it to the RAT selection game in Fig. 1. We encountered the following three undesirable outcomes even on this simple example.

1) **Sub-optimal convergence:** Our implementation of the Hart's RL-based algorithm when applying to the above network leads to the CE point $(p(1,1) = 0.70,\ p(1,2) = 0.08,\ p(2,1) = 0.20,\ p(2,2) = 0.02)$ that yields a payoff pair of (5.4, 14.8) the majority of the time. This equilibrium is neither fair ((5.4, 5.4) provides the best system fairness) nor throughput efficient ((5.4, 54) yields the highest overall throughput, albeit unfairly).
2) **Slow convergence:** The algorithm takes at least 6,000 iterations to converge on a simple 2 base stations – 2 users network! This is a significant problem for RAT selection where network conditions can change quickly, breaking the implicit assumptions of stable environment, required for the RL-based algorithm to converge.
3) **High numbers of switching:** The algorithm also requires up to 400 RAT switchings per user to converge. This is an another major constraint for real network implementation due to the challenge in providing seamless vertical handover between different RATs.

These issues of slow convergence, sub-optimal convergence and high numbers of switching of Hart's RL-based algorithm in [17] motivate the introduction of network-assisted feedback to the reinforcement learning based regret minimisation algorithm in the next section.

IV. Reinforcement Learning With Network-Assisted Feedback

To overcome the limitations of the Hart's RL-based algorithm as observed above, we propose a feedback model that uses network-assisted information from the network BS.[4] The main idea of our solution is to help users estimate their utilities more accurately using the limited information that is readily available at the BS. Using network feedback, the operators can also alter the trajectory of the algorithm. There have been several proposals for using network feedback to improve distributed RAT selection algorithms [8]–[14], but not for RL-based ones.[5] We show empirically via simulation in Section V.A that our algorithm, by using little extra information, achieves a faster convergence rate to the CE set than existing distributed RAT selection algorithms including a recent RL-based algorithm in [16].

A. Using Feedback to Update Network Measured Regret

The types of network feedback varys depending on the objectives of the network designers. In this paper, we use two types of feedback: (1) BS computed throughput $\bar{U}_t^k$, which indicates the actual throughput that a user could receive from the BS k at time t; (2) and the number of users n_t^k, which is the number of users currently connected to the BS k at time t. Providing the actual achievable throughputs can help users make informed decisions that lead to better outcomes by exploiting the actions that yield higher throughputs. Knowing the number of concurrent users at each BS will help users avoid exploring selections that result in poor performances. However, these types of information are not directly available to the end-users.

Since most networks have up-to-date and accurate measurements of these metrics, we propose to use this information to improve the performance of the Hart's RL-based algorithm in [17] for RAT selection. A number of mechanisms to distribute additional feedback information from BS to users have been standardised and can be used for this purpose, including the logical communication channel in IEEE standard 1900.4 [29], and the Access Network Query Protocol (ANQP) in IEEE 802.11u standards [30]. These protocols allow users to query information about the capabilities of the network (such as throughput, packet error rate, available services) prior to performing the authentication process.

As explained in Section III.A, users do not know their actual throughput. Their instantaneous estimations are often very noisy. By using network-assisted feedback, each user can estimate its obtainable throughput $U(k, \ell_t)$ if it switches to another BS k given its current action is $i_t = j \neq k$. The user then can compute network measured regrets $Y_t(j,k)$, which is a measure of the average regret for the user observed by the network at time t for not selecting other BS k instead of the actual BS j every time in the past, as follows.

1) Class-1 Throughput Estimation: Suppose $i_t = j$ is the action chosen by user A at time t. Using (1), the obtainable throughput if user A connects instead to BS k, is equal to R_A^k divided by $(n_t^k + 1)$, the total number of users sharing the BS k at time t, if user A joins.

$$U(k, \ell_t)|_{i_t = j \neq k} = \frac{R_A^k}{n_t^k + 1} \approx \frac{\sum_{\tau \leq t : i_\tau = k} (\bar{U}_\tau^k \times n_\tau^k)}{\nu_t^k \times (n_t^k + 1)}, \quad (7)$$

where $\nu_t^k = \sum_{\tau \leq t} \mathbb{1}_{\{i_\tau = k\}}$ counts how many times BS k has been chosen up to time t. R_A^k is obtained by taking the average

[4]Note that the feedback model is a model not an algorithm and RL is an algorithm.

[5]Note that in general learning theory, RL-based algorithms, their convergence and approximations are well studied.

of $(\bar{U}_\tau^k \times n_\tau^k)$ over ν_t^k – the periods when k was used.

Similarly, the number of users sharing the same BS k at time t can be estimated by taking the average number of users on k over the periods when k was used. That is,

$$n_t^k|_{i_t=j\neq k} = \frac{\sum_{\tau\leq t:i_\tau=k} n_\tau^k}{\nu_t^k}. \tag{8}$$

Replacing (8) into the denominator of (7), the estimate of $U(k,\ell_t)$ in (7) is then

$$\tilde{U}(k,\ell_t)|_{i_t=j\neq k} = \frac{\sum_{\tau\leq t:i_\tau=k}(\bar{U}_\tau^k \times n_\tau^k)}{\sum_{\tau\leq t:i_\tau=k}(n_\tau^k+1)}.$$

The BS observed regret measured at time t for class-1 RAT can be calculated as

$$\begin{aligned} Y_t(j,k) &= \frac{1}{t}\sum_{\tau\leq t:i_\tau=j}\left(\tilde{U}(k,\ell_\tau) - \bar{U}_\tau^j\right) \\ &= \frac{1}{t}\sum_{\tau\leq t:i_\tau=j}\left(\frac{\sum_{\tau\leq t:i_\tau=k}(\bar{U}_\tau^k \times n_\tau^k)}{\sum_{\tau\leq t:i_\tau=k}(n_\tau^k+1)} - \bar{U}_\tau^j\right). \end{aligned} \tag{9}$$

2) Class-2 Throughput Estimation: Suppose $i_t = j$ is the action chosen by user A at time t, then using (2), we obtain the throughput of user A if it connects to another BS k as

$$\begin{aligned} U(k,\ell_t)|_{i_t=j\neq k} &= \left(\sum_{a=1}^{n_t^k}\frac{1}{R_a^k} + \frac{1}{R_A^k}\right)^{-1} \\ &= \left[\left(\sum_{a=1}^{n_t^k}\frac{1}{R_a^k}\right)\left(1+\frac{(R_A^k)^{-1}}{\sum_{a=1}^{n_t^k}(R_a^k)^{-1}}\right)\right]^{-1} \\ &= \bar{U}_t^k\left[1+\frac{(R_A^k)^{-1}}{\sum_{a=1}^{n_t^k}(R_a^k)^{-1}}\right]^{-1} \approx \bar{U}_t^k\left[1-\frac{(R_a^k)^{-1}}{\sum_{a=1}^{n_t^k}(R_a^k)^{-1}}\right]. \end{aligned} \tag{10}$$

To obtain the final expression (10), in the last line we use the first order Taylor approximation $(1+x)^n \approx (1+nx)$ when $0 < x \ll 1$. This approximation is likely to hold as long as the number of users is large enough $n_t^k \gg 1$.

Assumption 2: To make the analysis simple, we assume that all the PHY rates R_a^k for all $a = 1, 2, \ldots n_t^k$ to a BS k are independent and identical distributed with a uniform distribution $R_a^k \sim \mathsf{U}(\alpha,\beta)$, where α and β denote the minimum and maximum PHY rates of all users.

Since each R_a^k is independent and identically distributed, they have the same expected value. Thus, the obtainable throughput if user A connects to another BS k, can be calculated as

$$\begin{aligned} \tilde{U}(k,\ell_t)|_{i_t=j\neq k} &\approx \bar{U}_t^k\left[1-\frac{\mathbf{E}\left\{\ (R_A^k)^{-1}\right\}}{\sum_{a=1}^{n_t^k}\mathbf{E}\left\{\ (R_a^k)^{-1}\right\}}\right] \\ &= \bar{U}_t^k\left(1-\frac{1}{n_t^k}\right). \end{aligned}$$

Proposition 1: The absolute error between the actual value $U(k,\ell_t)$ and the estimate $\tilde{U}(k,\ell_t)$ in our WiFi throughput estimation is bounded by

$$\frac{G}{n_t^k}\left(\frac{\beta}{\alpha}-1\right), \quad \textit{where } \beta \geq \alpha.$$

Proof: The absolute error is

$$\begin{aligned} &|U(k,\ell_t) - \tilde{U}(k,\ell_t)| \\ &= \bar{U}_t^k\left|\left(1-\frac{(R_A^k)^{-1}}{\sum_{a=1}^{n_t^k}(R_a^k)^{-1}}\right) - \left(1-\frac{1}{n_t^k}\right)\right| \\ &= \bar{U}_t^k\left|\frac{1}{n_t^k} - \frac{(R_A^k)^{-1}}{\sum_{a=1}^{n_t^k}(R_a^k)^{-1}}\right| \\ &\leq G\max\left\{\left|\frac{1}{n_t^k} - \frac{(1/\alpha)^{-1}}{n_t^k\ (1/\beta)^{-1}}\right|, \left|\frac{1}{n_t^k} - \frac{(1/\alpha)^{-1}}{n_t^k\ (1/\beta)^{-1}}\right|\right\} \\ &= \frac{G}{n_t^k}\max\left\{\left|1-\frac{\beta}{\alpha}\right|, \left|1-\frac{\alpha}{\beta}\right|\right\} = \frac{G}{n_t^k}\left(\frac{\beta}{\alpha}-1\right). \end{aligned}$$

■

Accordingly, we can conclude that the absolute error will be zero when $\beta = \alpha$, which assumes all users on BS k have the same PHY rates. Otherwise, if the number of users sharing the same BS n_t^k is large enough $n_t^k \gg G(\beta/\alpha - 1)$, the absolute error is also very close to zero.

Similarly, replacing $\bar{U}_t^k$ by the average of $\bar{U}_\tau^k$ over ν_t^k and using (8), $\tilde{U}(k,\ell_t)$ is then

$$\begin{aligned} \tilde{U}(k,\ell_t)|_{i_t=j\neq k} &= \frac{\sum_{\tau\leq t:i_\tau=k}\bar{U}_\tau^k}{\nu_t^k}\left(1-\frac{\nu_t^k}{\sum_{\tau\leq t:i_\tau=k} n_t^k}\right) \\ &= \frac{\sum_{\tau\leq t:i_\tau=k}\bar{U}_\tau^k(n_\tau^k-1)}{\sum_{\tau\leq t:i_\tau=k} n_\tau^k}. \end{aligned}$$

The BS observed regret measured at time t for class-2 RAT can be calculated as

$$\begin{aligned} Y_t(j,k) &= \frac{1}{t}\sum_{\tau\leq t:i_\tau=j}\left(\tilde{U}(k,\ell_\tau) - \bar{U}_\tau^j\right) \\ &= \frac{1}{t}\sum_{\tau\leq t:i_\tau=j}\left(\frac{\sum_{\tau\leq t:i_\tau=k}\bar{U}_\tau^k(n_\tau^k-1)}{\sum_{\tau\leq t:i_\tau=k} n_\tau^k} - \bar{U}_\tau^j\right). \end{aligned} \tag{11}$$

B. Reinforcement Learning With Network-Assisted Feedback (RLNF) Framework

We propose to fundamentally complement the Hart's RL-based algorithm in [17] with external feedback from the network to aid users in their RAT selection. Let Y_τ^k be the network feedback that the BS k sends to its connected user A at time τ. In this paper, the network feedback is a tuple $Y_\tau^k = (\bar{U}_\tau^k, n_\tau^k)$, where $\bar{U}_\tau^k$ is the BS computed per-user throughput at time τ and n_τ^k is the number of users on BS k at time τ. In our RLNF algorithm, the user then uses Y_τ^k to compute network measured regrets $Y_t(j,k)$ at time $t \geq \tau$.

Our main idea is to complement the user estimated regret B_t in [17] with the network observed assisted information Y_t at each time step t to speed up convergence towards the equilibria. We modify the probability of actions $p_{t+1}(k)$ in (6) with the combined regrets (B_t, Y_t) as in equation (12), shown at the bottom of this page, for any $0 < \epsilon \ll 1$. In order to implement this policy, each user needs 2 inputs: (1) the user observed throughputs $(U_1^k, \ldots, U_t^k)$ to compute user estimated regret $B_t(j,k)$ using equation (5); and (2) the network-assisted feedback $(Y_1^k, \ldots, Y_t^k)$ to compute network measured

regret $Y_t(j,k)$. The exact procedure to compute $Y_t(j,k)$ from network feedbacks was explained in Section V.B.

Our RLNF algorithm differs from the Hart's RL-based algorithm in [17] in the formula to update $p_{t+1}(k)$ in (12). Here, we make two changes to equation (6) in [17] for updating $p_{t+1}(k)$. First, $p_{t+1}(k)$ in (6) is a function of two inputs, i.e., $p_{t+1}(k) = f(\min\{B_t(j,k), m\})$, whereas in (12), we remove m in the min function and complement this function to take the extra information $Y_t(j,k)$ as $p_{t+1}(k) = f(\min\{B_t(j,k), Y_t(j,k)\})$. Thus, in our solution, not only the user observed regrets $B_t(j,k)$ but also the network measured regrets $Y_t(j,k)$ contribute to the update procedure of the user. Second, we do not use a constant proportionality factor μ as in (6), but normalise the vector of regret to get a probability vector. This is done to avoid needing to choose an appropriately large arbitrary parameter μ. As discussed in [17], a higher value of μ results in a smaller probability of switching and thus leads to a slower speed of convergence. It is not clear to us that the proof in [17] the convergence of Hart's RL-based procedure using (6) could be readily modified to include the form of normalisation we propose in (12).

There are three major terms in the formula (12). The first, $B_t^+(j,k)$, is the original regret as observed by the user in a manner similar to [17]. The second, $Y_t^+(j,k)$, is the extra "regret" observed at the BS. As the BS has a more complete view of the system than the individual users, $Y_t(j,k)$ is expected to take into account the information on network load conditions, which may not available under $B_t(j,k)$. Taking the minimum function of the two regrets guarantees that the sum $\sum_{k' \neq j} p_{t+1}^i(k')$ does not exceed 1. The last term, δ_t/m, is the weighted uniform distribution over I to guarantee that all probabilities at time $t+1$ are at least $\delta_t/m > 0$. This last term together with the scaling of the regrets by $(1-\delta_t)$ ensures that when t is small the algorithm explores different solutions to learn about the network environment. As the algorithm progresses, the regrets become the dominant factors in determining the selection probabilities.

Our algorithm for distributed RAT selection operated by each user is presented as follows.

Algorithm 1 Reinforcement Learning With Network-Assisted Feedback (RLNF)

1: *Exploration:* At the beginning, each user A takes sequential actions to explore all available choices $j \in S_A$ in order to learn possible payoffs and feedback from potential RATs.
2: *Initialisation:* Generate random uniform probability $p_1(j)$ for all $j \in S_A$.
3: **for** $t = 1, 2, \ldots$ **do**
4: *Action Selection:* Select action $i_t = j$ according to the probability distribution $p_t(j)$.
5: *Feedback Exchange:* Obtain feedback Y_t^j from the corresponding base station j.
6: *Regret Update:* for all $k \neq j \in S_A$
- Update the user estimated regret $B_t(j,k)$,
- Update the network measured regret $Y_t(j,k)$.

7: *Strategy Update:* Update $p_{t+1}(k)$ using (12).
8: **end for**

C. Convergence Properties

Theorem 1: If a player (i.e. player A) uses RLNF algorithm, its time average regret is guaranteed to approach the set of non-positive regrets almost surely irrespective of the behaviour of the other players, for finite payoffs and positive and finite feedback.

Proof: Please refer to Appendix for our proof which adopts the notation of [24]. ∎

Assumption 3: We assume that the payoffs are bounded and the network feedback $Y_\tau^k = (\bar{U}_\tau^k, n_\tau^k)$ is positive and finite for all τ, k. This assumption enables us to establish some convergence result for RLNF. In practice, all the payoffs and the feedback that we use (the number of users, the throughput) are finite and positive.

Theorem 2: If all players follow RLNF algorithm, the empirical distribution of joint play of all players $\bar{z}_t(s)$ converges almost surely as $t \to \infty$ to the set of correlated equilibria.

Proof: Please refer to Appendix. ∎

Remark 1: Contrary to most existing works that use the classical averaging theory for ordinary differential equations (ODEs) techniques to examines the convergence properties of their game algorithms [16], we use the differential inclusion (DI) framework in [24] to prove our Theorem. In our proof, if a single player uses the proposed procedure, its time average regret is guaranteed to approach its own set of non-positive regrets in the payoff space for any strategies of the other players. All players are required to follow the same algorithm in order to obtain the global convergence of the empirical distribution of joint actions of all players to the set of CE.

The following corollaries trivially follow from the proofs of Theorems 1 and 2 with small modifications for the construction of the probability vectors, and therefore omitted.

Corollary 1: Class-1 RAT selection games with the BS observed regret update in (9) *converges almost surely to the set of CE.*

Corollary 2: Class-2 RAT selection games with the BS observed regret update in (11) *converges almost surely to the set of CE.*

Remark 2: In the repeated game literature ([17], [23]), reference is made to "unconditional" or "internal" regrets.

$$p_{t+1}(k) = \begin{cases} (1-\delta_t)\min\left\{\dfrac{B_t^+(j,k)}{\epsilon + \sum_k B_t^+(j,k)}, \dfrac{Y_t^+(j,k)}{\epsilon + \sum_k Y_t^+(j,k)}\right\} + \dfrac{\delta_t}{m} & \text{if } k \neq j, \\ 1 - \sum_{k' \neq j} p_{t+1}(k') & \text{if } k = j. \end{cases} \quad (12)$$

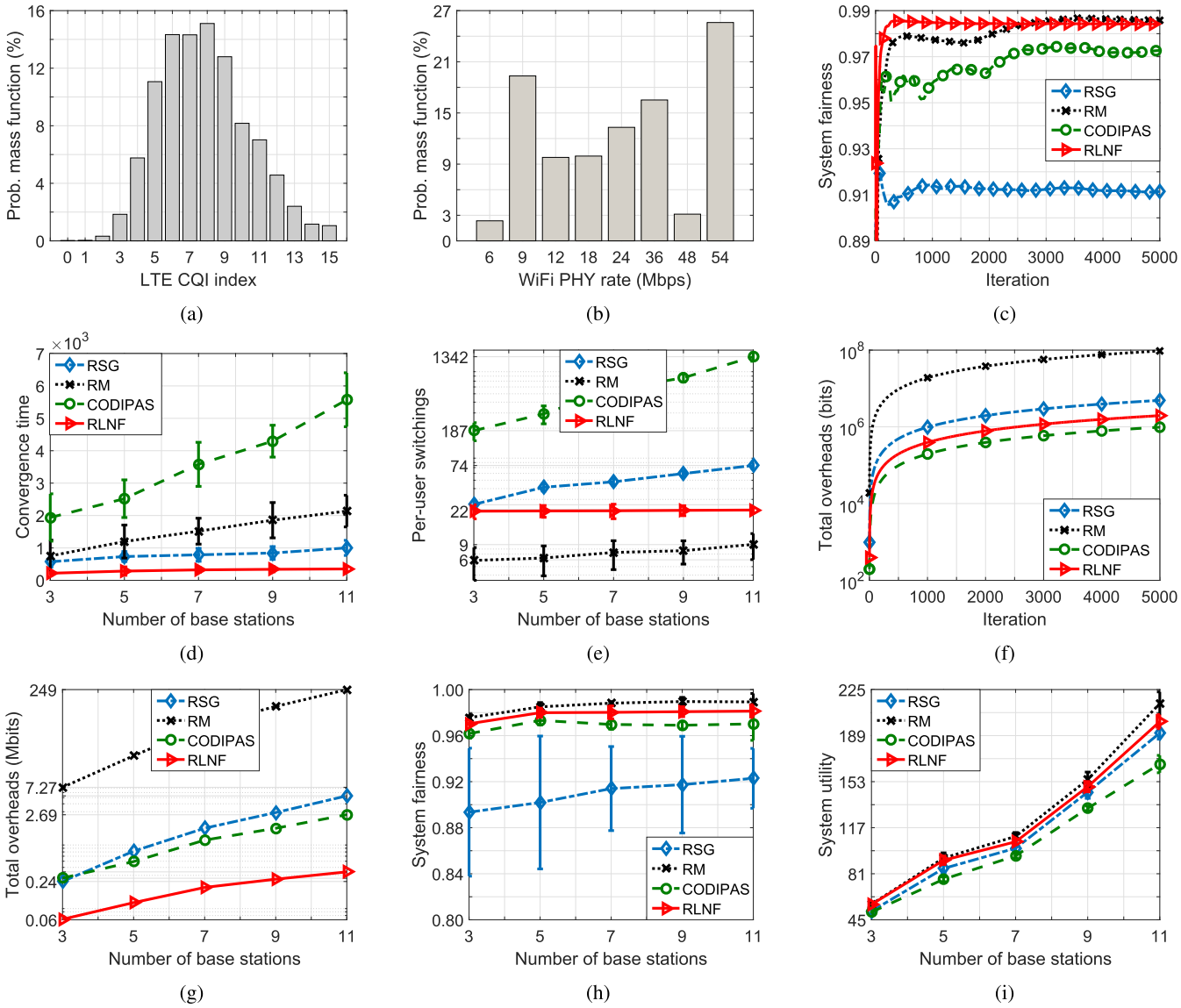


Fig. 2. (a) Example CQI distribution of a real-world LTE network; (b) PHY rate distribution of a real-world WiFi network; (c) Evolution of system fairness index J for different schemes; (d) Convergence time comparison with varying number of BSs; (e) Evolution of total overheads by different schemes; (f) Total overheads comparison with varying number of BSs; (g) Per-user switchings comparison with varying number of BSs; (h) System fairness J for varying number of BSs; (i) System utility comparison with varying number of BSs.

Rather than the case considered here ("conditional" or "external" regrets), the notion of unconditional regret involves a player reasoning about replacing each action played by a fixed strategy. In [17], it's shown that an RL procedure based on unconditional regrets converges to the *Hannan set* of global non-positive regrets if all players play that strategy. The Hannan set contains CE. Furthermore, it's argued in [24] that, in a heterogeneous system where some players may adopt different strategies, those players that use the unconditional regret based algorithm will themselves achieve non-positive unconditional regret. These approaches can be handled within the framework presented in this paper with appropriate modifications, although we don't address this (somewhat simpler) issue here.

V. Evaluation

We consider a heterogeneous wireless network environment with 2 different RATs (WiFi and LTE) in a narrow square area of 150 × 150 meters. We assume that WiFi BSs and users are located within the coverage area of one macro LTE BS at the center of the network. We follow the same network model in [5], that reflect real world WiFi BSs and users distribution. In this model, the connectivity and bandwidth between BSs and users are determined by their geographical distribution. We divide the given geographic area into 9 smaller, non-overlapping square-shaped areas and randomly place a WiFi BS within the borders of each small area. We then place a random number of users (up to 20 – the maximum number of local users for each WiFi BS) for each WiFi BS within the area. A user is considered to be a local user to BSs that are located in the same area of its location and to be a non-local users to the rest of the BSs in the network. We assume that each WiFi BS allocate a certain portion of its bandwidth ($0 \leq \kappa \leq 1$) to serve the non-local users ($\kappa = 1$ for the local users). The actual throughput of users A under

TABLE III
PHY RATE AND THE RSS FOR IEEE 802.11G [33]

PHY (Mbps)	6	9	12	18	24	36	48	54
RSS (dBm)	-90	-84	-82	-80	-77	-73	-72	>-72

the non-local BS k is equal to $\kappa \times \bar{U}_A^k$, where $\bar{U}_A^k$ is given in (2).

We use real network data from a tier-1 LTE operator in North America to simulate users' PHY rates to the macro LTE BS. In particular, we use the measured Channel Quality Indication (CQI) and map them to the possible data rates that a user can receive from a BS [31]. Fig. 2a shows an example CQI distribution of the real-world LTE BS from the dataset. In the simulation, maximum data rate of 35 Mbps per cell in LTE 20 MHz is assumed [31]. We then linearly divide the data rate into 16 different levels corresponding to the 16 CQI indexes. For example, the strongest CQI 15 correspond to the highest data rate of 35 Mbps and the median CQI 7 corresponds to 17.5 Mbps. A user's PHY rate to a BS is supposed to be unchanged over time.

In addition to LTE data, we also use the collected residential WiFi data [32] in setting up users' PHY rates to WiFi BSs. This data set provides traces of received signal strength (RSS) measurements of the WiFi BSs collected at the University of Colorado. These values are then converted to PHY rates based on Table III as follow. Fig. 2b shows an example of PHY rate distribution of the WiFi network from the simulated dataset.

To evaluate the performance of our proposed RLNF algorithm, we compare the performance of the following four distributed algorithms for RAT selection:

- RAT Selection Games (RSG) in [8]–[10]: All BSs broadcast their traffic information to all users. Thus, each user has the information on the number of other users on each BS and their PHY rates. At each iteration, user selects a BS that provides the highest throughput. This broadcasting assumption is similar to those in [11]–[13].
- Regret Matching (RM) in [14]: Users are assumed to have a global view of the network including the actions taken by other users and their historical PHY rates. Users apply the regret matching algorithm [23] to select their RATs.
- Combined Fully Distributed Payoff and Strategy Reinforcement Learning (CODIPAS) in [16]: Users learn and adapt their decisions based on their own observation of the rewards received from past experiences. At each iteration, using only this information, user selects the best available BS to maximize its utility. This is a state-of-the-art RL-based algorithm and has been shown to be superior to the Hart's RL-based scheme in [17].
- *Our Reinforcement Learning with Network-Assisted Feedback (RLNF)*: User data is not required to exchange among the users or the BSs. Each BS shares feedback only to its connecting users to assist them in their RAT selection decisions.

For comparison purposes, we use the following metrics:

- Total overheads (bits): amount of data exchanges between users and BSs. Lower overhead is preferable.
- Convergence time (iterations): required number of iterations to convergence. A fast convergence is desired since the wireless channel conditions may change quickly.
- Per-user switchings: maximum number of switchings required by all users to convergence. A small number of switching is desirable to minimise the cost for managing the vertical switching between RATs.
- Jain's fairness index, which is derived as

$$J = \frac{(\sum_{a=1}^{N} x_a)^2}{N \times \sum_{a=1}^{N} x_a^2},$$

  where x_a is the average throughput of user a and N is the number of users. Notes that the largest value 1 indicating the best fairness of the system, which guaranteeing the same throughput among all the users.
- System utility: sum of all users' average throughputs. Higher utilities benefit both mobile operators and service providers in offering higher bandwidth-services.

We would like to emphasise that users running our RLNF algorithm select their RAT by combining their individual observed throughput and the network feedback; whereas in all the other solutions, users make their RAT selection decisions based only on their own observations. For each network model and algorithm, the actual throughput $\bar{U}_A^k$ that a user A gets from the BS k depends on the other users that share the same BS, and is given in the equations (1) and (2). The instantaneous throughput U_A^k that a user A observes directly from its connecting BS k is a random number generated according to the Gaussian distribution $\mathsf{N}(\bar{U}_A^k, \sigma^2)$ with $\bar{U}_A^k$ mean and $e \times \bar{U}_A^k$ proportional standard deviation, where we assume the proportional noise factor is $e = 0.3$. This model and choice of parameters was used in [10].

We also set $\delta = 10^{-5} \ll 1$ and $\gamma = 0.1$ for all the simulations of RLNF algorithm. Note that large δ may cause the convergence to a large distance from the CE set. To compare the performance of different schemes versus the number of BSs, we fix the number of LTE BSs to 1 BS and vary the number of WiFi BSs from 2 BSs to 10 BSs. Thus the total BS number varies from 3 BSs to 11 BSs. All the results presented are averaged over 50 simulation runs. Each data point on the graphs is the average value shown with the standard deviation as an error bar.

A. Performance Comparison With Existing Algorithms

We first compare our RLNF against existing algorithms in convergence behaviour. The feedback used in RLNF is the BS computed throughput and the number of connected users. In our simulation, the BS computes the actual throughput for each user using equations (1) or (2) and provide them this number. Each BS also keeps track of the number of users currently connected to it and sends this information to its serving users.

Figs. 2c, 2d and 2e show, respectively, the evolution of system fairness index, the convergence time versus number of BSs and the number of RAT switchings for each user (per-user switching) versus number of BSs by different algorithms.

We observe that RLNF achieves the fastest convergence with a small number of per-user switchings among all algorithms. Our RLNF even outperforms the RM in convergence speed. It should be note that in standard RM in [17], payoffs of the players (mobile users) are not noisy. But in our simulation, in order to reflect practical consideration of real-world network for RAT selection, users running our RLNF receive the actual throughputs via network-assisted feedback; whereas in all the other schemes including RM, users only observe their noisy payoffs from their instantaneous throughputs. The network feedback in RLNF is therefore more accurate than the user observed throughput in RM and hence user running RM may take a longer time to learn the throughput in order to converge. Although RM obtains a smaller number of per-user switchings than RLNF, it requires a longer time to converge and exchanges significant communication overheads as we explain later in Figs. 2f and 2g. CODIPAS performs poorest in both convergence speed and per-user switchings metrics due to the lack of information on global network conditions.

We present, respectively, in Figs. 2f and 2g the total information exchange versus number of iterations and number of BSs across different algorithms in order to compare their overheads. We assume that 4 bits are used to represent the number of users or throughput. Let T be the number of iterations to convergence. The calculations of the information exchanges for each algorithm are summarised below.

- RSG: Each user obtains its payoff from its serving BS ($4 \times N$ bits). Each user also needs to receive the number of connecting users on Class-1 BSs and per-user throughput on Class-2 BSs ($4 \times N \times (M-1)$ bits) in order to calculate its expected throughputs of joining the other $(M-1)$ BSs. The total overheads are thus $4NM \times$ convergence time (bits) $\sim O(TNM)$.
- RM: Each user obtains its payoff from its serving BS ($4 \times N$ bits). Each user also needs to know the PHY rates and actions taken by other $(N-1)$ users in each iteration ($8N(N-1)$ bits). The total overheads are thus $(8N^2 - 4N) \times$ convergence time (bits) $\sim O(TN^2)$.
- CODIPAS: Each user receives its payoff directly from its serving BS without requiring any extra communication. The total overheads are just $4N \times$ convergence time (bits) $\sim O(TN)$.
- RLNF: Apart from the BS computed per-user throughput (4 bits), user also requires the number of users sharing the same BS (4 bits) from its connecting BS. The total overheads are then $8N \times$ convergence time (bits) $\sim O(TN)$.

Fig. 2f shows that with the same number of iterations CODIPAS has the lowest overhead performance. However, as shown in Fig. 2g, RLNF is the best algorithm to minimise overheads. CODIPAS, despite using less information to make decisions, requires higher overheads due to its slower convergence speed, i.e., larger T. Both require an order of magnitude less information exchange than RSG and RM algorithms, especially the later, and when the number of users is large. The reason is that their complexity is linear whereas the complexity of RM is quadratic and the complexity of RSG depends also on the total number of BSs in the network.

We now compare the performances of the algorithms on system fairness and system utility. The fairness and utility results are shown in Figs. 2h and 2i, respectively. As shown, RLNF achieves comparable fairness to RM. Both are better than the others in fairness metric. We also observe that RM, RLNF and CODIPAS achieve very good fairness indexes in all the cases compared to RSG. This can be explained by the fact that these three algorithms are designed to reach an efficient equilibrium points such as CE (RM and RLNF) or optimal-NE (CODIPAS) rather than converging to arbitrary NE as in RSG. Similarly, RLNF achieves very similar utility to the RM, and outperforms the other remaining algorithms.

For further evaluation of the scalability of the algorithms, we study the impact of network size (total number of users in the network) on the performances of different algorithms. We fix the total number of BS in the network to 5 BSs (composed of 1 LTE BS and 4 WiFi BSs). We then vary the number of local users per WiFi BS from 10 users/BS to 50 users/BS resulting in increasing the total number of users in the network from 40 users to 200 users. Fig. 3 shows the scalability behaviour of the algorithms with respect to the size of the network. Further experiments presented in Fig. 3 demonstrate the robust performance and stability of RLNF to variations in the network size as compared to relevant RAT selection schemes. Overall, RLNF achieves the fastest speed and lowest overheads, whilst guaranteeing competitive performance both in fairness and utility, as well as requiring a small number of per-user switchings as compared to the others.

B. Using Feedback to Change Convergence Points

One of the main difference between our RLNF and other game algorithms [8]–[14] is that our solution can flexibly support a wide range of policy-defined feedback. Under our framework, network operators can influence user decisions to achieve their objectives by tuning their network feedback information. In the following, we show how to apply different feedback in RLNF to achieve different convergence points.

As explained in Section IV, once the BS has computed per-user throughput, it can send the users this information to aid them in their RAT selections. This network-assisted information, however, does not need to be the actual value of per-user throughput, but can be functions of these throughputs. This type of feedback reflects information about the expected payoff that a user could receive from a BS. Let $\hat{U}_t^k = f(\bar{U}_t^k)$ be the feedback that the BS k sends to its connected users at time t. For simplicity, $\hat{U}_t^k$ can be defined as a function of $\bar{U}_t^k$ as below

$$\hat{U}_t^k = \begin{cases} \bar{U}_t^k\,(1+\gamma) & \text{if } R_a^k \geq \omega_1^k, \\ \bar{U}_t^k & \text{if } \omega_2^k < R_a^k < \omega_1^k, \\ \bar{U}_t^k\,(1-\gamma) & \text{if } R_a^k \leq \omega_2^k, \end{cases} \tag{13}$$

where $0 \leq \gamma \leq 1$ is some weighted parameter and $[\omega_1^k, \omega_2^k]$ are PHY rate thresholds defined by the network operator. Each network could use different γ and $[\omega_1^k, \omega_2^k]$ depends on its own policy. Note that the feedback $\hat{U}_t^k$ is equal to real actual throughput $\bar{U}_t^k$ when $\gamma = 0$.

The idea is that network feedback is tuned as a function of the user PHY rate. When user PHY rate on a BS k is

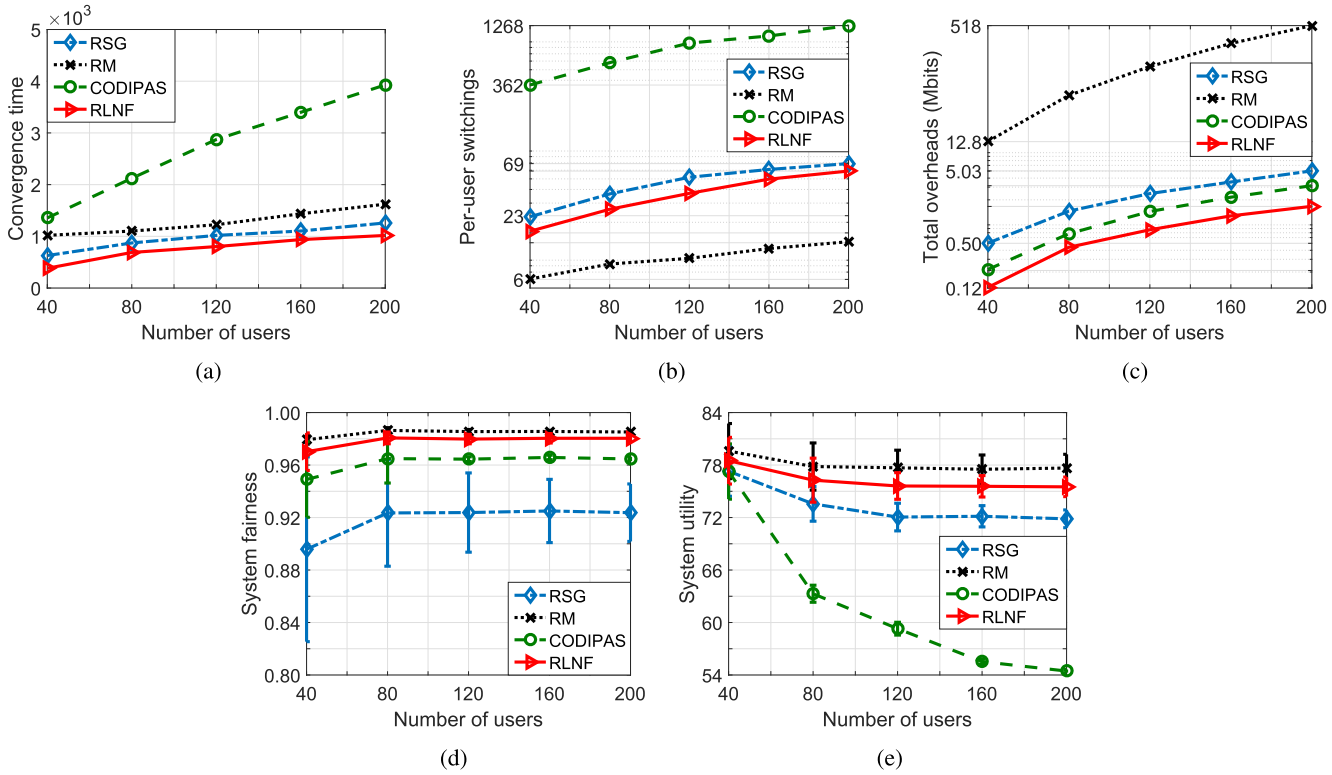


Fig. 3. Performance comparison of the different algorithms for increasing size of the network (number of users) on: (a) Convergence time; (b) Per-user switchings; (c) Total overheads; (d) System fairness; and (e) System utility.

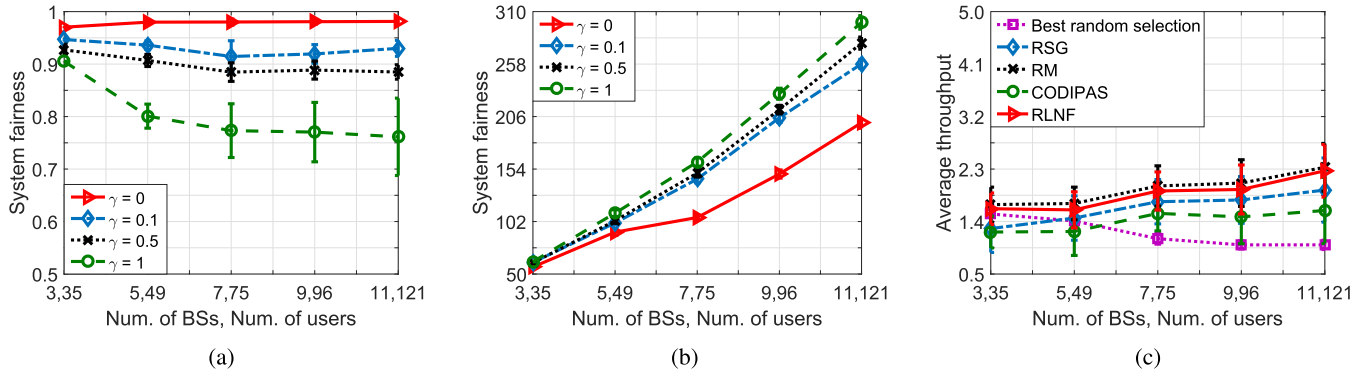


Fig. 4. (a) Impact of different feedback mechanism on system fairness; (b) Impact of different feedback mechanism on system utility; (c) Average throughput performance by different schemes in a heterogeneous situation where users use different learning strategies.

higher than a threshold $R_a^k \geq \omega_1^k$, the network encourages that user to select BS k by putting more weight on the feedback throughput. In contrast, when user PHY rate is lower than the threshold $R_a^k \leq \omega_2^k$, the network discourages that user from selecting the BS by reducing the feedback throughput.

In this experiment, instead of using feedback in term of the real actual throughput, we vary the weighted parameter γ according to the feedback form as in equation (13); and measure the performance of RLNF in terms of fairness and utility. We also set the PHY rate thresholds $[\omega_1^k, \omega_2^k]$ of WiFi and LTE BSs to be [36Mbps, 12Mbps] and [24Mbps, 10Mbps], respectively. Figs. 4a and 4b illustrate the impact of different feedback mechanisms on fairness and utility.

As shown, increasing γ improves the total utility, however reduces the system fairness. The reason behind this observation is that increasing γ will encourage users to select the BSs that offer the higher PHY rates, which results in providing them with better throughputs. Therefore, the total utility increases. At the same time, this higher utility comes with a cost of increasing disparities of users' throughputs, which also results in bringing down the system fairness. Obviously, there is a trade-off between fairness and utility. Depending on different policies, different feedback mechanisms could be defined to meet the operator's goals.

C. Performance of RLNF in Heterogeneous Environment

Lastly, we investigate the case where users do not use the same learning rule. Particularly, we simulate a network in a complex heterogeneous situation where half of the users play a random fixed strategy and the others play a random strategy at each iteration, except only one user using an adaptive

game algorithm. We randomly select one user among all the users and let that user applies our proposed RLNF. We then repeat the same simulation with different algorithms including RSG, RM and CODIPAS, respectively, for performance comparison purposes. The comparison of average throughput of the selected user running different learning algorithms is illustrated in Fig. 4c.

As shown, RLNF achieves very close performance to RM scheme and outperforms the others. Note that RLNF does not use global information of the network (how many players are, their actions and payoffs) as required in RM. RLNF achieves faster convergence and exchange significant less overheads, especially for a network with large number of BSs. We observed that the average throughput of users running a random fixed strategy heavily depends on the BSs they select as well as their PHY rates on these BSs. User runs random selection strategy at each iteration also obtains a very poor throughput when the number of BSs is large. The long-run payoffs of the RLNF user, however, does not depend on either its selected BSs or its PHY rates on any BSs as well as the number of BSs in the network. The result implies that such a user has no regret nor does it lose by committing to use RLNF rather than playing any other strategies. This demonstrates the efficiency of using RLNF in real networks where each user often plays different RAT selection strategy according to its own preference.

VI. Conclusion

We have studied the problem of RAT selection games in heterogeneous wireless networks. We have developed a new decentralised framework, called Reinforcement Learning with Network-Assisted Feedback (RLNF), that incorporates limited base station measurements in a user's RAT selection policy to achieve fast convergence to the set of correlated equilibria. Our RLNF, as compared to other algorithms, achieves faster convergence rate, lower signalling overheads with a small number of RAT switching per-user, whilst achieving competitive performance both in global network utility and user fairness. More importantly, by adopting an efficient feedback mechanism, RLNF enables mobile users to adapt their selection behaviours to various network feedback, resulting in behaviour that meets operator objectives while providing users with good performance. Lastly, we show that our solution guarantees non-positive regret in the long-run for any user applying RLNF, regardless of what other users might do and so can work in an environment where other users may not use RLNF. This is an important implementation issue as RLNF can be implemented within current standards. We have demonstrated the improved performance of RLNF compared to other related algorithms using realistic simulations.

Appendix A
Proof of Theorem 1

Let $C : Z \to \mathbb{R}^{m\times m}$ be defined by

$$[C(z)]_{j,k} = \sum\nolimits_{\ell\in\mathcal{L}} z(j,\ell)\left(U(k,\ell) - U(j,\ell)\right),$$

which is the expected regret for player A when substituting action k for action j under the joint distribution z of actions. Suppose we consider player A playing some action $i = j$ with probability one, then

$$\begin{aligned}[C(z^i)]_{j,k} &= \sum\nolimits_{\ell\in\mathcal{L}} \mathbb{1}_{\{i=j\}} y_\ell \left(U(k,\ell) - U(j,\ell)\right)\\ &= \mathbb{1}_{\{i=j\}}\left(U(k,y) - U(j,y)\right).\end{aligned}$$

Since player A cannot compute the first term as it only has access to the payoffs corresponding to actions it actually took, following [17], define an estimate of this term by

$$\tilde{U}(k,y)\mathbb{1}_{\{i=j\}} = \frac{p(j)}{p(k)} U(k,y)\mathbb{1}_{\{i=k\}},$$

which is computed from the regrets associated with the alternative action k weighted proportional to the relative probabilities of player A choosing action j versus k when those actions were actually taken. The associated pseudo regret matrix at stage t is now

$$B_t(j,k) = \frac{p_t(j)}{p_t(k)} U(k,y_t)\mathbb{1}_{\{i_t=k\}} - U(j,y_t)\mathbb{1}_{\{i_t=j\}}.$$

Thus, we have

$$\begin{aligned}\mathbf{E}\left\{B_t(j,k)|h_{t-1}\right\} &= p_t(k)\frac{p_t(j)}{p_t(k)}U(k,y_t) - p_t(j)U(j,y_t)\\ &= p_t(j)\left(U(k,y_t) - U(j,y_t)\right)\\ &= \mathbf{E}\left\{C_t(j,k)|h_{t-1}\right\},\end{aligned}$$

where h_{t-1} is the action history of the game until stage $t-1$. It can be seen that $B_t(j,k)$ and $C_t(j,k)$ are each bounded by $2mG/\delta_t$. The limit sets of the pair processes B_t and C_t also coincide since they both have the same conditional expected values (see [17] for more details and discussion). Then Theorem 7.3 of [24] can be applied and thus the two processes exhibit the same asymptotic behaviour.

Let $\bar{B}_t(j,k) = \sum_{\tau=1}^{t} B_\tau(j,k)$ be the time-average of $B_t(j,k)$. The average regret at stage t is thus a matrix B_t defined by

$$B_t(j,k) = \frac{1}{t}\sum_{\tau=1}^{t}\left[\frac{p_\tau(j)}{p_\tau(k)}U(k,y_\tau)\mathbb{1}_{\{i_\tau=k\}} - U(j,y_\tau)\mathbb{1}_{\{i_\tau=j\}}\right]$$

We then have the algebraic identity

$$\bar{B}_{t+1} - \bar{B}_t = \frac{1}{t+1}\left(B_{t+1} - \bar{B}_t\right)$$

holds. This result follows directly from the definition of average $\bar{B}_t$. Hence, the above discrete dynamics is a discrete stochastic approximation of the DI

$$\dot{\mathbf{w}} \in \hat{N}(\mathbf{w}) - \mathbf{w} \quad (\text{with } w = B_t), \tag{14}$$

where $\hat{N}$ is a mapping from $\mathbb{R}^m$ into the class of all subsets of $\mathbb{R}^m$ (called a *correspondence* on $\mathbb{R}^m$) that satisfies the various conditions outlined in Hypothesis 2.1 of [24] (see [24] for details).

Now define the matrix sequence

$$[M_t]_{j,k} = \min\left\{\frac{B_t^+(j,k)}{\epsilon + \sum_k B_t^+(j,k)}, \frac{Y_t^+(j,k)}{\epsilon + \sum_k Y_t^+(j,k)}\right\} \tag{15}$$

for $j \neq k$. We set $[M_t]_{j,j} = 1 - \sum_{k \neq j} [M_t]_{j,k}$ which is in $[0, 1]$ by Assumption 3 and virtue of (15). Thus M_t is a transition probability matrix on $\mathcal{S}$. So there is a probability vector μ_t such that $M_t^T \mu_t = \mu_t$.

The "no positive regret set" $D \subset \mathbb{R}^{m \times m}$ for player A is defined by

$$D = \left\{ g \in \mathbb{C}^{m \times m} : g(j,k) \leq 0, \forall\ (j,k) \right\}.$$

Evidently, D is a closed, convex subspace of $\mathbb{R}^{m \times m}$. Define the Lyapunov function $P(w) = \frac{1}{2}\|w^+\|^2$, with $\nabla P(w) = w^+ \geq 0$. Then P satisfies the following properties:

(i) P is continuously differentiable;
(ii) $P(w) = 0 \Leftrightarrow w \in D$;
(iii) $[\nabla P(w)]_i \geq 0$ for all $i = 1, \ldots, m$;
(iv) $\langle \nabla P(w), w \rangle > 0$ for all $w \notin D$.

Thus P is a potential function for D. Let $\Pi_D(w)$ be the convex projection onto D, then we have $w^+ = w - \Pi_D(w)$, and $\langle w^+, \Pi_D(w) \rangle = 0$. Let $\varphi : \mathbb{R}^{m \times m} \to 2^X$ given by

$$\varphi(w) = \begin{cases} (1 - \delta_n)\mu(w) + \dfrac{\delta_n}{m} & \text{if } w \notin D^1, \\ X & \text{if } w \in D^1, \end{cases} \tag{16}$$

where $\mu(w)$ denotes a probability vector computed from w^+ according to the process above.

Define a correspondence $\hat{N}$ on $\mathbb{R}^{m \times m} \setminus D$ by $\hat{N}(w) = C(\varphi(w) \times Y)$ so that $\hat{N}(w)$ contains all resulting average regrets. According to Lyapunov theory, to prove the approachability of w to D, it suffices to show that for any $w \in \mathbb{R}^{m \times m} \setminus D$ and some constant $\lambda > 0$,

$$\frac{d}{dt} P(w) = \langle \nabla P(w), \dot{w} \rangle \in \langle \nabla P(w), N(w) - w \rangle \leq -\lambda P(w),$$

meaning $\langle \nabla P(w), \theta - w \rangle \leq -\lambda P(w)$ for all $\theta \in \hat{N}(w)$ (see [24] for details).

Suppose that $w \notin D$, let $\theta = \mathbf{E}\left\{ \tilde{C}(\varphi(w), y) | h_{n-1} \right\}$, with $y \in Y$, which means

$$[\theta]_{j,k} = \varphi_j(w)\left(U(k,y) - U(j,y)\right).$$

Then consider

$$\begin{aligned} &\langle \nabla P(w), \theta \rangle \\ &\quad = \sum\nolimits_{j,k}^{m} \nabla P_{jk}(w) \varphi_j(w) \left(U(k,y) - U(j,y)\right) \\ &\quad = (1 - \delta_t) \sum\nolimits_{j,k} \nabla P_{jk}(w) \mu_j(w) \left(U(k,y) - U(j,y)\right) \\ &\qquad + \frac{\delta_t}{m} \sum\nolimits_{j,k} \nabla P_{jk}(w) \left(U(k,y) - U(j,y)\right) \\ &\quad = (1 - \delta_t) \sum\nolimits_{j} U(j,y) \\ &\qquad \times \left(\sum\nolimits_{k} \mu_k(w) \nabla P_{kj}(w) - \mu_j(w) \sum\nolimits_{k} \nabla P_{jk}(w) \right) \\ &\qquad + \frac{\delta_t}{m} \sum\nolimits_{j,k} \nabla P_{jk}(w) \left(U(k,y) - U(j,y)\right). \end{aligned} \tag{17}$$

In the third line we substituted for $\varphi_j(w)$ from (16), and in the last line we collected together all terms containing $U(j,y)$.

Let $\mu_t = \mu(w)$ be such a measure. Suppose that for every $j = 1, \ldots, m$, it holds that

$$\mu_j(w) \sum\nolimits_{k} \nabla P_{jk}(w) = \sum\nolimits_{k} \mu_k(w) \nabla P_{kj}(w),$$

then the first term in (17) is equal to zero. Noting that the payoff function $|U(.)|$ is bounded by G using Assumption 3, then

$$\begin{aligned} \langle \nabla P(w), \theta \rangle &= \frac{\delta_t}{m} \sum\nolimits_{j,k} \nabla P_{jk}(w) \left(U(k,y) - U(j,y)\right) \\ &\leq ||\nabla P(w)|| \frac{2G\delta_t}{m}. \end{aligned} \tag{18}$$

Next, consider

$$\begin{aligned} \langle \nabla P(w), w \rangle &= \left\langle w^+, w \right\rangle = \left\langle w^+, w^+ + \Pi_D(w) \right\rangle = ||w^+||^2 \\ &= 2\ P(w) \quad (\text{since } \left\langle w^+, \Pi_D(w) \right\rangle = 0). \end{aligned} \tag{19}$$

It follows that given $\epsilon > 0$, $||w^+|| \geq \epsilon$, one can choose $\delta_n > 0$ small enough such that

$$\begin{aligned} \langle \nabla P(w), \theta - w \rangle &= \langle \nabla P(w), \theta \rangle - \langle \nabla P(w), w \rangle \\ &\leq ||\nabla P(w)|| \frac{2G\delta_t}{m} - 2\ P(w) \leq -P(w). \end{aligned}$$

Thus from Lyapunov theory, the set D is a global attractor for the DI (14). Hence, the regret B_t and its corresponding conditional regret C_t will then approach D. Note that in our proof, Theorem 1 holds no matter what the other players do as long as all the payoffs are bounded. In other words, any user applying our RLNF will achieve "self-consistency" [17] (all its positive regrets approach zero in the long run). This completes the proof.

APPENDIX B
PROOF OF THEOREM 2

The proof follows immediately from how the "regret" measure is defined. Recall that

$$\begin{aligned} [C(z_t)]_{j,k} &= \sum\nolimits_{\ell \in \mathcal{L}} z_t(j, \ell_t) \left(U(k, \ell_t) - U(j, \ell_t)\right) \\ &= \sum\nolimits_{s_t \in S : i_t = j} z_t(s_t) \left(U(k, \ell_t) - U(j, \ell_t)\right), \end{aligned}$$

where $s_t = (i_t, \ell_t)$ is the joint action at t. On any convergent subsequence $\lim_{t \to \infty} z_t \to \Pi$, we get

$$\lim_{t \to \infty} [C(z_t)]_{j,k} = \sum_{s_t \in S : i_t = j} \Pi(s_t) \left(U(k, \ell_t) - U(j, \ell_t)\right) \leq 0.$$

Next, comparing with the definition of CE as in equation (3), the desired results follows.

REFERENCES

[1] Q. Chen, G. Yu, H. Shan, A. Maaref, G. Y. Li, and A. Huang, "Cellular meets WiFi: Traffic offloading or resource sharing?" *IEEE Trans. Wireless Commun.*, vol. 15, no. 5, pp. 3354–3367, May 2016.
[2] J. G. Andrews *et al.*, "What will 5G be?" *IEEE J. Sel. Areas Commun.*, vol. 32, no. 6, pp. 1065–1082, Jun. 2014.
[3] S. H. Chae, J.-P. Hong, and W. Choi, "Optimal access in OFDMA multi-RAT cellular networks: Can a single RAT be better?" *IEEE Trans. Wireless Commun.*, vol. 15, no. 7, pp. 4778–4789, Jul. 2016.

[4] A. Ahmed, L. M. Boulahia, and D. Gaiti, "Enabling vertical handover decisions in heterogeneous wireless networks: A state-of-the-art and a classification," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 2, pp. 776–811, Jun. 2014.
[5] O. B. Karimi, J. Liu, and J. Rexford, "Optimal collaborative access point association in wireless networks," in *Proc. IEEE INFOCOM*, Apr. 2014, pp. 1141–1149.
[6] S. Singh and J. G. Andrews, "Joint resource partitioning and offloading in heterogeneous cellular networks," *IEEE Trans. Wireless Commun.*, vol. 13, no. 2, pp. 888–901, Feb. 2014.
[7] V. Sagar, R. Chandramouli, and K. P. Subbalakshmi, "Software defined access for HetNets," *IEEE Commun. Mag.*, vol. 54, no. 1, pp. 84–89, Jan. 2016.
[8] E. Aryafar, A. Keshavarz-Haddad, M. Wang, and M. Chiang, "RAT selection games in HetNets," in *Proc. IEEE INFOCOM*, Apr. 2013, pp. 998–1006.
[9] E. Monsef, A. Keshavarz-Haddad, E. Aryafar, J. Saniie, and M. Chiang, "Convergence properties of general network selection games," in *Proc. IEEE INFOCOM*, Apr. 2015, pp. 1445–1453.
[10] A. Keshavarz-Haddad, E. Aryafar, M. Wang, and M. Chiang, "HetNets selection by clients: Convergence, efficiency, and practicality," *IEEE/ACM Trans. Netw.*, vol. 25, no. 1, pp. 406–419, Feb. 2017.
[11] Z. Du, Q. Wu, P. Yang, Y. Xu, J. Wang, and Y.-D. Yao, "Exploiting user demand diversity in heterogeneous wireless networks," *IEEE Trans. Wireless Commun.*, vol. 14, no. 8, pp. 4142–4155, Aug. 2015.
[12] S. Andreev *et al.*, "Intelligent access network selection in converged multi-radio heterogeneous networks," *IEEE Wireless Commun.*, vol. 21, no. 6, pp. 86–96, Dec. 2014.
[13] M. E. Helou, M. Ibrahim, S. Lahoud, K. Khawam, D. Mezher, and B. Cousin, "A network-assisted approach for RAT selection in heterogeneous cellular networks," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 6, pp. 1055–1067, Jun. 2015.
[14] L. Chen, "A distributed access point selection algorithm based on no-regret learning for wireless access networks," in *Proc. IEEE 71st Veh. Technol. Conf.*, May 2010, pp. 1–5.
[15] D. Niyato and E. Hossain, "Dynamics of network selection in heterogeneous wireless networks: An evolutionary game approach," *IEEE Trans. Veh. Technol.*, vol. 58, no. 4, pp. 2008–2017, May 2009.
[16] H. Tembine, "Fully distributed learning for global optima," in *Distributed Strategic Learning for Wireless Engineers*. Boca Raton, FL, USA: CRC Press, Apr. 2012, pp. 317–359.
[17] S. Hart and A. Mas-Colell, "A reinforcement procedure leading to correlated equilibrium," in *Economics Essays*. Berlin, Germany: Springer, 2001, pp. 181–200.
[18] H. P. Borowski, J. R. Marden, and J. S. Shamma, "Learning efficient correlated equilibria," in *Proc. IEEE 53rd Annu. Conf. Decision Control (CDC)*, Dec. 2014, pp. 6836–6841.
[19] D. D. Nguyen, H. X. Nguyen, and L. B. White, "Performance of adaptive RAT selection algorithms in 5G heterogeneous wireless networks," in *Proc. IEEE 26th Int. Telecommun. Netw. Appl. Conf. (ITNAC)*, Dec. 2016, pp. 70–75.
[20] R. Trestian, O. Ormond, and G.-M. Muntean, "Game theory-based network selection: Solutions and challenges," *IEEE Commun. Surveys Tuts.*, vol. 14, no. 4, pp. 1212–1231, 2012.
[21] C. Daskalakis, R. Frongillo, C. H. Papadimitriou, G. Pierrakos, and G. Valiant, "On learning algorithms for Nash equilibria," in *Algorithmic Game Theory*. Berlin, Germany: Springer, 2010, pp. 114–125.
[22] R. J. Aumann, "Correlated equilibrium as an expression of Bayesian rationality," *Econ., J. Econ. Soc.*, vol. 55, no. 1, pp. 1–18, Jan. 1987.
[23] S. Hart and A. Mas-Colell, "A simple adaptive procedure leading to correlated equilibrium," *Econometrica*, vol. 68, no. 5, pp. 1127–1150, Sep. 2000.
[24] M. Benaïm, J. Hofbauer, and S. Sorin, "Stochastic approximations and differential inclusions, part II: Applications," *Math. Oper. Res.*, vol. 31, no. 4, pp. 673–695, Nov. 2006.
[25] V. Krishnamurthy, O. Gharehshiran, and M. Hamdi, "Interactive sensing and decision making in social networks," *Found. Trends Signal Process.*, vol. 7, nos. 1–2, pp. 1–196, 2014.
[26] M. Bravo and M. Faure, "Reinforcement learning with restrictions on the action set," *SIAM J. Control Optim.*, vol. 53, no. 1, pp. 287–312, Jan. 2015.
[27] D. D. Nguyen, L. B. White, and H. X. Nguyen, *Adaptive Multiagent Reinforcement Learning with Non-positive Regret*. Cham, Switzerland: Springer, Dec. 2016, pp. 29–41.
[28] S. Hart and D. Schmeidler, "Existence of correlated equilibria," *Math. Oper. Res.*, vol. 14, no. 1, pp. 18–25, Feb. 1989.
[29] *IEEE Standard for Architectural Building Blocks Enabling Network-Device Distributed Decision Making for Optimized Radio Resource Usage in Heterogeneous Wireless Access Networks*, IEEE Standard 1900.4, Feb. 2009, pp. 1–130.
[30] *IEEE Standard for Information Technology-Telecommunications and information exchange between systems-Local and Metropolitan networks-specific requirements-Part II: Wireless LAN Medium Access Control and Physical Layer specifications: Amendment 9: Interworking with External Networks*, IEEE Standard 802.11, Feb. 2011, pp. 1–208.
[31] "Mobile broadband with HSPA and LTE—Capacity and cost aspects," Nokia Siemens Network, Espoo, Finland, White Paper, 2010. [Online]. Available: http://ec.europa.eu/information_society/newsroom/cf/dae/document.cfm?doc_id=4555
[32] C. Phillips and E. W. Anderson. (Oct. 2011). *CRAWDAD Dataset cu/cu Wart (v. 2011-10-24)*. [Online]. Available: http://crawdad.org/cu/cu_wart/20111024
[33] K. Pahlavan and P. Krishnamurthy, "Wireless LANs," in *Principles Wireless Access Localization*. Hoboken, NJ, USA: Wiley, Sep. 2013, pp. 357–404.

Duong D. Nguyen received the B.Sc. degree in electronic communication systems from the University of Plymouth and the M.Sc. degree in mobile and personal communications from King's College London, U.K., in 2008 and 2009, respectively. He is currently pursuing the Ph.D. degree with the School of Electrical and Electronic Engineering, The University of Adelaide. His current research focuses on signal processing challenges for emerging 5G wireless networks, in particular the application of game theory to resource allocation.

Hung X. Nguyen (M'11) received the Ph.D. degree in computer science and telecommunications from the Swiss Federal Institute of Technology, Lausanne, Switzerland. He is a Senior Research Fellow with the Teletraffic Research Centre, The University of Adelaide. He has authored or co-authored over 50 refereed papers in his research topics. His current research interests include wireless software-defined networks, cellular technologies, network measurements, tomography, and privacy preserving techniques.

Langford B. White (SM'00) received the B.Sc., B.E., and Ph.D. degrees from the University of Queensland, Australia, in 1984, 1985, and 1989, respectively. From 1986 to 1999, he was a Research Scientist with the Defence Science and Technology Organisation, Australia, where he was involved in radar and communications electronic warfare. Since 1999, he has been a Professor with the School of Electrical and Electronic Engineering, The University of Adelaide. His current research interests include signal processing, control, and artificial intelligence. From 2002 to 2006, he was a fellow with National ICT Australia Ltd. He was a recipient of the Australian Telecommunications and Electronics Research Board Medal for Best Young Researcher in 1996.