Load Balancing Models based on Reinforcement Learning for Self-Optimized Macro-Femto LTE-Advanced Heterogeneous Network

Sameh Musleh, Mahamod Ismail and Rosdiadee Nordin
Department of Electrical, Electronics and Systems Engineering, Faculty of Engineering and Built Environment, Universiti Kebangsaan Malaysia, 43600, Bangi, Selangor, Malaysia.
sameh.musleh@gmail.com

Abstract—Heterogeneous Long Term Evolution-Advanced (LTE-A) network (HetNet) utilizes small cells to enhance its capacity and coverage. The intensive deployment of small cells such as pico- and femto-cells to complement macro-cells resulted in unbalanced distribution of traffic-load among cells. Machine learning techniques are employed in cooperation with Self-Organizing Network (SON) features to achieve load balancing between highly loaded Macro cells and underlay small cells such as Femto cells. In this paper, two algorithms have been proposed to balance the traffic load between Macro and Femto cells. The two proposed algorithms are named as Load Balancing based on Reinforcement Learning of end-user SINR (LBRL-SINR) and Load Balancing based on Reinforcement Learning of Macro cell-throughput (LBRL-T). Both of the proposed algorithms utilize Reinforcement Learning (RL) technique to control the reference signal power of each Femto cell that underlays a highly loaded Macro cell. At the same time, the algorithm monitors any degradation in the performance metrics of both Macro and its neighbor Femto cells and reacts to troubleshoot the degradation in real time. The simulation results showed that both of the proposed algorithms are able to off-load end-users from highly loaded Macro cell and redistribute the traffic load fairly with its neighbor Femto cells. As a result, both of call drop rate and call block rate of a highly loaded Macro cell are decreased.

Index Terms—Load Balancing; LTE-A HetNet; Small Cells; Reinforcement Learning.

I. INTRODUCTION

One of the 3GPP technologies that meets the high demand for new services is LTE-A HetNet. It integrates various network structures and various cell types. This is for the purpose of offering new data and voice services, improved latencies and higher throughput for end-users. The main nodes of HetNets include High Power Nodes (HPNs) such as Macro eNodeBs, and Low Power Nodes (LPNs) such as Pico and Femto cells. LPNs are defined in 3GPP as small cells. They become important elements of LTE-A HetNet, and they contribute to improve the performance of the whole network in terms of increasing both of the link and system capacity, as well extending the network coverage in both outdoor and indoor networks [1]. The deployment of open-access Femto cells enables Macro cells to reduce the opportunity of being overloaded or congested with a high number of end-users. Moreover, the cost of deploying Macro sites to solve the problems of network capacity and coverage is reduced.

A Femto cell is a low power node. It becomes compulsory that many processes including the installation and troubleshooting of Femto cells need to be automated. This is for the reason that the end-user is not expected to have the enough technical knowledge to be able to install Femto cells or to troubleshoot them. As a result, the Self-Organizing Network (SON) for LTE-A is a new technology that consists of new concepts and functionalities to automate the operation of LTE-A HetNets towards better performance and higher quality of service [1]. Specifically, the operations of self-tuning and self-optimization are defined in SON-enabled LTE-A networks [2], SON is a recent development, and it is part of 3GPP standard for LTE-A [3]. Recently, diverse challenges related to SON-enabled HetNets have been widely researched in various international research projects including 3GPP projects [4],[5]. Various efforts that have been taken to develop advanced Radio Resources Management (RRM) algorithms to decrease the effect of interference in a dense LTE-A HetNets [6].

The traffic load balancing is one of the most demanding topics for both the automation and self-optimization processes in the context of LTE-A networks [7]. The high traffic volumes, as well the unbalanced traffic volumes which are generated from end-users are the motivation for load balancing techniques to be researched. The traffic load balancing is targeting to achieve the balance between LTE-A radio resources and end-users traffic. The process of load balancing affects the Grade of Service (GoS), which is specifically related to call maintainability. Parameters such as radiation pattern power [8], Handover power-margins [9] and reference signal power are optimized to cope with end-users traffic. There have been a few studies researched in the field of load balancing for Macro and small cells in HetNets [10, 11]. Unbalanced traffic is a prominent issue that should be investigated in-depth for indoor and outdoor HetNet deployment scenarios.

Reinforcement Learning (RL) is a technique that is specifically used for interactive learning [12]. It is based on Q-Learning (QL) technique which does not need a system defined by a formula or transfer function. As a result, it becomes an attractive technique to be used to optimize the operations of LTE-A radio access network in real time [13-16].

In this paper, two emerging load balancing techniques have been proposed to overcome the high traffic-load problem of Macro cells in LTE-A HetNet. Both of the
proposed techniques, named as LBRL-SINR and LBRL-T, are mainly employing Q-Learning method to process the degraded performance metrics of Macro cells and to deliver higher link quality for end-users.

II. RELATED WORK

Most researches, which are related to traffic load balancing in LTE and LTE-A are based on making adjustments to the handover or cell selection process in order to manage the traffic distribution between the neighbor cells [17]. The approaches in this field can be classified into Handover-based control and coverage control of a given cell. In the case of Handover-based control, the UEs are steered into specific cells by adjusting the handover offsets of each cell. In coverage control approach, eNodeB will either extend its coverage to reach more UEs or reduce its coverage in case of overloading so that more UEs will handover to its neighbor eNodeBs. The author in [18] explained a method for monitoring the usage of Resource Blocks (RBs) in eNodeB. Whenever the RBs utilization ratio crosses specific limit, it triggers high load status which will initiate optimizing eNodeB’s Reference-signal power. This will reduce the high load at the eNodeB and enable neighbor cells to collaborate in the offloading process.

The author in [19] presented a technique to optimize Jain’s Fairness Index. The proposed technique reallocates UEs towards underlay small cells, which are the Pico, Relay and Femto cells. Both of the Pico and Femto cells use wired-backhaul to connect to the closest eNodeB. On the other hand, Relay nodes use completely wireless connection to connect to its neighbor eNodeBs. In [20], the author proposed an algorithm that monitors eNodeB load based on the Handover process and the capability of neighbor eNodeBs. The algorithm triggers an offloading process whenever neighbor eNodeBs are found to have an adequate capacity. The technique could achieve noticeable performance improvements, especially on UE throughput and BLER.

In [21], the author proposed an algorithm to fairly distribute the eNodeB loads by making reductions in the Handover-overhead, which is necessary for initiating any Handover process. The algorithm is designed based on solving Multi-objective Optimization Problem. There are two conflicting targets to be controlled by the optimizer, signaling overhead and traffic load. A Higher weight is given by the optimizer to the desired target.

III. FORMULATION OF REINFORCEMENT LEARNING TECHNIQUE

An LTE-A HetNet is designed as a Multi-Agent Reinforcement Learning system, in which each Femto cell is defined as an agent [12]. Reinforcement learning deals with the issue of finding strategy for an autonomous agent to perceive and react in its environment to select optimal actions to reach its objective. For every action that the agent takes in its environment, a trainer sets a reward or penalty to trigger the agent to decide about a new state. The states are defined in this paper as a range of possible reference signal power values. An action is defined as the optimal reference signal power value. The agent is learning from the delayed reward in order to select actions that result in the highest possible value of cumulative reward. A Q-learning algorithm is able to achieve the most effective Q-value, based on delayed rewards. This is true regardless of the awareness of the agent about the impact of its actions on the system where actions are applied. Reinforcement learning techniques are associated with dynamic programming techniques, which are used to solve problems related to optimization. The agents collaborate together during the learning process to converge to an optimal policy faster. Meanwhile, each agent during this stage puts the learned policy into action separately, increasing the capability of the designed self-optimization algorithm to run in distributed manner. The nature of LTE-A HetNet is rapidly changing due to the dynamic change in parameters and values related to the mobility of User Equipment (UEs), multipath fading, changing traffic distributions, etc.

Each agent learns through the well-known Markov Decision Process (MDP), in which the agent is aware about a set S of discrete states. Additionally, there is a set A of actions for the agent to implement. At every time interval t of the optimization epoch, the agent acquires the current state s at before it selects a current action a at and executes it. The agent receives a reward r(s, a) and the environment turns to the next state s and a. Both of the δ and r are the main functions in the environment, and the agent might be unaware of them. In MDP, both of the functions δ(s, a) and r(s, a) have a direct correlation with the current state and action, rather than on previous states or actions.

The agent learns a policy π to decide about the next action a, depending on the current acquired state s which is, π(s) = a. A precise way to specify which policy π that the agent will learn is the policy that results in the greatest cumulative reward for the agent. In order to make this requirement specific and more accurate, we set the cumulative value Vπ(s) which is resulted from a random policy π from random first state s as follows:

\[ V_\pi(s) = r^t + \gamma r^{t+1} + \gamma^2 r^{t+2} + \cdots + \gamma^{t+k} r^{t+k} \]

(1)

where the order of reward values \( r^{t+k} \) is produced by starting from state s, and iteratively utilizing the policy π to choose actions as mentioned above (i.e., \( a^{t} = \pi(s^{t}) \), \( a^{t+1} = \pi(s^{t+1}) \) etc.).

Each Femto cell is defined as an agent, whereby it interacts in real time with the environment and selects an action in response to the changing system states. The agent depends on the current Q-values to have the highest possible reward. Meanwhile, it has to identify the actions that produce the highest reward in the long term.

Here \( 0 \leq \gamma < 1 \) is a constant value that shows the relative value of future reward compared to current reward. Specifically, the future reward which is yet to be received are discounted by \( \gamma^k \). If \( \gamma^k \) has the value of 0, then only the instant reward is considered. When \( \gamma \) value closes to 1, the priority is given to the future rewards than the instant reward.

The discounted cumulative reward is defined as \( V_\pi(s) \), it acquires the policy π from the first state s. Logically, further rewards should be discounted relative to immediate rewards because, generally, the agent would prefer to acquire the reward in the shortest possible time steps. We require that each Femto cell learns a policy π that produces the
maximum value of $V^r_\pi(s)$ for the total number of states $s$, which will be referred to as an optimal policy, denoted $\pi^*$:

$$\pi^* = \text{argmax}_\pi V^r_\pi(s)$$  \hspace{1cm} (2)

$V^r_\pi(s)$ is defined as the highest discounted cumulative reward that the agent can gain starting from the initial state $s$. In other words, it is the discounted cumulative reward achieved through executing the optimal policy that is started from state $s$.

It is a challenge for the agent to achieve the optimal policy $\pi^*$ because of the lack of training data which does not offer training examples in the form of $(s, a)$. However, the learner is informed about one thing, which is the sequence of the instant reward $r(s^k, a^k)$ for $k = 0, 1, 2, ...$. This data facilitates the process to learn a numerical evaluation function which can be represented by states and actions, then get the optimal policy in terms of this evaluation function.

One selection for evaluation function is $V^r_\pi(s)$. The proposed LBRL algorithms in this paper should give preference to state $s^1$ over state $s^2$ each time when $V^r_\pi(s^1)$ is higher than $V^r_\pi(s^2)$, as the cumulative future reward is higher than $s^1$. The algorithm policy makes a selection from the states space, and not from the actions space. However, in some cases $V^r_\pi(s)$ can be used to select from the actions space as well. The optimal action to be selected in state $s$ is the action $a$ that produces the highest instant reward $r(s, a)$ added to the amount $V^r_\pi(s)$ of the next state after it is discounted by $\gamma$ as shown in Equation 3.

$$\pi^*(s) = \text{argmax}_a [r(s, a) + \gamma V^r_\pi(\delta(s, a))]$$ \hspace{1cm} (3)

Recall that the variable $\delta(s, a)$ identifies the achieved state from applying action $a$ to state $s$. Further, an agent is defined in this paper as a Femto cell that underlays a Macro cell. The agent that runs LBRL algorithms adopts an optimal policy by learning $V^r_\pi(s)$, then the agent will be equipped with complete knowledge of the instant reward function $r$ and the state transition function $\delta$. As the agent has gained knowledge about the variables $r$ and $\delta$ which are employed by the environment to react to its actions, then the optimal action, $a$, for any state $s$ can be determined. Even though learning $V^r_\pi(s)$ is an efficient way to get the optimal policy, it can be used only when the agent has a complete knowledge of $\delta$ and $r$. This needs the capability to expect the instant result of both of the instant reward and future reward for each state-action pair. Practically, the agent will not be able to expect an accurate result of applying random action to a random state. Whenever the value of $\delta$ or $r$ is undefined, then the process of learning $V^r_\pi(s)$ is useless for choosing the optimal policy. As well, the agent will not be able to estimate Equation 2 in this case. So another evaluation function should be used by the agent for this framework.

The evaluation function $Q(s, a)$ can be determined as shown in Equation 4, so that its value is the highest discounted cumulative reward to be gained by starting from state $s$, initially and executing action $a$.

$$Q(s, a) = r(s, a) + \gamma V^r_\pi(\delta(s, a))$$ \hspace{1cm} (4)

Note that $Q(s, a)$ is exactly the quantity that is maximized in Equation 2 to choose the optimal action $a$ in state $s$. Therefore, we can rewrite Equation 2 in terms of $Q(s, a)$ as

$$\pi^*(s) = \text{argmax}_a Q(s, a)$$ \hspace{1cm} (5)

which indicates that learning $Q$-function instead of learning $V^r_\pi(s)$ will make the agent able to choose an optimal action even though the variables $r$ and $\delta$ are unknown for the agent.

Learning the $Q$-function is similar as learning the optimal policy. The main issue is about figuring out a trustworthy method to estimate $Q$ values from the instant values of reward, $r$. Such a method is possible to be achieved by iterative approximation. This conclusion is coming after noticing the very close relationship between $V^r_\pi$ and $Q$ in Equations 6 and 7 as follows:

$$V^r_\pi(s) = \max_a Q(s, \hat{a})$$ \hspace{1cm} (6)

That allows rewriting as:

$$Q(s, a) = r(s, a) + \gamma \max_a Q(\delta(s, a), \hat{a})$$ \hspace{1cm} (7)

which is an iterative equation that provides us the foundation for an algorithm that iteratively approximate $Q$.

A Q-learning algorithm learns by repeatedly decreasing the differences between the $Q$ values of the succeeding states. It is able to solve optimization problems that deal with systems which are undefined in closed form expression, and it depends on the Temporal Difference (TD) method during the learning process. To estimate the $Q$-value in Equation 7, an agent has the target to choose the action that produces the highest value of long term reward, $r$.

In Section III of this paper, there are two formulas that have been proposed to calculate the reward, $r$, for each of the proposed algorithms. The proposed LBRL algorithms are specified by firstly, controlling the transmitted power of the Reference Signal (RS) at each Femto cell. Secondly, the Reinforcement Learning (RL) as one of the machine learning techniques, which will convert each Femto cell to a smart node that is able to take a decision and auto-tune itself for an optimal state.

IV. MACRO-FEMTO SELF ORGANIZING NETWORK MODEL

The Self Organizing Network (SON) features are considered powerful development in the 4th generation (4G) of mobile networks that are pertaining to the next stage of development which includes 4G and beyond 4G networks [3]. SON features are used when there is rapidly changing traffic, highly fluctuating RF channel or to automate the operator policies which are specifically related to the mobile radio access network. Its main features are categorized into four categories, which are self-optimization, self-configuration, self-diagnosis and self-healing [18]. SON functions have been identified and used by multiple mobile service operators, as it leads to simplified operations and increasing profitability.

Our proposed algorithms utilize SON functions, which include self-diagnosis, self-healing, and self-optimization of Macro and Femto cells in LTE-A HetNet. In order to achieve fair distribution of end-users between highly loaded
Macro cell and its neighbor Femto cells, both of the proposed algorithms are mainly based on the self-optimization concept for SON-enabled LTE-A HetNet, which is mainly employing Reinforcement Learning (RL) and Q-learning techniques to offload end-users from the Macro cell into its neighbor Femto cells.

A set of three performance metrics for highly loaded Macro cell are the main inputs for each of the proposed algorithms, LBRL-SINR and LBRL-T. The three performance metrics are call block rate (B), call drop rate (D), and average SINR, which are specific inputs of LBRL-SINR algorithm. However, B, D, and cell throughput (T) are the specific inputs of LBRL-T algorithm. The SON module at each Femto cell is triggered only when a Macro eNodeB declares a high load state or an overload indicator (OI) is activated, then a Macro cell will trigger the LBRL algorithm to be executed at its neighbor Femto cells, as shown in Figure 1. The signaling between each Femto and Macro cell is carried over X2 or S1 interface. Each Femto cell will independently increase the reference signal (RS) power to increase its coverage region. As a result, the traffic in hot areas is redirected to lightly loaded areas under Femto cells, and thus load balancing is achieved.

The proposed SON architecture is distributed architecture and not centralized. In other words, both of LBRL algorithms do not need to connect to a database to exchange the performance metrics data, while the algorithm is running on live network. The normal signaling over X2 or S1 interface will be enough for each Femto cell to acquire the required performance metrics from its neighbor Macro cell.

Figure 1: Macro-Femto SON model

The three performance metrics which will be exchanged through X2 interface or S1 interface as an alternative. The three performance metrics which will be used to calculate the reward at Femto cell-i are: the average SINR of all end-users at both Macro cell and Femto cell-i at time t (SINR\textsubscript{m} and SINR\textsubscript{i}), Call Drop Rate at Macro cell and Femto cell-i at time t (D\textsubscript{m} + D\textsubscript{i}). The reward function is defined as follows:

\[ r_f^t = (w_1(SINR_{m}^t + SINR_{i}^t) + w_2(D_{m}^t + D_{i}^t)) + w_3(B_{m}^t + B_{i}^t) \]  

where \( w_1, w_2, \) and \( w_3 \) are the weights. SINR\textsubscript{m} is the average SINR for all end-users at time t, SINR\textsubscript{i} is defined as the SINR of UE (k) at Macro cell (m) as defined in Equation 9. The constant \( c \) is to keep the reward \( r_f^t \) value between 0 and 1.

\[ SINR_{m,k}(dB) = P_m + G_m - PL_{m,k}(l_{m,k} + n^2) \]  

where: 
\( P_m = \) downlink transmitted power from Macro cell (m) to end-user (k)  
\( G_m = \) downlink antenna gain of Macro cell (m)  
\( PL_{m,k} = \) Path loss between Macro cell (m) and end-user (k).
Load Balancing Models based on Reinforcement Learning for Self-Optimized Macro-Femto LTE-Advanced Heterogeneous Network

\[ I_{m,k} = \text{The received downlink interference at end-user (k) who connects to Macro cell (m)} \]

\[ n = \text{Thermal noise} \]

The downlink inter-cell interference model is simulated for LTE-A downlink. LTE-A employs Orthogonal Frequency Division Multiple Access (OFDMA) technique for its physical layer, which contributes in achieving higher spectral efficiency for LTE-A in comparison with the previous versions of mobile technologies. The smallest unit of bandwidth to be assigned for each end-user is the Physical Resource Block (PRBs). Each PRB serves a single end-user at a time. Hence, the risk of having intercell-interference is mitigated by the mentioned assignment scheme of PRBs.

As much as the value of the reward, \( r_f \), is high, as much as the Femto cell-i coverage becomes wider. As a result, the optimized reference signal power level will force more end-users to camp on the Femto cell instead of camping on the overlay Macro cell.

VI. LOAD BALANCING BASED ON REINFORCEMENT LEARNING OF MACRO CELL THROUGHPUT (LBRL-T)

This algorithm considers mainly the cell-throughput (T) for all UEs instead of the average SINR in the case of LBRL-SINR, to dynamically control the RS power at each Femto cell. It is assumed that the reference signal power of the Macro cell remains the same and is not subject to be changed by the algorithm. This is to ensure full network coverage and to minimize the chance of creating coverage holes. As at some instant, Macro cell and its neighbor Femto cell may reduce their coverage together at the same time, which will create coverage hole.

In this algorithm, the reward is estimated based on the cell throughput (T) of Macro cell. The T value is one of the main components that constructs the reward function (\( r_f \)) as shown in Equation 10. The state and action of Femto cell-i are modeled in the same way as LBRL-SINR in Section IV, while the process of estimating the reward is different from LBRL-SINR algorithm.

There are three performance metrics, which are required in order to estimate \( r_f \) in LBRL-T, three of the metrics are acquired from the Macro cell and its neighbor Femto cell-i simultaneously. The first metric is the average cell throughput at time \( t (T_{m}^{t} + T_{f}^{t}) \), the second metric is the Call Drop Rate at time \( t (D_{m}^{t} + D_{f}^{t}) \) and the third metric is the Call Block Rate at time \( t (B_{m}^{t} + B_{f}^{t}) \). The mentioned metrics construct the reward function which is defined as follows:

\[
\begin{align*}
    r_{f}^{t} = (w_1(T_{m}^{t} + T_{f}^{t}) + w_2(D_{m}^{t} + D_{f}^{t}) \\
    + w_3(B_{m}^{t} + B_{f}^{t})^n) / c
\end{align*}
\]

(10)

The LBRL-T algorithm keeps monitoring the cell throughput (T) to not degrade at any time instance after the new action, \( a \), is applied. The immediate response of the algorithm after an action, \( a \), is to estimate the new reward value, \( r_{f}^{t+1} \). The higher \( r_{f}^{t+1} \), the higher RS power value to be assigned to Femto cell-i, which means increasing the chance of Femto cell-i to off-load more end-users from its neighbor Macro Cell. As a result, an improved performance will be achieved by decreasing the chance for a Macro cell with high number of end-users to have high rates of dropped or blocked calls (D or B).

However, if the increment in the reference signal power at Femto cell-i was unnecessary or led to unstable performance in terms of causing higher Drop Calls Rate (D) or higher Block Calls Rate (B) at Macro cell side, the algorithm will detect the degraded B or D, and estimates new reward value, \( r_{f}^{t+1} \), in the next optimization epoch which should be lower than the previous reward, \( r_{f}^{t} \). As a result, an optimized action, \( a \), will be applied to reduce the RS power to lower level.

VII. SIMULATION ENVIRONMENT

An LTE-A Heterogeneous Network (HetNet) consists of two types of cells, Macro cells and underlying Femto cells. In 3GPP [22], dense LTE-A HetNet is defined as a heterogeneous network that consists of underlay small cells varies from 4 to 10 cells which are defined as neighbors to their overlay Macro cell. Our simulation scenarios are conducted on system-level simulation which is comprising 7 Macro cells and 42 underlay Femto cells as shown in Figure 3. A number of 6 Femto cells is distributed randomly within the coverage area of their neighbor Macro cell. As well, each Femto cell is defined as neighbor to its nearest overlay Macro cell. The underlay Femto cells are able to communicate with the Macro cell through X2 or SI interface to exchange performance metrics and load information.

The system topology as shown in Figure 3 consists of 7 Macro cells. The center Macro cell is simulated with high traffic load that is originated from a maximum of 100 end-users. The rest of 6 Macro cells is simulated with normal traffic load that is originated from a maximum of 20 end-users. The system bandwidth varies according to the cell type. Each Macro cell has total bandwidth of 100 MHz which is the total available bandwidth from deploying 5 Component Carriers (CCs), each CC provides a channel bandwidth of 20 MHz. Each Femto cell provides a channel bandwidth of 10 MHz. The traffic load of the center Macro cell in the 3 simulation scenarios is simulated to utilize 70% to 99% of the Macro cell bandwidth. Meanwhile, normal traffic load is simulated to utilize a maximum of 25% of the available bandwidth at each cell of the total 6 surrounding Macro cells.

Figure 3: System topology of dense LTE-A HetNet

ISSN: 2180-1843  e-ISSN: 2289-8131  Vol. 9 No. 1  51
Three simulation scenarios have been executed. They are: Fixed reference signal power allocation, dynamic reference signal power allocation by LBRL-SINR algorithm, and the third scenario is a dynamic reference signal power allocation by LBRL-T algorithm. In each of the three scenarios, each UE admits to either Macro cell or its neighbor Femto cell depending on which cell has higher reference signal power value, as shown in Figure 4. If the cell Overload Indicator (OI) is not active, this means that the cell is still able to provide RBs to any new end-user that requests a connection or call. Otherwise, the call/connection request from the end-user will be blocked. A dropped call is recorded if the received signal power of an end-user that has established connection with either Macro or Femto cell is lower than pre-determined threshold value of -110 dBm.

In Figure 6, the improved performance of Macro cell is shown through the reduced rate of dropped calls (D). In other words, the low Call Drop Rate (D) is an indicator for higher percentage of successful handovers (HO) among cells. When LBRL-SINR algorithm is triggered at an underlay Femto cell, it could show the lowest Call Drop Rate (D), as well it showed the lowest Call Block Rate (B) in comparison with both of the reference case and LBRL-T algorithm. This confirms that acquiring the average SINR of end-users instead of the average Cell-Throughput (T) contributes in making more accurate decisions by the QL optimizer to select the best RS power level at each Femto cell. More accurate reward values (rf) were fed to the QL optimizer when LBRL-SINR is triggered. As a result, the LBRL-T algorithm showed sub-optimal performance in comparison with LBRL-SINR, as shown in the Figures 5 and 6.

In the second and third simulation scenarios, both of LBRL-SINR and LBRL-T evolved to new values for reference signal power that fluctuated in the range of 19 ± 3 dBm at each underlay Femto cell. In Figure 7, a comparison is shown for the average reference signal power of the 6 Femto cells that underlay Macro cell 1 (Central Macro), where the LBRL algorithms were triggered and executed during one optimization cycle for each simulation scenario. At each Femto cell, the minimum RS power level was set to 10 dBm, which is the lowest RS power level where neither LBRL-SINR nor LBRL-T will go lower than this threshold.
The complexity and computational cost of LBRL-SINR and LBRL-T are negligible since the proposed algorithms take a few minutes for computing an output with all the needed calculations during each optimization epoch. In addition, the memory requirement is limited. The needed size of the look-up table is considered small, as it contains a set of 4 performance metrics ($B$, $D$, $\text{SINR}$ and $T$) to be exchanged between Macro cell and its neighbor Femto cell once an LBRL algorithm is triggered to run.

IX. CONCLUSION

This paper proposed two algorithms that optimize the degraded performance of LTE-A Macro cells due to high traffic load. The proposed algorithms utilize Reinforcement Learning (RL) techniques to auto-tune the reference signal power of Femto cells, this results in offloading end-users from a congested overlay Macro cell. Both of LBRL-SINR and LBRL-T algorithms optimize the RS power level of Femto cells in real time during every optimization epoch of an On-air Macro cell. As a result, the distribution of traffic load among Macro and Femto cells is improved, and lower rates of dropped calls and blocked calls is achieved for highly loaded Macro cell.

REFERENCES