Dynamic Functional Split Selection in Energy Harvesting Virtual Small Cells Using Temporal Difference Learning

Dagnachew A. Temesgene, Marco Miozzo, Paolo Dini

CTTC/CERCA, Av. Carl Friedrich Gauss, 7, 08860, Castelldefels, Barcelona, Spain
{dtemesgene, mmiozzo, pdini}@cttc.es

***Abstract*—Flexible functional split in Cloud Radio Access Network (CRAN) is a promising approach to overcome the capacity and latency challenges in the fronthaul. In such architecture, the baseband processing takes place partially at local base stations and the remaining processes are executed at the central cloud. On the other hand, we have seen a recent trend of powering base stations with ambient energy sources to achieve both environmental sustainability and profit advantages. As the base stations become smaller and deployed in densified manner, it is evident that baseband processing power consumption has a huge share in the total base station power consumption breakdown. Given that such base stations are powered by energy harvesting sources, energy availability conditions the decision on where to place each baseband function in the system. This work focuses on applying reinforcement learning techniques, in particular Q-learning and SARSA, for optimal placement of baseband functional split options in virtualized small cells that are solely powered by energy harvesting sources. In addition, a comparison of such online optimization solution with respect to offline performance bounds is provided.**

***Index Terms*—energy harvesting, virtual small cells, functional split, CRAN, Reinforcement learning, SARSA, Q-learning, temporal difference**

I. Introduction

Cloud Radio Access Network (CRAN) enables more efficient RAN resource utilization by centralized pooling of the Baseband (BB) processing units [1]. However, this comes at the cost of very high capacity and very low latency fronthaul. To overcome this challenge, flexible functional split between local base station (BS) sites and a central Baseband Unit (BBU) pool is proposed [2]. Significant relaxation on the tight latency and capacity requirements of the fronthaul can be achieved by executing part of the baseband processes locally while maintaining many of the centralization advantages that CRAN architecture offers. In addition, Heterogeneous CRAN (HCRAN) is proposed as an architecture that includes a presence of High-Power Nodes (HPNs) for control plane functions and coverage [3] to partially alleviate the fronthaul challenge in CRAN. Moreover, with the advent of Network Function Virtulization (NFV), network functions can be executed on general purpose computing hardware as virtual functions with Software Defined Networking (SDN) applied as a tool to realize the management and control of the movement of such functions [4].

Energy sustainability is one of the key pillars for future mobile network design and operation. This is mainly due to both environmental and economic concerns [5]. As a means of ensuring sustainability, Energy Harvesting (EH) technology is becoming widely applicable in mobile networks [5]. However, EH comes with its own unique challenge mainly due to unreliable energy sources. Hence, in Energy Harvesting Base Stations (EHBSs), it is important to intelligently manage the harvested energy and to ensure proper energy storage provision to avoid outage.

Most literature on intelligent energy management in EHBSs focus on a HetNet architecture with an intelligent switching on/off scheduling of base stations. The authors in [6] apply a ski-rental framework based online algorithm for optimal switch on/off scheduling. On the other hand, the authors in [7] apply reinforcement learning to optimize the energy usage.The authors in [8] apply Dynamic Programming (DP) to determine the optimal switch on/off policy of a HetNet.

Nevertheless, there is a gap in the literature in embedding EH and incorporating flexible functional split options in HCRAN. The functional splits give insight into considering more operative modes of BSs, in addition to switch on and off.

In our previous work [9], we have studied the performance bounds of dynamic placement of functional split options in an offline manner. The bounds are determined by solving a grid energy consumption minimization problem, based on a-priori knowledge of the system dynamics (traffic and energy arrivals) subject to battery constraints. On the other hand, this paper focuses on proposing an on-line algorithm for the dynamic placement of the functional split options in HCRAN scenarios with EH capabilities.

The main contributions of the paper are:

- Proposing a scenario of HCRAN involving central BBU pool and local small cells having capabilities of EH and dynamic functional split selection. Relying on NFV, the scenario enables dynamic movement of baseband functions between local small cells and central BBU pool according to the choice of the functional split;

- Applying online Reinforcement Learning (RL) based algorithms to find the optimal placement of functional split options for an HCRAN scenario involving a single Virtual Small Cell (vSC) and a Macro BS (MBS) with co-located BBU pool. The approach considers the traffic demand, energy reserve and forecasted energy arrival. In particular, Temporal Difference (TD) learning approach is used and both Q-learning and SARSA algorithms are applied;
- Presenting the performance of the RL based placement of functional split options through numerical results with comparison against offline performance bounds proposed in our previous work [9];

The rest of the paper is organized as follows. Section II describes the considered network scenario, the target functional split options and the power consumption model. In Section III, the RL based energy management with details of both Q-learning and SARSA algorithms are described. The results are discussed in Section IV including the training phase of the algorithms and their performance comparison against offline optimal policy. Finally, we draw our conclusions in Section V.

II. Network Model

We consider a RAN with a MBS with co-located BBU pool and one vSC. The vSC provides service to hotspots, whereas mobility and baseline coverage are provided by the MBS. The vSC is fully powered by solar energy plus batteries and is endowed with limited computational resource that can be used opportunistically, e.g., when enough energy is available, for part of the baseband signal processing tasks. The MBS with the BBU pool is powered by the electrical grid. The connection between the vSC and MBS is provided by a reconfigurable fronthaul and the BBU pool is capable of performing part of the baseband processing. The functional split options that can be applied for the vSC are given in [2]. Considering the potential centralization gains, we have selected the following functional split options as targets in this paper (shown in Fig. 1):

- Standard CRAN – all the baseband processing is done centrally at the BBU pool;
- MAC/PHY – the whole PHY layer processing takes place at vSCs, whereas MAC and above layers are done at the central BBU pool.

As a result, the vSC can be in one of the following three operative modes:

- Standard CRAN mode
- MAC/PHY mode
- switch-off

At each time step, the RL based algorithm decides the optimal operative mode of the vSC by considering the traffic demand, forecasted energy arrival and energy reserve. The details of the RL algorithms applied are given in Section III.

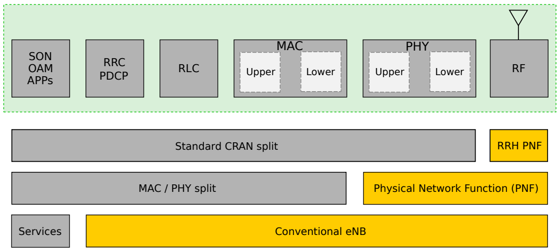

Fig. 1. Functional split options considered in the scenario

A. Power model

The power consumption of each split option is estimated based on the model introduced in [10], which is a general flexible power model of LTE base stations and provides the power consumption in Giga Operation Per Second (GOPS). Technology dependent GOPS to Watt conversion factor is applied to determine the power consumption in Watts. In this paper, we have mapped the various baseband processing tasks of the functional split options to their power requirement estimations. The main baseband processes associated with the split options are shown in Fig. 2.

The total base station power consumption is given by:

$$P_{\mathrm{BS}} = P_{\mathrm{BB}} + P_{\mathrm{RF}} + P_{\mathrm{PA}} + P_{\mathrm{overhead}} \qquad (1)$$

where P_{BB} is the power consumption due to baseband processing, P_{RF} is the power consumption due to RF, P_{PA} is the power consumption by the power amplifier and P_{overhead} is the overhead power consumption, e.g., cooling system.

The baseband power consumption, P_{BB}, is generally computed as:

$$P_{\mathrm{BB}} = P_{\mathrm{BB1}} + P_{\mathrm{BB2}} \qquad (2)$$

More in detail, P_{BB1} is given by:

$$P_{\mathrm{BB1}} = [P_{\mathrm{CPU}} + P_{\mathrm{OFDM}} + P_{\mathrm{filter}}] \qquad (3)$$

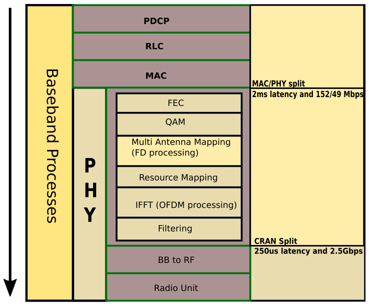

Fig. 2. BBU functions with the considered functional splits and the relevant fronthaul latency and bandwidth requirements (estimations based on [2])

where P_{CPU} is the idle mode power consumption, P_{OFDM} is the power consumption due to OFDM processes and P_{filter} is the power consumption due to filtering. In addition, P_{BB2}, is given by:

$$P_{\text{BB2}} = [P_{\text{FD}} + P_{\text{FEC}}] \tag{4}$$

where P_{FD} is the frequency domain processing power consumption and P_{FEC} is the power consumption due to FEC processes. Estimating these power consumption values mainly depends on bandwidth, number of antennas and the load fraction. In particular, P_{FD} and P_{FEC} are dependent on the traffic load. The power dependence on these factors can be both linear and exponential [10]. The baseband power consumption of the vSC depends on the adopted functional split option, in particular it is given as:

$$P_{\text{BB}}^{\text{vSC}} = \begin{cases} 0, & \text{if vSC is in CRAN} \\ P_{\text{BB1}} + P_{\text{BB2}}, & \text{if vSC is in MAC/PHY} \end{cases} \tag{5}$$

The power consumption of the MBS is determined by (1). Additional MBS power consumption is considered based on the operative mode of the vSC (e.g., additional P_{BB} for a vSC in CRAN split mode).

III. Algorithm

A. Reinforcement learning based energy management

RL is a learning paradigm that relies on learning by interacting with the environment without an exemplary supervision [11]. Formally, the RL framework is defined in terms of states, actions and rewards. Through the RL process, the agent executes a certain action and receives an immediate reward and as the result of the action, its environment will evolve to a new state. It is important to note that in RL, the rewards can be delayed. Hence, it is a sequential decision making process with the goal of maximizing cumulative reward.

For our network model, the objective of the RL based controller is to learn energy management policies through interaction with the environment. The controller decides the operative mode of the vSC at each time steps based on the traffic load, energy arrival and energy storage information. Let S_t be the state of the system at time t, the controller chooses an action A_t from action set $\boldsymbol{A}$, which translates to the operative modes of the vSC. As a result of this action, the environment returns an immediate reward r_t. Based on this r_t, the Q-value, $Q(S_t, A_t)$, which represents the goodness of taking a specific action from a given state, will be updated. This process of selecting a specific action and updating the Q-value continues sequentially for each time steps. The controller selects the action at time step t based on the specific RL algorithm it applies. The goal of such algorithms is to determine the Q-values for each state-action pairs with the goal of achieving an optimal policy in the long-term. In this paper, we have applied algorithms that belong to the class called TD learning.

B. TD learning

TD learning is a class of RL that is based on estimating the Q-values of state-action pairs by making an update at each time-step without waiting for a final return [11]. Hence these methods perform an update of their estimates at time-step $t+1$ based on their existing estimate at t and the methods are able to learn from raw experience without a model of an environment. The algorithms in TD learning can be an on-policy or off-policy methods. In on-policy algorithms, learning an optimal policy is done using the current estimate of the optimal policy where as, in an off-policy methods, learning to approximate the optimal behavior is done independently of the current policy being followed. In this paper, we specifically apply Q-learning and SARSA algorithms which belong to TD off-policy and on-policy algorithm classes respectively.

1) Q-Learning: Q-Learning is an off-policy RL algorithm that can learn the optimal Q-values for each state-action pairs. As long as all state-action pairs are visited and continued to be updated, Q-learning guarantee an optimal behavior regardless of the specific policy being followed throughout the learning phase. The equation for updating the Q-values is given by (6). The procedure of Q-learning algorithm is shown in Algorithm 1.

$$\begin{aligned} Q(S_t, A_t) = Q(S_t, A_t) + \alpha(r_t + \\ \gamma \max_{\boldsymbol{A}} Q(S_{t+1}, A) - Q(S_t, A_t)) \end{aligned} \tag{6}$$

where α is the learning rate, γ is the discount factor, A_t is the current action, r_t is the immediate reward, S_t and S_{t+1} are the current and the next state respectively.

Algorithm 1 Q-Learning Algorithm

```
Initialize $Q(S, A) \forall S \in S, A \in \boldsymbol{A}$ arbitrarily
for each episode do:
    Initialize $S$
    for each step, $t$, of episode do:
        Choose $A_t$ from $S$ using policy derived from Q
        Take action $A_t$, get reward $r_t$ and next state $S_{t+1}$
        update Q-value using (6)
        $S$ = $S_{t+1}$
    end for
end for
```

2) SARSA: SARSA is an on-policy technique of learning optimal policy in RL based problems. In SARSA, the update of the Q-values is done after observing a transition from one state-action pair to the next state-action pair and this transition is dependent on the policy used to select the actions at each time step. The equation for updating the Q-values is given by (7). The procedure of SARSA algorithm is shown in Algorithm 2.

$$Q(S_t, A_t) = Q(S_t, A_t) + \alpha(r_t + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t)) \tag{7}$$

where α is the learning rate, γ is the discount factor, A_t and A_{t+1} are the current and next actions respectively, r_t is the immediate reward, S_t and S_{t+1} are the current and the next state respectively.

Algorithm 2 SARSA Algorithm

```
Initialize $Q(S, A) \forall S \in S, A \in \boldsymbol{A}$ arbitrarily
for each episode do:
    Initialize $S$
    Choose $A_t$ from $S$ using policy derived from Q
    for each step, $t$, of episode do:
        Take action $A_t$, get reward $r_t$ and next state $S_{t+1}$
        Choose $A_{t+1}$ from $S_{t+1}$ using the policy
        update Q-value using (7)
        $S_t$ = $S_{t+1}$
        $A_t$ = $A_{t+1}$
    end for
end for
```

C. Algorithm Details

This section introduces the state, actions and reward functions that are defined for the dynamic functional split selection. The state, action and reward definitions are the same for both SARSA and Q-learning. However, they differ on how they update Q-values as shown in Algorithm 1 and 2.

1) State: The state is dependent on the energy storage level, the normalized traffic load and the harvesting condition. Hence at time step, t, the state is given by (8).

$$S_t = \{H_t, B_t, \rho_t\} \tag{8}$$

Where H_t is the harvesting condition (e.g. daylight and night hours), B_t and ρ_t are the normalized battery status and traffic load which are quantized into 4 and 5 levels, respectively. This quantization levels are sufficient for capturing state variations in state representations.

2) Action: The set of possible actions are the possible operative mode of the vSC. Hence, the action set are switch-off, CRAN split mode and MAC/PHY split mode.

3) Reward: The reward function determines the immediate reward the controller acquire as a result of taking a specific action. Since our goal is to maximize the harvested energy utilization (minimize the grid energy consumption by MBS), the reward definition should reflect this objective. The reward function is given as in (9).

$$r_t = \begin{cases} (1/(A + \rho_t)) - 1/(B * B_t), & A_t \text{ is switch-off} \\ (C * \rho_t) - (1/B_t), & A_t \text{ is CRAN} \\ (D * \rho_t) - (1/B_t), & A_t \text{ is MAC/PHY} \end{cases} \tag{9}$$

Where ρ_t and B_t are the normalized traffic load and battery level at time step t respectively. The rationale behind the reward function is:

- It is desirable to switch-off during very low traffic periods and to operate in one of the split modes (CRAN or MAC/PHY) otherwise. Hence the immediate reward has inverse relationship with the traffic load for switching-off action whereas it is directly proportional to the traffic load for both CRAN and MAC/PHY modes;
- For all actions, there is an immediate penalty which is inversely proportional to the battery level. This helps to avoid the level of battery from falling into very low levels.
- The constants A, B, C and D are used to emphasize the reward according to the desired behavior. For example, choosing higher D than C implies higher immediate reward for MAC/PHY action than for CRAN action.

IV. Performance Evaluation

A. Simulation Scenario

We consider a scenario where one vSC provides capacity enhancement to MBS with co-located BBU pool. The vSC equipped with a solar panel and a battery whereas the MBS with its BBU pool is powered by the grid. User activities are categorized based on [12] as heavy users with an activity of 900 MB/hr and ordinary users with an activity of 112.5 MB/hr. Moreover, we adopt the traffic profiles described in [13], in particular the residential, office and transport area traffic profiles are considered. The solar energy traces are generated using the SolarStat tool [14] for the city of Los Angeles.

The reference vSC power consumption values for our scenario are $P_{\text{RF}} = 2.6$ W and $P_{\text{PA}} = 71.4$ W. For P_{BB}, we consider 200 GOPS, 160 GOPS and 80 GOPS for P_{CPU}, P_{filter} and P_{OFDM} respectively. Moreover, the reference load dependent power consumption values are 30 GOPS, 10 GOPS and 20 GOPS for linear component of P_{FD}, non-linear component of P_{FD} and P_{FEC} respectively. As for the MBS, we consider $P_{\text{RF}} = 9.18$ W and $P_{\text{PA}} = 1100$ W. The baseband power consumption is 630 GOPS and 215 GOPS for the static ($P_{\text{CPU}} + P_{\text{OFDM}} + P_{\text{filter}}$) and load dependent components ($P_{\text{FD}} + P_{\text{FEC}}$), respectively. The power consumption overhead (P_{overhead}) is of 0 and 10% of the total power of the rest of the base station for the case of vSC and MBS, respectively. Other simulation parameters are given in Table I. The solar panel size and battery capacity shown in Table I are dimensioned based on the criteria that the vSC can be fully recharged on a typical winter day.

TABLE I
Simulation Parameters.

Parameter	Value
Solar panel size (m^2)	4.48
Transmission power of macro cell (dBm)	43
Transmission power of vSC (dBm)	38
Bandwidth (MHz)	5
Antenna	2x2
Battery capacity (kWh)	2
GOPS to Watt conversion factor	8

B. Training phase

The training of the TD learning algorithms is performed on a residential, office and transport area traffic profiles. The traffic load at vSC is generated by 90 UEs with 50% heavy users ratio. After simulations involving different values of the training parameters, the values given in Table II are selected as the best combinations. Similarly, for the reward function given in (9), the values of the constants A, B, C and D are chosen to be 1.15, 2, 10 and 20 respectively, as a result of a simulative evaluation. The training is done starting from January, by treating a month as one episode of training. The policy used for action selection during the training phase is an ϵ-greedy policy that can explore random action with probability of ϵ. The value of ϵ is multiplied by an ϵ-discount factor after each episode. The training phase battery status of the vSC in residential area is given in Fig. 3 and Fig. 4 for Q-Learning and SARSA algorithms respectively, for one seed of training. It can be noted that during initial hours of training, the algorithm is unstable and the battery reaches very low level in many occasions. The training is performed with different seeds to add randomness in the energy arrival process. On average over 10 different seeds, the Q-learning achieves stability after about 1750 hours of training whereas SARSA reaches stability after about 2000 hours of training. This is the average training duration computed for 10 seeds and can be interpreted as the duration after which the algorithm can be deployed for operation while improving its performance. Maintaining the battery level above the threshold is an important stability requirement since a wrong decision that results in a lower than threshold battery level can have negative consequences such as, damage in the storage system [15]. For this reason, the battery level is a good indicator to determine the stability of the algorithm.

C. Policy description and comparison with optimal offline policy

This section describes the policies obtained by Q-learning and SARSA and compares them with the offline optimal policy described in [9]. The offline optimization is based on dynamic programming, in particular shortest path search, to determine the optimal policy for each hour of operation by taking into account the battery level, energy arrival and traffic requests. The offline optimization is used as a bound to evaluate how the performance of the online approaches is close to the optimum.

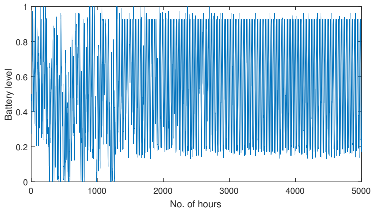

Fig. 3. Battery level during the training phase of Q-Learning algorithm

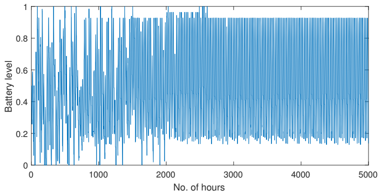

Fig. 4. Battery level during the training phase of SARSA algorithm

For both SARSA and Q-Learning, the training is performed for one year and the evaluation is done for a year of operation after the training. The training and evaluations are done for ten different seeds. A sample of the output of the policies for a week of January and for a residential traffic profile is shown in Fig. 5. As it is shown, both Q-learning and SARSA are able to switch-off during low traffic periods as it is the case also for the optimal offline policy. However, the optimal offline policy and the TD policies differ in the selection of the operative modes. Both SARSA and Q-learning choose the MAC/PHY split mode for most hours of operation, whereas the offline approach chooses the MAC/PHY and CRAN splits evenly. A sample output of the policies for July week and residential traffic profile is shown in Fig. 6. This month is characterized by high energy income. As a result, it can be observed that the policies adjust the operative mode decisions accordingly. All the three policies switch-off during very low traffic periods and all the three policies select MAC/PHY mode predominantly. In addition, SARSA shows higher switch-off rate with almost negligible CRAN selection mode whereas Q-learning maintains relatively lower switch-off rate by selecting CRAN mode during some hours of operation. The policy outputs for office and transport area traffic profile also show similar behavior (not shown here due to space limitation). Residential profile is shown here as it represents traffic peaks during night when energy

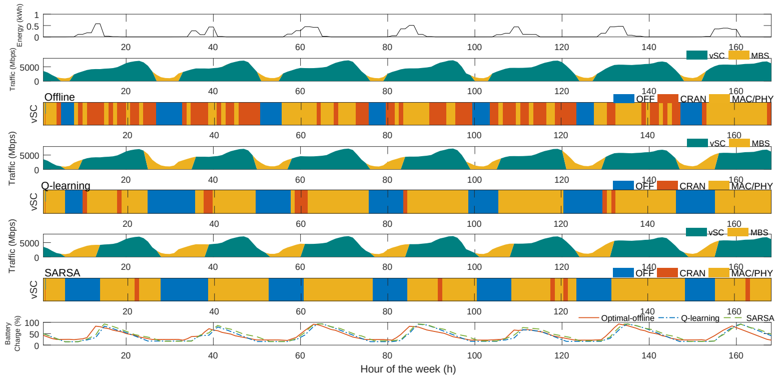

Fig. 5. Functional split selection results in a residential area scenario for a week of January (hour 0 to hour 168; Monday from 0 - 23 hr). The traces show the amount of harvested energy, the amount of mobile traffic handled by vSC and MBS and operative mode of the vSC for a January week for offline, Q-Learning and SARSA policies.

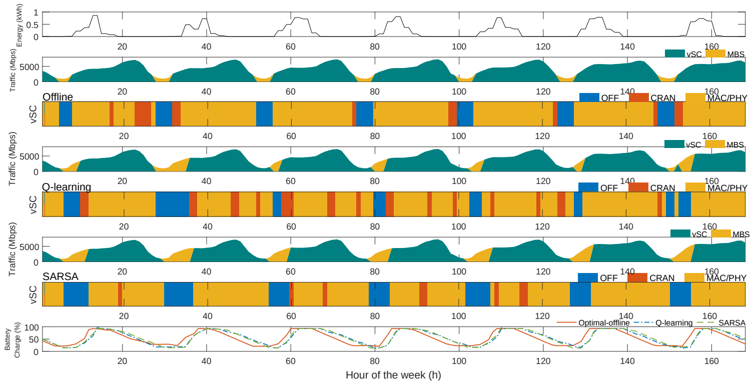

Fig. 6. Functional split selection results in a residential area scenario for a week of July (hour 0 to hour 168; Monday from 0 - 23 hr). The traces show the amount of harvested energy, the amount of mobile traffic handled by vSC and MBS and operative mode of the vSC for a July week for offline, Q-Learning and SARSA policies.

TABLE II
TRAINING PARAMETERS VALUES

Parameter	Q-Learning	SARSA
α	0.8	0.8
γ	0.9	0.9
ϵ	0.5	0.65
ϵ-discount	0.1	0.2

arrival is zero. Hence it is the most representative profile to evaluate the performance of the TD learning policies.

The policies output for a year of operation for three traffic profiles, namely residential, office and transport area are shown in Table III. The results are the averages of ten different simulations running over one year of operation. As it is shown, both Q-learning and SARSA are able to consume a grid energy which is very close to the amount of grid energy consumed by the offline optimal policy. For a year of operation, the average annual grid energy consumption by Q-learning policy is only 2.5%, 1.9% and 1.6% higher than the amount consumed by the offline policy, in residential, office and transport areas respectively. Moreover, for SARSA, the respective residential, office and transport area average annual grid energy consumption are 4.3%, 2.6% and 2.3% higher than the annual consumption by the offline policy. Furthermore, both Q-learning and SARSA are able to achieve a total offloaded traffic of more than 90% with respect to the total offloaded traffic by the offline policy. The TD learning policies also approximate the optimal offline policy in terms of predominat operative mode selection rate. In all the three profiles, MAC/PHY is an operative mode with highest selection rate by the offline policy and this is also reflected by the TD learning policies. In MAC/PHY split mode, most of the baseband processes takes place at vSC utilizing the harvested energy. Hence, the high MAC/PHY split selection rate helps the TD learning policies to achieve close to the optimum annual grid energy consumption. The TD learning polices tend to have relatively higher switch-off rate than the offline policy. This shows the conservative nature of the proposed online policies as compared to the offline approach.

It can also be noted that SARSA policy shows relatively higher CRAN split rate than Q-learning. This explains the higher grid energy consumption by SARSA than Q-learning. Hence, for all traffic profiles, Q-learning performed better both in terms of grid energy consumption and offloaded traffic. This can be attributed to the nature of Q-values update in Q-learning. As an off-policy method, Q-learning learns the optimal behavior by choosing the maximum return that can be gained from one state-action transition to the next one, regardless of what the current policy might choose.

Finally, it is worth noting here that single agent TD learning algorithms are proved to perform optimally given the system model [16]. However, the two proposals in this paper rely on different models with respect to the one used to solve the offline optimization problem in [9]. In particular, the reward function used by SARSA and Q-learning algorithms presents minor modifications to the cost function of the offline optimization, which results in the different performance as presented in this section.

TABLE III
POLICY COMPARISONS (R = RESIDENTIAL, O = OFFICE, T = TRANSPORT, OP = OPTIMAL-OFFLINE, Q = Q-LEARNING , S = SARSA)

Profile	Policy	Grid energy (KWh)	Offloaded traffic (10^7Mbps)	Switch-off rate (%)	CRAN rate (%)	MAC/ PHY rate (%)
R	OP	6780	3.47825	18.26	19.89	61.84
	Q	6952	3.1838	24.3	10.3	65.4
	S	7074	3.1	26	15. 02	58.97
O	OP	6572	1.450	26.55	1.148	72.29
	Q	6700	1.397	26.88	5.41	67.71
	S	6748	1.364	25.77	9.4	64.82
T	OP	6507	0.680	26.63	0.26	73.1
	Q	6611	0.573	30.29	1.5	68.2
	S	6661	0.531	34.6	6.72	58.61

V. CONCLUSIONS

In this paper, we have proposed an optimal functional split placement of energy harvesting virtual small cell that relies on central BBU pool for part of baseband processing. Temporal Difference learning and more specifically Q-Learning and SARSA algorithms are applied to determine the optimal functional split configurations. In particular, three operative modes namely, CRAN split mode, MAC/PHY split mode and switching off have been targeted. Such online approaches are evaluated and compared with respect to an offline optimal policy. Simulation results prove that the two proposed methods perform close to the optimal bounds and confirm the validity of our approach. In addition, Q-learning is observed to perform better than SARSA both in the amount of annual grid energy consumption and offloaded traffic.

ACKNOWLEDGEMENT

This work has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 675891 (SCAVENGE) and by Spanish MINECO grant TEC2017-88373-R (5G-REFINE).

REFERENCES

[1] "C-RAN: the road towards green RAN," *China Mobile Research Institute, White Paper*, 2011.

[2] "Virtualization for small cells: overview," *Small cell forum*, 2015.

[3] M. Peng, Y. Li, J. Jiang, J. Li, and C. Wang, "Heterogeneous cloud radio access networks: A new perspective for enhancing spectral and energy efficiencies," *IEEE Wireless Communications*, vol. 21, no. 6, pp. 126–135, 2014.

[4] H. Hawilo, A. Shami, M. Mirahmadi, and R. Asal, "Nfv: state of the art, challenges, and implementation in next generation mobile networks (vepc)," *IEEE Network*, vol. 28, no. 6, pp. 18–26, Nov 2014.

[5] G. Piro, M. Miozzo, G. Forte, N. Baldo, L. A. Grieco, G. Boggia, and P. Dini, "Hetnets powered by renewable energy sources: Sustainable next-generation cellular networks," *IEEE Internet Computing*, vol. 17, no. 1, pp. 32–39, 2013.

[6] G. Lee, W. Saad, M. Bennis, A. Mehbodniya, and F. Adachi, "Online ski rental for ON/OFF scheduling of energy harvesting base stations," *IEEE Transactions on Wireless Communications*, vol. 16, no. 5, pp. 2976–2990, 2017.

[7] M. Miozzo, L. Giupponi, M. Rossi, and P. Dini, "Distributed Q-learning for energy harvesting heterogeneous networks," in *2015 IEEE International Conference on Communication Workshop (ICCW)*, June 2015, pp. 2006–2011.

[8] N. Piovesan and P. Dini, "Optimal direct load control of renewable powered small cells: A shortest path approach," *Internet Technology Letters*, 2017.

[9] D. A. Temesgene, N. Piovesan, M. Miozzo, and P. Dini, "Optimal Placement of Baseband Functions for Energy Harvesting Virtual Small Cells," in *2018 IEEE Vehicular Technology Conference: VTC2018-Fall*, August 2018.

[10] C. Desset, B. Debaillie, V. Giannini, A. Fehske, G. Auer, H. Holtkamp, W. Wajda, D. Sabella, F. Richter, M. J. Gonzalez, H. Klessig, I. Gdor, M. Olsson, M. A. Imran, A. Ambrosy, and O. Blume, "Flexible power modeling of LTE base stations," in *2012 IEEE Wireless Communications and Networking Conference (WCNC)*, April 2012, pp. 2858–2862.

[11] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press Cambridge, 1998, vol. 1, no. 1.

[12] G. Auer, O. Blume, V. Giannini, I. Godor, M. Imran, Y. Jading, E. Katranaras, M. Olsson, D. Sabella, P. Skillermark *et al.*, "D2. 3: Energy efficiency analysis of the reference systems, areas of improvements and target breakdown," *EARTH*, vol. 20, no. 10, 2010.

[13] F. Xu, Y. Li, H. Wang, P. Zhang, and D. Jin, "Understanding mobile traffic patterns of large scale cellular towers in urban environment," *IEEE/ACM Transactions on Networking (TON)*, vol. 25, no. 2, pp. 1147–1161, 2017.

[14] M. Miozzo, D. Zordan, P. Dini, and M. Rossi, "SolarStat: Modeling photovoltaic sources through stochastic markov processes," in *2014 IEEE International Energy Conference (ENERGYCON)*, May 2014, pp. 688–695.

[15] L. Lu, X. Han, J. Li, J. Hua, and M. Ouyang, "A review on the key issues for lithium-ion battery management in electric vehicles," *Journal of power sources*, vol. 226, pp. 272–288, 2013.

[16] C. J. Watkins and P. Dayan, "Technical note: Q-learning," *Machine Learning*, vol. 8, no. 3, pp. 279–292, May 1992. [Online]. Available: https://doi.org/10.1023/A:1022676722315