Softwarized and Distributed Learning for SON Management Systems

Tony Daher, Sana Jemaa, Laurent Decreusefond

Softwarized and Distributed Learning for SON Management Systems

Tony Daher, Sana Ben Jemaa
Orange Labs
44 Avenue de la Republique 92320 Chatillon, France
Email:{tony.daher,sana.benjemaa}@orange.com

Laurent Decreusefond
Telecom ParisTech
23 avenue d'Italie, 75013 Paris, France
Email:laurent.decreusefond@mines-telecom.fr

***Abstract*—Self-Organizing Networks (SON) functions have already proven to be useful for network operations. However, a higher automation level is required to make a network enabled with SON capabilities respond as a whole to the operator's objectives. For this purpose, a Policy Based SON Management (PBSM) layer has been proposed to manage the deployed SON functions. In this paper, we propose to empower the PBSM with cognition capability in order to manage efficiently SON enabled networks. We focus particularly on the implementation of such a Cognitive PBSM (C-PBSM) on a large scale network and propose a scalable approach based on distributed Reinforcement Learning (RL): RL agents are deployed on different clusters of the network. These clusters should be defined in such a way that the RL agents can learn independently. As the interaction between these clusters may evolve in time due for instance to traffic dynamics, we propose a flexible implementation of this C-PBSM framework with dynamic clustering to adapt to network's evolutions. We show how this flexible implementation is rendered possible under Software Defined Networks (SDN) framework. We also assess the performance of the proposed distributed learning approach on an LTE-A simulator.**

I. Introduction

Autonomic management of mobile networks first appeared with the Self-Organizing Networks (SON) concept [1]. SON standards were first introduced in the 8th release of 3GPP [2], including self-configuration, self-optimization and self-healing functions. Ever since, the SON concept gained popularity and operators showed a lot of interest in enhancing their networks with such autonomic capabilities, hence reducing their Operational Expenses (OpEx).

SON functions are already in the deployment phase in LTE networks in many countries. However, these functions are until now deployed independently. This means that the autonomy is at individual SON functions level, and not at the whole network's level. Because the deployed functions have different individual objectives, such a deployment may lead to conflicts and incompatibility, resulting in suboptimal configurations of the network. A real self organized network is however a network where all the deployed SON functions would cooperate in order to respond to the operator's needs as a whole, and not individually, i.e. the desired autonomous entity would be the network as a whole. Whence the necessity of an entity that ensures the management of SON functions: resolving SON conflicts and managing the SON functions so that the operator's high level objectives (global Key Performance Indicators (KPI) targets) are satisfied. These topics have been addressed by several works in the framework of the Semafour European project [3]. In this work, we are particularly interested in the management of SON functions, so they respond efficiently to the operator's objectives. Some solutions have already been proposed in this context such as the Policy Based SON Management (PBSM) concept that was investigated in [3].

In the next section, we will present an overview of the state of the art and state the problematic and contribution of this paper. The following section presents the proposed distributed RL approach and Software Defined Network (SDN) architecture to enhance the PBSM with cognitive capabilities. In section IV we provide a brief review of the RL theory and describe the considered scenario, we then evaluate in the same section the performances of the distributed learning on an LTE-A simulator. We conclude the paper in the last section and discuss future works.

II. State of the Art and Paper Contribution

A. State of the Art

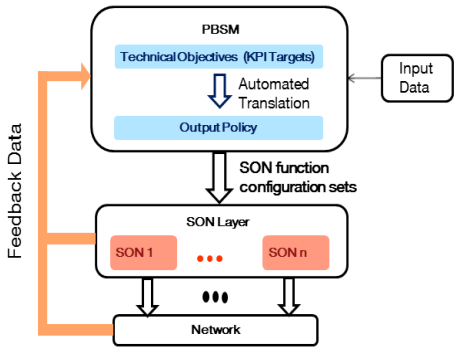

Fig. 1: PBSM functional description

As shown in figure 1, the PBSM translates high level operator objectives into configuration policies to be enforced on the deployed SON functions [4]. These functions

are usually provided by SON vendors as black boxes. The operator has few information about the running algorithm because of proprietary issues. SON Configuration Value (SCV) sets represent a mean for the operator to control and pilot the SON functions deployed in its networks (an SCV set is a collection of threshold values, parameter range, step size ... of the algorithm running the SON function). Furthermore, network KPIs and the SON function behavior depend also on the network context i.e. cell location, time of the day, cell type (macro, pico, femto...) and should also be considered in the process as well.

In [5, 6] the authors propose an approach for PSBM based on the so called SON Function Models (SFM). SFM is a mapping between SCV sets and KPI outcomes of a certain SON function. These models are considered to be provided by the SON vendor and generated using simulations. Nevertheless, this approach presents some drawbacks because it relies strongly on input models such as SFM. These models are obtained by simulation and do not reflect the network reality, especially that the SON function's behavior depends on the context of the cell where it is deployed and on the actions of other deployed SON functions. This drawback becomes even more critical with the increasing complexity of future networks. Also, if the operator has trust issues with the simulations of the SON vendor, or if the latter does not wish to provide the SFMs to the operator, the proposed approach has no solutions but to test SCVs randomly, which may lead to serious performance degradations and delay the convergence.

B. Contributions

It is agreed that cognitive capabilities need to be introduced to the PBSM [3]. A cognitive function is an autonomous and intelligent loop that is able to learn from past actions and observations of its environment states dynamics in order to improve future decisions and adapt to eventual environment changes. Artificial intelligence techniques such as Reinforcement Learning (RL) algorithm [7] can be used as a mean to enhance systems with cognition. The advantages of the RL approach is that there is no need for input models, the PBSM is able to explore and learn the optimal policy starting from scratch. Nonetheless, RL convergence time depends strongly on the size of the set of actions and the set of environment states. In our case, the set of action and states grows fast with the size of the network and the number of SON functions and their possible configurations. We hence propose to deal with this scalability problem through a distributed learning approach, by deploying independent RL agents over network sectors with low interaction with each others.

On the other hand, the emerging SDN architectures introduce new levels of abstraction and flexibility, bringing high expectations in overcoming the limitations of current network architectures, and simplifying the control and management process of network elements [8]. SDN can hence be considered as a mean to facilitate a dynamic deployment of autonomous control agents in the network. This is why for the PBSM, we propose to use the SDN approach to implement a distributed RL framework, that adapts well with the network dynamics. We henceforward refer to the proposed cognitive PBSM as C-PBSM.

The contribution of this paper resides therefore in proposing a distributed RL solution for the cognitive SON management that is able to find optimal policies satisfying the operator's objectives. We also propose an SDN based architecture that can be used to implement the distributed RL agents. This approach is able to adapt to the network dynamics and evolution as will be explained later in the paper. The distributed learning approach was tested on a use case where several SON functions were deployed in an LTE-A heterogeneous network and showed to improve the performances of the network.

III. Distributed Learning Under an SDN Framework

A. Dynamic and Distributed Learning for SON Management

We consider a network section where several SON functions are deployed in a distributed manner i.e. each SON function has several instances deployed on different cells. Each function observes local measurements and KPIs, and takes actions in line with its objectives. We consider that the SONs are provided by several SON vendors and are seen by the operator as black boxes. As stated previously, the operator can steer the behavior of a SON function in a direction or another by configuring it through the SCV sets. Recall that the C-PBSM's objective is to translate the operator's objectives, given as input to the RL agent, into appropriate SCV sets as shown in figure 2.

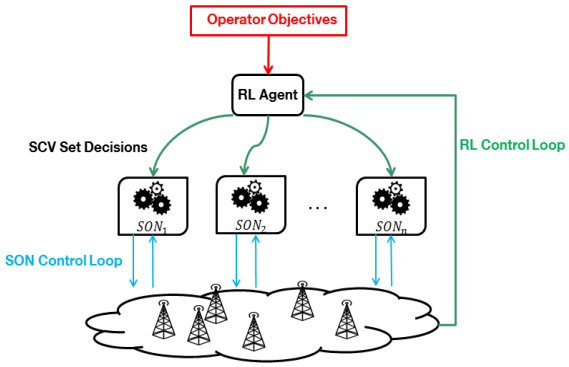

Fig. 2: C-PBSM

Each deployed SON instance (SON_i) has its own control loop (SON Control loop). The outer loop is the RL control loop: the agent observes global KPIs and metrics from the network and takes appropriate actions (in this case RL actions consist of the SCV configurations of the deployed SON instances). Through RL, the C-PBSM is able to observe the system and learn from past experience and use

this knowledge to find the optimal policy corresponding to the specified operator's objective.

In [9] the authors showed that a centralized RL algorithm (Q-learning) is able to learn the optimal policy, satisfying the operator's objectives. However, when the network size, the number of SON functions and instances, as well as the number of possible SCV sets for each SON function grow, the action space that has to be explored by the RL agent explodes, leading to a very slow convergence. This is a well known problem in the RL framework, known as the scalability problem. In fact most of real life problems have huge action and state space and standard RL algorithms fail to efficiently deal with such environments. Such scalability issues can be alleviated through distributed learning.

On the other hand, Radio Access Networks (RAN) are complex dynamic environments. This is mainly due to the fact that in RAN, multiple devices such as user equipments, base stations, radio heads ... are accessing the same medium and interfering with one another. Changing certain configurations of a single cell, for example antenna tilt or transmit power, will affect neighboring cells in term of interference, coverage, traffic load, etc. Furthermore, RAN are highly influenced by the users traffic dynamics, which in turn depend on the human activity. Traffic varies in 2 dimensions: time (hour of the day, period of the year) and space (office, residential or rural areas, etc). Furthermore, the traffic profiles may change on the long term due to the evolution of mobile services (services requiring higher data rates, machine type communications, ultra reliable and low latency services, etc), modifications of the RAN topology (the installation of new sites) and also due to the human activity and the urban expansion (construction of a mall or a new residential building).

Opting for a distributed learning approach in such environments is not straight forward, and must be considered with caution because learning agents may interact with one another. In fact, because of the complexity of the environment and the correlation between cells, changes made by the RL agent to a certain cell will affect the environment of neighboring cells that may belong to the domain of another agent, hence interfering with each others observations and, consequently, slowing considerably the convergence of the learning process if not preventing it at all. Hence, RL agents should be distributed in such a way that each one of them is responsible of a cluster of neighboring cells such that the different learning processes have minimal interaction. We propose hence to run a clustering process based on network metrics, before deploying the RL agents, in order to define the clusters with minimal interaction. This interaction can be expressed in terms of interference level, traffic flux (expressed through handover metrics) and the overlapping coverage area (small cells deployed in the coverage are of macro cells for example) between cells. The deployment process is presented in figure 3. The clustering entity gathers the required metrics from the network and outputs the relevant clusters to a controller. The latter will then associate each RL agent to a network cluster. Once deployed, each RL agent learns a local optimal policy over a cluster of the network according to an operator defined objective: the agent configures the SON functions deployed in the considered cluster, and observes the KPI outcomes and the states of the local system as well (traffic, network configuration ...).

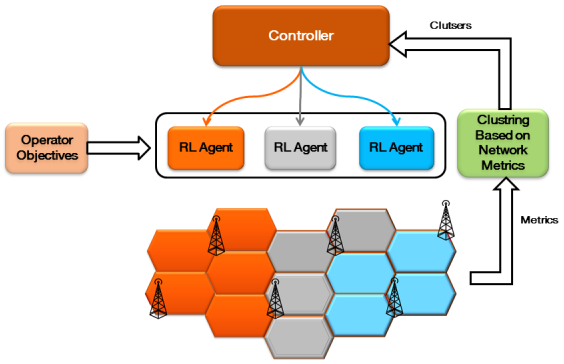

Fig. 3: Distributed RL

On the other hand, because of the traffic and network evolutions mentioned previously (due to new mobile services, changes in the RAN topology, urban expansion, etc), the defined clusters should be kept monitored and adapted to these changes. The clustering process should hence be dynamic, meaning that new clusters can be introduced and existing clusters can be modified on the run. This requires a dynamic redistribution of new learning agents or modification of the environment of the deployed ones. Whence the need of an architecture able to provide such levels of abstraction and flexibility. This can be provided by the SDN framework, as will be discussed in the following section.

B. Software Defined Networks for SON Management

As stated in the introduction, the SDN concept is based on separating the control plane and the data plane and logically centralizing the control process. Such a separation permits the 2 planes to evolve separately, hence introducing new levels of flexibility and abstraction. Also, it helps providing a global and centralized view of the network, thus facilitating the introduction and operation of control processes in the network [8].

Although the first notable applications of SDN architectures were oriented towards core networks and data centers [11, 12], the SDN concept gained also popularity in mobile networks such as OpenRadio and SoftRAN [13, 14]. AutoSDN in its turn is an SDN controller for RAN oriented towards autonomic-network management [15, 16]. AutoSDN's objective is to introduce a new abstraction level to self organized networks that enables SON programmability. With AutoSDN, the SON functions become software applications that the controller

compiles and executes. The parameters and metrics of the network elements are then monitored and updated by the controller's southbound interface. This approach ensures higher flexibility to the autonomic management of RAN. In this paper however, our objective is to introduce flexibility to higher levels of self organizations, namely the C-PBSM.

In fact, SDN and Autonomic Network Management (ANM) are related in the sens that they both seek to simplify and improve the management of complex heterogeneous networks, hence reducing the operational cost and delivering better quality of service to the clients [17]. Furthermore, SDN can be used as a mean to facilitate the introduction of automation and intelligent control loops in complex networks with multiple technologies, layers and vendors. In this paragraph we discuss an SDN architecture, based on the SDN for RAN in [16] that could be adopted to deploy a dynamic and distributed RL for SON management as represented in figure 4. The

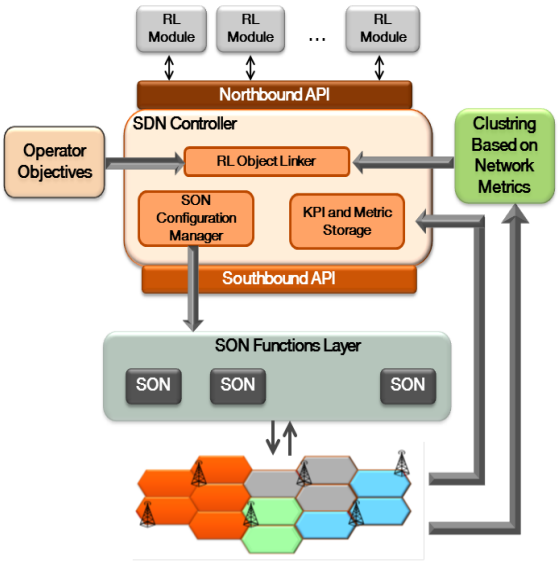

Fig. 4: C-PBSM based on Distributed RL deployment through SDN

learning agents are instantiations of the RL modules, each of them encapsulates the RL code, inputs and output. The inputs of a RL algorithm include the operator's objective and the set of KPIs, metrics and indicators describing the observed environment and the reward functions. The output is the agent's decision, which is in this case the SCV sets of the SON function instances monitored by the agent. The code of the RL object can be any RL algorithm that the agent runs and that permits it to explore and learn the optimal policy. The RL modules are loaded and compiled in the RL object linker, where the operator's objective is specified as well as the network cluster that will be controlled by each agent. The modules are then executed in the controller. The clusters are specified after clustering based on network metrics and topology and are continuously monitored and modified as stated previously. Updated clusters can hence be reintroduced on the fly in the controller, enabling the RL process to adapt accordingly. This applies also for the eventual objective changes. The RL algorithms require constant observations of the environment and the generated rewards. The KPI and metric storage entity keeps an inventory of network KPIs and metrics that can be accessed by the agents. After observing their environments' states and reward, the agents take local decisions and forwards them to the SON configuration manager. The SON configuration manager then enforces the RL decisions in the concerned deployed SON functions instances. In this architecture, the RL modules are merely high level software applications, which simplifies the introduction of new RL algorithms in the system and makes their deployment more flexible and hence adapted to the dynamics of the radio environment. In the next section we briefly introduce the theory behind RL and describe a scenario of SON management using C-PBSM.

IV. C-PBSM Based on Distributed Reinforcement Learning

A. The Reinforcement Learning Framework

RL as defined in [7] is learning how to achieve a goal by interacting with the environment. The learning agent observes the state of the environment and based on this state, it takes an action. The environment reacts by triggering a state change according to a probability distribution that depends on the enforced action and the previous state, and generates a numerical reward. The agent observes the new state and the reward and takes a new action following the new observation. The aim of the agent is to maximize the long term perceived reward. In other words the objective is to find a mapping between states and actions in order to maximize the following expression:

$$R_t = \sum_{k=0}^{T} \gamma^k r_{t+k+1} \quad (1)$$

Where $\gamma \leq 1$ is a discount factor and r_t is the immediate reward perceived at iteration t. The long term perceived reward R_t represents the agent's insight of the rewards in future states starting from iteration t.

There is a variety of RL algorithms in the literature, almost all of them are based on the estimation of the value function and the action-value function. The value functions is defined as follows:

$$V^{\pi}(s) = \mathbf{E}_{\pi}\{R_t | s_t = s\} = \mathbf{E}_{\pi}\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s\} \quad (2)$$

It reflects how good it is for the agent, in terms of the expected perceived rewards in the future states, to be in a certain state s when following a policy π. Respectively, the action-value function, also known as the Q-function, reflects how good it is for the agent, in terms of the

expected perceived rewards in the future states, to be in a certain state s and choosing action a, when following a policy π:

$$Q^{\pi}(s,a) = \mathbf{E}_{\pi}\{R_t|s_t = s, a_t = a\} \\ = \mathbf{E}_{\pi}\{\sum_{k=0}^{\infty}\gamma^k r_{t+k+1}|s_t = s, a_t = a\} \quad (3)$$

Q-learning is a well known RL algorithm that was shown to converge to an optimal policy in single agent setting [18]. It has the advantage of being a bootstrapping algorithm (i.e. it updates its estimates based on other learned estimates, which accelerates the convergence) and it does not need to know the environment transition model (which is the case of most real life problems). This is the algorithm we are going to adopt for the use case study further in the paper. It is however important to note that in the distributed case, the theoretical convergence proofs for single agent Q-learning do not hold. This is due to the fact that an agent's decision can change its environment, but also the environment of other agents. Whence the importance of defining clusters with minimal interactions, as described previously. This model has however been used successfully in many cases [10, 19, 20].

RL techniques have already showed their efficiency in finding optimal policies for individual SON functions such as Mobility Load Balancing (MLB) and enhanced Inter Cell Interference Coordination (eICIC) [21, 22], but also in SON conflict resolution [23, 24] and in different aspects of cognitive radio [10, 19] and many other aspects of networks. In the following we introduce the validation scenario and show the simulation results.

B. Scenario Description

We consider a network section where several SON functions instances are deployed. Let $\mathbb{F}$ be the set of SON functions deployed in the network. N_f is the number of deployed instances of a SON function f and $\mathbb{V}_f$ is the set of possible configuration sets (SCV sets) for SON f, $\forall f \in \mathbb{F}$. Assuming that all instances of a certain SON function f have the same set of possible configuration sets $\mathbb{V}_f$, we define the global action set as:
$\mathbb{C} = (\mathbb{V}_{f_1})^{N_{f_1}} \times (\mathbb{V}_{f_2})^{N_{f_2}} \times ... \times (\mathbb{V}_{f_{|F|}})^{N_{f_{|F|}}}$ where $f_1, f_2, ..., f_{|F|} \in F$.
We also consider the following reward function:

$$r = \sum_i \omega_i . K_i \quad (4)$$

where K_i are normalized KPIs and ω_i are weights that are set by the operator. They are positive and add up to one. They reflect the operator's priority to optimize the corresponding KPI. The optimal policy maximizes the perceived reward in different states of the network.

The LTE-A network is represented in figure 5. We consider the following SON functions:

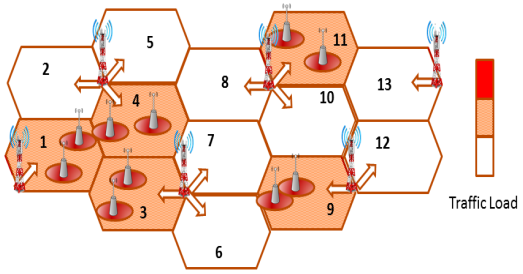

Fig. 5: Network Model

a) MLB: Deployed on each macro cell, its objective is to balance the traffic load between the macro cells by tunning the Cell Individual Offset (CIO) of macro cells. Note that the User Equipment (UE) connects to the cell from which it receives the highest power and CIO.

b) Cell Range Expansion (CRE): Deployed on each small cell, it tunes the CIO of the small cells to balance the load between small cells and the associated macro cell (each small cell is deployed to serve a traffic hotspot inside a coverage region of a macro cell).

c) eICIC: Deployed on macro cells with small cells in their coverage area. eICIC manages Almost Blank Subframes (ABSF) transmissions of macro cells in order to protect small cell edge users from macro downlink interference (ABSF are frames where only control and cell-specific reference signals are transmitted at reduced power).

We consider the following SCV sets:

- MLB:
  - Off: Function is turned off
  - SCV1: Soft configuration
  - SCV2: Aggressive configuration
- CRE:
  - SCV1: Soft configuration
  - SCV2: Aggressive configuration
- eICIC
  - Off: Function is turned off
  - SCV1: Function is turned On

For this scenario, we will consider an RL agent i on each macro cell (if the macro cell has pico cells, they will belong to the same learning cluster as the macro). Let $\mathbb{I}$ bi the set of agents. Each agent has an action space $\mathbb{A}_i$ depending on the SON deployed in its cluster and a state space $\mathbb{S}_i$ depending on the network topology:

- Action space of agent i: $\mathbb{A}_i = (\mathbb{V}_{f_1})^{N^i_{f_1}} \times (\mathbb{V}_{f_2})^{N^i_{f_2}} \times ... \times (\mathbb{V}_{f_{|F|}})^{N^i_{f_{|F|}}}$ where N^i_f is the number of instances of SON function f in network cluster i.
- State space of agent i: $\mathbb{S}_i = \mathbb{A}_i \times (\mathbb{V}_{MLB})^n$ where n is the number of first tier neighbor cells where MLB is deployed and $\mathbb{V}_{MLB}$ is the set of SCV sets of MLB (we consider that only the configuration of the MLB SON function influences the first tier neighbor cells).

- A reward function: $r_i = \omega_1(1-\sigma_i)+\omega_2\bar{t}_i+\omega_3\bar{t}'_i$ where σ_i is the load variance, $\bar{t}_i$ the average user throughput and $\bar{t}'_i$ the average pico cell edge user throughput in network cluster i

The deployment of learning agents in this scenario is rather straightforward and adapted to the considered topology. However when considering more complex topologies with traffic evolutions, a dynamic clustering should be implemented, as described previously in the paper. For this use case however, we focus on the learning task and performance of the agents. The pseudo-code of the distributed Q-learning algorithm for C-PBSM is presented the following.

Distributed Q-learning Algorithm for C-PBSM

for each agent i
Initialize $Q_i(s,a)$ arbitrarily
Initialize s_i
for t=1,...,T
for each agent i
- pick action a_i for state s_i according to ϵ-greedy policy:

$$a_i = \begin{cases} \underset{a_i \in A_i}{argmax}\{Q_i(s_i,a_i)\} & \text{with probability } 1-\epsilon \\ rand(A_i) & \text{with probability } \epsilon \end{cases} \quad (5)$$

- observe new state s'_i
- observe perceived reward $r(s_i,a_i)$
- update Q_i function as follows:

$$Q_i(s_i,a_i) \leftarrow Q_i(s_i,a_i)+\alpha[r+\gamma \underset{a'_i \in A_i}{max} Q_i(s'_i,a'_i)-Q_i(s_i,a_i)] \quad (6)$$

Until convergence

C. Performance Evaluation

We consider a system level simulator built on the 3GPP standards [2] to simulate the C-PBSM. The simulator is semi-dynamic, performing correlated snapshots with a time resolution of 1 second, taking into account path loss and shadowing. Users arrive in the network according to a Poisson arrival, download a file of a fixed size, and leave the network once the download is complete. Users can be dropped due to lack of coverage. We consider only downlink traffic. We run the C-PBSM for different operator objectives and compare the output policy with a baseline approach where the SON functions are configured with static configurations that are usually adopted by operators in similar networks.

The results are shown in figure 6. The graph plots the average achieved reward of the optimal policy identified by the C-PBSM (DQL) and the one achieved by the static configuration, for different operator objectives. The results show that the C-PBSM is able to adapt to different operator objectives and is able to learn and enforce through distributed learning a configuration policy that globally performs always better than a static SON configuration.

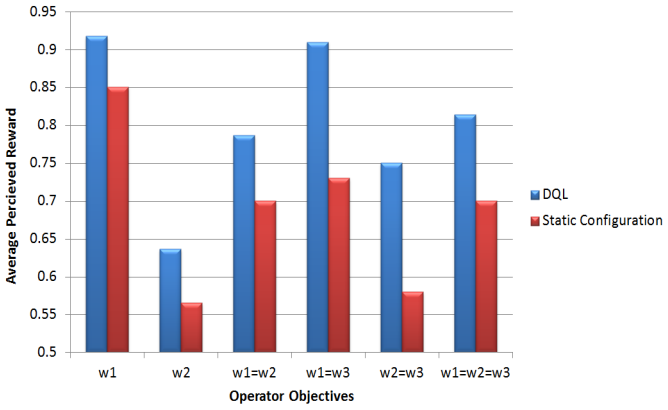

Fig. 6: Performance Comparison

The variations in the perceived rewards between different objectives are due to the normalization of the KPIs considered in the reward function defined previously.

V. Conclusion

In this paper we have presented an approach to enhance SON management in radio networks with cognitive capabilities as well as an implementation based on an SDN architecture. The cognitive capabilities of the PBSM can be achieved through a distributed RL approach that scales well with the size of the network and the big number of possible SON configurations. We have also motivated the need to define network clusters with minimal interactions with each other so that the independence of the distributed learning processes is preserved. These clusters should be continuously updated in order to adapt to the evolutions of the network. An architecture based on the SDN paradigm has been proposed to enable the required flexible and dynamic implementation and deployment of RL agents in the network. The C-PBSM has been evaluated on an LTE-A simulator and the results showed that the global policy enforced by the C-PBSM performs better than a static SON configuration, and is able to adapt to different operator objectives.

In more complex cases, where there is more traffic and topology diversity, as well as more SON functions and SCV sets, the average gain of the RL is expected be even higher than in the scenario considered in this paper. The intuition behind this statement is that the distributed learning obtains its gain from exploiting the differences and diversity of network clusters, and finding the optimal SON configuration for each one of them. This is why in future works, we are motivated to consider more complex network topologies, scenarios and traffic evolutions. In such scenarios, the dynamic clustering of cells and distributing the learning agents accordingly can be better investigated. The scalability and the computational limits of the proposed SDN controller should be studied as well.

References

[1] S. Hämäläinen, H. Sanneck and C. Sartori, *LTE self-organising networks (SON): network management automation for operational efficiency,* John Wiley & Sons, 2012.

[2] 3GPP TR 36.902, "Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Self-configuring and self-optimizing network (SON) use cases and solutions (Release 8)", May 2011.

[3] SEMAFOUR project web page http://fp7-semafour.eu/.

[4] C. Frenzel, S. Lohmüller and L.C. Schmelz, "Dynamic, context-specific SON management driven by operator objectives," *IEEE Network Operations and Management Symposium (NOMS)*, 2014.

[5] S. Hahn and T. Kürner, "Managing and altering mobile radio networks by using SON function performance models," *IEEE Wireless Communications Systems (ISWCS)*, 2014.

[6] S. Lohmüller, L.C. Schmelz and S. Hahn, "Adaptive SON management using KPI measurements," *IEEE Network Operations and Management Symposium (NOMS)*, 2016.

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

[8] D. Kreutz, et al., "Software-defined networking: A comprehensive survey," *Proceedings of the IEEE*, 103(1), pp.14-76, 2015.

[9] T. Daher, S.B. Jemaa and L. Decreusefond, "Q-Learning for Policy Based SON Management in wireless Access Networks," *IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, 2017.

[10] A. Galindo-Serrano and L. Giupponi, "Distributed Q-learning for aggregated interference control in cognitive radio networks," *IEEE Transactions on Vehicular Technology (VTC)*, 2010.

[11] N. McKeown, et al., "OpenFlow: enabling innovation in campus networks," *ACM SIGCOMM Computer Communication Review*, pp.69-74, 2008.

[12] S. Jain, et al., "B4: Experience with a globally-deployed software defined WAN," *ACM SIGCOMM Computer Communication Review*, pp.3-14, 2013.

[13] M. Bansal, et al., "Openradio: a programmable wireless dataplane," *ACM proceedings of the first workshop on Hot topics in software defined networks*, pp. 109-114, 2012

[14] A. Gudipati, et al., "SoftRAN: Software defined radio access network," *ACM proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking*, pp. 25-30, 2013.

[15] G. Poulios, et al., "Autonomics and SDN for self-organizing networks," *IEEE Wireless Communications Systems (ISWCS)*, 2014.

[16] K.Tsagkaris, et al., "An open framework for programmable, self-managed radio access networks," *IEEE Communications Magazine*, 53(7), pp.154-161, 2015

[17] K. Tsagkaris, et. al, "Customizable autonomic network management: integrating autonomic network management and software-defined networking," *IEEE Vehicular Technology Magazine*, 10(1), pp.61-68, 2015

[18] C.J. Bubeck and P. Dayan, "Q-learning," *Machine learning,* vol. 8, pp. 279-292, 1992.

[19] M. Dirani and A. Altman, "Distributed Q-learning for interference control in OFDMA-based femtocell networks," *IEEE Transactions on Vehicular Technology (VTC)*, 2010.

[20] L. Panait and S. Luke, "Cooperative multi-agent learning: The state of the art," *Autonomous agents and multi-agent systems*, pp. 387-434, 2005.

[21] S.S. Mwanje and A. Mitschele-Thiele, "A q-learning strategy for lte mobility load balancing," *IEEE Personal Indoor and Mobile Radio Communications (PIMRC)*, 2013.

[22] M. Dirani and A. Altman, "A cooperative reinforcement learning approach for inter-cell interference coordination in OFDMA cellular networks," *IEEE Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt)*, 2010.

[23] O. Iacoboaiea, O. Sayrac, S.B. Jemaa and P. Bianchi, "SON conflict diagnosis in heterogeneous networks," *IEEE Personal, Indoor, and Mobile Radio Communications (PIMRC)*, 2015.

[24] O. Iacoboaiea, O. Sayrac, S.B. Jemaa and P. Bianchi, "Coordination for parameter conflict resolution: A reinforcement learning framework," *IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, 2014.