8-25-2016

Distributed Cognitive RAT Selection in 5G Heterogeneous Networks: A Machine Learning Approach

Juan Samuel Perez Rodriguez

Juan Samuel Pérez Rodríguez

Candidate

Electrical and Computer Engineering

Department

This thesis is approved, and it is acceptable in quality and form for publication: *Approved by the Thesis Committee:*

Prof. Sudharman K. Jayaweera, Chair

Prof. Manel Martínez-Ramón, Member

Prof. Christos Christodoulou, Member

Distributed Cognitive RAT Selection in 5G Heterogeneous Networks: A Machine Learning Approach

by

Juan Samuel Pérez Rodríguez

B.S., Electronics and Communications Engineering,
Instituto Tecnológico de Santo Domingo, 2010

THESIS

Submitted in Partial Fulfillment of the
Requirements for the Degree of

Master of Science
Electrical Engineering

The University of New Mexico

Albuquerque, New Mexico

July, 2016

Dedication

To my Lord and Savior Jesus Christ, for His continuous blessings to my life. To my parents, Juan Manuel and Elizabeth, for their support, encouragement and instruction. To my sisters Pamela and Paola, for their love and prayers.

"Trust in the Lord with all your heart, and do not lean on your own understanding. In all your ways acknowledge him, and he will make straight your paths."
– Proverbs 3:5-6

Acknowledgments

I am very grateful to my advisor, Professor Sudharman K. Jayaweera, for his invaluable support and guidance throughout my research. I would also like to thank Alan and Jane Lennox, Rex and Claudia Delaney, and Deborah Boro (Bu). Their loving encouragement and assistance was instrumental during my studies.

Distributed Cognitive RAT Selection in 5G Heterogeneous Networks: A Machine Learning Approach

by

Juan Samuel Pérez Rodríguez

B.S., Electronics and Communications Engineering,
Instituto Tecnológico de Santo Domingo, 2010

M.S., Electrical Engineering, University of New Mexico, 2016

Abstract

The leading role of the HetNet (Heterogeneous Networks) strategy as the key Radio Access Network (RAN) architecture for future 5G networks poses serious challenges to the current cell selection mechanisms used in cellular networks. The max-SINR algorithm, although effective historically for performing the most essential networking function of wireless networks, is inefficient at best and obsolete at worst in 5G HetNets. The foreseen embarrassment of riches and diversified propagation characteristics of network attachment points spanning multiple Radio Access Technologies (RAT) requires novel and creative context-aware system designs. The association and routing decisions, in the context of single-RAT or multi-RAT connections, need to be optimized to efficiently exploit the benefits of the architecture.

However, the high computational complexity required for multi-parametric optimization of utility functions, the difficulty of modeling and solving Markov Decision Processes, the lack of guarantees of stability of Game Theory algorithms, and the rigidness of simpler methods like Cell Range Expansion and operator policies managed by the Access Network Discovery and Selection Function (ANDSF), makes neither of these state-of-the-art approaches a favorite. This Thesis proposes a framework that relies on Machine Learning techniques at the terminal device-level for Cognitive RAT Selection.

The use of cognition allows the terminal device to learn both a multi-parametric state model and effective decision policies, based on the experience of the device itself. This implies that a terminal, after observing its environment during a learning period, may formulate a system characterization and optimize its own association decisions without any external intervention. In our proposal, this is achieved through clustering of appropriately defined feature vectors for building a system state model, supervised classification to obtain the current system state, and reinforcement learning for learning good policies.

This Thesis describes the above framework in detail and recommends adaptations based on the experimentation with the X-means, k-Nearest Neighbors, and Q-learning algorithms, the building blocks of the solution. The network performance of the proposed framework is evaluated in a multi-agent environment implemented in MATLAB where it is compared with alternative RAT selection mechanisms.

Contents

List of Figures **xi**

List of Tables **xiv**

1 Introduction **1**

1.1 Motivation 1

1.2 The User Association Problem in 5G Heterogeneous Networks: Literature Survey 3

1.2.1 Perspectives for user association 3

1.2.2 Criteria for user association 4

1.2.3 Proposals for user association 8

1.3 Chapter summary 19

2 Solution Description **21**

2.1 Problem Analysis 21

2.2 Proposed Solution 22

2.3 Contributions of the Thesis 24

2.4 Learning the User/Network State Model 25

2.4.1 The feature vectors. 27

2.4.2 Clustering 37

2.5 Detecting the Cognitive States 43

2.5.1 The role of supervised classification in our framework 43

2.5.2 The k-Nearest Neighbors algorithm 44

2.6 Machine Learning based User Association Policy 45

2.6.1 States and actions. 45

2.6.2 Reinforcement learning and the reward function. 46

2.6.3 The Q-learning algorithm. 49

2.6.4 A modified Q-learning for learning afterstate value functions. . 50

3 Clustering simulations 53

3.1 Introduction 53

3.2 Testing Data Set 55

3.3 One-dimensional clustering 57

3.3.1 Downlink SINR data clustering 57

3.3.2 BS Load data clustering 59

3.3.3 User Mobility data clustering 62

3.4 Bi-dimensional data clustering 62

3.4.1 Downlink SINR vs. Application Class data clustering 63

3.4.2 BS Load vs. Application Class data clustering 65

3.4.3 User Mobility vs. Application Class data clustering 67

3.4.4 BS Load vs. Downlink SINR data clustering 68

3.4.5 Peer ID vs. User Mobility 69

3.5 Three-dimensional data clustering 70

3.5.1 Downlink SINR vs. Application Class vs. User Mobility . . . 70

3.5.2 Peer ID vs. Downlink SINR vs. BS Load 71

3.6 Chapter Summary 73

4 Reinforcement learning simulations 75

4.1 Introduction 75

4.2 Initial tests with Q-learning. 76

4.2.1 Learning a stationary policy. 76

4.2.2 Using the full cognitive model. 79

4.3 Multi-agent simulation set-up 82

4.3.1 Decision Mechanisms. 82

4.3.2 SNR Calculation. 83

4.3.3 Cell Load Calculation. 84

4.3.4 Computation of Rewards. 85

4.4 Multi-agent simulation results. 87

4.4.1 3 client nodes 2 serving nodes scenario. 87

4.4.2 6 client nodes 2 serving nodes scenario. 91

4.4.3 10 client nodes 10 serving nodes scenario. 95

4.5 Chapter Summary 99

5 Conclusions and Future Work 100

References 103

List of Figures

1.1 Perspectives for the User Association Problem 4

2.1 High level overview of the proposed solution. 24

2.2 Collection of feature vectors 28

2.3 Regions of QoS requirement 34

2.4 Illustration of the output of X-means output for 3 RATs observed . 39

2.5 Elements of the Decision Process considering an observation and an afterstate. 51

3.1 Distribution of the Testing Data Set for $N = 300, 500, 1173, 42579$. . 56

3.2 Clustering of Downlink SINR Data (1) 58

3.3 Clustering of Downlink SINR Data (2) 58

3.4 Clustering of Downlink SINR Data (3) 59

3.5 Clustering of Base Station Load Data (1) 60

3.6 Clustering of Base Station Load Data (2) 60

3.7 Clustering of Base Station Load Data (3) 61

3.8 Clustering of User Mobility Data 62

3.9 Clustering of DL SINR and Application Class Data (1) 63

3.10 Clustering of DL SINR and Application Class Data (2) 64

3.11 Clustering of DL SINR and Application Class Data (3) 64

3.12 Clustering of BS Load and Application Class Data (1) 65

3.13 Clustering of BS Load and Application Class Data (2) 66

3.14 Clustering of User Mobility and Application Class Data 67

3.15 Clustering of BS Load and DL SINR Data 68

3.16 Clustering of Peer ID and User Mobility Data 69

3.17 Clustering of DL SINR, Application Class and User Mobility Data . 70

3.18 Clustering of Peer ID, DL SINR and BS Load Data (1) 71

3.19 Clustering of Peer ID, DL SINR and BS Load Data (2) 72

4.1 Cognitive states model generated for Reinforcement learning tests . 76

4.2 State dynamics of the system. 77

4.3 Highlighted location of clusters 28 and 43. 78

4.4 Accumulated rewards by Q-learning after learning a stationary policy. 79

4.5 Accumulated rewards by Q-learning handling the full cognitive model. 80

4.6 Performance comparison of Q-learning for different values of ε 81

4.7 Performance comparison of Q-learning for different values of α . . . 81

4.8 Network Topology. 3 client nodes and 2 serving nodes 88

4.9 Network simulation results. 3 static clients nodes and 2 serving nodes. 89

4.10 Network simulation results. 3 mobile clients nodes and 2 serving nodes 90

4.11 Network Topology. 6 client nodes and 2 serving nodes 92

4.12 Network simulation results. 6 static clients nodes and 2 serving nodes 93

4.13 Network simulation results. 6 mobile clients nodes and 2 serving nodes 94

4.14 Network Topology. 10 client nodes and 10 serving nodes 96

4.15 Network simulation results. 10 static clients nodes and 10 serving nodes. 97

4.16 Network simulation results. 10 mobile clients nodes and 10 serving nodes. 98

List of Tables

2.1 Suggested mapping of Tier classes. 31

2.2 Proposed mapping of Application classes. 34

2.3 Example of Application classes based on nDPI supported protocols . 35

4.1 Q-Table after convergence 79

4.2 Log-Normal Fading Propagation Model parameters 84

4.3 RAT Parameters for Network Simulation 84

Chapter 1

Introduction

1.1 Motivation

The perpetual increase in the demand of broadband access in today's world, supported by the need for ubiquitous high-definition multimedia, interactive applications and augmented reality techniques is far from being satisfied by 4G networks. In the next few years, the introduction of in between 2.9 and 13 billion connected Internet of Things (IoT) units, as forecasted by Gartner [1], only increases this gap.

The challenges of ever-growing demand are forcing engineers to ask themselves: "What is next?" While many different ideas are fighting in the arena to shape a new fifth-generation technology standard, there is increasing consensus that what is envisioned for 5G may come from multiple infrastructures, technologies, and topologies. Indeed Chih-Lin I, chief scientist of wireless technologies of China Mobile, verbalized this idea: "In the past, we talked about 1G, 2G, 3G, and 4G in a quite narrow sense of next-generation mobile standards," but "what we call 5G should be renamed 5G Era because that is a wide sense of 5G" [2]. Such a broad sense is justified by

Andrews et al. in their statement: “an incremental approach [of existing 4G technologies] will not come close to meeting the demands that networks will face by 2020” [3].

5G networks research is expected to lead to: x1000 increase in capacity and negligible latency in the connections, new market-driven ways of spectrum sharing and access, major shifts towards virtualization in both the core and radio access networks, the increasing importance of the IoT devices, unparalleled energy efficiency, the introduction of new communication standards (e.g., mmWave) and Massive MIMO techniques, and an increasing integration of past, current and future cellular, WiFi and Device-to-Device Communications (D2D) standards. [3–5]

Undoubtedly, this increasing integration of wireless wide area networks, local area networks and D2D standards will be achieved through the Heterogeneous Networks (HetNets) strategy, under the principle of extreme densification and offloading. A mixture of several types of radio base stations with different cell sizes that support agressive spatial reuse is what we call a HetNet. This strategy aims to improve the area spectral efficiency (i.e., more active nodes per unit area and Hz) of a wireless network. In [3] extreme densification and offloading is considered as one of the key approaches to achieve increased data rate. In [6], it is elevated over the spectrum shortage challenge as the key area for achieving the x1000 increase in network capacity needed for meeting the demand that is forecasted for 2020. In [4] the HetNet architecture is presented as the driving architecture for 5G networks, spanning various Radio Access Technologies (RATs), cell-sizes, topologies and frequency bands.

In this Thesis, we analyze the problem of determining which RAT standard and spectrum to utilize and which BS(s) or users to associate (a.k.a, the User Association Problem) within the context of 5G HetNets. “To the mobile user, who may be within range of many BSs or WiFi access points (APs) over dozens of different frequency bands, all that really matters is whether some of them can jointly deliver the rate and latency that the user applications require" [6]. However, this is easier to say than to do. 5G HetNets turn the classic, and extremely important, user association function into a complex decision process for the network.

The following section of this Chapter summarizes our survey of the different approaches and perspectives found in the literature related to the User Association Problem in 5G and HetNets. After a brief analysis of the objectives guiding these approaches, we expound the criteria used for the association decisions, and proceed to discuss the state of the art proposals for single-RAT and multi-RAT association. The following chapters of this Thesis describe our proposed solution and guide through our experimentation of its feasibility.

1.2 The User Association Problem in 5G Heterogeneous Networks: Literature Survey

1.2.1 Perspectives for user association

The consulted literature reflects three main perspectives for the user association problem in HetNets. (1) The user association problem can be studied from a load balancing perspective (e.g., [6–10]), with the main goal of improving the experience of both macro-cell users and offloaded users. (2) The user association problem can also be studied from an enhanced mobility perspective (e.g., [11, 12]), with the objective of making the mobility as seamless and transparent as possible to the user, and (3) Finally, a Self Organizing Network (i.e., distributed autonomous intelligence) perspective (e.g., [13–15]), in order to reduce the overhead, increase the network efficiency and ease the management tasks for highly diverse and complex networks.

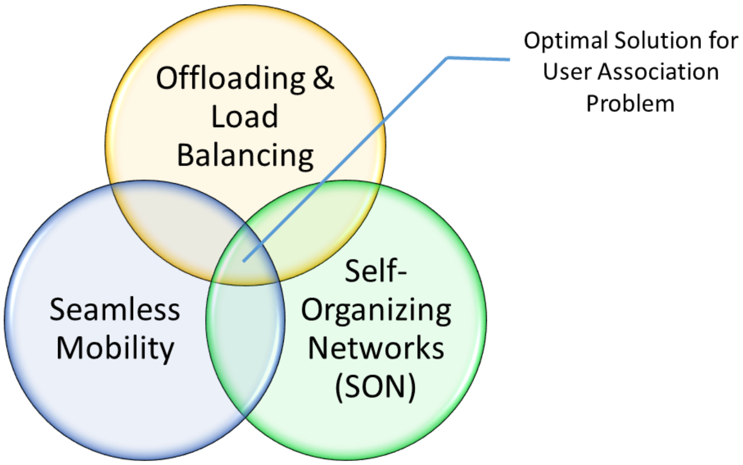

Offloading & Load Balancing
Optimal Solution for User Association Problem
Seamless Mobility
Self-Organizing Networks (SON)

Figure 1.1: An optimal solution for the user association problem needs to integrate the benefits of network load balancing, an enhanced user experience through seamless mobility and has to be flexible enough to automatically adapt to changes and failures while keeping a low overhead.

Although the solutions present in the literature often lean towards one of these perspectives, it is not difficult to realize that these objectives are not necessarily mutually exclusive. Indeed, we suggest that an ideal solution for the user association problem for HetNets is found at the intersection of these three perspectives, as illustrated in Fig. 1.1.

1.2.2 Criteria for user association

SINR

Traditionally, the simplest user association mechanism used in cellular systems relies on Downlink Radio Frequency (RF) measurements: The user terminal connects to the BS that will result in the highest SINR (signal to interference plus noise ratio). The logic behind this algorithm, known as the max-SINR rule, is built on top of the results of Information and Communication Theory; namely, the error rate and maximum achievable data rate with negligible error in a communication channel are affected directly by the level of SNR. The higher the SNR (or SINR), the better the performance of the radio link. A good radio link performance is, then, assumed to be equivalent to a good networking experience.

Cell Load

More recently, new proposals are questioning these assumptions and advocate for the use of additional criteria for user association [6]. The use of the level of congestion (load) of the cells has gained momentum. The cell load must be considered because the throughput perceived by the final user is a time average of its optimal bit rate over a number of radio resources that such user has been allocated during the measured interval. Therefore, it is expected that a user will experience a very poor service from a highly-loaded cell regardless of how high the SINR measurement might be (under simplistic scheduling assumptions).

Although applicable to macrocell selection, these new proposals are specially important in the context of HetNets. When the architecture of a network includes macro- and small cells (i.e., micro-, pico- and femto-cells), the reduced coverage of the latter makes them less attractive to the terminal devices using the max-SINR algorithm, regardless of how loaded the macrocells might be. This can lead to major load imbalance, particularly when both network tiers are using the same spectrum. However, load-based user association is a complex combinatorial optimization problem with exponential growth as the network size increases. Hence, simpler indirect load-aware methods have been proposed.

Panah et al. [8] use optimization over a utility function based on throughput measurements. In their solution a single small-cell has two radio interfaces; one of them is an LTE interface, while the other one is a Wi-Fi interface. Both interfaces operate on orthogonal carriers. Initial "warm up" frames in which the user connects and sends data to both interfaces allow for the calculation of the average achievable throughput. The distribution of users in between the interfaces (i.e., the RAT selection decision) depends on the results of these measurements.

Kwon et al. [9], on the other hand, rely on the use of a binary indicator of congestion for controlling the admission to highly-loaded BSs. In their solution, depending on the average queue sizes, both the terminal and the BS make use of explicit congestion notification (ECN) information in the IP header of the forwarded packets. The terminal informs the congestion condition to the Access Network Discovery and Selection Function (ANDSF) server of the LTE network. The BS information is updated in the database and makes the cell not eligible for future access requests while the congestion condition is still present.

Other criteria

Besides SINR and cell load, other criteria are becoming increasingly important for the user association problem:

Much of the literature that addresses offloading or UE-steering from a cellular network to a WiFi network is mostly concerned with the cell-edge users of the cellular network. Therefore, characterizing the state of a user as cell-edge condition can be used for simple offloading. This improves the overall throughput of the macro-cell system and the service quality of the cell-edge users. For example, Gonçalves et al. [10] propose a device-based solution in which the devices threshold their Reference Signal Received Power (RSRP) with a value that is broadcasted as part of the standardized signaling of LTE/LTE-A networks. Based on this operation, the device recognizes whether it is in a cell-edge state and requests suitable nearby WiFi offloading options from the ANDSF.

The level of mobility of the device can be an important criteria. A high mobility user can be better served by an association with a macro base station with a more extensive coverage than by WiFi APs or other cells with smaller coverage. Recognizing the state of a device as static or mobile, therefore, can be used to restrict the number of unnecessary network scans and reduce ephemeral connections with small cells. This results in improvements in the connection stability and terminal power efficiency. Although the mobility of a user can be derived from GPS information, this is a very low power-efficient technology to be used on a regular basis. Other creative methods have been recommended: the authors of [16] propose the use of the accelerometer that is built-in in many modern smartphones, while the authors in [12] propose an algorithm that infers device mobility using the received signal strength indicator (RSSI) and the base station ID (BSID) information.

Context Awareness

It is likely that in the near future other parameters will be considered relevant for the continuous co-optimization of a HetNet, specially for the user association task. There is potential, for instance, in the use of Uplink RF measurements and other context-awareness information as useful criteria. Context awareness is the collection of information from the network, devices, applications, the user behavior and its environment, acquired in order to optimize network processes and personalize services for the users [4, 17]. Some interesting examples of the use of context-awareness information for our case of interest:

(1) It is desirable to analyze optimal association algorithms that are application-dependent. This could be used, for example, for offloading users running applications with loose QoS requirements to small cells enabled with random access technologies (e.g., WiFi APs). On the other hand, users of delay-constrained and interactive real-time applications could be assigned to macrocells equipped with channel-allocation access technologies. If simultaneous connections are possible to many RATs, such distribution could be done on a per-IP-flow basis. (2) The presence of proximate devices can be exploited by prioritizing D2D communications in certain scenarios, with gains in delay and power consumption. (3) Provided that the small cells backhaul will be a major challenge, and probably, the main bottleneck of a HetNet, a backhaul-aware algorithm is also desirable. At the time of writing this Thesis, we are not aware of concrete proposals that consider these criteria.

1.2.3 Proposals for user association

Optimal distribution of resources is not a new problem; it is found in economics, load balancing among computer servers and even in scheduling operations in wireless networks. However, provided the non-uniform nature and the considerable amount of parameters and constraints of HetNets, no approach seems to be uniquely optimal. Nevertheless, it is possible to extract some guidelines from the consulted literature that seem to have led most of the analyzed solutions. We understand that practical proposals will need to, at least, consider these expectations.

The *simplicity of procedures* has been favored by the experts. Thus, elegant mathematical approaches like relaxed optimization of utility functions, Markov Decision Processes and Game Theory algorithms have been deprecated. Instead, biasing methods, which have been claimed to provide a surprising close-to-optimal performance if the biasing values are chosen carefully [6], and the use of coordinating nodes in the network like the ANDSF, are preferred. Furthermore, the opinions favor the use of a *hybrid approach* that combines uncoordinated and coordinated elements [4] and distributed (client-based) and centralized (server-based) [15] aspects in the solution. An uncoordinated solution (like a non-cooperative game) could be suboptimal compared to a coordinated proposal but reduces the complexity of the optimization and the overall signaling overhead. Distributed algorithms have been shown to provide considerably good results under simplistic assumptions [6] but tradeoff the mobile network operator (MNO) visibility and control over the traffic and the user experience [12]. A good example of a useful compromise is suggested by [11]: A hybrid centralized-distributed approach would allow the exchange of rules with a centralized node that do not control directly the behavior of the nodes, but restrict their actions.

Finally, it seems clear that the transformation of the cellular networks by means of *Network Function Virtualization (NFV) and Software Defined Network (SDN) technologies*, the *advances in networking protocols* and *innovative HetNet architecture design* will carry with them substantial simplifications and flexibility. In future 5G networks, it is likely that a device will see itself as associated to the network instead than to a radio station. In this sense, the user will share a connection with multiple tiers (i.e., macrocells, micro-cells, APs and other devices) simultaneously. Such connections might be suggested by the network, based on context-awareness data obtained from the user device and the network state. Hence, the user will enjoy the benefits of a fully coordinated infrastructure. The traffic will be aggregated at the end device by powerful Connection Managers. These concepts may simplify the network selection process and will ensure ubiquitous connectivity and seamless mobility.

Single-RAT user association

In this section we comment on the technical approaches found in literature for the user association to a single RAT (i.e., a single network attachment point) when multiple RATs are available in a HetNet.

1. Optimization of utility functions.

   Normally, network load functions subject to a given SINR constraint or (log-utility) throughput functions are utilized. The main inconvenience of this approach is its high complexity. The authors in [6] present a series of assumptions for relaxing the constraints of the optimization problem that allow to create a low-complexity distributed algorithm using dual decomposition techniques. Shen and Yu [7] use a coordinated descent approach over a dual function, instead. Their method provides them with the flexibility of achieving both offloading and power control with a balance in between a centralized and distributed design. Other authors suggest the use of exhaustive search or heuristics for solving the problem using a centralized approach. [8]

2. Markov Decision Processes.

   The idea of this method is to choose the actions that maximize the future expected reward. The challenges for applying this technique are in the modeling of the system (i.e., how to define adequate states and state transition probabilities) and in the difficulty of solving the MDP as the network grows.

3. Game Theory.

   Game theory is a convenient way to model and analyze distributed uncoordinated interactions among rational agents. The main challenges of this approach are stated by Andrews et al.: (1) “The convergence of the resulting algorithm is, in general, not guaranteed." (2) "Even if the algorithms converge, they do not necessarily provide an optimal solution". (3) "There is no closed-form expression to characterize the relationship between a performance metric and the network parameters." [6]. This is exemplified in [18], where Aryafar et al. explore the Nash equilibria and Pareto-efficiency of single-class RAT selection games and when a mixture of classes are present. Their results show convergence of the Nash equilibria for the first case and the need of appropriate hysteresis policies for achieving the equilibria in the second.

4. Biasing (a.k.a, Cell Range Expansion) and Blanking.

   By the use of biasing, the users can be offloaded to smaller cells using an association bias. A bias, in this context, is an artificial offset value added to the SINR measurements in order to encourage the association with small cells. Provided that "biasing effectively expands the range/coverage area of small cells, so is referred to as cell range expansion (CRE)" [6]. Blanking, or Almost Blank Subframes (ABS), as it is known in 3GPP documentation, is an interference management method that allows to boost the SINR received from small cells. "Blanking, refers to shutting off the macrocell transmissions for some fraction of the time, preferably while the biased small cell users are being served" [3]. This technique is specially useful when both macro- and small cells are operating using the same frequency bands. The combination of both techniques has been shown to increase edge rates by as much as 500% [3]. Andrews et. al. [6] provide rules of thumb for the selection of optimal biases and blanking percentages under fairly simple assumptions:

   - Optimal biasing is considerably aggresive (e.g., 20 dB or more) in out-of-band offloading. The optimal offloading bias decreases as the network density increases.

- Bias values for co-channel offloading are around 5-10 dB if blanking is not used and can increase to aggressive values as blanking is increased. “The small cell density does not affect the optimal offloading bias, because the interference they cause affects all users equally”.
- The optimal amount of blanking grows in proportion to the small cell density if offloaded users can also be served during non-blanked time slots; however, for plausible small cell deployments (5-7 micro-cells per each macro-cell) and independently of the serving of offloaded users strictly during ABS or not, it is approximately 50%.

5. Policy-based.

   The Access Network Discovery and Selection Function (ANDSF) is a core network entity defined in the 3GPP standards created in order to regulate the interaction and offloading tasks between 3GPP and non-3GPP networks. The ANDSF interacts directly with the terminals by the use of XML over IP through the S14 interface defined in [19, 20]. The ANDSF allows the mobile devices to discover available access networks and assists in choosing the best candidate, based on operator-defined policies (e.g., best QoS, lowest charges, etc.). The ANDSF exchanges three types of information through the S14 interface:

   - Discovery Information: A list of prioritized available networks within range.
   - ISMP: Operator-defined rules for selection of one active access network.
   - ISRP: Rules for access selection for multiple simultaneous IP connections.

   There are many advantages to this approach: (1) Improved visibility and control over the BS selection. (2) Simplicity, due that it is only based on rules, and (3) It is an already standardized mechanism for LTE/LTE-A networks.

In the consulted literature, the potential of the ANDSF framework is clear. For instance, only slight modifications of the standards are required in order to achieve reduced power consumption in the process of network scanning [16], congestion control [9], and offloading of cell-edge users [10].

Nonetheless, a policy-based approach is suboptimal. Further, the major shortcoming of this method is eloquently commented by Gonçalves et al.: the “ANDSF specifications, [...] still do not define procedures and messaging allowing for real time network data to be sent to the UE, like real-time radio link load or congestion status. These factors are of utmost importance when deciding to switch from one RAT to another, so that a UE is not switched to a lower performance network” [10]. Of course, if such real-time procedures and messaging would be created, they should be optimized in order to reduce the network overhead as much as possible.

Multi-RAT user association

This section expounds multi-RAT user association by means of simultaneous connections to several radio nodes operating different wireless technologies. This seems to be a relatively unexplored area with many opportunities for research. Below, we enumerate the efforts found in the consulted literature in this direction:

1. Cellular standards extended to Unlicensed Band.

   The simplest approach to a simultaneous multi-RAT connection in the consulted literature is the use of LTE in the 5 GHz unlicensed band, also known as ‘LTE-U’. The development of LTE-U is based on the significant amount of underutilized spectrum in the WiFi bands and is undergoing standardization in 3GPP Release 13.

The main challenge of LTE operating in the WiFi band is the risk that LTE could take over all the spectrum and degrade significantly the performance of the WiFi systems that operate in the band. This is caused by both the design of LTE as a licensed spectrum technology and to the Listen-Before-Talk (LBT) mechanism implemented in WiFi. Before any transmission, a WiFi interface senses the use of the channel and restricts itself from sending data if the channel is busy. Provided that LTE doesn't have a similar mechanism, coexistence methods need to be considered. A simple solution would be to implement the LBT algorithm for LTE-U; another option would be to use coexistence gaps where the channel is simply not used by LTE allowing for the action of WiFi. Al-Dulaimi et al. [21] propose a system that uses both concepts and integrates a carrier frequency switching mechanism in order not to interrupt the LTE-U transmission. There is considerable space for research of coexistence solutions using spectrum underlay and spectrum interweave schemes.

Regardless of what the final specifications of LTE-U could be, the strategy of Qualcomm [22] is of our interest. They propose the use of LTE-U on demand as an offloading method for LTE/LTE-A users using the Carrier Aggregation (CA) feature (3GPP Release 10/11). "The same small cells that offer LTE Advanced on licensed spectrum will also offer it on the 5 GHz unlicensed spectrum". The end user would be simultaneously connected to both RATs: The LTE-U carrier would be used only for the data plane, "while all the control signaling would happen on the licensed band, where the QoS is ensured". Under this scenario, the user benefits from the seamless mobility and additional spectrum, while the MNO reduces expenditures by using a common infrastructure.

2. Multipath Traffic Aggregation.

   Most modern mobile devices posses multiple network interfaces such as Ethernet, WiFi, 3G, WiMAX, 4G-LTE and Bluetooth. Theoretically, it is possible to use these multiple interfaces simultaneously for data transfer and aggregate the traffic received at the end device. These techniques are also known as "path diversity in IP networks" in the literature [23]. This is fundamentally different from the regular practice today. Observe that, although most of these interfaces are available nowadays in the mobile devices, they are activated on-demand by the user and its utilization is normally restricted to some specific function. For instance, a cellular-network connection is replaced by WiFi as the device broadband connection whenever the latter is available (in order to reduce costs and energy use). Instead, path diversity advocates automated coordination mechanisms that allow to use the different interfaces simultaneously. Considerable benefits are expected in terms of seamless mobility, increased traffic capacity, decreased probability of network outage, ubiquitous network access, and even improved energy efficiency in some circumstances.

   Research work has demonstrated that the most effective techniques for multipath support operate at the Routing and Transport layers [24], so we focus on them in our discussion. Coordinated simultaneous data transport over more than one interface can be achieved using fundamentally three different schemes.

   First, the most straightforward way is to send different IP flows (characterized by the TCP/IP header 5-tuple) over different interfaces.

   Second, Mobile IP mechanisms can be also employed. In the case of MIPv6, for example, each interface of the device can be assigned a care-of-address while another interface is connected to a Foreign Network. In the context of 3GPP networks, Dual Stack Mobile IP (DSMIP) presents two flavors that can be used: MAPCON and IFOM. MAPCON enables multiple packet data network (PDN) connections with more than one access point name (APN); hence, it provides connection management for both APNs (the one for 3GPP access and the one for non-3GPP access). On the other hand, IFOM provides only one PDN connection with the same APN for both access networks and enables a per-IP-flow granularity control. In other words, every IP flow can be seamlessly moved to the 3GPP access network or non-3GPP access network [9].

Thirdly, it is possible to transport a single data stream over multiple interfaces. Most of the current research for this third option is based on the Multipath TCP (MPTCP) protocol [24]. "Each subflow contains a subset of the packets which form the data stream. At the receiving station, these subflows have again to be reassembled before being handed to the application" [25]. It is easy to argue that the increase in complexity is compensated by the robustness obtained against network outages and the gains in terms of traffic capacity. Such gains are observed because of the 'aggregated pipe' bandwidth that is available to *each* TCP connection. Notice that, in the previously discussed options, the goodput of each TCP connection is limited by the bitrate capacity of the individual link by which it is sent.

3. Network Virtualization and Software Defined Networking.

   There is no doubt that NFV and SDN are essential drivers for the next generation of wireless HetNets. NFV enables network functions that were traditionally tied to hardware appliances to run on close-to-cloud-computing infrastructure in a data center. The point is that there will be a degree of reuse of the infrastructure used in commercial cloud. Although the idea of NFV originated with the purpose of reducing costs, many experts agree that the main benefits will be seen in other areas: the elasticity to support network demands, and the ease to quickly create, test and implement new network features without affecting the service of the current users. Thus, it can represent an invaluable competitive advantage for network operators. The concept of SDN is an architectural principle for networks. It is used to refer to a separation of the routing control plane of the network from the forwarding devices. This decoupling allows to centralize the decision engine of the network (that achieves total visibility of the network) and control the forwarding devices though open Application Programming Interfaces (APIs) like Openflow.

Their implementation will affect specially the mobile core network but will certainly extend progressively to the access network edge. Indeed, interesting ideas have come to light in which the principles of SDN have been applied to the RAN. The SoftRAN proposal [26] is one of such. The OpenRadio platform [27], although still in initial stages, is also a very promising proposal as the dual of Openflow for wireless packet forwarding. Virtualization applied to the RAN will allow network slicing and enhanced mobility by the implementation of Virtual Base stations or Virtual Access Points. These virtual nodes will group a series of physical network nodes as a single geographical entity. We agree with Bangerter et al. that the maximum benefit of such advances will be observed in a Multi-RAT environment: “An integrated virtual radio network will enable joint management and simultaneous use of radio resources across different radio technologies to significantly improve radio capacity and enhance coverage and wireless link reliability.” [4]

Of course, such framework will require new ways of connection management by the network. For instance, Soldani and Manzalini [28] introduce the concept of the ‘neural bearer’, which they define as a bearer graph. It is supposed to enable multidimensional carrier-grade communication paths simultaneously through different radio interfaces. This construction is the key element of their vision of what 5G will be. They envision a 5G Operating System (5G OS) that orchestrates a high capacity and diversified RAN infrastructure under a software-defined approach.

4. New RAN architectures.
   Finally, without the possibility of simultaneous multi-RAT connections, innovative architectures for the RAN would not exist or would be very limited.

It is the opinion of [4] and [29] that the evolution of the HetNet architecture with respect to small cells will consist of variations of the 'anchor-boost' design. In this design, the macrocell operates as an 'anchor' base station and is primarily responsible for signaling, while the small cells operate as 'booster' base stations and are mainly responsible for offloading data traffic. This decoupling of signaling (coverage) and data (capacity) can be also exploited in order to increase power efficiency by giving the capability to the anchor node to turn on or off the booster nodes, depending on the demand. A decoupling of routing control and the data-forwarding plane (SDN ideas) can also be added to the picture by making the signaling node also the routing control node. Finally, by assigning interference management tasks to the signaling node we obtain the idea of 'phantom-cells' existing in 3GPP literature.

The gains of these new RAN architectures are expected to be maximized when multiple RATs spanning various standards are considered in the equation. It is likely that new RAN architectures will also take advantage of D2D communications in order to complement the infrastructure-based network. However, the optimal role for D2D communications in HetNets is still an open area of research.

1.3 Chapter summary

Summarizing, ultra-densification and offloading through multi-RAT HetNets will be the key strategy for increased bandwidth capacity in 5G networks. Under this assumption, the user association or RAT selection process is a very important problem with repercussions in network load balancing, seamless mobility and Self-Organizing Networks (SON) functions.

In this survey we presented the state of the art of proposals related to the User Association Problem in 5G and HetNets. We analyzed the main criteria for user association decisions, DL SINR and cell load, and we also discussed the opportunities for exploring additional parameters like user mobility and context-awareness information.

We considered two main categories of user association, single-RAT association and multi-RAT association. In the first case, the simplicity of biasing and policy-based mechanisms make them preferable against other approaches. Additional research is necessary in order to fully characterize biasing under non-simplistic assumptions. It would also be desirable to see novel proposals of real-time policy-based mechanisms that generate low overhead. In the case of multi-RAT association, multipath traffic aggregation, network decoupling and virtualization are still (mostly) theoretical propositions that should be tested extensively for several different settings.

Interest has been growing lately around the idea of the use of cognition and the use of machine learning for enhancing the adaptability and efficiency of the networking tasks [13–15]. Cognition is currently seen as one of the essential enabling technologies behind Self-Organizing Networks. Its applicability seems very promising for both single-RAT and multi-RAT association. To the best of our knowledge, cognition and machine learning techniques are almost completely unexplored areas of research concerning the user association problem.

Chapter 2

Solution Description

2.1 Problem Analysis

Most mobile terminals today support the use of multiple Radio Access Technologies (RATs) (e.g., LTE, WiMAX, Wi-Fi, Bluetooth) through the use of several network interfaces or software defined radio (SDR) technology. By means of a combination of the services enabled by these different RATs, true ubiquity of mobile broadband services is becoming a reality. In 5G HetNets, the variedness and availability of network attachment points through diversified RATs will continue to mature and evolve. This development will turn the classic, and extremely important, user association problem into a complex decision process in order to guarantee efficacy and efficiency of the HetNet architecture.

The conventional mechanism for user association in cellular networks, the max-SINR rule (see subsection 1.2.2), has proven to be insufficient in the context of HetNets [6, 15]. Here, differences in transmission power and coverage of the cells can lead to inefficiency in terms of network load balancing. Multi-parametric optimization solutions have been proposed in the literature to overcome the limitations of the max-SINR RAT selection approach by combining it with other criteria (e.g., cell load). However, the computational complexity of these approaches have made the task very difficult [6]. Furthermore, the associated problem of efficient and practical multi-parametric modeling (representation of the network state) has not been completely solved [11, 13]. This is particularly important because the need for integrating multiple parameters in the optimization of network functionalities seems to be a design requirement for future networks. There is increasing agreement to look at these parameters under the umbrella of context-awareness [4, 13, 15, 17].

Finally, besides efficiently exploiting the multi-RAT (i.e., multi-standards) capabilities of the network nodes and performing context-aware optimization, any suitable RAT selection algorithm in 5G HetNets needs to address the following objectives: Optimization of the network load balancing, enhancement of the user mobility experience, and effective autonomous adaptation to changes in the network while keeping a reduced overhead (see Fig. 1.1).

2.2 Proposed Solution

We propose a distributed cognitive framework for RAT selection for 5G HetNets. Here, the term RAT is used to describe a network attachment point (i.e., a Base Station (BS), Access Point (AP), or another terminal device) for a certain wireless network technology. Each RAT is considered independently by our framework even when several of them are concentrated at a single physical node. Our proposed solution implements machine learning algorithms in order to satisfy the objectives above while providing a meaningful approach for context-aware multi-parametric modeling that echoes the principles of [14]. Our solution advocates the use of cognition at the device-level in order to learn optimal, or at least reasonably well-performing, decision policies based on the experience of the device itself. Cognition is important since there may likely not be a "one-size-fits-all" rule for association.

Figure 2.1 presents an overview of our proposed solution. As observed, it can be segmented into three modules/stages: (1) Learning the user/network state model. (2) Detecting the cognitive states. (3) Learning an effective user association policy. Our final goal is to obtain a decision policy to pick the best action (i.e., to associate with a RAT), given the state of the terminal. For that, we propose to use reinforcement learning (specifically, Q-learning). The set of all possible system states $\mathcal{S}$ is the set of clusters from the first stage, obtained using the X-means algorithm (see Fig. 2.1). The current state s_t is the mapping of a current feature vector observed by the terminal device to one of those clusters. This mapping is done using the k-Nearest Neighbors algorithm. We define the set of actions to be the set of available RATs.

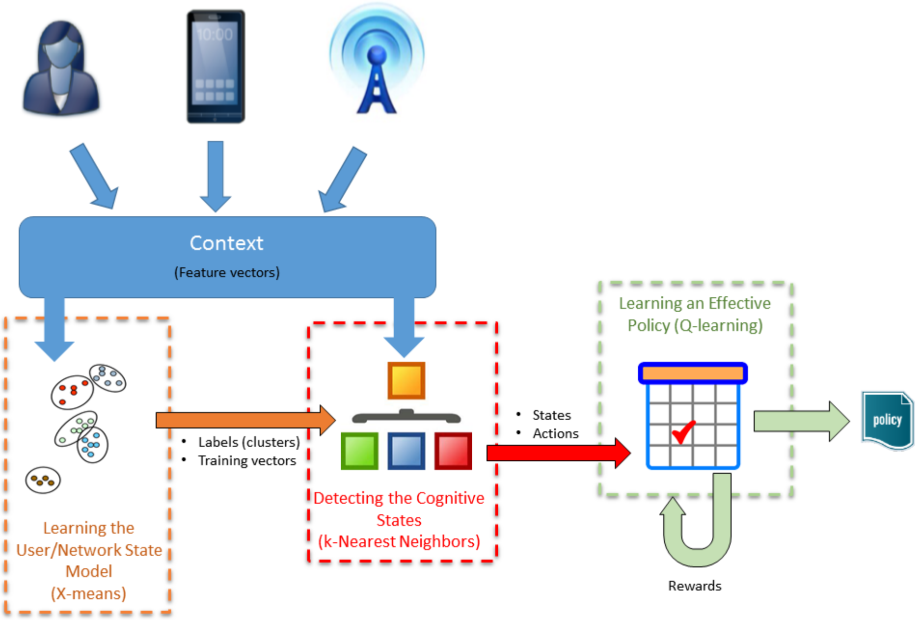

Context
(Feature vectors)
Labels (clusters)
Training vectors
Learning the User/Network State Model (X-means)
Detecting the Cognitive States (k-Nearest Neighbors)
States
Actions
Learning an Effective Policy (Q-learning)
policy
Rewards

Figure 2.1: High level overview of the proposed solution.

2.3 Contributions of the Thesis

The main contributions of this Thesis are: (1) A framework with intrinsic modularity for handling the user association problem. (2) The suggested use of reinforcement learning over supervised machine learning due to its low computational complexity and flexibility for learning an effective user association policy within the large number of diverse situations that a 5G HetNet may represent. (3) The proposition that the use of unsupervised machine learning is crucial for achieving the desired adaptability and practicality in order to formulate the user/network state model and populate the cognitive database. (4) The choice and standardization of a minimal set of recommended descriptors (features) for integrating context-awareness information in the user/network state models. (5) Recommendations for achieving an effective clustering of feature vectors with the desired characteristics for our solution. (6) The exploration of two variants of Q-learning for the reinforcement learning stage.

The novelty in the proposed approach is that the set of states are learned by the device itself and the number of states is also learned rather than pre-specified. This strategy can be really flexible in practice. This implies that a terminal, after collecting data during a training period, may formulate a system characterization and optimize its own association decisions without any external intervention. In the following we will refer to these states as cognitive states, because they are supposed to be learned cognitively by each terminal device.

The following three sections of this Chapter discuss the conceptual foundations of each stage of our solution: Clustering of feature vectors for building a system state model, supervised classification to obtain the current system state, and reinforcement learning for learning good policies, respectively.

2.4 Learning the User/Network State Model

Efficiency of 5G HetNets requires user association mechanisms that consider multiple criteria. This is achieved through "context-awareness", which means that the cognitive entities of the system are aware of the conditions of the network and the user needs and behavior, and are capable of integrating these criteria into the decision-making process. However, formulating system models that are context-aware can be a daunting task, especially for our case of interest. The state definitions must be reasonable across the envisioned heterogeneity and diversity of architectures, communication standards, international regulations, applications and protocols and deployment needs of future 5G networks. Furthermore, representing these states such that the overall processing and memory requirements as well as network overhead remain reasonable is important for efficiency.

A centralized solution that involves bookkeeping all possible states in the core network nodes for all the users and making decisions for them, would demand a considerable computational capacity, even for a relatively small HetNet. Even more concerning is the overhead necessary for updating the states information in real-time if this requires knowledge that is only available or can be estimated at the device-level (e.g., user mobility, proximity of Device-to-Device Communication (D2D) enabled terminals).

Thus, it is desirable, to adopt a distributed design and build meaningful user/network states at the terminal level. As illustrated in Fig 2.1, in the first stage of our framework, we propose clustering of appropriately defined feature vectors as a method for autonomously constructing suitable system models for the user association task. The choice of the standardized fields (or parameters) for formulating the feature vectors is extremely important in order to reduce the dimensionality and ensure significance. This approach allows the system models to be custom-made for each node because the created cognitive states will depend on the specific situations it has experienced. Note also that handling all this information at the device-level dramatically decreases the overhead requirements and leads to low-complexity algorithms that try to optimize the gains of individual user according to its particular needs.

We, however, acknowledge that some network state information that could be crucial for globally optimal user association and should be included in the feature vectors may not easily be perceived, or even inferred, by the terminal device (e.g., base station load and backhaul state). We propose minimal modifications of the standards for allowing real-time broadcasts of this information during broadcast channels transmissions. It can be argued that the amount of radio resources used for transmitting these values is negligible and, as we will see in the following sections, the benefits in terms of Quality of Experience and load balancing surpass the associated technical costs.

2.4.1 The feature vectors.

The user state at any given time n is defined in terms of a collection of parameters that characterizes the mobile device and RAT situation in the network at that particular time. Such parameters can be collected and grouped as a tuple in order to formulate a vector $\mathbf{x}$ of d descriptors or features. Thus, we may represent an observation of the user/network state as a point in a d-dimensional feature space:

$\mathbf{x} = \{x_1, x_2, ..., x_d\}^T$, for $\mathbf{x} \ \epsilon \ \mathbb{R}^d$

Figure 2.2 shows a terminal device that collects feature vectors related to three available RATs. The mobile device needs to collect relevant statistics that reflect both network state information and user needs/behavior information. The network state information could be obtained through Application Programming Interface (API) calls to the available network interfaces and broadcast messages, while estimating the user behavior might require sensing and packet-level analysis with the corresponding data post-processing.

Next, we list a collection of candidate network-level and user-level parameters useful for achieving context-awareness, relevant to the user association problem:

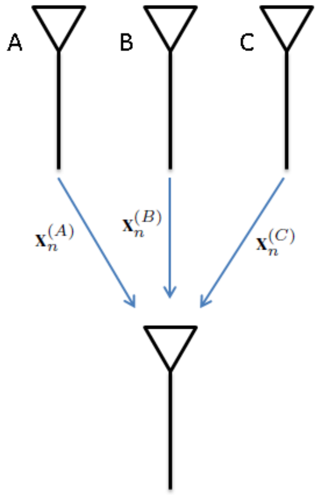


Figure 2.2: At each time n, a client node collects feature vectors related to RATs A, B and C.

- Network-level context:

  1. Tier class (e.g., macro-cell 3G, macro- cell 4G, small-cell 4G, D2D, WiFi).
  2. Peer ID (i.e., Base Station ID (BSID) or MAC Address of the available RAT).
  3. Network carrier frequency.
  4. BS/AP load status.
  5. BS/AP congestion status.
  6. BS/AP backhaul status.
  7. Interference levels.
  8. Uplink communication channel information (e.g., UL-SINR).
  9. Downlink traffic information (e.g., number of connected clients, data packet size, queue length, packet loss rate).

- User-level context:

    1. User profile (e.g., user identity, user preferences on media quality, user activity patterns, user level of distraction).
    2. User billing plan.
    3. Roaming information.
    4. Application (e.g., Skype, Netflix, SAP).
    5. Type of Application (e.g., machine-type data, video streaming, web browser).
    6. Application Quality of Service (QoS) requirements.
    7. Quality of Experience (QoE) metrics.
    8. Downlink communication channel information (e.g., DL-SINR).
    9. Uplink traffic information (e.g., data packet size, packet generation rate, queue length, packet loss rate)
    10. User location: Absolute (e.g., GPS-based), relative (indoor/outdoor/cell-edge), zone (urban/suburban/rural).
    11. User Mobility (i.e., static/mobile).
    12. Device context (e.g., battery state, CPU load, mobile phone characteristics and capabilities).

The following discussion offers a detailed view of the feature vector data specification as proposed in our solution. Four main goals must be achieved with the choice of parameters for constructing the feature vectors:

1. Pertinent and sufficient context-awareness data must be considered in order to achieve load balancing, seamless mobility and Self-Organizing Networks (SON) objectives.
2. Reduced dimensionality of the feature vectors. Using too many features may degrade the performance of clustering and increases the complexity of supervised learning algorithms [30].
3. Ease of data acquisition. The data acquisition task in real world implementations should not require considerable processing.
4. Simplicity, for ensuring clarity while explaining our ideas.

Thus, without compromising essential context-awareness data to achieve the desired goals, we will consider in our analysis the use of the following as the minimal set of descriptors for our feature vectors: (1) Tier class, (2) Peer ID, (3) BS Load, (4) DL SINR, (5) Application Class, and (6) User Mobility. With this minimal set of descriptors we have enough information to uniquely identify the RAT (descriptors 1 and 2), a simple representation of the multi-agent effect on the network condition (descriptor 3), an indicator of the radio link quality (descriptor 4), and a characterization of the user needs and behavior (descriptors 5 and 6). Hence, it is an abbreviated but sufficient definition to achieve our intent.

Tier class

We assume that the network tier (i.e., the class of wireless network technology and cell size) of the active connection or potential connections can be obtained from network broadcast messages or directly from the terminal device's network interfaces. Table 2.1 shows the suggested mapping of values for this parameter. Entries 9-12 are reserved for new 5G standards.

Table 2.1: Suggested mapping of Tier classes.

Value	Network Tier
1	3G - macrocell - EV-DO
2	3G - small cell - EV-DO
3	3G - macrocell - HSPA
4	3G - small cell - HSPA
5	4G - macrocell - FDD - LTE
6	4G - small cell - FDD - LTE
7	4G - macrocell - TDD - LTE
8	4G - small cell - TDD - LTE
9	(Reserved)
10	(Reserved)
11	(Reserved)
12	(Reserved)
13	4G - macrocell - WiMAX
14	4G - small cell - WiMAX
15	Wi-Fi b/g/n (2.5 GHz)
16	Wi-Fi 802.11n (5 GHz)
17	Bluetooth
18	D2D

Peer ID

The Peer ID is a unique identifier of a network attachment point (i.e., a RAT). The RAT, or Peer node, to which the client node is able to connect, could be a base station (BS), an access point (AP) or another terminal device. Usually, the standard-specific identifiers used for base stations are Base Station IDs (BSIDs), while access points and devices are identified by MAC Addresses. However, in our framework we propose the standardization of this descriptor by having the terminal node map each new BSID or MAC Address observed in its collected feature vectors to a positive integer index. We assume that the standard-specific identifiers may be easily obtained from network broadcast messages.

BS Load

The BS Load status is a value that, we propose, can be easily computed by each radio access network node in real-time and transmitted in broadcast messages (broadcast channels or beacon intervals) with negligible cost. For standardizing this parameter, we propose that the result of the computation must be a discrete-valued representation of its time-averaged radio resource utilization (i.e., a percentage of resource blocks or time slots commited for transmission during a time interval). For our analysis, we only consider the BS Load in the downlink transmission (i.e., RAT to terminal device).

The BS Load status is the key parameter for achieving an efficient distributed algorithm for load balancing. In case a node may not be able to compute and broadcast this value, a default value ‘0’ would be used for this parameter, with the subsequent loss of the network load-awareness dimension in the characterization of the corresponding cognitive states.

DL SINR

The traditionally used Downlink Signal to Interference plus Noise Ratio (SINR) is still useful for our model for characterizing the quality and reliability of the communication channel. Slightly different nomenclature, calculation methods and ranges of values of DL SINR are expected from different standards. For example, WiFi (802.11 b/g) defines SNR values on the interval [5,40], while LTE interfaces may compute SINR values on the interval [-5.0,39.38], and WiMAX users may perceive CINR values on the interval [0,40].

These values are usually measured from pilot carriers by the network interfaces, according to the specifications of each communication technology standard. We don't propose any standardization other than recognizing the appropriate nomenclature of the downlink SINR in dB for each standard. These values can be collected from the respective device's network interfaces through API calls. We are assuming that all possible values observed will be contained in the interval [-99,99].

Application Class

Being able to distinguish in real-time the software applications that the end user is running is also an important criterion. From a network performance perspective, and for reasons of scalability and practicality, a model that characterizes the QoS requirements of the applications is more useful than simply identifying the particular protocols being used. Hence, we propose the protocol identification as a first stage and a QoS class mapping as a second stage for formatting the data of this descriptor. Protocol identification can be obtained through packet-level analysis [31] or lightweight deep-packet-inspection (DPI) engines [32, 33] running at the terminal.

Figure 2.3 illustrates our approach for characterizing levels of QoS for different types of applications, according to combinations of throughput requirements and delay tolerance. Table 2.2 summarizes our model, and Table 2.3 provides an example of how the proposed scheme of application classes can be applied to protocols that the deep-packet-inspection engine nDPI can identify [34].

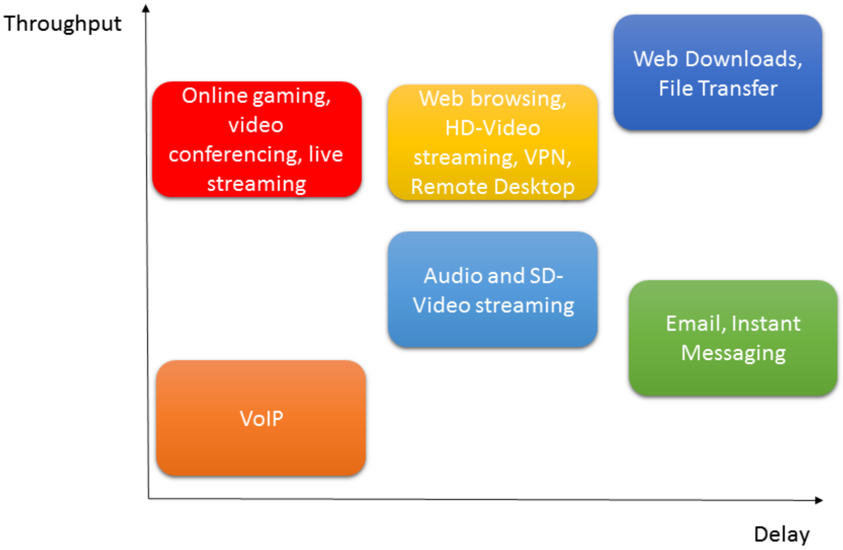

Throughput
Online gaming, video conferencing, live streaming
Web browsing, HD-Video streaming, VPN, Remote Desktop
Web Downloads, File Transfer
Audio and SD-Video streaming
Email, Instant Messaging
VoIP
Delay

Figure 2.3: Regions of QoS requirement of different types of applications.

Table 2.2: Proposed mapping of Application classes.

Application Class	QoS Requirements	Example applications
0	Unknown or unclassified / Non-persistent protocol	DNS, DHCP
1	Moderate throughput, High delay tolerant	Email, Instant Messaging, Sensor Data
2	High throughput, High delay tolerant	Web Downloads, File Transfer
3	Moderate throughput, Moderate delay tolerant	Audio and SD-Video streaming
4	High throughput, Moderate delay tolerant	Web browsing, HD-Video streaming, VPNs, Remote Desktop
5	Low throughput, Low delay tolerant	VoIP
6	Moderate to High throughput, Low delay tolerant	Interactive applications (e.g., online gaming), video conferencing, live streaming

Table 2.3: Example of Application classes based on nDPI supported protocols

	Class 0	**Class 1**	**Class 2**	**Class 3**	**Class 4**	**Class 5**	**Class 6**	**N/A**
TCP/IP Stack Protocols	DNS IPP MDNS NTP NETBIOS ICMP	POP SMTP IMAP	FTP NFS TFTP		HTTP TELNET			
Database			PostgreSQL MySQL TDS msSQL Oracle Redis					
File sharing and P2P	I23V5 Socrates		DirectDownloadLink AppleJuice DirectConnect WinMX PANDO Filetopia iMESH Kontiki OpenFT Kazaa/Fasttrack Gnutella eDonkey Bittorrent OFF Soulseek AFP StealthNet Aimini DropBox Apple iCloud WindowsUpdate RSYNC UbuntuONE Microsoft cloud					
Instant Messaging		QQ GaduGadu IRC Popo Jabber MSN Oscar MEEBO WhatsApp	Snapchat					
Network Management	STUN Whois-DAS	Collectd OpenSignal						NetFlow_IPFIX sFlow
Authentication	Kerberos LDAP Radius							
Other	Fiesta Florensia MOVE USENET DCERPC Corba ZeroMQ Simet VMware	HTTP Actsync						

	Class 0	**Class 1**	**Class 2**	**Class 3**	**Class 4**	**Class** 5	**Class 6**	**N/A**
Routing, Tunneling and Service Discovery	SSDP IGMP IGMP SCTP OSPF IP in IP Teredo				PPTP IPSEC GRE SSL OpenVPN CiscoVPN TOR HotspotShield VPN			BGP VRRP EGP
Web		Twitter Gmail	Instagram		Yahoo FaceBook Google Maps Google SSL over HTTP HTTP Proxy Citrix CNN Wikipedia 99Taxi			
Remote Support					RDP VNC PCAnywhere SSH TeamViewer			
Streaming	QUIC			OGG Icecast PPLive PPStream Zatoo Apple iTunes Spotify GloboTV Deezer	AVI Flash MPEG QuickTime RealMedia Windowsmedia MMS Youtube Netflix RTSP			
VoIP and videoconferencing						RTP MGCP IAX Skype Viber CiscoSkinny Megaco WhatsApp Voice KakaoTalk NOE RTCP	CitrixOnline Apple (Facetime) Webex H323	
Online Gaming	Feidian				Twitch	TeamSpeak	XBOX Battlefield Quake Steam Halflife2 WorldofWarcraft Guildwars MapleStory WARCRAFT3 WorldofKungFu Starcraft	

Finally, it is important to mention that our feature vector must only contain the highest class of all the detected applications, thereby characterizing the most stringent QoS demands of the user as its behavior. For example, if our packet-level or DPI analysis shows that both FTP and VoIP are being used simultaneously by the user, the Application Class descriptor would only reflect the class of the most sensitive application (in this case VoIP); therefore, the descriptor value would be ‘5’.

User Mobility

The User Mobility status could be inferred through GPS-based calculations or any of the other methods mentioned in subsection 1.2.2. This parameter is the key descriptor for enhancing the mobility experience for the end user. We propose a binary descriptor; hence, characterizing the user as mobile if it shows the value ‘1’, or as static if it shows the value ‘0’. Extending this definition to accommodate for several levels of mobility (e.g., static, walking speed, vehicle speed, very high speed) is straightforward if positive integer values are assumed for representing each level.

2.4.2 Clustering

The role of clustering in our framework

From a machine learning perspective, the feature vectors are simply patterns. These patterns can be divided into a set $\mathcal{S} = \{1, ..., K\}$ of K groups of similar characteristics or clusters, based on some measure of similarity. In our framework, these resulting clusters represent different cognitive states the terminal device can be in. Note that these states are derived from the past data observed by the terminal device. Thus, the proposed approach builds a system model relying on the multi-parametric context information contained in the feature vectors.

This idea of multi-parameter cognition (based on unsupervised machine learning techniques) for modeling and data representation is not new [14]. Zorzi et al. indicate that "... when applied to the real-world problems, unsupervised learning can exploit the huge amount of data that comes without any label to build rich internal representations [of the sensory world]" and relate this idea with the notion of generative models and representation learning [13]. Surprisingly, to the best of our knowledge, this approach, however, had not yet been applied in the context of the user association problem.

In our framework, we use the X-means algorithm to generate the cognitive states. X-means is a clustering algorithm with relatively low complexity that, in addition to classifying data into a set of clusters, attempts to estimate de number of clusters K from the data itself. In our proposed solution, X-means runs off-line, utilizing the training feature vectors collected from all the different RATs. We have observed in our tests with X-means, that under certain conditions, the resulting state space $\mathcal{S}$ is clearly segmented into subsets of clusters, each associated with a particular RAT, as illustrated in Fig. 2.4. This reveals an effective characterization of distinct cognitive states. In other words, if we assume that there are only 3 RATs available (see Fig. 2.2), without loss of generality, we may formalize our definition of a cognitive state s as follows:

$$s \in \mathcal{S} = \{s_1^{(A)}, \cdots, s_{K_1}^{(A)}, s_1^{(B)}, \cdots, s_{K_2}^{(B)}, s_1^{(C)}, \cdots, s_{K_3}^{(C)}\},$$

where $K_1 + K_2 + K_3 = K$ is the total number of cognitive states, and each element in the set $\mathcal{S}$ represents a cluster of feature vectors associated with one of the available RATs (in this case, A, B, or C).

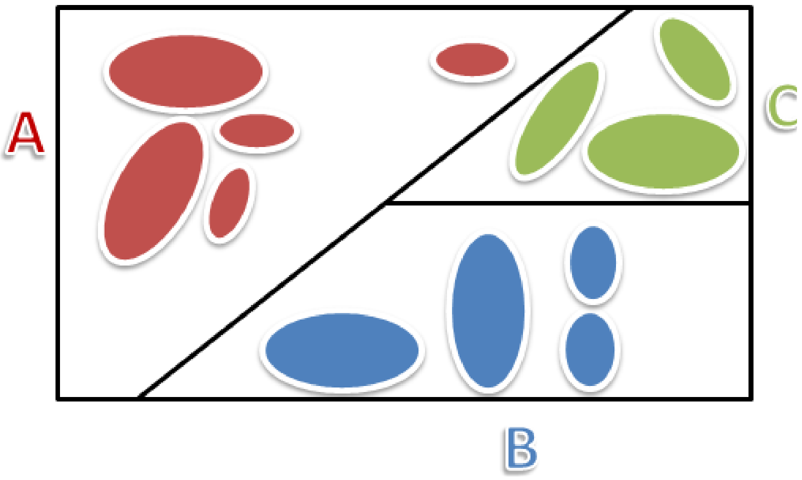


Figure 2.4: The output of X-means is a state space with clearly differentiated regions corresponding to the different RATs observed. Each region may contain multiple clusters (i.e., multiple cognitive states).

The X-means algorithm

X-means is a variation of the K-means clustering algorithm. Both algorithms assume underlying spherical Gaussian likelihoods. For the K-means algorithm we need to specify the desired number of clusters K in which the data would be divided, while X-means attempts to estimate it, within a possible range of values, from the data itself.

Assume that we have a sequence of N training feature vectors $\{\mathbf{x}_i\}_{i=1}^{N}$ collected by the terminal device from all the different RATs observed in the past. While explaining the X-means algorithm, we will drop momentarily the notation that reflects the association of a cluster with a given RAT,
$s \in \mathcal{S} = \{s_1^{(A)}, \cdots, s_{K_1}^{(A)}, s_1^{(B)}, \cdots, s_{K_2}^{(B)}, s_1^{(C)}, \cdots, s_{K_3}^{(C)}\}$ from above, and use
$s \in \mathcal{S} = \{s_1, s_2, ..., s_K\}$, instead. Still, each element in the set $\mathcal{S}$ represents a cluster of feature vectors, interpreted as a cognitive state. Let $\mathcal{C} = \{\mu_1, \mu_2, ..., \mu_K\}$ denote a set of centroid vectors with cardinality K. Each centroid vector $\mu_m \in \mathcal{C}$ is associated

to a particular cluster $s_m \in \mathcal{S}$ for $m \in \{1, ..., K\}$.

The K-means algorithm tackles the clustering task as a minimization of the following distortion measure [35], the distance of each feature vector $\mathbf{x}_i$ assigned to a cluster s_m from its associated centroid vector μ_m:

$$J(r, \mathcal{C}) := \frac{1}{2}\sum_{i=1}^{N}\sum_{m=1}^{K} r_{ij}||\mathbf{x}_i - \mu_m||^2,$$

where $r = \{r_{ij}\}$ and r_{ij} is an indicator, which is 1 if, and only if, $\mathbf{x}_i$ is assigned to cluster s_m; and, $||\cdot||^2$ denotes squared Euclidian distance.

This joint minimization of J with respect to r and $\mathcal{C}$ is performed in a two-step process (see Algorithm 1): (1) Each vector $\mathbf{x}_i$ is assigned to the cluster s_m with the closest associated centroid vector μ_m. (2) Each centroid vector μ_m is recomputed based on the feature vectors assigned to the associated cluster s_m in the previous step. The algorithm iterates until the cluster assignments or the centroid locations do not change significantly.

The X-means algorithm deals with two problems simultaneously: (1) It needs to figure the right number of clusters and (2) classify the observations to the different clusters. "The X-means algorithm treats the problem of unknown number of clusters as a model selection problem" [36] because the collected feature vectors can be seen as generated by a given unknown model with K parameters. In general, the model selection problem is to choose the model that better explains the given data while avoiding overfitting. Different criterion can be used to manage this tradeoff. The most widely used criteria include the "Akaike information criterion" (AIC) and the "Bayesian information criterion" (BIC). The original X-means algorithm [37] makes use of the BIC [38]. The interested reader is referred to our sources for an extensive discussion. For the second problem, the X-means algorithm relies on the K-means algorithm.

X-means will try to estimate the number of clusters K within the range $K_{min} \leq K \leq K_{max}$, where the values $K_{min}, K_{max} \in \{1, 2, ..., N\}$, need to be specified by the user. In each iteration the X-means runs the K-means for a given number of clusters. Next, it splits each of the clusters into two subclusters and runs the K-means again with $K = 2$ within the region of each parent cluster. Then, it compares if there was any improvement in the model by making the new partitioning. In the case(s) in which there was no improvement by partitioning, the parent cluster is preserved. Otherwise, the value of K is updated. If the current number of clusters $K > K_{max}$ or the cluster assignment did not change significantly from the last iteration, the algorithm will stop. Detailed pseudo-code can be found in Algorithm 1.

Algorithm 1 X-means algorithm

```
 1: procedure K-MEANS ALGORITHM
 2:     Initialize the cluster centroids $\mu_1, \mu_2, ..., \mu_K$ to random values.
 3:     for $m = 1, ..., K$ do
 4:         Assign $\mathbf{x}_i$ to cluster $s_{\bar{m}}$ if $\bar{m} = \underset{m}{\text{argmin}}||\mathbf{x}_i - \mu_m||^2$
 5:     end for
 6:     for $m = 1, ..., K$ do
 7:         Set centroid $\mu_m$ as a sample approximation: $\mu_m = \frac{1}{|s_m|} \sum_{\mathbf{x} \epsilon s_m} \mathbf{x}_i$, where $|s_m|$
    denotes the cardinality of the cluster $s_m$.
 8:     end for
 9:     if convergence condition is not satisfied then
10:         Go to step (3).
11:     else
12:         Stop.
13:     end if
14: end procedure
15:
16: Define $K_{min}$ and $K_{max}$.
17: Initialize $K = K_{min}$
18: Set $\hat{K} = K$.
19: Run K-means algorithm procedure for current value of K.
20: for $m = 1, ..., K$ do
21:     Replace the centroid $\mu_m$ by two centroids $\mu_{m_{(1)}}$ and $\mu_{m_{(2)}}$ by moving the original
    centroid $\mu_m$ along a random vector proportionally to the cluster size.
22:     Run K-means algorithm with $K = 2$ over the cluster $s_m$.
23:     Replace or retain the parent centroid based on the model selection criterion (e.g.,
    AIC or BIC).
24: end for
25: Update $\hat{K}$ according to the decisions to replace or retain the parent centroids.
26: if convergence condition is not satisfied and $\hat{K} \leq K_{max}$ then
27:     Set $K = \hat{K}$.
28:     Go to step (19).
29: else
30:     Stop.
31: end if
```

2.5 Detecting the Cognitive States

2.5.1 The role of supervised classification in our framework

At time t, each device must determine the best RAT association based on an observed feature vector $\mathbf{x}_t$. Although there exist many supervised classification techniques to classify a new feature vector $\mathbf{x}_t$ into one of the cognitive states generated by the X-means algorithm in the previous section, we advocate for the use of possibly the simplest approach, the k-Nearest Neighbors (kNN) algorithm.

As depicted in Fig. 2.1, in our proposed approach, the training vectors and classes are supplied by the clustering stage preceeding the kNN classifier. Let $\mathbf{y}_t$ be a tuple of feature vectors collected at decision time t by the terminal device:

$$\mathbf{y}_t := [\mathbf{x}_t^{(A)}, \mathbf{x}_t^{(B)}, \cdots]^T,$$

where $\mathbf{x}_t^{(\cdot)}$ denotes a feature vector collected at time t by the terminal device, associated to an available RAT.

The classification rule $f : \mathbf{x}_t^{(\cdot)} \to s^{(\cdot)} \in \mathcal{S}$, is a deterministic function, that maps each feature vector in $\mathbf{y}_t$ to a cognitive state (i.e., the clusters). Hence, the output of the classifier is essentially the set of possible cognitive states the device may reach at the next time instant.

2.5.2 The k-Nearest Neighbors algorithm

Supervised machine learning techniques can be used to build classifiers that use examples properly labeled by a “supervisor” as ground truth. Based on these examples, the classifier infers a decision rule to classify a new feature vector. Some well-known supervised learning classifiers found in literature are Support Vector Machines (SVM), Multilayered Feed-forward Artificial Neural Networks and Decision Trees [13, 30].

The kNN rule is a simple supervised learning algorithm that assigns the new feature vector $\mathbf{x}_t$ to the m-th class, where $m \in \{1, ..., K\}$, to which the majority of the k closest training feature vector(s) belong.

According to [39], the performance of this algorithm is somewhat surprising, even in its most basic form. For the case when $k = 1$, as the amount of training vectors N tends to infinity, it is possible to lower bound its performance to twice the error rate of the Optimal Bayesian classifier: “Using only random labeled examples but knowing nothing about the underlying distributions, we can (in the limit) achieve an error rate no worse than twice the error rate that could be achieved by knowing everything about the probability distributions. Moreover, we can do this with an extremely simple rule that bases its decision on just the nearest neighbor to the feature vector we wish to classify.” As the value of k grows, however, there is no guarantee that the performance will be better. The result will depend on the underlying distributions. The authors suggest a rule of thumb for implementation: by choosing $k = h(N)$, such that $h(N) \xrightarrow{N\to\infty} \infty$ and $\frac{h(N)}{N} \xrightarrow{N\to\infty} 0$ (e.g., choosing $h(N) = \sqrt{N}$), the performance of the kNN classifier approaches that of the Bayesian decision rule.

2.6 Machine Learning based User Association Policy

2.6.1 States and actions.

We may assume that the terminal device is able to recognize the cognitive state that corresponds to the currently active RAT association out of all computed classifier decisions from the second stage of our framework (see subsection 2.5.1) and denote it by $s_t \in \mathcal{S}$. Let $\mathcal{A}_t$ denote the set of actions available to the device at time t, where actions $a_t \in \mathcal{A}_t$ are identified as the available RATs. Choosing an action implies enforcing standard-specific procedures (i.e., network entry, handover, etc.) necessary for the new association. Notice that in our framework, the set of states reachable by the device at time $t+1$, denoted by $\mathcal{S}_{t+1}$, depends on the chosen action at time t. Continuing with the 3-RAT example of Figs. 2.2 and 2.4, assume that the state of the terminal at time t is $s_t = s_2^{(A)}$ and the available action set is $\mathcal{A}_t = \{A, B, C\}$. Then,

$$\begin{aligned}
&\text{if } a_t = A, \text{ then } \mathcal{S}_{t+1} = \{s_1^{(A)}, s_2^{(A)}, \cdots, s_{K_1}^{(A)}\}\\
&\text{if } a_t = B, \text{ then } \mathcal{S}_{t+1} = \{s_1^{(B)}, s_2^{(B)}, \cdots, s_{K_2}^{(B)}\}\\
&\text{if } a_t = C, \text{ then } \mathcal{S}_{t+1} = \{s_1^{(C)}, s_2^{(C)}, \cdots, s_{K_3}^{(C)}\}.
\end{aligned}$$

Although it is not our main interest in this Thesis, we may argue that this idea can be easily extended to traffic flow association within the context of simultaneous multi-RAT connections. In that case, choosing an action would imply enforcing routing decisions for IP flows.

2.6.2 Reinforcement learning and the reward function.

Reinforcement learning refers to a category of unsupervised machine learning techniques useful for learning an effective sequence of actions (i.e., a policy) from experience in order to achieve a goal. In reinforcement learning, a decision-making agent receives a reward (i.e., a feedback) based on the action it chooses. This reward informs the agent of the "goodness" of the action. In selecting the next actions, the agent tries to find a balance between exploration (choose untested actions) and exploitation (selection of actions already identified as beneficial) in order to reach at a globally optimal policy for maximizing the rewards.

Reinforcement learning has been proposed for achieving autonomous behavior in cognitive architectures (cognitive radios and cognitive networks) and as an enabling technology for Self-Organizing Networks [13, 40, 41]. Also, it has been claimed to be robust against noise [42], to outperform heuristic algorithms [43] and to have acceptable performance in multi-agent (decentralized) environments [40, 44].

A reward function defines the goal in a reinforcement learning problem. The rewards determine the immediate desirability of the system states. Next we present a list of network performance metrics that, in general, could be used for specifying a reward function for learning a good RAT selection policy. We have categorized them guided by the ideas present in [45]:

- Quality of Service (QoS).

  1. Downlink/Uplink Throughput (min/avg/max).
  2. HTTP and RTP Round Trip Time (RTT) Delay.
  3. Jitter.
  4. TCP performance (retransmissions, duplicated ACKs, etc.).
  5. Packet loss rate.

- Cost.

  1. Amount of transmissions necessary to achieve a specific goal.
  2. Cost of actions performed (e.g., cost of interrupting IP flows).
  3. Signaling overhead.
  4. Resource consumption (e.g., battery power)
  5. Handover rate.

- Flexibility/Reaction Time.

  1. Mode change time (i.e., time it takes to associate to a certain RAT).

- Failure avoidance (robustness).

  1. Mean time before failure.
  2. Attach success rate.
  3. Application connection success rate.
  4. Call drop rate.
  5. Incomplete handover rate.
  6. Packet retransmission rate (at both MAC and Transport layers).

Since the user association problem requires multi-objective optimization that jointly maximizes the user perceived average throughput and QoS while minimizing service interruptions due to mobility conditions, in our proposal we define the reward function as

$$R_t(s_{t-1}, s_t, a_{t-1}) = r_t \cdot U(r_t), \tag{2.1}$$

and

$$\begin{aligned} r_t = \; & \beta \cdot g(\text{Avg\_Measured\_Throughput}) - \lambda \cdot h(\text{HTTP\_RTP\_RTT\_Delay}) \\ & - \zeta \cdot c(\text{Handovers\_Calldrops}), \end{aligned}$$

where:

(1) $R_t(s_{t-1}, s_t, a_{t-1})$ is the delayed reward function computed at instant t that evaluates the consequences of the action a_{t-1} taken at instant $t-1$ while in state s_{t-1} that led to the current state s_t.

(2) $U(\cdot)$ is the Heaviside step function, for ensuring that $R_t(\cdot)$ is non-negative.

(3) β, λ and ζ are arbitrary coefficients defined on the interval [0,1].

(4) $g(\cdot)$, $h(\cdot)$ and $c(\cdot)$ are suitably defined non-decreasing reward and cost functions of network performance metrics. $g(\cdot)$ and $c(\cdot)$ are also restricted to be non-negative.

By using actual measurements in computing these rewards, the optimization is performed over a more accurate characterization of the network behavior.

2.6.3 The Q-learning algorithm.

As shown in Fig. 2.1 and mentioned in our introduction, the third stage of our proposed framework relies on the model generated in the first stage to learn a good association policy. We propose the use of the Q-learning algorithm for this reinforcement learning stage. The Q-learning algorithm maintains a table (known as the Q-table) of values that represent a quantification of the goodness of taking a particular action when in a given state. Each table entry, $Q(s_t; a_t)$ is associated with a state-action pair, where $s_t \in \mathcal{S}$ and $a_t \in \mathcal{A}_t$.

In our case, each Q-value is a measure of "quality" of switching the currently active RAT association to either a different RAT or keeping it unchanged. Note that, provided we defined the states as multi-parametric representations, the Q-Learning algorithm decisions are inherently context-aware.

The Q-values are updated as:

$$Q(s_{t-1}, a_{t-1}) \leftarrow (1-\alpha)Q(s_{t-1}, a_{t-1}) + \alpha[R_t(s_{t-1}, s_t, a_{t-1}) + \gamma \max_{a_t} Q(s_t, a_t)] \quad (2.2)$$

where:

(1) $Q(s_{t-1}, a_{t-1})$ denotes the Q-table entry defined by taking action a_{t-1} while being at state s_{t-1} at decision instant $t-1$.
(2) $\alpha : 0 < \alpha < 1$ is called the learning rate.
(3) $\gamma : 0 \leq \gamma < 1$ is called the discount factor.

Once the Q-table is learned, the actions are selected according to:

$$a_t^* = \begin{cases} \underset{a_t}{\operatorname{argmax}}\, Q(s_t, a_t) & \text{with probability } 1 - \epsilon \\ P(\mathcal{A}_t) & \text{with probability } \epsilon \end{cases}, \tag{2.3}$$

where

(1) $\epsilon : 0 \leq \epsilon \leq 1$ is called the exploration rate.

(2) $P(\mathcal{A}_t)$ is some probability distribution over the set of actions $\mathcal{A}_t$ defined using heuristics for exploration purposes.

2.6.4 A modified Q-learning for learning afterstate value functions.

The Q-value update equation (2.2) updates state-action pairs in order to approximate an optimal action-value function. However, in our research we also explored the strategy of modifying the Q-learning algorithm in order to learn *afterstate*-value functions. Here, we adopt the notion of afterstates coined by Sutton and Barto in [46]. An afterstate reflects the system conditions immediately after the agent has made a decision. This may be different from the conventional state, co-produced by underlying random processes, observed at the next decision instant. “Afterstates are useful when we have knowledge of an initial part of the environment’s dynamics but not necessarily of the full dynamics. For example, in games we typically know the immediate effects of our moves. We know for each possible chess move what the resulting position will be, but not how our opponent will reply” [46]. Afterstate value functions take advantage of this kind of knowledge and may produce more efficient learning.

In our case, an afterstate would be defined as the cognitive state a terminal device would be in immediately after making a RAT association decision. Notice that this may be different from the cognitive state reached by the device in the next decision interval. The latter, although associated to the same RAT as the afterstate, will also depend on the environment dynamics. Let $\hat{s}_{t+1} \in \mathcal{S}$ denote the afterstate reached immediately as a consequence of making decision a_t at decision interval t. We may consider the afterstate $\hat{s}_{t+1}$ as an estimation of the cognitive state s_{t+1} that will be observed at the next decision instant $t+1$.

In our proposed framework, at every decision interval t, the output of the kNN classifier maps each feature vector $\mathbf{x}_t$ in the observation $\mathbf{y}_t$ to a cognitive state $s^{(\cdot)}$ (refer to Fig. 2.1 and to subsection 2.5.1). Let $\hat{\mathcal{S}}_t \subset \mathcal{S}$ denote the set resulting from these mappings containing the candidate afterstates. The actual afterstate will depend on the chosen action at each decision instant t. Figure 2.5 illustrates the elements of the Decision Process considered in the present discussion for a single decision interval.

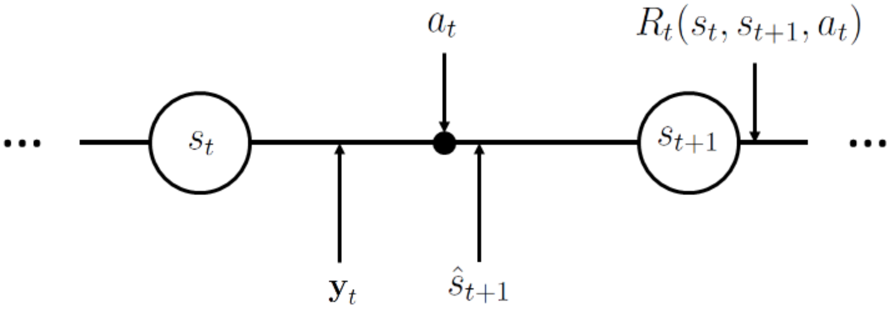


Figure 2.5: Elements of the Decision Process considering an observation $\mathbf{y}_t$ and an afterstate $\hat{s}_{t+1}$.

Each Q-table entry in this modified version of Q-learning is, thus, defined as a state-afterstate pair. The intention behind this approach is to take advantage of the multi-parametric context embedded in both the state and afterstate information for optimizing the decisions. In other words, our Reinforcement learning mechanism may avoid selecting a RAT that is predicted to provide low rewards if it is able to estimate its future state when making the association decisions. This approach may provide performance gains over the conventional Q-learning algorithm.

In this case, the Q-table entries are updated as:

$$Q(s_{t-1}, \hat{s}_t) \leftarrow (1-\alpha)Q(s_{t-1}, \hat{s}_t) + \alpha[R_t(s_{t-1}, s_t, a_{t-1}) + \gamma \max_{\hat{s}} Q(s_t, \hat{s})],$$

where $Q(s_{t-1}, \hat{s}_t)$ denotes the Q-table entry in the modified version of Q-learning, defined by reaching the afterstate $\hat{s}_t$ after taking action a_{t-1} while being at state s_{t-1} at decision instant $t-1$.

And, the actions are selected as follows:

$$a_t^* = \begin{cases} \underset{a_t}{\text{argmax}}\ Q(s_t, \hat{s}_{t+1}(a_t)) & \text{with probability } 1-\epsilon \\ P(\mathcal{A}_t) & \text{with probability } \epsilon \end{cases},$$

where the notation $\hat{s}_{t+1}(a_t)$ emphasizes the afterstate as a function of an available action a_t at time t.

We expect that, as long as the afterstates $\hat{s}_{t+1}$ are a good estimation of the future states s_{t+1}, this modified version may provide better performance than the regular Q-learning presented in the previous subsection. The accuracy of the estimation will depend on how fast the network conditions and user behavior might change and on the amount of cognitive states learned for every RAT.

Chapter 3

Clustering simulations

3.1 Introduction

Clustering is the means by which, we propose, a multi-parametric system state model can be created cognitively in our framework. The terminal device builds such a model based on its particular observations. The clustering operation generates two important outputs:

1. A set $\mathcal{S}$ of clusters of feature vectors observed in the past. As explained in the previous chapter, this set is used in the reinforcement learning stage as the set of cognitive states.
2. A set $\mathcal{C}$ of cluster centroids. The centroids are a practical (in terms of a reduced cardinality compared to the set of observations) and meaningful representations of the cognitive states. This set of cluster centroids has tremendous relevance for our analysis in this chapter. As we will see, the location of the centroids allows us to evaluate the performance of the clustering operation according to the needs of our proposed framework.

This chapter presents our testing procedure and observations of the X-means clustering algorithm using simulated data for our proposed solution. We implemented K-means and X-means in MATLAB using *squared Euclidian distance* as a distance measure. The *Bayesian Information Criterion (BIC)* [38] was chosen as the model selection criterion for X-means.

A detailed report on our initial executions of the X-means algorithm using 6-dimensional data vectors has not been included in this Chapter. However, the results demonstrated that fine-tuning was necessary. Indeed, we are working on a hard classification problem in which using all features simultaneously without any adjustment of the data produced misclassification. Thus, we decided to reformulate our testing methodology: We studied and tested the clustering of the available data of each feature separately, adjusting parameters to optimize the individual outcomes, as necessary. After acceptable results were obtained, we proceeded by increasing the dimensionality of the data points by combining features, and continued testing. The adjustable parameters of the K-means and X-means algorithms are: K_{min}, the initial number of clusters; K_{max}, the maximum number of clusters allowed; and ε_K and ε_X, the convergence tolerances for K-means and X-means, respectively. For all our tests we have configured $\varepsilon_K = \varepsilon_X = 0.001$.

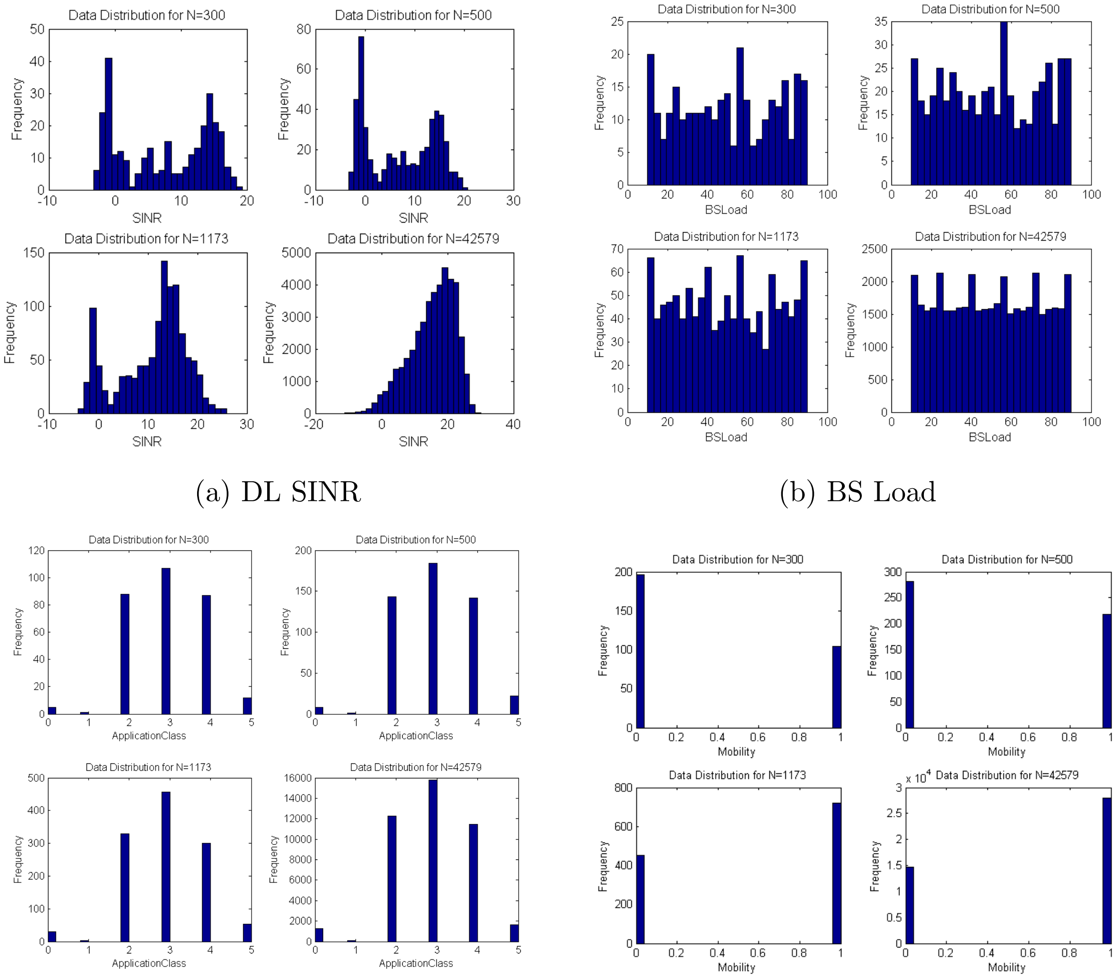
The following sections present the characteristics of the data used for testing and discuss the results of one-dimensional, two-dimensional and three-dimensional clustering operations by combining several descriptors of our proposed feature vectors. The test reports have been organized chronologically and the adjustments of parameters at each stage of our tests has been commented.

3.2 Testing Data Set

For our tests we only consider 5 out of the 6 proposed descriptors for our feature vectors. Our testing data set contains a total of 42,579 samples of DL SINR measurements, associated to different Peer IDs, acquired from an operational LTE network. The data for the other features was simulated or derived as follows:

1. Application Class: This data was simulated using a 7-sided coin flip in conformity with the distribution of protocols presented in [47] and then classified according to the scheme proposed in the previous chapter.
2. BS Load: This data was simulated assuming *uniform distribution* over the set $\{10, 11, ..., 90\}$.
3. User Mobility: Based on the GPS coordinates associated with the DL SINR measurements, we were able to derive the speed at which the device was moving and assigned the state '1' to a sample that was taken while moving at at least 10 km/h, and the state '0', otherwise.

The individual distribution of the DL SINR, BS Load, Application Class and User Mobility data are showed in Fig. 3.1 for different values of sample size N. It is important to mention that, for $N = \{1, ..., 1173\}$, the samples are associated with a single Peer ID.



(a) DL SINR

(b) BS Load

(c) Application Class

(d) User Mobility (based on GPS data)

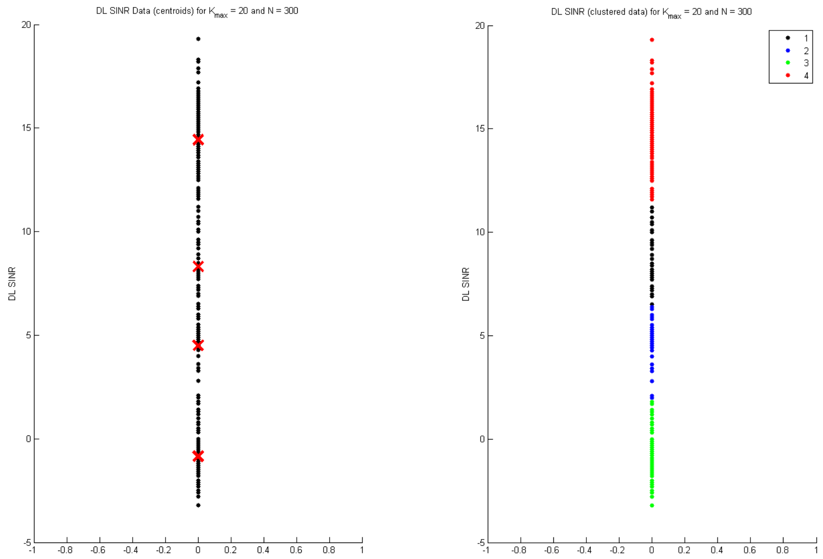
Figure 3.1: Distribution of the Testing Data Set for $N = 300, 500, 1173, 42579$.

3.3 One-dimensional clustering

In this section we consider the clustering of DL SINR, BS Load and User Mobility data individually. We evaluate the output of X-means visually. Unless otherwise specified, the location of the centroids over the original data are shown in the subfigure (a), while in subfigure (b), colors are used for displaying the data according to their associated clusters.

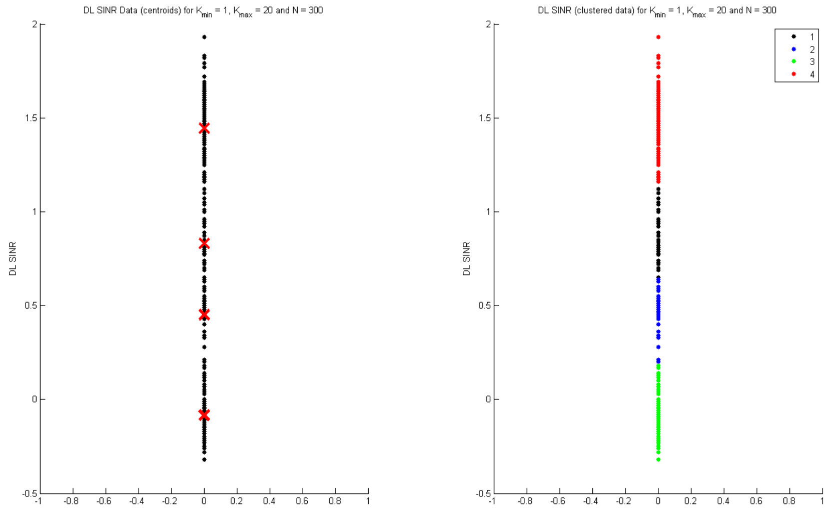
3.3.1 Downlink SINR data clustering

Our first step was to verify that a correct clustering was taking place considering a single descriptor from our feature vectors, the DL SINR. In all the experiments of the present subsection we considered $K_{min} = 1$. Figure 3.2 shows a visualization the output of the X-means algorithm. From Fig. 3.3 we observe that a simple scaling does not affect the clustering results. However, the clustering results changed considerably when increasing the sample size. As verified in Fig. 3.4, by accounting for 500 samples instead of 300, X-means determined that 15 clusters were present, instead of 4.



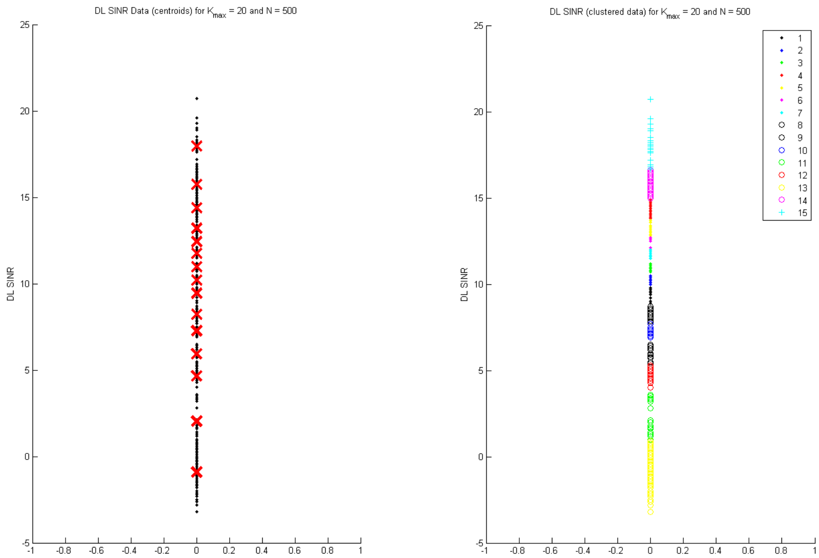
(a) Centroids (b) Clusters

Figure 3.2: Clustering of DL SINR values for $K_{max} = 20$ considering $N = 300$ samples.



(a) Centroids (b) Clusters

Figure 3.3: Clustering of DL SINR values scaled by a factor of 0.1 for $K_{max} = 20$ considering $N = 300$ samples.



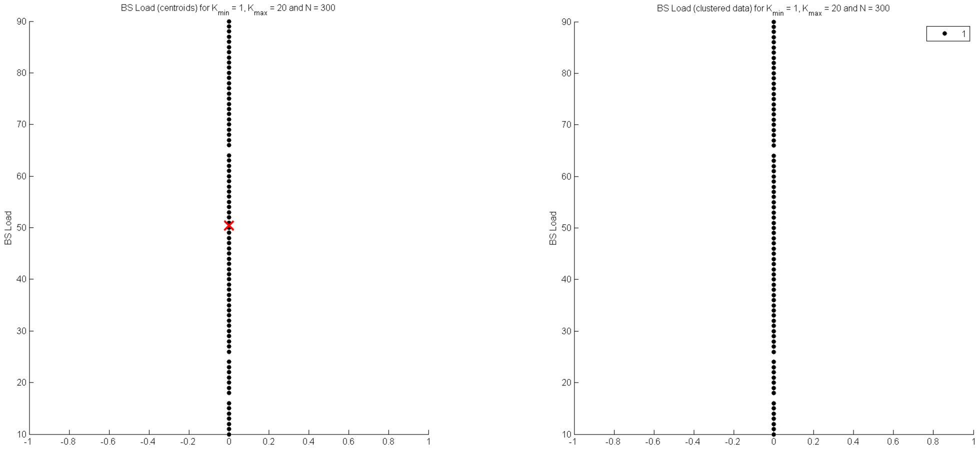
(a) Centroids

(b) Clusters

Figure 3.4: Clustering of DL SINR values for $K_{max} = 20$ considering $N = 500$ samples.

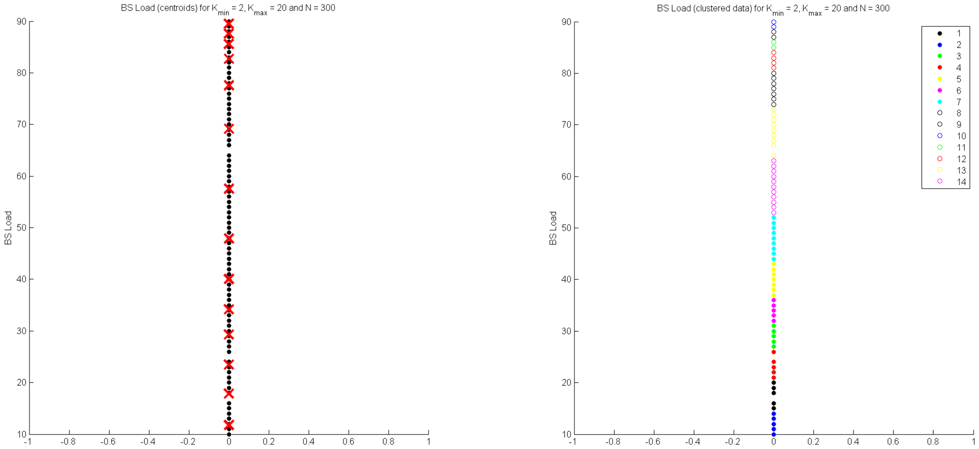
3.3.2 BS Load data clustering

We had to set at least $K_{min} = 2$ to obtain multiple clusters at the output of the X-means. This is shown in Figs. 3.5 and 3.6.



(a) Centroids (b) Clusters

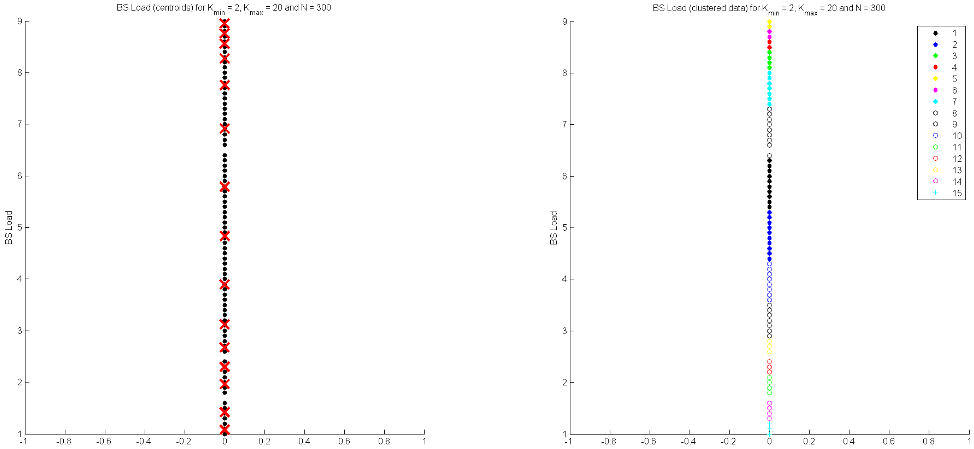
Figure 3.5: Clustering of BS Load values for $K_{min} = 1$, $K_{max} = 20$ considering $N = 300$ samples.



(a) Centroids (b) Clusters

Figure 3.6: Clustering of BS Load values for $K_{min} = 2$, $K_{max} = 20$ considering $N = 300$ samples.

Similar to the DL SINR case, scaling does not affect significantly the output of the X-means algorithm:



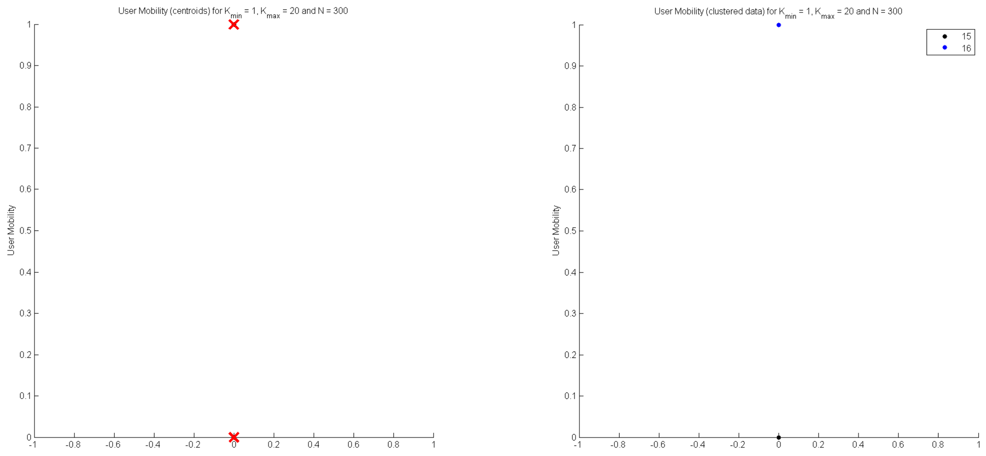
(a) Centroids

(b) Clusters

Figure 3.7: Clustering of BS Load values scaled by a factor of 0.1 for $K_{min} = 2$, $K_{max} = 20$ considering $N = 300$ samples.

3.3.3 User Mobility data clustering

As expected, this is a trivial case for the clustering algorithm.



(a) Centroids

(b) Clusters

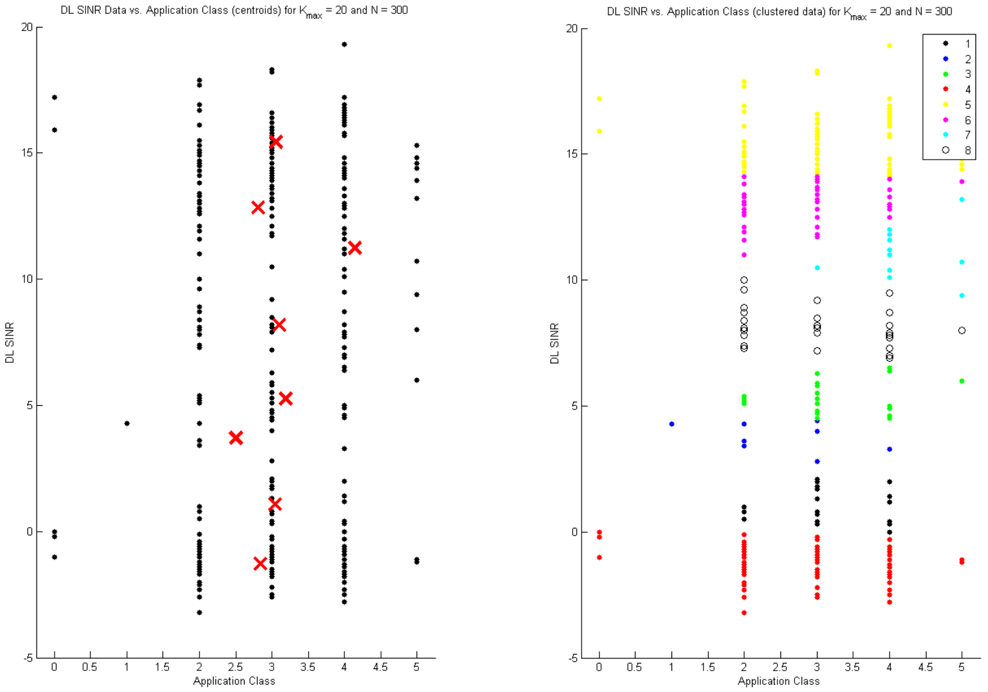
Figure 3.8: Clustering of User Mobility values for $K_{min} = 1$, $K_{max} = 20$ considering $N = 300$ samples.

3.4 Bi-dimensional data clustering

Our next step was, then, to observe the results of clustering bi-dimensional vectors in which we combined each of the descriptors analyzed in the previous section with the Application Class data. Also, the interesting cases of BS Load vs. DL SINR and Peer ID vs. User Mobility were considered.

3.4.1 Downlink SINR vs. Application Class data clustering

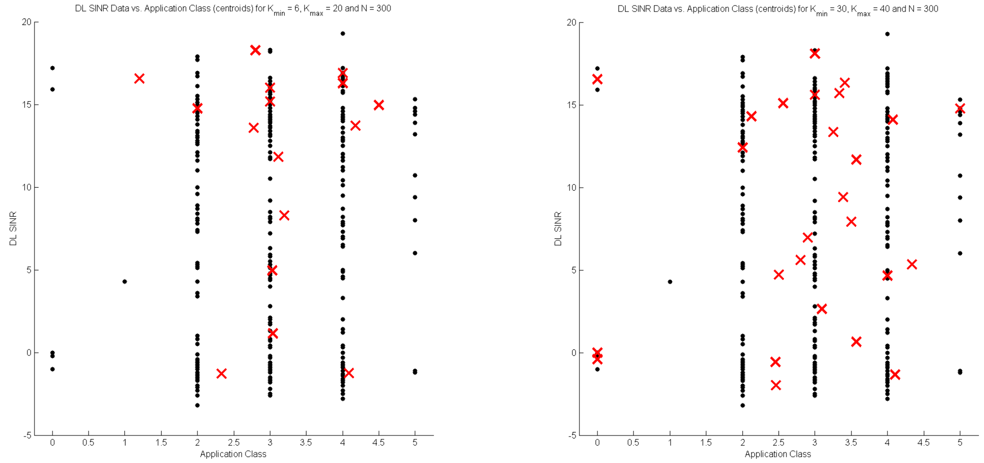
Our results, shown in Fig. 3.9, demonstrate that the DL SINR values have a more prominent influence in the location of the centroids. Hence, the vectors are grouped along the DL SINR values. Figure 3.10 shows how adjusting the parameters K_{min} and K_{max} provides better results concerning the centroid locations. However, the best output observed required applying scaling to the DL SINR values. This simple operation enhanced the results significantly, even for low values of K_{min} and K_{max}, as depicted in Fig. 3.11.



(a) Centroids

(b) Clusters

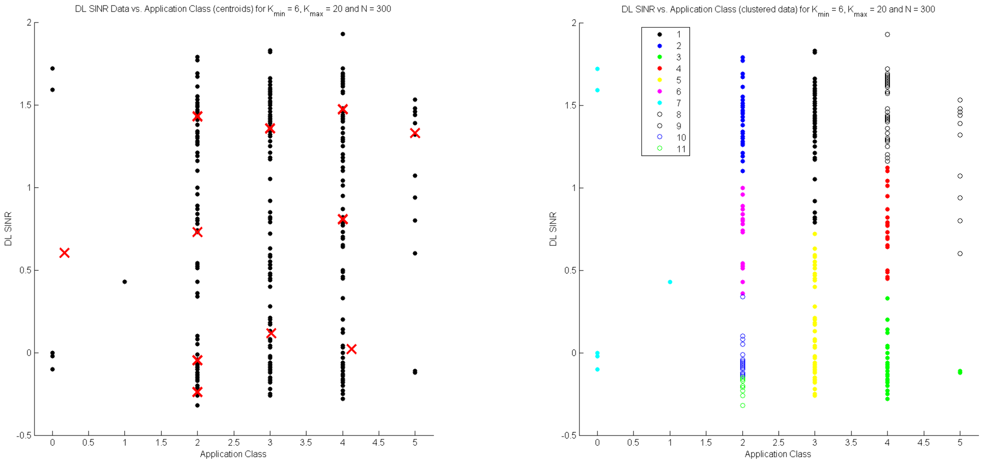
Figure 3.9: Clustering of feature vectors formed by DL SINR and Application Class data for $K_{min} = 1$, $K_{max} = 20$ considering $N = 300$ samples.



(a) Centroid locations for $K_{min} = 6$, $K_{max} = 20$

(b) Centroid locations for $K_{min} = 30$, $K_{max} = 40$

Figure 3.10: Clustering of feature vectors formed by DL SINR and Application Class data considering $N = 300$ samples and different values of K_{min} and K_{max}.



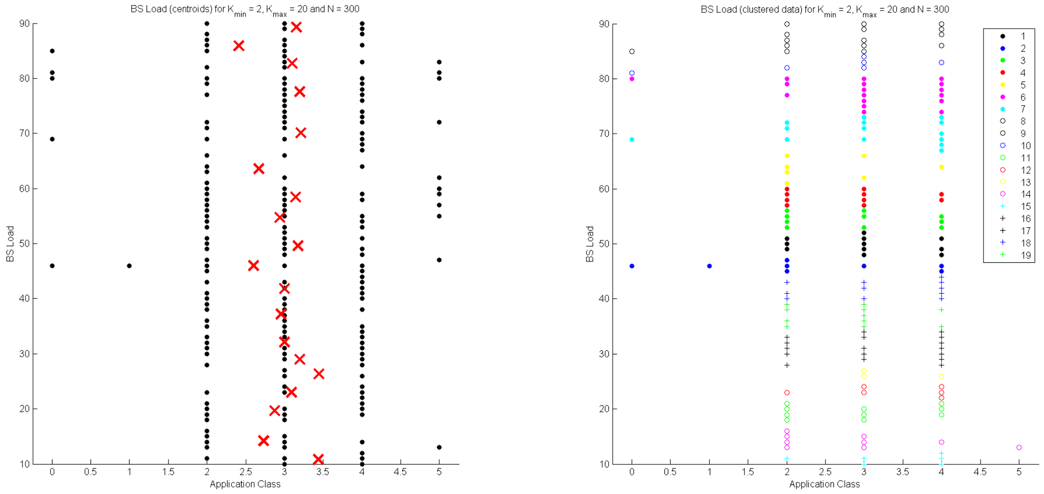
(a) Centroids

(b) Clusters

Figure 3.11: Clustering of feature vectors formed by DL SINR and Application Class data for $K_{min} = 6$, $K_{max} = 20$ considering $N = 300$ samples. The DL SINR values have been scaled by a factor of 0.1

3.4.2 BS Load vs. Application Class data clustering

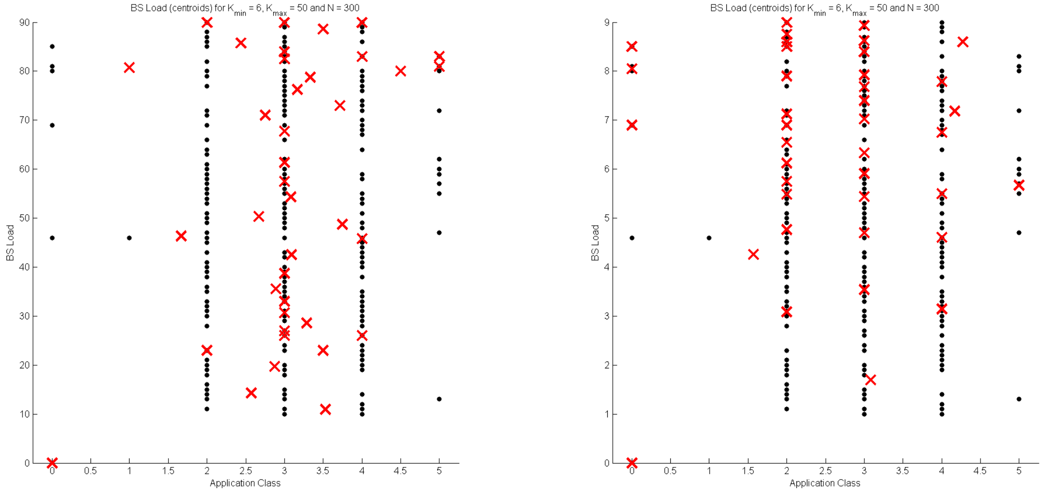
When combining the BS Load values with the Application Class values we obtain a very similar behavior to that the immediately previous subsection. The BS Load values have a more prominent influence in the location of the centroids (Fig. 3.12). If we apply the same technique of scaling the BS Load values, as we did with the DL SINR values, we visualize similar improvements in the output of the X-means algorithm. This is evident in Fig. 3.13 where the same values are adopted for the parameters K_{min} and K_{max} and although the resulting amount of clusters generated is similar when X-means converges (i.e., 41 and 39, respectively), the classification is superior in the case when the BS Load values are scaled.



(a) Centroids

(b) Clusters

Figure 3.12: Clustering of feature vectors formed by BS Load and Application Class data for $K_{min} = 2$, $K_{max} = 20$ considering $N = 300$ samples.



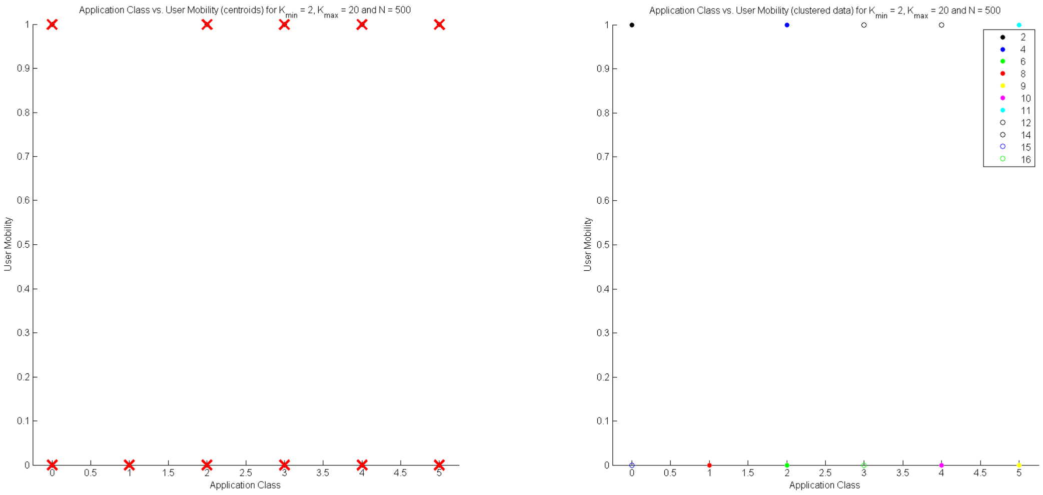
(a) Centroid locations for $K_{min} = 6$, $K_{max} = 50$

(b) Centroid locations for $K_{min} = 6$, $K_{max} = 50$. The BS Load values have been scaled by a factor of 0.1

Figure 3.13: Clustering of feature vectors formed by BS Load and Application Class data considering $N = 300$ samples.

3.4.3 User Mobility vs. Application Class data clustering

The results here were very straightforward; however, we needed to set at least $K_{min} = 2$ to obtain more than a single cluster at the output of X-means.



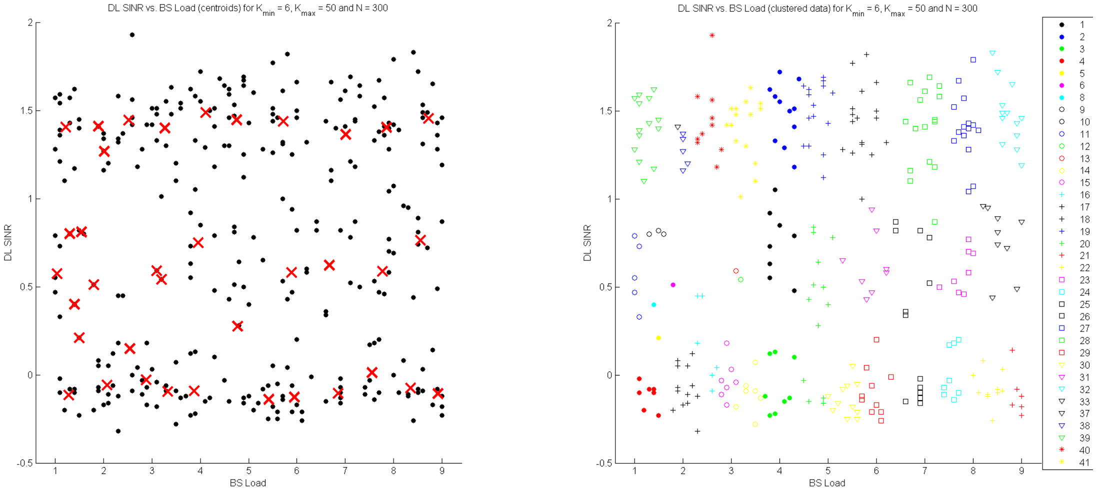
(a) Centroids

(b) Clusters

Figure 3.14: Clustering of feature vectors formed by User Mobility and Application Class data for $K_{min} = 2$, $K_{max} = 20$ considering $N = 500$ samples.

3.4.4 BS Load vs. Downlink SINR data clustering

Using the scaled version of the data of both descriptors offered appreciably good results. Fig. 3.15 is very illustrative of how out of 300 observations from a single base station, by means of clustering, we have obtained 41 representative points of combinations of radio link quality and congestion levels that can be recognized by our framework as cognitive states. Each of this 41 regions of the space spanned by this two descriptors can be associated to different expected throughput capacities.



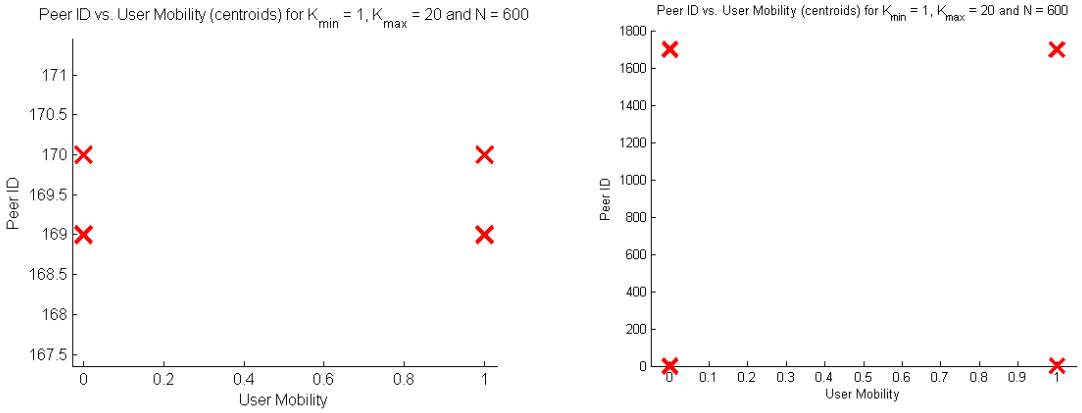
(a) Centroids

(b) Clusters

Figure 3.15: Clustering feature vectors formed by DL SINR and BS Load data for $K_{min} = 6$, $K_{max} = 50$ considering $N = 300$ samples. The vectors are scaled by a factor of 0.1

3.4.5 Peer ID vs. User Mobility

This is another interesting experiment because it combines the descriptors with the largest and lowest expected values. As is possible to confirm in Fig. 3.16, the results are satisfactory for reasonable values of Peer ID.



(a) Peer ID taking values from the set {169,170}

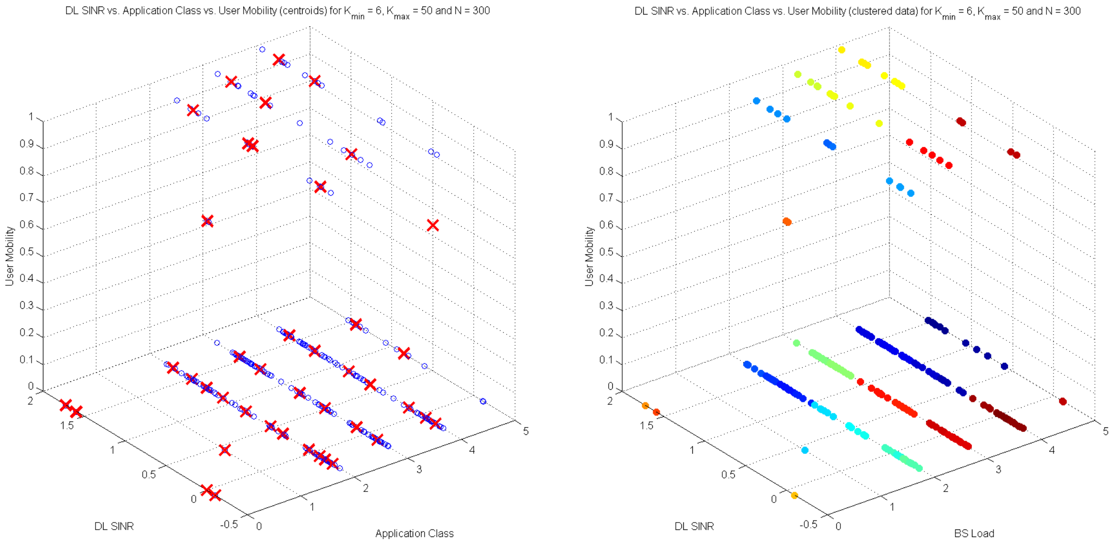
(b) Peer ID taking values from the set {2,1700}

Figure 3.16: Clustering of feature vectors formed by Peer ID and User Mobility using $K_{min} = 1$, $K_{max} = 20$ considering 300 samples from each Peer ID.

3.5 Three-dimensional data clustering

3.5.1 Downlink SINR vs. Application Class vs. User Mobility

Figure 3.17 displays reasonably good classification results of X-means obtained by generating only 48 clusters.



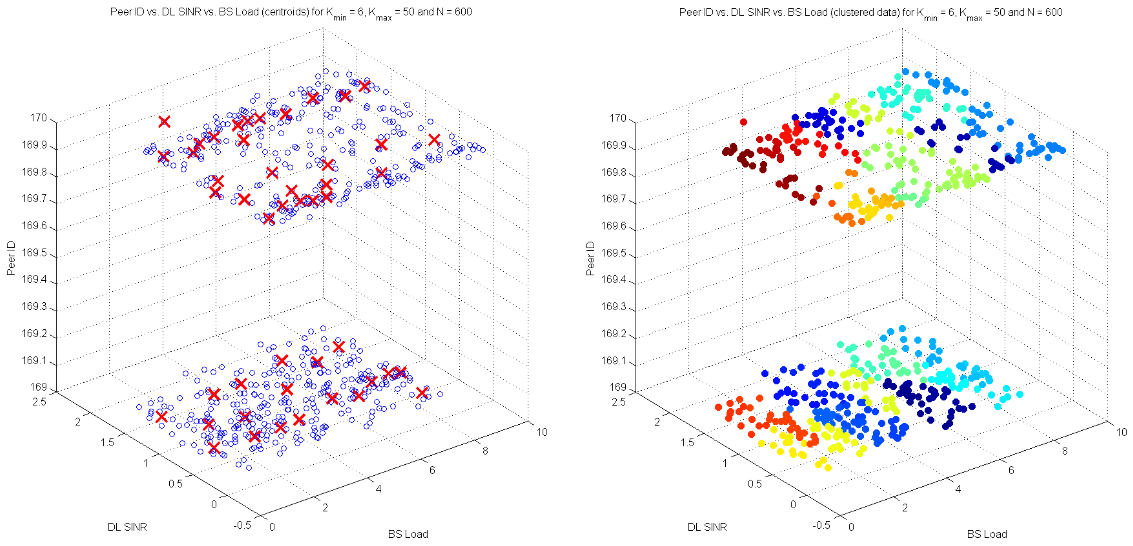
(a) Centroids

(b) Clusters

Figure 3.17: Clustering feature vectors formed by DL SINR, Application Class and User Mobility data for $K_{min} = 6$, $K_{max} = 50$ considering $N = 300$ samples. The DL SINR values have been scaled by a factor of 0.1.

3.5.2 Peer ID vs. Downlink SINR vs. BS Load

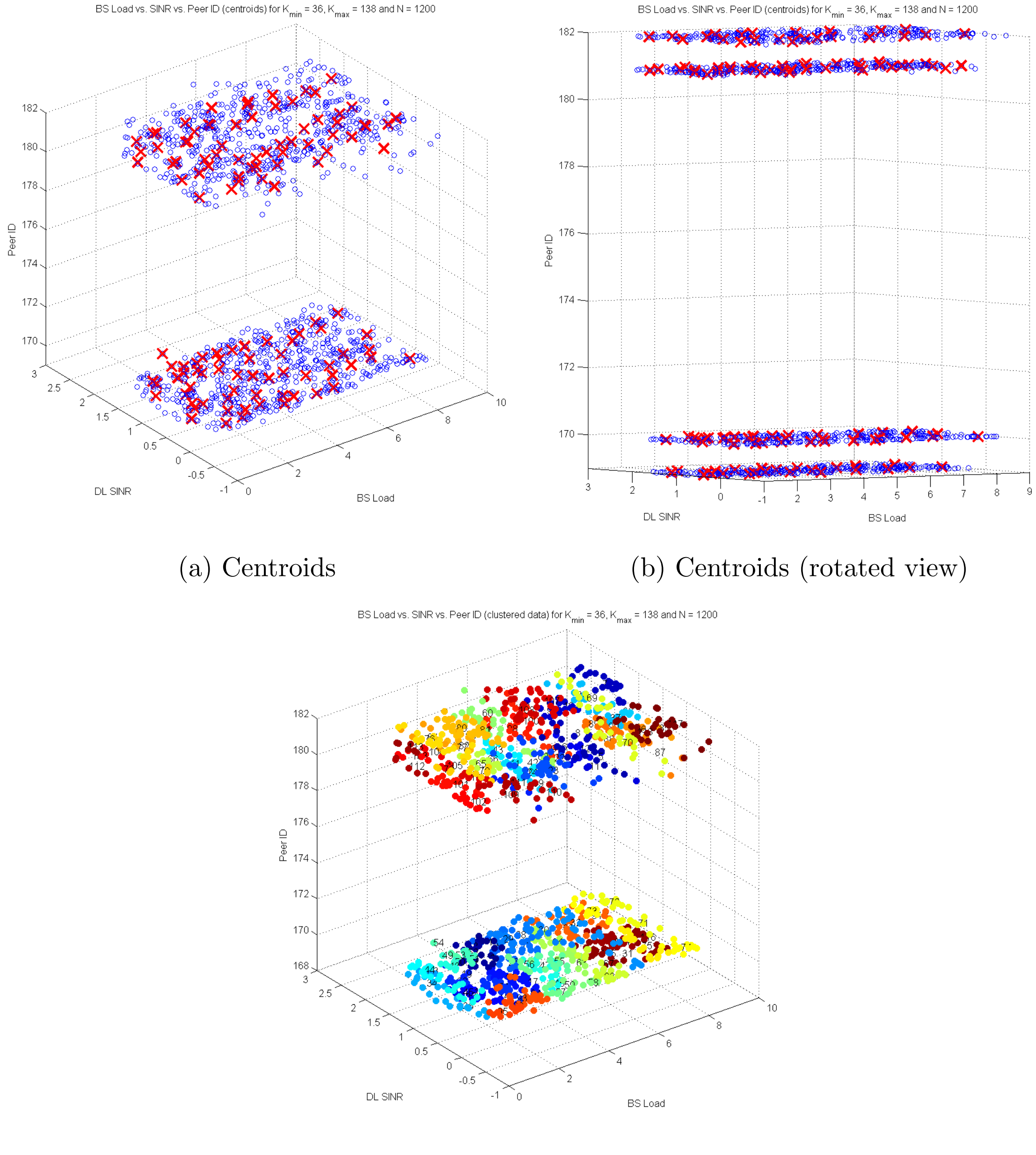
This case is very meaningful because not it illustrates how different combinations of BS Load and DL SINR values can represent different states across distinct network connections. Figures 3.18 and 3.19 indicate that X-means was able to satisfactorily group 600 data points associated with 2 different Peer IDs in 47 clusters, and 1200 data points associated with 4 different Peer IDs in 118 clusters, respectively.



(a) Centroids

(b) Clusters

Figure 3.18: Clustering of feature vectors formed by Peer ID, DL SINR and BS Load data for $K_{min} = 6$, $K_{max} = 50$ considering 300 samples from Peer ID 169 and 300 samples from Peer ID 170. Both the SINR and BS Load values have been scaled by a factor of 0.1.



(a) Centroids

(b) Centroids (rotated view)

(c) Clusters

Figure 3.19: Clustering of feature vectors formed by Peer ID, DL SINR and BS Load data for $K_{min} = 36$, $K_{max} = 138$ considering 1200 samples equally distributed from Peer IDs 169, 170, 181 and 182. Both the DL SINR and BS Load values have been scaled by a factor of 0.1.

3.6 Chapter Summary

The results of our experiments with the X-means algorithm have been satisfactory according to the needs of our proposed framework. Next, we briefly describe the most important observations during our test procedure:

- The standardization of the feature vectors.
  Much of the success of the experiments is due to the standardization of the descriptor data of the feature vectors. Such standardization has enabled us to have control over the characteristics of the input data and guarantees that only two of our features are noisy (i.e., DL SINR and BS Load).

- The scaling of DL SINR and BS Load values.
  The scaling of these parameters was decisive for confirming X-means as a useful tool for our solution. The scaling factor of 0.1 worked well for our testing data set as well as for all the simulations performed in the next chapter. Future research might indicate that this value is not suitable for real life implementation and require other rules of thumb.

- The values of K_{min} and K_{max}. Recommendations:
  (1) It is not a good idea to start with $K_{min} = 1$. In some occasions, provided the distribution of the data, X-means would not split this original cluster, producing unsatisfactory results.
  (2) As a general principle for our proposal, it is desirable to reduce the amount of clusters at the output of X-means. Empirically, we have noticed that the resulting number of clusters K is a non-decreasing function of K_{max}. Hence, choosing this parameter must be done carefully.

Let $d_{\mathbb{Z}}$ denote the set of descriptors in our feature vectors defined to contain non-negative integer values, according to our specification (i.e., Tier Class, Peer

ID, Application Class, User Mobility). We suggest to search for low values for K_{min} and K_{max} through the following exhaustive search algorithm:

Algorithm 2 Exhaustive search method for choosing values for K_{min} and K_{max}

```
 1: Initialize Flag = 0
 2: Initialize $K_{min} = 10$ and $K_{max} = N/10$.
 3: Run X-means algorithm procedure for current values of $K_{min}$ and $K_{max}$.
 4: for $m = 1, ..., K$ do
 5:     if centroid descriptor values $\mu_m(d_{\mathbb{Z}}) \notin \{0, 1, 2, ...\}$ then
 6:         Set Flag = 1.
 7:     end if
 8: end for
 9: if Flag = 1 then
10:     Set Flag = 0.
11:     Increase $K_{min} = K_{min} + \alpha$ and $K_{max} = K_{max} + \beta$
12:     Go to step (3).
13: else
14:     Stop.
15: end if
```

In the pseudo-code in Algorithm 2, $\mu_m(d_{\mathbb{Z}})$ denotes the set of values of the descriptors $d_{\mathbb{Z}}$ of the m-th centroid generated by the run of X-means in Step (3), and $\beta > \alpha$, for $\alpha, \beta \in \mathbb{N}$. We also have assumed that at least $N = 100$ feature vectors are being used for generating the clusters. This search method increases the values of the parameters until all the clusters are assigned to locations that satisfy our descriptor value specifications. This is the minimum requirement for an acceptable classification in our framework. This exhaustive search method has provided desired results in less than 20 iterations using $\alpha = 5$, $\beta = 10$ for our testing data set.

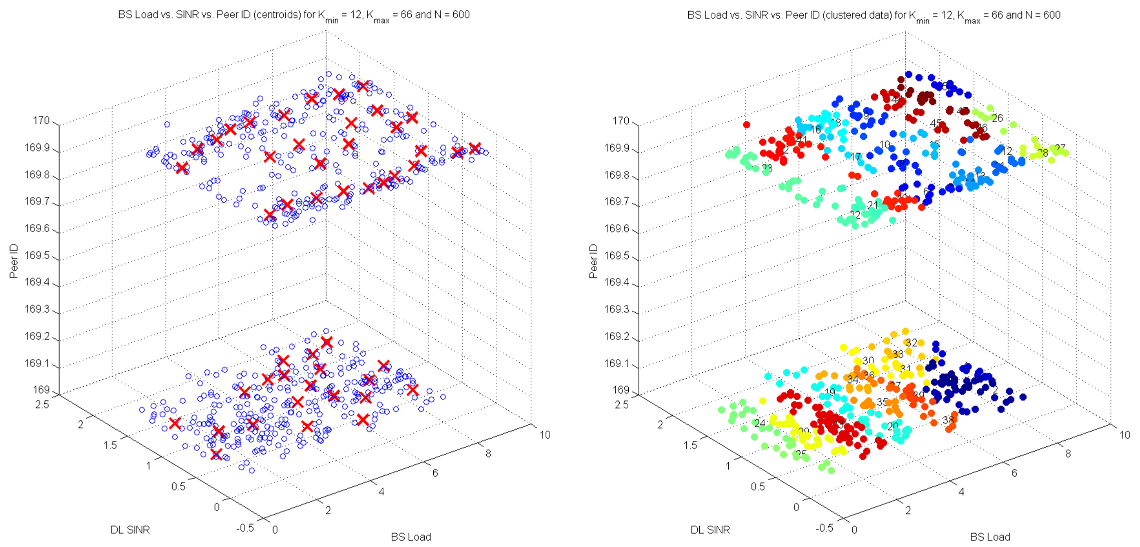
Chapter 4

Reinforcement learning simulations

4.1 Introduction

In this chapter, we present the results of MATLAB simulations that evaluate a simplified version of our proposed framework for user association in a multi-agent environment. The cognitive system models for each client node in our simulations were created using 3-dimensional feature vectors formed by the descriptors Peer ID (i.e., RAT index), downlink SINR, and the downlink Cell Load (i.e., an average of radio resource utilization). An example is presented in Fig. 4.1, where each cluster represents regions of combinations of DL SINR and Cell Load across two different network attachment points.

The following section presents our initial approach for verifying that the Q-learning algorithm was able to learn a good policy relying on the cognitive states model and a suitable reward function. Then, we proceed to lay out general assump-



(a) Centroids

(b) Clusters

Figure 4.1: X-means clustering of feature vectors formed by Peer ID, DL INR and BS Load, for $K_{min} = 12$, $K_{max} = 66$. 300 samples from Peer ID 169 and 300 samples from Peer ID 170 were used. Both the SINR and BS Load values have been scaled by a factor of 0.1.

tions and settings for our multi-agent (network) simulations. The remaining sections of this chapter show results and evaluate the performance of both Q-learning-based RAT selection approaches proposed in subsections 2.6.3 and 2.6.4, compared with other decision mechanisms in network scenarios of increasing complexity.

4.2 Initial tests with Q-learning.

4.2.1 Learning a stationary policy.

Our very first approach was conditioning the input of the algorithm in order to generate a stationary policy that was easy to predict. Therefore, for each instant t, we considered only two test feature vectors in our observation $\mathbf{y}_t = [x^{(170)}, x^{(169)}]$.

Here, $x^{(170)} = [7.8, -0.11, 170]^T$ is a feature vector associated to the RAT 170, and $x^{(169)} = [3.6, 2.17, 169]^T$ is a feature vector associated to the RAT 169. This simulates a situation in which two possible network attachment points are available at each decision instant and the network conditions remain constant regardless of the terminal's association decision. The cognitive states of Fig. 4.1 were used for our tests. The vectors $x^{(170)}$ and $x^{(169)}$ in $\mathbf{y}_t$ were classified by the kNN algorithm in clusters 28 and 43, respectively. Hence, the possible states of the terminal are $s_t \in \mathcal{S} = \{28, 43\}$, and the actions, $a_t \in \mathcal{A}_t = \{169, 170\}$. The state dynamics representation of this system is depicted in Fig. 4.2.

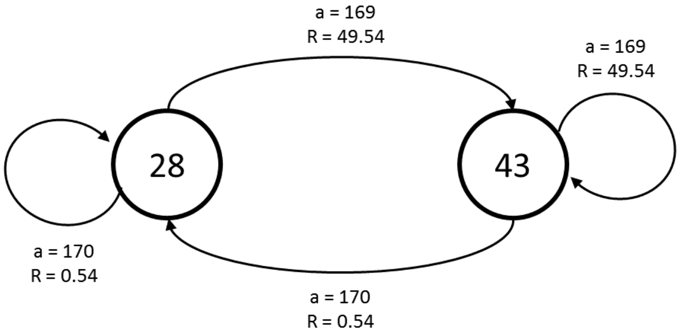


Figure 4.2: State dynamics of the system.

Let T_{at}^{max} be the mapping of the SINR value contained in the test feature vector of the current RAT association to the maximum theoretical throughput capacity according to the LTE standard [48]. Let L_{a_t} be the BS Load value contained in this feature vector as well. The rewards are computed as:

$$R_t(s_{t-1}, a_{t-1}) = \ \beta \cdot g(T_{a_t}^{max}, L_{a_t}) = \ \beta \cdot (10 - L_{a_t}) \cdot T_{a_t}^{max}, \tag{4.1}$$

where $\beta = 0.001$ is chosen for convenience.

Notice that states 28 and 43 have highly differentiated rewards because they represent distant regions of the DL SINR-BS Load plane of RATs with similar capabilities (Fig. 4.3).

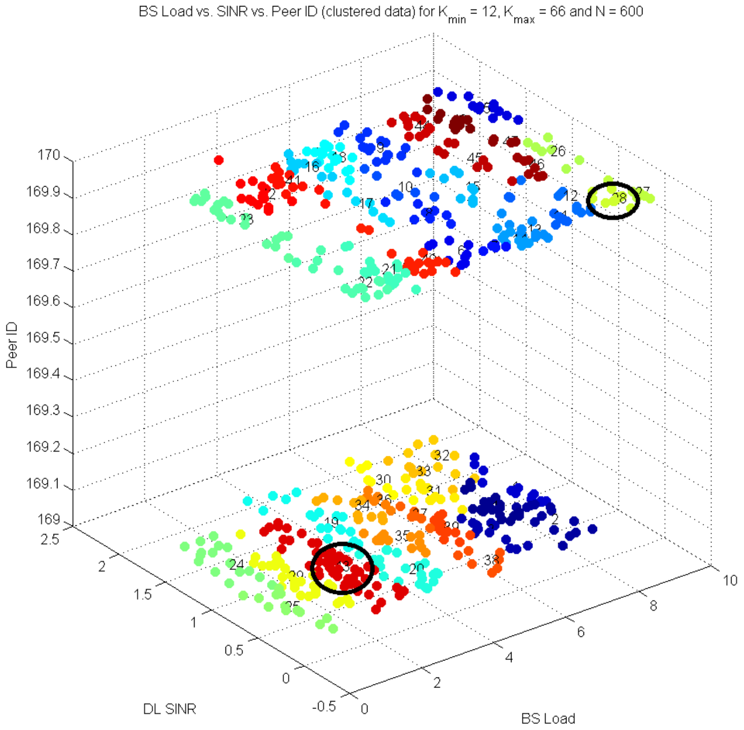


Figure 4.3: Highlighted location of clusters 28 and 43.

Therefore, an optimal policy that maximizes the rewards is a stationary policy that always chooses $a_t = 169$, as suggested by our results. Table 4.1 shows the values of the Q-table for the relevant entries, registered after the algorithm converged in the 162^{th} iteration. Figure 4.4 compares the performance of the Q-Learning algorithm with the maximum theoretical performance and with the performance of a decision mechanism based on random actions.

Table 4.1: Q-Table after convergence

States	Actions	
	170	169
28	49.86	161.11
43	103.27	164.98

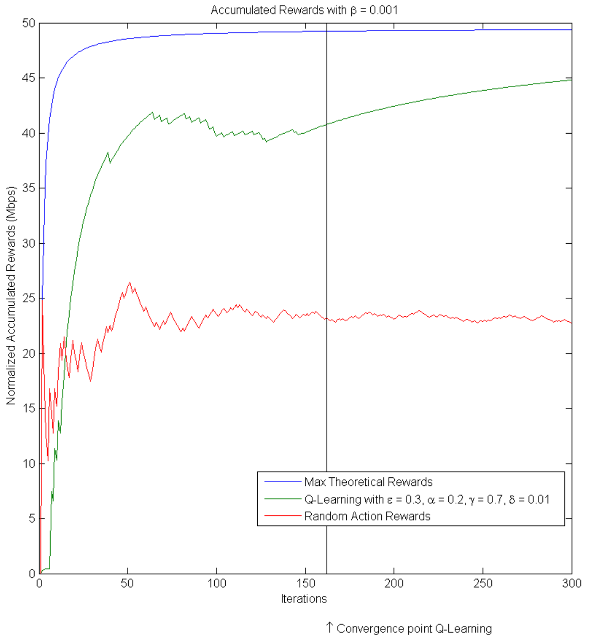
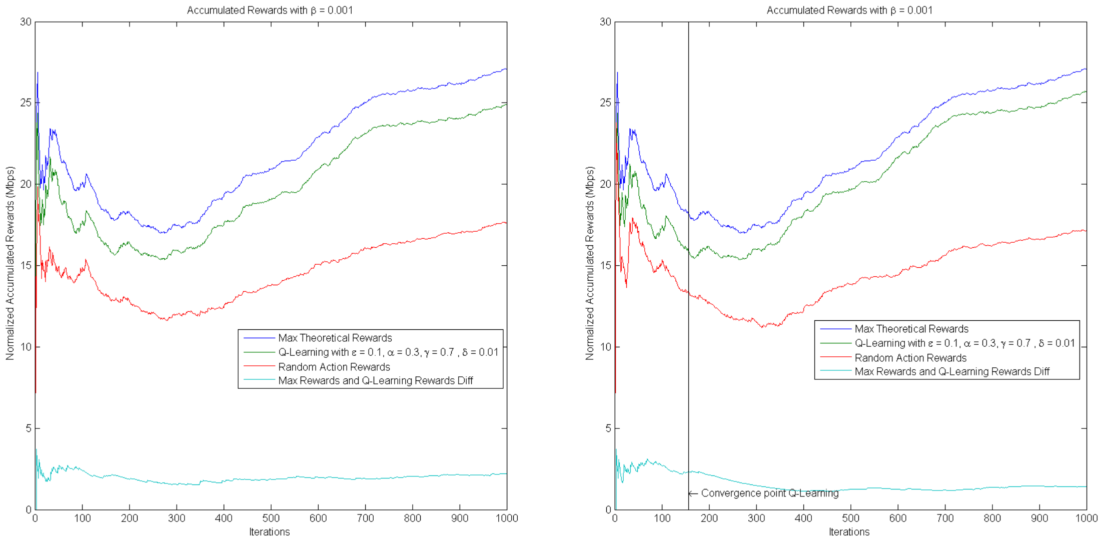


Figure 4.4: Q-learning averaged accumulated rewards compared with the maximum possible rewards and random action rewards after 300 iterations. Q-learning was run using $\varepsilon = 0.3$, $\alpha = 0.2$, $\gamma = 0.7$, $\delta = 0.01$.

4.2.2 Using the full cognitive model.

In this second stage we did not restrict the values of the observation as in the previous stage; instead, a subset of 1000 feature vectors from our testing dataset was used. In this case, the system dynamics are more complex. Not only the terminal state s_{t+1} is affected by the association action a_t, but also different feature vectors from the test dataset associated with the same RAT simulate underlying random processes that may cause that $s_t \neq s_{t+1}$ without changing the current RAT association. However, as verified in Fig. 4.5, with the proper tunning, Q-learning was able to approximate the maximum possible rewards.

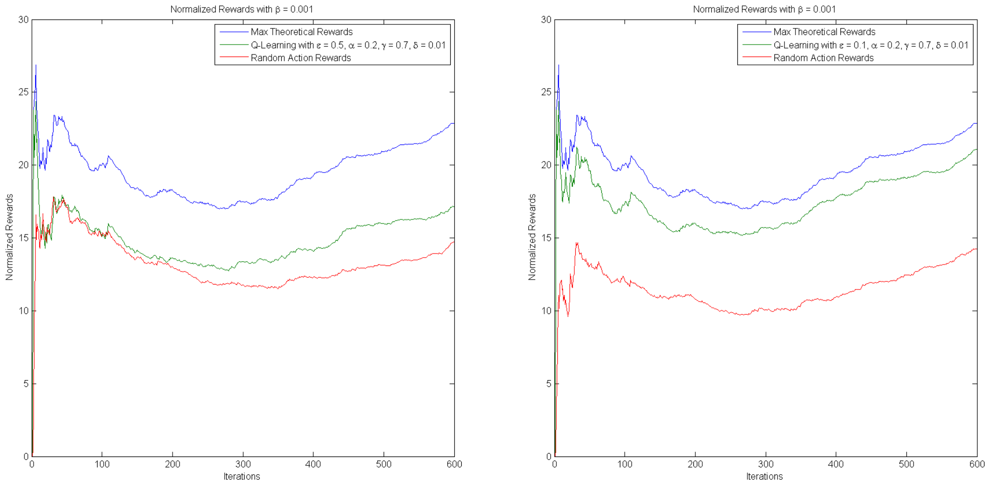


(a) Without convergence

(b) With convergence

Figure 4.5: Q-learning averaged accumulated rewards compared with the maximum possible rewards and random action rewards for 1000 iterations. The full 48-cognitive state model was used. Q-learning was run using $\varepsilon = 0.1$ before convergence and $\varepsilon = 0$ after convergence, $\alpha = 0.3$, $\gamma = 0.7$, $\delta = 0.01$.

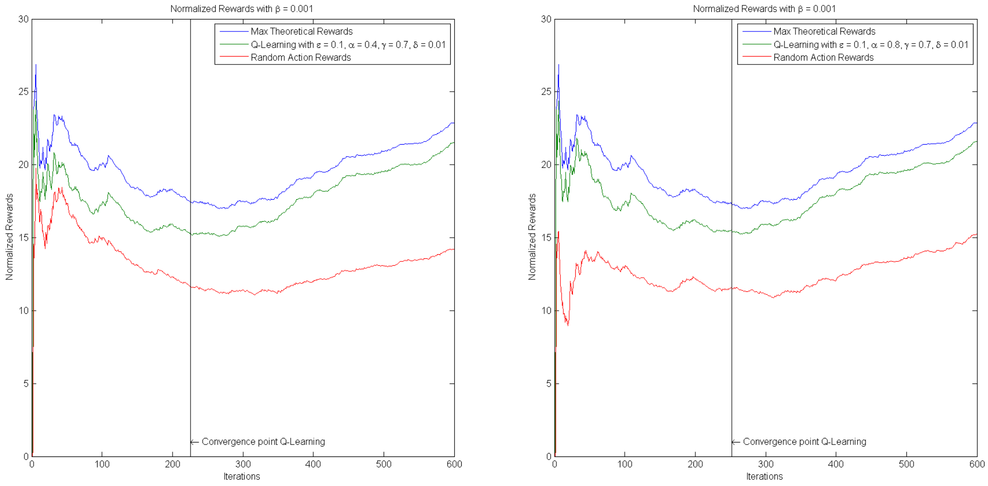
Our results indicated that the most important factor for fine-tuning Q-learning for our testing dataset was the exploration rate ε. A reduction in the value of the exploration rate produced a steady increase in the accumulated rewards (Fig. 4.6). We did not observe major differences in the results when using different values of discount factor γ or the step-size α. The latter is exemplified in Fig. 4.7.



(a) Exploration rate $\varepsilon = 0.5$

(b) Exploration rate $\varepsilon = 0.1$

Figure 4.6: Q-learning averaged accumulated rewards compared with the maximum possible rewards and random action rewards for 600 iterations. Q-learning was run using $\alpha = 0.2$, $\gamma = 0.7$, $\delta = 0.01$ for different values of ε.



(a) Learning rate $\alpha = 0.4$

(b) Learning rate $\alpha = 0.8$

Figure 4.7: Q-learning averaged accumulated rewards compared with the maximum possible rewards and random action rewards for 600 iterations. Q-learning was run using $\varepsilon = 0.1$ before convergence and $\varepsilon = 0$ after convergence, $\gamma = 0.7$, $\delta = 0.01$ for different values of α.

4.3 Multi-agent simulation set-up

Our simulation compares the network behavior of different user association mechanisms, namely, a random decision mechanism, the max-SINR rule and our proposed framework, under common radio-frequency (RF) and node mobility conditions. We restricted our implementation to the DL transmission and have assumed that there is no interference. We have also taken for granted that any client node has access to any serving node in the topology, which can be interpreted as each client node is equipped with an appropriate network interface for accessing any RAT being offered.

4.3.1 Decision Mechanisms.

Three decision methods are compared in our simulation:
(1) Random decision mechanism: At each decision instant, each client node will decide to associate with any of the available serving nodes based on a uniform probability distribution over the number of the available nodes.
(2) Max-SINR mechanism: The max-SINR rule associates a client node with the serving node that provides the highest average Downlink SINR in its initial network connection. For stability's sake, this association does not change unless the average Dowlink SINR related to the associated serving node drops below a certain threshold d_{thres} and the detected Downlink SINR of other available serving nodes surpass it by a certain hysteresis value d_{hyst}. In our implementation, the first time step is the only one considered as the initial network connection.
(3) Q-learning-based decision mechanism: This is the mechanism that corresponds to our proposed solution. We have analyzed both the conventional Q-learning, that tries to learn an action-value function, and the modified version of Q-learning, that tries to learn an afterstate-value function.

4.3.2 SNR Calculation.

At each simulation time step, the SNRs of the client nodes relative to each serving node are recomputed according to (4.2), using the parameters of tables 4.2 and 4.3:

$$SNR_{dB} = P_{dBm} - PL(d)_{dB} - N_{dBm} \tag{4.2}$$

where, P_{dBm} is the serving node transmit power in dBm, $PL(d)_{dB}$ is the path loss of the wireless link in dB that depends on the distance d between the client node and serving node, and N_{dBm} denotes the client node Noise Floor in dBm.

The model adopted in our simulation for the mean path loss term in (4.2), log-normal fading, is widely recommended in the literature [49–51] as a simple model that captures the essence of signal propagation and large scale (or slow) fading caused by blockages in the signal path for wireless channels:

$$PL(d)_{dB} = Ls(d_0)_{dB} + 10nlog_{10}(\frac{d}{d_0}) + X^{\sigma}_{dB}, \tag{4.3}$$

where:

(1) d_0 is a known close reference distance in the far field of the transmitting antenna in Km.

(2) $Ls(d_0)_{dB} = 32.44 + 20log_{10}(d_0) + 20log_{10}(f)$ is the path loss (specifically, the free-space loss) at a known reference distance d_0 between the transmitter and receiver in Km. using a carrier frequency f in MHz.

(3) n is a path-loss exponent obtained from empirical measurements.

(4) d is the distance between the transmitter and receiver in Km.

(5) X^{σ}_{dB} denotes a zero mean Gaussian random variable with variance units in dB.

Table 4.2: Log-Normal Fading Propagation Model parameters

Parameters	Macrocell	Microcell	Indoors
d_0 (Km.)	1.0	0.1	0.001
n	5.1	4.0	3.0
σ (dB)	8.0	8.5	7.0
P_{dBm}	40.0	30.0	23.0

Table 4.3: Radio Access Technology Parameters [52, 53]

Parameters	LTE FDD 10 MHz	LTE FDD 20 MHz	WiFi G	WiFi B
Carrier Frequency (GHz)	1.8 (Band 3)	2.5 (Band 7)	2.4	2.4
N_{dBm}	-94	-92	-88	-94

4.3.3 Cell Load Calculation.

Let I and J denote the total number of client nodes and serving nodes in the simulation, respectively. Note that, at each simulation time step, the decisions of the i-th client node, for $i \in \{1, \cdots, I\}$ associated to the j-th serving node, for $j \in \{1, \cdots, J\}$, will affect its load. Let T_i^{req} be the required throughput of the i-th client node. Let T_{ij}^{max} be the mapping of SNR values to the maximum theoretical throughput according to the LTE and Wi-Fi standards [48, 54]. Then, assume a time-sharing scheduling mechanism that allocates to each connected client node a time-slice according to the following proportion:

$$w_{ij} = \frac{T_{ij}^{max}}{\sum\limits_{i^{(j)}} T_{ij}^{max}}, \tag{4.4}$$

where, $\sum\limits_{i^{(j)}}$ denotes a summation over all the client nodes currently associated to the j-th cell. This scheduling strategy favors the nodes of relatively higher SNR with longer allocations. The average cell load L_j of the j-th cell is defined as

$$\begin{aligned} L_j &= \mathbb{E}\left[\frac{T_{ij}^{req}}{w_{ij}\cdot T_{ij}^{max}}\right] \\ &= \sum_{i^{(j)}} w_{ij}\cdot\left[\frac{T_{ij}^{req}}{w_{ij}\cdot T_{ij}^{max}}\right] \\ L_j &= \sum_{i^{(j)}} \frac{T_i^{req}}{T_{ij}^{max}}. \end{aligned} \tag{4.5}$$

4.3.4 Computation of Rewards.

In our simulations, the individual rewards R_t^i obtained by the i-th client node corresponding to each decision instant t, are computed using the following simplified form of (2.1):

$$R_t^i(s_{t-1}, s_t, a_{t-1}) = \ \beta\cdot g(T_i^{req}, L_j), \tag{4.6}$$

where $\beta = 0.001$ is chosen for convenience, and

$$g(T_i^{req}, L_j) = \begin{cases} T_i^{req} & \text{if } L_j \leq 1.0 \\ T_i^{req}/L_j & \text{if } L_j > 1.0 \end{cases}.$$

An analysis of our reward function allows us to predict the behavior of the simulation:

1. Network simulation settings that do not generate congestion (i.e. $< 100\%$ load values) in any of the available serving nodes using any of the decision mechanisms will achieve the same performance (i.e., will obtain the same rewards). We would expect that in such scenarios, under assumptions of low mobility of the nodes, the max-SINR algorithm would achieve the overall most efficient decisions in terms of lowest generated load.

2. We expect the rewards obtained by our proposed RAT selection mechanism to be superior to the other mechanisms when congestion is generated by the policies of all the decision mechanisms.

3. We expect that Q-learning will need higher number of iterations to converge in scenarios of high mobility compared to scenarios of low mobility.

4. We expect that the Random Decision mechanism will outperform the max-SINR algorithm as more serving nodes become available if we don't penalize the handover rate.

5. We expect the max-SINR algorithm to perform poorly in a HetNet environment with congested macrocells.

6. We expect the conventional Q-learning to outperform the modified version when the afterstates are a poor estimation of the future states and when kNN misclassification occurs. For the modified version of Q-learning, classification errors not only affect the states, but also the action set at each decision interval. Otherwise, we expect the afterstates Q-learning to obtain at least the same rewards than the conventional Q-learning.

4.4 Multi-agent simulation results.

This section presents results corresponding to network simulations. In all cases the client nodes were configured with high traffic rate requirements (i.e., $T_i^{req} = 40.0$ Mbps). Besides from the aggregate rewards computed, the smoothed cell load generated by each decision mechanism is also presented. The cell load is calculated using (4.6), and values higher than 1.0 have been truncated. The smoothing operation is necessary because the cell load can be very noisy and difficult to interpret when a large number of simulation time steps are used. To reduce the effects of these fluctuations, we smooth the original cell load values using an averaging rectangular sliding window of size W.

4.4.1 3 client nodes 2 serving nodes scenario.

In this case, our network consists of two LTE FDD macrocells using 10 MHz carriers, and three client nodes. The location of the serving and client nodes at the beginning of the simulation is depicted in Fig. 4.8. Figures 4.9 and 4.10 present the corresponding results to experiments in which static and random walk-based mobility models have been adopted, respectively. The Q-learning-based mechanisms outperform both other decision methods by means of achieving a better load balancing. In terms of the rewards, the performance of the Q-learning-based mechanisms is equivalent, obtaining 26% higher rewards than the random decision mechanism and 45% higher rewards than the max-SINR rule for a network with low mobility. These values decrease to 14% and 42%, respectively when mobility is present. The modified version of Q-learning obtains the presented rewards with a fraction of the RAT association changes of its counterpart; roughly, at least, 38% less association changes per time step after convergence.

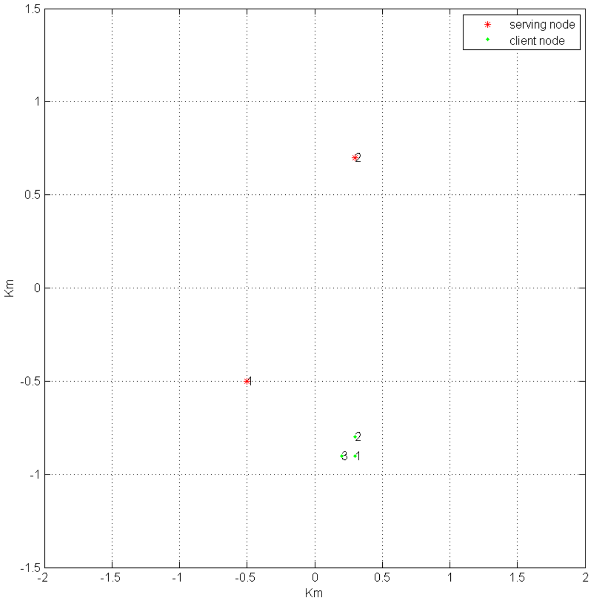
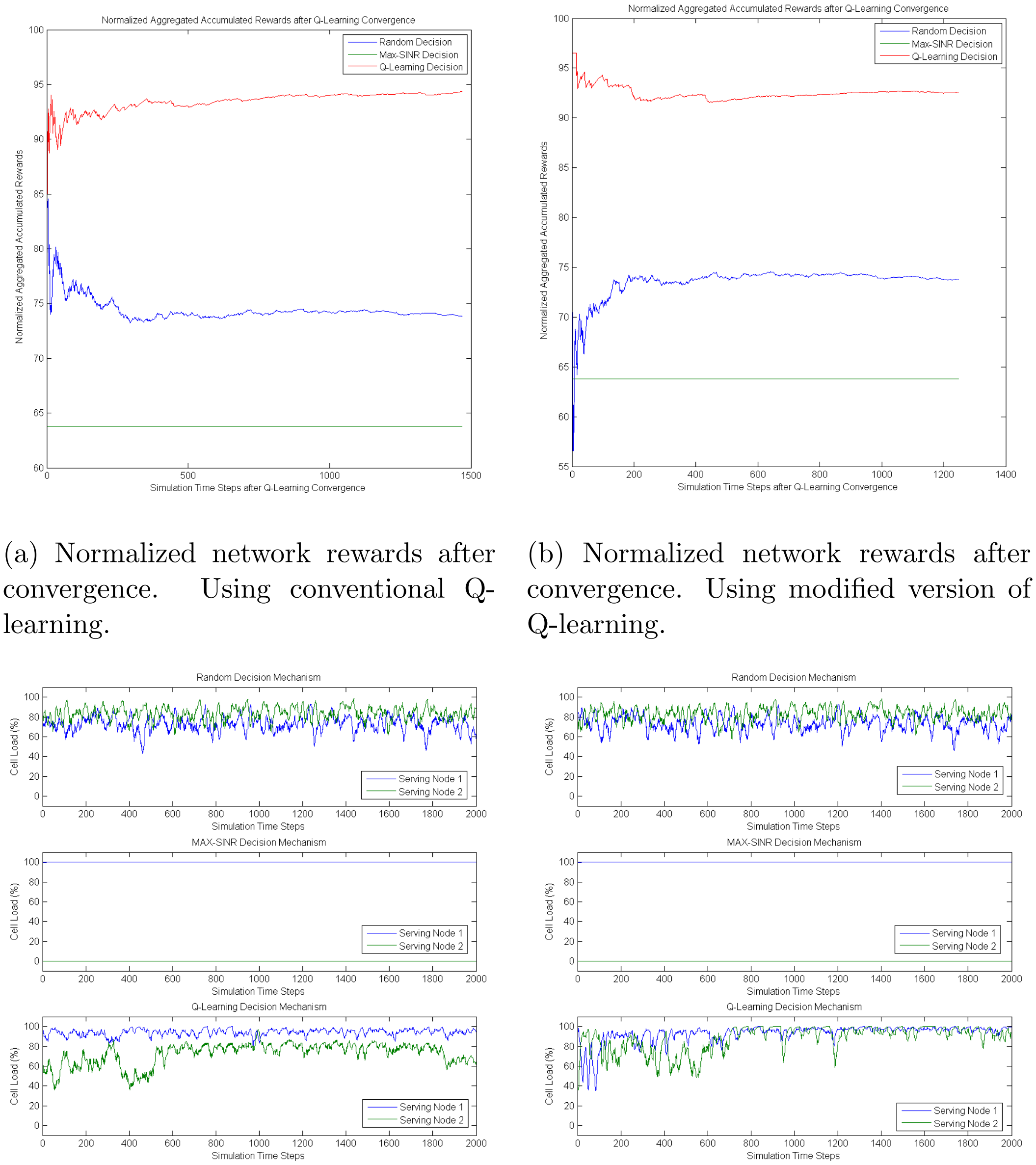


Figure 4.8: Network Topology for 3 client nodes and 2 serving nodes scenario.



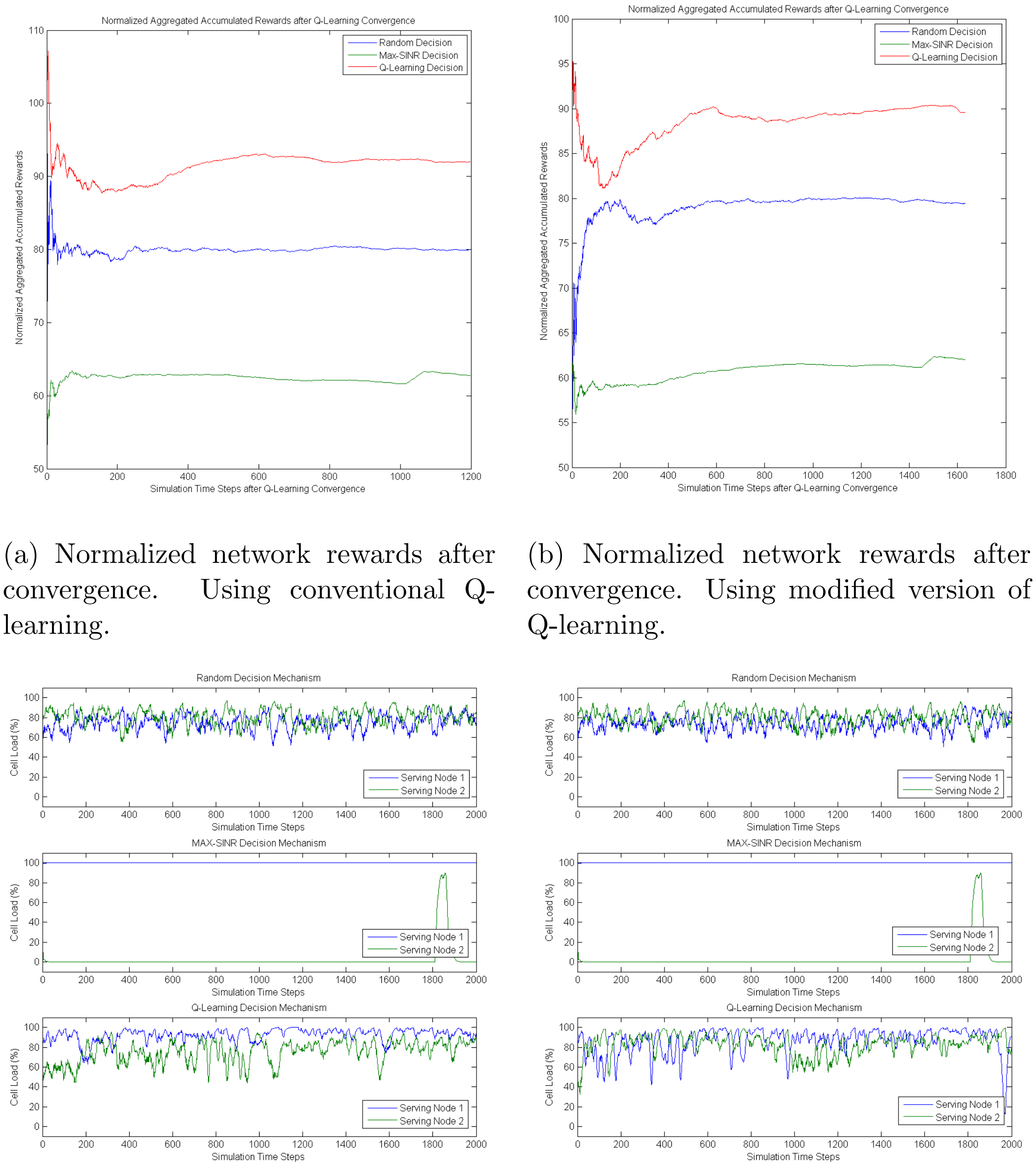
(a) Normalized network rewards after convergence. Using conventional Q-learning.

(b) Normalized network rewards after convergence. Using modified version of Q-learning.

(c) Smoothed Cell Load using $W = 21$. Using conventional Q-learning.

(d) Smoothed Cell Load using $W = 21$. Using modified version of Q-learning.

Figure 4.9: Simulation results for 3 static client nodes and 2 serving nodes scenario. Showing normalized network rewards after all Q-tables have converged and smoothed cell load per decision mechanism. Q-Learning parameters: $\varepsilon = 0.35$ before convergence and $\varepsilon = 0.1$ after convergence, $\alpha = 0.3$, $\gamma = 0.7$, $\delta = 0.01$. Max-SINR parameters: $d_{thresh} = 10$ dB and $d_{hyst} = 5$ dB. kNN parameters: $k = 3$



(a) Normalized network rewards after convergence. Using conventional Q-learning.

(b) Normalized network rewards after convergence. Using modified version of Q-learning.

(c) Smoothed Cell Load using $W = 21$. Using conventional Q-learning.

(d) Smoothed Cell Load using $W = 21$. Using modified version of Q-learning.

Figure 4.10: Simulation results for 3 mobile client nodes and 2 serving nodes scenario. Showing normalized network rewards after all Q-tables have converged and smoothed cell load per decision mechanism. Q-Learning parameters: $\varepsilon = 0.35$ before convergence and $\varepsilon = 0.1$ after convergence, $\alpha = 0.3$, $\gamma = 0.7$, $\delta = 0.01$. Max-SINR parameters: $d_{thresh} = 10$ dB and $d_{hyst} = 5$ dB. kNN parameters: $k = 3$

4.4.2 6 client nodes 2 serving nodes scenario.

For this scenario, we keep the two LTE FDD macrocells of the previous scenario, but we configure six client nodes. The initial location of the serving and client nodes is depicted in Fig. 4.11. Similar to the previous subsection, Figures 4.12 and 4.13 presents the corresponding results to experiments in which static and random walk-based mobility models have been adopted, respectively. Again, in terms of rewards, the Q-learning-based decisions achieve comparatively similar rewards and outperform by 8% the random mechanism and by 13% the max-SINR method in a network with low mobility. In high mobility, the proposed framework still has better performance but by a smaller margin, namely 3% compared to the random decision mechanism and 12% compared to the max-SINR. It is important to remark, nevertheless, that any of the Q-learning-based mechanisms in either of the experiments of this topology register at least 34% less changes in the associations than the random decision process. In this case, also, the modified version of Q-learning generates at least 28% less overhead per time step after convergence than its counterpart.

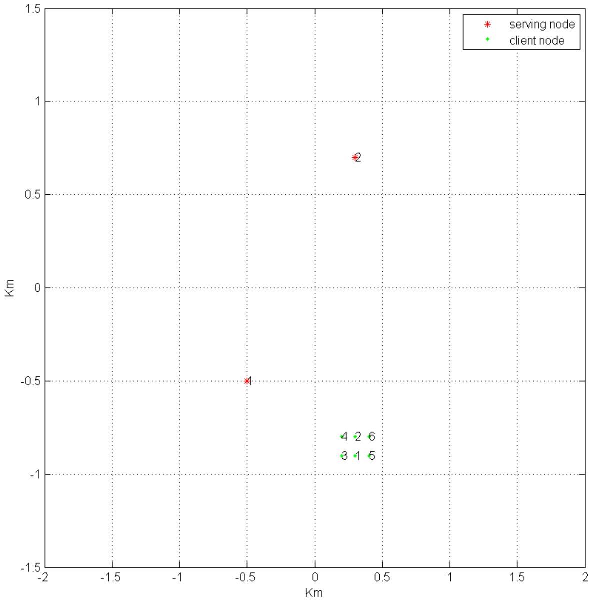
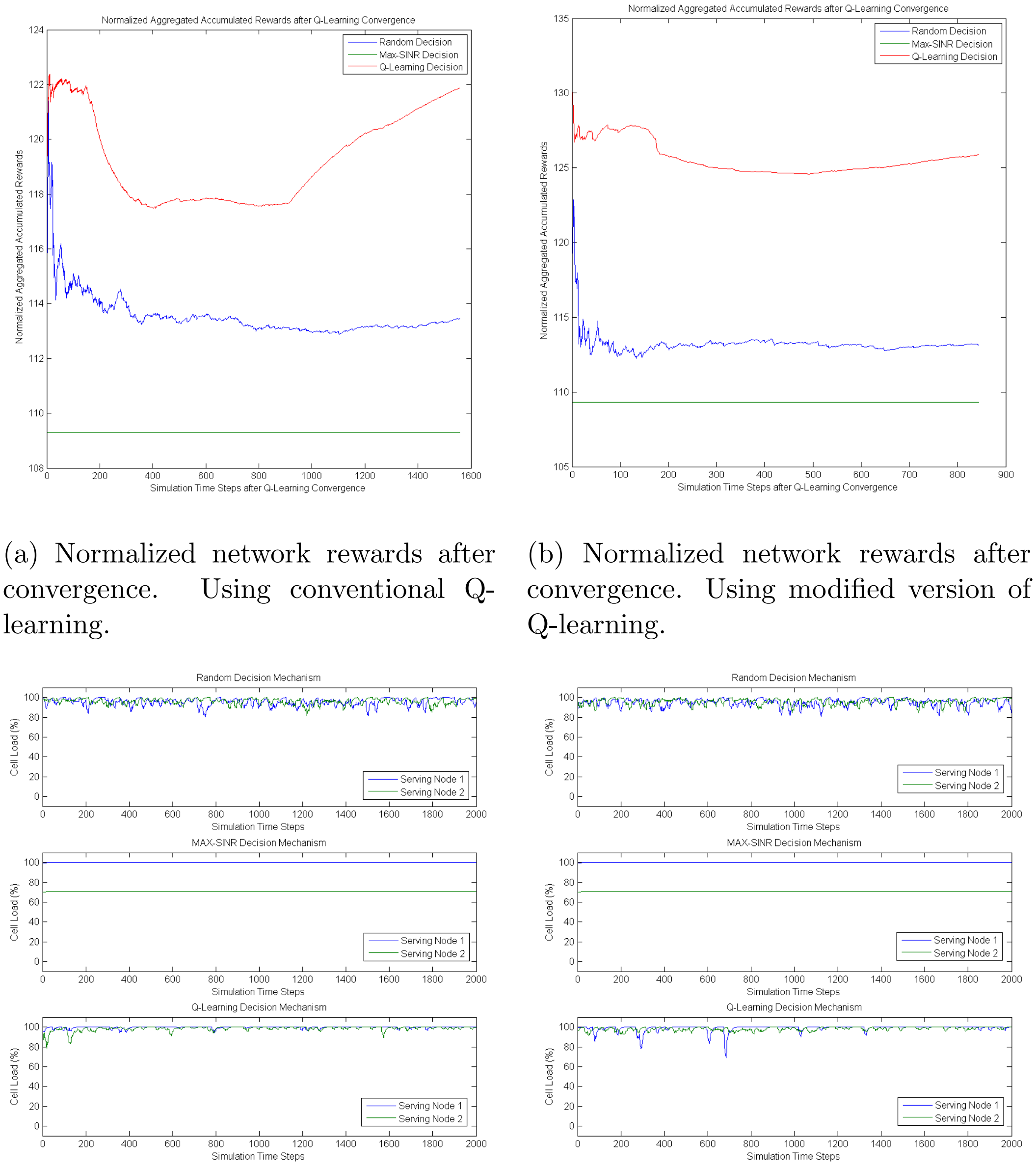


Figure 4.11: Network Topology for 6 client nodes and 2 serving nodes scenario.



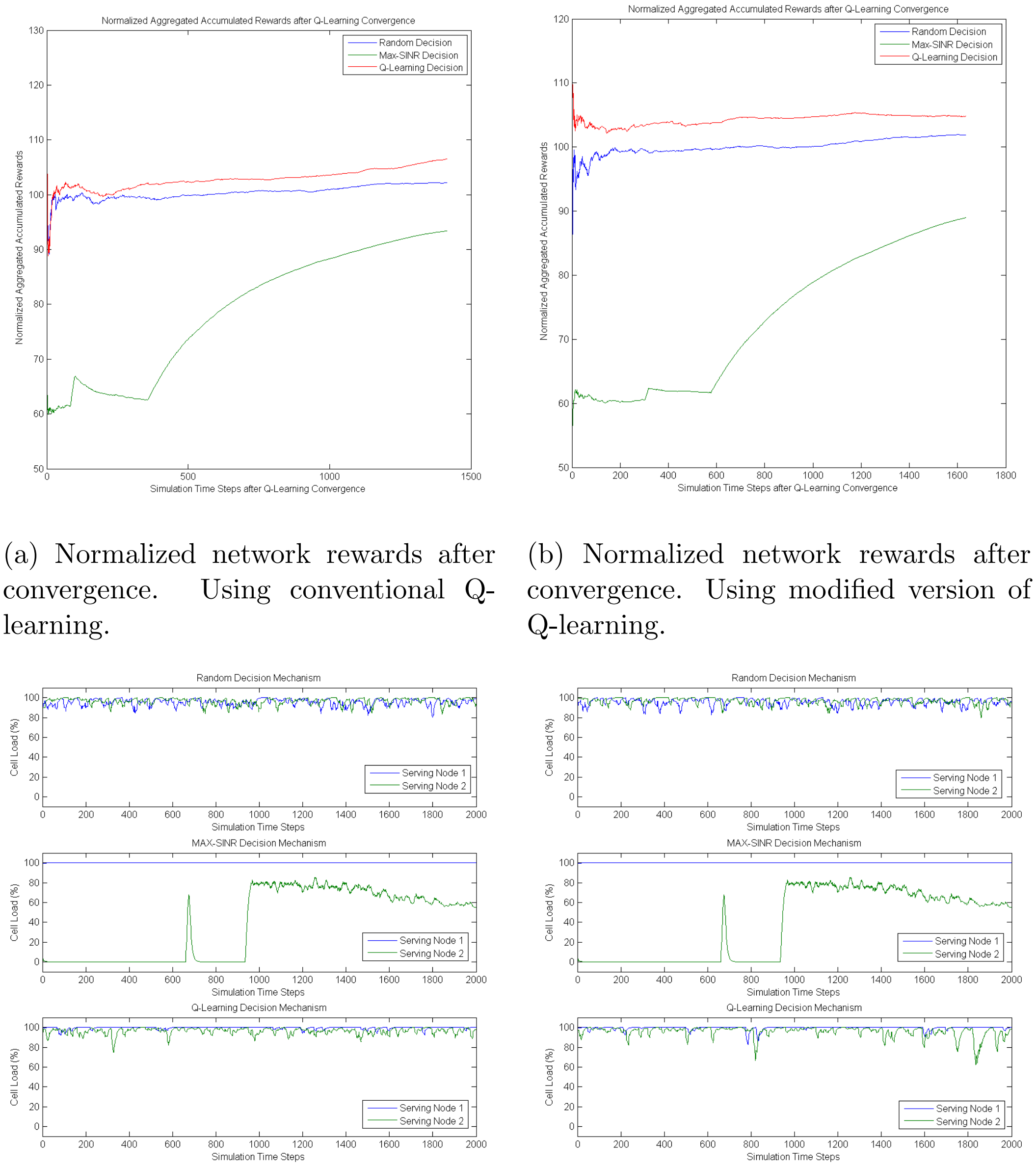
(a) Normalized network rewards after convergence. Using conventional Q-learning.

(b) Normalized network rewards after convergence. Using modified version of Q-learning.

(c) Smoothed Cell Load using $W = 21$. Using conventional Q-learning.

(d) Smoothed Cell Load using $W = 21$. Using modified version of Q-learning.

Figure 4.12: Simulation results for 6 static client nodes and 2 serving nodes scenario. Showing normalized network rewards after all Q-tables have converged and smoothed cell load per decision mechanism. Q-Learning parameters: $\varepsilon = 0.35$ before convergence and $\varepsilon = 0.1$ after convergence, $\alpha = 0.3$, $\gamma = 0.7$, $\delta = 0.01$. Max-SINR parameters: $d_{thresh} = 10$ dB and $d_{hyst} = 5$ dB. kNN parameters: $k = 3$



(a) Normalized network rewards after convergence. Using conventional Q-learning.

(b) Normalized network rewards after convergence. Using modified version of Q-learning.

(c) Smoothed Cell Load using $W = 21$. Using conventional Q-learning.

(d) Smoothed Cell Load using $W = 21$. Using modified version of Q-learning.

Figure 4.13: Simulation results for 6 mobile client nodes and 2 serving nodes scenario. Showing normalized network rewards after all Q-tables have converged and smoothed cell load per decision mechanism. Q-Learning parameters: $\varepsilon = 0.35$ before convergence and $\varepsilon = 0.1$ after convergence, $\alpha = 0.3$, $\gamma = 0.7$, $\delta = 0.01$. Max-SINR parameters: $d_{thresh} = 10$ dB and $d_{hyst} = 5$ dB. kNN parameters: $k = 3$

4.4.3 10 client nodes 10 serving nodes scenario.

In this case, our setup is a small HetNet consisting of seven LTE FDD macrocells using 10 MHz carriers, three LTE FDD microcells using 20 MHz carriers, and ten client nodes. The location of each node at the beggining of the simulation is depicted in Fig. 4.14. Figure 4.15 presents the network behavior assuming static client nodes, and Fig. 4.16, assuming a random walk-based mobility model. In the first case, the Q-learning-based methods generated the highest differences observed when compared with the alternative mechanisms during our tests. In between 47%-64% higher rewards compared to the random decision mechanism, and in between 127%-154% higher rewards compared to the max-SINR rule. The numbers are more moderate but equally reassuring for a network with high mobility. In between 33%-38% higher than the random decision mechanism and in between 14%-20% higher than the rewards obtained by the max-SINR decisions. Finally, in both experiments the modified Q-learning obtains 10% more rewards than the conventional Q-learning generating, at least, 43% less changes in the association decisions per time step after convergence.

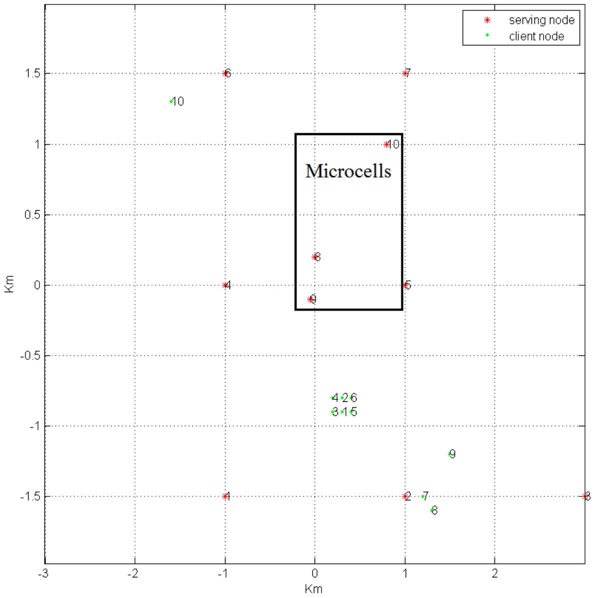
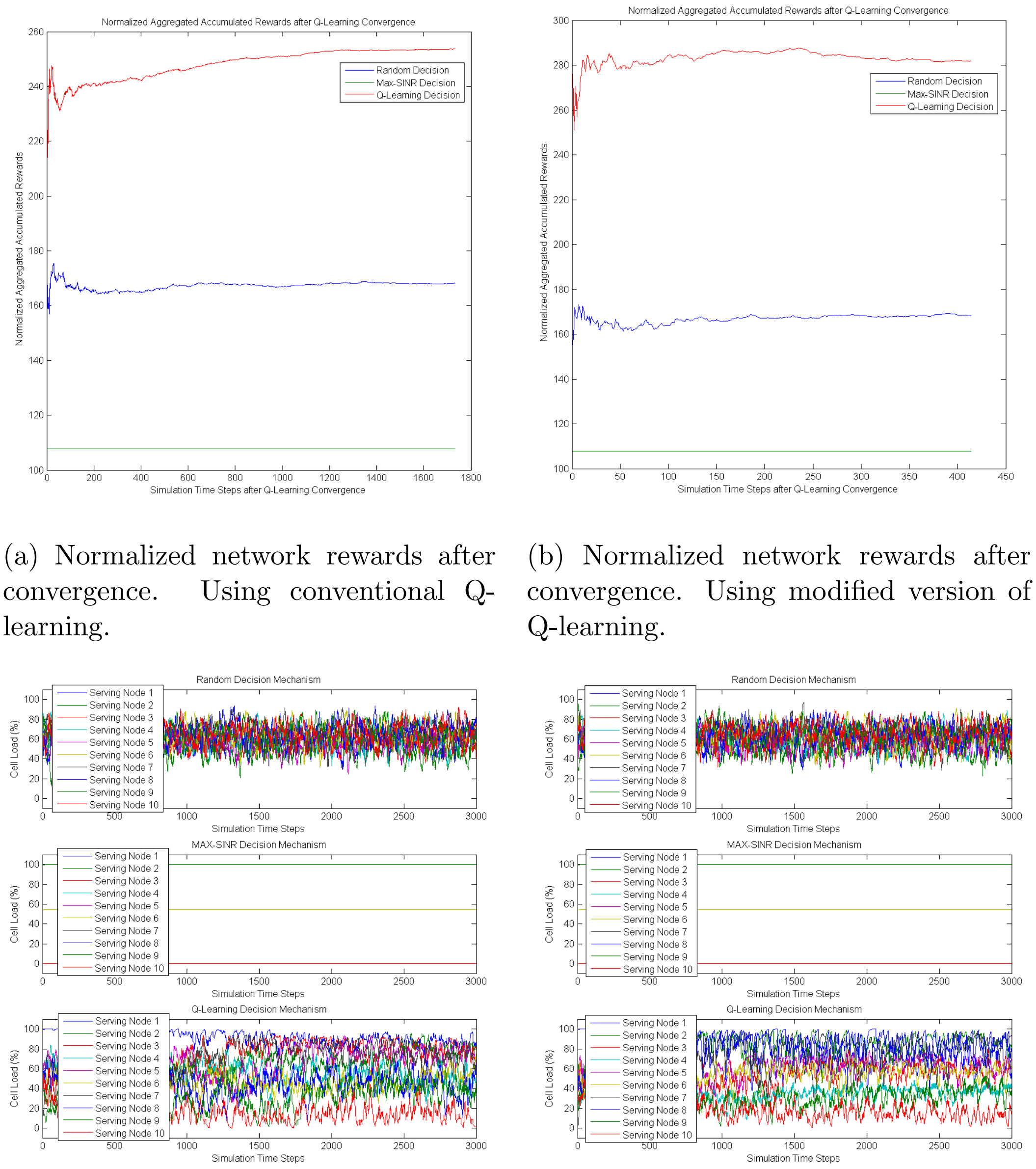


Figure 4.14: Network Topology for 10 client nodes and 10 serving nodes scenario.



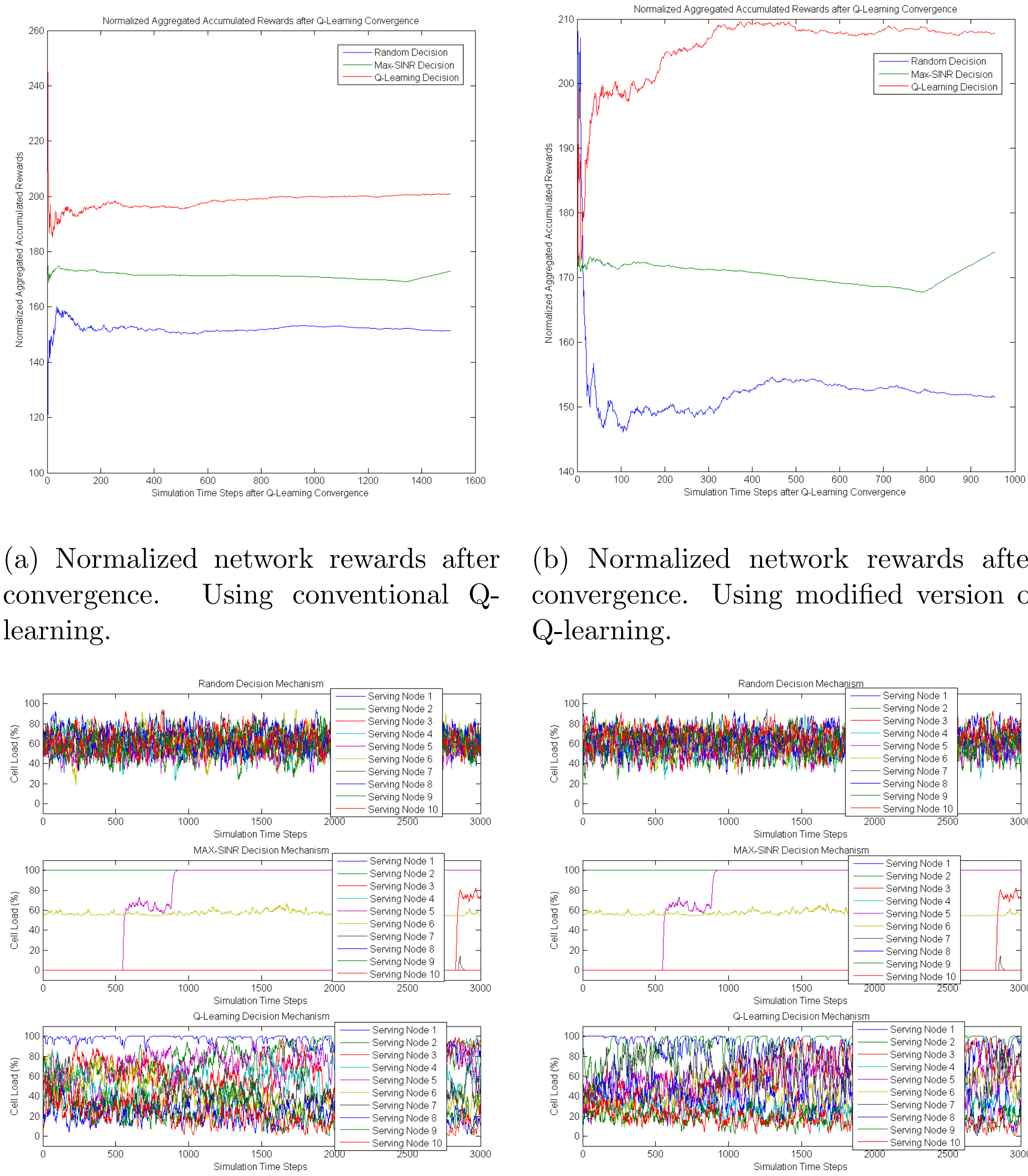
(a) Normalized network rewards after convergence. Using conventional Q-learning.

(b) Normalized network rewards after convergence. Using modified version of Q-learning.

(c) Smoothed Cell Load using $W = 21$. Using conventional Q-learning.

(d) Smoothed Cell Load using $W = 21$. Using modified version of Q-learning.

Figure 4.15: Simulation results for 10 static client nodes and 10 serving nodes scenario. Showing normalized network rewards after all Q-tables have converged and smoothed cell load per decision mechanism. Q-Learning parameters: $\varepsilon = 0.35$ before convergence and $\varepsilon = 0.1$ after convergence, $\alpha = 0.3$, $\gamma = 0.7$, $\delta = 0.01$. Max-SINR parameters: $d_{thresh} = 10$ dB and $d_{hyst} = 5$ dB. kNN parameters: $k = 3$



(a) Normalized network rewards after convergence. Using conventional Q-learning.

(b) Normalized network rewards after convergence. Using modified version of Q-learning.

(c) Smoothed Cell Load using $W = 21$. Using conventional Q-learning.

(d) Smoothed Cell Load using $W = 21$. Using modified version of Q-learning.

Figure 4.16: Simulation results for 10 mobile client nodes and 10 serving nodes scenario. Showing normalized network rewards after all Q-tables have converged and smoothed cell load per decision mechanism. Q-Learning parameters: $\varepsilon = 0.35$ before convergence and $\varepsilon = 0.1$ after convergence, $\alpha = 0.3$, $\gamma = 0.7$, $\delta = 0.01$. Max-SINR parameters: $d_{thresh} = 10$ dB and $d_{hyst} = 5$ dB. kNN parameters: $k = 3$

4.5 Chapter Summary

In this chapter we have presented the underlying theoretical background and implementation details of a wireless network simulation in MATLAB. We have designed it with the goal of effectively comparing the performance of our proposed solution with other two user association mechanisms of interest.

Q-learning-based methods and kNN have proven to be useful at the task of handling the cognitive model for learning good policies. Our proposed solution was shown to be consistently effective in a multi-agent environment, in the presence of congestion and varying RF conditions, at learning policies that obtain the highest rewards from the three decision mechanisms analyzed.

Moreover, we compared the performance of the conventional Q-learning and a modified version that tries to learn afterstate value-functions. We found that, although both approaches are almost equivalent in terms of the rewards obtained, the latter learns policies that are more stable and seems to be better suited for being applied in more specific situations (e.g., larger networks, or networks with low mobility). The tradeoff is increased complexity, extended convergence time and high risk in performance degradation by classification errors.

Chapter 5

Conclusions and Future Work

In this Thesis we have proposed a distributed cognitive framework based on machine learning algorithms for the RAT association problem in 5G Hetnets. We have justified the formulation of our framework with a discussion of the challenges of the User Association Problem as seen in current literature. Our proposed framework can learn simple state representations out of the terminal experience and user behavior, thus, reducing the complexity of the core network design requirements. Also, it allows multi-objective optimization of the association decisions while incurring minimal network overhead. Our simulation results, although based on a simplified version of our propositions, showed the feasibility and benefits of our framework. However, still there are many opportunities for continuing research.

An analysis of the fine-tunning of the reward function parameters to achieve the expected improvement of network performance is necessary. A second step in this direction would be a comprehensive assessment of the stability and fairness of the proposed solution.

It is desirable to extend the multi-agent simulations to more realistic scales, topologies and user behaviors by, for example, using Poisson point process (PPP) to model user and base station location and implementing the solution in a specialized network simulator that might generate more realistic traffic patterns.

Also, developing methods that provide some degree of visibility and control to mobile network operators (MNO) is a must, in order to implement our proposal in a real network. A straightforward approach would be to integrate our solution with servers like the ANDSF, and use the rules as guidelines that restrict the possible actions set at each decision interval. Maybe this could be implemented using PUSH messages in order to keep a reduced overhead.

Making the solution more robust by developing rules of thumb, and enhancing or replacing the specific machine learning algorithms adopted for each module of the framework, is also of interest. For example, implementing a classifier with higher accuracy and versatility than the kNN would encourage the choice of the modified Q-learning over the conventional Q-learning for mobile environments, with expected improvement in stability. In the case of clustering, it is important to derive general rules for the scaling of feature values to achieve the desired clustering. Moreover, the use of pruning methods that make the solution robust in the presence of time-varying probabilistic models would considerably increase the flexibility of the solution.

The definition of additional features for the feature vectors, that might be essential for future 5G networks and are still not foreseeable at the time of writing this Thesis, also calls for exploration.

Finally, the implementation of the framework in a physical terminal is the final objective. Each device will present its own challenges and will require special adaptations of the solution. However, we understand that applying our framework to IP flow association is already a possibility on multi-RAT capable terminals.

References

[1] J. Gonzalvez, “Samsung Electronics sets 5G speed record at 7.5 Gb/s,” *IEEE Vehicular Technology Magazine*, vol. 10, no. 1, pp. 12–16, March 2015.

[2] C. I, M. A. Uusitalo, and K. Moessner, “The 5G Huddle,” *IEEE Vehicular Technology Magazine*, vol. 10, no. 1, pp. 28–31, March 2015.

[3] J. G. Andews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. K. Soong, and J. C. Zhang, “What will 5G be?” *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 6, pp. 1065–1082, June 2014.

[4] B. Bangerter, S. Talwar, R. Arefi, and K. Stewart, “Networks and devices for the 5G Era,” *IEEE Communications Magazine*, vol. 52, no. 2, pp. 90–96, February 2014.

[5] E. G. Larsson, O. Edfords, F. Tufvesson, and T. L. Marzetta, “Massive MIMO for Next Generation Wireless Systems,” *IEEE Communications Magazine*, vol. 52, no. 2, pp. 186–195, February 2014.

[6] J. G. Andews, S. Singh, Q. Ye, X. Lin, and H. Dhillon, “An overview of load balancing in HetNets: Old myths and open problems,” *IEEE Communications Magazine*, vol. 21, no. 2, pp. 18–25, April 2014.

[7] K. Shen and W. Yu, “Distributed pricing-based user association for Downlink Heterogeneous Cellular Networks,” *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 6, pp. 1100–1113, June 2014.

[8] A. Y. Panah, S. Yeh, N. Himayat, and S. Talwar, “Utility-based radio link assignment in multi-radio heterogeneous networks,” in *GC’12 Workshop: International Workshop on Emerging Technologies for LTE-Advanced and Beyond-4G, IEEE 2012*, Anaheim, CA, December 2012, pp. 618–623.

[9] Y. M. Kwon, J. S. Kim, J. Gu, and M. Y. Chung, “ANDSF-based congestion control procedure in heterogeneous networks,” in *International Conference on*

Information Networking (ICOIN), IEEE 2013, Bangkok, January 2013, pp. 547–550.

[10] L. C. Gonçalves, P. S. ao, N. Souto, and A. Correia, “Addressing Cell Edge Performance by Extending ANDSF and Inter-RAT UE Steering,” in *11th International Symposium on Wireless Communications Systems (ISWCS), IEEE 2014*, Barcelona, January 2014, pp. 465–469.

[11] V. Yazici, U. C. Kozat, and M. O. Sunay, “A new control plane for 5G Network architecture with a case study on unified handoff, mobility and routing management,” *IEEE Communications Magazine*, vol. 52, no. 11, pp. 76–85, November 2014.

[12] D. S. Kim, Y. Noishiki, and H. Yokota, “Efficient ANDSF-assisted Wi-Fi Control for Mobile Data Offloading,” in *9th International Wireless Communications and Mobile Computing Conference (IWCMC), IEEE 2013*, Sardinia, July 2013, pp. 343–348.

[13] M. Zorzi, A. Zanella, A. Testolin, M. de Filippo de Grazia, and M. Zorzi, “Cognition-Based Networks: A new perspective on Network Optimization using Learning and Distributed Intelligence,” *IEEE Access*, vol. 3, pp. 1512–1530, September 2015.

[14] F. Gao and K. Zhang, “Enhanced multi-parameter cognitive architecture for Future Wireless Communications,” *IEEE Communications Magazine*, vol. 53, no. 7, pp. 86–92, July 2015.

[15] H. ElSawy, H. Dahrouj, T. Y. Al-Naffouri, and M.-S. Alouini, “Virtualized cognitive network architecture for 5G cellular networks,” *IEEE Communications Magazine*, vol. 53, no. 7, pp. 78–85, July 2015.

[16] K. Doppler, C. B. Ribeiro, and J. Kneckt, “On efficient discovery of next generation local area networks,” in *Wireless Communications and Networking Conference (WCNC), IEEE 2011*, Cancun, Quintana Roo, March 2011, pp. 269–274.

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

[18] E. Aryafar, A. Keshavarz-Haddad, M. Wang, and M. Chiang, “RAT Selection Games in HetNets,” in *Proceedings INFOCOM, IEEE 2013*, Turin, April 2013, pp. 998–1006.

[19] *3GPP TS 23.402*, Architecture enhancements for non-3GPP accesses (Release 11), Rev. V11.1.0, December 2011.

[20] *3GPP TS 24.312*, Access Network Discovery and Selection Function (ANDSF) Management Object (MO) (Release 11), Rev. V11.1.0, December 2011.

[21] B. Bangerter, S. Talwar, R. Arefi, and K. Stewart, "5G Communications Race: Pursuit of More Capacity Triggers LTE in Unlicensed Band," *IEEE Vehicular Technology Magazine*, vol. 10, no. 1, pp. 43–51, March 2015.

[22] (2013) Extending LTE Advanced to unlicensed spectrum. Qualcomm Incorporated. [Online]. Available: https://www.qualcomm.com/media/documents/files/white-paper-extending-lte-advanced-to-unlicensed-spectrum.pdf

[23] M. V. D. Schaar, *Multimedia over IP and Wireless Networks: Compression, Networking, and Systems.* Burlington, MA: Academic Press, 2007, ch. 17.

[24] D. Zhou and W. Song, *Multipath TCP for user cooperation in wireless networks (ebook).* Springer, 2014.

[25] A. Konsgen, A. Singh, and C. Goerg, "Tutorial: Multipath support in IP networks," in *5th International Conference on Information and Automation Sustainability (ICIAFs), IEEE 2010*, Colombo, December 2010, pp. 37–38.

[26] A. Gudipati, D. Perry, L. E. Li, and S. Katti. (2013) SoftRAN: Software Defined Radio Access Network. [Online]. Available: http://web.stanford.edu/~skatti/pubs/hotsdn13-softran.pdf

[27] M. Bansal, J. Mehlman, S. Katti, and P. Levis. (2012) OpenRadio: A programmable wireless dataplane. [Online]. Available: https://users.ece.cmu.edu/~vsekar/Teaching/Spring15/18731/papers/Wireless_Bansal.pdf

[28] D. Soldani and A. Manzalini, "Horizon 2020 and beyond: On the 5G Operating System for a true digital society," *IEEE Vehicular Technology Magazine*, vol. 10, no. 1, pp. 32–42, March 2015.

[29] Z. Zaidi, V. Friderikos, and M. A. Imran, "Future RAN architecture: SD-RAN through a general-purpose processing platform," *IEEE Vehicular Technology Magazine*, vol. 10, no. 1, pp. 52–59, March 2015.

[30] P. Casas. (2012) Understanding network traffic: An introduction to Machine Learning in networking. [Online]. Available: http://www.tma-portal.eu/wp-content/uploads/2011/12/TMA_phd_Machine_Learning.pdf

[31] (2016) Tshark. [Online]. Available: https://www.wireshark.org/docs/man-pages/tshark.html

[32] L. Deri, M. Martinelli, T. Bujlow, and A. Cardigliano, "nDPI: Open-source high-speed deep packet inspection," in *International Wireless Communications and Mobile Computing Conference (IWCMC), IEEE 2014*, August 2014, pp. 617–622.

[33] T. Bujlow, V. Carela-Espanol, and P. Barlet-Ros, "Independent comparison of popular DPI tools for traffic classification," *Computer Networks*, vol. 76, pp. 75–89, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.comnet.2014.11.001

[34] (2016) nDPI: Open and extensible LGPLv3 Deep Packet Inspection Library (list of supported protocols). NTOP. Retrieved on March 16th, 2016. [Online]. Available: http://www.ntop.org/products/deep-packet-inspection/ndpi/

[35] A. Smola and S. V. N. Vishwanathan, *Introduction to Machine Learning.* United Kingdom: Cambridge University Press, 2008.

[36] S. K. Jayaweera, *Signal Processing for Cognitive Radios.* Hoboken, New Jersey: Wiley, 2015, ch. 8.

[37] D. Pelleg and A. Moore. (2000) X-means: Extending K-means with Efficient Estimation of the Number of Clusters. School of Computer Science, Carnegie Mellon University. [Online]. Available: https://www.cs.cmu.edu/~dpelleg/download/xmeans.pdf

[38] G. Schwarz, "Estimating the Dimension of a Model," *The Annals of Statistics*, vol. 6, no. 2, pp. 461–464, March 1978.

[39] S. R. Kulkarni and G. Harman. (2011) Statistical learning theory: A tutorial. [Online]. Available: http://www.princeton.edu/~harman/Papers/SLT-tutorial.pdf

[40] S. K. Jayaweera, *Signal Processing for Cognitive Radios.* Hoboken, New Jersey: Wiley, 2015, pp. 364–372, 681–688.

[41] M. L. Littman, N. Ravi, E. Fenson, and R. Howard, "Reinforcement learning for autonomic network repair," in *Proceedings of the International Conference on Autonomic Computing (ICAC'04), IEEE 2004*, May 2014, pp. 284–285.

[42] R. Razavi, S. Klein, and H. Claussen, "A fuzzy reinforcement learning approach for self-optimization of coverage in LTE networks," *Bell Labs Tech. J.*, vol. 15, no. 3, pp. 153–175, December 2010.

[43] S. Whiteson and P. Stone, "Towards autonomic computing: Adaptive network routing and scheduling," in *International Conference on Autonomic Computing, IEEE 2004*, May 2004, pp. 286–287.

[44] A. Galindo-Serrano and L. Guipponi, "Distributed Q-learning for aggregated interference control in cognitive radio networks," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 4, pp. 1823–1834, May 2010.

[45] J. A. McCann and M. Huebscher. (2004) Evaluation issues in autonomic computing. [Online]. Available: http://pubs.doc.ic.ac.uk/autonomic-eval/autonomic-eval.pdf

[46] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction.* Cambridge, Massachusetts; London, England: The MIT Press, 2012, ch. 6.

[47] (2011) Allot MobileTrends: Global Mobile Broadband Traffic Report H1/2011. Allot Communications. [Online]. Available: http://www.allot.com/resource-library/mobiletrends-h1-2011/

[48] *3GPP TS 36.213*, Physical layer procedures (Release 9), Rev. V9.3.0, September 2010.

[49] A. Goldsmith, *Wireless Communications.* New York: Cambridge University Press, 2005, ch. 2.

[50] B. Sklar, *The Mobile Communications Handbook*, 2nd ed. New York: CRC Press and IEEE Press, 1999, ch. 18.

[51] J. B. Andersen, T. S. Rappaport, and S. Yoshida, "Propagation Measurements and Models for Wireless Communications Channels," *IEEE Communications Magazine*, vol. 33, no. 1, pp. 42–49, January 1995.

[52] *3GPP TS 36.101*, User Equipment (UE) radio transmission and reception (Release 8), Rev. V8.15.0, November 2011.

[53] (2015) Intel 7265 802.11ac 2x2 DualBand Combo PCIe x1 Card Datasheet. HP. Retrieved on May 5th, 2016. [Online]. Available: http://www8.hp.com/h20195/v2/GetPDF.aspx/c04694748.pdf

[54] (2014) What is the relationship between data rate, SNR, and RSSI? Aruba Networks. Retrieved on April 18th, 2016. [Online]. Available: http://community.arubanetworks.com/t5/Controller-Based-WLANs/What-is-the-relationship-between-data-rate-SNR-and-RSSI/ta-p/178312