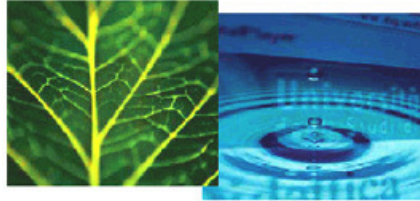


PhD Dissertation



**International Doctorate School in Information and
Communication Technologies**

DISI - University of Trento

**BRIDGING THE GAP BETWEEN THEORY AND
IMPLEMENTATION IN COGNITIVE NETWORKS:
DEVELOPING REASONING IN TODAY'S NETWORKS**

Christian Facchini

Advisor:

Prof. Fabrizio Granelli

University of Trento

Co-Advisor:

Prof. Nelson L.S. da Fonseca

State University of Campinas, SP, Brazil

December 2011

Abstract

Communication networks are becoming increasingly complex and dynamic. The networking paradigm commonly employed, on the other hand, has not changed over the years, and, as a result, performs poorly in today's environments. Only very recently, a new paradigm named cognitive networking has been devised with the objective to make networks more intelligent, thereby overcoming traditional limitations and potentially achieving better performance. According to such vision, networks should be able to monitor themselves, reason upon the environment, act towards the achievement of specific goals and learn from experience. Thus far, several cognitive network architectures have been conceived and proposed in the literature, but, despite researchers seem to agree on the need for a holistic approach, their architectures pursue such a global vision only in part, as they do not consider networks nor network nodes in their entirety.

In the present work, we analyze the aspects to be tackled in order to enable this holistic view and propose to base reasoning on both intra- and inter-node interactions, with the ultimate aim to devise a complete cognitive network architecture. After a thorough analysis of advantages and drawbacks of generic reasoning framework, we select the most apt to form the basis on which to build the cognitive network we envision. We first formalize its application in network environments, by determining the steps to follow in the process to equip traditional network with cognitive capabilities. Then, we shift the focus from the design side to the implementation side, by identifying the problems that could be faced when realizing such a network, and by proposing a set of optional refinements that could be taken into account to further improve the performance in some specific situations. Finally, we tackle the problem of reducing the time needed for the cognitive process to reason.

Validation through simulations shows that explicitly considering cross-layer intra- and inter-node interactions when reasoning has a twofold effect. First, it leads to better performance levels than those that can be achieved by today's non-intelligent networks, and second, it helps to better understand existent causal relationships between variables in a network.

Keywords

[cognitive network, adaptive network, cross-layering, network modeling]

A Gemma e Alice

Acknowledgments

This achievement would not have been possible without the help and support of many people, and this page is here to thank them all.

First, I would like to thank my advisor, Prof. Fabrizio Granelli. It was an invaluable opportunity to have you as mentor, not only for the doctoral program, but also for the Master's and Bachelor's degrees. Thanks to you I could grow up as a student, first, and as a researcher, then. You taught me how to perform research independently and objectively, fostered in me the ability to look at problems from different perspectives, and put your trust in me by giving me increasing responsibility, since the very beginning.

I would like to thank Prof. Nelson L.S. da Fonseca, who has been my co-advisor during the PhD program and allowed me to visit him at the University of Campinas in Brazil and become part of his research team. You have been an excellent guide and role model: from you I have learned many things, the value of dedication, drive, and determination, in particular. Thank you especially for doing literally everything to make me have a wonderful time in Brazil.

A special thanks goes also to Dr. Oliver Holland, who agreed to be my supervisor and made it possible for me to spend a period at King's College London and work as part of his research team. I have really enjoyed working with you: doing field research was fantastic—definitely beyond any expectation!

Fabrizio, Nelson, and Oliver, the three of you have been far more than mentors to me, you have been and are wonderful persons.

Agora, os agradecimentos mais pessoais. Quero agradecer a galera do LRC: Nelson, que já citei, Carlos, Cesar, Luciano, Flavio, Daniel, Joana, Jorge, Neumar, Cleo, Pedro e André. Se o tempo que passei no Brasil foi tão legal, foi só graças a vocês. E se eu aprendi o sentido de conceitos como amizade, honestidade e hospitalidade, foi também só graças a vocês. Não poderia ter pedido nada melhor que encontrar vocês no meu caminho.

Thanks to the friends I met at King's College, Oliver, Paul, Andrej, Reza, and Helen. I enjoyed every minute with you, but especially the insightful talks we had—not to mention the pints of Guinness.

Non posso poi dimenticare di ringraziare tutti i “compagni di open space”, ancor prima amici, che meglio di chiunque altro conoscono gioie e dolori del dottorato. Grazie dunque a Alessandro, Andrea, Igor, Nicola, Paolo, Edoardo, Luca e Gianluca. Grazie anche agli amici di sempre, quelli che riescono a farti staccare dalla ricerca e ti portano al bar, ad arrampicare, in montagna o in moto e magari senza saperlo ti regalano un'intuizione su come andare avanti. Grazie in particolare a Max, Federica, Andrea, Lorena, Paolo, Alessio e al Gruppo Guzzisti Anonimi Trentino. Un ringraziamento speciale va a Ilaria, che ho

avuto la fortuna di incontrare in questo percorso, che ha saputo spronarmi nei momenti più difficili, aiutarmi in quelli più complicati, e offrirmi il suo supporto, sempre. Grazie davvero per esserci. Un grazie va anche a Martina, che nello stesso percorso ho perso di vista, ma che mi è stata vicina agli inizi ed è stata presente quando ne avevo bisogno.

Infine, un ringraziamento speciale va a mio padre, un punto fermo non solo durante questo percorso, ma in tutta la vita, che mi ha supportato in tutte le scelte, in special modo quelle più difficili, e che ha saputo consigliarmi sempre per il meglio. Se sono qui ora, è soprattutto grazie a te.

Contents

Contents	ix
List of Tables	xi
List of Figures	xiii
1 Introduction	1
1.1 The Context	1
1.2 The Problem	3
1.3 The Solution	4
1.4 Innovative Aspects of the Thesis	8
1.5 Structure of the Thesis	9
2 State of the Art	11
2.1 Cognitive Architectures and Guiding Principles	11
2.2 Research Challenges	14
2.2.1 Reasoning and Learning	14
2.2.2 Adaptation	15
2.2.3 Information Representation	16
2.3 State-of-the-art Solutions	17
2.3.1 Reasoning and Learning	17
2.3.2 Adaptation	19
2.3.3 Information Representation	20
2.4 Research Directions	21
2.5 Conclusion	23
3 The proposed approach	25
3.1 Towards a Model for Quantitative Reasoning in Cognitive Nodes	27
3.1.1 Introduction	27
3.1.2 Fuzzy Cognitive Maps	29
3.1.3 Fuzzy Cognitive Maps for Cognitive Nodes	32
3.1.4 The Proposed Approach	38
3.1.5 Conclusion	40
3.2 The Cognitive Service-Oriented Infrastructure: An Application Example	41
3.2.1 Introduction	41

3.2.2	A Cognitive Service-oriented Infrastructure	43
3.2.3	Reasoning in Cognitive Service-Oriented Infrastructures: the Fuzzy Cognitive Maps	44
3.2.4	Validation	46
3.2.5	Conclusion	60
3.3	Cognitive Rate Adaptation in Wireless LANs	62
3.3.1	Introduction	62
3.3.2	Related Works	63
3.3.3	The Proposed Approach	64
3.3.4	Simulation Results and Discussion	73
3.3.5	Conclusion	79
3.4	Dynamic Green Self-Configuration of 3G Base Stations using FCMs	80
3.4.1	Introduction	80
3.4.2	Related Works	81
3.4.3	The Proposed Energy-efficient Architecture	83
3.4.4	Embedding Fuzzy Cognitive Maps in Radio Network Controllers	85
3.4.5	Simulation Scenario	95
3.4.6	Results	98
3.4.7	Conclusion	100
3.5	Dimensionality Reduction in Fuzzy Cognitive Maps	103
3.5.1	Introduction	103
3.5.2	Motivation	104
3.5.3	Fuzzy Cognitive Maps in Cognitive Processes	105
3.5.4	Identification of non-Relevant Cross-layer Relations	105
3.5.5	Test Case	110
3.5.6	Conclusion	113
4	Conclusion	117
	Bibliography	121
A	List of Publications	129
B	List of the acronyms used	131

List of Tables

3.1	Characteristics of the measuring process, depending on node location and layers involved	37
3.2	Domains of the input variables in the first scenario	48
3.3	Domains of the input variables in the second scenario	58
3.4	Cumulative amount of data transferred by the algorithms. Average over 20 random simulations. The proposed scheme is referred to Fuzzy Cognitive Map (FCM), in the case it does not use information related to the Signal-to-Noise Ratio (SNR), and FCMS, in the case it does.	78
3.5	Parameters utilized for the validation. Unity of measurement appears next to the value, surrounded by square brackets.	88
3.6	Simulation configuration parameters. d is the distance in km. Traffic modeling parameters obtained from [1].	97
3.7	Domains of the input variables	112
3.8	Cross-layer relations characterizing the test case analyzed	112
3.9	Performance achieved using different subsets of elements to perform the reasoning. Student's t distribution with 29 d.o.f., $p = 0.95$	112

List of Figures

1.1	Broadband and mobile subscriptions in OECD countries. Elaboration of the data published on www.oecd.org/sti/ICTindicators	2
1.2	Global IP traffic growth as forecasted by Cisco. Elaboration of the data published in [2]. Managed IP includes corporate IP WAN traffic and IP transport of television and video on demand.. . . .	3
1.3	The cognition loop (adapted from [3])	6
1.4	Alternative loops	6
1.5	Scope of: a) Software-defined radio, b) Cognitive radio, c) Cognitive network	6
1.6	Domains of cognitive radios, cognitive radio networks, and cognitive networks. Adapted from [3].	8
3.1	Fuzzy Cognitive Map example	29
3.2	FCM inference process example. As can be seen, the reasoning process results into the cycle limit $(1, 0, 0) \rightarrow (0, 1, 0)$ in three iterations. Concepts mapped to the $\{0, 1\}$ domain and threshold set to 0.5.	30
3.3	Merging multiple Fuzzy Cognitive Maps. Arrows intentionally omitted for clarity.	31
3.4	Possible evolution of the TCP congestion window and its transformation into binary concept.	34
3.5	Transformation of delay measurements into a concept. Threshold can be chosen as the maximum tolerable delay.	35
3.6	Cognitive networking and service-oriented architecture stacks (based on [4])	44
3.7	A Web Service Policy (WS-Policy) example	45
3.8	Validation steps. First, a database is populated with the simulations of all possible combinations of the input variables. Then, the prediction skills of the cognitive engine are tested against the simulation results.	47
3.9	Network topology in the first scenario. <i>AP</i> denotes the access point while $\{t_1, t_2, \dots, t_N\}$ are the terminals.	47
3.10	Temporal diagram of a possible service negotiation (time flows downward). Cognitive capabilities allow the service provider to publish a new service description, according to the current network conditions.	49

3.11	WS-Policy (based on [5]) describing the service provided in the first scenario under optimal network conditions: service consumers can opt for the audio quality they want (Application-level Packet Error Rate (PER) greater or smaller than 5%). When conditions worsen, and the cognitive entity is not able to effectively thwart wireless channel imperfections, the WS-Policy will be updated, for instance by removing the bold part, implying that the service provider cannot guarantee calls characterized by a PER lower than 5%.	50
3.12	FCM employed in the first scenario. e , d , r , i , and n stand for packet error rate, physical data rate, maximum number of retransmissions, voice packet interval, and number of calls supported, respectively.	52
3.13	Performance achieved enabling the FCM update at each step. $\eta = 0.5$. Compare with Figure 3.14.	53
3.14	Performance achieved enabling the FCM update only in case of prediction errors. $\eta = 0.5$. The sub-graph in the middle shows that, by updating the FCM only when errors occur, predictions are more reliable. Compare with Figure 3.13.	54
3.15	Prediction performance achieved by the reasoning entity in the first scenario. $\eta = 0.2$. When predictions are correct, the structure of the FCM does not change.	55
3.16	Throughput as a three-valued concept. $\eta = 0.1$	56
3.17	Impact of the learning parameter η on the prediction reliability. Confidence intervals computed with $p = 0.995$, DoF= 14.	57
3.18	Feedback example. The action profile tested at t_1 (high data rate, d , low voice packet interval, i , and low retransmission number, r) does not permit to achieve a greater number of calls while providing an acceptable quality. The action profile tested at t_2 (high d , high i , and high r), on the contrary, seems to allow the system to forward all the calls while providing the callers with an acceptable quality.	58
3.19	Network topology in the second scenario	58
3.20	The WS-Policy (based on [5]) describing the service provided in the second scenario states that users can be choose between two levels of performance, under optimal network conditions. In case the cognitive engine cannot keep network conditions at an optimal level, the WS-Policy may be republished to inform potential users that only a best effort service will be available (bold part removed).	59
3.21	FCM employed in the second scenario. t , n , e , d , r , and f stand for throughput, number of nodes, bit error rate, physical data rate, Request to Send (RTS)/Clear to Send (CTS) handshake, and fragmentation, respectively.	60
3.22	Prediction performance achieved by the reasoning entity in the second scenario – $\eta = 0.3$. When predictions are correct, the structure of the FCM does not change.	61

3.23	Specific implementation of the proposed cognitive data rate control mechanism. Station N2 gathers information about throughput, Frame Error Rate (FER) and Signal-to-Noise Ratio (SNR) that it sends to station N1.	65
3.24	High-level view of the proposed cognitive network architecture	67
3.25	Action model of a sensing variable. Different groupings identify different functions, as implemented in the simulator. Function names are reported as labels for each grouping.	69
3.26	Action model of the cognitive entity. The striped background indicates activities that logically belong to the sensing entity, but are implemented in the cognitive entity for convenience. The squared background indicates activities that belong to the reasoning formalism and may differ depending on the specific formalism employed—they are reported in the diagram for completeness. Dashed actions are not mandatory.	71
3.27	Specific implementation of the mechanism to avoid sub-optimal solutions based on a three-step counter. Every time the timer expires the counter is decremented, and different actions are taken. Different time intervals are used, in order to limit the use of the reinforcement mechanism.	72
3.28	Simulation setup. Numbers along the paths indicate the second in which the node started and stopped. For instance, interfering node A started moving at 0 seconds, stopped at 120, started again at 230 and finally stopped completely at 350 seconds. The shadowed area around nodes of interest N1 and N2 represents the transmission range.	73
3.29	Throughput comparison with other data rate adjustment algorithms.	75
3.30	Details of the comparison with other data rate adjustment algorithms.	76
3.31	(a) Evolution of throughput and data rate as a function of time. (b) Evolution of causality as a function of time. The letters d , e , and t stand respectively for data rate, errors FER, and throughput, respectively.	77
3.32	How the proposed architecture fits in a UTRAN. Node B monitors the environment and sends the data to the Radio Network Controller (RNC), which upon reasoning and learning, drives the acting modules in Node B, thereby closing the cognitive loop.	84
3.33	Relations among the concept sets and the FCM.	87
3.34	Extreme-case time evolutions of variable values and thresholds	89
3.35	Lagged-coordinated plot of the evolution of an FCM edge when updated by different learning algorithms. Starting point is $(0, 0)$. The sequence of variations is as follows: 2 positive, 10 negative, and 3 positive. Learning rate η set to 0.25, unless otherwise stated.	93
3.36	Differentiation of multiple solutions through the assignment of a performance score ($\lambda = 1, \mu = 2$). At $t = 9$, solutions that could improve en and snr , en but not snr , snr but not en , neither en nor snr , would score 1.13, 0.13, 1, and 0, respectively.	94
3.37	Layout of the simulated scenario. The coverage radius r is set to 600 m in the simulations.	95

3.38	Hourly variation of traffic load as a percentage of busy hour load over a typical day for a mobile network operator in London, UK.	96
3.39	(a) Consumed energy in traditional and cognitive base stations, and (b) energy saved by employing the cognitive scheme over a period of 72 hours. (c): example of the evolution of the blocking rate.	99
3.40	Evolution of the action concepts: (a) use of higher frequencies (<i>hi</i>) and (b) use of tri-sectorized mode (<i>tri</i>)	101
3.41	Example of evolution of the causal relationships: (a) between <i>hi</i> (use of higher frequencies) and the quality-related concepts, and (b) between <i>tri</i> (use of tri-sectorized mode) and the quality-related concepts.	102
3.42	Toy example showing the first steps in the application of the Logic Circuit Minimization approach with $c = 3$ and $n = 6$. First, the elements of the FCM are serialized. Then, reasoning is performed considering different combinations of the FCM elements and assigned a binary value indicating the performance level obtained. For instance, considering f_{ca} alone, or f_{ca} and f_{cb} (second and third combination) allowed the reasoning process to achieve an acceptable performance level, whereas considering f_{cb} only did not.	106
3.43	Decomposition as AND-expressions of the logic formula B , hypothetically synthesizing the Karnaugh map resulting from the toy example depicted in Figure 3.42	107
3.44	Extreme-case time evolutions of a cause-effect relation	108
3.45	Test case network topology	110
3.46	Test case Fuzzy Cognitive Map. Concepts t , n , e , d , r , and f stand for throughput, number of nodes, error rate, physical data rate, RTS/CTS handshake, and fragmentation, respectively.	111
3.47	Karnaugh map of the test case (reduced for better clarity)	111
3.48	Evolution of the discriminating index of the cross-layer relations in the test case. For abbreviations, refer to Table 3.8.	114

Chapter 1

Introduction

This chapter describes the research problem at a high level and glances at the potential solutions.

Section 1.1 will describe the context of the research problem and will discuss its positioning in the area of information and communication technologies. The problem itself will be analyzed in Section 1.2, while Section 1.3 will hint at possible ways to solve it. We will review in Section 1.4 the innovative aspects that characterize the work presented in this document and we will lay out its structure in Section 1.5.

1.1 The Context

In the last few decades, communication networks have grown considerably in size, and the growth has not stopped yet. According to a recent report on information technology by the Organisation for Economic Co-operation and Development (OECD), among the OECD economies the development of the broadband infrastructure still retains high priority “both in general terms and as part of the economic recovery” [6]. Indeed, the data shown in Figure 1.1a indicate that, though in the most recent years the annual rate has sensibly decreased, the total number of broadband subscriptions is still rising.

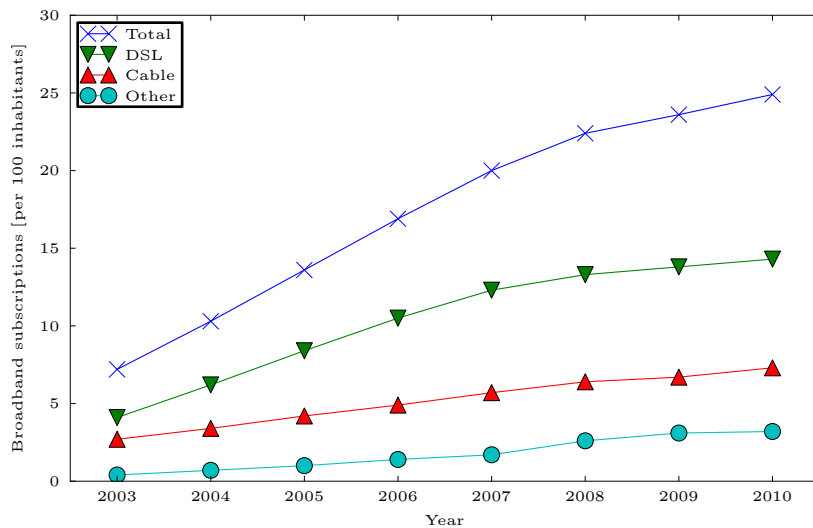
Similar considerations can be drawn by analyzing the evolution of the number of mobile subscribers in the same period, as shown in Figure 1.1b. In this area of communications as well, figures seem bound to increase in the next future.

Given these premises, it is clear that global traffic is also expected to grow, and, according to the estimates performed by Cisco Systems and reported in Figure 1.2, such growth will likely be exponential in time [2].

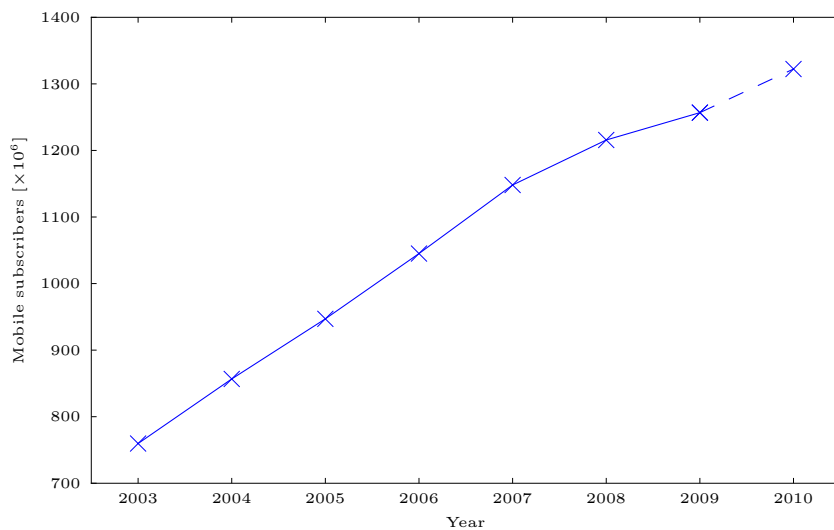
In other words, as both the amount of end users and traffic volumes are bound to rise, nodes both at the edges of networks and in the core networks will inevitably expand in number, and the amount of interactions among said nodes will continue to rise accordingly. The number of such interactions, potentially increasing as the square of the number of nodes, has caused networks to become more complex and dynamic—and still does so.

Evidence of both higher complexity and greater dynamics can be easily recognized in today’s networks.

One of the aspects of complexity that can be observed in networks is known as emer-



(a) Average number of broadband subscriptions per 100 inhabitants in the OECD countries.



(b) Total number of mobile subscribers in the OECD countries (in millions). Data for year 2010 is projected.

Figure 1.1: Broadband and mobile subscriptions in OECD countries. Elaboration of the data published on www.oecd.org/sti/ICTindicators.

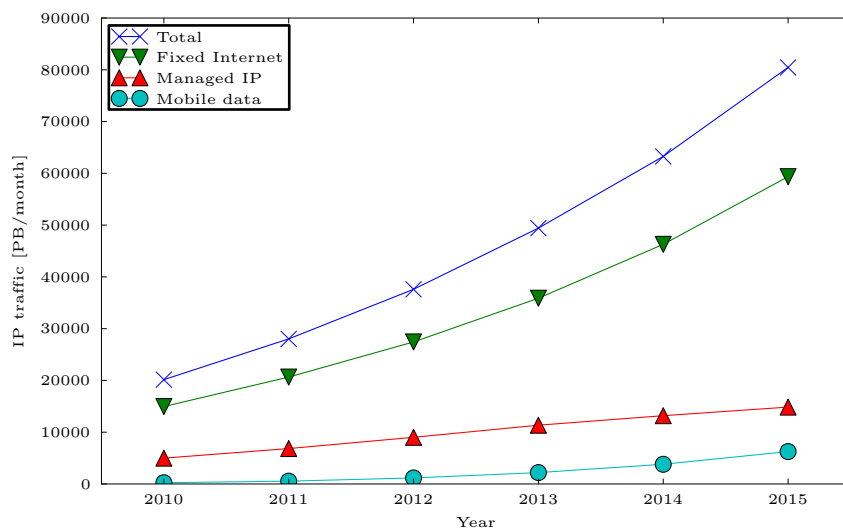


Figure 1.2: Global IP traffic growth as forecasted by Cisco. Elaboration of the data published in [2]. Managed IP includes corporate IP WAN traffic and IP transport of television and video on demand.

gence. As a general term, emergence indicates that complex evolution patterns may appear from multiple, possibly non-complex, interactions. Specifically for the networking research area, emergence means that to fully understand a network, not only its nodes must be studied, but also the interactions among them. More importantly, at a finer scale, even the behavior of a node cannot be completely explained by studying separately the protocols it is made of without taking into account the relationships existing among them.

The evolution of network dynamics is the other aspect we mentioned. Early networks can be seen as quasi-static environments, the most peculiar features of which being the homogeneity of devices and the fact that they were mostly wired. Heterogeneity of network devices and communication channels—among which we should most notably recall the introduction of wireless communications—and more recently, heterogeneity of applications, along with the already cited increase in size, made networks start becoming more and more dynamic.

The most evident direct consequence is undoubtedly an increased difficulty in managing and tuning networks so as to achieve and continually maintain optimal levels of performance.

1.2 The Problem

Albeit it is a fact that networks have grown in terms of size, and consequently, in terms of complexity and dynamics, no major breakthroughs have been introduced in network management.

Let us consider, for instance, that the principles of operation upon which today's

networks are based have substantially remained unchanged. The fact that the seven-layer ISO/OSI “model of architecture for connecting open systems” [7] is still the de facto reference model is a fitting example in this sense. The motivation behind such a design choice is to be sought in compatibility. The adoption of the ISO/OSI model ensured compatibility throughout the years, ultimately favoring modest yet repeated modifications and improvements over a clean-slate approach that would otherwise not allow legacy technology to function anymore.

However, although it is acknowledged that these principles are sound, it is likewise evident that they carry intrinsic implications that allow the creation of networks performing optimally only in static and well-defined environments, such as they were in the beginning. On the other hand, they are unable to adapt effectively to complex and dynamic environments where conditions keep changing. This represents a limitation, in that in such environments only sub-optimal performance can be achieved. A relevant example in this sense is the behavior of transport protocols on either wireless or high bandwidth-delay product links. As they were conceived, in none of these environment they can reach optimal performance [8].

Two crucial observations should be discussed at this point.

The first observation regards timing. Operations at different levels of the stack are characterized by different timings. Protocols at the link layer act upon receiving feedback from the next recipient, which is usually among the closest neighbors. On the contrary, transport protocols receive feedback after the so-called round-trip time, i.e. the time it takes for a packet to travel from source to destination (at the same layer) and back. The destination in this case may be placed several hops away, even in another continent. In general, as we go down the protocol stack, feedback times tend to reduce and become as short as fractions of seconds. The need to react within this timescale definitely rules out human intervention.

The second observation is about determinism. Even though intervention by human beings is not an option, if the environment shows to be, to some extent, deterministic, one could rely on algorithms to perform continuous optimization. This could well be the case for early networks. However in today’s networks, as we pointed out, increased complexity and dynamics concur in increasing unpredictability. Essentially, this means that algorithms incapable of adaptation are likely to fail in such environments.

1.3 The Solution

New network paradigms were recently proposed to lower the complexity of network management and to achieve better performance in dynamic environments. A noteworthy approach in this sense—the principles of which are at the basis of this work—is known as the cognitive networking paradigm [9]. To fully understand its origins we have to take a few steps back in history, at the time software (and software-defined) radios were born.

The first mention of the term software radio can be traced back to 1985, when it was used to indicate an “ultra-fast data processor, configured as a digital radio receiver” that had been developed within the Space System Technology Group, at Garland, Texas [10]. The term software-defined radio (SDR) is allegedly attributed to Mitola, who, years

later, refined the original concept to represent the ensemble of: (i) a set of Digital Signal Processing (DSP) primitives, (ii) a system to combine said primitives into communication-related functions, and (iii) a set of digital processors to be used as hosts [11]. Today, the term is used to refer more simply to radio transceivers, in which functions that are typically implemented in hardware are implemented as software.

Independently from the definition, the implications of such a revolutionary system are paramount, flexibility likely being the most important. For example, let us consider the transmission of signals using different modulation techniques while exploiting the same hardware, or the opportunity to employ algorithms (e.g. to operate at different frequencies) that would not otherwise be possible to include.

In 1999 a novel technology, building on the advantages brought by Software Defined Radios (SDRs) and attempting to add intelligence to flexibility, was proposed and labeled cognitive radio. Cognitive radios are reconfigurable radio devices based on SDRs that, in addition, are able to perform a cognitive process: they reason about the environment they are placed in and act in order to maximize spectrum utilization, all while continuously learning [12]. This process, graphically represented in Figure 1.3, is usually referred to as the cognitive loop, and is peculiar not only to cognitive radios but also to cognitive networks (and, to a certain extent, to cognitive architectures in general). It must be said that different representations of said loop exist [13, 12]. However, barring minor nuances, all versions share the same steps shown in Figure 1.3. Different versions appear in other fields as well. Connoisseurs of the military art may have already heard of Col. Boyde's OODA loop in Figure 1.4a, devised to understand opponents' moves [13]. Researchers in the autonomic area sure know about the existence of the autonomic loop in Figure 1.4b [14]. Finally, keen observers may have noticed some resemblances with Deming's cycle for quality improvement (Figure 1.4c). Notably, the sequence of stages described by each loop is the process human beings normally go through when coping with a problem. Indeed, it is no wonder that all such loops stem from the scientific method, according to which hypotheses are made (plan/decide), experiments performed (act/execute/do), and results evaluated (sense/observe/monitor/check).

An important aspect to stress is that, in cognitive radios only a part of the node is intelligent: the physical and the medium access layer, i.e. at most the two layers at the bottom of the seven-layer ISO/OSI reference model, representing the interface between a network node and the communication medium. The network perspective is generally not considered. The approach aiming to fill this gap was proposed in 2005 under the name cognitive networking [13].

Actually, in cognitive networks the idea is not restricted to include just the network layer in the cognitive process. Rather the aim is to add intelligence to network nodes in their entirety and not to just parts of them. For better clarity, the reader is referred to Figure 1.5, which compares the scope of SDRs, cognitive radios and cognitive networks.

Another fundamental difference between cognitive radios and networks partly derives from the different scope that characterizes each technology. Whereas cognitive radios aim to local optimization, cognitive networks are characterized by a network-wide perspective, thanks to which they attempt to realize global optimization. In other terms, the cognitive process peculiar to cognitive networks operates in view of some defined end-to-end goals.

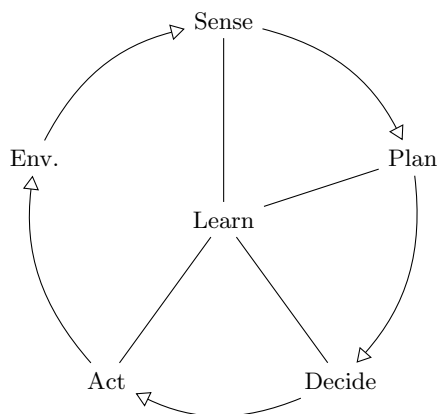


Figure 1.3: The cognition loop (adapted from [3])

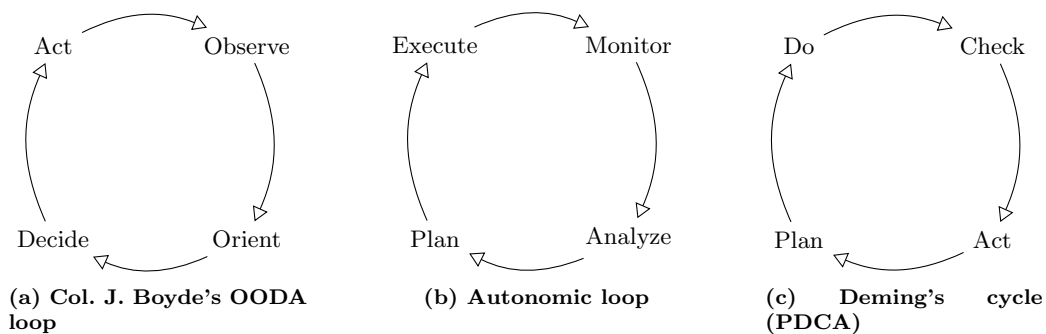


Figure 1.4: Alternative loops

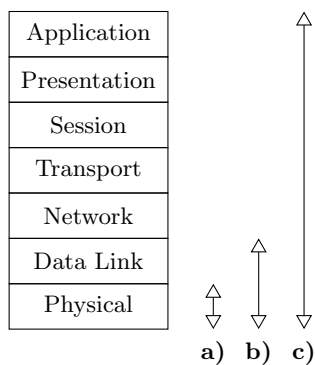


Figure 1.5: Scope of: a) Software-defined radio, b) Cognitive radio, c) Cognitive network

For completeness, another movement should also be mentioned, that goes by the name of autonomic communication [15, 8]. It represents an evolution of the autonomic computing concept, first stated by IBM [16]. The target is to have systems (autonomic computing) and networks (autonomic communication) manage themselves and adapt as the environment evolves, so that the overall management complexity is significantly lowered. The underlying idea is based on the concept governing human body's vital functions: in the same way the nervous system takes care of such vital functions without ever involving the brain, a network should be able to manage itself, thus relieving technicians from having to deal with the burden of network management.

The areas investigated by either the paradigms presented overlap to some extent with one another and this is reflected in a noteworthy confusion in the terminology adopted in the literature. Different research groups tend to identify their work with a paradigm instead of the other, often to stress some precise characteristics (the ability to reason and learn, and the end-to-end performance, in the case of cognitive networking, or the self-management aspect, in the other). However, most works can be thought of as belonging to both the paradigms, since often being able to self-manage implies the capability to reason, and vice versa.

Fortuna and Mohorčič [3] have suggested considering autonomic those networks that use a mere algorithmic approach to perform their tasks. Networks where nodes are required to be capable of reasoning and learning should instead be called cognitive. Significant in this sense and similar in spirit is the position of Clark *et al.* in their vision paper [17], in which it is stated that cognitive techniques are somewhat in contrast with algorithmic approaches. Indeed, algorithmic solutions should not be regarded as cognitive, as opposed to approaches that are capable of learning through experience. Throughout the remainder of this work we will adhere to this classification and consider cognitive those networks that reflect this consideration.

Finally, two remarks should be made. First, it is of utmost importance to draw a clear distinction between cognitive networks and cognitive radio networks. A great difference lies in the domain they belong to: the former can be either wired or wireless, whereas the latter only wireless. This concept is represented graphically in Figure 1.6¹. An even greater difference concerns the scope: while cognitive radio network may refer to a bare ensemble of cognitive radios and may hence describe a network in which devices try to improve performance only locally, in a cognitive network actions always strive towards better global performance. This, in particular, means that not necessarily a cognitive radio network is also a cognitive network.

Second, it is a fact that cross-layer techniques are utilized in virtually any cognitive network implementation. Nevertheless, cognitive networking must absolutely not be identified simply with cross-layering. What distinguishes the two concepts is adaptation combined with learning: while cross-layering can be seen as a memoryless technique for spreading information across the layers of a node's protocol stack aiming (at most) at "static" stack-wide optimization, cognitive networking aims at network-wide optimization through continuous adaptation and learning.

¹The wireless access point icon is copyright of DevCom. All other icons are copyright of Ben Fleming (Media Design).

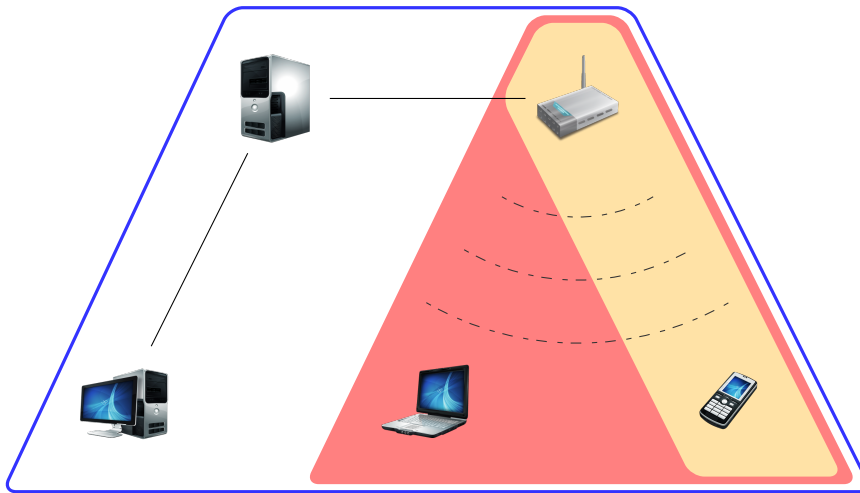


Figure 1.6: Domain of cognitive radios ■, cognitive radio networks ■, and cognitive networks □. Adapted from [3].

1.4 Innovative Aspects of the Thesis

Though the cognitive networking research area is very recent, several works already exist in the literature. Among them, some theorize about the best tools a cognitive entity should use to carry out their optimization tasks [18, 13]. Others, instead, adopt specific reasoning techniques, but often do not justify the choice behind such techniques [3].

This work centers around a reasoning formalism known as Fuzzy Cognitive Map (FCM), which can be used to perform causal reasoning. The first innovative aspect lies in the adoption of such a tool to explicitly exploit cause-effect relationships among variables in the protocol stack of network nodes. The motivation behind this choice is grounded on the idea that variables at different layers of the protocol stack share cause-effect relationships. Indeed, though according to the layering principles protocols at one layer should be independent from one another, what happens in reality is that actions performed at one level may lead to variations in others. The fact itself to use cross-layer relationships as the basis of reasoning was not discussed until now, to the best of the author's knowledge. In addition, it appears that the cognitive networks proposed in the literature, when addressing the reasoning problem, do not explicitly consider cross-layer relations.

Novelty is not restricted to the sheer application of this tool. Indeed, as the reasoning formalism in question originates from the need to analyze problems in fuzzy domains, such as social science [19], a formal approach to embed it in network nodes was missing. The second innovative aspect of this work is the development and illustration of such general approach.

Finally, FCMs were devised to be used by human beings. A typical example of application would be a person reading a political document: thanks to FCMs she would be able to draw and understand causal implications underlying the problem presented in the text. The key point here is that the analysis has been made possible because *she herself* drew the causal relationships and framed the problem according to the formalism. In cognitive networking, there should be minimal, if at all, intervention by human beings. Conse-

quently, for the formalism to work without any external intervention, techniques have been devised and introduced in this work, which represent the third element of novelty.

1.5 Structure of the Thesis

The rest of the thesis is structured as follows. In Chapter 2 we discuss the state of the art in the cognitive networking research area. We will learn what has been proposed so far in the literature to cope with the research problem described in Section 1.2. With Chapter 3 we move into the core of this document, and learn the details of this research work. Specifically, Section 3.1 introduces the reasoning formalism chosen for this work. Then, two sections are devoted to show the potential of the tool: Section 3.2 demonstrates how it can be used for prediction purposes, whereas Section 3.3 presents an application scenario in which reasoning is directed to enable intelligent actions. Refinements that can be applied to the system are discussed in Section 3.4, while Section 3.5 focuses on the reduction of timings needed to carry out the reasoning process. Finally, concluding remarks and insights about future possible research directions are provided in Chapter 4.

Chapter 2

State of the Art

So far several architectural designs have been presented in the research community of cognitive networking. Likely due to the novelty of the topic, several proposals aim to provide design guidelines by presenting general cognitive architectures, rather than showing a specific implementation. Section 2.1 presents the most relevant works in this sense and discusses some of the indications given. Section 2.2 analyzes some specific aspects of the research challenges present in the field and Section 2.3 assesses some possible ways to tackle them. Finally, we identify in Section 2.4 some of the primary research directions to be explored and draw the conclusion in Section 2.5.

2.1 Cognitive Architectures and Guiding Principles

The pioneer work envisioning cognition-enhanced networks has been undertaken by Clark *et al.* [17], who focus primarily on the ability of a network to self-recover. Motivated by the fact that further improving already existent algorithms is not enough to achieve such an objective, they suggest to revise the networking paradigm by introducing a key component that they call “Knowledge Plane” (KP). The KP they conceive is basically a distributed cognitive system equipped with reasoning and learning capabilities and designed as a closed-loop control system. It is in charge of creating and maintaining a model of the network at a high-level point of view and its ultimate purpose should be that of being able to properly advise network elements about what to do in specific situations. According to the researchers’ view, the KP spans both vertically through the protocol stack and horizontally across the network nodes and should be designed taking into consideration some precise tenets. Specifically, the KP should:

- take advantage of the different observations that can be made in different points of the network, that is, it should exploit observation diversity;
- implement a unified approach to solve problems, avoiding ad-hoc solutions, which may appear simpler at first but sub-optimal in the long run;
- include network edges, in order to exploit their knowledge;

- be able to function in dynamical, continuously changing environments, also in the presence of misleading and/or incomplete information, and under conflicting high-level goals.

It is important to note that these basic principles are shared by virtually any other cognitive-network-related work present in the literature. As a matter of fact, Clark’s KP is a possible instantiation of the already mentioned cognitive loop, depicted in Figure 1.3.

The researchers also envision fairly advanced features for their architecture, such as the ability of the KP to reason by itself on information trade-off and routing, i.e. understand when local information is to be preferred to measurements of the global situation of the network, and understand what and where knowledge is needed in given situations. Interestingly, though, they admit of the possibility for the cognitive architecture to be driven by human input. Indeed they advance the hypothesis that there might be occasions in which only a person would be able to discern the best action to take and give the cognitive system proper guidance.

In summary, however, there seems to be no agreement in the literature on which cognitive network model should be taken as a reference. Rather, there is a proliferation of architectures based on the KP concept, with minor adjustments.

The first formal research about cognitive networks has been performed by Thomas *et al.*, who, with regard to the characteristics a cognitive architecture should have, list: (i) extensibility, flexibility, and proactivity; (ii) the capability by the decision process to use network metrics as input and provide actions as output; (iii) the capability to achieve higher performance levels with respect to traditional networks. Besides offering a definition for this new paradigm, also delineate a possible framework for the development of cognitive networks, logically divided in three layers, mirroring the defining aspects of a generic cognitive entity (behavioral, computational, neuro-physical) [13]. Specifically, the guiding end-to-end objectives are specified at the top layer, where they are also redefined in terms of local objectives. Such local objectives are successively handed to the middle (cognitive) layer, which reasons about them and selects the appropriate actions to take. According to the actions selected, the bottom layer sets the tunable elements it controls. It also senses the environment and reports to the cognitive layer, in order to make it learn, thereby closing the cognitive loop. A remarkable characteristic of this cognitive network model is that no strict rules are defined about how the three layers should be mapped with respect to the actual network nodes. This means that, for instance, the cognitive process may be performed by just one cognitive element, comprising all the devices of a network, or by multiple elements, each one in charge of controlling just a part of the ensemble. In other words, no design limits are imposed. Insightful is also the evaluation of network performance as a function of some characteristics of the cognitive nodes, to wit, the degree of selfishness, of information, and of control on the other elements they have: the researchers demonstrate that not always characteristics that could be deemed beneficial, like altruism or perfect knowledge of the environment, lead to better results.

Several architectures are built on the belief that a more holistic approach is needed. Mähönen *et al.* [18], for instance, acknowledge that information has to be extracted from all layers but assert that so far no attempts towards the joint analysis of the behavior of a communication system and its cross-layer interactions have been made. To them,

the cognitive networking paradigm is particularly appealing, as it represents a valid alternative to basic cross-layer optimization. Emphasis is also given on the collaborative aspect: according to the researchers, a local view of the network is simply not sufficient to perform global optimization. Collaboration is, therefore, fundamental to perform distributed information gathering with the objective to build a comprehensive knowledge of the environment. Lastly, it is interesting to note that the architecture they propose is heavily based on that by Clark *et al.* [17], with the difference that, in their opinion, both the sensing and the reasoning parts have to be executed in real-time.

The thesis of holism is supported also by Sutton *et al.* [20], who hold that the limited scope of communication protocols prevents a node from performing a global optimization. In this sense, they propose an architecture characterized by a network-wide scope, similar to that peculiar to the already mentioned KP, extending over both the protocol stack and the network itself. An interesting point of such architecture is that the cognitive process is explicitly split between two distinct entities: a reconfigurable entity is in charge of manipulating sensors and actuators and a separate cognitive entity takes care of reasoning and learning tasks. This separation may give rise to concerns regarding how commands and measurements flow between the cognitive entity and the reconfigurable entity, and how much should one expect communication overhead to be. However, no specific details are offered in merit. According to their vision, the cognitive entity can comprise all the layers of a node or even just a subset of them, whereas the reconfigurable entity involves reconfigurations not only inside a single node but possibly also in each node of the network. It is not clear, though, whether more than one cognitive entity is allowed in a network or not. Nor it is clear how, in the case there exist multiple cognitive entities, problems related to multiple reconfiguration issued by different cognitive entities can be tackled, or, in more general terms, how cooperation should be achieved.

Another project, named CogNet [21], aims to use all the information each layer can provide. A particularly interesting feature of this model with respect to the ones previously analyzed is the introduction of a memory. Thanks to it, this architecture possesses the ability to exploit the temporal and spatial characteristics associated with information gathered from the various layers and use a repository where experience can be stored and consulted. The cognitive process in this case is accomplished by an entity they refer to as the CogPlane, which runs joint-layer optimization algorithms, distributed throughout the network. However, in the test case provided, the situation when multiple cognitive planes interact with one another is not investigated.

The m@ANGEL project [22] is worth a mention because of its particular area of application. It focuses on the use of cognition for managing infrastructure wireless networks, and specifically, it concerns the development of a cognitive access architecture to be used in the so-called “Beyond 3G” era, where it is foreseen that wireless communications will converge to a common access infrastructure. In such an access network, reconfiguration may happen at all layers; the core network, on the other hand, is not meant to be cognitive at all. In the authors’ view, cognitive access networks represent a means to overcome the issues that would arise if networks were purely based on cooperation, e.g. too high inter-dependence and too high a level of coordination required among network operators. However, the authors also specify that, with respect to cooperative networking, cognitive

networking is a complementary rather than a supplementary technique, and therefore co-operation schemes between network operators still have to be present, at least to a certain degree.

Finally, the architecture designed by Raychaudhuri *et al.* [23] should also be cited, although tailored for the evaluation and design of cognitive radio networks rather than cognitive networks. Precisely because it is targeted at cognitive radio networks, it is interesting to note that it explicitly considers not only the physical and data link layers but also the network layer. This has a twofold implication. First, it implies that layer-three protocols (and, in general, protocols at even higher layers) should not be thought of as being independent from the rest and should be included in the cross-layer optimization procedure, so as to achieve joint optimization of the stack. Second, it is a clear sign that also cognitive radio network research is leaning toward the higher layers of the stack, thus trying to include the end-to-end scope peculiar to cognitive networks.

As a side note, it is significant to observe that the research area of cognitive networking (and adapting/intelligent networking in general) aroused increasing interest in the last few years. Meaningful in this sense are the ARAGORN project [24], funded through the seventh Framework Programme (FP) and focusing on both cognitive radios and cognitive networks, and the End-to-End Reconfigurability project (E²R) [25] and its continuation, the End-to-End Efficiency project (E³) [26], funded respectively through the sixth and the seventh FPs and studying the cognitive networking paradigm for the previously mentioned Beyond 3G era.

Another representative initiative is the even more recent approval of the COST Action IC0902, devoted to the study, in the mid-term, and to the standardization, in the long-term, of cognitive networks [27].

2.2 Research Challenges

Research in the area of cognitive networking is at an early stage [3, 27]. Consequently, several problems still have to be resolved. Among the greatest challenges are reasoning and learning, adaptation, and information representation.

2.2.1 Reasoning and Learning

Despite the wide consensus in the literature about the need for holistically considering network nodes, many examples provided seem not to consider nodes as wholes. Some of the works outlining cognitive architectures stress the importance of involving all the layers of the communication stack, hinting at requirements and constraints to be respected, without however offering sample implementations [18, 22, 20]. Other works present cognitive entities that span over all the layers of the stack but validate it considering only a reduced subset of these layers, and, as a consequence, exploiting a limited subset of the information available [13]. Even more important, to the best of the author's knowledge, no research has specifically taken into consideration cross-layer interactions in the reasoning process, fundamental to fully enable a truly holistic approach: indeed, such interactions, be they implicit or explicit, concur in defining the overall performance of a network and

therefore are to be considered when performing the cognitive process.

To cope with these challenges, an appropriate reasoning technique has to be chosen. However, as noted in [3], in the networking architectures proposed in the literature only a few reasoning techniques are employed and often their use is not completely justified over the available alternatives.

Reasoning exploits the knowledge that an entity has been able to build by learning to draw new inferences and beliefs [17]. Similar is the definition by Langley *et al.* [28], who state that to reason is to infer conclusions starting from beliefs that are already present in an agent's mind. Therefore, a necessary condition is for a cognitive entity to possess the capability (i) to draw relationships among such beliefs and (ii) to use them in order to derive some conclusions.

Thomas *et al.* suggest to think of cognition as a machine learning problem [29]. Formalisms like neural networks, genetic algorithms, expert systems, and control theory (e.g., Kalman filters and learning automata) can therefore be employed [13]. However, rather than on analyzing the theoretic aspect of cognition, they focus more on providing practical guidelines for bringing intelligence in communication networks, by arguing that a cognitive process need not involve exactly one reasoning strategy. On the contrary, depending on the type of problem to be solved and the objectives to be achieved, more than one technique can be used, possibly creating hybrids.

Along the same lines is the opinion of Mähönen *et al.*, who propose to equip cognitive network nodes with an ensemble of tools to perform reasoning tasks [18]. Besides the already mentioned neural and Bayesian networks, also pattern recognition and classification techniques, and multidimensional optimization algorithms, like simulated annealing or genetic algorithms, are included in such a toolbox. The authors, though, give no indication about what technique should be preferred in a given situation nor on what basis it should be selected.

Eventually, independently from the technique used, what seems to really matter is convergence time: whatever the solution chosen, it is of utmost importance that its convergence time be less than any environmental change time [13].

Another key research topic is learning, which can be seen as a planned and ordered gathering of knowledge, and can happen in several ways. Being strictly connected to reasoning, it is given little attention as a topic on its own. Besides, learning techniques are usually straightforward and reflect those techniques actually used by living organisms. According to [17], knowledge can be built thanks to external input, the same way children learn at school thanks to their teachers, or by trial and error, as happens with people facing new problems when no one is available to give indications. Other techniques to boost learning cited are by analogy and by generalization.

2.2.2 Adaptation

Complementary to reasoning is adaptation [30], i.e. the process of turning reasoned decisions into actions.

More specifically, after reasoning, the node must adapt itself so that the decisions taken by the cognitive entity can be put in practice. However, being a cognitive network characterized by a global scope, it is possible that more than one node have to adapt

themselves jointly, in order for the effect to take place. Adaptation means therefore that nodes have to be able to coordinate themselves to carry out a common goal, or in more general terms, to extend the cognitive process to the whole network.

Network-wide adaptation in cognitive networks, though, seems not to have received much attention in the literature thus far. Instead, proposals have focused more on the need of inter-node communications (or lack thereof). It can be argued that inter-node communication may not be fundamental in cognitive networks, as there exist studies on cognitive networking that do not consider it. For instance, this is the case presented in [21], in which the centralized architecture of the wireless network analyzed suggests that information exchange among the base station and the wireless clients may be not necessary. However, other researchers plan instead to have their cognitive devices explicitly talk with other peers [22, 18], stating that the more the shared information, the higher the performance could be, in principle. This may not always be true, though, as in some scenarios better performance can be achieved when nodes do not share information with one another [9].

2.2.3 Information Representation

The topic of representation is partly related to both learning and adaptation. Many researchers agree on the need of a means to represent objectives and knowledge. Inevitably, pursuing the vision of a knowledge-centric network makes knowledge representation fairly important [17]. Information representation plays a key role also in optimization processes: as optimization can be performed with respect to different, possibly contrasting, aspects (e.g. performance, reliability), representation is necessary to establish priorities and decide to which dimension optimization should aim [14]. Goals of the system, as well, should be clearly expressed so that no misinterpretation can take place. However, even when goals were not correctly specified, the system should be able to act reasonably well anyway [16].

Linked to information representation is information propagation, both within nodes and across the network. As a general guideline, a desirable feature of the information propagation scheme to be used is that it should be independent from the underlying communication technology [14].

As for its implementation, it can be analyzed from both the semantic and the syntactic points of view. The semantic questions concern *what* information we want network nodes to exchange. Updating all the nodes in a network with all the network knowledge may not be the best option, mainly for two reasons: first, the waste of resources may be too high and second, an “information overload” may be burdensome, not to say misleading, for the reasoning process. To this end, it is imperative to understand which nodes need what kind of information, so that can be defined what to pass and to whom [17]. Syntax is about *how* messages should be passed and how they should be defined. This translates into the definition of ad-hoc protocols that have to be efficient from the point of view of resource consumption, and explicitly or implicitly negotiate and ask for resource reservation. All this while also paying particular attention to timings, that is, *when* messages should be passed.

2.3 State-of-the-art Solutions

Several techniques can be used as the reasoning core of a cognitive entity, yet some of them could be more fitting than others. The characteristics of the main potential reasoning formalisms are surveyed in Section 2.3.1. In Section 2.3.2, instead, we illustrate how literature copes with the issues related to adaptation and we conclude the analysis in Section 2.3.3 by identifying how information is represented and exchanged.

2.3.1 Reasoning and Learning

A popular technique used to infer general conclusions is represented by first-order logic [28] and is typically used by agents who can rely on certain knowledge [3]. However, communication networks can hardly be classified as certain-knowledge domains. It is not unlikely to foresee situations, e.g. in wireless environments or wired peer-to-peer scenarios, in which nodes go on and off at any time, causing abrupt changes in topologies. Nor it is uncommon that services offered by servers become suddenly unavailable with little notice, if any at all. As a consequence, employing first-order logic schemes in cognitive nodes may not appear as the best choice. This seems also to be acknowledged in the literature, since, to the best of the author's knowledge, there is no cognitive network architecture based on such a formalism.

Some researchers deem expert systems, i.e. systems aiming to store human experts' knowledge in a specific field, useful to perform reasoning in cognitive network, provided the problem to be solved is characterized by a limited number of variables [13]. Others, however, maintain that the overly narrow domain of application peculiar to expert systems clashes with the concept of cognitive architecture, which should instead seek to reason across a variety of diverse domains [28]. The author, however, does not have any knowledge of cognitive networks based on this paradigm.

Structural equation modeling, although not a proper artificial intelligence technique, has been mentioned as a potential formalism for reasoning purposes in [31]. However, the same authors admit that such a formalism is more suitable for confirming already defined causal structures (hypothesis testing), rather than discovering them. This could place a limit in cognitive entities, which may be prevented from adapting to new situations, thus not evolving over time.

Heuristic optimization algorithms, like simulated annealing, genetic algorithms or swarm intelligence [32], are used in order to automatically find optimal solutions, and can be likened, to some extent, to other reasoning methods. Examples of applications of genetic algorithms have been given for cognitive radio networks. Newman *et al.*, for instance, adopt such technique to build a cognitive radio decision engine with the final intent to determine optimal radio transmission parameters [33], while Friend *et al.* apply it to optimize dynamic spectrum allocation in a distributed environment [34]. However, some researchers assert that, although applicable in a variety of contexts, such techniques should be preferred when the environment is well-known and the problem is centralized [13]. For this reason, in distributed scenarios, like wireless ad-hoc networks where the nodes may be highly scattered over the environment, such techniques are generally less favorable with respect to others.

Neural networks are often considered as a standard artificial intelligence technique and, thus far, have been applied to a wide range of applications [35], including cognitive networks [3], where they have been used for carrying out cognitive routing functions [36, 37]. The main drawback lies in the fact that they are black boxes: once a neural network reaches a solution, its inner structure does not necessarily reflect the motivation behind that outcome [33]. In another way, the actual relationships that exist among the variables of a system are not reflected by the configuration of the neural network that led to the solution [31]. On the one hand, this may not represent a problem in case we, as designers, are not interested in the mechanisms of a network. On the other hand, if we want to gain some insights into a network's internals, so that we can understand potential problems and build more and more efficient solutions, neural networks are inapplicable.

Bayesian networks are another reasoning tool traditionally associated with artificial intelligence, their peculiarity being the capability of representing cause-effect relationships among variables of a given problem. Generally they are suited, as well as neural networks, to being applied when knowledge is uncertain [3]. As they are based on directed acyclic graphs, their major limitation lies in the impossibility to deal with causality loops [38, 31]. Most notably, Bayesian networks were already envisioned by Clark *et al.* in their pioneer work on cognitive networking [17].

Strictly related to Bayesian networks are Markov random fields. Similarly to Bayesian networks, they are generative models, i.e. they represent a probability model for all the variables of a problem; however, unlike Bayesian networks, they need not represent causal relationships. Other formalisms derived from Markov random fields are Markov logic networks, which combine Markov random fields and first-order logic, and conditional random fields, which are a discriminative type of model, i.e. they model the dependency of unobserved variables on observed variables. Markov random fields (and all models based on them), being represented by means of undirected graphs, suffer less from the limitation peculiar to Bayesian networks about loop-free networks: when loops are present in the problem structure, the inference process is performed by approximate algorithms (so-called loopy belief propagation), which, however, are not guaranteed to converge. It is also worth noting that the undirected nature of such structures prevents them to handle induced dependencies¹. As far as the author knows, no cognitive network is based on any of these models.

Fuzzy Cognitive Maps (FCMs) are mathematical structures for modeling dynamical systems. They emphasize the cause-effect relationships present among the variables of a system and upon them they base reasoning [19]. From a graphical point of view, they are direct graphs, in which nodes represent generic concepts (be them events, variables or other generic entities) and edges portray the cause-effect relationships among those concepts. FCMs offer some advantages over other potential reasoning techniques. First, differently from the edges in a neural network, those in an FCM reflect the actual relationships among the variables of the problem being inspected. Then, in opposition to the cases of both Bayesian and Markov networks, the inference procedure used in FCMs

¹Situation that happens when two nodes of the model are marginally independent but become conditionally dependent thanks to a third common node. For example, let A, B represent weather conditions in two independent cities and let C represent the quantity of fallen water in the two cities. A, B are marginally independent, but turn conditionally dependent given C .

is computationally light, involving only vector-by-matrix multiplications and threshold operations and can be applied independently from the presence of causality loops. Moreover, it is guaranteed to converge to a solution provided that the vertices have their value in a discrete set [39]. Finally, multiple FCMs can be exchanged and merged with one another (upon a weighting operation, if necessary), mimicking the exchange of opinions peculiar to human beings: this is an advantageous feature when dealing with uncertain knowledge, thus being better suited in such environments than other formalisms, such as, for instance, first-order logic. Nevertheless, FCMs are characterized by some drawbacks as well. The most limiting drawback is that abductive reasoning, i.e. the process of discovering the causes that led to some effects, is NP-hard, exactly as in the case of Bayesian networks [40].

With respect to learning, the techniques used so far in the cognitive networking field are usually straightforward, such as the use of a knowledge base to be updated as new situations are experienced [41, 21]. Specific techniques are used to update connections in neural networks and FCMs. Popular algorithms commonly employed in neural networks are backpropagation learning [37], in which error with respect to a target outcome is repeatedly back-propagated through the network, and reinforcement learning [14], in which a reinforcement signal is a measure of the performance level achieved after the system has performed a set of actions. As for FCMs, updating techniques are based on Hebbian learning, according to which connections between concepts that are activated together should be given more weight [42].

2.3.2 Adaptation

Papers in the literature provide hints at techniques that could be used to accomplish network-wide adaptation, but examples of implementations are limited.

Clark *et al.* have in mind a distributed and decentralized model that lets them partition the network so that different parts can work toward the achievement of divergent goals [17]. However, they acknowledge that classic artificial-intelligence techniques are not suited to be employed in distributed environments and suggest that robust, highly dynamic algorithms are needed.

They also mention multi-agent systems, a tool to model and design systems populated with intelligent agents, each capable of (and responsible with) its own actions, as a framework that could potentially represent a base for building such algorithms. However, they cautiously warn that this may not be the best solution, depending on the timing constraints posed by a network environment.

In [43], Friends *et al.* declare to be in favor of a distributed implementation of the cognitive process, claiming it is more convenient when compared to the centralized counterpart. Their preference stems not only from the typical advantages such a model offers, but also from the benefits it presents in the specific case of cognitive networking. First, a centralized architecture would lead to a greater communication overhead. Second, in environments populated by nodes that already have cognitive capabilities (e.g. where there are some cognitive radios), the centralization of cognition would inevitably lead to a greater waste of channel resources.

According to this point of view, Thomas proposes several formalisms to study the

behavior of a network [9]. Besides the already cited multi-agent systems, he advances the use of game theory, that aims to analyze the interactions occurring among a population of rational decision-makers who attempt to behave strategically. He makes use of game theory to devise a cognitive network, and, though acknowledging that such theory accurately captures the behavior of selfish agents, he urges that methods more apt to explore cooperative behavior should be sought. He does not consider cooperative games, where players are allowed to constitute coalitions and commit agreements. However, the author has no notion of cognitive networks in which cooperative game theory has been applied. One potential drawback of game theory is that, usually, assumptions are made in order to make game analysis tractable: common assumptions in this sense are a homogeneous player set (which means nodes have to share similar characteristic), or that players have complete information (that is, nodes exactly know what other nodes want and what they can do). Were such assumptions not acceptable, a game would likely become more complicated and, possibly, not presenting equilibrium points.

Another tool Thomas promotes as a means to investigate interacting elements is the set of so-called interaction models, thanks to which it should be feasible to identify fixed points of complex systems. Some of the models listed are infinite particle systems (like the Ising model, originally conceived as a model for the analysis of ferromagnetism), and Petri nets, to model general distributed discrete systems.

2.3.3 Information Representation

Unsurprisingly, most of the works offering practical insights on information representation, resort to the use of markup languages, such as the Extensible Markup Language (XML).

For example, though pertaining to the field of autonomic computing, an interesting solution comes from Kephart and Chess [16], for whom autonomic elements should register the services they offer in a public registry, such as the Universal Description, Discovery and Integration (UDDI). Ultimately, said registry will contain high-level descriptions of the objectives and policies offered by autonomic elements, and how they should be invoked.

Another suggestion is to use the Radio Knowledge Representation Language (RKRL), first promoted by Mitola [12, 13] with the objective to represent radio knowledge through the use of structured, yet natural, language.

Finally, the last scheme for the representation of knowledge worth mentioning is the DARPA Agent Markup Language (DAML) by the Defense Advanced Research Projects Agency (DARPA), that capitalizes on both XML and the Resource Description Framework (RDF) to support ontologies for web objects. Nevertheless, a caveat is necessary: whatever the scheme, valuable information would probably come from other places, such as in Management Information Bases (MIBs) [17].

As for information propagation, it is acknowledged that in order to foster cross-network communication a signaling architecture must first be devised [44]. However, to the best of the author's knowledge, so far no mechanism has been proposed specifically for cognitive networks, and only generalized guidelines have been discussed.

Regardless of the signaling method, two broad categories can be distinguished: either

we implement an in-band signaling scheme, where signaling messages are mixed with data messages, or an out-of-band signaling scheme, where messages constitute a separate communication. A signaling scheme belonging to the first category may be realized by embedding information in header fields that are normally not used by a protocol. Among the advantages of such a solution are the facts that no overhead is added and that it is possible to reuse already existent protocols. However, nodes along a path may misclassify such ‘enhanced’ packets as malicious or malformed and discard them. Other drawbacks are represented by the limited room to allocate information, and the fact that information passing becomes dependent on the communication technology used (i.e. the protocols). An out-of-band signaling scheme needs a separate communication setup (and possibly the design of a new protocol). The quantity of information that can be sent is greater than in the complementary scheme, but greater are also complexity and overhead.

Depending on the architecture chosen for the cognitive process, different message passing schemes can be deployed. In a centralized network a natural option would be that of a master-slave communication scheme, but direct communications among slaves may be desirable as well. In a distributed network, a naïve scheme could be that of flooding. Such an approach is not recommended, not only because of the great overhead, but also because not every node may be interested in receiving a particular message. For instance, two neighboring nodes may harbor the same beliefs about the environment: as a consequence neither of the two will benefit from the other’s information. Naturally, more elaborate protocols can be employed: as an example, a seemingly popular choice in distributed scenarios is represented by epidemic protocols, i.e. protocols that to spread information imitate the diffusion typical of diseases, as they are generally scalable and failure resistant [45, 17, 14].

2.4 Research Directions

It has been argued that interactions between protocols inside a node and among nodes themselves play a critical role in determining network performance. As a consequence, it becomes of paramount importance to evaluate which reasoning techniques are capable of dealing with both intra- and inter-node relationships, and among them, to select the most appropriate to be employed.

The general problem a cognitive node has to face is that of finding an action that, given a situation could lead to another, generally more advantageous, situation. By rephrasing the problem in terms of causality, we easily see that this is equivalent to finding a set of causes that could lead to some desired effects, which, in general, turns out to be an NP-hard problem, regardless of the reasoning formalism chosen. This may represent an issue when analyzed and compared to the ephemerality of the solution: as a solution must be found before the environment changes, it is very likely there will be only little time to devote to search. Besides, it is not straightforward to predict how much this time lapse will be. Supposedly, it could range from milliseconds, according to the coherence time of a wireless channel, to seconds, as the round-trip time of a transport level segment in a multi-hop chain-topology ad-hoc network, to even minutes, as the duration of a file transfer.

Techniques to reduce the dimensionality of the problem have, therefore, to be sought. Alternatively, parallel solutions may be promoted, such as storing the experience acquired in a long-term memory, to be consulted with the purpose of reducing search times and avoid local maxima.

Partly related is the fact that most of the reasoning formalisms surveyed in Section 2.3.1 do not consider the temporal dimension of causality, which could, in principle, represent a limitation when dealing with communication networks: since different layers are involved in the cognitive process, one could expect that causal relationships are characterized by different timings. In case this aspect represents a major concern in the reasoning phase, more sophisticated techniques or even completely different mathematical models may have to be employed.

The continuous update of the reasoning engine itself must also not be underestimated. Independently from the formalism employed, learning is fundamental to obtain accurate results over time. Proper learning techniques must be chosen, possibly depending on the context in which the cognitive entity is placed.

As for the adaptation/coordination aspect, neither the centralized nor the distributed model can be discarded a priori. Even more important, a cognitive network may feature both schemes and switch from one to the other in an adaptive manner. Ideally, the cognitive architecture should impose no restrictions on the communication mechanism, thus allowing full flexibility.

The same concept can be applied to cross-network information exchange. Also in this case, both in-band and out-of-band signaling schemes are to be investigated and neither of the two is to be a priori rejected. In demanding scenarios chances are that, due to their lower overhead, in-band schemes represent the best choice. If resources are not a problem, e.g. in optical or delay-tolerant networks, out-of-band schemes are likely to offer greater flexibility of application. The bottom line is the same as before: no constraints on signaling should be imposed by the cognitive architecture. Minor communication-related research questions are, for instance, whether signaling messages should be read only by the communication end-points or also by nodes along the way. In general, whether (and how) intermediate nodes would actually benefit from learning signaling information cannot be known beforehand and it likely depends on the application scenario.

Security is another direction carrying important implications, yet barely considered by the cognitive networking research community. Important, especially in view of possible real-world implementations, is to determine if and how much signaling schemes are prone to security issues, and understand if and how techniques such as authentication and encryption can be implemented, without impairing the overall performance.

Central is also the definition of formal indicators to measure the performance levels that characterize the proposed cognitive network architecture. This will allow network designers to precisely evaluate the performance levels of a cognitive network with respect to non-intelligent traditional networks. Moreover, performance (along with weak and strong points) of different cognitive network architectures could be identified and compared. In fact, it is possible that some cognitive architectures may perform better than others in certain situations. For this reason, following the introduction of such metrics, real-world scenarios must be carefully defined, so that the efficiency of cognitive networks can be

exhaustively evaluated.

2.5 Conclusion

The choice of a reasoning formalism to use as the base of a cognitive network is of great importance, as it can potentially affect all the other aspects, namely learning, adaptation, and information representation/passing.

Considering the research directions outlined in Section 2.4, Fuzzy Cognitive Maps, though characterized by some weak points, seem to offer some advantages over other reasoning techniques. Representing a valid candidate to address the research challenges of the research field, they have been chosen as the basis for this work.

In this context, techniques aimed at reducing reasoning time will focus on the elimination of unimportant edges and nodes. Likewise, learning will concentrate on the strengthening of the edges that best reflect real-world cause-effect relationships. Besides, thanks to the possibility to be exchanged and merged with other FCMs, they naturally lend themselves to be employed in multiple nodes, thus imposing no limitation on the vision of an all-distributed cognitive network.

Chapter 3 will deal with all these aspects, starting with the introduction of the chosen tool, its application in different scenarios, and the refinements that can be applied to adapt the formalism to different scenarios.

Chapter 3

The proposed approach

This chapter aims to describe the approach proposed to solve the problem analyzed in Section 1.2.

Section 3.1 will introduce a formalism known as Fuzzy Cognitive Map (FCM), created in the '80s as a tool to represent causal reasoning [19]. Initially devised to be used in “soft knowledge domains”, such as history and social science, throughout the years it has been applied to problems in different fields. The main purpose of this section is to familiarize with the tool and explain the steps of the conceived process that are needed in order to use such a tool for the cognitive networking paradigm. The basis of reasoning in this framework will be laid, as well as the fundamentals of learning.

An application example will be given in Section 3.2, in which the concepts illustrated so far will be further extended to devise a whole intelligent architecture, to which we will refer as Cognitive Service-Oriented Infrastructure (CoSOI). By analysis and simulation of this architecture in two wireless scenarios, centralized and decentralized, we will see how FCMs can empower cognitive networks.

Whereas the application examples in Section 3.2 are limited to prediction, i.e. they show the potential of FCMs to forecast future situations, Section 3.3 will take it a step further by enabling actions, thereby closing the cognitive loop (Figure 1.3). Specifically, this section will show that actions can be triggered automatically to achieve optimality in certain situations. The problem tackled here is that of rate adaptation in Wireless Local Area Networks (WLANs), in which the application of the cognitive formalism allows the rate to be adjusted automatically, depending on the environmental conditions. A thorough description of the implementation details completes the section.

As the efficacy of the tool is demonstrated, Section 3.4 dwells on some refinements that can be introduced in the system, at virtually all stages of the cognitive cycle. First, at the sensing level, the details of pre-processing operations such as time averages, discretization, and threshold operations will be analyzed; next, novel learning schemes will be explained; and finally, a technique will be shown for choosing a solution when multiple options are available. The application of this refinements has been validated via simulations of a mobile access network, in which the goal was to abate energy consumption while not increasing the blocking rate.

The work presented in Section 3.5 runs somewhat in parallel with what presented in the other sections and it is complementary, at the same time. As reasoning can be a

time-consuming task to perform if too many variables are taken into account, the idea developed here focuses on the identification of variables that do not add any information to the reasoning process and could, therefore, be discarded reducing reasoning times yet not affecting reasoning effectiveness. Simulations show that the algorithm proposed can indeed distinguish relevant from irrelevant concepts.

All sections in this chapter are loosely based on the author's published works. Section 3.1 contains part of work published in [46], Section 3.2 includes parts published in [47], Section 3.3 draws from [48] and [49], Section 3.4 is based on [50] and [51], while Section 3.5 on [52].

3.1 Towards a Model for Quantitative Reasoning in Cognitive Nodes

Cognitive networks are proposed in the framework of evolution of network architectures as novel paradigms to provide autonomous in-network reasoning to support end-to-end goals. While several articles are available that propose different approaches, the problem of reasoning still represents a challenging issue. This section aims to propose a mathematical model, based on Fuzzy Cognitive Maps, to support and provide a quantitative tool for implementing reasoning in the nodes of a cognitive network. In particular, the idea is to provide a methodology to enable network nodes to represent the complex interactions that happen within their protocol stacks and across the network. The potential utility of the proposed scheme is then validated on a sample scenario, outlining good results and relevant potential for future developments¹.

3.1.1 Introduction

Following the current evolution of communication networks, the Future Internet is expected to become service-oriented and to provide support to a wide variety of applications, ranging from browsing and data transfer to more complex and interactive applications that go far beyond the triple-play service vision—e.g. data, voice and video—and including augmented and virtual reality interactions, HDTV broadcasting and video on demand, online gaming experience, etc.

However, different services are characterized by different quality constraints on the communication infrastructure, yet are bound to the rigid layered architecture designed in the '70s. In such framework, performance assurance and optimization represent a hard task, due to the fact that each protocol is designed to work with different goals and without being aware of the behavior of the other protocols either in the network as well as within the same node.

Indeed, interactions among protocols at different layers of the protocol stack can potentially jeopardize the overall performance, and are difficult to quantify, foresee and control. It is clear that single layer optimization does not represent a suitable approach to provide controlled quality (being performance dependent on the interaction of all layers). Moreover, cross-layering (as it is usually found in the literature) is not going to be the ultimate solution, since it is often “static” and focuses on specific objectives, not considering the dynamics of the networking infrastructure but also the variable and evolving needs of the user applications and services.

In addition to “internal” interactions described above, performance is heavily influenced by the complex interactions among network elements (end-systems, routers, etc.), requiring proper coordination of devices along the data path to support Quality of Ser-

¹The work presented in this section has been partially supported by the Italian National Project: Wireless multiplatform active access networks for QoS-demanding multimedia Delivery (WORLD), under grant number 2007R989S.

Part of this work was partially done while the author was a visiting Ph.D. student at the State University of Campinas (SP, Brazil) in the framework of the EUBRANEX Erasmus Mundus External Cooperation Window (EM ECW) EU programme.

Part of this work was published in the proceedings of the 3rd IEEE Workshop on Enabling the Future Service-Oriented Internet – Towards Socially-Aware Networks, at the GLOBECOM '09 international conference, Honolulu, HI, USA, 2009 [46].

vice (QoS) constraints.

Clearly, the need for suitable architectures to provide end-to-end service-oriented performance control and optimization is emerging as a central point in the design of the Future Internet.

This point seems to be well addressed by the cognitive networking paradigm, first proposed by Thomas *et al.* [13]. Cognitive networks are composed of intelligent nodes, capable of reasoning about the environment they live in and acting in order to meet a global end-to-end goal, while continuously learning about the operating context and consequences of its actions. Cognition should enable to reduce the complexity of management of the network by means of its dynamic adaptation and self-configuration in order to support the time-varying requests of the users.

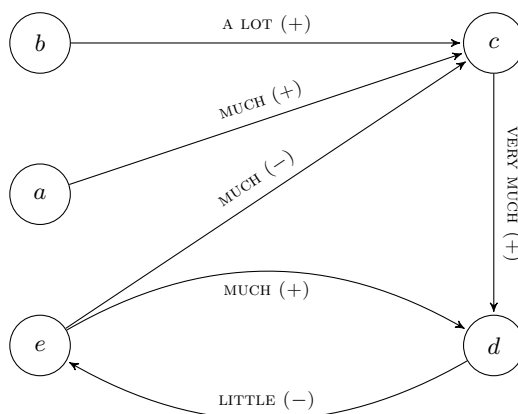
This “cognition loop”, graphically represented in Figure 1.3, is central to any cognitive architecture. According to this cycle, during the *sensing* stage a cognitive entity acquires knowledge about the environment and pre-processes it. Afterward, the reasoning, driven by end-to-end principles, takes place during the *planning* stage while the actual decision is taken in the *decision* stage. Finally, in the *acting* stage, actions are executed and their effect evaluated, by sensing the environment again.

Cognitive networking undoubtedly needs cross-layering to operate. In this framework, cross-layering represents the means to provide optimization, while the “cognitive engine” represents the learning, adaptation and decision process which drives it to achieve end-to-end goals.

Several cognitive network architectures are available in the literature, dealing with different stages of the cognition loop.

Among the others, *planning*, i.e. reasoning, and *learning* are the parts of the cycle that have received by far the most attention. Relevant considerations are drawn in [28], where Langley *et al.*, discussing the research challenges related to cognition, argue that in order to support reasoning, cognitive architectures must be capable of “representing relationships among beliefs”. In the author’s opinion, such skill can be developed by means of tools like Bayesian networks, neural networks, and first order logic. A more practical perspective is provided by Thomas *et al.* [13], who list some other machine learning algorithms apt to be used for the reasoning process, such as genetic algorithms, learning automata, and expert systems. However, rather than giving some indications about which particular technique should be used, they offer two insightful guidelines: (i) they suggest that the choice of the right technique (or techniques) largely depends on the problem to be solved, and (ii) whatever the algorithm chosen, it needs to converge before environmental conditions change. In [18], the authors advance the idea that more than one reasoning technique could be selected for a particular situation and propose to equip cognitive networks with a whole set of tools, rather than just a single tool. However, no explicit mention is given on how to build such a set of tools. A remarkable approach is provided in [53], where the use of fuzzy logic is motivated by the idea of reproducing the way human beings reason.

As correctly summarized in [3], the techniques used in cognitive networks for reasoning purposes are limited in number and often not justified over other available alternatives. In addition, it appears that the cognitive networks proposed in the literature, when ad-



(a) Graphical representation

$$F = \begin{pmatrix} 0 & 0 & 0.6 & 0 & 0 \\ 0 & 0 & 0.9 & 0 & 0 \\ 0 & 0 & 0 & 0.8 & 0 \\ 0 & 0 & 0 & 0 & -0.2 \\ 0 & 0 & -0.6 & 0.6 & 0 \end{pmatrix}$$

(b) Mathematical representation (adjacency matrix)

Figure 3.1: Fuzzy Cognitive Map example

dressing the reasoning problem, do not explicitly consider cross-layer relations.

In this framework, the aim of this section is to introduce the usage of graph-like structures, commonly known as Fuzzy Cognitive Maps (FCMs), to explicitly represent cross-layer and network-wide interactions, and use such information as a base for the reasoning process.

The remainder of this section is organized as follows. We begin Section 3.1.2 by describing the basics of Fuzzy Cognitive Maps. In Section 3.1.3 we analyze how this tool can be employed to foster reasoning in a cognitive network node and we propose our approach in Section 3.1.4.

3.1.2 Fuzzy Cognitive Maps

FCMs are mathematical structures conceived in 1986 by Kosko [19] as a means for modeling (possibly dynamical) systems through the causal relationships that characterize them. A pictorial representation of a hypothetical FCM is shown in Figure 3.1a. Figure 3.1b shows the adjacency matrix of the graph, which is an alternative widely used representation for FCMs².

Graphically, an FCM is rendered as a directed graph, in which a node represents a generic concept (e.g. an event, a process, a variable or other generic entities) and edges between any two concepts mean that there is a causal relation between them, the cause being the node from which the arrow starts.

²Throughout the remainder of the text, the term FCM is used to refer to both the formalism and the graph/matrix.

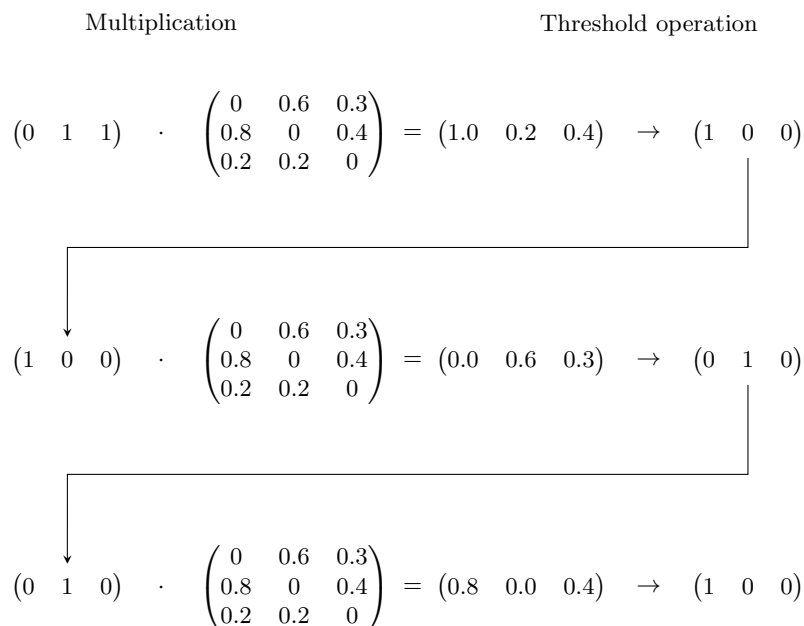


Figure 3.2: FCM inference process example. As can be seen, the reasoning process results into the cycle limit $(1, 0, 0) \rightarrow (0, 1, 0)$ in three iterations. Concepts mapped to the $\{0, 1\}$ domain and threshold set to 0.5.

In the simplest case, the domains of nodes and the weights of edges are discrete. In more complex FCMs, however, nodes can be mapped on larger sets, depending on the detail the designer wants to achieve. Indeed, nodes and edges can in principle be fuzzy, in that they may take any value in the continuous sets $[0, 1]$ and $[-1, 1]$, respectively. Though the use of larger sets generally results in a greater flexibility of a model, it is often the case that FCMs concepts are mapped on the discrete set $\{0, 1\}$ and edge labels mapped on $\{-1, 0, 1\}$. Such maps are better known as simple FCMs and are particularly suited to obtain a preliminary model of a problem.

A zero-valued concept denotes that the concept is *off*, *inactive*, in a *low-state* or it can even mean that it is not considered at all. Conversely, a concept set to one means it is regarded as *high* or *active*. Edge labels measure the degree of causality; values of either $+1$ or -1 denote a strong causal relationship, positive in the former case and negative in the latter. A zero-valued label means that the two concepts are not causally related to each other. Notably, no concept can cause itself, hence edges leaving and entering the same node cannot exist. Equivalently, the trace of the adjacency matrix of any FCM yields zero.

The state of a system having n distinct concepts is a vector of dimensions 1-by- n . In the inference process, this vector is repeatedly multiplied by the FCM matrix and the result thresholded each time, until it converges either to a fixed point or to a limit cycle. The complete procedure is shown using a toy example in Figure 3.2.

FCMs offer some advantages over other potential reasoning techniques (neural networks, Bayesian and Markov networks).

Unlike both Bayesian and Markov networks, the inference procedure used in FCMs,

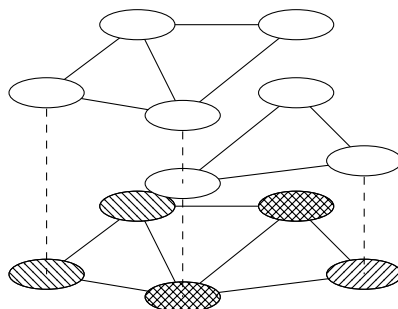


Figure 3.3: Merging multiple Fuzzy Cognitive Maps. Arrows intentionally omitted for clarity.

since it involves exclusively vector-by-matrix multiplications and thresholding operations, has a low computational footprint.

Moreover, FCMs are more powerful when there are causality loops in the problem; for such problems Bayesian network cannot be applied. In Markov networks, “loopy belief propagation” algorithms can be applied, but convergence is not guaranteed. Additionally, the FCM inference process can be applied to both loop-free and non-loop-free problems. Most notably, inference is guaranteed to converge to either a fixed point or a limit cycle, provided that concepts take their values in any finite discrete set [39].

Neural networks have also been employed to model dynamic systems. However, once a neural network reaches a solution, the conformation of its links does not necessarily reflect the actual relationships tying together the system variables [31, 39]. On the contrary, the edges in an FCM faithfully represent those relations and are, therefore, more appropriate to analyze both direct and indirect cross-layer interactions.

Another considerable advantage is the possibility of exchanging and merging together multiple FCMs, resembling operations people do when they exchange their opinions. This aspect has its roots in the primary purpose for which FCMs were created, i.e. to allow experts to represent their causal knowledge about some situation. Different experts may have different opinions about the same matter, and may encode differently their beliefs, hence drawing conflicting FCMs. Merging helps to smooth (possibly divergent) beliefs and biases, thereby reducing the possibility of biased reasoning. Moreover, weights can be employed to give more or less credit to each expert. Interestingly, as shown in Figure 3.3, the augmented FCM can be composed of potentially non-overlapping FCMs, thus, enabling the exchange of knowledge in case the domain of knowledge of cognitive entities is different, which undoubtedly is an advantageous feature when dealing with uncertain scenarios, such as, for instance, communication networks.

However, there exist some disadvantages as well. One major drawback concerns the automatic synthesis of FCMs: FCMs were not conceived for being constructed starting from observational data. They were initially devised in the social science field as a tool to be used by experts, in order for them to formally express their beliefs about a given matter. For this reason, autosynthesis of FCMs is difficult, mostly because, for non-humans, cause-effect relationships between variables are generally more complex to detect than simple correlations [31].

Another limitation of FCMs is the restricted ability to deal with the complementary

process of inference, i.e. abductive reasoning. Abduction is the process of stating which causes are responsible for a given effect: in case of FCMs, as well as in Bayesian and Markov Networks, it requires the solution of an NP-hard problem [40].

Despite such problems, the benefits deriving from the use of FCMs in cognitive networking appear to outweigh their drawbacks. This is also supported by the fact that FCMs have been used in different domains for many different purposes, including for instance the creation of medical decision support systems [54] and the simulation of virtual worlds [55]. The interested reader can find a thorough survey in [56].

3.1.3 Fuzzy Cognitive Maps for Cognitive Nodes

FCMs represent a suitable model for dynamic systems and can also represent (or embed) cross-layer and network-wide interactions, thereby enabling a truly holistic approach to the problem of cognition in intelligent networks.

The following paragraphs discuss the issues related to the usage of FCMs to support reasoning in cognitive nodes.

3.1.3-A Reasoning

The purpose of a cognitive network is to enable end-to-end performance optimization. As performance depends on several interacting factors and parameters associated to different layers of the protocol stack, the focus of this section is on cross-layer interactions. As a consequence, it appears logic to map concepts to communication protocol internals. However, the same considerations can be applied also when analyzing network-wide interactions, as we will see in Section 3.3.

To this end, we propose to distinguish three classes of concepts. A first class comprises the concepts related to quality-of-service metrics the system pursues; examples of concepts in this class may be “application-level throughput” and “end-to-end delay”. A second class of concepts includes all the concepts related to the environment in which the cognitive entity is; concepts such as “bad channel conditions” and “network congestion” belong to this category. A third class includes the set of actions (or a subset) that each protocol can perform; concepts like “use of Request to Send/Clear to Send (RTS/CTS)” and “packet fragmentation” fit here.

From a more rigorous perspective, we can define the system state vector \mathbf{s} as composed of the sub-vectors $(\mathbf{q}, \mathbf{e}, \mathbf{a})$ that represent the three classes discussed before, that is:

- \mathbf{q} , of quality-related concepts,
- \mathbf{e} , of environment-related concepts, and
- \mathbf{a} , of action-related concepts.

This representation allows for the reformulation of the tenets introduced in [13]: the FCM needs to converge to a solution state $\mathbf{s}^* = (\mathbf{q}, \mathbf{e}, \mathbf{a}^*)$ by finding a vector \mathbf{a}^* such that the constraints expressed by \mathbf{q} are satisfied before environmental conditions \mathbf{e} change.

This means that the FCM is used to find which causes generate a specific effect, or in more general words, to perform abductive reasoning [28], which, as previously mentioned,

is shown to be a NP-hard problem [40]. However, this issue can be mitigated, by noticing that the search space is reduced: only the elements of \mathbf{a}^* are to be found, since both \mathbf{q} and \mathbf{e} are given and the cognitive entity cannot change any element of them. Abductive reasoning is especially problematic when the system state is composed of a high number of variables. Noteworthy, as the search space is determined by the dimension of \mathbf{a} , and provided its dimension is small enough, a simple brute force approach can be feasible. This aspect can be addressed in several ways. For example, though at the outset, when the FCM has a few (if any) edges the actions to form \mathbf{a}^* can be chosen in no particular order, it could be helpful to devise a module to be used in parallel with the cognitive entity to store experience. Such a module, as the cognitive node experiments new situations, could keep track of the most likely combinations of \mathbf{a}^* that yield a desired result, so that they can be evaluated and chosen according to their probability, potentially allowing the skipping of the exhaustive search phase. It should also be noted that multiple sets of actions can lead to the same desired situation; thus, if there is no interest in a particular combination, search time can potentially be further reduced. This issue is, however, out of the scope of this section and will be inspected in a more detailed form in Section 3.5.

As mentioned in the previous section, concepts can be mapped to different domains: from simple discrete binary sets to continuous sets.

The binary discrete set $\{0, 1\}$ is certainly the simplest domain, and it often fits many concepts. This is the case of all those concepts that inherently convey an “on-off” or “existent-nonexistent” meaning, such as, for instance, “the use of RTS handshake” or “congestion in the network”. Both examples represent dichotomies. Intuitively, RTS can either be employed or not, and either there is congestion or there is not. No other situation is acceptable. As a result, when RTS is not active or there is no congestion, the related concepts will be set to the null value—and to the unitary value, otherwise.

Other concepts are better expressed by another binary discrete set: $\{-1, 1\}$. Variables mapped to this set have the capability to completely invert the causal relationships that bind them to other concepts. As an example, let us consider the classic Transmission Control Protocol (TCP) congestion window mechanism (not to be confused with the “congestion” concept in the previous example). It would be far-fetched to map it on the $\{0, 1\}$ domain: it is not possible to identify situations where the congestion window is *on* and situations where it is *off*. Instead, it is possible to note that increasing the window size generally causes the throughput to increase but at the same time also increases the probability of congestion. When the first signs of congestion appear, the window is suddenly reduced, to prevent the congestion from getting worse. The concept of “TCP congestion window” can, thus, be reasonably well represented by the $\{-1, 1\}$ domain. In this case we will have positive causality with respect to the throughput when increasing, negative causality when decreasing. A visual example of this mapping is provided in Figure 3.4.

This line of reasoning can be further extended. Let us suppose we devise an enhanced version of the congestion window mechanism described in the previous paragraph. Let us assume that, besides increasing and decreasing, the window possesses another degree of freedom, i.e. it can maintain its value. The concept can, thus, become trivalent, that is, it can be mapped to the domain $\{-1, 0, 1\}$ [39].

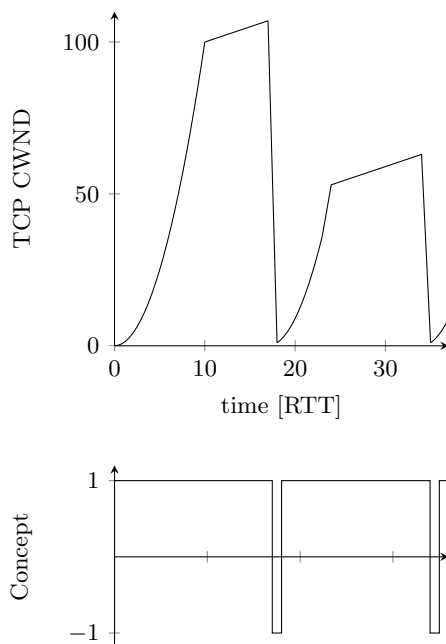


Figure 3.4: Possible evolution of the TCP congestion window and its transformation into binary concept.

Extensions are virtually infinite. Let us suppose that, for a certain problem, modeling already mentioned congestion window as simply increasing or decreasing is not precise enough. In such a case it is possible to employ domains with more than three levels, such as $\{-1, -0.5, 0, 0.5, 1\}$, where intermediate steps transmit the idea of “slightly increasing” or “slightly decreasing” and allow a finer modeling detail. Ultimately, discrete domains may be dismissed in favor of continuous domains, that could allow, in theory, a greater (ideally infinite) degree of precision.

However, to the best of the author’s knowledge, it has not been shown yet that the use of domains characterized by a higher number of levels leads to more detailed solutions. Nor has anything been said about the accuracy of FCMs with mixed concepts (e.g. binary and ternary), as they do not seem to have been studied in the literature thus far. Furthermore, for the sake of tractability, (i) discrete domains have been preferred to continuous domains, and (ii) among them, those with fewer levels are most likely the ones to provide the more efficient solutions.

The reason behind the first point is that, when employing discrete-valued concepts, the inference process leads to either fixed points or limit-cycle solutions. Continuous domains are discouraged, since inference can result into chaotic behavior.

The inference process is also the reason behind the second assertion. One should prefer low cardinality sets for mapping concepts because, given the number of concepts in an FCM, c , and the number of levels of the domains, l , inference is guaranteed to reach a solution within l^c steps [39]. Therefore, increasing the number of levels can potentially greatly dilate the convergence time for inference.

It should be noted that not all variables potentially included in the reasoning formalism can be *directly* mapped onto discrete domains. An example of such variable is the “end-to-

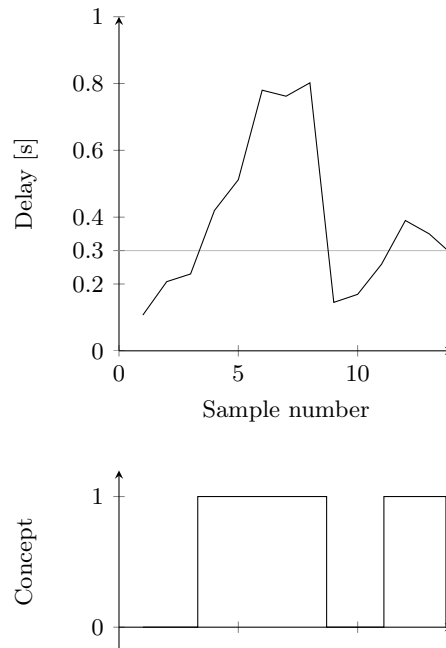


Figure 3.5: Transformation of delay measurements into a concept. Threshold can be chosen as the maximum tolerable delay.

end delay” that packet experience when traveling across a network. Its natural domain is continuous and only half-bounded: $[0, \infty)$. Pre-processing of any such variables is needed in order to map them on a discrete and bounded domain. The simplest pre-processing operation enabling this mapping is the comparison of the variable values against threshold. The designer can identify a threshold as the value below or above which a variable becomes meaningless. For instance, in the case of the “end-to-end delay” in a voice communication network, the designer can set the threshold to the maximum tolerable delay, as shown in Figure 3.5. Alternatively, a more general approach that works in case of variables (uniformly or non-uniformly) distributed over both continuous and discrete domains is to determine thresholds by inspecting basic statistics, such as mean or median values. However, finding the optimal threshold value is not straightforward, as it strongly depends on the problem and greatly impacts the performance that can be achieved.

Another question is whether it is possible to comprise mutually exclusive concepts. This is the case, for instance, of a node equipped with two different implementations of TCP, which clearly cannot be used at the same time. In such a case we could resort to the formalism used in Competitive FCMs [54], in which such concepts are linked to each other by means of totally negative edges: this way when either one of the concepts is triggered, the other is deactivated.

3.1.3-B Learning

After identifying the concepts and their domains, the structure of the problem must be outlined. Indeed cognitive nodes must perform continuous learning—making it possible to spot which concepts are causally related to one another and which are not (and, ideally,

also detect redundant concepts).

If the designer has full knowledge of the problem, he can connect the concepts with one another, specifying the degree of causality as edge labels. This, however, is rarely the case. Most often, information is missing and just a few a-priori beliefs are available. In this case, the designer needs to embed the information available in the FCM and let the graph organize itself by means of learning so as to reproduce the causal beliefs that reflect the environment.

Learning algorithms used in FCMs mimic the way human beings learn. Humans can infer that there is some sort of causality between two events when they perceive simultaneous changes [42]. Similarly, the basic idea is to register variations of the concepts and update the label of the edges connecting them.

One of the most popular learning rules to update an FCM is called Differential Hebbian Learning (DHL). In the following paragraphs we will describe the principles behind it.

To explain how it works, let us refer, without loss of generality, to the case of a simple FCM. If two concepts change their value from 0 to 1, it can be inferred that the positive variation in one concept has caused a positive variation in the other. In the case the two concepts change their value from 1 to 0, it means that a negative variation in the cause leads to a negative variation in the effect. In both cases, the sign of the variation is the same for causes and effects, and it can be stated that there is a positive causality between the two concepts. Differently, when a positive variation in one concept leads to a negative variation in the other, and vice versa, it is possible to state that there is a negative causality between the two concepts.

Formally, if we denote by C_i and C_j two generic concepts, by \dot{C}_i and \dot{C}_j their variation over time (time derivatives), and by f_{ij}^t the *variation* of the label of the edge connecting concept C_i with concept C_j at time instant t , the differential Hebbian law states that:

$$\dot{f}_{ij}^t = -f_{ij}^{t-1} + \dot{C}_i^t \dot{C}_j^t \quad (3.1)$$

As can be seen, derivatives encode changes and the product correlates these changes. In Equation (3.1) variation of concept C_i occurs before variation of concept C_j . The other way around, if C_j were the first concept changing, the edge label to be updated would be f_{ji} . The negated value of the edge label at the previous time instant is included in the right-hand side of the equation to prevent that a spurious simultaneous variation impacts indefinitely.

Accordingly, the edge f_{ij} at step t is computed as the value it had at time step $t - 1$ plus the variation:

$$f_{ij}^t = f_{ij}^{t-1} + \dot{f}_{ij}^t = \dot{C}_i^t \dot{C}_j^t \quad (3.2)$$

As can be seen, the inclusion of $-f_{ij}^{t-1}$ in Equation (3.1) allows the edge label to be reset to zero when either one of the concepts does not vary, thereby indicating that the previous causal relationship no longer holds.

Depending on the application, such an update may be too abrupt and a less responsive behavior can be preferred. In such case, the responsiveness of the algorithm can be modified by introducing a parameter η in (3.2), known either as “learning rate” or as “decreasing learning coefficient” [55] to help smoothing edge variations. Its value is,

Table 3.1: Characteristics of the measuring process, depending on node location and layers involved

Location	Layers	Direct/Indirect	Software/Hardware
Edge	2-7	Direct (usually) and indirect	Software
	1		Software/Hardware
Core	3	Indirect	Software

generally, defined in the set $(0, 1]$: values close to zero result in a slowly changing FCM, while values close to unity produce a highly responsive FCM.

Accounting for the learning rate η , Equation 3.2 can be rewritten as:

$$f_{ij}^t = f_{ij}^{t-1} + \eta \left(-f_{ij}^{t-1} + \dot{C}_i^t \dot{C}_j^t \right) \quad (3.3)$$

It should be noted that, with the introduction of such parameter, it is possible that edge labels take values in sets other than the basic set $\{-1, 0, 1\}$. This, however, does not affect the duration of reasoning.

3.1.3-C Sensing and Acting

Sensing and acting are two fundamental activities, complementary to each other, that should be independent from the reasoning formalism employed.

Sensing can be decomposed in two main functions: measurement and communication of the values extracted. The specific operations involved in measuring obviously depend on the variable measured. However, it is possible to delineate a few general characteristics, based on some properties of the node involved in the process. In fact, depending on the location of the node and the layers at which measuring is performed, measurements can be taken directly or indirectly, and via software or hardware. Table 3.1 summarizes this classification clearly.

For edge nodes, measuring at any of the layers between 2 and 7 included (i.e. at any layer from data link to application) can be generally done via software, provided that it is possible to access the internals of the operating system running on the node, and (for layer 2) of the driver of the networking interfaces. Physical layer (layer 1) functionalities, on the other hand, are generally implemented in hardware and, though in principle hardware probes could be used, in practice their use is not common at all. Logically, this is not the case when layer 1 functionalities are implemented in software, as in software-defined radios, for which ad-hoc programs can be written.

For nodes in the core network—routers essentially—the measuring process is generally more problematic. Independently from their use (i.e. whether they are conceived for operating in LAN or WAN environments), they hardly allow arbitrary code to be run. Needless to say, measuring in this case is limited, and must be carried out indirectly, e.g. by probing the node using specially crafted packets.

Eventually, the sensed values will have to be communicated to the reasoning entity. For this task we can distinguish between local or remote, in-band or out-of-band, and pull-type or push-type communications.

For local communication, i.e. the exchange of data within a single network node, overhead costs are virtually nonexistent. On the other hand, if data is transmitted over the

network, overhead is present and careful decisions must be taken in terms of frequency and size of the data messages sent.

Then, a signaling method must be chosen. In-band communication exploits the same logical channel used for the transmission of data, whilst out-of-band communication is done by setting up a different channel, to be used exclusively for the transmission of such kind of information.

Particularly important is deciding on the nature of updates, whether they should be requested by the reasoning entity (pull scheme) or autonomously sent by the node performing the sensing (push scheme). The pull scheme involves at least a couple of messages to be sent: a query must be transmitted by the entity requesting the data, to which a response containing the data will follow. Opposedly, according to the push scheme, updates are sent by a sensor, regardless if they have been requested. Advantages and drawbacks are present in either scheme. The former may reduce the overall number of messages exchanged, as data is exchanged only when the cognitive entity needs it. The latter, on the other hand, may reduce the overall implementation complexity.

With regard to the acting stage, what happens is that the cognitive entity communicates the actions to be taken to the corresponding protocols in the stack. Basically, either a direct communication path or an indirect communication path has to be implemented.

Direct communication implies implementing new interfaces in a protocol. The great advantage of such an approach is that it does not limit the designer's choices of implementation. Not to mention that the designer can define a specific communication scheme, as well as implement new actions. The drawback is that this approach requires rewriting parts of the protocol, which may not always be a feasible option. For instance, in case of protocols that belong to the lowest layers of the stack, we may have no access to their code, due to either limitations of the operating system or hardcoded code in the network device.

Conversely, indirect communication tricks a protocol into the belief of experiencing a particular situation, to which it will react with a precise action. Clearly, this approach preserves the protocol nature, but at the same time it does not allow the implementation of new actions.

In summary, however, no general rules can be provided for the implementation of sensing and acting. The trade-offs implied by each design choice depend on the application and the scenario in which the cognitive network will be deployed.

3.1.4 The Proposed Approach

Summarizing the discussion above, the proposed approach to implement FCMs in cognitive nodes can be explained as a three-step process:

- 1) To identify which specific variables of the various communication protocols could be of interest for the intended application.

In particular, concepts should be categorized according to what they represent: everything that can be tuned should belong to the 'action' class, everything that cannot be controlled directly and conveys a QoS-related meaning should belong to the 'QoS'

class, and everything else (basically variables that cannot be controlled and are not QoS-related) should belong to the ‘environment’ class.

As an example, let us suppose we want to model the behavior of TCP over the wireless medium. Some concepts that can be used are: the TCP congestion window, the packet error probability, and the throughput. If we are allowed to drive the congestion window, then such a concept belongs to the ‘action’ class. No direct control can be exercised on the error probability, which should therefore be classified as an ‘environment’ concept. Finally, the throughput represents a measure of the performance and should belong to the ‘QoS’ class. If we label the concepts with the letters w , e , and t , in order, the resulting preliminary FCM will be:

$$F' = \begin{pmatrix} 0 & f_{we} & f_{wt} \\ f_{ew} & 0 & f_{et} \\ f_{tw} & f_{te} & 0 \end{pmatrix} \quad (3.4)$$

It should be noticed that the elements on the diagonal are set to zero by default, as concepts cannot cause themselves. Instead, all the other elements can be non-zero, at least in principle.

- 2) To define the domain of each variable, avoiding continuous sets and keeping discrete sets as small as possible (or else being prepared to increased computational complexity).

In finding the right domain for boundless concepts, it could be of help to think of a threshold, so that greater (or lower) values entail the same causality as the threshold value.

With respect to the example we have introduced in 1), the domain of the TCP congestion window could be the discrete set $\{-1, 0, 1\}$, useful to represent situations when the window increases, decreases or remains stable. The error probability could instead be mapped to the discrete set $\{0, 1\}$, indicating absence or presence of errors; however, a proper threshold has to be chosen (for instance, according to the transmission modulation used), so to distinguish values of the error rate that affect the system behavior from values that do not affect it. Finally, the same domain could be apt for modeling the throughput: the lowest value can be used to represent non-satisfactory situations, the highest to represent favorable situations.

- 3) To design and implement the algorithm for building and updating the FCM, embedding into the matrix any available a-priori knowledge.

As it will be shown in Section 3.2, the choice and configuration of the algorithm is critical for achieving satisfactory results.

For instance, the fact that the throughput does not cause changes in neither one of the other concepts is translated by setting to zero the last row of the FCM in 3.4. The possible final version of the FCM can then be written as follows:

$$F'' = \begin{pmatrix} 0 & f_{we} & f_{wt} \\ f_{ew} & 0 & f_{et} \\ 0 & 0 & 0 \end{pmatrix} \quad (3.5)$$

It should be noted that the first two points are more similar to a pre-processing stage, rather than the reasoning stage itself. However, the two stages cannot be separated from one another and have to be accomplished in a jointly fashion.

3.1.5 Conclusion

In this section a novel tool to support reasoning in cognitive nodes has been illustrated. It uses the Fuzzy Cognitive Map framework, thanks to which it is possible to model dynamical systems by exploiting cause-effect relationships that characterize their internals.

In the context of cognitive networking, such tool is able to represent information about cause-effect relationships among operating parameters at different layers—thus providing a way to embed information about cross-layer interactions within the cognitive cycle of a network node.

The steps to be taken in order to implement this reasoning formalism in a network node have been described. Specifically, we have explained how to identify and include concepts in the formalism, how reasoning is performed, and how the system can be kept up to date by means of continuous learning.

3.2 The Cognitive Service-Oriented Infrastructure: An Application Example

Service-Oriented Architectures (SOAs) and Service-Oriented Infrastructures (SOIs) were proposed to support the evolution and composition of services on heterogeneous networking infrastructures. However, neither SOA nor SOI address the issues related to the performance of the underlying network infrastructure. This section shows that such issues can be mitigated by supporting the SOA paradigm using cognitive networks, which are networks that adopt continuous adaptation and intelligent communications, so that transparency to underlying technology, protocol independence and quality of service can be promoted³.

3.2.1 Introduction

SOA is an architectural paradigm that makes Information Technology (IT) environments more flexible by allowing the creation of loosely coupled services which are location transparent and protocol independent. A SOA should be supported by a proper Service-Oriented Infrastructure, to provide service consumers with the greatest possible transparency in relation to the underlying technologies while providing the best possible quality of service.

Based on such paradigm, a SOA should work independently of the specific underlying networking infrastructure, regardless of it being wired or wireless. Remarkably, however, as a consequence of a growing interest in the availability of services in mobile and nomadic scenarios, wireless-based SOAs have emerged in the recent past, leading to a significant effort to overcome the challenges in designing suitable solutions to SOAs over wireless network.

Sánchez-Nielsen *et al.* identify in service provision a significant challenge and attempt to solve the problems behind it by proposing to tailor SOAs to deal with the limitations of the wireless scenario, characterized by finite bandwidth and limited terminal processing capabilities [57].

Invoking services in a wireless context is thought to be one of the most challenging tasks according to Sen *et al.* [58], since a mobile terminal can experience frequent handoffs and connection can be easily lost. To ameliorate such problem, they propose to separate issues related to the medium from those related to services by means of a method they call “context sensitive binding”: limitations due to the use of a wireless channel can be masked so that technologies already developed for wired networks can be employed. However, connectivity can also be disrupted when there is no handoff, since the physical characteristics of a wireless channel can vary widely over time.

Service invocation and discovery introduce novel challenges to SOA over wireless networks, especially in ad-hoc networks [59, 60], in which service providers can move around, thus, thwarting service finding procedures.

³The work presented in this section has been partially supported by the Italian National Project: Wireless multiplatform mimo active access networks for QoS-demanding multimedia Delivery (WORLD), under grant number 2007R989S.

Part of this work was published in issue no. 4/1 of the Journal of Internet Engineering (special issue on service-oriented architectures), 2010 [47].

Icons used in the illustrations are copyright of Jakub Steiner. The versions for Dia are copyright of Thiago Ribeiro.

Actually, the wireless environment undermines the capability of supporting stable quality of service, which is a major objective of SOA. More importantly, this happens at different scales, or, in network terms, at different layers of the communication stack of a node, possibly involving fluctuation of channel characteristics at physical layer, hand-offs at the Medium Access Control (MAC) layer, and routing functions at the network layer. However, the existing proposals [57, 58, 59, 60] focus on specific aspects, leading to single-layer and scenario-specific optimization approaches.

Nonetheless, performance depends on all layers of the protocol stack and, as a consequence, a holistic approach is needed to identify proper solutions to such issues. Cognitive networking can provide such holistic approach.

In general, it can be noticed that it is possible to identify two systems: the Service-Oriented Architecture and the Transport Network (or Communication Infrastructure). They are loosely coupled and promoting a higher degree of collaboration between these two can lead to the concept of Cognitive Service-Oriented Infrastructure. Indeed, SOA and the network infrastructure have complementary goals and functionalities: SOA aims at platform-independent service delivery, while the network infrastructure aims at providing the best possible performance to the data flows. The interesting aspect is that one has the capability that the other misses: a SOA is not able to request proper quality of service since it does not have control of the underlying technologies, but it perfectly knows the requirements of the service; conversely, the network infrastructure is generally not aware of Quality of Service (QoS) requirements of the applications and services.

In line with that, this section proposes a novel architecture that facilitates interaction between SOA and the network infrastructure. This is achieved by decoupling SOA-related aspects from network-related aspects, as in [58], in addition to promoting cross-signaling between SOA and the network so that their mutual goals can be reached.

The proposed approach capitalizes on the fact that the granularity of events that trigger network adaptation to variation of traffic load and service requirements are related to protocols operating at different layers of the protocol stack. Moreover, intelligence should be embedded in the network stack to support context-aware decision-making processes [13]. In this context, cognitive networks, being autonomous and able to sense their operational environments as well as to learn from past experience so that decisions towards specific goals can be taken and continuously refined, nicely fill the gap between the needs of SOA and network infrastructure.

This section aims to describe a Cognitive Service-Oriented Infrastructure (CoSOI) able to properly support the requirements of the Service-Oriented Architectures. Collaboration and signaling between SOAs and cognitive networks are described, as well as how to identify a reasoning engine to efficiently deliver services over wireless networks. The wireless scenario is chosen not only because it represents a significant technical challenge, as described in the previous paragraphs, but also because it is adopted by a high percentage of today's network and service users.

The rest of the section is organized as follows. Section 3.2.2 shows how cognitive networks can be used to foster the deployment of Service-Oriented Architectures. Section 3.2.3 illustrates how the mathematical tool introduced in Section 3.1.2 to identify cognitive interactions in a system can be employed to realize the proposed Cognitive

Service-Oriented Infrastructure (CoSOI). Section 3.2.4 presents the validation of the framework through two relevant case studies, while Section 3.2.5 provides concluding remarks.

3.2.2 A Cognitive Service-oriented Infrastructure

The cognitive networking paradigm was conceived to deal with the increasing complexity of network management and the unsatisfactory performance of current networks, which results from their lack of capacity of adaptation to highly dynamic environments [13].

In cognitive networks, nodes are intelligent and are meant to capture information about the surrounding context, to perform reasoning on this information, to execute some action based on conclusions drawn and, finally, to learn from achieved results. This cycle of actions, graphically represented in Figure 1.3, is commonly referred to as the “cognition loop” and it is the core upon which cognitive architectures are built. The cognitive paradigm focuses on continuous, adaptive optimization, not only within a single node, but also at network-wide perspective. Thus, its effectiveness heavily relies on classic cross-layering techniques.

Actually, the main feature of cognitive networks is the ability to provide better global performance than that achievable by networks based on legacy technology (i.e. TCP/IP based). Ultimately, the approach followed by cognitive networks should lead to stable performance levels, even in heterogeneous or wireless network scenarios. Clearly, such characteristic supports stable and sound quality of service provisioning, necessary to the deployment of SOAs.

By inspecting the general architecture of a SOA depicted on the top part of Figure 3.6, it can be noticed that the *transport* layer is the layer which all functions rely on. Moreover, the transport layer is concerned with the transfer of requests and responses between service provider and service consumer [4]. Protocols commonly employed in the SOA transport layer are the Hypertext Transfer Protocol (HTTP), the Simple Mail Transfer Protocol (SMTP) and the Java Message Service (JMS), all of which in the TCP/IP protocol stack are classified within the *application* layer and use either the Transmission Control Protocol (TCP) or the User Datagram Protocol (UDP). These protocols, jointly operating with those at the other (lower) layers, can significantly impact the overall network performance. However, each protocol operates following the layering principle, having its own goals and functionalities, regardless of the other protocols at different layers operating within the same node or across the network. Along this line, cognitive networking can be seen as an enabler for SOAs: the cognitive node/entity, by interacting with the Quality of Service plane, can guide the whole protocol stack so that (end-to-end) quality of service requirements can be supported (Figure 3.6, bottom part).

The diagram shown in Figure 3.6 represents the proposed Cognitive Service-Oriented Infrastructure (CoSOI), in which interaction between the upper (SOA) and lower (transport infrastructure) sections is enabled and adaptivity to the transport infrastructure is provided.

To further underline the potential benefits deriving from the proposed CoSOI architecture, let us exemplify, with the aid of Figure 3.6, a typical situation, in which a potential

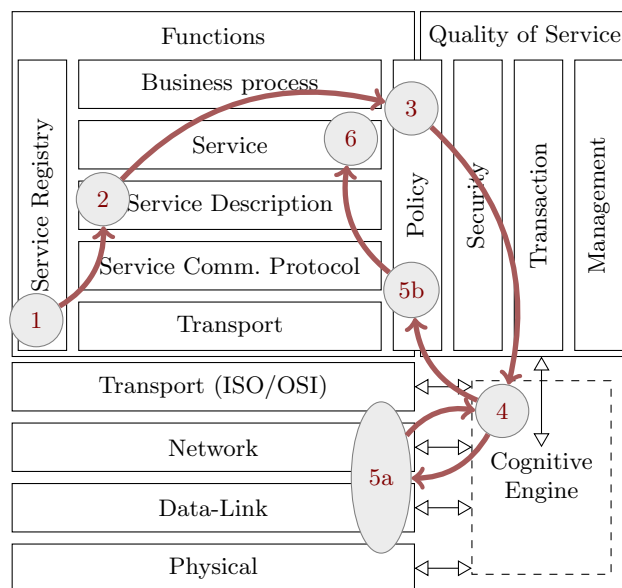


Figure 3.6: Cognitive networking and service-oriented architecture stacks (based on [4])

consumer looks for a particular service. The consumer will first access the service registry, in order to get more information about the service itself, such as how to invoke it and what should be expected (step 1). Successively, it will analyze the offered service through its description (2), promptly forwarded by the registry, and will eventually decide to use it. The decision is finalized by actually invoking the service. At this point, service user and service provider are bound by a bilateral agreement: if, on the one hand, the service consumer agrees to comply with the rules specified by the policy layer (3), on the other it expects a seamless delivery of the service itself (according to what the description states). To provide the user with the quality described in the schema, the policy layer instructs the cognitive engine about the quality requirements that must be met. The cognitive engine will keep monitoring the environment and reasoning about it with respect to the related quality goals (4), with the aim to hide all the factors possibly leading to bad quality of experience by continuously reconfiguring all the protocols in the stack (5a). In case the cognitive engine failed to mask channel imperfections, it could provide feedback to the policy layer (5b), allowing it to redefine the offered service (6). In summary, Figure 3.6 underlines that the cognitive engine should be able to reconfigure the whole transport infrastructure and interact with some SOA functionalities to gather service requirements and possibly give feedback related to the operating context. In this framework, it is easy to outline that a CoSOI can be effectively integrated with limited modifications in existing SOA environments, therefore providing a satisfactory level of interoperability.

3.2.3 Reasoning in Cognitive Service-Oriented Infrastructures: the Fuzzy Cognitive Maps

Thomas *et al.* [13] claim that the reasoning technique should be selected based on the dynamics of the context, in order to devise a proper architecture. In line with that, actions

```
<wsp:Policy
  xmlns:wsp="http://www.w3.org/ns/ws-policy"
  xmlns:wsrmp="http://docs.oasis-open.org/ws-rx/wsrmp/200702">

  <wsp:ExactlyOne wsp:Usage="Required">
    <rmp:RMAssertion>
      <rmp:InactivityTimeout
        Milliseconds="600000" />
      <rmp:BaseRetransmissionInterval
        Milliseconds="1000" />
      <rmp:ExponentialBackoff />
      <rmp:AcknowledgementInterval
        Milliseconds="200" />
    </rmp:RMAssertion>

    <rmp:RMAssertion>
      <rmp:InactivityTimeout
        Milliseconds="600000" />
      <rmp:BaseRetransmissionInterval
        Milliseconds="3000" />
      <rmp:ExponentialBackoff />
      <rmp:AcknowledgementInterval
        Milliseconds="400" />
    </rmp:RMAssertion>
  </wsp:ExactlyOne>
</wsp:Policy>
```

Figure 3.7: A WS-Policy example

should be taken before the context changes, so that the quality of service experienced by a service consumer remains the same. This represents a crucial feature in SOA.

A promising technique in this direction is represented by Fuzzy Cognitive Maps (FCMs) [46], which are graphical structures that can be used for reasoning on cross-layer interactions inside a network node that affect the network performance. As the basics of FCMs have been laid out in Section 3.1.2, we will examine how they can be employed to enhance the performance of SOIs.

To illustrate the use of the cognitive paradigm in the context of SOAs, let us first define what a Web Service Policy (WS-Policy) is. A WS-Policy, as described in the relative World Wide Web Consortium (W3C) recommendation [61], is the specification of a web service in terms of the characteristics and constraints that said service possesses.

Let us suppose a web service is characterized by the WS-Policy shown in Figure 3.7.

A service consumer, when invoking such hypothetical service, must select one of the two policy assertions established by the provider. Supposedly, as both the acknowledgment interval and the base retransmission interval are set to lower values, the first assertion requires more bandwidth than the second one, hopefully providing a better quality experience.

Assuming the consumer has enough processing capability, it will clearly try to invoke the service through the most demanding assertion. In case the underlying transport medium is a wired, non-congested network, chances are that the service is delivered flawlessly. However, if the network is wireless, there is a great chance that the service will suffer from frequent disconnections, causing interruptions and eventually making the final user QoS experience vary.

By continuous monitoring of network conditions and the ad-hoc planning of actions, a cognitive entity would aim at maximizing some end-to-end objective, and could ideally prevent such an adverse situation from happening, or at least mitigate its effects. Specifically, the cognitive network would attempt to find an action (or a set of actions) that produce a desirable result, or, in terms of FCMs, to find the causes leading to a precise outcome.

In order to populate the FCM, it is necessary to find on which concepts reasoning should be based, and a preliminary step is to distinguish three possible categories to classify such concepts, following the guidelines defined in Section 3.1.4.

The most logical class of concepts that can be identified is that related to protocol operations. Concepts in this category represent the state of the protocols running in a node and the edges represent the interactions among them. Examples of such concepts are “the use of Request to Send (RTS) handshake” or “high physical data rate”.

The second category stems from the observation that network efficiency is affected not only by protocol actions but also by environmental conditions. The peculiarity of concepts related to environmental conditions is that nodes have no direct control on them, yet can influence them indirectly. By way of illustration, let us suppose we identify the concept indicating the wireless channel status and let us name it “bad channel conditions”. Obviously, a node cannot magically turn it off. Rather, it can turn on the “fragmentation” concept, which will try to hinder the “bad channel conditions”. Similarly, a node has no power to directly turn off “congestion” in a network. Still, it can act on “TCP congestion window” that ultimately will affect the congestion network state.

Finally, in order to reason and find the actions needed for a desired outcome to happen, the FCM needs to be guided by global quality objectives, which will form the third and last category of concepts to be included in the reasoning phase. Reasonably, such objectives should be suggested by the Quality of Service plan of a SOA. The service provider will express the requirements to the cognitive engine, which, in turn, will use them to build appropriate concepts. Back to the example in Figure 3.7, the WS-Policy contains two assertions, differing by the bandwidth (more properly, throughput) requirement. The cognitive entity should be able to analyze such assertions and deduce that a potential concept could be “high throughput situation”. However, it should be noted that there is no need for the cognitive entity to deduce concepts automatically; they can also be defined manually, a priori, by human operators.

3.2.4 Validation

In this section, two study cases are presented to illustrate the use of the proposed methodology (see Sections 3.1.4 and 3.2.3). Such examples are employed to validate the capability of FCMs to learn the relationships among the parameters and performance metrics, leaving the issues related to the ‘action’ phase for Section 3.3.

The whole testing procedure consists of two steps (Figure 3.8):

1. First, for each study case, all the possible scenarios are simulated, i.e. all the combinations of the input variables are inspected. For each simulation, a quality metric is measured and recorded, along with the input variables for that simulation, in a *scenario database*.

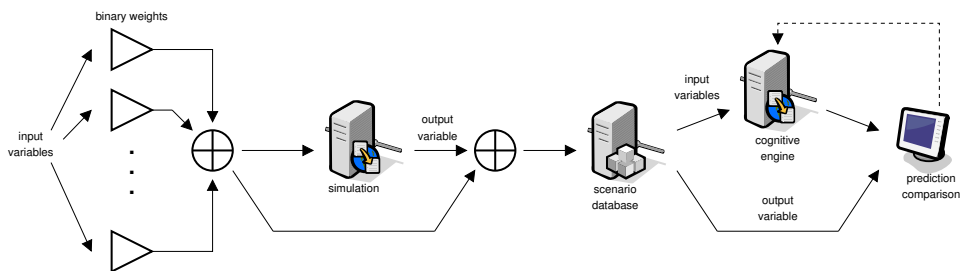


Figure 3.8: Validation steps. First, a database is populated with the simulations of all possible combinations of the input variables. Then, the prediction skills of the cognitive engine are tested against the simulation results.

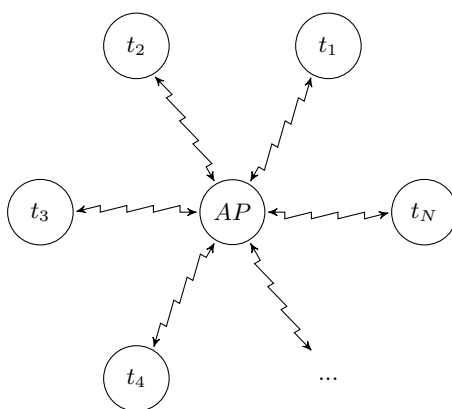


Figure 3.9: Network topology in the first scenario. *AP* denotes the access point while $\{t_1, t_2, \dots, t_N\}$ are the terminals.

2. Second, the proper validation takes place. We shuffle the data base, in order to (i) avoid that similar situations are placed in the same area, and (ii) randomize the experiment. After shuffling the database, the input variables of each scenario are fed into the cognitive engine, which uses them to predict the related quality metric. The prediction is evaluated against the previously stored value.

The cognitive engine learns, i.e. updates its beliefs by means of the Differential Hebbian Learning rule, discussed in Section 3.1.3, every time it is presented with a new scenario. This implies that concepts change synchronously and that prediction is triggered by changes in the operating scenario.

3.2.4-A Call Admission Control over an Infrastructure WiFi Cell

The first scenario we propose to validate the proposed approach concerns voice-over-Internet Protocol (IP) delivery over an infrastructure wireless network, whose topology is depicted in Figure 3.9. The test case is based on the data produced for the experiments in [62], where the performance of a Voice-over-IP (VoIP) Wi-Fi system is investigated. More specifically, experiments were conducted to measure how many VoIP calls the system supports given specific quality constraints.

In this scenario, the Access Point (*AP*) is the only node in the network equipped with

Table 3.2: Domains of the input variables in the first scenario

Property	Values	Abbreviation
Frame error rate	$\{10^{-9}, 10^{-8}, \dots, 10^{-1}\}$	e
Physical data rate	$\{1, 2, 5.5, 11\}$ Mb/s	d
Maximum number of retransmissions	$\{0, 1, \dots, 5\}$	r
Voice packet transmission interval	$\{10, 20, \dots, 90\}$ ms	i

cognitive capabilities and it is in charge of transferring VoIP calls generated by the N terminals in the network. The goal of CoSOI is to support the maximum number of VoIP flows from mobile users offering an appropriate level of quality of service.

Network conditions change over time and four input variables at different layers of the ISO/OSI protocol stack have been considered as representative of such changes, namely:

- the frame error probability,
- the data rate at the physical layer,
- the maximum number of retransmissions of a packet, and
- the voice packet transmission interval.

All such variables can potentially affect the number of calls supported by the system, subject to specific quality constraints. Table 3.2 summarizes the range of values of these variables.

In this framework, the cognitive entity is used to predict the number of calls supported by the system at a given time instant. This would enable the AP to perform call admission control of VoIP calls, given the specified quality constraint. Another, more advanced, use of the cognitive capabilities involving the ‘action’ part would be to allow the service provider to renegotiate the quality parameters of the call with the service consumer, by republishing the service according to the WS-Policy reported in Figure 3.11, as summarized in Figure 3.10: the call can still be placed, but, for instance, the quality of the voice can be significantly poorer. Furthermore, the cognitive entity could control the cognitive stack to try and mask channel imperfections, while also negotiating the quality parameters with the policy layer of the SOA stack. A scenario example involving the masking of channel imperfections is further discussed at the end of this section.

The first step to implement reasoning is to identify and classify concepts. Concepts related to protocol operations are those the cognitive node can modify, i.e. “the data rate at the physical layer”, “the maximum number of retransmissions of a packet”, and “the voice packet interval”. The only concept related to the scenario conditions is “the frame error rate”, as it depends on external parameters (i.e. behaviour of the wireless link, modulation, etc.), and the cognitive entity clearly has no direct control on it. Finally, “the number of calls supported”, being the only output, will represent the quality metric guiding the reasoning process: based on it, the cognitive entity can simply inform the other terminals, negotiate the quality parameters of the service, or even take actions to make the communication medium really transparent to the service.

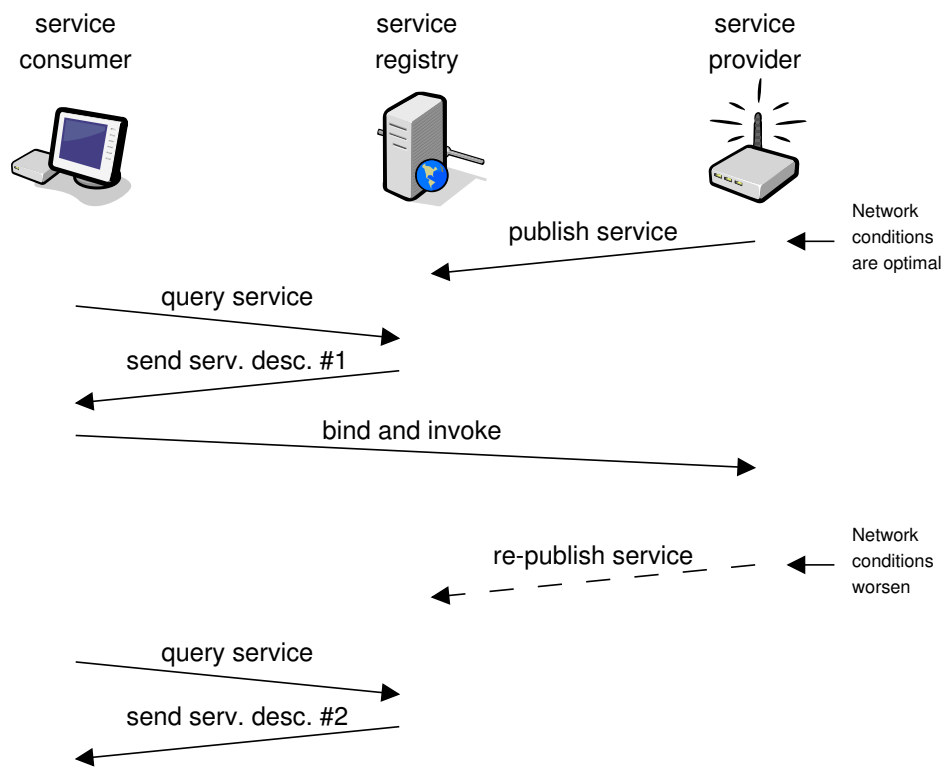


Figure 3.10: Temporal diagram of a possible service negotiation (time flows downward). Cognitive capabilities allow the service provider to publish a new service description, according to the current network conditions.

```

<wspes:Policy
  xmlns:wspes="..." xmlns:qos="..." xmlns:op="...">
  <wspes:ExactlyOne>
    <wspes:All>
      <wspes:Assertion name="AudioQuality"
        assertiontype="Capability">
        <wspes:Expression>
          <wspes:Parameter>PER</wspes:Parameter>
          <wspes:Value>0.05</wspes:Value>
          <wspes:ExactlyOne>
            <wspes:Operator>greater</wspes:Operator>
            <b>wspes:Operator>lower</b>wspes:Operator>
          </wspes:ExactlyOne>
        </wspes:Expression>
        <wspes:Expression>
          <wspes:Parameter>End-to-End Delay
          </wspes:Parameter>
          <wspes:Value>100</wspes:Value>
          <wspes:Unit>ms</wspes:Unit>
          <wspes:Operator>lower</wspes:Operator>
        </wspes:Expression>
      </wspes:Assertion>
    </wspes:All>
  </wspes:ExactlyOne>
</wspes:Policy>

```

Figure 3.11: WS-Policy (based on [5]) describing the service provided in the first scenario under optimal network conditions: service consumers can opt for the audio quality they want (Application-level PER greater or smaller than 5%). When conditions worsen, and the cognitive entity is not able to effectively thwart wireless channel imperfections, the WS-Policy will be updated, for instance by removing the bold part, implying that the service provider cannot guarantee calls characterized by a PER lower than 5%.

The second step is concerned with finding suitable domains for the concepts. Some of the concepts described are characterized by continuous domains. However, it would be beneficial to translate them into discrete sets. Indeed, all concepts in the data base can be mapped to a binary domain: for instance, either we have a high physical data rate or a low one, either we have a long voice packet interval or a short one, and so on. As mentioned in the paragraph entitled “Reasoning”, under Section 3.1.3, pre-processing is needed. In particular, by a simple comparison against a threshold, it is possible to map the concepts onto discrete (binary) sets. Threshold values can be found in several ways. They can be known beforehand or expressly specified by the designer, such as the voice packet interval⁴. Alternatively, these values can be either learned following standard specifications and recommendations, such as the maximum number of retransmissions or the physical data rate in IEEE 802.11 networks are specified on the relative standard [63], or they can be estimated through extensive testing, like the frame error rate or the number of calls supported. After the processing, we would end up having situations of high and low data rate, high and low number of retransmissions, and so on. The number of supported calls has been filtered as well, and mapped to the domain $\{0, 1\}$, equivalent to low- and high-throughput situations.

It is worth noting that some concepts are already characterized by discrete domains. However, there may be values in such domains that can be misleading, since they could

⁴It could be objected that a variable such as the packet interval is not actually specified by a network designer, rather by the VoIP application used. However, it is the designer that ultimately states *which* application should be used.

be unrepresentative neither of a high nor of a low state. For instance, this is the case of the physical data rate. In this scenario the data rate can be set to 1, 2, 5.5, or 11 Mb/s. While data rates of 1, 2, and 11 Mb/s can be safely classified as representative of low- or high-throughput situations, 5.5 Mb/s is an ambiguous value, as it falls on the threshold separating the two categories. In order to eliminate such misleading values and keep only the entries that are unambiguously high or low, a gray interval around the threshold can be defined: those values falling within the gray interval will be discarded.

For this test case all the combinations of the variables have been simulated. As a consequence, all the variables can be regarded as characterized by a uniform distribution over their domains. Therefore, the average value of the domain of each variable is selected as the threshold for the pre-processing operations.

The third and final step deals with the actual implementation of the FCM. The complete structure of the FCM can be outlined beforehand, by embedding the a-priori knowledge we have on the problem. First it can be noted that the frame error rate is the only true independent variable and all the other concepts are somehow affected by it. Indeed, it makes more sense that the frame error rate implies some change in the other concepts, rather than the other way round. This means that no edges in the FCM point to the packet error rate concept; instead, the packet error rate concept points to all the others. Next, we assume that the number of calls cannot cause any other concept, as it represents the “output” of the system: this is translated by setting to zero all the elements in the row corresponding to the number of calls. Ultimately, the action concepts (physical data rate, maximum number of retransmissions, and voice packet interval) are controllable parameters and thereby can be assumed as independent from one another. They, however, have some causal impact on the number of calls that can be forwarded. The resulting FCM is depicted in Figure 3.12, and mathematically defined by:

$$F_1 = \begin{pmatrix} 0 & 0 & 0 & 0 & f_{dn} \\ 0 & 0 & 0 & 0 & f_{in} \\ 0 & 0 & 0 & 0 & f_{rn} \\ f_{ed} & f_{ei} & f_{er} & 0 & f_{en} \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (3.6)$$

Subscript letters refers to the variables of the problem, as abbreviated in Table 3.2 (n , which does not appear in the aforementioned table as it is the only output variable, denotes the number of calls supported by the system).

As can be noted, due to the nature of the problem, no loop is present.

All relationships have been specified during the third design step, so no other relationship between any two elements can be expected to appear during the system evolution. As a consequence, the differential Hebbian law is used to update the already existing causality links. Initially, we let the learning algorithm update the FCM at each step. Results are presented in Figure 3.13, where the effect of only two causes (for clarity’s sake), namely the physical data rate and the number of retransmissions, has been studied. Then, we let the learning algorithm be applied only when the prediction is wrong. This, however, implies the presence of some external knowledge in the form of a feedback loop, capable of comparing what has been forecast to what actually happened, thereby enabling or disabling the Differential Hebbian Learning (DHL) algorithm. Results of the

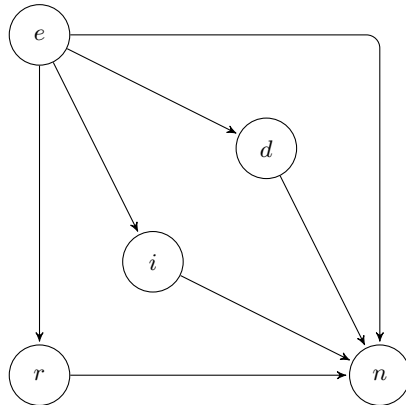


Figure 3.12: FCM employed in the first scenario. e , d , r , i , and n stand for packet error rate, physical data rate, maximum number of retransmissions, voice packet interval, and number of calls supported, respectively.

CoSOI with this enhancement are shown in Figure 3.14. As can be seen, better results can be achieved by modifying the FCM edges only when prediction fails.

In each figure the graph at the top shows the real number of calls supported by the system (continuous line) and the prediction given by the FCM (dashed line).

The graph in the middle depicts how reliable is the prediction, by performing the Negated Exclusive OR (XNOR) operation between the real data and the predicted data. A high logical value appears when the reasoning engine correctly predicts the number of calls admitted by the system. Conversely, a low logical value indicates that the prediction is wrong. More specifically, if the FCM predicts that a high number of calls (or low) can be supported, and the real number is actually high (or low) the graph shows a high logical value (correct prediction); in the opposite case, the graph shows a low logical value (wrong prediction).

The graph at the bottom shows how the edges of the FCM vary as a function of time: a comparison between Figure 3.13 and Figure 3.14 shows that continuous adaptation of the FCM leads to a worse performance and a more unstable FCM.

Remarkably, updating FCM edges only when the prediction is not correct allows the reasoning engine to score on average up to 10% more corrected guesses than in the case in which edges are updated regardless of the correctness of the prediction.

Figure 3.15 shows another run, where a different learning parameter has been employed. It is interesting to notice how causal relationships between the input variables and the output variable (the number of calls supported by the system) evolve as a function of time. Such relations tend to stabilize as time progresses and the cognitive entity succeeds in predicting the actual state of the system in many occasions: by having such a cognitive entity negotiate the quality of service parameters with the policy layer in the service provider, the imperfection and the capacity fluctuations of the wireless medium could be masked most of the time, and terminals could be given the chance of experiencing stable quality of service. For the scenario considered, an improvement of 11.34% ($\pm 1.53\%$, confidence interval computed with $p = 0.995$, DoF= 14) has been measured over the case in which a static decision is applied, e.g. deciding that the number of calls supported is

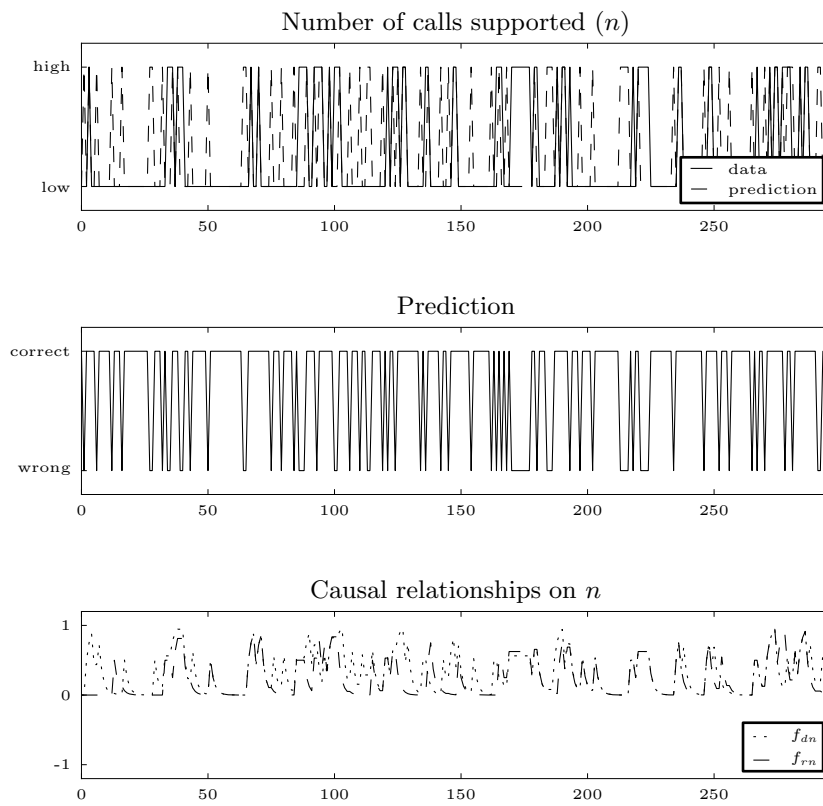


Figure 3.13: Performance achieved enabling the FCM update at each step. $\eta = 0.5$. Compare with Figure 3.14.

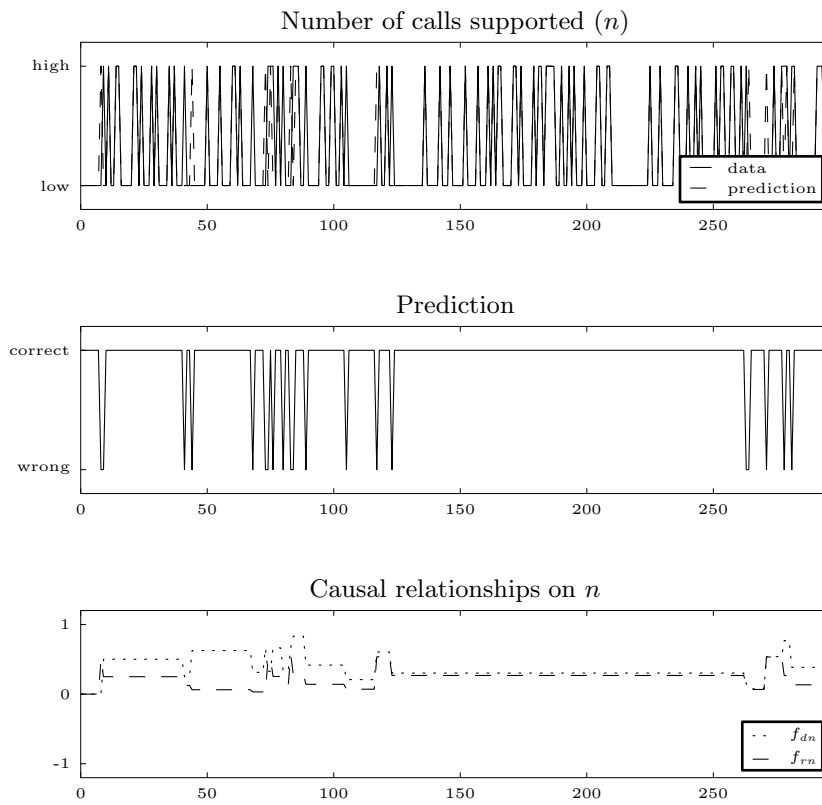


Figure 3.14: Performance achieved enabling the FCM update only in case of prediction errors. $\eta = 0.5$. The sub-graph in the middle shows that, by updating the FCM only when errors occur, predictions are more reliable. Compare with Figure 3.13.

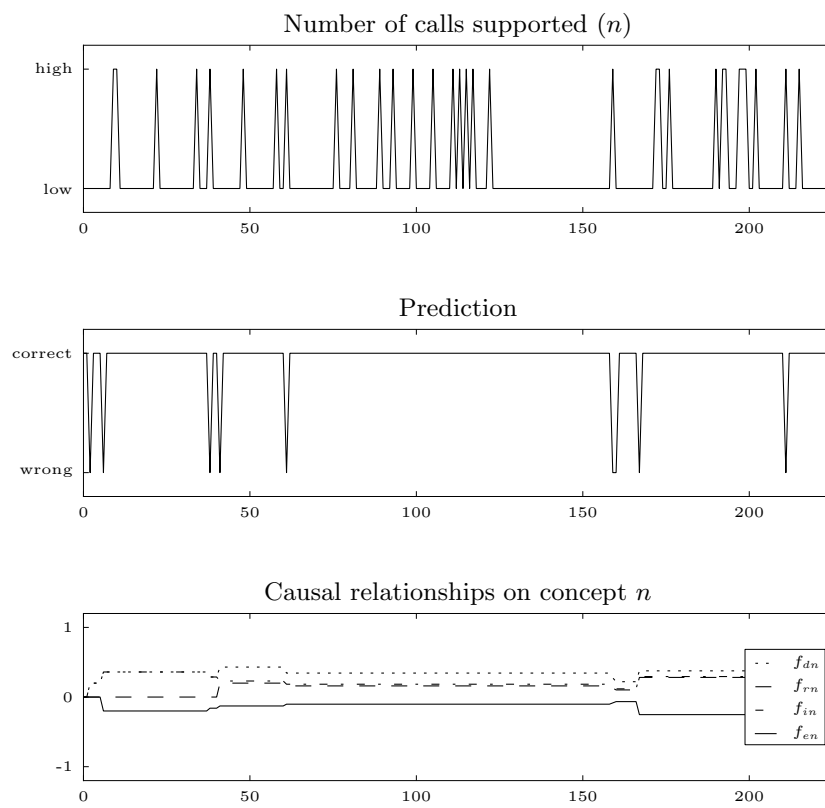
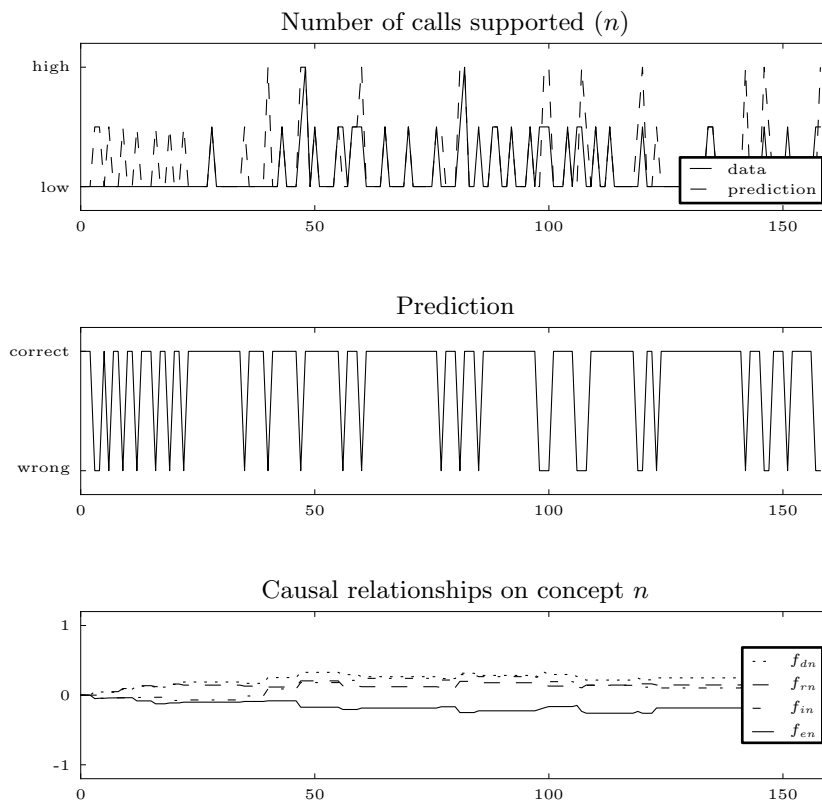


Figure 3.15: Prediction performance achieved by the reasoning entity in the first scenario. $\eta = 0.2$. When predictions are correct, the structure of the FCM does not change.

Figure 3.16: Throughput as a three-valued concept. $\eta = 0.1$.

always low (which happens about 85% of the times).

A slightly different situation is reproduced in Figure 3.16, in which the number of calls has been mapped as a three-valued concept, meaning the FCM could predict not only low- and high-throughput values, but medium values as well. Due to the greater complexity, predictions are less accurate, and heavily depend on the thresholds used for the mapping operation. Tests showed that the improvement is about 5.23% ($\pm 2.90\%$, confidence interval computed with $p = 0.995$, DoF= 14), remarkably lower if compared with the value achieved by considering the number of calls as a two-valued concept.

The impact of different values for the learning rate parameter η is investigated in Figure 3.17, which shows the evolution of the performance as a function of η . Specifically, it illustrates what is the performance obtained predicting the throughput value by means of the FCM (continuous line) and that obtained by applying a static decision as previously mentioned (in this case the decision made is equivalent to betting that the throughput availability is always high). For the specific scenario, a less dynamic FCM ($\eta \in [0.1, 0.4]$) is able to achieve better results than a more dynamic FCM.

Finally, to further underline what benefits the cognitive networking paradigm brings

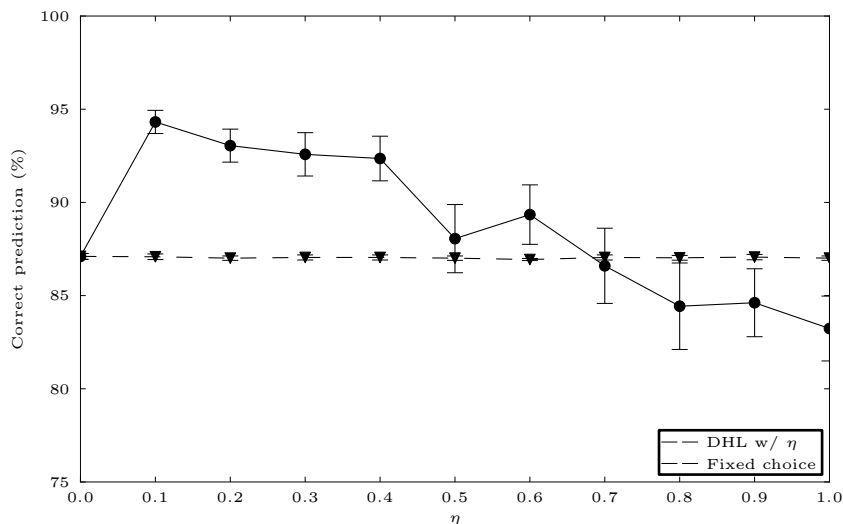


Figure 3.17: Impact of the learning parameter η on the prediction reliability. Confidence intervals computed with $p = 0.995$, DoF= 14.

to SOAs, we discuss in the following a situation showing the potential of the proposed CoSOI where also the ‘action’ part is enabled.

Let us suppose that, at a given time, the number of calls that can be forwarded is low. In that precise moment, additional clients request the service provider to place their call. Normally, in such a situation a non-intelligent service provider would forward all the calls, offering the relative owners a poor quality of service. A cognitive service provider, on the other hand, knowing that it would be impossible to forward a greater number of VoIP flows while also granting an acceptable quality, could decide not to accept their requests, at least as long as such adverse conditions remain. An even smarter service provider, however, would exploit its reasoning capabilities to try and find an action profile potentially leading to a better situation, where a higher number of calls can be supported.

Figure 3.18 depicts such a behavior. Despite the high data rate, at time t_0 , only a low number of calls can be transferred. As a consequence, the cognitive engine looks for a combination of actions that can result in a higher number of calls. The combination analyzed at t_1 implies to decrease the voice packet interval, i , and lower the retransmission number, r , while keeping the data rate high. However, as the number of calls, n , stays low, such a combination seems not to be successful. The combination analyzed at t_2 involves increasing all the variables: as witnessed by the increase in the number of calls, that combination should allow the system to forward a higher number of calls while not giving up to the quality of service requirements promised.

3.2.4-B TCP Throughput Estimation in an Ad-hoc Network

The second scenario is an IEEE 802.11-based wireless ad-hoc network with a chain topology, as shown in Figure 3.19. At one end of the chain, there is a service provider,

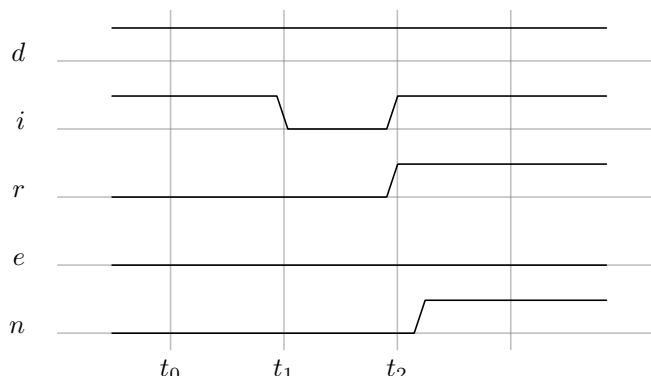


Figure 3.18: Feedback example. The action profile tested at t_1 (high data rate, d , low voice packet interval, i , and low retransmission number, r) does not permit to achieve a greater number of calls while providing an acceptable quality. The action profile tested at t_2 (high d , high i , and high r), on the contrary, seems to allow the system to forward all the calls while providing the callers with an acceptable quality.

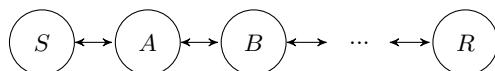


Figure 3.19: Network topology in the second scenario

while at the other end there is a service consumer.

The only cognitive node is the one in which the service provider is implemented. In this case, instead of the number of calls that can be placed at a given time, the cognitive engine controls the throughput of the system. A possible WS-Policy describing the service is shown in Figure 3.20.

The input variables considered for this scenario are: the bit error probability, the data rate at the physical layer, the number of nodes in the network, the use of fragmentation, and the use of the RTS/Clear to Send (CTS) mechanism. The only output variable is the variable to be predicted by the reasoning mechanism, i.e. the throughput that can be achieved by the network. The range of variation of the variables is summarized in Table 3.3.

Some variables have domains similar to those analyzed in the previous scenario, but there are also variables that are inherently binary, for example, the use of fragmentation and the use of RTS/CTS handshake. This clearly facilitates the mapping of variables onto concepts, as no pre-processing operation is needed. Regarding the other variables, since,

Table 3.3: Domains of the input variables in the second scenario

Property	Values	Abbreviation
Bit error rate	$\{0, 3 \cdot 10^{-8}, 3 \cdot 10^{-7}, \dots, 3 \cdot 10^{-5}\}$	e
Physical data rate	$\{1, 2, 5.5, 11\}$ Mb/s	d
Number of nodes	$\{3, \dots, 9\}$	n
MAC fragmentation	{on, off}	f
MAC RTS/CTS handshake	{on, off}	r

```

<wspes:Policy
  xmlns:wspes="..." xmlns:qos="..." xmlns:op="...">
  <wspes:ExactlyOne>
    <wspes:All>
      <wspes:Assertion name="TransferSpeed"
        assertiontype="Capability">
        <wspes:Expression>
          <wspes:Parameter>Throughput
          </wspes:Parameter>
          <wspes:Value>350</wspes:Value>
          <wspes:Unit>kbps</wspes:Unit>
          <wspes:ExactlyOne>
            <wspes:Operator>greater</wspes:Operator>
            <wspes:Operator>lower</wspes:Operator>
          </wspes:ExactlyOne>
        </wspes:Assertion>
      </wspes:All>
    </wspes:ExactlyOne>
  </wspes:Policy>
    
```

Figure 3.20: The WS-Policy (based on [5]) describing the service provided in the second scenario states that users can be choose between two levels of performance, under optimal network conditions. In case the cognitive engine cannot keep network conditions at an optimal level, the WS-Policy may be republished to inform potential users that only a best effort service will be available (bold part removed).

similarly to the study case presented in Section 3.2.4-A, the combinations of input variables have been exhaustively simulated, also in this case thresholds are identified with the mean values of the variable domains. However, differently from what has been done in the previous scenario, no gray area around the thresholds is defined for these input variables. As a result, the cognitive entity attempts to classify every possible situation, whereas in the other case, prediction was restricted only to situations presenting unambiguous input values.

In the considered case, no automatic data rate fallback algorithm is employed, and both fragmentation and RTS/CTS mechanisms, as well as the number of nodes, are independent from any other variable. The four considered variables affect the error rate, and all together exert their influence on the throughput. This means that the resulting FCM is represented by:

$$F_2 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & f_{et} \\ f_{de} & 0 & 0 & 0 & 0 & f_{dt} \\ f_{re} & 0 & 0 & 0 & 0 & f_{rt} \\ f_{fe} & 0 & 0 & 0 & 0 & f_{ft} \\ f_{ne} & 0 & 0 & 0 & 0 & f_{nt} \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (3.7)$$

and its graphical counterpart is shown in Figure 3.21.

The resulting performance of the cognitive entity is plotted in Figure 3.22: the evolution of the output variable as a function of time is shown at the top of the figure.

The two figures at the bottom show that, after an initial transient phase, the causal relationships among the concepts in the FCM stabilize and allow the reasoning mechanism to score above 94% correct predictions (second graph from the top of Figure 3.22). In particular it improves performance by about 4.58% ($\pm 0.54\%$, confidence interval com-

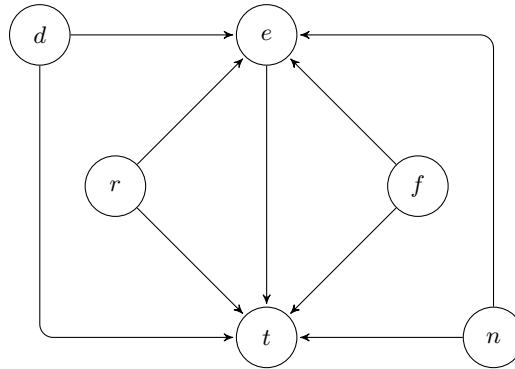


Figure 3.21: FCM employed in the second scenario. t , n , e , d , r , and f stand for throughput, number of nodes, bit error rate, physical data rate, RTS/CTS handshake, and fragmentation, respectively.

puted with $p = 0.995$, DoF= 14) with respect to the case in which we keep our decision fixed. In other terms, in most occasions, the imperfections of the communication channel could be masked, granting the service consumer the ability to experience stable quality of service.

3.2.5 Conclusion

This section illustrated an application example of the tool introduced in Section 3.1.

It aimed to define a flexible and adaptable solution to support the quality of service requirements of services in a SOA scenario. Specifically, it addressed the issue of how information and interactions among the different layers of the protocol stack can be exploited to develop a reasoning engine enabling cognitive networks to support SOAs deployment in heterogeneous networks. We have referred to the union of these entities as Cognitive Service-Oriented Infrastructure.

After explaining the operations of CoSOI from a high-level point of view, we proposed two case-studies, both focused on wireless networks. The first one dealt with call admission control in a centralized environment, whereas the second dealt with throughput estimation in a distributed environment.

Results illustrate the potential of the proposed architecture, by validating its capability to learn the relationships among parameters and performance in a highly variable scenario such as wireless networks.

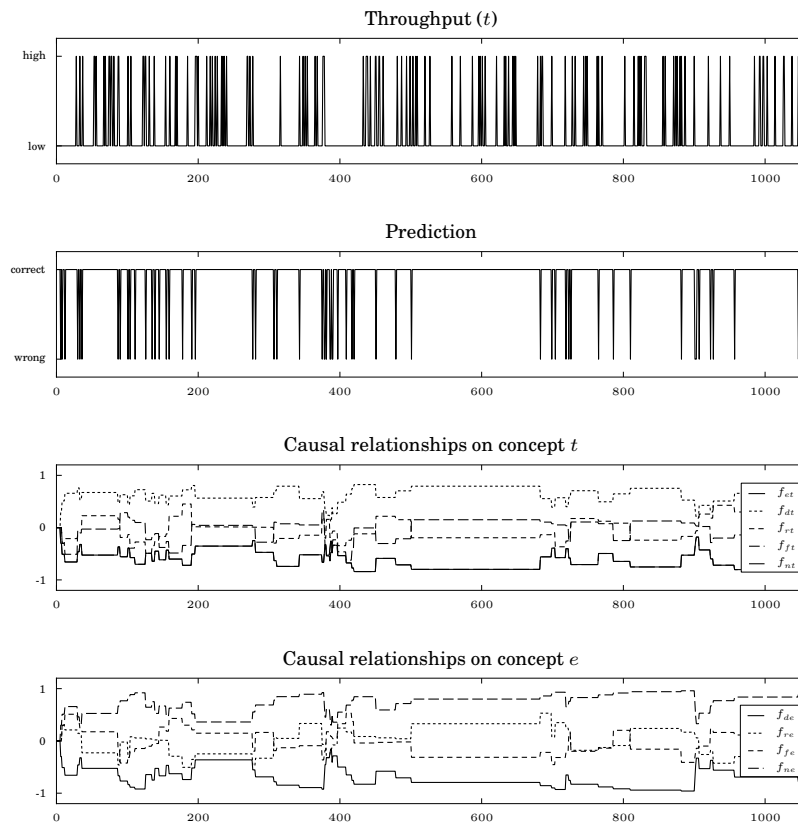


Figure 3.22: Prediction performance achieved by the reasoning entity in the second scenario – $\eta = 0.3$. When predictions are correct, the structure of the FCM does not change.

3.3 Cognitive Rate Adaptation in Wireless LANs: Inference using Fuzzy Cognitive Maps

Rate adaptation represents a relevant issue in optimization of Wireless Local Area Network (WLAN) performance. This section proposes to employ the cognitive approach we described in Section 3.1 to perform rate adaptation, which is able to learn cause-effect relationships without any a-priori knowledge. Special attention is given to the details related to the implementation of the cognitive architecture. Results demonstrate the potential of the proposed scheme⁵.

3.3.1 Introduction

IEEE 802.11 is the most widely adopted standard for wireless indoor communications. It defines physical layer and data-link layer specifications to be adopted for high-speed communications in a WLAN [63].

To overcome the common problems that impair wireless communications, such as noise, multi-path fading and interference, the standard specifies a possible set of modulations that can be selected to transmit data, based on current channel conditions. The idea is that, when needed, a station can change its modulation, thereby tuning the trade-off between robustness and achievable performance.

As channel conditions may vary over time, adaptive schemes for properly choosing a modulation are needed. However, the definition of algorithms to perform dynamic rate switching is not specified by the IEEE 802.11 Standard [63]. As stated in paragraph 9.6:

Some PHYs have multiple data transfer rate capabilities that allow implementations to perform dynamic rate switching with the objective of improving performance. The algorithm for performing rate switching is beyond the scope of this standard, but in order to ensure coexistence and interoperability on multirate-capable PHYs, this standard defines a set of rules to be followed by all STAs⁶.

As a consequence, different approaches have been proposed in literature [64, 65, 66, 67], all based on relevant a-priori knowledge of the problem.

Recently, a novel networking paradigm, called “cognitive networking”, has been developed, with the purpose to equip network nodes with some sort of intelligence, so that they are able to cope with varying network conditions and adapt themselves in order to enhance performance.

Though in its early infancy, the cognitive networking paradigm has successfully been applied to specific problems, such as throughput prediction [68] and rate adaptation in WLANs [69].

Following the promising results of the cognitive approach, we propose in this section a cognitive rate adaptation scheme that makes no use of a-priori knowledge. Instead, it performs causal reasoning and learning, based only on measurements collected by the receiver and aims to maximize the end-to-end throughput. Results demonstrate that

⁵Part of this work was published in the proceedings of the IEEE International Conference on Communications (ICC'11), Kyoto, Japan, 2011 [48].

⁶PHY stands for “Physical Layer”, STA stands for “(wireless) station”

achieved performance is in line with—and in some cases better than—the above mentioned rate-adaptation algorithms.

The remainder of this text is structured as follows. Section 3.3.2 reviews the major data rate adjustment algorithms and shows a comparison with the proposed scheme, which is described and employed in a wireless ad-hoc network in Section 3.3.3. Performance is evaluated in Section 3.3.4 by comparison with classic data rate adjustment algorithms. Finally, Section 3.3.5 concludes the section by offering some consideration on the proposed scheme.

3.3.2 Related Works

The first data rate control algorithm being published is the Automatic Rate Fallback (ARF) mechanism, designed to work with the first version of the standard, in which only two data rates were available [64]. The algorithm keeps track of the number of missed acknowledgment frames (ACKs): if two consecutive frames are lost, the algorithm selects the lowest data rate. The highest data rate is selected when either ten ACKs in a row are correctly received or a countdown timer expires. Noteworthy, ARF has been applied also to later versions of the standard, with multiple data rate capabilities. The main drawback of this approach is the lack of stability in slowly changing scenarios. Data rate oscillations are present because, after ten ACKs are received correctly, the algorithm will nonetheless try to increase the data rate, even though the previous rate was the optimal one, so that it is forced to back off if necessary.

The Adaptive ARF (AARF) algorithm aims to solve such problem by implementing a binary exponential backoff-based counter [67]. Dynamic adaptation of such counter results in lowering the the number of unnecessary modifications.

Both ARF and AARF are incapable of discerning between errors due to medium characteristics and errors due to collisions with other interfering nodes. Such lack of capacity to distinguish the nature of losses is a problem in presence of bursty interference: in case of successive collisions, both algorithms decrease the data rate although this is not necessary, thus affecting the overall performance.

The Collision-Aware Rate Adaptation (CARA) algorithm addresses this problem by means of the Request to Send/Clear to Send (RTS/CTS) handshake [66]. The rationale is that transmission errors of RTS frames are most likely due to collisions. On the other hand, errors in data transmitted after the RTS/CTS handshake should not be due to collisions (since the channel is reserved). This way, the transmitter is able to distinguish the nature of errors and decreases the data rate only in case of channel errors.

The AARF-Collision Detection (AARF-CD) scheme [65] is similar to CARA. It is grounded on the AARF scheme and it makes use of the RTS/CTS frames in case errors are due to channel contention. The main difference to CARA is that, in AARF-CD, the RTS/CTS handshake is enabled when the data rate is increased and disabled when the data rate is decreased. This follows the rationale that the first failure is probably due to channel errors, and reserving the channel is useless.

A characteristic that is common to the mentioned methods is that all use specific a-priori information.

Differently, a cognitive approach should in principle be able to act without such knowledge, and it should adapt to different situations without any modification. For example, the cognitive system proposed in [69] entrusts the cognitive engine to determine which is the best rate to be used. Performance history as a function of the data rate is stored in a knowledge base. Using as mean the value of data rate that gave the best performance, a random distribution is generated. The drawn random value indicates the data rate that should be used.

The approach we propose learns by experience, as well, but it employs reasoning based on causal relationships detected by a reasoning engine rather than on near-past experience. The main concept is to identify and update cause-effect relationships, in such a way to incrementally build knowledge on the phenomenon to control (data rate, in the considered case) with the ultimate goal to maximize throughput.

The proposed data rate control mechanism is based on a feedback loop from the receiver node (N2) to the sender (N1), to which information about measured throughput, Signal-to-Noise Ratio (SNR) and Frame Error Rate (FER) is provided—whenever relevant changes of each considered parameter take place.

3.3.3 The Proposed Approach

The proposed data rate control mechanism is shown in Figure 3.23. Section 3.3.3-A will discuss how the problem can be adapted to the reasoning formalism introduced in Section 3.1, whereas Section 3.3.3-B will analyze the details related to the implementation of the architecture.

3.3.3-A Translating the Problem for the Reasoning Formalism

As can be seen in Figure 3.23, station N1 is in charge of reasoning and acting. It implements a cognitive engine based on Fuzzy Cognitive Maps (FCMs), a tool that enables causal reasoning and that we introduced and illustrated in Section 3.1.

To summarize, let us just say that they can be seen as directed labeled graphs, in which nodes represent generic concepts, and edges between any two nodes represent the causal relationships that exist among them.

Let us now determine the FCM for the considered problem, as outlined in Section 3.1.4. By first inspecting the problem, we can identify the concepts that play primary roles.

Clearly, the main purpose of the mechanism is to tune the data rate. As a consequence we can define concept d to denote data rate tuning and classify it as an ‘action’ concept.

Data rate will be adjusted according to the current level of noise and amount of collisions, both of which cause the error rate to vary. Therefore, it is reasonable to combine them and form concept e , symbolizing the frame error rate, and classify it as an ‘environment’ concept. It could be objected that if we considered collisions and noise separately, we could achieve better results. However, it is not realistic to suppose that a node can distinguish such events, unless it implements a mechanism similar to that present in CARA and AARF-CD, which is not the case in this implementation.

The final aim of the data rate management scheme we propose is to maximize data throughput. This can be translated into the creation of the throughput concept t , which

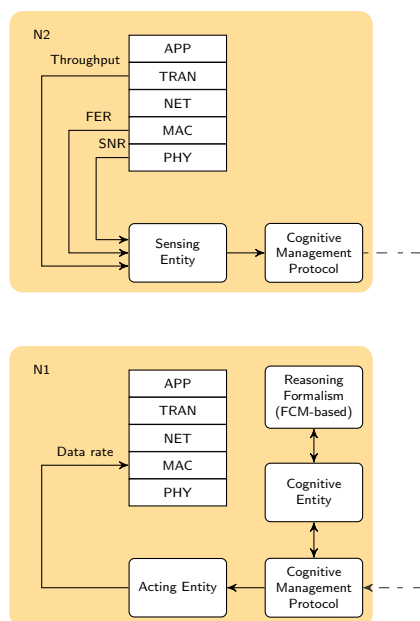


Figure 3.23: Specific implementation of the proposed cognitive data rate control mechanism. Station N2 gathers information about throughput, Frame Error Rate (FER) and Signal-to-Noise Ratio (SNR) that it sends to station N1.

belongs to the ‘QoS’ class.

The resulting general adjacency matrix will be the following:

$$F = \begin{pmatrix} 0 & f_{ed} & f_{et} \\ f_{de} & 0 & f_{dt} \\ f_{td} & f_{te} & 0 \end{pmatrix} \quad (3.8)$$

As we explained previously, elements of the matrix represent the causal relationships that exist between any two concepts, and can be mapped on the continuous set $[-1, 1]$. Negative values indicate that there is an inverse causality between the considered concepts, while positive values indicate there is a direct causality. As an example, let us refer to element f_{dt} , which measures the causal relation between the data rate and the throughput. A positive value means that increasing (decreasing) the data rate should cause an increase (decrease) of throughput. A negative value means that increasing the data rate should cause the throughput to decrease (and vice versa).

Regarding concept domains, we opted to apply the binary set $\{-1, 1\}$ to all concepts. The use of such a domain allows cause-effect relationships to exist even when concepts decrease (or reach a low value). To clarify this consideration, let us realistically suppose that there is an edge between the error rate concept (e) and the throughput concept (t) and that its label has a negative value. Depending on the domain on which concepts are mapped, such relationship may convey different meanings. If they are mapped on $\{0, 1\}$, it means that when the error rate is low, no cause-effect relationship exists between the two concepts. If they are mapped on $\{-1, 1\}$, it means that a decrease in the error rate is likely to cause an increase in terms of throughput.

As described in Section 3.1.3-A, reasoning is done by multiplying the state vector, i.e. the vector containing the values of all the concepts at a given time, by the adjacency matrix representing the FCM, and by repeating the multiplication using the result obtained, until the operation converges (either to a fixed point, or to a limit cycle).

The FCM learns by updating the relations between its concepts, which is done in real time. The specific learning algorithm we chose to use in this scenario is the Differential Hebbian Learning (DHL), which we thoroughly described in Section 3.1.3-B. Such algorithm is based on the observation that humans tend to infer causal relationships when they notice correlated variations in two variables [42]. Accordingly, a generic matrix element f_{ij} is updated proportionally to the product of the time derivatives of concepts C_i and C_j . The reader is referred to Equations (3.1)–(3.3) to inspect the process more in depth. The equation governing the learning process (3.3) is reported here for convenience:

$$\dot{f}_{ij}^t = -f_{ij}^{t-1} + \dot{C}_i^t \dot{C}_j^t$$

Elements on the diagonal are set to zero and never updated: this is because we assume that no concept can cause itself. For additional information on FCMs we point the interested reader to Section 3.1.

3.3.3-B Implementation Details

Using Figure 3.24 as reference, let us first review the main components that constitute the general architecture and then discuss the implementation for the specific problem.

The general architecture

The components of the proposed architecture are:

- acting and sensing variables;
- acting and sensing entities;
- the cognitive entity;
- the cognitive management protocol.

Acting variables are used to reconfigure a particular aspect of a node. The interface is simple on purpose and it consists only of an instruction pair, the meaning of which depends on the specific element to be adjusted. For instance, if we want to control the congestion window of the transport protocol, the command pair will be ‘increase’ and ‘decrease’. If we want to regulate packet fragmentation, the instructions will be ‘turn on’ and ‘turn off’. Though the idea of designing a simplified interface is valid regardless of the reasoning formalism employed, it is particularly fitting when using an FCM with binary concepts.

Sensing variables are in charge of monitoring events and transmitting relevant updates to the cognitive entity. We chose to have only push-type notifications. Therefore, the cognitive entity cannot request any update and has to reason based on the available information. In the proposed implementation, to distinguish significant changes from random variations, an approach based on Exponentially Weighted Moving Average (EWMA)

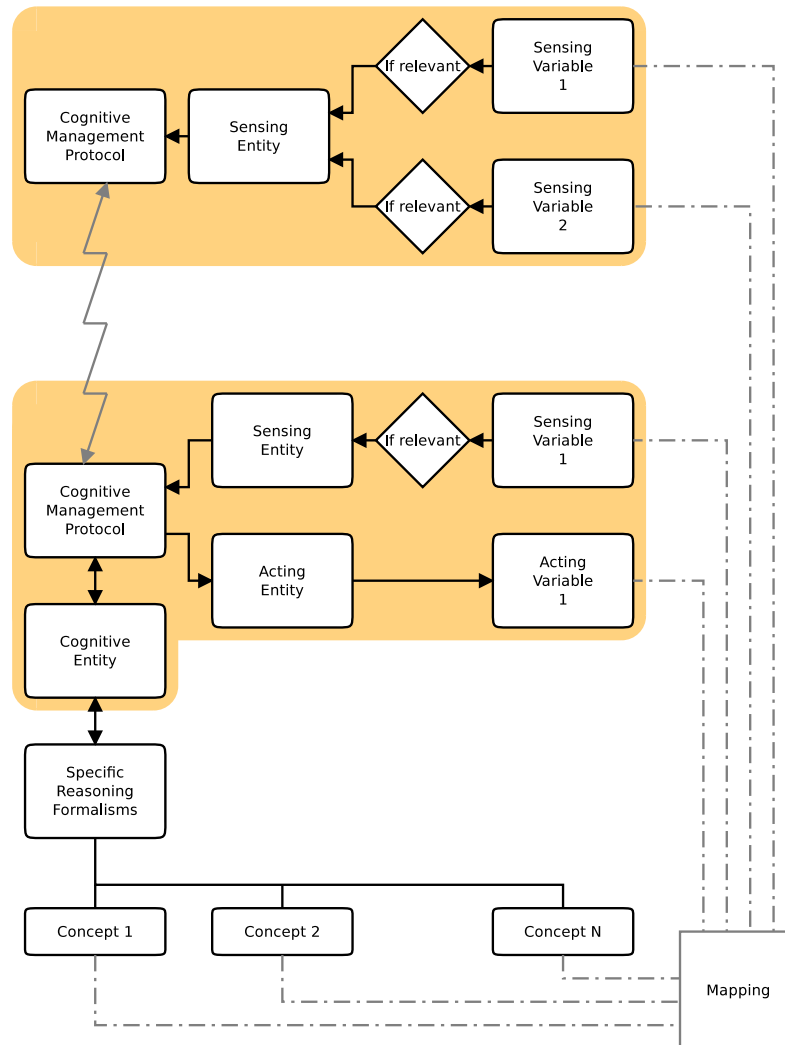


Figure 3.24: High-level view of the proposed cognitive network architecture

control charts is used [70]. For each source of information a control chart is drawn. The controlled quantity is an estimation of the mean value, computed as an EWMA of the actual mean, and upper and lower control limits (UCL and LCL) are proportional to the standard deviation. When the estimation crosses either one of the boundaries, an update is sent to the sensing entity.

Acting and sensing entities function as aggregators of acting and sensing variables, respectively. Their main task is to minimize communication overhead, while supporting the cognitive functionality of the network.

The role of the cognitive entity is to reason, by interrogating a specific reasoning formalism, and guide the acting variables. Though FCMs have been chosen as the reasoning formalism for this implementation, it should be stressed that the adoption of a specific reasoning scheme can naturally be replaced in favor of other approaches. It was decided on purpose to separate the cognitive entity from the reasoning formalism, in order not to constrain network designers in any way.

According to our vision, the cognitive management protocol is implemented in all network nodes and is used to exchange sensed information and tuning commands. It is worth to notice that while a cognitive node must implement the cognitive management protocol, there is no need for it to have any sensing or acting variables at all. For instance, a centralized wireless access point may be equipped with reasoning capabilities, but receive updates from and send commands to wireless stations.

Furthermore, though a single cognitive entity is used in this implementation, in principle no constraints are placed on the number of cognitive entities that could be installed. Multiple entities may coexist in the same network, and can coordinate with each other via the cognitive management protocol.






The specific implementation

Results from the reasoning process are used to adjust the data rate to be used to transmit data frames.

As Station N2 has only sensing capabilities, it monitors transport-layer throughput, frame error rate and signal-to-noise ratio of incoming frames.

To better analyze the achievable performance while being able to evaluate the communication overhead, a fictitious wired channel is added to provide feedback information. Information from N2 to N1 is sent thanks to the previously mentioned push-mechanism: if N2 detects relevant variations in the monitored variables, it updates N1 with the sensed values.

The process can be explained by looking at Figure 3.25, which shows the action model of a generic sensing variable, as it has been implemented⁷. Measurements are saved in a vector, possibly after some pre-processing. An example of pre-processing at this stage is the creation of a frame error rate measure from the separate measurements of the number of the frames correctly and incorrectly received. As soon as the first measurement is stored in the array, a timer is run. At its expiration, the number of measurements is checked:

⁷All action models included in this text are loosely based on the representation known as Business Process Model and Notation (BPMN). Briefly,  indicates the start of a process,  indicates the end,  indicates an intermediate event triggering some action,  and  denote positive and negative answers, in order.

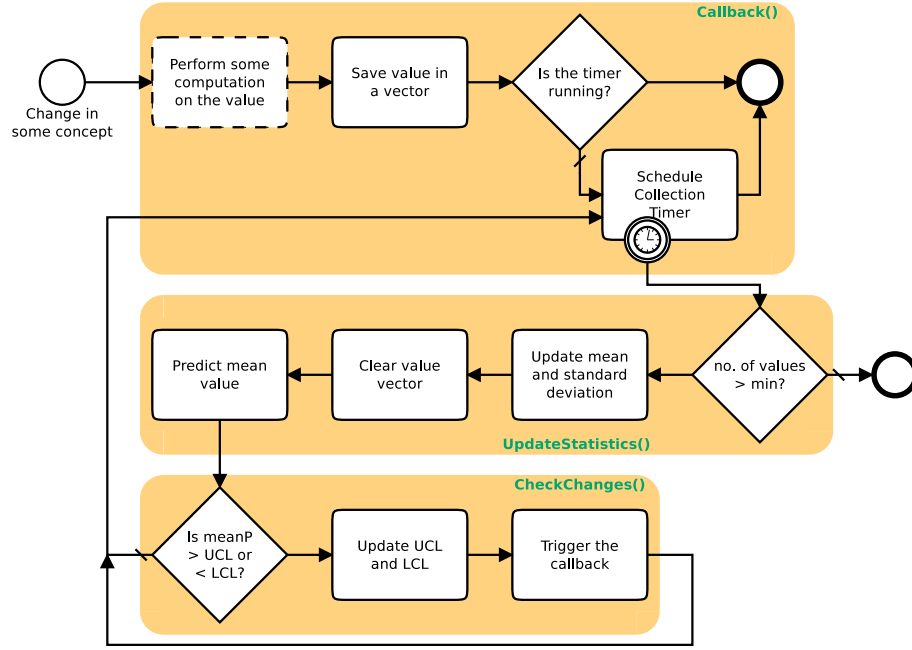


Figure 3.25: Action model of a sensing variable. Different groupings identify different functions, as implemented in the simulator. Function names are reported as labels for each grouping.

if it is lower than a minimum, another round of measurements is scheduled, in order to ensure significant averages. Upon reaching a sufficient number of measurements, standard deviation (σ) and mean (μ) are computed. An EWMA-based prediction $\hat{\mu}$ of the actual mean value is performed using a weighting coefficient β in the continuous interval $(0, 1]$, i.e.:

$$\hat{\mu}^t = \beta \cdot \mu^t + (1 - \beta) \cdot \hat{\mu}^{t-1} \quad (3.9)$$

UCL and LCL are updated only when the controlled value ($\hat{\mu}$) crosses either boundary. The new values are computed according to the following formulae:

$$UCL = \hat{\mu} + C\sigma \cdot \sqrt{\frac{\beta}{2 - \beta}} \quad \text{and} \quad LCL = \hat{\mu} - C\sigma \cdot \sqrt{\frac{\beta}{2 - \beta}} \quad (3.10)$$

where β is the same value used as the weighting coefficient in the EWMA-based prediction and C is a real parameter in $(0, +\infty]$.

The cognitive engine in N1 encodes the causal relationships among the monitored variables and forwards the data to the reasoning formalism to have it reason on the problem.

By way of example, let us suppose that N2 sends two subsequent updates indicating there has been an increase of the error rate followed by a decrease of throughput. Accordingly, the reasoning formalism will encode the rule that errors and throughput are bound by a negative causal relation. As the throughput value has decreased, the reasoning formalism is requested to provide a solution. If a solution can be found, it will be implemented through the acting entity, which adjusts the data rate. However, it is also

possible that the reasoning process is incapable of providing any solution or that the solution found is ineffective. These situations can happen either because learning capabilities are still not well developed or because adjusting the data rate is useless in that particular situation. In either case, a reinforcement mechanism kicks in and modifies the data rate.

A deeper analysis of the functioning of the cognitive entity can be illustrated by referring to Figure 3.26. The entire process is triggered by an update sent by a sensing variable through the relative sensing entity. Due to the asynchronous nature of events, a timer is used in order to avoid enforcing false causal relationships. Indeed, if the time gap between two updates is relatively small, it is unlikely that the one happened before caused the other. As a consequence it is reasonable to consider two such events as contemporary. It should be noticed that, from the logical point of view, such functionality belongs to the sensing entity. However, in order to ease the implementation of the scheme in the simulator, after verifying that this development choice would not have impacted on the overall results, it has been implemented within the cognitive entity. This is highlighted in Figure 3.26, where the functions of the sensing entity are drawn on a striped background.

Causes are sought for the variables that have undergone a change: the idea is that changes happened previously may have induced the changes in the concepts being evaluated now. Afterwards, the latter are labeled as possible causes for any potential variation happening in the immediate future.

Two timers have been introduced in this implementation with the aim to (i) reduce the number of unnecessary oscillations, and (ii) avoid that the reasoning engine yields sub-optimal solutions as much as possible. The first objective is accomplished by enforcing a solution for a specific amount of time. Independently from other changes, the action profile remains active until the timer expires (**PersistenceTimer**). This prevents the reasoning engine from continually trying to find and impose new solutions in a fast-changing environment. The second objective is achieved by monitoring the quality concepts: as soon as one such concept enters a non-optimal state another timer is fired (**BadShapeTimer**). The timer is promptly deactivated if the concept in question returns to the optimal state. When the timer expires the concept is checked and, in case it is still in a non-optimal state, the reasoning engine is queried to find a new solution. A possible implementation of such a mechanism is shown in Figure 3.27, in which the use of a three-step counter makes it possible to employ different actions every time the timer expires: the first time the action profile is inverted and kept for a specific amount of time (**ShortDelay**). Then it is enforced for a second round and finally it is inverted again (to reflect the original solution) and kept for a longer time interval (**LongDelay**).

After checking the timers, the reasoning formalism is interrogated. In the case it succeeds in finding a proper solution, the actions specified therein are enabled and enforced by means of the previously mentioned timer.

If no solution can be found, the process stops. Indeed, if the reasoning formalism does not find a solution, it may be either because no action can improve the current situation or because the reasoning formalisms *believes* that no action can improve the current situation. In the former case, nothing can be done. In the latter case, however, an additional refinement can be developed.

Two are the possible sub-cases that lead to this situation: either the reasoning formal-

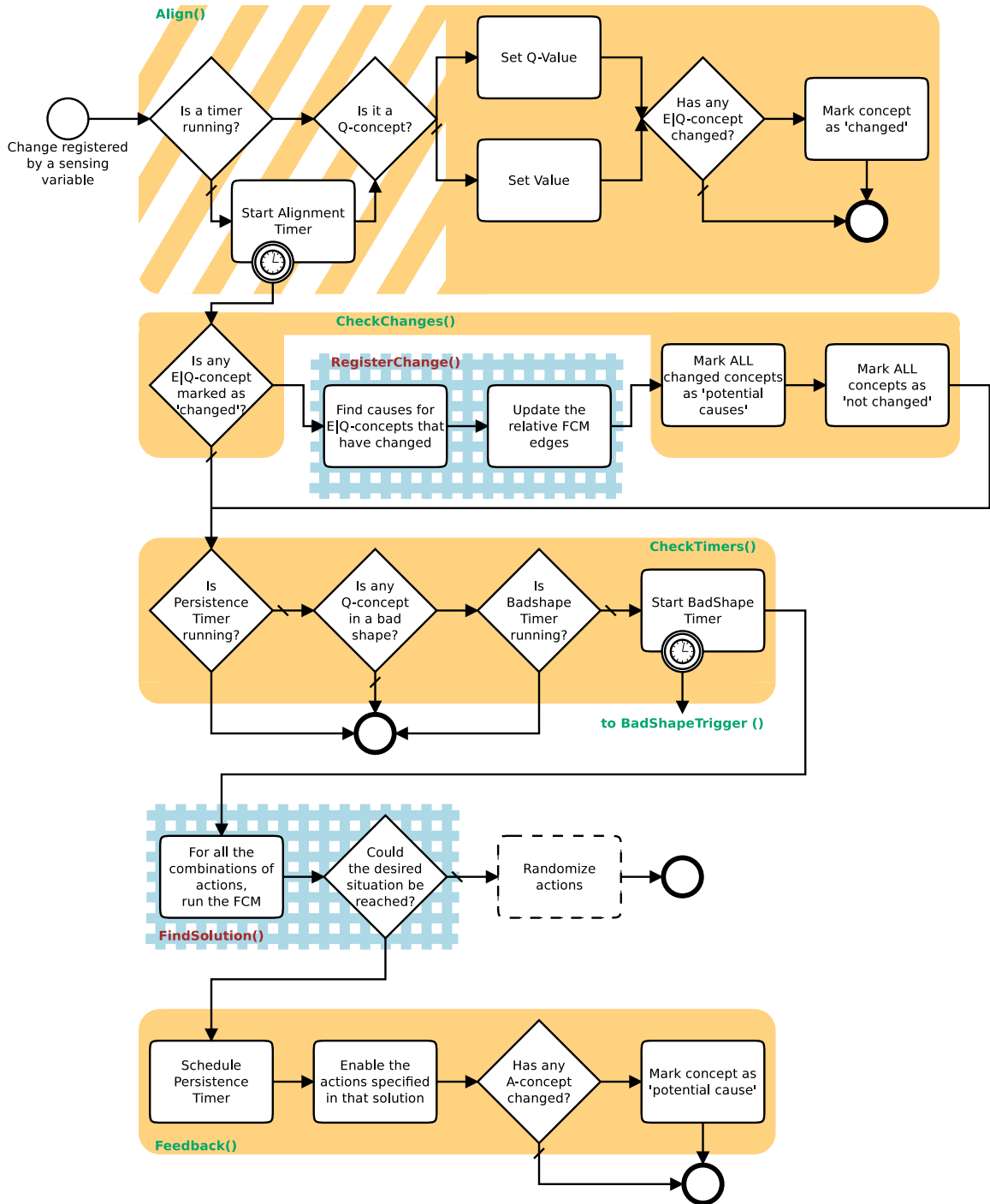


Figure 3.26: Action model of the cognitive entity. The striped background indicates activities that logically belong to the sensing entity, but are implemented in the cognitive entity for convenience. The squared background indicates activities that belong to the reasoning formalism and may differ depending on the specific formalism employed—they are reported in the diagram for completeness. Dashed actions are not mandatory.

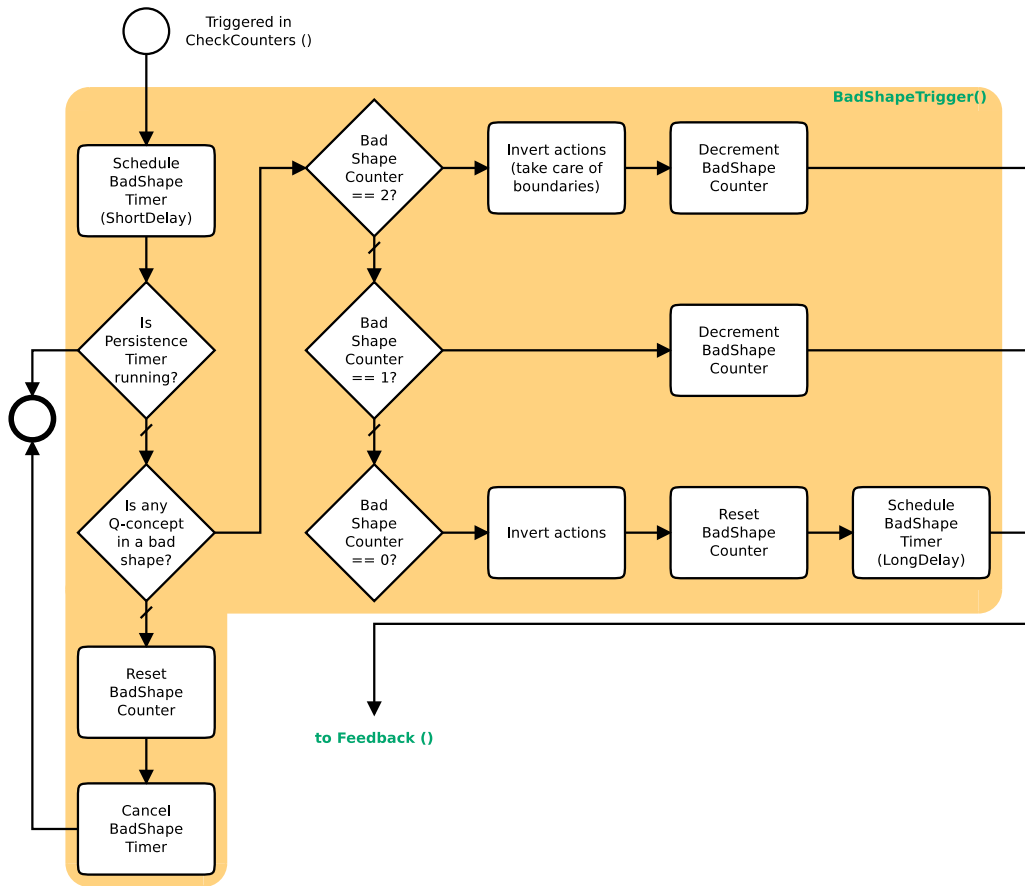


Figure 3.27: Specific implementation of the mechanism to avoid sub-optimal solutions based on a three-step counter. Every time the timer expires the counter is decremented, and different actions are taken. Different time intervals are used, in order to limit the use of the reinforcement mechanism.

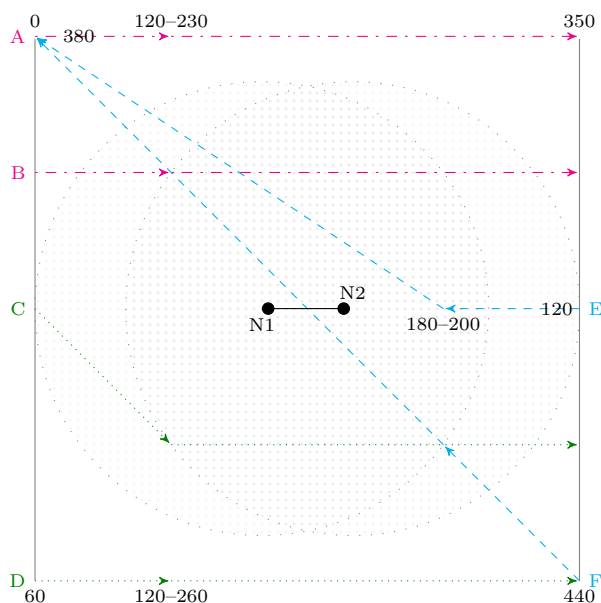


Figure 3.28: Simulation setup. Numbers along the paths indicate the second in which the node started and stopped. For instance, interfering node A started moving at 0 seconds, stopped at 120, started again at 230 and finally stopped completely at 350 seconds. The shadowed area around nodes of interest N1 and N2 represents the transmission range.

ism has incorrectly encoded experience or it has no experience at all. The solution for both sub-cases is to include a function that randomizes the action profile (dashed block in Figure 3.26), with the purpose to ensure that the reasoning formalism actually learns from experience.

To motivate such consideration let us think of a person with no a-priori knowledge of the world whatsoever (this may be the case of a newborn, for instance). If no one instructs her how to behave (indirect experience), she will have to learn by trial and error (direct experience).

The same observation holds even if that person has developed (for whatever reason) wrong beliefs. Imagine her in a room with nothing but a light switch and a window. Suppose that as she turns on the switch for the first time (i.e. she makes experience of the switch), the window cracks open. Whereas other human beings, could think of such a situation as a coincidence, she, having no other experience, is clearly led to believe that turning on the switch *causes* the window to open. Again, barring any external aid, this wrong belief can be corrected by repeated manipulation of the switch, which will eventually show that there is no correlation between the two events.

In the same way, if we exclude any type of external intervention, e.g. by experts, the only means by which the reasoning formalism can learn is through trial and error.

3.3.4 Simulation Results and Discussion

Simulations are performed using the NS-3 network simulator. Network setup is shown in Figure 3.28. N1 and N2 are the nodes of interest (cognitive): N1 continuously transmits

data to N2 using the User Datagram Protocol (UDP) as the transport protocol, while adjusting the transmission data rate in order to maximize the end-to-end throughput. Nodes A to F form three pairs of non-cognitive interfering nodes that exchange data to each other using UDP. All nodes are wireless, and equipped with IEEE 802.11b capabilities. The choice of amendment “b” of the IEEE 802.11 standard is motivated by the fact that such version is logically equivalent to other variants for the application of the proposed scheme, yet providing more control over the simulated scenario. Cognitive nodes hold their position throughout the whole simulation, while non-cognitive nodes wander around following predefined paths (drawn as dashed, dotted, and dash-dotted lines in Figure 3.28), with speeds of 1 to ≈ 1.4 m/s.

Such a scenario is interesting because it shows how data rate control algorithms adjust the data rate in presence of errors due to collisions. Comparison of the proposed scheme is done with four well-known adaptive algorithms reviewed in Section 3.3.2, i.e. ARF [64], AARF [67], AARF-CD [65], and CARA [66].

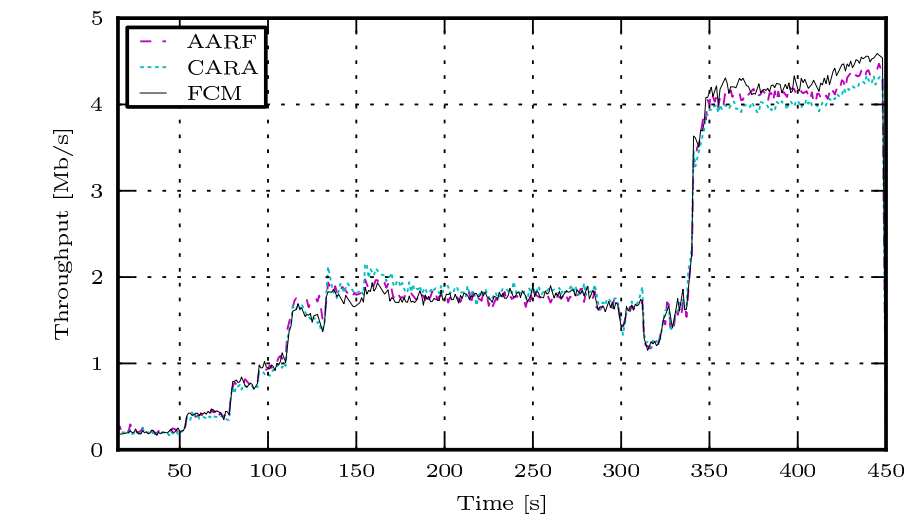
Simulation results are summarized in Table 3.4 and in Figure 3.29a. Though the data in the table is referred to all the algorithms, the graph only shows a subset of them for better visualization⁸.

Based on the review performed in Section 3.3.2, it could be expected that the algorithms that are able to discern between losses perform better than the other algorithms. However, the algorithms that cannot distinguish the nature of losses (ARF and its evolution AARF) achieve better results than those that can (CARA and AARF-CD). This can be motivated by the fact that both CARA and AARF-CD rely on transmission of control frames (RTS/CTS), which inevitably reduce the amount of channel resources that can be used to carry out the actual transmission of data.

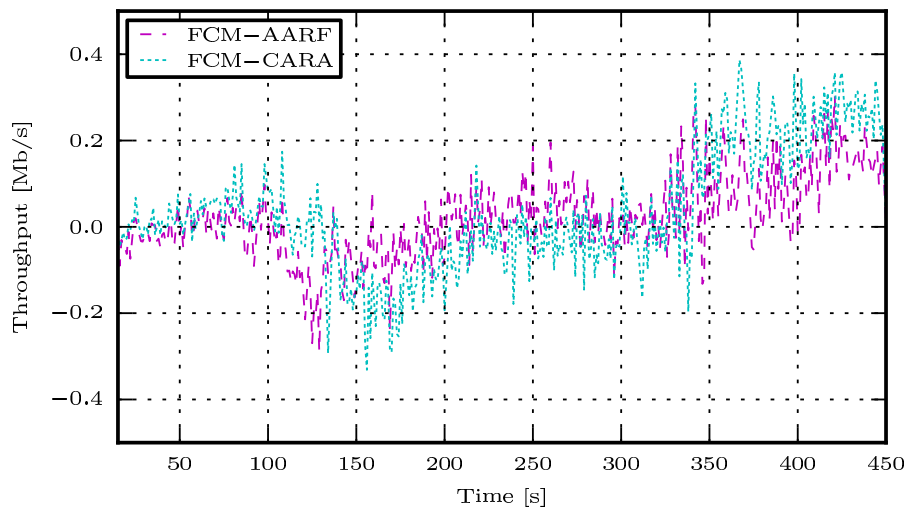
It is also interesting to evaluate the behavior of the cognitive approach, which is not able to distinguish the nature of losses. In particular, it is worth noting that the cognitive approach, unlike the other approaches, starts with no a-priori knowledge. In this view, it is relevant to check that the performance is at least in line with the other well-known schemes, based on detailed empirical or mathematical models. In fact, as can be seen in Figure 3.29a, the temporal evolution of the simulation shows that our approach is in line with the others, while manages to transfer a greater amount of data, as reported in Table 3.4. Figure 3.29b shows the throughput difference between the proposed approach and the other two data rate adjustment algorithms analyzed. After an initial phase in which the performance level is slightly lower, the cognitive approach is able to perform better, thanks to its learning capabilities.

Figure 3.31a provides additional details on the learning algorithms by showing the evolution of the throughput (solid curve) and the data rate (dotted curve) concepts as a function of time. It can be noticed that the data rate oscillates in the initial part between the values of 5.5 and 11 Mb/s. This means that the reasoning formalisms did not converge to any solution and the reinforcement mechanism has been triggered. After some time, the cognitive engine recognizes that the throughput increased, after the data rate had decreased to 5.5 Mb/s. Therefore, it infers that there is a negative causal relationship between the data rate and the throughput (element f_{dt} in the adjacency matrix in Eq. 3.8),

⁸The complete version of the graphs are available at <http://disi.unitn.it/~facchini/icc11/>

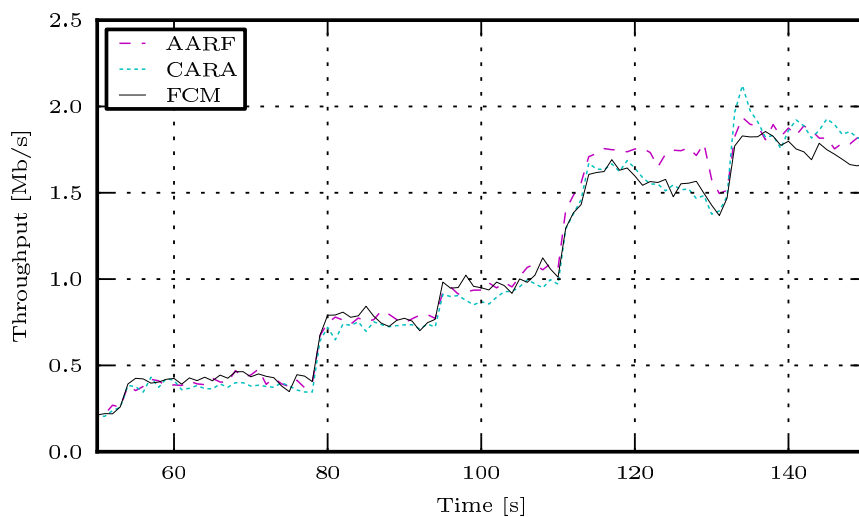


(a) Overall behavior

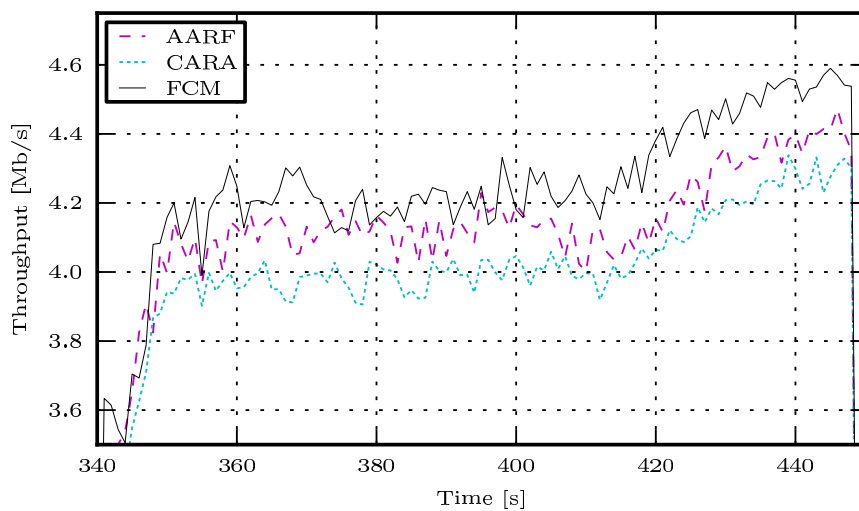


(b) Throughput difference

Figure 3.29: Throughput comparison with other data rate adjustment algorithms.

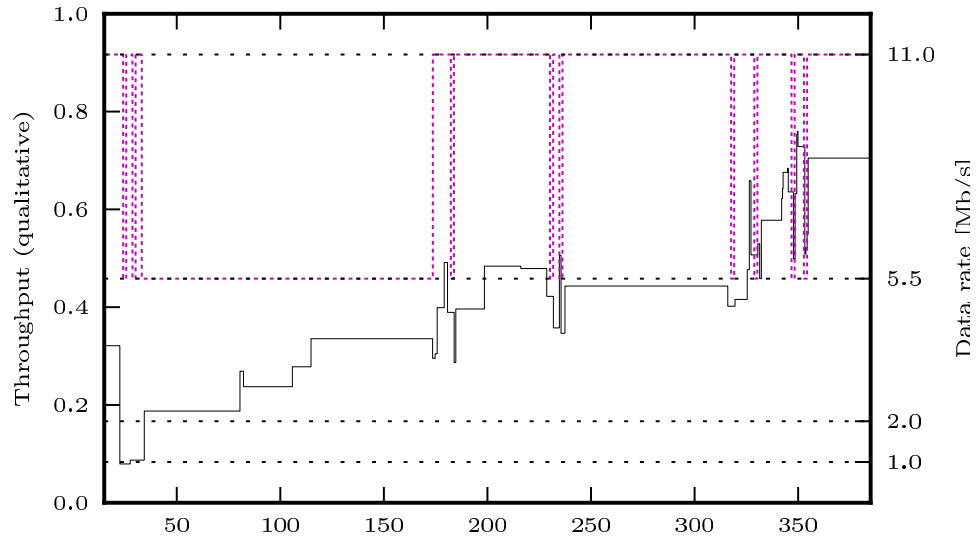


(a) Detail of the 50-150 s period

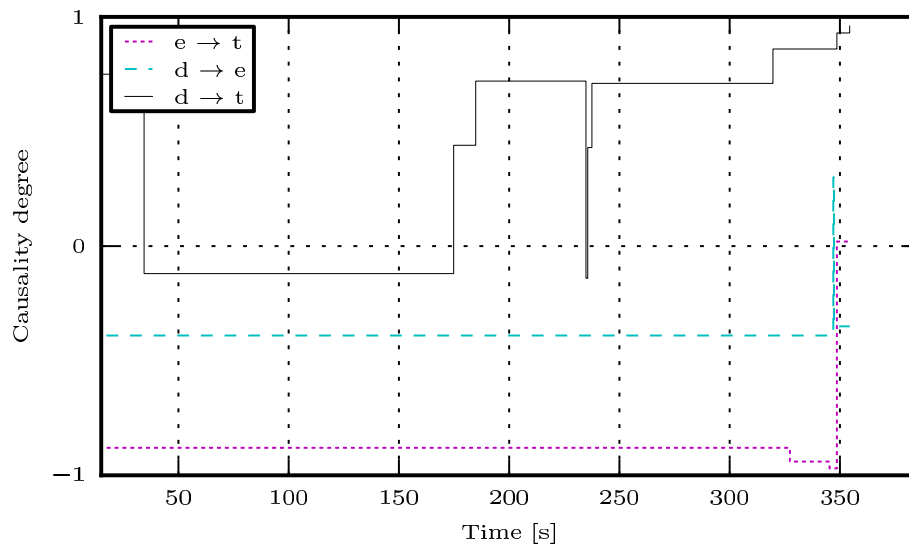


(b) Detail of the 340-450 s period

Figure 3.30: Details of the comparison with other data rate adjustment algorithms.



(a)



(b)

Figure 3.31: (a) Evolution of throughput and data rate as a function of time. (b) Evolution of causality as a function of time. The letters d , e , and t stand respectively for data rate, errors FER, and throughput, respectively.

Table 3.4: Cumulative amount of data transferred by the algorithms. Average over 20 random simulations. The proposed scheme is referred to FCM, in the case it does not use information related to the SNR, and FCMS, in the case it does.

Algorithm	Avg. [Mb]
AARF	915.70
AARF-CD	902.50
ARF	916.86
CARA	909.80
FCM	923.62
FCMS	915.71

meaning that if the data rate increases, the throughput will likely decrease. Such rule can be seen in Figure 3.31b, represented by the solid line, which around 30 seconds crossed the zero line.

Oscillations can be observed in different parts of the graph: each of these events indicates that the reinforcement mechanism kicked in and, consequently, the reasoning formalism has learned from past experience.

It is worth mentioning that, on average, update messages about throughput and frame error rate are sent approximately every 6.4 and 6.8 seconds, respectively. The amount of updates is relatively small (65 to 70 on average), especially given that the number of data packets exchanged is on the order of 8×10^5 . However, the quantity of exchanged messages can be further lowered, considering that measurements related to almost contemporary events could be grouped together at the level of the sensing entity, as outlined in Section 3.3.3-B. Furthermore, in principle, such data could be piggybacked inside layer 2 acknowledgment frames, thereby reducing the overhead even more.

Interestingly, the version using the information about the SNR of frames performs worse than the other. This can be explained by a wrong encoding of causal relations between the SNR and the other variables. This results into misleading the reasoning process which, occasionally, may provide sub-optimal solutions. This is directly related with the representation of variables by the reasoning formalism. Though FCMs can in principle be based on either continuous or discrete domains, binary FCMs are preferred for convergence properties that continuous maps do not possess [39]. However, problems may arise when mapping inherently continuous variables onto binary domains, as we discussed in a preliminary study of FCM-based cognitive processes [46]. In particular, the mapping operation implies losing quantitative information and the reasoning formalism may not be able to distinguish between optimal and sub-optimal solutions. It is, therefore, necessary to resort to external methods, such as the reinforcement mechanisms described in the previous section.

Finally, the achieved performance increment should be put in relation with the added complexity. The process of finding a proper action to solve a particular situation, or in other words, to find which causes lead to a given effect, is commonly referred to as “abductive reasoning”, which is known to be an NP-hard problem in the case of FCMs and a few other formalisms such as Bayesian and Markov Networks [40]. In the scenario proposed, this does not represent a problem, as the search space consists only of one

‘action’ concept, i.e. it is reduced to the minimum. However, as it will be shown in Section 3.5, it is possible to classify cross-layer interactions (i.e. elements of the adjacency matrix) and identify those that are not relevant to the reasoning process and can be safely eliminated [52], thereby reducing the dimension of the search space.

3.3.5 Conclusion

In this section we have proposed to apply the cognitive scheme that we introduced and described in Section 3.1 to enable finer data rate adaptation in WLAN environments. The proposed scheme is based on a learning process which enables to identify the cause-effect relationships without any a-priori knowledge of the scenario, which makes the approach adaptable to other scenarios and applicable in broader contexts.

The focus of this section was on the implementation details of the architecture. We identified its elements, detailed their functionalities and thoroughly described their implementation model.

Experimental results demonstrate that the proposed scheme provides similar or better results with respect to the most interesting rate adaptation schemes in complex mobility scenarios.

3.4 Dynamic Green Self-Configuration of 3G Base Stations using Fuzzy Cognitive Maps

Cellular networks are rapidly evolving towards the fourth generation, thus providing a global infrastructure for wideband mobile network access. In the current situation, most of the energy consumption of such technology is due to cellular base stations, which are not properly energy efficient. This section addresses the problem of energy efficiency in cellular base stations by taking advantage of the principles of cognitive networking, which promotes the creation of intelligent networks, capable of self-configuring with minimal human intervention. The focus of this text is on the refinements that can be made to the cognitive scheme described in the Section 3.1 rather than on the specific problem. Said refinements concern the pre-processing stage, the learning stage, and the acting stage. Regarding pre-processing we delve into the steps needed to convert a generic measurement into a discrete value apt to be used by the reasoning engine. As for the learning and acting stages, we will introduce new updating schemes and a particular method to select a solution when multiple options are available. The feasibility of the proposed approach is then demonstrated through simulations⁹.

3.4.1 Introduction

Global emissions of greenhouse gases (GHGs) represent the footprint of the development of humanity on the world, especially after industrialization. The ICT sector itself roughly accounts for 2% of today's global carbon footprint [71], but figures are expected to significantly increase in the forthcoming years, with forecasts predicting levels in 2020 around three times as they were in 2002. However, ICT is also forecast to concur, both directly and indirectly, in reducing global emissions of about five times its own footprint, potentially leading to approximately €600 billion savings [72].

The most significant direct effect is that the telecom infrastructure is expected to grow significantly, ultimately being responsible for the 13% of the total sector footprint. Considering also that power generation in ICT is acknowledged to be one of the main causes behind the increase of man-made greenhouse gases, the importance of energy optimization in the telecom infrastructure is evident.

In current mobile cellular networks, base stations are usually kept powered on and operating all day long, pursuing the vision of an “always-on” network. As power consumption in mobile networks is mainly due to base stations, which account for almost 80% of the total [73], it is no wonder that several research efforts have tried to intervene directly on the functioning of transceivers, at different levels of details.

Most of the works in the literature address the problem of energy consumption from a static point of view, by applying static optimization algorithms [73, 74, 75]. While such

⁹The work presented in this chapter was partially done while the author was a visiting research associate at the King's College London (United Kingdom).

This work was partially supported by the Research Project GREENET (PITN-GA2010264759), the Green Radio Core Research Program of the Virtual Centre of Excellence in Mobile & Personal Communications, Mobile VCE, www.mobilevce.com, the ICT-ACROPOLIS Network of Excellence, FP7 project number 257626, www.ict-acropolis.eu, and COST Actions IC0902 and IC0905 “TERRA”.

Part of this work has been submitted for publication to the Elsevier Journal on Computer Networks (special issue on green communications), 2012 [50].

an approach works fine in quasi-static or highly predictable scenarios, the same may not hold in case the environment is subject to dynamic changes or in case of emerging and differentiated usage of the communication infrastructure. As an example, even if it is known that the average number of mobile users in a cell follows a predictable pattern [76], the actual number may be markedly different from the mean, thereby reducing the effectiveness of energy saving mechanisms implementing a static approach. Sharing such considerations, this work is aimed to explore the possibility to develop dynamic energy-reduction schemes.

In this context, the cognitive networking paradigm represents a potential approach to pursue energy efficiency. The term “cognitive network” denotes a communication network in which components (or, at least, a subset of them) intelligently and proactively act towards a specific goal, while continuously learning from past experience. Clearly, the main advantage of such a paradigm is continuous adaptation.

Cognitive networking is a relatively recent research field, stemming from cognitive radio technology. It aims to extend the principles underlying cognitive radios and apply them to the whole communication protocol stack according to a network-wide perspective [13].

Works in this area are traditionally targeted at reducing management complexity or optimizing Quality-of-Service-related metrics. In this section, we propose to adapt cognitive networking principles to address the problem of energy saving in mobile networks. More precisely, we employ Fuzzy Cognitive Maps (FCMs), a mathematical tool that has been introduced and described in depth in Section 3.1. We exploit it to analyze causal relationships between energy consumption in mobile base stations and other variables characterizing a mobile network, in order to identify the most appropriate run-time decisions to reduce energy consumption while maintaining a suitable performance level. It should be noted that a dynamic approach such as the one proposed in this document, though it can, in principle, substitute the “static” schemes, can also be used in conjunction to them.

The rest of the text is structured as follows. In Section 3.4.2 we review the most related works. Section 3.4.3 is devoted to the description of the proposed cognitive architecture, which will be analyzed in detail in Section 3.4.4. Validation of the proposed scheme is done through simulations: the simulation scenario together with the details of simulations are described in Section 3.4.5, while results are presented and discussed in Section 3.4.6. Concluding remarks are offered in Section 3.4.7.

3.4.2 Related Works

As mentioned in the introductory section, several works try to cope with the problem of energy consumption by turning on and off mobile network components—base stations, radio modules, parts of the network itself.

For instance, Tipper *et al.* [74] observe that powering down transceivers may not be convenient in certain scenarios, e.g. for security reasons or because it does not comply with regulatory constraints, and advance the idea to *dim* mobile networks. Specifically, they propose to lower base stations transmission power, diminish the number of frequency slots available, and reduce high data rate services so as to achieve, in order, coverage, frequency, and service dimming.

Other works aim to reduce energy consumption by turning on and off base-station based on the traffic load of a cell [75, 77]. Such approach is based on the observation that the traffic load in real-world mobile networks alternates busy periods with quiet periods in a periodic fashion, to the point that it can be approximated with a sinusoid [76]: during low-peak traffic periods the system is underutilized and energy can be saved by switching off inactive base stations, provided that coverage is guaranteed by neighboring cells. This scheme is shown to potentially reduce energy consumption of about 25 to 30% [77].

Following an even finer degree of control, Saker *et al.* illustrate a scheme to save energy by reducing the number of active transceivers based on the current traffic load [73]. Results show that by implementing sleep modes in a mixed 2G/3G network, it is possible to save up to 66% of the power used in a traditional network, while still being able to retain a blocking rate as low as 0.2%.

This work differs from [75], [77] and [78] in that it focuses on a single base station, rather than multiple base stations in a network. We also avoid putting to sleep a base station in its entirety. Instead, we allow a finer degree of control by activating or deactivating subsets of the transceiving modules that compose the base station. Similarly to [74], we consider powering off radio modules at the higher frequencies if the number of customers is small enough to exclusively fit the lower band (or the other way around, depending on inter-cell interference). However, we assume that radio coverage remains unchanged as we do not allow transmission power to be dimmed. Instead the base station is allowed to switch operational mode, from omni-directional to tri-sectorized and vice versa, depending on the context. In general, differently from all the mentioned approaches, our scheme aims to independently adapt to the context variations, using minimal a-priori information, while also discovering cause-effect relationships among the variables constituting the problem.

Only a few works exist that merge together the cognitive networking paradigm and green communications. One such example is the architecture developed within the End-to-End Efficiency (E³) European project [26]. Although the E³ architecture aims primarily to maximize spectrum and radio resources utilization while reducing configuration complexity, it can in principle optimize the power consumption thanks to long- and short-term decisions taken by different modules (Dynamic Self-Organizing Network Planning and Management module and Self-x for Radio Access Networks module) to be installed in the mobile network.

However, rather than with respect to the E³ architecture, it is more appropriate to position our approach with respect to the reasoning techniques employed in the approaches advanced by the cognitive networking research community. Examples of reasoning formalisms proposed thus far in the literature include neural networks [36], Bayesian and Markov networks [17], and optimization algorithms in general [13].

We use Fuzzy Cognitive Maps (FCMs), a tool that makes reasoning to be based on cause-effect dependencies. FCMs are graphically represented through directed labeled graphs, in which edges are causal relationships that tie together two variables (technically referred to as “concepts”). This is an advantage over neural networks, in which the conformation of edges does not necessarily show the real dependencies between the variables [31]. The advantage over both Bayesian and Markov networks lies in the capability of dealing with causality loops. Whereas, in presence of such loops, in the former

frameworks reasoning can be difficult to perform, the inference process in FCMs works straightforwardly while maintaining a low degree of complexity.

The interested reader can find a more detailed description of FCMs in Section 3.1, and a discussion of its advantages and drawbacks with respect to other reasoning techniques in Section 2.3.1.

3.4.3 The Proposed Energy-efficient Architecture

This section introduces the proposed architecture. For clarity, sub-Section 3.4.3-A describes the main modules of the proposed cognitive architecture, while sub-Section 3.4.3-B explores into more detail the employed energy saving mechanisms.

3.4.3-A The Modules Involved in the Cognitive Loop

Though the approach here proposed can be applied to any generic mobile network, we focus on the Universal Terrestrial Radio Access Network (UTRAN), and use parameters that refer to the High Speed Downlink Packet Access (HSDPA) protocol.

Two are the main components of a UTRAN: the Radio Network Controller (RNC) and the Node B. The RNC is in charge of controlling the Nodes B that are linked to it and managing the available radio resources. Examples of functions carried out by the RNC are handover control, admission control and power control. The Node B, also referred to as base station¹⁰, acts as a transceiver and allows the terminals (User Equipments, technically) to access the core network. It performs low level functions such as signal processing, modulation, and diversity combination.

Before explaining how we propose to equip the UTRAN with cognitive capabilities, let us describe what such capabilities are with the help of the so-called cognitive cycle first introduced in Figure 1.3.

The cognitive cycle is an abstraction that represents the fundamental activities to be carried out by a cognitive entity [3]. The environment is sensed, and the information collected enables the cognitive entity to reason, i.e. to assess the possible actions that can be taken (plan) and finalize a decision (decide). Finally, action takes place, and the environment is again sensed so as to evaluate the effects produced.

Figure 3.32 analyzes how such steps can be embedded in the UTRAN. As Node B acts as an interface between the terminals and the network, it is the appropriate device to perform environmental monitoring. Reasoning and learning, however, are best allocated at the RNC level. The main reason is that a single RNC can potentially drive multiple base stations and can thus aim at global optimization, which would otherwise be remarkably more difficult to achieve. Ultimately, actions can be undertaken only by Node B, properly instructed by the corresponding RNC.

Actions, in the problem posed in this section, deal with energy saving and are analyzed in detail in Section 3.4.3-B, whereas the definition of a reasoning entity suitable for installation on the RNC is described in Section 3.4.4. In particular, Section 3.4.4-A describes how the information collected by sensing modules and the actions of acting modules can be transformed into concepts that can be used in the reasoning formalism.

¹⁰The terms “Node B” and “base station” will be used interchangeably throughout the remainder of the text.

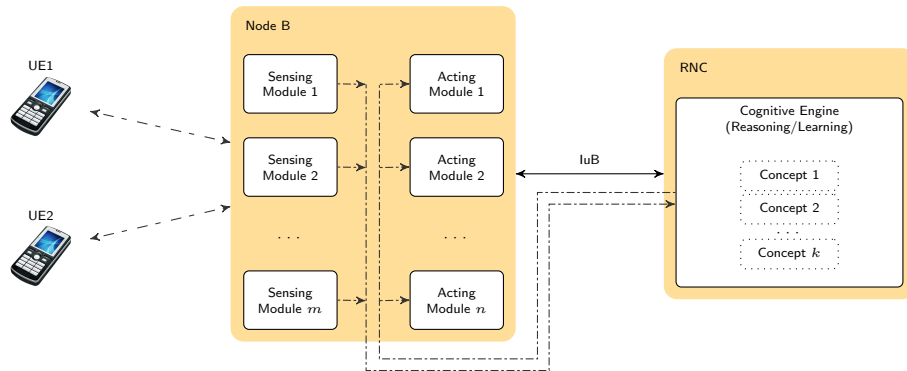


Figure 3.32: How the proposed architecture fits in a UTRAN. Node B monitors the environment and sends the data to the Radio Network Controller (RNC), which upon reasoning and learning, drives the acting modules in Node B, thereby closing the cognitive loop.

3.4.3-B Energy Saving Mechanisms

The energy saving techniques considered in this work are applicable to scenarios in which networks operate in different frequency bands covering the same geographical area. Such scenarios are almost routinely the case in many areas around the world, and their proliferation is likely to increase in the future with new technologies and additional bands, such as the earmarked International Mobile Telecommunications (IMT)-Advanced bands, coming into operation.

In this context, two energy saving techniques are investigated. Both are based on the dynamic redistribution of traffic load or users between bands, but aim at achieving different effects [51]. One technique aims to allow radio network equipment in the bands that the users originated from to be switched off or put into stand-by mode. The other aims to optimize propagation characteristics for the given scenario, dependent on traffic load, shadowing, and other factors. These two techniques are described in more detail in the following sub-sections.

Opportunistic Reallocation to Power Down Radio Network Equipment

Two scenarios are considered: (i) the opportunistic reallocation of all traffic load or users from a band to another band, in order to allow the band that the users or traffic load originated from to be entirely switched off, and (ii) the opportunistic reallocation of a sufficient number of users or traffic load to allow the cell to operate in omni-directional mode instead of sectorized mode, while still adequately carrying all offered traffic. Power consumption dependence on transmission power is limited in most radio base stations, so any increase in required transmission power due to, e.g., a reduction in antenna gain in omni-directional mode, is not likely to have a very significant effect on overall power consumption. Given this, switching off the largest number of radio chains possible, for the sectorization switching example through reducing sectors from, e.g., three (sectorized) to one (omni-directional), is usually considered a reasonable approach to achieve energy saving. Besides, though not investigated in this work, it is inherently clear that entirely opportunistic switching off the cell is also a good solution, should appropriate measures

be taken to avoid any negative consequences of such an action.

This particular solution is clearly suited to locations and times where the deployed networks' capacity is higher than traffic load. In many cases, it is necessary to deploy a considerably large network capacity to cover the peak hour load for example, but at other times all such capacity might not be needed. In such cases, the extra capacity could be switched off in order to save energy. This work considers such capacity being provided across multiple bands, and theorizes the ability to re-allocate users between bands to allow network equipment to be switched off.

Opportunistic Reallocation to Improve Propagation

This solution is based on the ability to opportunistically re-allocate users or traffic loads to lower frequency spectrum to improve propagation, hence reducing necessary transmission power. This, however, will often have the negative effect of increasing inter-cell interference in frequency reuse scenarios, if the density of base stations is high.

Such a possibility can nevertheless be mitigated by considering some specific factors when finding the optimal spectrum available to perform the re-allocation. Such factors are, for instance, the traffic area-density, required base station density and propagation distance, and the frequency-dependent propagation characteristics in the locality. It is noted that in a number of cases the final result may still be the same, i.e., that it is preferable to allocate to the lower frequency spectrum opportunistically, when possible.

This solution is applicable like in the previous case at times when spectral capacity is in excess. Such cases might be operable in non-busy periods or office hours in a business district scenario, or at vacation times.

3.4.4 Embedding Fuzzy Cognitive Maps in Radio Network Controllers

Devised in the '80s as a mathematical tool to help experts discover causal implications in social science problems [19], Fuzzy Cognitive Maps (FCMs) have been applied to different domains, from the simulation of virtual worlds [55] to industrial control systems [39].

As briefly mentioned, they are represented as directed labeled graphs. Nodes are called concepts and can potentially represent any variable that characterizes the system that is being modeled, such as for instance “the amount of users at a given time” or “the consumed energy”. The labeled edges express the strength of the causal relationship binding together two concepts.

In their simplest form, concepts take values in the binary discrete set $\{0, 1\}$. Zero conveys the idea that the concept in question is *inactive* (or *low* or *off*, depending on the concept nature). Conversely, a value of one means that the concept is *active*. It should be noted, though, that other sets, such as $\{-1, 1\}$ and $\{-1, 0, 1\}$, are commonly chosen. Whereas a null value practically absorbs any causal implication between two concepts, the effect of a negative value is to invert causality—which is what actually happens in some situations. As for edge labels, typical values lie in the continuous real interval $[-1; +1]$. The closer to the boundaries the value is, the stronger the causal implication is: positive

or negative, depending on the boundary it approaches, whether the right one or the left one.

The inference process is computationally lightweight and, most importantly, guaranteed to converge in a finite number of steps, provided that concepts are mapped on discrete sets. The process involves repeated multiplications between the vector of all concepts, referred to as the system state and representing the current state of the system, and the adjacency matrix of the FCM studied.

For further details regarding specific aspects of FCMs, such as node and edge domains or the reasoning process, the reader is referred to Section 3.1.2.

We propose to equip base stations with such cognitive capabilities, in order to save energy while adapting to the changing environment.

According to the procedure originally defined in Section 3.1.3, we will define in the following sections a proper FCM for the problem of energy saving in mobile networks.

3.4.4-A Identification of the Concepts Characterizing the Problem

As outlined in Section 3.1.4, the first step towards the definition of a FCM involves the identification of the concepts that will compose the system state.

To increase clarity, let us examine the whole process under a slightly different perspective, using set theory. With reference to Figure 3.33, we can define a set C of all the concepts that characterize the system under study. We can think of such concepts as belonging to different sets:

- Set A comprises all concepts on which the reasoning entity has direct control;
- Set Q collects all concepts that the reasoning entity cannot control directly but that are interesting because they give feedback on the achieved performance;
- Set E collects all concepts on which the reasoning entity has no direct control nor carry relevant information regarding the performance.

As an example, let us consider a generic wireless network. Transmitting stations might enable packet fragmentation to hinder data corruption due to channel noise, which would otherwise reduce the throughput. The key concepts in such scenario are “fragmentation”, “noise”, and “throughput”. According to our framework, fragmentation can be directly controlled and belongs to set A . Throughput and noise cannot be controlled, but while the former is a relevant performance metric and would belong to set Q , the latter is not and would therefore be put in set E .

This toy example clearly shows that sets Q and E together form the complement of A , that is, $Q \cup E = \bar{A}$. It also shows that not necessarily all the variables have to be taken into account in the definition of the system state. Indeed, as can be seen in Figure 3.33, $S \subseteq C$, meaning that some variables may not be considered, depending on the problem formulation. With respect to the toy example we devised, we do not consider the jitter experienced by stations ($\in Q$), as we are not interested in it.

Once concepts are found and classified, it is possible to create the system state vector, $\mathbf{s} = (\mathbf{a}, \mathbf{q}, \mathbf{e})$, where:

$$\mathbf{v} = (v_i, \dots, v_{n_V}) \quad v_i \in V \quad \forall (\mathbf{v}, V) \in \{(\mathbf{a}, A), (\mathbf{q}, Q), (\mathbf{e}, E)\} \quad (3.11)$$

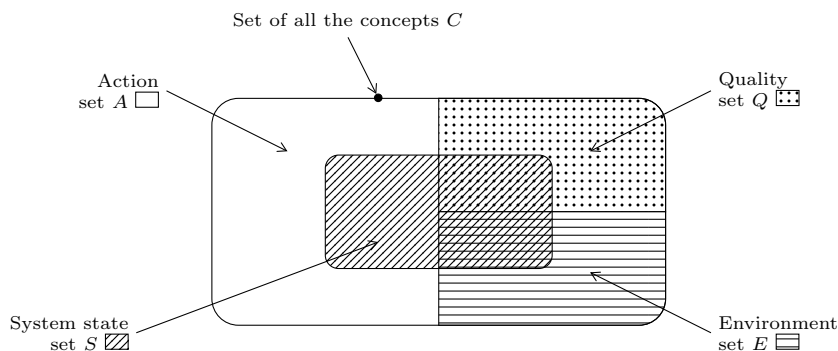


Figure 3.33: Relations among the concept sets and the FCM.

and n_V is the cardinality of generic set V .

The FCM needs to converge to a solution state $\mathbf{s}^* = (\mathbf{a}^*, \mathbf{q}, \mathbf{e})$ by finding a vector \mathbf{a}^* such that the constraints expressed by \mathbf{q} are satisfied before environmental conditions \mathbf{e} change.

For the problem considered in this section, i.e. energy saving in cellular base stations, we selected the following concepts:

- Concepts in $S \cap A$: the use of higher frequencies (*hi*), the use of tri-sectorized operational mode (*tri*);
- Concepts in $S \cap Q$: the energy consumption (*en*), the blocking rate (*br*), the Signal to Interference-plus-Noise Ratio (*snr*);
- Concepts in $S \cap E$: the amount of voice users (*v*), the amount of users browsing the web (*h*), the amount of users that transfer data (*f*).

3.4.4-B Definition of Concept Domains

The second step in defining the FCM involves the identification of concept domains and the pre-processing operations needed to perform the mapping operation.

The number of steps needed to make the reasoning process converge depends on the domains on which the concepts are mapped and the number of concepts themselves [39]. More precisely, the inference process reaches a solution within l^c steps, l being the number of levels of concept domains and c being the number of concepts.

For this reason, it seemed appropriate to adopt only binary sets. Specifically, the set $\{-1, 1\}$ has been used as the domain for all concepts except the blocking rate. In fact, the blocking rate is inherently bounded between 0 and 1. It is reasonable in this case to prefer the use of the discrete set $\{0, 1\}$.

It should be noted that only concepts in $S \cap A$ can be naturally mapped on such interval. Pre-processing operations are needed in order to map all other concepts. The following paragraphs are devoted to explain such operations in detail. When applicable, variable names used in the simulating program are reported in parenthesis, using a **mono-spaced font**. Values used are reported in Table 3.5.

Parameter name	Value
DELTA_PERIOD	180 [s]
GRACE_PERIOD	5 [s]
HYSTERESIS	threshold \pm 5%
MIN_SAMPLES	20

Table 3.5: Parameters utilized for the validation. Unity of measurement appears next to the value, surrounded by square brackets.

Raw data averages

A collection period (`DELTA_PERIOD`) is defined during which raw measurements for each of the variables in $S \cap E$ are acquired. After this period, if the number of samples collected is greater or equal than a target value (`MIN_SAMPLES`) first and second order statistics are computed.

If the number of samples is lower, the sensor is polled at small regular intervals (`GRACE_PERIOD`), until the minimum number of samples is reached. This procedure ensures that a minimum number of samples is collected, so as to reduce the chance of spurious variations.

Discretization

Some variables, such as the blocking rate in this example, are inherently scaled to be properly discretized. However, for other variables, before proceeding with the actual discretization, it is necessary to perform a transformation on the variable, so as to map it onto a restricted domain.

Generally a linear transformation is employed. In such case, given x_i the input value of a generic variable, o_l and o_h the left and the right boundaries of the mapping domain, the output x_o will be:

$$x_o = (x_i - x_{min}) \frac{o_h - o_l}{x_{max} - x_{min}} + o_l \quad (3.12)$$

In Eq. (3.12) x_{min} and x_{max} denote the minimum and maximum value that x_i can reach. As these values may not be known beforehand, if the input value is less than the current minimum or greater than the maximum, the output value will be set to the o_l or o_h and the relative limit updated.

Since variables o_h and o_l in this problem are set to -1 and $+1$, respectively, Eq. (3.12) simplifies to:

$$x_o = \frac{2 \cdot x_i - x_{min} - x_{max}}{x_{min} - x_{max}} \quad (3.13)$$

It should be noted that other transformations can be employed. In particular, if the variable is characterized by a compressed dynamic, logarithmic or exponential transformations may be preferred.

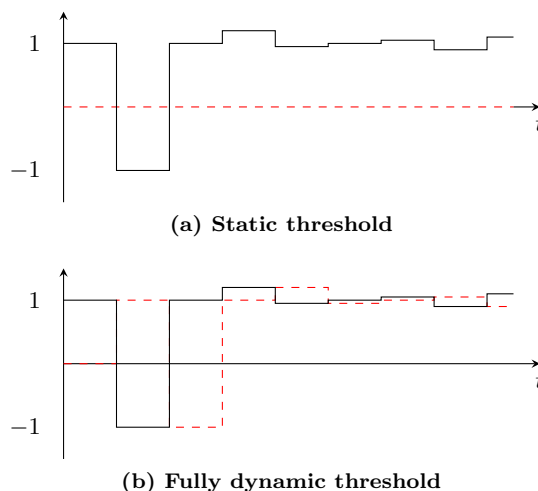


Figure 3.34: Extreme-case time evolutions of variable values and thresholds

Adaptive threshold and hysteresis

The resulting value is ultimately controlled against a threshold, in order to convert the continuous value to either element of the discrete set specified at design time.

Choosing a static threshold may not be an optimal solution under some circumstances: if the variable stabilizes around a specific value which is not close to the chosen threshold but keeps varying, no variation will ever be registered and the system could possibly settle to a sub-optimal point. This situation is depicted in Figure 3.34a.

The implementation of an adaptive threshold reduces the likelihood of such an event. In the extreme case, such a threshold is set to the immediately older value recorded, as shown in Figure 3.34b. The drawback in this case is that even small variations around the threshold are perceived as radical changes.

To counteract this side-effect it is straightforward to think of solutions in between. An example may be computing the threshold as a moving weighted average between the actual value of the variable and the previous threshold.

For this particular problem, though, we have resorted to a different solution, by introducing a hysteresis (**HYSTERESIS**), proportional to the extension of the mapping domain.

3.4.4-C Definition and Update of the Fuzzy Cognitive Map

The third and last step is about embedding of any a-priori knowledge of the problem to the FCM. Let us denote by f_{ij} the edge of the FCM that departs from i and arrives to j , i , and j being generic concepts in S .

1. We assume that concepts in the same set are *causally* independent from one another. Considering the set $S \cap Q$ as an example, this means that, for instance, the variation of the number of users that browse the web has no causal implication to (and from) the variation of the number of the users that place voice calls. This means that no edges arrive or depart from concepts that belong to the same class:

$$f_{ij} = f_{ji} = 0 \quad \forall i, j \in \{S \cap V\}, V \in \{A, Q, E\} \quad (3.14)$$

2. Concepts in the action set are not directly caused by any other concept. Instead they are triggered by the reasoning process. This translates into the fact that no edges point to any action concept, that is:

$$f_{ia} = 0 \quad \forall a \in \{S \cap A\}, \forall i \in \{S \cap V\}, V \in \{Q, E\} \quad (3.15)$$

3. Similarly, we state that concepts related to quality metrics do not cause any variation in the concepts related to the environment. As an example, users will decide to call, browse the web and download files ignoring channel conditions (Signal to Interference and Noise Ratio (SINR), blocking rate and energy consumed by the base station). Mathematically:

$$f_{qe} = 0 \quad \forall q \in \{S \cap Q\}, \forall e \in \{S \cap E\} \quad (3.16)$$

4. We also know that both actions increase the number of frequency slots available, and, as a direct consequence, reduce the blocking rate. Therefore we may want to embed such information, by properly setting $f_{hi,br}$ and $f_{tri,br}$.

The resulting FCM is as follows:

$$F = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & f_{v,en} & f_{v,br} & f_{v,snr} \\ 0 & 0 & 0 & 0 & 0 & f_{h,en} & f_{h,br} & f_{h,snr} \\ 0 & 0 & 0 & 0 & 0 & f_{f,en} & f_{f,br} & f_{f,snr} \\ f_{hi,v} & f_{hi,h} & f_{hi,f} & 0 & 0 & f_{hi,en} & f_{hi,br} & f_{hi,snr} \\ f_{tri,v} & f_{tri,h} & f_{tri,f} & 0 & 0 & f_{tri,en} & f_{tri,br} & f_{tri,snr} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (3.17)$$

In order to keep the FCM updated a learning algorithm has to be employed. Learning algorithms emulate human learning: just like human beings infer causality between two events when they perceive concomitant variations that respect a chronological order, learning rules for FCMs modify the labels of the edges that connect two concepts that experienced some change in a relatively short period of time.

If a concept experiences a positive variation and soon after another concept experiences a positive variation as well, it can be assumed that the two concepts are bound by a positive causal relationship. The same holds if both the variations experienced are negative. Conversely, if they undergo alternate variations (positive the first and negative the second, or vice versa), it can be inferred that there is a negative causal relationship that connects the two concepts.

A popular learning rule, that we discussed in Section 3.1.3-B, is known as Differential Hebbian Learning (DHL), and updates the edges in an FCM proportionally to the value of variations of the concepts [42]. The rule was first written in Eq.(3.3), but it is reported here for convenience. If we indicate by f_{ij}^t a generic edge between concepts C_i and C_j at time t and by \dot{C}_i the variation of concept i at time t , the DHL rule states that:

$$f_{ij}^t = f_{ij}^{t-1} + \eta \left(-f_{ij}^{t-1} + \dot{C}_i^t \dot{C}_j^t \right)$$

The parameter $\eta \in (0; 1]$ is known as “learning rate” [55] and its purpose is to lower the responsiveness of the algorithm, which could otherwise produce too abrupt updates.

A peculiar property of DHL is that it does not account much for the *causal history* of an edge. Even if many variations in a row take place, all sharing the same polarity, it takes only a few steps for the edge to change value (and sign). Another feature of DHL regards the importance of consecutive variations of the same type. As can be seen in Figure 3.35a, variations are assigned different levels of importance depending on how many variations of the same type happened immediately before.

In an effort to address these aspects, we devised two novel rules, that we called Exponential Backoff Learning (EBL) and Linear Learning (LL).

Exponential Backoff Learning

Similarly to DHL, this rule assigns smaller values to consecutive variations sharing the same polarity. However, when opposite variations happen, the algorithm will set the edge to the values that were previously been used, retracing all the steps backwards. This way, if the reasoning engine has witnessed n variations of the same type, after the edge in question crossed the zero, it will take exactly n variations of the other type for the edge to reach the zero value again. The behavior of such a rule is shown in Figure 3.35b. Mathematically, it is defined by the following equations (in which sgn denotes the sign operator):

$$f_{ij}^t = f_{ij}^{t-1} + \alpha^t \quad (3.18a)$$

If $\text{sgn}(\dot{C}_i^{t-1}\dot{C}_j^{t-1}) \neq \text{sgn}(\dot{C}_i^t\dot{C}_j^t)$:

$$\alpha^t = -\alpha^{t-1} \quad (3.18b)$$

If $\text{sgn}(\dot{C}_i^{t-1}\dot{C}_j^{t-1}) = \text{sgn}(\dot{C}_i^t\dot{C}_j^t)$:

$$\alpha^t = \alpha^{t-1} \cdot \begin{cases} \eta & \text{if } \text{sgn}(\dot{C}_i^t\dot{C}_j^t) > 0 \text{ and } f_{ij}^{t-1} > 0 \text{ or} \\ & \text{sgn}(\dot{C}_i^t\dot{C}_j^t) < 0 \text{ and } f_{ij}^{t-1} < 0 \\ \frac{1}{\eta} & \text{if } \text{sgn}(\dot{C}_i^t\dot{C}_j^t) > 0 \text{ and } f_{ij}^{t-1} < 0 \text{ or} \\ & \text{sgn}(\dot{C}_i^t\dot{C}_j^t) < 0 \text{ and } f_{ij}^{t-1} > 0 \\ 1 & \text{otherwise} \end{cases} \quad (3.18c)$$

Linear Learning

There might be situations in which all variations should be treated the same way, i.e. all considered of the same importance. To reach this objective we created the Linear Learning (LL) rule, by slightly modifying the DHL, and eliminating any reference to the value taken by the edge at the previous time step.

Accordingly, edges are updated based only on the polarity of the variation, that is:

$$f_{ij}^t = f_{ij}^{t-1} + \eta \cdot \text{sgn}(\dot{C}_i^t\dot{C}_j^t) \quad (3.19)$$

where, again, sgn denotes the sign operator. Clipping is done to prevent edge values from falling off of the $[-1; 1]$ interval, as also shown in Figure 3.35c.

3.4.4-D Action Profile Selection

It is possible that the reasoning formalism identifies multiple solutions for a given problem, or, in other words, it may find that different action profiles can be used in a specific situation.

However, because of the intrinsic discrete nature of concepts, evaluating which solution should increase the performance the most is not immediate. To address this aspect, a method based on the assignment of a performance score has been developed.

Without loss of generality, let us consider an FCM that deals with binary concepts, as the one described in this section. Upon interrogation, such FCM yields a binary system state (one for solution), telling which quality-related concepts can be improved (and which not), by using which actions (and which not).

The idea is to assign each action profile \mathbf{a}^* a score σ that depends not only on the state of the concepts a given action profile is able to improve (q^*), but also on their current state (q_c):

$$\sigma(\mathbf{a}^*) = \sum_{q^* \in s^*(\mathbf{a}^*)} \sigma_{q^*}(q^*, q_c) \quad (3.20)$$

in which σ_{q^*} , which is the score associated to a quality concept q , is defined as follows:

$$\sigma_{q^*}(q^*, q_c) = \begin{cases} 0 & \text{if } q^* = \nabla \\ \lambda \in \mathbb{R} & \text{if } (q^*, q_c) = (\Delta, \nabla) \\ \lambda \cdot e^{-\Delta_t/\mu} \quad \lambda, \mu \in \mathbb{R} & \text{if } (q^*, q_c) = (\Delta, \Delta) \end{cases} \quad (3.21)$$

in which symbols ∇ and Δ denote bad and good performance levels, respectively, and Δ_t is the time elapsed since the concept has registered a good performance level.

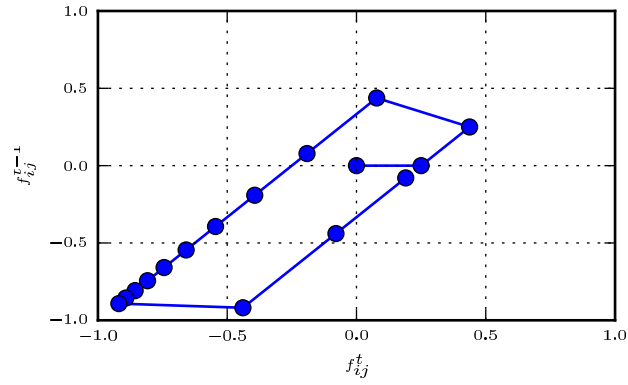
Eq. (3.21) indicates that if the state of the quality-related concept:

- cannot be improved, no score is assigned;
- can be improved, and it is currently experiencing a bad situation, the maximum score is given;
- can be improved, but it is already in a good state, a variable score is given, depending on how much time has elapsed since it entered such state.

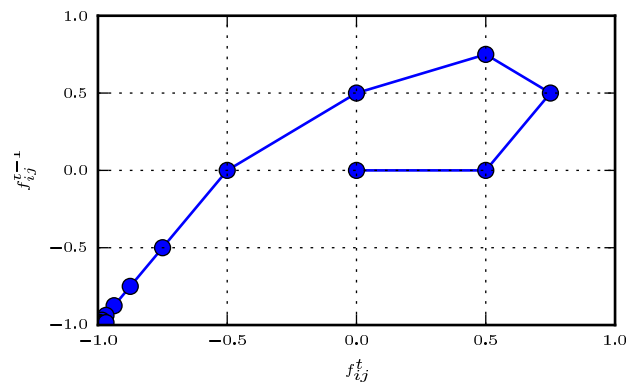
Once scores are computed, the action profile corresponding to the greatest value is selected.

To better understand how the scoring system works, let us refer to the example shown in Figure 3.36, in which two quality concepts that we introduced in Section 3.4.4-A are depicted, namely, *en* and *snr*.

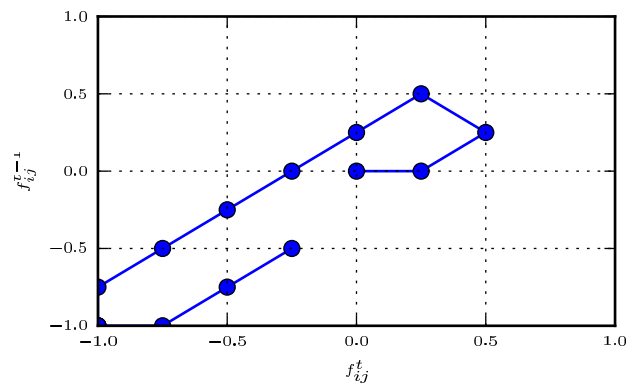
The first thing to notice is that the meaning of good/bad situations is different, depending on the concept. Whereas for *snr*, high values of the concept represent a good situation (Δ), for *en* is the opposite, i.e. high values means the base station consumes too much (∇).



(a) Differential Hebbian Learning (DHL)



(b) Exponential Backoff Learning (EBL) - $\eta = 0.5$



(c) Linear Learning (LL)

Figure 3.35: Lagged-coordinated plot of the evolution of an FCM edge when updated by different learning algorithms. Starting point is $(0, 0)$. The sequence of variations is as follows: 2 positive, 10 negative, and 3 positive. Learning rate η set to 0.25, unless otherwise stated.

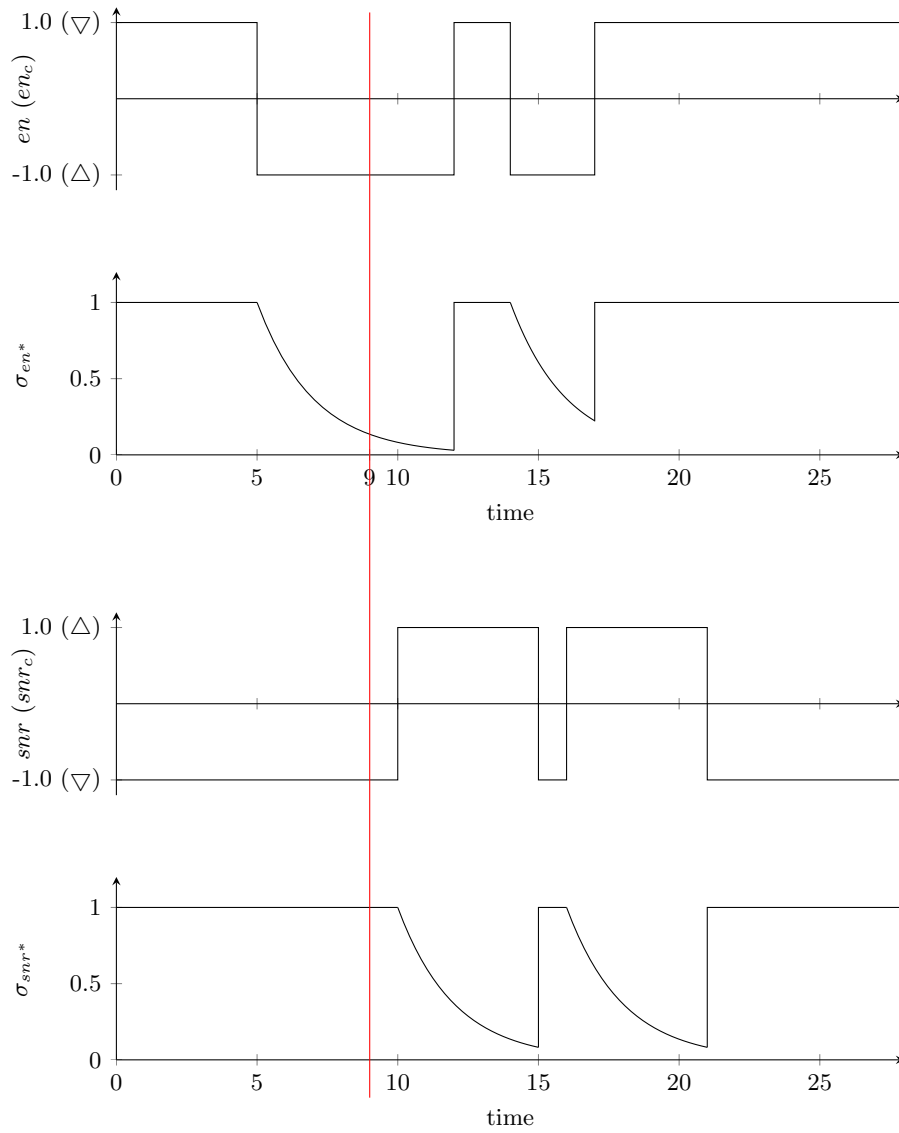


Figure 3.36: Differentiation of multiple solutions through the assignment of a performance score ($\lambda = 1, \mu = 2$). At $t = 9$, solutions that could improve en and snr , en but not snr , snr but not en , neither en nor snr , would score 1.13, 0.13, 1, and 0, respectively.

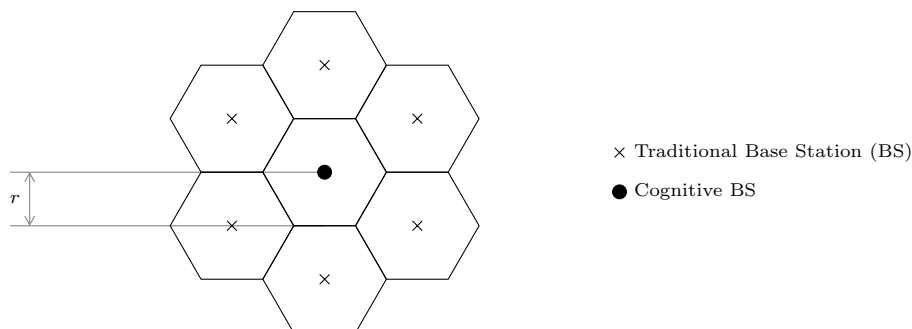


Figure 3.37: Layout of the simulated scenario. The coverage radius r is set to 600 m in the simulations.

Next, suppose that at 9 seconds, the reasoning formalism is queried and the action profile proposed as solution is able to lead both concepts in a good state. Let us use $\lambda = 1$ and $\mu = 2$ for the computation. Since the concept associated to SINR is not in a good state ($snr_c = \nabla$), eq. (3.21) tells us that the score assigned to it is 1. On the other hand, the concept associated to the energy is in a good state since $t = 5$. Therefore, $\Delta_t = 4$, and the score assigned to it is $\exp(-4/2) \sim 0.13$. Overall, the action profile scores 1.13.

3.4.5 Simulation Scenario

The simulation scenario is populated by seven base stations arranged according to the classical honeycomb structure, as shown in Figure 3.37. Measurements are taken from the central base station, for symmetry reasons.

The cell in the center is served by a base station that is equipped with cognitive capabilities. Such cognitive capabilities are based on the FCM designed in Section 3.4.4, and allow the base station to reason about the environment to reduce energy consumption while monitoring the blocking rate. All other base stations in the network do not employ any cognitive scheme and maintain all radio modules enabled at all times.

The simulating platform focuses on the periods of active communications between the base stations and the user terminals associated to it, so that it is possible to monitor energy consumption.

Terminals are static during their communications and are distributed over the coverage area of a base station following a uniform random distribution. Associations and de-associations to/from a base station follow a Poisson process, with parameters λ and μ , in order. Both λ and μ depend on the type of communication carried on: voice call, web browsing, data transfer. Therefore, we define the two parameters for each traffic category, resulting in six parameters: $\lambda_v, \mu_v, \lambda_h, \mu_h$, and λ_f, μ_f .

Each base station has a peak busy load of 50 users, weighted by the data shown in Figure 3.38 in order to reflect real-world situations. Weights reflect the actual hourly load measured in a Vodafone 3G cell in London and were obtained via internal communication within the Mobile VCE Core 5 Green Radio research program.

Voice traffic is modeled after the well-known Brady six-state model [79]. In our case, to compute energy consumption, we restricted our attention to a subset of the six states

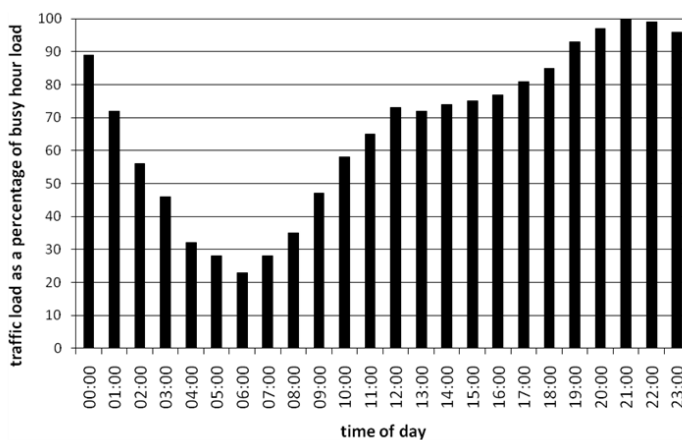


Figure 3.38: Hourly variation of traffic load as a percentage of busy hour load over a typical day for a mobile network operator in London, UK.

of the original model, focusing only on the states for which there is a transmission from the base station to the terminal, i.e. when the other end of the communication is active. We assumed an average call duration of one minute, resulting in a μ_v of $1/60$.

Web traffic has been modeled as a continuous repetition of two states: a downloading period to retrieve a page from the web, and a waiting period, to parse and read the page [1]. The download time depends on the size of the web page and potential embedded objects. Object sizes follow a truncated lognormal distribution, while the number of the embedded objects in a web page follows a truncated Pareto distribution. Reading and parsing times can be found by sampling an exponential distribution. Assuming that a web session for a mobile user lasts, on average, five minutes, we fixed μ_w to a value of $1/300$.

The model for File Transfer Protocol (FTP) sessions is similar to that for web sessions, except for the fact that there is no parsing time [1]. The download time exclusively depends on the size of the object to be transferred, which follows a Pareto distribution. Reading time is, again, modeled by an exponential random variable. Considering that a mobile user is not likely to make extensive use of FTP, we hypothesize an average session duration of two minutes, equivalent to a μ_f of $1/120$.

In the simulations we fix the composition of traffic as follows: 50% voice traffic (α_v), 40% web browsing (α_h) and 10% FTP traffic (α_f). Traffic categories are independent from one another. By approximating each type of traffic as an $M/M/\infty$ queue it is possible to find the relative birth rates as:

$$\lambda_i = N\alpha_i\mu_i \quad i \in v, h, f \quad (3.22)$$

The environment broadly reflects that of an HSDPA network. The main parameters that characterize the system are reported in Table 3.6.

Base stations all operate on two bands, centered at 2GHz and 5GHz, and are characterized by a coverage radius of 600 m. The from-the-socket power P_M is computed

Parameter	Value
System configuration	Broadly reflecting HSDPA Rel. 5
Spectral efficiency	0.8 b/s/Hz
Bandwidth per HSDPA band	5 MHz
Channel path loss models [80]	2 GHz: $128.1 + 37.6 \cdot \log(d)$ 5 GHz: $141.52 + 28 \cdot \log(d)$
HSDPA pilot power	20% of cell power budget
Average voice session duration ($1/\mu_v$)	60 s
Web page size distribution	Truncated lognormal
Web page average size (min./max.)	10710 B (100 B/2 MB)
Embedded objects size distribution	Truncated lognormal
Embedded objects average size (min./max.)	7758 B (50 B/2 MB)
Number of embedded objects distribution	Pareto, $k = 2$ and $\alpha = 1.1$
Avg. number of embedded objects per page (min./max.)	5.65 (0/53)
Web traffic reading time (OFF duration)	Exponential, mean 30 s
Web traffic parsing time (OFF duration)	Exponential, mean 0.13 s
Average web session duration ($1/\mu_w$)	300 s
FTP file size distribution	Pareto, k computed from the mean and α , and $\alpha = 1.5$
FTP file average size	2 MB
FTP file reading time (OFF duration)	Exponential, mean 180 s
Average ftp session duration ($1/\mu_f$)	120 s

Table 3.6: Simulation configuration parameters. d is the distance in km. Traffic modeling parameters obtained from [1].

according to a well known linear function of the transmission power P_{tx} [81], namely:

$$P_M = a \cdot P_{tx} + b \quad (3.23)$$

According to internal documentation within the Mobile VCE Green Radio research program, it has been shown that an HSDPA base stations consumes 857 W at 100% transmission power and 561 W at 20% transmission power. Constants a and b have been computed by regression and are equal to 9.25 from-the-socket Watts per transmission Watt and 487 Watts.

Regarding user capacity, a base station can accommodate at most 22 users per band when operating in omni-directional mode and up to 15 users per band per sector when operating in tri-sectorized mode.

3.4.6 Results

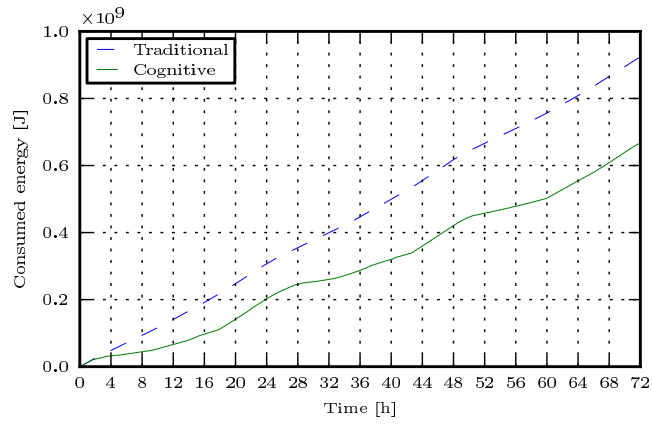
Simulation time covers three days, starting from midnight. As we intend to simulate three consecutive working days, the hourly variation reported in Figure 3.38 does not change from day to day.

The dashed line in Figure 3.39a represents the energy consumption by a traditional base station, i.e. when all six modules are always on. As can be expected, the curve is a linear function of the time, showing no change in the behavior of the base station. The behavior of the cognitive base station is represented by the solid line, which resembles a piecewise linear function. The curve reveals that at times when the user load is low, it is possible to save energy by switching off part of the radio modules. Conversely, when there is a high user load all radio modules must be kept active and no energy saving is possible. Such periods are the intervals in which the solid curve runs in parallel with the dashed curve, i.e. from hour 18 to 28, from hour 42 to 50, and from hour 60 to 72.

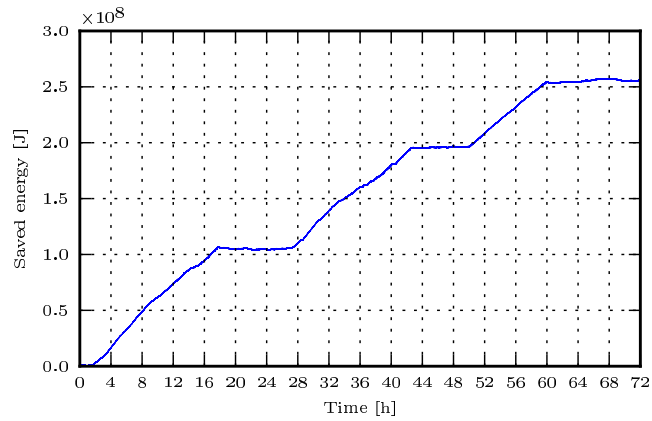
Figure 3.39b shows the difference between the energy consumed by a traditional base station and the cognitive base station. The periods in which energy saving is not possible can be recognized more clearly. By analyzing the slopes it can be observed that our scheme allows to save about 6.25 MJ/h or 1.7 kW during such periods, equivalent to 50% with respect to the traditional case. Considering also the periods in which all radio modules are active, energy saving decrease to 3.5 MJ/h or almost 1 kW, i.e. about 25% of the energy consumed in the traditional case.

Inevitably, there is a trade-off between energy saving and blocking rate. As the thin line in Figure 3.39c indicates, the blocking rate tends to increase when a subset of the transceivers is turned off. High peaks of the blocking rate can be registered by averaging the value using a window equal to the collection period (180 s). However, it should be noted that it may be that only a few users try and join the network during such intervals, hence the difference between local maxima and following minima is often pronounced. The thick curve represents the all-time average, in which the blocking rate remains below 5% throughout the simulation.

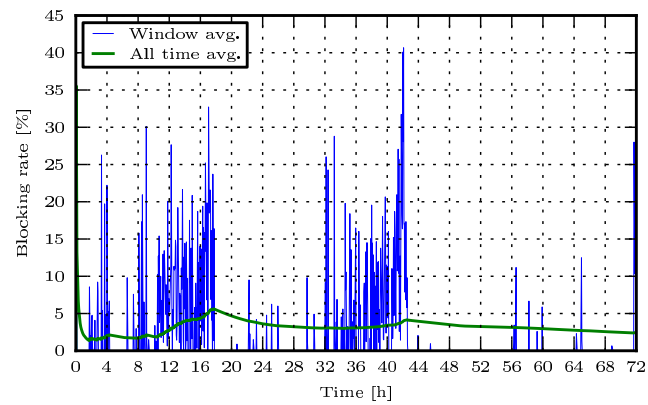
Figure 3.40 reproduces the behavior of the radio modules in the cognitive base station. In such graphs, the step curves are either high or low, symbolizing the use or not, in order, of the extra capacity. Figure 3.40a shows the use of the higher band, whereas Figure 3.40b



(a)



(b)



(c)

Figure 3.39: (a) Consumed energy in traditional and cognitive base stations, and (b) energy saved by employing the cognitive scheme over a period of 72 hours. (c): example of the evolution of the blocking rate.

shows the use of the tri-sectorized mode. The oscillations during the low-activity periods hint at the fact that an even greater saving could be, in principle, possible.

For the sake of clarity, we also report the causal relationships between the action concepts and the quality-related concepts (Figure 3.41). It can be noticed that the cause-effect relations between any action concept and the blocking rate are negative and approach the lower bound (-1). This means that using the higher band in conjunction with the lower band and using the tri-sectorized mode causes the blocking rate to decrease. A similar effect happens with energy: turning on the radio modules causes the consumption of energy to rise (positive cause-effect relation). The relation with the SINR is less clear. Whereas the employment of tri-sectorized mode seems to increase the SINR, when turning on the higher band it an increase in the SINR may not follow. This could be explained by the fact that, although it is true that switching on more radio modules increases the available capacity and should lower the interference, it is likewise true that interference depends also on the activity in the neighboring cells, which is neither modeled by the designed FCM, nor it can be controlled.

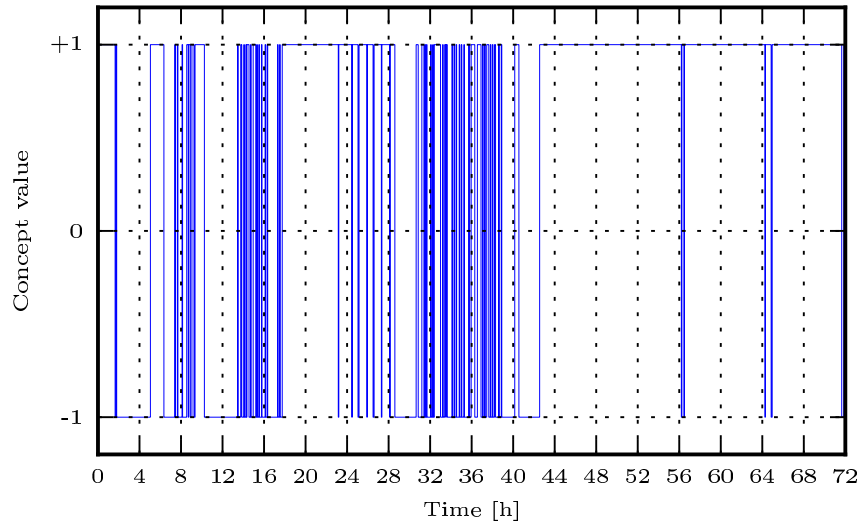
3.4.7 Conclusion

As the ICT sector is partly responsible with the increase in the global carbon footprint, methods are being developed to reduce the amount of energy consumed in the area of telecommunications.

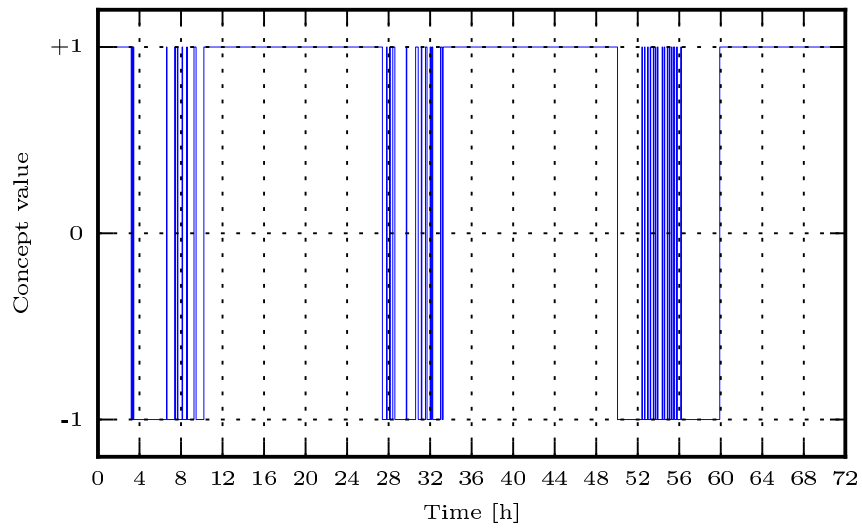
We have described how this problem can be tackled by exploiting the cognitive networking vision, according to which networks should be able to self-configure in view of a specific goal, limiting the need of human intervention. Specifically, we have proposed a novel dynamic scheme to perform energy saving in HSDPA base station by driving the selection of the active transceivers based on Fuzzy Cognitive Maps.

We have introduced several refinements to the system, i.e. adaptive thresholds to convert continuous measurements into discrete concepts, novel learning algorithms to account more for the causal history of concepts, and a scoring mechanism to choose the most promising solution when multiple options are available.

Results demonstrate that the cognitive paradigm enables to save a relevant amount of energy in operation under realistic traffic patterns. As a consequence, the proposed architecture seems suitable to support dynamic configuration of cellular base stations. Clearly, the achieved savings correspond to an increase in the blocking probability. The identification of a refined methodology to enable a more flexible trade-off between energy saving and blocking rate will be subject of further investigation on the topic, as well as a generalization of the proposed architecture to enable collaboration among several neighboring base stations.

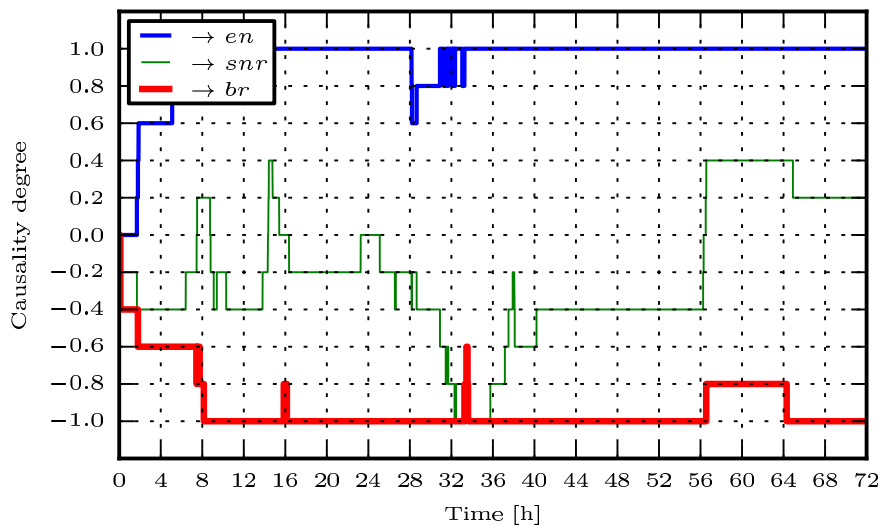


(a)

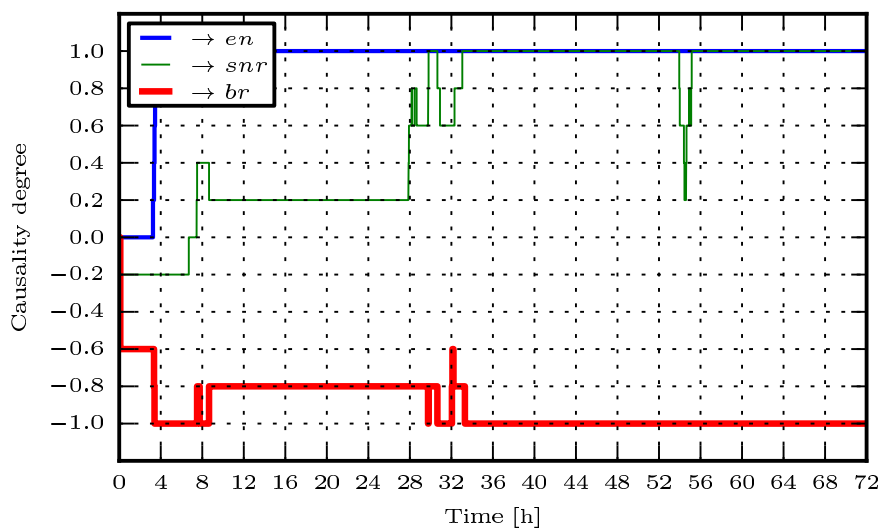


(b)

Figure 3.40: Evolution of the action concepts: (a) use of higher frequencies (*hi*) and (b) use of tri-sectorized mode (*tri*)



(a)



(b)

Figure 3.41: Example of evolution of the causal relationships: (a) between hi (use of higher frequencies) and the quality-related concepts, and (b) between tri (use of tri-sectorized mode) and the quality-related concepts.

3.5 Dimensionality Reduction in Fuzzy Cognitive Maps

Cognitive networks were recently proposed to cope with the complexity and the dynamics of network managements, exploiting reasoning to adapt the behavior of protocols. As we have seen in the previous sections, among the reasoning formalisms that can be employed in cognitive networks, Fuzzy Cognitive Maps (FCMs) seem to be very promising, as they potentially allow the cognitive process to consider cross-layer interactions in the characterization of the performance of a network node. However, when considering a high number of cross-layer interactions, reasoning schemes can be too time consuming and may not provide a suitable solution as environmental conditions change. In order to decrease the demand of reasoning time it is of utmost importance to discover which cross-layer relationships carry relevant information to the cognitive process. This section discusses how to make such distinction. Moreover, it proposes a metric to evaluate the influence that a cross-layer interaction has on the cognitive process¹¹.

3.5.1 Introduction

Recent advances in communications led to significant improvements in network performance at the price of increased management complexity. In order to cope with such complexity, new networking paradigms have been proposed. In this scenario, “cognitive networking” attempts to embed networks with intelligence to facilitate monitoring, reasoning, and acting towards the achievement of performance goals [13].

Reasoning is undoubtedly the most critical step of the cognitive process. Indeed it needs to balance the potential lack of precision of the information obtained from sensing activities, and determines the actions to be taken in the acting step. As a consequence, reasoning has received significant attention in the literature. Although there is a general consensus that reasoning should exploit information from all layers of the protocol stack of a network node [18, 20, 82], it seems there is no general agreement on the reasoning technique to be employed for that. Neural networks, Bayesian networks, expert systems, multidimensional optimization algorithms, and techniques from the fields of control theory and pattern recognition have been employed for this goal [13, 18]. Moreover, quite often, the techniques are chosen empirically, without proper justification [3]. In addition to that, although it has been acknowledged that reasoning schemes must converge to a solution before the environment significantly changes [13], and that different events might require different time-sensitive responses [12], to the best of the author’s knowledge, apparently no work focuses on reasoning time.

Since this work focuses on cognitive processes based on FCMs, which exploit the information on cross-layer interactions for reasoning, the procedures introduced here are targeted to identify the interactions that are most relevant for the cognitive process. In particular, relevant cross-layer relations can be identified by their causal strength and variability; weak relations and variable causality can be ignored, reducing reasoning time

¹¹The work presented in this section was partially done while the author was a visiting Ph.D. student at the State University of Campinas (SP, Brazil) in the framework of the EUBRANEX Erasmus Mundus External Cooperation Window (EM ECW) EU programme.

Part of this work was published in the proceedings of the IEEE GLOBECOM ’10 international conference, Miami, Florida, USA, 2010 [52].

without affecting the effectiveness of the cognitive process. This part of the thesis introduces novel methods to classify cross-layer interactions and identify those relevant to reasoning in cognitive processes based on FCMs. Such identification is a fundamental step towards reducing reasoning time and consequently improving the performance of cognitive networks.

The remainder of this text is organized as follows. Section 3.5.2 motivates the choice of FCMs and describes the peculiar characteristics. Section 3.5.3 delves into the importance of distinguishing relevant from irrelevant cross-layer interactions in FCM-based reasoning. Sections 3.5.4 and 3.5.5 propose a procedure to operate such differentiation and provide an example. Finally, Section 3.5.6 draws the conclusions.

3.5.2 Motivation

Despite the existing consensus in the literature about the need for the holistic consideration of network node operation, examples provided so far seem to pay little attention to the issue. Indeed, some cognitive network architecture proposals stress the importance of involving all layers of the protocol stack to observe requirements and constraints. However, they do not offer implementations of the target scheme [18, 20] or they propose implementations on a limited subset of the available layers [82]. Although it is deemed that cognitive network architectures will make intense use of cross-layering [13] and that cross-layer interactions impact the performance of network nodes [83], so far no attempt has been made to account for the impact of cross-layer interactions on the performance of network nodes [18]. Even more important, though it is natural to make use of cross-layering in cognitive networks [84, 30], to the best of the author's knowledge, no research work has *explicitly* taken into consideration cross-layer dependencies in the reasoning process, which is fundamental to fully enable a truly holistic approach. Only recently, a reasoning mechanism based on FCMs, capable of reasoning on cross-layer interactions has been proposed [46].

Introduced in the '80s [19] as a tool for causal reasoning, FCMs can be used to model dynamic systems, since they emphasize the cause-effect relationships among the systems' internal variables. Mathematically, FCMs can be represented by directed labeled graphs (and, thus, by adjacency matrices).

To better understand the whole section, let us briefly review the most prominent characteristics of an FCM.

Nodes in an FCM symbolize causal objects, i.e. general concepts that can entail and/or be entailed by other concepts. Edges symbolize causal relationships between two concepts. Edge labels express the degree of such causality and can take any value in an interval of real numbers. The domains of the labels are usually zero-centered (often between -1 and $+1$), so that causality can be positive, negative or non-existent (in which case there is no edge). Since the graph is directed, the direction of an edge discriminates the cause (where the edge starts) from the effect (where the edge ends).

Reasoning is based on the edges of an FCM. More importantly, the amount of edges (i.e. the number of causal relationships) directly impacts on the time needed to find a solution: the more the edges are, the slower the reasoning process converges to a solution.

The focus of this section is, therefore, on the identification of weak, irrelevant causal

relationships, that do not add any valuable information to the reasoning process and, if eliminated, can reduce the time needed to reach a solution.

Though this schematic description is sufficient to understand the ideas presented in the remainder of the text, the interested reader is referred to Section 3.1.2 for a more detailed explanation.

3.5.3 Fuzzy Cognitive Maps in Cognitive Processes

The capability of FCMs of representing systems through the relationships of their internal mechanisms matches adequately with the representation of cross-layer interactions in nodes of cognitive networks.

A detailed description of the application of FCMs to cognitive networks was introduced in Section 3.1.3. In summary, it was shown that the state of a node (technically referred to as the system state) is composed of three vectors of concepts: (i) the vector \mathbf{q} , related to quality requirements specific to the communications which the node is involved with, (ii) the vector \mathbf{e} , comprising environment-related concepts, and (iii) \mathbf{a} , the vector of the actions a node can take. The fundamental problem is, thus, to find a particular value of \mathbf{a} , \mathbf{a}^* , so that the constraints given by \mathbf{q} are met, before the environmental conditions specified by \mathbf{e} change.

Solving this kind of problem implies the use of abductive reasoning, and we have to resort to exhaustive search to find an exact solution. The problem is somewhat mitigated by the fact that the search space can be reduced. Since only \mathbf{a} can be modified by a node, and assuming, without loss of generality, that the node domain is a binary set such as $\{-1, 1\}$ or $\{0, 1\}$, the search space has $2^{\dim(\mathbf{a})}$ elements (i.e. $2^{\dim(\mathbf{a})}$ combinations to try) which makes exhaustive search faster.

For each combination, the inference procedure converges (either to a fixed point or to a limit cycle) within l_c steps, where $l \geq 2$ is the cardinality of the discrete domain onto which concepts are mapped, and c is the number of concepts [39]. Clearly, it is beneficial to keep both l and c as small as possible.

As it is often the case, variables are naturally mapped to the smallest possible domain (the binary domain $\{0, 1\}$), which cannot be further reduced. As a consequence, though it can be difficult once the FCM has been designed, it is important to lower as much as possible the number of concepts c needed for the computation, taking care to avoid impinging the performance of the reasoning process.

A concept cannot be removed from the computation without first confirming that all the edges linking it to other concepts do not impact the cognitive process. The problem, then, is to find a metric to identify which edges can be discarded, i.e. which cross-layer interactions are not relevant for the reasoning process.

3.5.4 Identification of non-Relevant Cross-layer Relations

This sub-section discusses the relevance of cross-layer relations for the reasoning process and how to identify those which are relevant. First, two standalone approaches are presented. An exhaustive search approach is described in 3.5.4-A, while a lightweight, yet

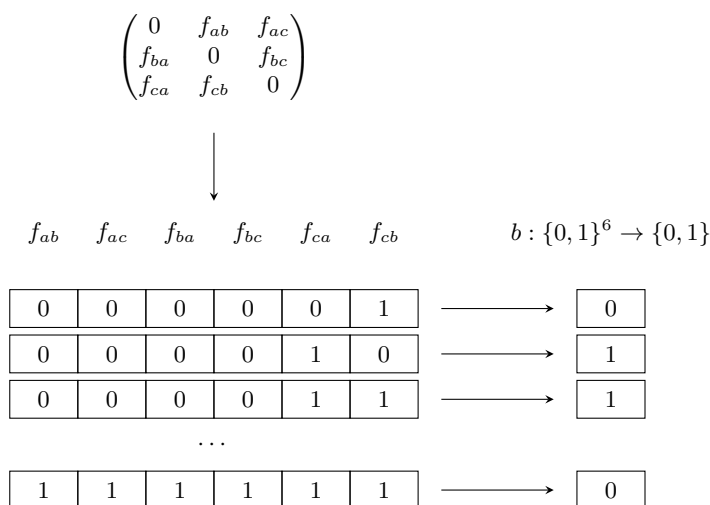


Figure 3.42: Toy example showing the first steps in the application of the Logic Circuit Minimization approach with $c = 3$ and $n = 6$. First, the elements of the FCM are serialized. Then, reasoning is performed considering different combinations of the FCM elements and assigned a binary value indicating the performance level obtained. For instance, considering f_{ca} alone, or f_{ca} and f_{cb} (second and third combination) allowed the reasoning process to achieve an acceptable performance level, whereas considering f_{cb} only did not.

also approximate, approach is described in 3.5.4-B. Finally, a joint approach combining the advantages of the two methods presented is described in 3.5.4-C.

3.5.4-A An Exhaustive Approach: Logic Circuit Minimization

Let us suppose a system state has c distinct concepts. Thus, the matrix of the associated FCM is a c -by- c matrix. Without loss of generality, let us also suppose that the FCM is a matrix with n non-zero elements, $n \leq c^2 - c$ (the elements on the diagonal are set to zero by definition). Then, reasoning will be based *at most* on n elements. However, not all the elements may be relevant to the reasoning process.

The possible combinations of the n elements are $2^n - 1$, excluding the trivial combination $(0, 0, \dots, 0)$. We can think of each combination as a binary string of length n , in which each element is a logic variable that is set to 1 when it is considered by the reasoning process, or to 0, otherwise. Each combination will lead to a particular performance level, that can be classified by a binary variable as well: the value 1 means that the performance achieved is satisfactory, while the value 0 expresses the opposite. In other words, the reasoning process can be seen as a binary function $b : \{0, 1\}^n \rightarrow \{0, 1\}$. Figure 3.42 illustrates the process in case of a generic 3-concept FCM.

The process, as described, bears some analogies with the analysis of the output of a digital logic circuit starting from the inputs, in which the objective is to derive a logic formula representing the circuit. Exactly as in the case of a logic circuit, also in this context it is possible to draw a so-called Karnaugh map, which associates an output value for every possible combination of the inputs. By synthesizing the part of the map in which values are set to 1, i.e. where the performance level is deemed acceptable, a formula B can

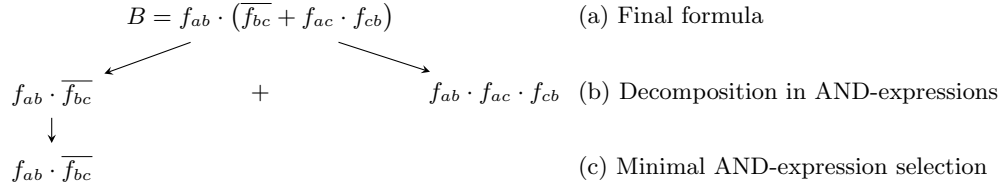


Figure 3.43: Decomposition as AND-expressions of the logic formula B , hypothetically synthesizing the Karnaugh map resulting from the toy example depicted in Figure 3.42

be derived. If we recall that each element in the map (and, thus, in the final formula) is an edge of the FCM and represents a cross-layer relationship, it is clear that the formula indicates which FCM elements should be taken into account by the cognitive process and which should be left out.

The formula can be composed by two types of terms: (i) non-negated variables, and (ii) negated variables. Both kinds of terms are important to the reasoning process, in that both have some effect on the inference result. However, they are important in different ways. FCM elements that appear as non-negated variables should be considered when running the inference method because they lead to meaningful results, whereas FCM elements that appear as negated variables should be avoided, because their use can mislead the reasoning process and lead to wrong predictions.

All the terms that are not present in the formula should be considered unimportant, i.e. taking them into account does not change the overall result. Hence, only the elements appearing in the final formula are really important to be carried out by the reasoning task, while all the others can be safely ignored. Consequently, ignoring such elements means that the causal relationships that they represent are weak, and deleting them can lead to the isolation of some concepts.

It should be noticed that it is possible to reduce the number of concepts even further. To illustrate this concept, let us refer to Figure 3.43. As shown in (a), the final formula will likely contain both OR and AND operations. If we refer to any set of terms connected only by the AND operator as an “AND-expression”, then we can say that expanding the formula will lead to several AND-expressions connected to one another by the OR operator. This is reflected in step (b). Each of these AND-expressions is equivalent to any other, since by definition they all lead to acceptable performance. Therefore, only the AND-expressions comprising the fewest non-negated elements could, in principle, be considered, and, among them, those involving the smallest number of concepts could be selected as the final formula, as shown in step (c).

Nevertheless, care should be taken when deciding whether adopting the full equation or the minimum AND-expression present in it. One could be tempted to choose the second option over the first, because such choice is less demanding from the point of view of the computation and should lead to the same performance anyway.

However, it should be observed that the full formula lists all the combinations that lead to acceptable performance levels. Hence, bad performance can be caused only by not having some concept in the matrix, i.e. concepts that should have been considered at the design stage, but were not.

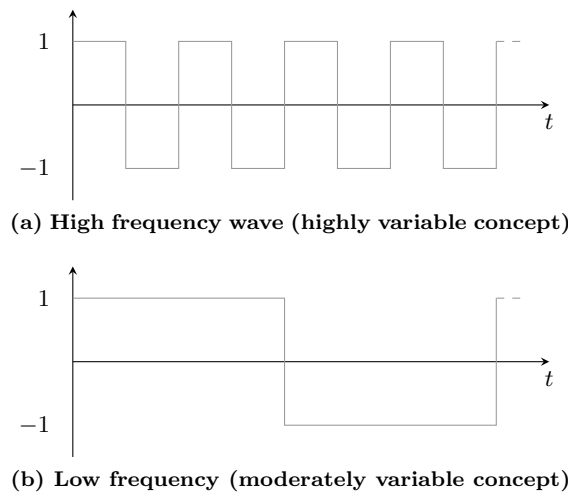


Figure 3.44: Extreme-case time evolutions of a cause-effect relation

Instead, the minimum AND-expression comprises a subset of all possible combinations. Bad performance can be experienced not only by failing to include important concepts when defining the FCM (as in the previous case), but also (and more likely) in the case some of the terms present in the formula lose their relevance as the time passes.

It is, thus, reasonable to expect that the minimum AND-expression is less reliable. If, on the one hand, there is a gain on the complexity of reasoning, on the other hand overhead must be added (e.g. by performing more frequent recalculations) so as to ensure reliability.

3.5.4-B A Lightweight Approach: The Discriminating Index

As mentioned in Section 3.5.2, edge labels in FCMs measure the strength of the cause-effect relation between two variables. As a consequence, the higher the absolute value of an edge label, the greater the cause-effect relationship. Therefore, the modulus (or magnitude) of the edge labels can be used for discriminating the edges that are relevant to the reasoning process from those which are not.

As the values of the edge labels fluctuate, an average value needs to be computed. The modulus can be taken either before or after the averaging operation. In practice, neither order is robust. Let us consider two extreme situations (exemplified in Figure 3.44): a high-frequency and a low-frequency polar unitary square waves, symbolizing rapid (in the first case) and slow (in the second case) variations of causality.

In a context in which the values of most edge labels change slowly (Figure 3.44b), an edge label that oscillates rapidly between positive and negative values (Figure 3.44a) can be a sign of an unreliable causal connection, potentially misleading the reasoning process. As a consequence, it would be beneficial to distinguish the two situations.

Averaging before taking the modulus yields a null values for both the causal relations in Figure 3.44, and it does not allow to discriminate between them. Averaging after taking the modulus yields one, which is likewise useless as, also in this case, no distinction between the two relations is possible.

This suggests selecting the oscillation between positive and negative values as a second discriminating feature. More specifically, the speed of oscillations can be approximated by the frequency of zero-crossings, i.e. the number of times the value of an edge label crosses the horizontal axis (null value). Causal connections that are characterized by a high variability can then be discarded in favor of the others.

However, as the edge labels are set to zero at the outset, it is likely that they experience a transient phase, during which they can oscillate around the axis, thus scoring a high number of zero-crossings. Considering the absolute number of zero-crossings would result in the exclusion of such edges from the reasoning process since the beginning. For this reason, it is advisable to normalize the number of zero-crossings of the edge label values with respect to the maximum number of zero-crossings registered.

In summary, the proposal is to build a discriminating metric based on two values, namely, the averaged absolute magnitude of the value of edge labels and the number of zero-crossings of edge labels themselves. Since the aim of this study is to show that it is possible to recognize which cross-layer relations influence reasoning, and not to show which is the best classifier, it was decided to simply combine these values by means of multiplication, and then adopt a simple one-dimensional classifier.

The choice of the multiplication as the feature extraction operation can be explained by looking at the final equation describing the discriminating index $d_{f_{ij}}(t)$ for a generic FCM element f_{ij} at time t :

$$d_{f_{ij}}(t) = \frac{1}{t} \int_0^t |f_{ij}(\tau)| d\tau \cdot \left(1 - \frac{zc(f_{ij})}{\max_{f_{ij}}\{zc(f_{ij})\}} \right) \quad (3.24)$$

where zc is a function returning the number of zero-crossings of a sequence since the beginning. As we are interested only in the elements that are characterized by high magnitude and low variability, the use of multiplication allows us to quickly discard all the other combinations, i.e. (i) high magnitude and high variability, (ii) low magnitude and high variability, (iii) low magnitude and low variability. Indeed, the resulting index will be low in all these cases.

Once the index is computed, a classifier needs to be employed, in order to discern the elements that are fundamental for the reasoning process from those that are not. As there is no a-priori knowledge about the probability density of the data points, it is advisable to employ a non-parametric classification technique. The Parzen window algorithm [85] was chosen to estimate the probability density function fitting the data¹². The classifier yields a multimodal distribution, based on which we can distinguish the cluster of relevant elements from the cluster of non-relevant elements. The cluster associated to the greatest discriminating index comprises the most important variables. All the other clusters, if any, contain non-relevant variables.

As the network evolves, it may be the case that some FCM elements lose their importance whereas others have their relevance increased. It is, therefore, clear that the discriminating index must be frequently updated during the lifetime of the cognitive process.

¹²There is plenty of non-parametric density estimation techniques in the literature. However, the choice of one technique over another is beyond the scope of this work.

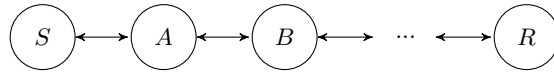


Figure 3.45: Test case network topology

3.5.4-C A Hybrid Method

Both the methods illustrated in Sections 3.5.4-A and 3.5.4-B can identify which cross-layer relationships are relevant to the reasoning process.

The main advantage of the logic circuit minimization approach is that it provides a closed-form solution to the problem. The formula is particularly helpful, since it explicitly specifies which elements reasoning should be based on and which elements can be neglected. The drawback is that, to synthesize the formula, the performance of all possible cases must be recorded, which is both time and resource consuming.

The discriminating index approach, too, identifies the elements of an FCM that are important for the reasoning process. Although it finds which elements influence the reasoning process, it cannot discriminate positive from negative influences (i.e. negated or non-negated variables, in the terms of the logic circuit minimization approach). In other words, the discriminating index alone helps to distinguish the *strength* of the influence an element has, but not its *kind*.

This may represent an issue only when negated terms appear in the Boolean equation synthesizing the Karnaugh map. More specifically, if, after expanding the formula, each AND-expression contains at least one negated term, then the use of the discriminating index will inevitably lead to poor performance. However, if at least one AND-expression is exclusively composed of non-negated terms, then the two approaches are equivalent, in the sense that they allow the reasoning process to reach the same performance.

To maintain the positive aspects of both schemes, a solution is to merge them and derive a joint method: the discriminating index is employed to preliminarily reduce the dimensionality of the problem, by discarding irrelevant FCM elements, while the logic circuit minimization is applied to definitively discern the positively influencing elements from the others. The resulting pseudocode is listed as algorithm 1.

3.5.5 Test Case

The scenario set up to validate the methodology proposed in this section involves a wireless ad-hoc multihop network based on the IEEE 802.11b standard, and was simulated using the NS-3 simulator. Nodes are positioned according to a chain topology, as depicted in Figure 3.45. In such configuration, the node at one end (S) starts an FTP transfer towards the node at the other end of the chain (R), lasting 100 seconds. To simulate the FTP protocol, it was decided to employ the NS-3 on/off application, specifying a duty cycle of 100%, and a data rate of 10 Mb/s. TCP Reno was used as the transport protocol and the segment size was set to 1460 bytes. Each node can communicate only with its immediate neighbors and routing is static.

Five input parameters have been selected: number of nodes, bit error rate, physical data rate, the use of RTS/CTS, and the use of layer-two fragmentation. Their range of

Algorithm 1 “Hybrid” approach

```

1:  $maxzc \leftarrow 1$ 
2:  $FCM\_R \leftarrow FCM$  //  $FCM\_R$  is the matrix containing the elements important for reasoning
3: if zerocrossing then
4:   update  $maxzc$ 
5:    $rel\_v.clear()$  //  $rel\_v$  is the vector containing the relevant elements
6:   for  $i, j = 1 \in \{1, 2, \dots, c\}, i \neq j$  do
7:     if Classifier(DiscrIndex( $FCM[i, j]$ )) == relevant then
8:        $rel\_v.push(FCM[i, j])$  // the classifier output can be either relevant or not
9:     end if
10:  end for
11:   $\hat{n} \leftarrow \dim(rel\_v)$ 
12:  for  $rel\_v[1] \in \{0, 1\}$  do
13:    for  $rel\_v[2] \in \{0, 1\}$  do
14:      for ... do
15:        for  $rel\_v[\hat{n}] \in \{0, 1\}$  do
16:           $Kmap[rel\_v[1], rel\_v[2], \dots, rel\_v[\hat{n}]] \leftarrow RunFCM(rel\_v[1], rel\_v[2], \dots, rel\_v[\hat{n}])$ 
17:          //  $Kmap$  contains the performance for every combination of the relevant elements
18:        end for
19:      end for
20:    end for
21:  end for
22:  for all  $FCM\_R[i, j] \notin \text{MinimizationAlgorithm}(Kmap)$  do
23:     $FCM\_R[i, j] \leftarrow 0$ 
24:    // suppose the MinimizationAlgorithm returns just the list of non-negated variables
25:  end for
26: end if

```

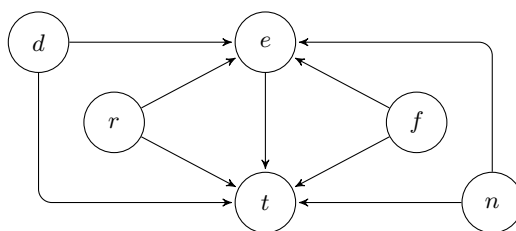


Figure 3.46: Test case Fuzzy Cognitive Map. Concepts $t, n, e, d, r,$ and f stand for throughput, number of nodes, error rate, physical data rate, RTS/CTS handshake, and fragmentation, respectively.

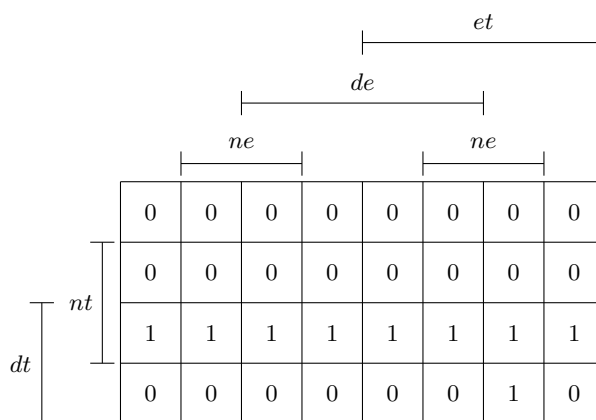


Figure 3.47: Karnaugh map of the test case (reduced for better clarity)

Table 3.7: Domains of the input variables

Property	Abbr.	Value
Number of nodes	n	$\{3, \dots, 9\}$
PHY bit error rate	e	$\{0, 3 \cdot 10^{-8}, 3 \cdot 10^{-7}, \dots, 3 \cdot 10^{-5}\}$
PHY data rate	d	$\{1, 2, 5.5, 11\}$ Mb/s
MAC fragmentation	f	$\{\text{on}, \text{off}\}$
MAC RTS/CTS handshake	r	$\{\text{on}, \text{off}\}$

Table 3.8: Cross-layer relations characterizing the test case analyzed

Cause concept	Effect concept	Abbr.
Error rate	Throughput	et
Physical data rate	Throughput	dt
Physical data rate	Error rate	de
Number of nodes	Throughput	nt
Number of nodes	Error rate	ne
RTS/CTS handshake	Throughput	rt
RTS/CTS handshake	Error rate	re
Fragmentation	Throughput	ft
Fragmentation	Error rate	fe

Table 3.9: Performance achieved using different subsets of elements to perform the reasoning. Student's t distribution with 29 d.o.f., $p = 0.95$.

Elements considered	Relative improvement [%]	Confidence interval
All 9 elements considered	+4.67	0.29
$dt(nt + ne \cdot et \cdot \overline{de})$	+5.04	0.31
$dt \cdot nt$	+5.04	0.31
$dt \cdot ne \cdot et \cdot \overline{de}$	+4.34	0.45
$dt \cdot ne \cdot et \cdot de$	-3.27	0.48
$nt + ne \cdot et \cdot \overline{de}$	-5.69	0.03

variation are summarized in Table 3.7. Such parameters were chosen because it is known that they all exert some effect on network performance. For instance, fragmentation and data rate fallback mechanisms were introduced to thwart physical layer impairments. The goal is to analyze whether some of the parameters have little influence on the proposed predictive model and are not worth to be considered or all of them are important and all should be comprised by the reasoning process.

Each combination of the inputs has been simulated 15 times and the results averaged, in order to eliminate potential spurious values. The cognitive engine was used to predict the throughput level, given the inputs.

As can be inferred from the FCM reported in Figure 3.46, at the outset there were 9 cross-layer relations that could potentially be used by the reasoning engine. For the sake of clarity, they are listed in Table 3.8, where, along with the names of the two concepts linked by the cross-layer relations, an abbreviation is introduced, by joining the letters of the concepts involved. For example, according to this notation, dt denotes the relationship between data rate and throughput.

The application of the procedure used for minimization of logic circuits (described in Section 3.5.4-A) made it possible to find the combinations of cross-layer relationships upon which reasoning should be based. The final formula, that can be derived by means of the Karnaugh map shown in Figure 3.47, is:

$$B = dt (nt + et \cdot \overline{de} \cdot ne) \quad (3.25)$$

Thus, only 5 out of 9 cross-layer relations can influence the reasoning process. Besides that, one of them, i.e. the relationship between data rate and errors, appears negated, which means it should not be considered by the inference mechanism. It could be objected that, although it is well known that fragmenting packets decreases the error rate, the formula suggests not to consider fragmentation. However, it should be noted that this does not mean that fragmentation does not reduce the error rate. Rather, it means that the causal relationship between fragmentation and error rate is not meaningful to the reasoning process. In particular, the relation between fragmentation and error rate is weaker or oscillates more than the relations among the other variables.

Table 3.9 shows some of the results that can be achieved by the reasoning process, as a function of the elements considered: as long as the terms of either one of the AND-expressions are included in the reasoning process, performance is positive. When this condition fails, the reasoning process is bound to yield bad results. As illustrated in Figure 3.48, the discriminating index approach managed to capture the relevance of the terms appearing in Equation 3.25. As expected, de appears among the relevant terms, and nothing points out it should not be considered during reasoning.

3.5.6 Conclusion

The reasoning process in cognitive networks must converge to a solution before the operating environment changes. Therefore, it is crucial to maintain reasoning times as low as possible, while conserving high reliability of the results.

According to the reasoning formalism presented in Section 3.1, to achieve the ultimate objective of reducing reasoning times, it is mandatory to first reduce the number of cause-

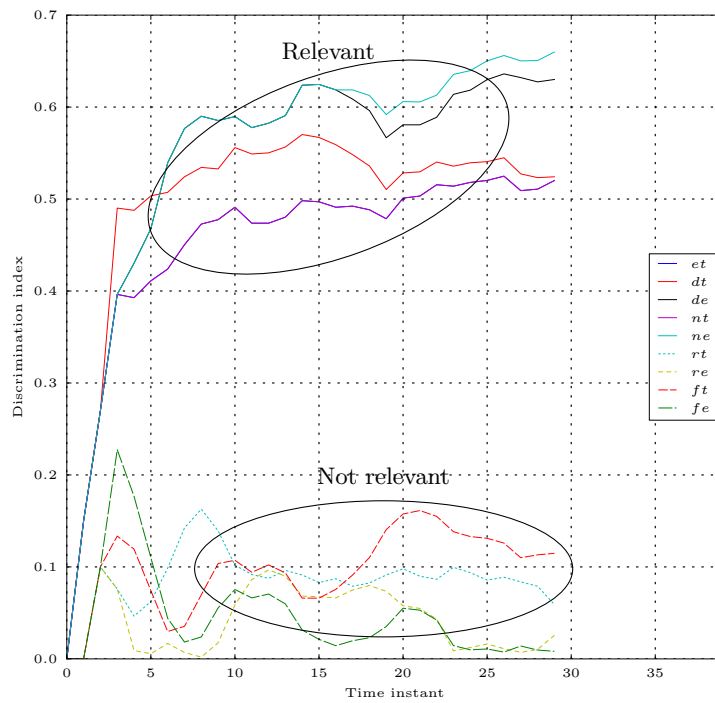


Figure 3.48: Evolution of the discriminating index of the cross-layer relations in the test case. For abbreviations, refer to Table 3.8.

effect relationships considered, without affecting the effectiveness of the reasoning process. Along this direction, in this section a method has been proposed to identify the cross-layer relations that are not relevant to the reasoning process, i.e. those relations that do not exert any influence on it. The method is composed of two steps. In the first step it attempts to reduce the dimensionality of the problem, keeping only the relations characterized by a high causal significance and low variability. In the second step it aims to discover the relevant cross-layer relations, among those that were not discarded during the first phase, by means of exhaustive search.

The proposed scheme is validated through simulations, to underline it represents a possible solution for reducing reasoning time.

Chapter 4

Conclusion

Cognitive networking is a paradigm that aims to address two of the major problems affecting today's communications networks: management complexity and inability to achieve optimal performance in dynamic environments. The core idea is to make networks intelligent, capable of learning, reasoning, and acting in order to both manage themselves and reach higher performance levels, all with minor intervention—ideally none—by humans. Cognitive network nodes can be thought of as complex systems, whose performance highly depends on non-obvious cross-layer interactions. For this reason, a largely accepted opinion among researchers is to base cognitive processes on a holistic vision of network nodes. However, the cognitive networks that have been presented in the literature so far do not implement such a holistic vision in a satisfactory way and several questions on how to pursue this line of research still remain open.

In this work, we have presented a cognitive network architecture that capitalizes on both intra- and inter-node cross-layer interactions, to make this vision possible.

We employed a framework known as Fuzzy Cognitive Map (FCM) as the core of the reasoning process. Such formalism makes it possible to both achieve better performance in dynamic scenarios and understand the cause-effect relationships that bind together the variables internal to network nodes. The basis of the work has been laid in Section 3.1, in which we introduced and thoroughly described the formal process to embed this framework in a network environment. Furthermore, we precisely identified and discussed the problems that can be commonly encountered while performing such process.

In Section 3.2 we presented an instance of FCM-based cognitive network. We combined it with the principles behind service-oriented architectures to derive a cognitive service-oriented infrastructure. We offered an implementation in two different wireless scenarios, one centralized and one distributed, and showed its potential for performing call-admission control and throughput estimation, respectively.

The focus of Section 3.3 was on the details of the implementation. After discussing them, we illustrated a complete example of the architecture. All its components were presented and analyzed, and the design choices motivated. An application example was given to show how the cognitive process can be used to adjust the data rate in a wireless node. Simulations of the architecture demonstrated that implementing the cognitive paradigm allows to achieve better results.

In Section 3.4 the architecture was further refined, by adding some tweaks at the

different stages of the cognitive loop. Adaptive thresholds and hysteresis were introduced in the sensing stage, new learning rules were proposed, and a novel model to select among different solutions was introduced.

Finally, in Section 3.5 we proposed a procedure to isolate non-relevant cross-layer relationships. We showed that such relationships do not add any valuable information to the reasoning process and can be safely discarded, thereby allowing a potential reduction of the time devoted to reasoning.

In summary, in this work a complete working cognitive architecture has been devised, designed and simulated. However, some research aspects still remain open. Research topics worth investigating include the fusion of multiple FCMs and the thorough definition of a scheme to exchange commands, measurements, and the FCMs themselves.

Regarding the former aspect, it should be noted that our design choices pose no restrictions whatsoever on the number of cognitive entities that could be installed in a network. Multiple entities may coexist and can coordinate with each other and exchange their beliefs via the cognitive management protocol that we envisioned. However, care must be taken before deploying such a scheme. Indeed, though it is straightforward to combine multiple maps with one another [42], it is not obvious that this operation will lead to greater performance achievements.

Remarkably important are the effects that actions taken by a reasoning entity have on other reasoning entities. As an example, let us assume we want to extend the scenario described in Section 3.4. Let us suppose we have two neighboring cognitive base stations, and let us also assume that it is possible for the terminals in a cell to join the other if connection is unavailable. It is possible that one of the two base stations will learn that it can remain turned off for the major part of the day, while still measuring a low blocking probability (in fact users can join the other base station). This demonstrates the first issue: apparently, selfish behavior could lead to greater savings for oneself, but to greater loss for the other. In this case, the exchange of FCMs would exacerbate the situation even more. Both reasoning entities would be led to believe that the best option is to turn off as many radio modules as possible, which is clearly a not worthy solution.

One simple option would be for a reasoning process to take into account variables belonging to the *other* reasoning process and set common goals for all the entities. Clearly, the biggest drawback is that such a solution does not scale. Another idea is to resort to either cooperative or non-cooperative game theory, which could highlight what are the points of equilibrium in such a system. From this perspective, the exchange of FCMs becomes a means to make the game *complete*, i.e. a game in which players know exactly what are the beliefs of their opponents.

Another, partly related, concern regards the reputation of cognitive entities. A means is needed to evaluate the reputation of a cognitive entity, so to decide whether its beliefs are worth being used. Such a situation is even more problematic if we remove the assumption that no malicious nodes can be present in the network. If that happens, reasoning entities could be led to acquire bad new notions, with the ultimate result of worsening the overall performance.

The second aspect concerns the definition of a scheme to exchange data, actions and beliefs. We already envisioned the presence of a so-called cognitive management protocol.

However, we refrained from establishing any rule regarding syntax, semantics, and timings. The major aspects to tackle are the definition of a specific format that unambiguously represents FCMs, as well as raw measurements or particular actions. Information, as we have mentioned in Section 3.3, could be embedded in unused fields of the headers in a packet [30] or even in the so-called extension headers, implemented in version 6 of the IP protocol [86].

However, to be truly regarded as a remarkable contribution to the state of the art, such a protocol should be designed in a very general way. It should be apt to be employed by different cognitive architectures that share the same needs of exchanging measurements, commands and beliefs. Ideally, a mechanism should be devised that lets each cognitive entity in a network know services and capabilities offered by other entities. As a consequence, each cognitive entity could decide by itself what information it will need and what other components could drive. Finally, it is of paramount importance to assess the impact of missing or delayed signaling messages, and decide whether connectionless protocols can be used to transport such data or connection-oriented protocols are necessary for the functioning of the system.

Bibliography

- [1] IEEE C802.20-03/80, *Traffic models for IEEE 802.20 MBWA system simulations, ver. 2*, July 2003. Available online at http://www.ieee802.org/20/DropBox/IEEE802.20_Tfc_Modeling_ToC_Baseline_V02.pdf.
- [2] Cisco Systems, “Cisco visual networking index: Forecast and methodology, 2010-2015.” White Paper, June 2011.
- [3] C. Fortuna and M. Mohorčič, “Trends in the development of communication networks: Cognitive networks,” *Comput. Netw.*, vol. 53, no. 9, pp. 1354–1376, 2009.
- [4] M. Endrei, J. Ang, A. Arsanjani, S. Chua, P. Comte, P. Krogdahl, M. Luo, and T. Newling, *Patterns: Service-Oriented Architecture and Web Services*. IBM Corp., July 2004. Available online.
- [5] S. Chaari, Y. Badr, and F. Biennier, “Enhancing web service selection by QoS-based ontology and WS-Policy,” in *ACM Symposium on Applied Computing*, (Fortaleza, CE, Brazil), pp. 2426–2431, ACM, 2008.
- [6] OECD, *OECD Information Technology Outlook 2010*. OECD Publishing, 2010.
- [7] H. Zimmermann, “Osi reference model—the iso model of architecture for open systems interconnection,” *Communications, IEEE Transactions on*, vol. 28, pp. 425–432, Apr. 1980.
- [8] R. Mortier and E. Kiciman, “Autonomic network management: some pragmatic considerations,” in *Proceedings of the 2006 SIGCOMM workshop on Internet network management, INM '06*, (New York, NY, USA), pp. 89–93, ACM, 2006.
- [9] R. W. Thomas, *Cognitive Networks*. PhD dissertation, Virginia Tech, Virginia, USA, 2007.
- [10] P. Johnson, “New research lab leads to unique radio receiver,” *E-System TEAM*, vol. 5, pp. 6–7, May 1985.
- [11] J. Mitola, “Software radios—survey, critical evaluation and future directions,” in *National Telesystems Conference (NTC)*, pp. 13–23, IEEE, 1992.
- [12] J. I. Mitola, “Cognitive radio for flexible mobile multimedia communications,” in *IEEE International Workshop on Mobile Multimedia Communications (MoMuC'99)*, (San Diego, CA, USA), pp. 3–10, 1999.

- [13] R. W. Thomas, L. A. DaSilva, and A. B. MacKenzie, "Cognitive networks," in *1st IEEE Int. Symposium on New Frontiers in Dynamic Spectrum Access Networks (DySPAN)*, pp. 352–360, 2005.
- [14] S. Dobson, S. Denazis, A. Fernández, D. Gaiti, E. Gelenbe, F. Massacci, P. Nixon, F. Saffre, N. Schmidt, and F. Zambonelli, "A survey of autonomic communications," *ACM Trans. Auton. Adapt. Syst.*, vol. 1, no. 2, pp. 223–259, 2006.
- [15] M. Smirnov, "Autonomic communication: Research agenda for a new communication paradigm," Nov. 2004.
- [16] J. O. Kephart and D. M. Chess, "The vision of autonomic computing," *IEEE Computer*, vol. 36, no. 1, pp. 41–50, 2003.
- [17] D. D. Clark, C. Partridge, J. C. Ramming, and J. T. Wroclawski, "A knowledge plane for the internet," in *Conf. on Applications, Technologies, Architectures, and Protocols for Computer Communications*, (Karlsruhe, Germany), pp. 3–10, ACM, 2003.
- [18] P. Mähönen, M. Petrova, J. Riihijarvi, and M. Wellens, "Cognitive wireless networks: Your network just became a teenager," in *25th Conference on Computer Communications*, (Barcelona, Spain), 2006.
- [19] B. Kosko, "Fuzzy cognitive maps," *Int. J. Man-Mach. Stud.*, vol. 24, no. 1, pp. 65–75, 1986.
- [20] P. Sutton, L. E. Doyle, and K. E. Nolan, "A reconfigurable platform for cognitive networks," in *1st Int. Conf. on Cognitive Radio Oriented Wireless Networks and Communications (CROWNCOM)*, pp. 1–5, 2006.
- [21] B. S. Manoj, R. R. Rao, and M. Zorzi, "On the use of higher layer information for cognitive networking," in *IEEE Global Telecommunications Conference (GLOBECOM)*, pp. 3568–3573, 2007.
- [22] P. Demestichas, V. Stavroulaki, D. Boscovic, A. Lee, and J. Strassner, "m@ANGEL: autonomic management platform for seamless cognitive connectivity to the mobile internet," *IEEE Comm. Mag.*, vol. 44, no. 6, pp. 118–127, 2006.
- [23] D. Raychaudhuri, N. B. Mandayam, J. B. Evans, B. J. Ewy, S. Seshan, and P. Steenkiste, "CogNet: an architectural foundation for experimental cognitive radio networks within the future internet," in *1st ACM/IEEE Int. Workshop on Mobility in the Evolving Internet Architecture*, (San Francisco, California), pp. 11–16, ACM, 2006.
- [24] Community Research and Development Information Service for Science, Research and Development (7th FWP), "Adaptive reconfigurable access and generic interfaces for optimisation in radio networks (ARAGORN)," Jan. 2008. <http://www.ict-aragorn.eu>.

-
- [25] D. Bourse, S. Buljore, A. Delauter, T. Wiebke, M. Dillinger, J. Brakensiek, K. Moessner, K. El-Khazen, and N. Alonistioti, "The End-to-End reconfigurability (E2R) research," in *Proceedings of the SDR Forum Technical Conference*, (Orlando, USA), 2003.
- [26] A. Kaloxylos, T. Rosowski, K. Tsagkaris, J. Gebert, E. Bogenfeld, P. Magdalinos, A. Galani, and K. Nolte, "The e3 architecture for future cognitive mobile networks," in *IEEE 20th Int. Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, pp. 1601–1605, Sept. 2009.
- [27] European Science Foundation, "Action IC0902: Cognitive radio and networking for cooperative coexistence of heterogeneous wireless networks," May 2009. http://w3.cost.esf.org/index.php?id=110&action_number=IC0902.
- [28] P. Langley, J. E. Laird, and S. Rogers, "Cognitive architectures: Research issues and challenges," *Cognitive Systems Research*, vol. 10, pp. 141–160, June 2009.
- [29] R. W. Thomas, D. H. Friend, L. A. DaSilva, and A. B. MacKenzie, "Cognitive networks: Adaptation and learning to achieve End-to-End performance objectives," *IEEE Comm. Mag.*, vol. 44, no. 12, pp. 51–57, 2006.
- [30] D. Kliazovich, F. Granelli, and N. L. S. D. Fonseca, *Architectures and Cross-Layer Design for Cognitive Networks*, ch. 2. World Scientific Publishing, 2009.
- [31] D. Fletcher, D. Nguyen, and K. Cios, "Autonomous synthesis of fuzzy cognitive maps from observational data: Preliminaries," in *IEEE Aerospace Conference*, pp. 1–9, 2005.
- [32] K. Herrmann, G. Mühl, and K. Geihs, "Self management: the solution to complexity or just another problem?," *Distributed Systems Online, IEEE*, vol. 6, no. 1, 2005.
- [33] T. R. Newman, B. A. Barker, A. M. Wyglinski, A. Agah, J. B. Evans, and G. J. Minden, "Cognitive engine implementation for wireless multicarrier transceivers," *Wireless Communications and Mobile Computing*, vol. 7, no. 9, pp. 1129–1142, 2007.
- [34] D. H. Friend, M. Y. ElNainay, Y. Shi, and A. B. MacKenzie, "Architecture and performance of an island genetic algorithm-based cognitive network," in *5th IEEE Consumer Communications and Networking Conference (CCNC)*, (Las Vegas, NV, USA), pp. 993–997, 2008.
- [35] P. Marrow, "Nature-Inspired computing technology and applications," *BT Technology Journal*, vol. 18, pp. 13–23, Oct. 2000.
- [36] M. D. Benedetto and L. D. Nardis, "Cognitive routing models in UWB networks," in *3rd Int. Conf. on Cognitive Radio Oriented Wireless Networks and Communications (CrownCom)*, pp. 1–6, 2008.
- [37] E. Gelenbe, R. Lent, A. Montuori, and Z. Xu, "Cognitive packet networks: Qos and performance," in *10th Int. Symposium on Modeling, Analysis and Simulation of Computer and Telecommunications Systems (MASCOTS)*, pp. 3–9, 2002.

- [38] Z.-Q. Liu, "Causation, bayesian networks, and cognitive maps," *Acta Automatica Sinica*, vol. 27, no. 4, pp. 552–566, 2001.
- [39] A. K. Tsadiras, "Comparing the inference capabilities of binary, trivalent and sigmoid fuzzy cognitive maps," *Information Sciences*, vol. 178, no. 20, pp. 3880–3894, 2008. Special Issue on Industrial Applications of Neural Networks, 10th Engineering Applications of Neural Networks 2007.
- [40] Y. Miao and L. Zhi-Qiang, "On causal inference in fuzzy cognitive maps," *Fuzzy Systems, IEEE Transactions on*, vol. 8, no. 1, pp. 107–119, 2000.
- [41] D. Kliazovich, N. Malheiros, N. L. da Fonseca, F. Granelli, and E. Madeira, "Cog-Prot: a framework for cognitive configuration and optimization of communication protocols," in *Mobile Lightweight Wireless Systems: 2nd Intl. ICST Conference (Mobilight)*, vol. 45, (Barcelona, Spain), pp. 280–291, Springer-Verlag New York, May 2010.
- [42] B. Kosko, *Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence*. Prentice-Hall, 1992.
- [43] D. H. Friend, R. W. Thomas, A. B. MacKenzie, and L. A. DaSilva, *Distributed Learning and Reasoning in Cognitive Networks: Methods and Design Decisions*, ch. 9, pp. 223–246. Wiley-Interscience, July 2007.
- [44] F. Granelli, P. Pawelczak, R. V. Prasad, K. P. Subbalakshmi, R. Chandramouli, J. A. Hoffmeyer, and H. S. Berger, "Standardization and research in cognitive and dynamic spectrum access networks: IEEE SCC41 efforts and other activities," *IEEE Comm. Mag.*, vol. 48, no. 1, pp. 71–79, 2010.
- [45] K. Leibnitz, N. Wakamiya, and M. Murata, *Biologically inspired networking*, ch. 1, pp. 1–21. Wiley-Interscience, July 2007.
- [46] C. Facchini and F. Granelli, "Towards a model for quantitative reasoning in cognitive nodes," in *3rd IEEE Workshop on Enabling the Future Service-Oriented Internet—Towards Socially-Aware Networks*, (Honolulu, Hawaii, USA), pp. 1–6, Dec. 2009.
- [47] C. Facchini, F. Granelli, and N. L. da Fonseca, "Cognitive service-oriented infrastructures," *Journal of Internet Engineering*, vol. 4, no. 1, pp. 269–278, 2010. Special issue on Service-Oriented Architectures.
- [48] C. Facchini, F. Granelli, and N. L. da Fonseca, "Cognitive rate adaptation in wireless LANs," in *IEEE Int. Conf. on Communications (ICC) - Communications QoS, Reliability and Modelling Symposium (CQRM)*, (Kyoto, Japan), June 2011.
- [49] R. L. Gomes, C. Facchini, F. Granelli, E. Madeira, and N. L. da Fonseca, "Um mecanismo de raciocínio para redes cognitivas baseado em inferência bayesiana," in *SBRC 2011 - WRA*, June 2011.
- [50] C. Facchini, O. Holland, F. Granelli, N. L. da Fonseca, and H. Aghvami, "Dynamic green self-configuration of 3G base stations using fuzzy cognitive maps," *Computer Networks, Special Issue on Green Communications*, 2012. Submitted.

-
- [51] O. Holland, C. Facchini, H. Aghvami, O. Cabral, and F. Velez, *Green Radio Communication Networks*, ch. Opportunistic Spectrum and Load Management for Green Radio, pp. 56–99. College Station, Texas: Cambridge University Press, 2012.
- [52] C. Facchini, F. Granelli, and N. L. da Fonseca, “Identifying relevant cross-layer interactions in cognitive processes,” in *IEEE Globecom 2010 - Communications QoS, Reliability and Modelling Symposium (GC10 - CQRM)*, (Miami, Florida, USA), Dec. 2010.
- [53] N. Baldo and M. Zorzi, “Cognitive network access using fuzzy decision making,” in *IEEE International Conference on Communications (ICC '07)*, pp. 6504–6510, June 2007.
- [54] C. D. Stylios, V. C. Georgopoulos, G. A. Malandraki, and S. Chouliara, “Fuzzy cognitive map architectures for medical decision support systems,” *Applied Soft Computing*, vol. 8, no. 3, pp. 1243–1251, 2008. Forging the Frontiers – Soft Computing.
- [55] J. A. Dickerson and B. Kosko, “Virtual worlds as fuzzy cognitive maps,” in *IEEE Virtual Reality Annual Int. Symposium*, (Seattle, WA, USA), pp. 471–477, 1993.
- [56] J. Aguilar, “A survey about fuzzy cognitive maps papers,” *International Journal of Computational Cognition*, vol. 3, pp. 27–33, 2005.
- [57] E. Sánchez-Nielsen, S. Martin-Ruiz, and J. Rodriguez-Pedrianes, “An open and dynamical service oriented architecture for supporting mobile services,” in *6th Intl. Conf. on Web engineering (ICWE)*, pp. 121–128, 2006.
- [58] R. Sen, R. Handorean, G.-C. Roman, and C. Gill, *Service-Oriented Computing Imperatives in Ad Hoc Wireless Settings*, ch. 12. Idea Group Inc., 2005.
- [59] L. Juszczak, J. Lazowski, and S. Dustdar, “Web service discovery, replication, and synchronization in ad-hoc networks,” in *1st Intl. Conf. on Availability, Reliability and Security (ARES)*, (Vienna, Austria), pp. 847–854, Apr. 2006.
- [60] T. Halonen and T. Ojala, “Cross-layer design for providing service oriented architecture in a mobile ad hoc network,” in *5th Intl. Conf. on Mobile and Ubiquitous Multimedia (MUM)*, p. 11, 2006.
- [61] A. S. Vedamuthu, D. Orchard, F. Hirsch, M. Hondo, P. Yendluri, T. Boubez, and Ümit Yalçınalp, “Web services policy 1.5 - framework,” Sept. 2007.
- [62] F. Granelli, D. Kliazovich, J. Hui, and M. Devetsikiotis, “Performance optimization of Single-Cell voice over WiFi communications using quantitative Cross-Layering analysis,” in *Managing Traffic Performance in Converged Networks*, vol. 4516/2007 of *Lecture Notes in Computer Science*, pp. 386–397, Springer Berlin, 2007.
- [63] “IEEE standard for information Technology-Telecommunications and information exchange between Systems-Local and metropolitan area Networks-Specific requirements - part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications,” *IEEE Std 802.11-2007 (Revision of IEEE Std 802.11-1999)*, June 2007.

- [64] A. Kamerman and L. Monteban, “WaveLAN[®]-II: a high-performance wireless LAN for the unlicensed band,” *Bell Labs technical journal*, vol. 2, no. 3, pp. 118–133, 1997.
- [65] F. Maguolo, M. Lacage, and T. Turetletti, “Efficient collision detection for auto rate fallback algorithm,” in *IEEE Symp. on Computers and Communications (ISCC)*, (Marrakech, Morocco), pp. 25–30, July 2008.
- [66] J. Kim, S. Kim, S. Choi, and D. Qiao, “CARA: collision-aware rate adaptation for IEEE 802.11 WLANs,” in *IEEE INFOCOM*, (Barcelona, Spain), pp. 1–11, Apr. 2006.
- [67] M. Lacage, M. H. Manshaei, and T. Turetletti, “IEEE 802.11 rate adaptation: A practical approach,” in *7th ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, (Venice, Italy), pp. 126–134, ACM, 2004.
- [68] N. Baldo and M. Zorzi, “Learning and adaptation in cognitive radios using neural networks,” in *5th IEEE Consumer Communications and Networking Conference (CCNC)*, (Las Vegas, NV, USA), pp. 998–1003, 2008.
- [69] L. Chaves, N. Malheiros, E. Madeira, I. Garcia, and D. Kliazovich, “A cognitive mechanism for rate adaptation in wireless networks,” in *Modelling Autonomic Communications Environments*, vol. 5844 of *Lecture Notes in Computer Science*, pp. 58–71, Springer Berlin / Heidelberg, 2009.
- [70] M. G. Kallitsis, G. Michailidis, and M. Devetsikiotis, “Measurement-based optimal resource allocation for network services with pricing differentiation,” *Performance Evaluation*, vol. 66, pp. 505–523, Sept. 2009.
- [71] A. Bianzino, C. Chaudet, D. Rossi, and J. Rougier, “A survey of green networking research,” *Communications Surveys & Tutorials, IEEE*, pp. 1–18, 2010. Accepted for inclusion in a future issue of the journal.
- [72] The Climate Group London and The Global e-Sustainability Initiative (GeSI), “Smart 2020: Enabling the low carbon economy in the information age,” 2008.
- [73] L. Saker and S. E. Elayoubi, “Sleep mode implementation issues in green base stations,” in *IEEE 21st Int. Symposium on Personal Indoor and Mobile Radio Communications (PIMRC)*, pp. 1683–1688, Sept. 2010.
- [74] D. Tipper, A. Rezgui, P. Krishnamurthy, and P. Pacharintanakul, “Dimming cellular networks,” in *IEEE Global Telecommunications Conference (GLOBECOM)*, pp. 1–6, Dec. 2010.
- [75] E. Oh and B. Krishnamachari, “Energy savings through dynamic base station switching in cellular wireless access networks,” in *IEEE Global Telecommunications Conference (GLOBECOM)*, pp. 1–5, Dec. 2010.
- [76] S. Thajchayapong and J. M. Peha, “Mobility patterns in microcellular wireless networks,” *IEEE Transactions on Mobile Computing*, vol. 5, pp. 52–63, Jan. 2006.

-
- [77] M. A. Marsan, L. Chiaraviglio, D. Ciullo, and M. Meo, "Optimal energy savings in cellular access networks," in *IEEE Int. Conference on Communications Workshops (ICC)*, pp. 1–5, June 2009.
- [78] L. Chiaraviglio, D. Ciullo, M. Meo, and M. A. Marsan, "Energy-efficient management of UMTS access networks," in *21st Int. Teletraffic Congress (ITC)*, pp. 1–8, IEEE, Sept. 2009.
- [79] P. Brady, "A model for generating on-off speech patterns in two-way conversation," *The Bell System Technical Journal*, vol. 48, pp. 2445–2472, 1969.
- [80] O. Cabral, F. Meucci, A. Mihovska, F. J. Velez, N. R. Prasad, and R. Prasad, "Integrated common radio resource management with spectrum aggregation over non-contiguous frequency bands," *Wireless Personal Communications*, vol. 59, pp. 499–523, Feb. 2011.
- [81] F. Richter, A. J. Fehske, and G. P. Fettweis, "Energy efficiency aspects of base station deployment strategies for cellular networks," in *70th Vehicular Technology Conference Fall*, pp. 1–5, IEEE, Sept. 2009.
- [82] B. S. Manoj, R. R. Rao, and M. Zorzi, "Cognet: a cognitive complete knowledge network system," *IEEE Wireless Commun. Mag.*, vol. 15, pp. 81–88, Dec. 2008.
- [83] V. Kawadia and P. R. Kumar, "A cautionary perspective on Cross-Layer design," *IEEE Wireless Commun. Mag.*, vol. 12, no. 1, pp. 3–11, 2005.
- [84] N. Baldo and M. Zorzi, "Fuzzy logic for Cross-Layer optimization in cognitive radio networks," *Communications Magazine, IEEE*, vol. 46, no. 4, pp. 64–71, 2008.
- [85] R. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. Wiley, 2, illustrated ed., 2001.
- [86] O. Holland, T. Mahmoodi, V. Friderikos, and A. H. Aghvami, "Cross-layer optimization: Network layer involvement," in *World Wireless Research Forum (WWRF 20)*, (Ottawa, Canada), 2008.

Appendix A

List of Publications

- [1] C. Facchini and F. Granelli, “Game theory as a tool for modeling cross-layer interactions,” in *IEEE Int. Conf. on Communications (ICC)*, (Dresden, Germany), pp. 1–5, June 2009.
- [2] C. Facchini and F. Granelli, “Towards a model for quantitative reasoning in cognitive nodes,” in *3rd IEEE Workshop on Enabling the Future Service-Oriented Internet—Towards Socially-Aware Networks*, (Honolulu, Hawaii, USA), pp. 1–6, Dec. 2009.
- [3] C. Facchini, F. Granelli, and N. L. da Fonseca, “Cognitive service-oriented infrastructures,” *Journal of Internet Engineering*, vol. 4, no. 1, pp. 269–278, 2010. Special issue on Service-Oriented Architectures.
- [4] C. Facchini, F. Granelli, and N. L. da Fonseca, “Identifying relevant cross-layer interactions in cognitive processes,” in *IEEE Globecom 2010 - Communications QoS, Reliability and Modelling Symposium (GC10 - CQRM)*, (Miami, Florida, USA), Dec. 2010.
- [5] C. Facchini, F. Granelli, and N. L. da Fonseca, “Cognitive rate adaptation in wireless LANs,” in *IEEE Int. Conf. on Communications (ICC) - Communications QoS, Reliability and Modelling Symposium (CQRM)*, (Kyoto, Japan), June 2011.
- [6] R. L. Gomes, C. Facchini, F. Granelli, E. Madeira, and N. L. da Fonseca, “Um mecanismo de raciocínio para redes cognitivas baseado em inferência bayesiana,” in *SBRC 2011 - WRA*, June 2011.
- [7] O. Holland, C. Facchini, H. Aghvami, O. Cabral, and F. Velez, *Green Radio Communication Networks*, ch. Opportunistic Spectrum and Load Management for Green Radio, pp. 56–99. College Station, Texas: Cambridge University Press, 2012.
- [8] C. Facchini, O. Holland, F. Granelli, N. L. da Fonseca, and H. Aghvami, “Dynamic green self-configuration of 3G base stations using fuzzy cognitive maps,” *Computer Networks, Special Issue on Green Communications*, 2012. Submitted.

Appendix B

List of the acronyms used

AARF Adaptive ARF

AARF-CD AARF-Collision Detection

ACK acknowledgment frame

AP Access Point

ARF Automatic Rate Fallback

BPMN Business Process Model and Notation

BS Base Station

CARA Collision-Aware Rate Adaptation

CoSOI Cognitive Service-Oriented Infrastructure

CTS Clear to Send

DAML DARPA Agent Markup Language

DARPA Defense Advanced Research Projects Agency

DHL Differential Hebbian Learning

DSP Digital Signal Processing

EBL Exponential Backoff Learning

EWMA Exponentially Weighted Moving Average

FCM Fuzzy Cognitive Map

FER Frame Error Rate

FP Framework Programme

FTP File Transfer Protocol

HDTV High-Definition Television

HSDPA	High Speed Downlink Packet Access
HTTP	Hypertext Transfer Protocol
IEEE	Institute of Electrical and Electronics Engineers
IMT	International Mobile Telecommunications
IP	Internet Protocol
ISO	International Organization for Standardization
IT	Information Technology
JMS	Java Message Service
LAN	Local Area Network
LCL	Lower Control Limit
LL	Linear Learning
MAC	Medium Access Control
MIB	Management Information Base
OECD	Organisation for Economic Co-operation and Development
OSI	Open Systems Interconnection
PER	Packet Error Rate
PHY	Physical Layer
QoS	Quality of Service
RDF	Resource Description Framework
RKRL	Radio Knowledge Representation Language
RNC	Radio Network Controller
RTS	Request to Send
SDR	Software Defined Radio
SINR	Signal to Interference and Noise Ratio
SMTP	Simple Mail Transfer Protocol
SNR	Signal-to-Noise Ratio
SOA	Service-Oriented Architecture
SOI	Service-Oriented Infrastructure
TCP	Transmission Control Protocol

UCL Upper Control Limit
UDDI Universal Description, Discovery and Integration
UDP User Datagram Protocol
UE User Equipment
UTRAN Universal Terrestrial Radio Access Network
VoIP Voice-over-IP
W3C World Wide Web Consortium
WAN Wide Area Network
WLAN Wireless Local Area Network
WS-Policy Web Service Policy
XML Extensible Markup Language
XNOR Negated Exclusive OR