Computer Networks 53 (2009) 1354–1376

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Trends in the development of communication networks: Cognitive networks

Carolina Fortuna *, Mihael Mohorcic

Department of Communication Systems, Jozef Stefan Institute, Jamova 39, Ljubljana, Slovenia

ARTICLE INFO

Article history:
Received 27 March 2008
Received in revised form 13 December 2008
Accepted 13 January 2009
Available online 20 January 2009

Responsible Editor: M.C. Vuran

Keywords:
Cognitive networks
Self-aware
Cross-layer
Autonomic networks
Knowledge representation
Cognition loop
Learning
Reasoning
Protocol stack
OSI

ABSTRACT

One of the main challenges already faced by communication networks is the efficient management of increasing complexity. The recently proposed concept of cognitive network appears as a candidate that can address this issue. In this paper, we survey the existing research work on cognitive networks, as well as related and enabling techniques and technologies. We start with identifying the most recent research trends in communication networks and classifying them according to the approach taken towards the traditional layered architecture. In the analysis we focus on two related trends: cross-layer design and cognitive networks. We classify the cognitive networks related work in that mainly concerned with knowledge representation and that predominantly dealing with the cognition loop. We discuss the existing definitions of cognitive networks and, with respect to those, position our understanding of the concept. Next, we provide a summary of artificial intelligence techniques that are potentially suitable for the development of cognitive networks, and map them to the corresponding states of the cognition loop. We summarize and compare seven architectural proposals that comply with the requirements for a cognitive network. We discuss their relative merits and identify some future research challenges before we conclude with an overview of standardization efforts.

1. Introduction

The area of information and communication technologies is one of the fastest changing areas, with related services and applications having enormous and almost immediate impact on diverse aspects of the modern society, including inter-human relations, economy, education and entertainment. In this respect the development of reliable and robust yet flexible and future proof communication infrastructure capable of real-time, secure and cost effective delivery of data is of utmost importance to increase the user's perceived quality of life by facilitating human-to-human as well as human-to-machine communication almost anywhere and anytime, providing services such as e-health, e-learning and e-payments. Future networks will be ever more complex, extending towards ubiquitous communications, and will provide a broad range of other services and applications, from remote managing of an intelligent house to advanced real-time navigation systems.

In spite of the increased complexity, future networks should be easily maintainable and their capabilities should be continuously improved and upgraded by relying as little as possible on human intervention. In order to meet this demand, the networking research community proposed a new paradigm for networking: the cognitive network [1–3]. Architectures that fall under this paradigm include a cognitive process that can sense current reality, plan for the future, make a decision and act accordingly. It is generally agreed that cognitive networks have the ability to think, to learn and to remember [2,3].

* Corresponding author. Tel.: +386 14773114.
E-mail addresses: carolina.fortuna@ijs.si (C. Fortuna), miha.mohorcic@ijs.si (M. Mohorcic).

1389-1286/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.comnet.2009.01.002

The capabilities of a cognitive network can be highly distributed or extremely centralized depending on the engineering tradeoffs for each specific network. In general, a cognitive network is formed of a set of distributed cognitive entities (agents), which are somehow ''smart" in the way that they have certain reasoning capabilities and are connected in a network. In this network the cognitive entities interact with each other; they can cooperate, act selfishly or a combination of the two. While functioning in this environment, the cognitive entities are able to learn and take decisions in such way as to reach an end-to-end goal (or optimize a set of end-to-end goals). These end-to-end goals are dictated by the business and user requirements [4,3]. Developing and maintaining such a network is an extremely challenging task and has enormous potential, especially in the area of network management

A cognitive network needs to evolve over time: its set of technologies has to be updated by removing deprecated and adding new ones; its set of tools that help managing complexity should be added and removed in a plug and play fashion. Thus, the architecture of the cognitive network should be flexible and should lead to a modular and highly scalable infrastructure. Furthermore, the cognitive network must be self-aware: it should be able to know what is happening inside, what it must do; it must be able to determine appropriate actions to achieve goals and to learn while doing all these. It should be self-configuring, self-optimizing, self-healing and self-protecting in a *cognitive* way. Developing of such a network necessitates state-of-the-art knowledge and tools from various fields of science carefully engineered into a complex and efficient system.

In this paper, we analyze some recent trends in the development of communication networks and investigate in more detail the concept of cognitive networks. Cognitive networks are promising to be the major step towards efficient and autonomic management of increasing complexity of communication networks. In Section 2, we briefly discuss the current necessities from user's and network operator's point of view and provide an objective analysis of trends for the future, carried out using an ontology editor. Section 3 presents the five approaches of the research community towards developing communication networks, paying special attention to two of them that fit in the concept of cognitive networks. Section 4 provides a conceptual overview and the definition of cognitive networks as well as enabling technologies and tools, while Section 5 summarizes different architectures for cognitive networks proposed in the literature and discusses their relative merits. Current standardization activities are briefly mentioned in Section 6 while Section 7 concludes the paper.

2. Current necessities and research directions

In the history of telecommunications, development has always been driven by humans' need to communicate, i.e. reliably transmit ever increasing amount of information across increasing distances. Over time this resulted in the current landscape of telecommunications characterized by large variety of technologies that are offering various ways to connect users with other users or with application servers. Worldwide standardization efforts are enabling interoperability and integration of legacy, new and emerging technologies. However, communication networks became increasingly complex and more difficult to manage, requiring increasingly specialized tools and human operators for their maintenance, configuration and optimization.

From the user's point of view the necessities in the world of telecommunications, as it is today, are: higher bandwidth or alternative solutions (since the full bandwidth of a link is scarcely used at full capacity) capable of accommodating the traffic; QoS for the wide range of applications that fixed/mobile terminals support and services that network operators deliver; more flexibility in choosing service providers and access technologies; security; reliability; and low costs. These necessities derive from the users' thirst for digital content.

From the network operator's point of view, some of the main necessities are: complexity management; security; scalability; fault tolerance; fast integration of new technologies; and a good business model [5]. The network operator has to create adequate premises for delivering the digital content.

These user's and network operator's necessities are actually forming the basis for research activities currently underway in the area of cognitive networks, as we show in the following. In general, research directions in communications can be classified in eight broad categories: theory, signal processing, networks, software, user satisfaction, security, management, and new/next generation protocols and architectures. In an attempt to obtain an objective big picture of the trends in research areas as well as a quantitative estimation of the ongoing work, we used Ontogen, a semi-automatic ontology editor [6], to analyze the conference proceedings of IEEE Globecom 2006 and 2007, totaling 2011 papers. Fig. 1 presents a topic map which is a 2D visualization of the IEEE Globecom papers based on processing paper titles and abstracts. Papers addressing similar issues are positioned close to each other, thus forming clusters which represent the main topics approached by the papers within this particular conference. The keywords that appear on the map are the most descriptive for the cluster of papers projected in the corresponding area.

Two main clusters can be easily distinguished on top left side of Fig. 1, transversally disposed, one on the upper side of the figure and the other on the lower side. The upper cluster of papers on system and network level aspects is described by relevant keywords such as *networking, sensor, wireless, node, routes, mobile, protocol, algorithms, scheme* and *scheduling*. The relevant keywords describing the lower cluster of papers on propagation channel and radio interface aspects are*code, channels, estimates, MIMO, OFDM, receiver, frequency, systems, fading* and *antenna*. The density of the two clusters decreases towards the left where a merger between them can be also noticed (see zoomed area in Fig. 1). In this area, keywords such as *relay, cooperative* and *cognitive* can be found denoting emerging trends in wireless relay, cooperative and cognitive networks. The content of these documents is somewhere between the two main, and traditional, areas of research: low layers such as physical and data link and

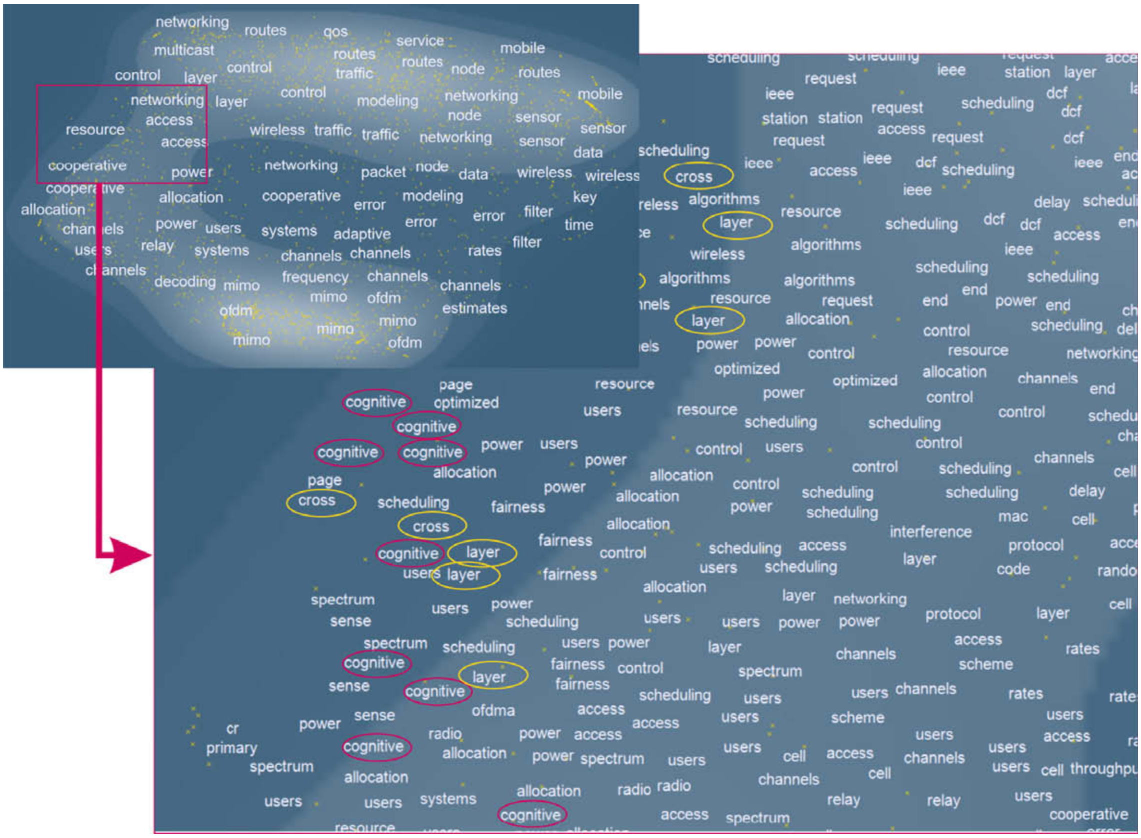

networking routes qos service mobile
multicast routes routes node routes
control layer control traffic
networking layer control modeling networking mobile
resource access node sensor sensor
access wireless traffic traffic networking sensor data
cooperative power networking packet node data wireless wireless
cooperative allocation cooperative error modeling key
allocation error error filter time
channels power users systems adaptive
users relay systems channels channels rates filter
channels decoding mimo frequency channels channels estimates
ofdm mimo ofdm mimo
mimo mimo ofdm
scheduling request ieee station layer
ieee request station station request scheduling dcf
scheduling request access request dcf
cross ieee access ieee dcf scheduling ieee
algorithms resource ieee delay scheduling
layer algorithms scheduling dcf dcf access
wireless ieee
algorithms algorithms scheduling scheduling
resource request end end power end
layer allocation control scheduling
power power control resource networking
optimized optimized allocation channels
page resource power control end
cognitive optimized users resource scheduling scheduling control
cognitive control users control
cognitive cognitive power users power channels
allocation allocation control scheduling
page power allocation control scheduling scheduling delay
cross scheduling fairness power scheduling mac cell
cross allocation interference
fairness scheduling access protocol
cognitive layer control users scheduling layer code
users layer fairness allocation
spectrum layer networking
sense users power scheduling users users power power protocol layer cell
spectrum scheduling users power layer access rates
cognitive fairness control spectrum channels scheme
sense layer fairness
cognitive scheduling users users channels rates
ofdma access scheme users
cr power sense access access users access
primary radio power access users users
cognitive allocation power spectrum users cell access channels users
spectrum cell users cell throughput
allocation radio radio channels
users users systems allocation relay relay users
cognitive access spectrum cooperative
resource

Fig. 1. Map of IEEE Globecom 2006 and 2007 papers.

high layers such as network, transport and application. Nevertheless, the keyword *cognitive* appears in the map closer to the lower cluster, meaning that the respective papers address cognitive radio aspects while the keywords *cross* and *layer* can be noticed close to both clusters.

Table 1 lists the number of papers in the cluster described by the keyword *cognitive* in the IEEE Globecom 2006 conference and in the IEEE Globecom 2007 conference and the number of papers that contain the keyword *cognitive*. This keyword typically refers to self-aware networks. The table also lists the number of papers contained in the cluster described by the keyword *cross-layer*, the number of papers that contain the keyword *cross-layer* and the total number of papers presented at the conference in each year. Numbers obtained from the ontology editor and from the word search slightly differ due to two reasons: first, the machine learning algorithm that helps building the ontology is not 100% accurate; and second, if a word is contained within a paper title or abstract, it does not necessarily mean that the paper investigates that particular subject. Finally, there are two important conclusions that can be derived from the numbers presented in the table: first, cognitive networks, including cognitive radios and cross-layer design, are still in their infancy; and second, these two areas are growing. In the following we will have a closer look at these two research areas.

3. Research directions towards future development

With respect to the reference layered architecture of the networks, we can identify five approaches of the research community towards future development.

First, a considerable number of researchers incrementally improve existing techniques and algorithms. This is a traditional, focused approach that is trying to optimize local goals and behaviors and is a necessary part of

Table 1
Cognitive and cross-layer papers.

	Conference	Cognitive	Cross-layer	Total number of papers
Ontology	Globecom 2006	~18	~39	1008
	Globecom 2007	~57	~96	1003
Word search	Globecom 2006	~15	~66	1008
	Globecom 2007	~56	~84	1003

research. Nevertheless, Clark et al. argue in [1] that improving on techniques and algorithms is not sufficient to meet the necessities of today.

The second approach that can be identified in research work is the creation of new or improved protocols. Obviously, this approach is also necessary in the area to keep up with new technologies (such as wireless), new realities (such as insufficient IPv4 addressing), etc.

The third research direction, less prolific than the others, investigates fundamental architectural changes by creating new abstractions as observed in [7] or by mathematically modeling the current architecture in a series of papers by Chiang et al. [8–10]. Chiang et al. notice that network protocols in a layered architecture have been developed on an ad-hoc basis, solving communication on a specific layer. Most of these protocols can be modeled in a mathematical way and future protocols and architectures can be holistically analyzed and systematically designed as distributed solutions to some global optimization problem [10].

The fourth and fifth approaches are relatively new and refer to breaking the layering and bringing cognition to networking. Although breaking the layering was researched and surveyed already, we briefly summarize it a as it serves as the basis for bringing cognition to networking, which is the main focus of this survey paper.

3.1. Breaking the layering

As mentioned above, the fourth and relatively novel research direction that can be noticed in communications refers to breaking the traditional layered structure of communication architectures and is often called cross-layer design and optimization. There are two basic cross-layer design approaches as identified in [11]: implicit cross-layer design and explicit cross-layer design. In the following, we will have a look at these two approaches to cross-layer design, see design proposals for these basic approaches as well as some benefits and disadvantages.

3.1.1. Implicit cross-layer design

With implicit cross-layer design there is no explicit violation of the reference layered architecture, in the way that no layers are merged and no new interfaces are created. The violation of the reference architecture occurs implicitly during the design stage. With this approach, a protocol on a specific layer (the designed layer) is designed taking into account the processing performed at a certain fixed layer as shown in [7,11] and Fig. 2a. While this approach can optimize one or perhaps more goals, it limits network flexibility and upgradeability. By employing such a design, one network layer cannot be changed without performing the corresponding changes to the second one [7]. Even more, if several layers are designed in such way, it is challenging if not impossible to estimate whether or not the overall system will behave in the desired manner, given the large number of variables its behavior depends on.

3.1.2. Explicit cross-layer design

Explicit cross-layer design refers to an explicit violation of the reference layered architecture in that it leads to merging and/or splitting of layers, creation of new interfaces and/or creation of new layers. Three design trends to explicit cross-layer design can be identified: merging/splitting adjacent layers, uni/bi-directional communication between (non-)adjacent layers and vertical calibration.

3.1.2.1. Merging/splitting layers. Layered protocol stacks have evolved over time, stacks have been designed for fixed voice centric and then data centric networks, for high capacity core networks and for wireless networks. The traditional evolution trend, especially in core networks, has typically consisted of collapsing the layers as shown in [12]. In multiple access networks, driven by the increasing demand for bandwidth and reliability, the Data Link Layer (DLL) of the OSI was split into two sub-layers, Media Access Control (MAC) and Logical Link Control (LLC) [13, p. 228]. MAC is closely related to the physical (PHY) layer while the LLC acts as mediation between the MAC and the network layer technologies. In wireless networks, PHY and MAC layers tend to merge into a super-layer to increase spectral efficiency and reliability. The idea of splitting a layer into sub-layers is illustrated in Fig. 2b and merging adjacent layers into a super-layer is conceptually illustrated in Fig. 2c. By splitting a layer into sub-layers, the outer interfaces are preserved and new interfaces are created between the new sub-layers. A super-layer that has been created from merging layers keeps the outer interfaces with the neighboring non-merged layers and it does not create new interfaces in the stack; it uses services provided by the lower layer and provides services via primitives to the upper layers. Typically, these types of architectures optimize for a single goal and when the goal of the network changes, such architecture has difficulties adapting to the changes [14].

3.1.2.2. Interactions between layers. The second trend in cross-layer design emerged from the need to improve the performance of wireless networks. It investigates inter-layer interaction between protocols as depicted in Fig. 2d. Unlike the previous trend, this one proposes interactions (uni/bi-directional communication or sharing the internal information) between protocols residing on several (non-)adjacent layers of the protocol stack, thus creating new interfaces. Kawadia et al. argue in [15] that good architectural designs lead to proliferation and longevity, an example of such architecture being the layered Internet architecture. The success of the Internet architecture is widely based on its modularity: between each two layers there is an interface that defines services that the lower layer has to deliver to the upper layer, and peer layers residing on different devices communicate using protocols that define the message formats. Protocols are used to implement the services they are required to offer to the upper layers and they can be upgraded or replaced as long as the services to the upper layer are being provided. This design decoupled services from protocols [13] and opened the road for scalability and flexibility. Adding interfaces between layers without having a set of rules to do so can render the architecture meaningless, unguided cross-layer optimization can lead to unintended performance degradation as shown in [15] and scalability will be seriously

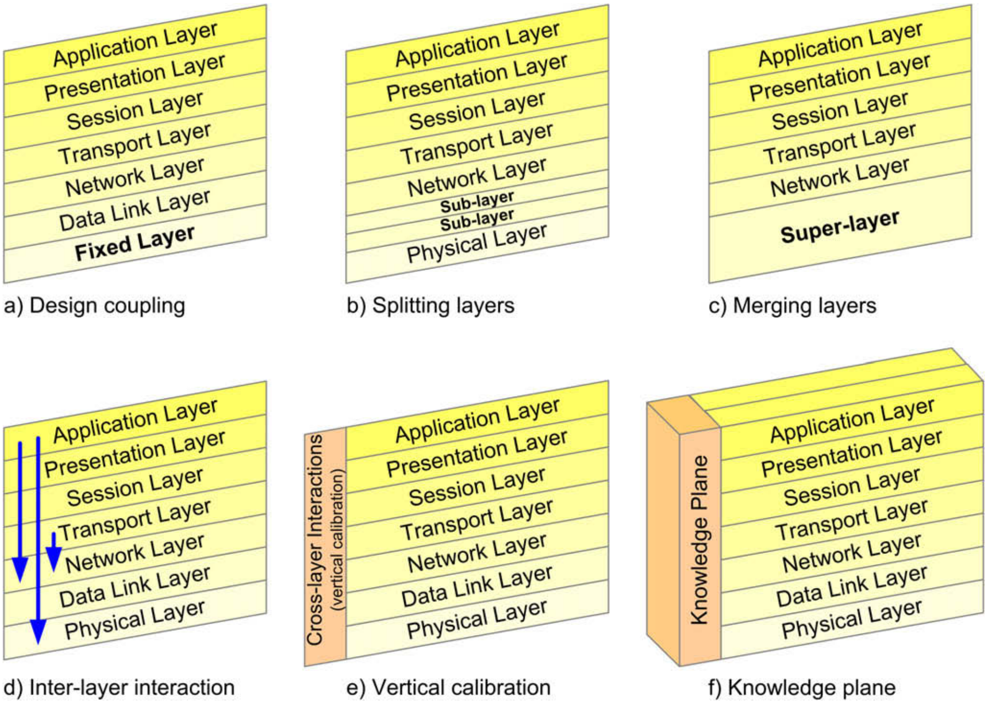


Fig. 2. Trends in breaking the layered architecture.

affected. Conflicts between independent adaptations at different layers can lead to adaptation loops [7,2].

3.1.2.3. Vertical calibration. In order to avoid the risk of "interface creep" [2], certain cross-layer designs use a different paradigm to avoid direct bidirectional communication between (non-)adjacent layers. This trend in breaking the layering uses a common vertical plane that spans across all layers and that can be used by each layer if it chooses to. This concept, illustrated in Fig. 2e, is called vertical calibration [7,2] and it does not seem to limit scalability and flexibility as other cross-layer design proposals do. According to Srivastava et al. [7], this approach performs joint optimizations at all the layers of the stack to satisfy an application level utility. The layers of the protocol stack are interfaced with a shared database, which acts as a new layer and provides structured information storage/retrieval services as well as controlling and sensing the status of layers [2]. Carneiro et al. describe in [14] a cross-layer manager for wireless networks that seems to have similar characteristics as the vertical calibration plane. The cross-layer manager interfaces the layers of the 4G wireless protocol stack, getting event notifications from the layers and exposing state variables to them. This manager is capable of handling heterogeneous wireless networks as well.

Cross-layer designs are algorithmic approaches to solve shortcomings of the traditional protocol layering. Every cross-layer design proposal serves to highlight a specific shortcoming of the traditional protocol layering [16]. These designs are protocol stack centric [4] in the way that they solve protocol stack specific problems, not end-to-end network goals so they do not have significant impact on solving heterogeneous network specific problems. Another shortcoming of these designs derives from the fact that they are based on algorithmic approaches. A system designed on such an approach does not learn and adapt according to an end-to-end communication goal, but preserves a rigid structure and repeats ill functioning adaptations that previously lead to poor performance [2].

3.2. Bringing cognition to communication networks

Internet of today is a passive network that relies heavily on human intervention: data is exchanged between the edge nodes which hold the intelligence of the network. The core network transports the data without having much knowledge about it. In case of some malfunction that prevents the data from being delivered, the edge can recognize that there is a problem, but the core cannot determine what the problem is about, not to mention solving it. In such case, a human operator is required to intervene and fix the problems. In most circumstances, it is not even possible for the human to specify in high-level terms (such as natural language) the solution to the malfunction and the network to solve it (by self-configuration). Instead, the operator has to interact with the core network using device specific configurations to resume functioning.

3.2.1. The Knowledge Plane (KP)

Clark et al. were the first to propose a new kind of network [1] which is aware of itself and its surroundings, thus a self-aware network, able to learn, decide and act according to those decisions to reach high-level goals, thus a network which employs cognition. Such network should, in the first phase, be able to recognize malfunctions and explain them in a (perhaps restricted) natural language. Then it should suggest ways of solving the malfunction and, finally, it should fix the problems itself. In other words, the network becomes self-aware and, as a consequence, it becomes able to self-configure, self-adapt, self-heal and self-manage. The network should be able to perform all this controlled by the *knowledge plane* (KP), an entity that spans over layers and devices as depicted in Fig. 2f, therefore having local as well as global knowledge of the network and its elements.

According to Clark et al. [1], the KP "*is a pervasive system within the network that builds and maintains high-level models of what the network is supposed to do, in order to provide services and advice to other elements of the network*" while heavily relying on tools from artificial intelligence (AI) and cognitive systems.

The definition of the KP and its attributes makes it dependent on cross-layer approach. Take as an example a device situated at the edge of the network, connected via a wireless access technology and running several applications. PHY/MAC cross-layer solutions are engaged at getting the most out of the wireless channel. Even more, these layers can communicate with higher layers, for instance to adapt the video encoding to the state of the propagation channel. While this type of local selfish cross-layer optimization might prevent complete service interruption, it is not able to ensure good quality of experience for the user. Now assume there is a learning module inside this device with extra knowledge on the user's habits or moving patterns. This module could control service delivery in such way as to deliver highly desired services at the highest possible quality, while postponing others until better conditions are met. Going a step further, a connection manager could understand the capabilities of incompatible access technologies in the area and, given application requirements and previous knowledge, request services from these technologies to meet high-level goals. The knowledge about the access segment, adaptation and reconfiguration attempts and/or application knowledge could be send to the access network which would be able to cluster it, integrate it with extra knowledge it possesses, determine actions, take complex decisions and perform reconfigurations.

With this example we show that the *edge involvement* of the KP makes use of cross-layer information that can be gathered and processed on the device to update the knowledge about the "self" and the environment it functions in. The KP can use this knowledge to perform cognitive changes (cross-layer adaptations and reconfigurations) and/or share relevant part of this knowledge with other nodes sharing the KP. Thus, the *compositional structure* of the KP ensures it can reach a *global perspective*. The functioning and enabling techniques of the KP are modeled by the *cognitive framework*[1].

3.2.2. Making networks aware: knowledge representation and reasoning

In [17], Mitola claims that "to be termed 'cognitive', a radio must be self-aware". By self-aware he refers to a *computationally accessible* description of its own structure. Extending this beyond the radio domain, it can be said that for a network to be cognitive, it must be self-aware [3, p. xix], that means it should have knowledge about itself, its building blocks and their interconnection, it should be able to share this knowledge and reason based on this. In [18], context awareness is seen as prerequisite for achieving cognition. By context awareness, the authors refer to awareness about the "self" and about the "world" and define it as the "information that surrounds and gives semantic meaning to an entity".

Alternative views and definitions of context and context awareness can be found in [19]. However, defining what exactly is "self", "world" and "context" depends on the domain of application, the design choices, etc. and is beyond the scope of this paper.

In summary, knowledge empowered networks seem to be the only way towards reaching the point where a network can configure itself, explain itself and repair itself [1]. But this raises a fundamental question: "How to represent this knowledge and how to reason about it?". In [20], the authors argue that a knowledge representation is most fundamentally "a surrogate, a substitute for the thing itself", thus a *model* [21, p. 48]. This model, the reasoning based on it and updating it can be implemented in several ways, depending on the focus (i.e. on what is wanted to be captured). Thus tables, Boolean circuits, neural networks, artificial languages, etc. can all be viewed as forms of knowledge representation. According to Davis et al. [20], intelligent reasoning and corresponding knowledge representation can be classified into five categories according to their origins: mathematical logic, psychology, biology, statistics and economics.

3.2.2.1. Knowledge representation for management. The work in [22] is an illustrative example for two types of structured knowledge representations for telecommunication network management: ontological and Bayesian network representations. They use ontology to represent domain specific knowledge and knowledge about Bayesian networks. This conceptual knowledge enables interoperability and data integration but also allows machine based reasoning through inference. The domain ontology is then used to construct Bayesian networks in an automatic fashion. The obtained Bayesian network is a probabilistic model of the telecommunications network and is used for fault management.

In [5] it is argued that self-aware entities need ontological models that encode domain specific knowledge. In building their system (called FOCALE, described in Section 5) they use information models and ontologies for modeling the managed elements of a network. The information models are high-level, generic models designed to be able to model all managed elements. Vendor specific facts, encoded as Object Management Group (OMG) compliant data models, are derived from these information models to describe the managed devices produced by a specific vendor. Each data model has one or more ontologies associated with it. These ontologies encode definitions, relationships and semantics, things that cannot be encoded by information or data modeling languages [23]. With this approach, the FOCALE system is able to reason about the facts represented in the data models by using Web Ontology Language (OWL) ontologies of the meta-models associated with the respective data models [23]. The system supports description logic (DL) reasoners and querying with SPARQL and other query languages for OWL (see [23]). This complex approach involving both semantic and non-semantic technologies for modeling a system described as self-governing seems to have been taken for two reasons. The first is to ease sharing of information residing in private knowledge bases and the second is to keep backward compatibility with existing standards.

An approach for modeling infrastructure wireless networks with the scope of resource management can be found in [24]. They use Bayesian networks to model the capability of a node according to a corresponding configuration. The model encodes information about the coverage area and average bit rate that can be achieved in that area. The approach in [25] is similar as both of them are coming out of E2R project [26].

The case study in [4] demonstrates the use of learning automata for wireless resource management. The learning automaton holds a model of the environment. This model is created using delayed rewards based on the reaction of the environment to previous actions [27, p. 8]. The environment is unknown and random (e.g. wireless path) and learning automata seem fit for modeling such unreliable environment.

The next example of utilization of ontological knowledge representation are the Knowledge-Based Networks (KBNs) [28,29]. KBNs are semantically enhanced content based networks in which content publishers use ontological representations of their information while subscribers use semantic queries to find relevant information. In [29], heterogeneous ontologies from different publishers are semantically inter-mapped to enhance querying. This technology is applied to the problem of exchanging policies related to dynamic spectrum access in a self-managed network. In this context, publishers can be seen as entities that are self-aware of their capabilities and advertise them. On the other hand, subscribers are aware of their needs and query for resources able to satisfy them.

3.2.2.2. Knowledge representation for quality of service. From several papers modeling QoS related aspects, we were able to identify two areas of application for semantic QoS modeling. First one is horizontal QoS provisioning which is concerned with finding, matching, providing and monitoring QoS based services between two parties [30–32]. The second one is vertical QoS provisioning in which application QoS requirements are semantically translated into lower level requirements (platform dependent or platform independent) [33]. However, in [34,35] a system that encompasses both approaches is described.

The work in [34] uses multilayer neural networks (MNN) to model and estimate user preferences with respect to wireless services. The MNN are trained with user feedback data to build a model of the user's perception of past provision of wireless services. Based on this model, an agent is able to estimate the satisfaction of the user with respect to a new, previously unseen service. The goal is to choose a wireless service that would please the user without requiring extensive interaction [35].

In [30], a QoS ontology (DAML-QoS) modeling QoS profiles, properties and metrics is defined. A matchmaking algorithm determines the published services that semantically match the ones inquired. A measurement system makes sure that the service agreement is respected. This ontology has been ported to OWL [31] and apparently work for constructing a QoS ontology that can be submitted for standardization is underway [32]. The approach in [33] seems suitable for achieving semantic translation between platform independent and platform dependent QoS requirements. However, the paper does not go that low along the OSI stack with the translation as ''low level metrics are not easily understandable and applicable for application developers".

In [37], self-awareness is achieved through on-line self monitoring and measurements with the final scope of offering QoS to users. The essence of the work is to validate a test bed for QoS routing. At each node (router), the network (all possible outgoing routes) is modeled using random neural networks (RNN). Smart packets (SPs) are sent to discover new routes in the network and acknowledgement packets (ACK) are used to bring back information discovered by the SPs. The network model (i.e. the weights of the RNN) is updated with the new information. This model tells the core what actions to take, thus eliminating one of the current shortcomings of core networks as stated as previously explained.

3.2.2.3. Knowledge representation for security. The work in [38] focuses on security semantics and tests the prioritization of types of content and flow over an enterprise network. It uses semantic technology to enable fine grained, highly specialized services for policy based networking. A packet sent through the network has an OWL semantic tag which is used by the policy decision points encountered on the way for reasoning and inferring the set of operations to perform. This approach, in which knowledge is encoded into packets, allows the core to be aware of what data it is carrying and what the purpose of this data is as called for in [1].

In [39], security domain knowledge was manually encoded in a special purpose unit of the Cyc knowledge base (KB). The KB is periodically updated with information gathered by sentinels placed on client machines. In essence, the CycSecure application, takes the information pulled from the sentinels, transforms it into CycL (Cyc internal representation language), and builds and maintains a model of the network based on the pre-existing ontologies of networking, security and computing concepts. This system is able to explain in natural language what the vulnerabilities of the network are and allows high-level input from the user.

An ontology modeling network attacks has been created in [40] using Resource Description Framework Schema (RDFS) formalism and Prolog for reasoning. The authors further illustrate the benefits of using ontology in this particular context. Based on the ontology, they update the KB to always have an accurate model of the state of the network with respect to attacks. The work in [41] focuses on a specific security problem, namely distributed denial of service attacks (DDoS). As an intermediate step towards modeling DDoS attacks, the authors built an ontology of such attacks.

3.2.2.4. Discussion. In this section, we showed that networks of the future will have to be based on knowledge, we briefly addressed the problem of knowledge representation and summarized relevant work related to knowledge representation and reasoning for networks. Table 2 synthesizes this research work and its main features. The first four rows of the table present research for knowledge representation and reasoning for management purposes, the next four rows for QoS while the last three for security.

The columns of the table list key-features of research on knowledge representation and reasoning for networks. The first column contains the name of the project, of the prototype or just an intuitive name for the research work. The second column lists the scope for which knowledge representation and reasoning are used. The third column refers to the type of model used for representing the thing under investigation. The fourth column lists the representation language, where the case, and the fifth column lists the type of the representation (symbolic, where an artificial language is used, and numeric otherwise). The sixth column notes whether the model is updated or not. Finally, the last column lists the reasoning engine used in the experiments.

As can be seen from the third column, ontology is the most used form of knowledge representation in the surveyed papers. An ontological representation is a structured knowledge representation using concepts such as actions, time and physical objects [21]. Domain specific (management, QoS and security) ontologies were built and in some studies they were extended [39] or were joined [22,30] with other domain ontologies as the nature of the problem being solved required knowledge about other areas as well.

An OWL [42] approach was mainly used as the language for ontology representation. Resource Description Framework (RDF) [43], RDF Schema (RDFS), DARPA Agent Markup Language (DAML) [44] and CycL are other representation languages that were used. OWL, RDF and RDFS are all standardized by the World Wide Web Consortium (W3C), DAML extends RDF and RDFS while CycL is a proprietary language. All these languages allow representing semantic knowledge and are used in the semantic web [45]. Unified Modeling Language (UML), Managed Object Format (MOF), Object Constraint Language (OCL), and Query/View/Transformation (QVT) are all Object Management Group (OMG) [46] specifications used for network management but unable to provide semantics [5]. The lack of semantics in those representations is the motivation for building OWL meta-models that empower semantic translations for automatic translations for autonomic management [23].

Ontological models make use of reasoning engines to infer new knowledge from the one encoded in the model. Jena [47], Pellet [48], FaCT [49], RacerPro [50] (see Table 2) are all description logic reasoners based on tableau calculus. Jess [51] is a rule-based engine, Cyc [52] has various modules for different types of reasoning while Prolog is a well known logic programming language.

It can be seen in Table 2 that besides ontological models, also neural networks, Bayesian networks and learning automata are used for modeling management and QoS related aspects. These techniques are best suited to model uncertain knowledge and perform reasoning under incomplete information [21, p. 462]. Even though several papers that address cognition related issues for communication networks [1,17,18] strongly argue for the use of conceptual knowledge encoding (i.e. ontology), most likely that will not suffice, if for no other reason then for the complexity and high resource consumption of this technology. Reasoning for this kind of knowledge representation is based on

Table 2
Approaches to network related knowledge representation and reasoning.

Research	Scope	Model type	Repres. language	Repres. type	Model update	Reasoning engine
Node management [22]	Network management	Ontology + Bayesian network	OWL	Symbolic and numeric	Yes	Jena
FOCALE [3, p. 23,23]	Backbone network management	Information and data models + ontologies	UML, MOF, OCL, QVT + OWL	Symbolic	?	RacerPro, Pellet, FaCT
Node management [24,26,25]	Wireless infrastructure management	Bayesian network	–	Numeric	Yes	Bayesian network
Cognitive network [4]	Wireless resource management	Learning automaton	–	Numeric	Yes	Learning automaton
Knowledge-based networks (KBN) [28,29]	Semantic publish-subscribe networks	Ontologies	RDF, OWL	Symbolic	Yes	RacerPro, Jena, Pellet
Cognitive agent for the personal router [34]	Wireless service selection	Multilayer neural network	–	Numeric	Yes	Multilayer neural network
DAML-QoS [30], OWL-QoS [31]	Semantic QoS matchmaking and measure	Ontologies	DAML + OIL, OWL	Symbolic	No	Jena
Cognitive packet networks (CPN) [37]	QoS aware routing	Random neural network	–	Numeric	Yes	Random neural network
Semantic tags [38]	Semantics for secure enterprise networking	Ontology	OWL	Symbolic	No	Jess
CycSecure [39]	Network risk assessment	Ontology	CycL	Symbolic	Yes	Cyc
Ontology for intrusion detection [40]	Modeling computer attacks	Ontology	RDFS	Symbolic	No	Prolog

logic, but logic cannot always solve the problems. When only partial knowledge or unreliable knowledge is available, then technologies modeling uncertainty are required [21, p. 462]. In this sense symbolic and numeric knowledge representations should both be used to complement each other.

Typically, with a good prior model, less training data is required to obtain the posteriori model than with a less good prior model. An illustrative example in [22] uses Bayesian networks (numeric) for modeling the node (probabilistic model with incomplete knowledge), but the construction of this model makes use of prior domain expert knowledge encoded in an ontology (logic model with domain knowledge). There is also a way to combine probability theory, which is able to handle uncertainty, with deductive logic, which is able to exploit structure, resulting in probabilistic logic [53].

In summary, the form (or combination of forms) of representation used it is not important provided it is adequate for the task. Suggestions for approaches to knowledge representations for computer networks (e.g. [18]) and sources of knowledge representation techniques in AI (e.g. [21]) are widely available. Also, as discussed in this section, there exist several approaches towards knowledge-based networks. However, most seem to be describing architectures and frameworks, presenting initial approaches to build knowledge representations for computer networks. Very few present actual simulation results and even fewer show performance comparisons; in fact, we were not able to find any representative for this last category.

Another issue with modeling is whether the models are updated or not. It can be seen in Table 2 that ontology based models are typically not updated or if they are, this happens in a manual or semi-automatic way. As opposed to this, the models which use numeric representation are all updated automatically. Evaluations are required on how reliable the ontology based models are in the context of communication networks. Are they actually fit for the use inside the cognition loop, or are they more useful in interpreting and preparing relevant data to be presented to the user as done in [39]. Proving the utility of these models for critical 99.999% systems is still an open challenge.

3.2.3. The cognition loop

All systems that are able to adjust their functioning according to changes in their environment are based on feedback information. Cognitive networks are no exception in this respect, so they will also use a control loop, also called cognition cycle [17, p. 47], feedback loop [4], context based adaptation loop [18]. According to Thomas et al. [4], the loop employed by a cognitive network should be based on the concept of the Observe-Orient-Decide-Act loop originally used in the military, augmented by learning and following end-to-end goals to achieve cognition. In [18], the loop also has a communicating capability for communicating with other loops in a distributed environment.

The cognition cycle as described by Mitola [17, p. 48] features the following states: observe, orient, plan, decide, act and learn. It uses the orient module for classifying stimuli and does not explicitly encompass policies. In [18], a taxonomy of cognitive capabilities can be found. They introduce a context based adaptation loop having five high-level states. The first state is Sense which is presented as having the following components: context gathering, measure, monitor, pre-process and approximate. The second state is Analyze and is responsible with interpretation and abstraction, consistency check, prediction, data fusion, reasoning, model management and update, and learning. The third state is the Decide state that is formed of the following components: select action, prune solution space, apply policies and constraints, evaluate alternatives, optimize and make the decision. The Reconfigure state is the fourth and it implements decisions, selects components, downloads components and sends alarms and warnings. The last state is the Communicate state and this is responsible with sending and receiving commands and data between cognitive engines distributed around the network.

Partially inspired by the approaches mentioned above, we define a reference cognition loop as depicted in Fig. 3, consisting of six states (sense, plan, decide, act, learn, policy). In our view, the self-aware network will employ sensors to sense the environment (Sense). The observations captured by the sensors will be further used for planning (Plan), but they will also be fed to a learning module able to learn and remember (build a model from the) useful observations (Learn), which can aid the decision making module in the future (Decide). The planning module determines potential actions, i.e. strategies to be followed based on observations and policies stored in the policy module (Policy). The decision module decides on the actions to be taken based on possible moves (actions) and (learned) experience. Finally, the actuators (Act) are responsible with executing the adequate changes (reconfigurations). The learning module is the one best connected in the sense that it can learn from several sources: from sensor data, from strategies, from decisions and from actuators, and can correlate and infer from this knowledge.

According to Mitola [17, p. 48], the described cognition cycle would occur under normal operating conditions. In "under stress" operating conditions, such as the ones where a power failure occurs, some modules can be bypassed. This is done by assigning priorities to the sensor output such that a stimulus classified as "Urgent" would trigger actions immediately.

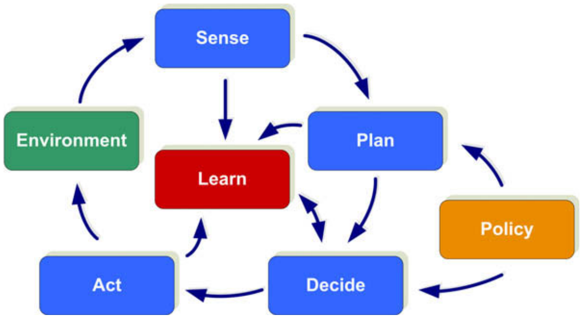


Fig. 3. The cognition loop.

3.2.3.1. Loops for management. In [34], the authors investigate a cognitive agent for wireless network selection which is designed to hide the complexity of the wireless environment from the user. The selection problem is decomposed into four elements that enhance the agent to select the network which is most suitable to user preferences. First, user's feedback that the decision making process will be used is captured. Second, the available services are evaluated against learned user preferences. Third, the agent decides when to change services and which new service to select based on user's preferences, context and goals. Fourth, the value of previously unseen services is predicted. Using this approach, the agent continuously monitors the wireless environment and selects the best service according to the current model of user preferences. However, when the user is unsatisfied (or changes preferences), the model is updated and a new selection is made to satisfy preferences.

In [4], a case study of a cognitive network is considered. The work addresses distributed wireless resource management issues with the scope of maximizing connection lifetime. Each node of the network employs its own cognition loop. The cognition loop gets sensor information from a directional reception antenna and controls an omnidirectional transmission antenna that has the role of the actuator. The plan, decide and learn modules of the cognition loops are implemented by a learning automaton [27, p. 8]. With learning automaton, the plan module of the loop allows different available alternatives (e.g. received power levels). The decision module selects one of the alternatives based on the reaction of the environment to one of the previous actions. These reactions are learned, but they are not completely reliable because the responses from the environment are random.

The node management in wireless infrastructure [24] has two end-to-end goals: maximize the QoS levels allocated to users, and minimize the number of reconfigurations. They also use a feedback loop for solving the problem. In the monitoring phase the cognitive node gathers information about the number of users, their mobility and traffic. The planning and decision phases use Bayesian networks and a four phase strategy for choosing the best reconfiguration. They make use of element profiles and network operator policies to determine possible reconfigurations. In case of a well performing reconfiguration, the probability of choosing it in the future is augmented by reinforcement learning.

Other related cognition loops can be found in [5,18,54–57]. These are mainly high-level views of cognition loops used for various types of management tasks.

The end-to-end goal in [5] is to perform device configuration according to high-level (restricted natural language) policies. The described architecture configures, monitors and reconfigures when necessary a network of managed elements (e.g. routers). The managed element is subject to monitoring in order to assess its local performance. Reasoning and learned facts are used to analyze data and events specific to the device to determine the actual operational state. The autonomic manager checks if this state complies with policies and triggers the definition of new configurations if necessary.

In [18], the architecture for a cognitive node is introduced. This node uses a cognitive process, namely a Sense–Analyze–Decide–Reconfigure loop, specified using process algebra notations. In [54], a cognition cycle is used for dynamically building and adapting a node's protocol stack. A stack manager gets as inputs the node's current configuration and a set of available components. Based on these inputs and with the help of the cognition cycle it outputs a new network protocol stack.

A Cognitive Resource Manager (CRM) and its conceptual architecture are introduced in [55]. The CRM's functioning is based on a cognition cycle adapted from Mitola [17] and aims at enabling autonomic optimization of the communication stack as a whole, thus acting as an intelligent vertical calibration (Fig. 2e). The intelligence would be based upon methods from the field of AI.

The cognitive resource manager in [56] uses multi-agent systems in non-cooperative nodes. The case study employs multi-agent constraint optimization to optimize inter WLAN handovers with the end scope of load balancing. The work in [57] introduces an extended cognitive cycle for joint radio resource management and computing resource management. Actually the extended cognitive cycle consists of two loops, one responsible for the radio environment while the second is responsible for the computing environment. They do not provide information about AI techniques used in the modules of the cycle.

3.2.3.2. Loop for overlays. The scope of the CogNet project [58] is to build an experimental cognitive radio network for testing cognitive techniques. It targets the first three layers of OSI. In [59], they evaluate a multi-overlay network layer, based on the Host Identity Protocol (HIP) operating on top of IP, with respect to the delay introduced by the addition of HIP and security. However, they envisage that the multi-overlay network layer can adapt based on sensing and learning and that the usage of a particular overlay will be decided based on reasoning and learning.

3.2.3.3. Loop for security. The CycSecure application [39] makes use of an incomplete cognitive loop. It uses daemons installed on machines in the network that collect local information and send it to the server when polled. A human operator can examine and modify the network model, query and view network statistics. The system is able to generate possible attack plans based on the information gathered from the system and the internal knowledge base. Based on these attack plans, the human operator can decide for remedy measures to increase the security of the system.

3.2.3.4. Discussion. In this section, we have showed that a loop or cycle having suitable states will empower the cognitive engine. Such systems will be able to behave in a reactive way but, even more important, they are also expected to be able to behave pro-actively through learning and planning. It can be seen that there are quite some high-level descriptions of possible cognition loops but not so many implementations, trials and validations. Most of them focus on management issues (as foreseen in [17, p. 57]), one focuses on overlays and one on security. Practical results are expected to be shown by recently financed mainstream projects [60] employing cognitive loops in their prototype systems.

Table 3 synthesizes relevant work implementing cognitive loops. The first three rows list work for management purposes while the last one focuses on security. The columns of the table list the technologies used for the states of the loop. Compared with Table 2, this table has less entries. This is also due to the fact that the symbolic models based on ontologies from Table 2 were not included in cognition loops, at least we were not able to find published work on this topic. The only exception to this is [39] but they do not use a full loop, as learning is missing and they also use a human operator to close the loop. Otherwise, it is worth noting that numerical methods of dealing with uncertainty are used for the Plan and Decide state of Refs. [34,35,4,24]. For the learning state, reinforcement learning seems to be the most popular approach.

We consider that with software going ever lower in the implementation of protocol stacks, cognition loop based systems will gain increasing interest as an alternative to management of complex systems. This research area is widely unexplored so far mainly due to physical constraints that are now beginning to diminish.

4. Cognitive networks

In this section, we analyze several existing definitions for cognitive networks, and we argue that two elements are essential for developing a cognitive network (CN): the knowledge representation and the cognition loop. We also discuss the differences and similarities between cognitive radio, cognitive network and cognitive radio network. Next, we discuss the framework proposed in [4] for introducing cognition to communication networks. The main part of the section focuses on methods from AI that seem applicable for developing CNs. We provide a summary of several types of intelligent agents (IAs), map them to the functional states of the cognitive loop. As we go along, we also refer to existing research on CNs which makes use of the respective type of IA, where available.

4.1. Terminology

4.1.1. How it started

The word *cognitive* refers to an entity that is able to perform some kind of conscious intellectual activity such as thinking, reasoning, learning or remembering in order to make sense of its surroundings. This word was first used in communication networks to refer to a technology by Mitola as he introduced the *cognitive radio* [17]. By cognitive he meant "the mix of declarative and procedural knowledge in a self-aware learning system" which would give the radio (by radio he referred to PHY and MAC layers in OSI reference model) a computationally accessible description of its structure and intelligence to operate a software radio.

It is important to note that he introduced the concept of cognitive radio (CR) at the time of early work on software defined radio (SDR) which represents a shift in approach from the traditionally hardware defined radio. Very briefly, what Mitola proposed was a software defined radio stack (PHY and MAC) which would be self-aware in the sense that it would hold a model based semantic representation of its structure and the environment (see radio knowledge representation language in [17]). This stack would use a cognition cycle to interact with the environment, learn and adjust its functioning in a cognitive fashion. This confirms the discussion in Section 3 that a radio to be cognitive it must feature a semantic representation of knowledge and a cognition cycle, thus incorporating technologies from the field of AI.

4.1.2. How it evolved

The original understanding of CR, however changed as the FCC defined it as "a radio that can change its transmitter parameters based on interaction with the environment in which it operates" [61]. It added "This interaction may involve active negotiations with other spectrum users and/or passive sensing and decision making (smart radio) within the radio. The majority of CRs will probably be SDRs, but a CR does not necessarily use software, nor does it need to be field programmable." This sounds very much

Table 3
Approaches to the implementation of cognition loops.

Research	Sense	Plan	Decide	Learn	Policy	Act
Personal router [34,35]	Personal router (wireless services, user feedback)	Multi-layer neural network in [34], Markov decision process in [35]		Reinforcement learning	User preferences	Personal router (wireless service)
Cognitive network [4]	Directional reception antenna (power)	Learning reward-interaction automaton			–	Omnidirectional transmission antenna (power)
Node management [24]	Reconfigurable node (users, applications, mobility)	Bayesian network (bit rate, coverage)		Reinforcement learning	Element profiles (transceiver, RAT, spectrum, user classes, application, terminals), network operator policies	Reconfigurable node (change RAT, spectrum)
CycSecure [39]	Daemon	Simple hierarchical ordered planner	Cyc	–	–	User

like reconfigurable radio (in the sense of parameter reconfigurations), dynamic spectrum access and/or open spectrum access, and represents a significant drift from the CR concept as defined by Mitola. The survey in [62] actually shows that the largest part of the research community sees CR in the sense of FCC as opposed to the definition of Mitola.

We would like to emphasize that, according to the dictionary [64], the word *cognitive* used as an adjective to a noun means:

- of, relating to, being, or involving conscious intellectual activity (as thinking, reasoning, or remembering);
- based on or capable of being reduced to empirical factual knowledge.

Thus, a CR should be based on knowledge acquired either empirically, or through learning (remembering). Additionally it can make use of thinking and reasoning, therefore the semantic analysis of the notion leads to Mitola's definition. In the remainder of this paper, when we refer to CR, we mean CR according to Mitola's definition.

4.1.3. How it continues

The notion of *Cognitive Network* was defined in [4] and it seems to have been inspired by the KP described in [1] as a "*distributed cognitive system that permeates the network*". In [4], the authors define the CN as a network with a cognitive process that can perceive current network conditions, plan, decide, act on those conditions, learn from the consequences of its actions, all while following *end-to-end* goals. This loop, the cognition loop, senses the environment, plans actions according to input from sensors and network policies, decides which scenario fits best its end-to-end purpose using a reasoning engine, and finally acts on the chosen scenario as discussed in the previous section. The system learns from the past (situations, plans, decisions, actions) and uses this knowledge to improve the decisions in the future.

This definition of CN does not explicitly mention the knowledge of the network; it only describes the cognitive loop and adds *end-to-end* goals that would distinguish it from CR or so called cognitive layers [4]. We consider this definition of CN incomplete since it lacks knowledge which is an important component of a cognitive system as discussed so far in this paper and also in [1,3,17,18].

In particular, Balamuralidhar and Prasad [18] gives an interesting view of the role of ontological knowledge representation: "The persistent nature of this ontology enables proactiveness and robustness to 'ignorable events' while the unitary nature enables end-to-end adaptations." We consider this statement essential for CNs. The authors of [18] also state that the "core persistent and unitary ontology is what will distinguish cognitive systems from merely adaptive systems which may also need ontology to function but will not have such a notion of 'self'."

In this survey, we consider that a CN is a communication network augmented by a KP [1]. This KP can span vertically over layers (thus making use of cross-layer design, as discussed in Section 3) and/or horizontally across technologies and nodes (thus covering a heterogeneous environment) as depicted in Fig. 2f. The KP needs at least two elements:

- A representation of relevant knowledge about the scope (device, homogenous network, heterogeneous network, etc.).
- A cognition loop which uses AI techniques inside its states (learning techniques, decision making techniques, etc.).

We surveyed the existing research work for these two elements and conducted relevant discussion in Section 3.2.

This view on CN can be seen as CR extended up the stack and across the network. The difference between cognitive radio (CR), cognitive network (CN) and cognitive radio network (CRN) regards only the scope and not the high-level technological approach it uses [65]. Fig. 4 illustrates our view, with CR spanning over the wireless link, thus PHY and MAC layers, CRN spanning over wireless networks (i.e. access, backhaul, homogeneous and/or heterogeneous) and across the entire protocol stack. CN is the most general being able to span over the entire communication system, including the core network. This survey focuses on CNs, for further reading on CR and CRN we point the interested reader to the seminal work in this area [17] as well as to extensive surveys in [62,63].

4.2. Cognitive framework

In [4], the authors proposed a cognitive framework for guiding research in this area. They see it as a software framework that ties high-level requirements with the underlying network using the cognitive process as presented in Fig. 5. The topmost layer allows specification of end-to-end goals by an application, user or resource. These end-to-end goals are then translated into requirements, or

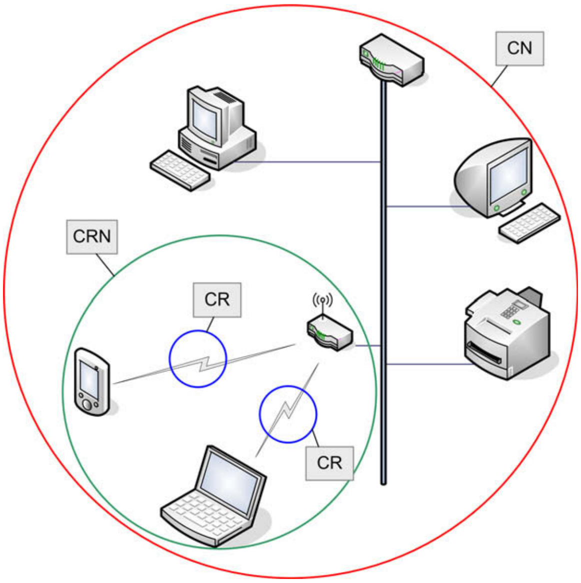


Fig. 4. The scope of CR, CRN and CN illustrated on LAN segment of the network.

policies, for the underlying mechanisms using a specification language that interfaces the topmost layer with the cognition layer. The cognition layer uses sensors to monitor the network and a network API to perform reconfigurations of the software adaptable network (SAN) in order to meet the end-to-end goals.

The cognitive process can operate in a centralized way, spanning over a large network, or in a totally distributed manner at a device level. In the first case, it might be too expensive to centralize all the network specific information that the cognition loop requires, while in the second case there might be too little knowledge available to pursue end-to-end network goals. In reality, the deployment of the cognitive functionality in a network will depend on the network specific problems and will be an engineering decision. However, it is important that the cognitive framework is designed in such way as to be modular, easily upgradeable and scalable in order to be able to accommodate existing as well as next generation technologies and applications.

4.3. Intelligent agents for CNs

The starting point towards developing a CN is the intelligent agent (IA). This section presents existing and emerging AI techniques that can prove useful for developing agents for CNs.

According to Russell and Norvig [21, p. 32], an agent is central to AI. It is an entity that perceives the environment through sensors and acts upon that environment through actuators. This is the so called "weak" definition of agency while "stronger" definitions take into account functions and characteristics of the agent [66, p. 8,21, p. 32]. Among different classifications of agents, we will consider as a reference the one established at IBM, which uses three dimensions to describe agents (see Fig. 6). The first dimension is the *Agency,* which determines the degree of "autonomy and authority vested in the agent". The second dimension is the *Intelligence*, which describes the degree of reasoning and learned behavior. Finally, the third dimension is *Mobility*, which specifies the degree to which agents travel through the network [66, p. 9].

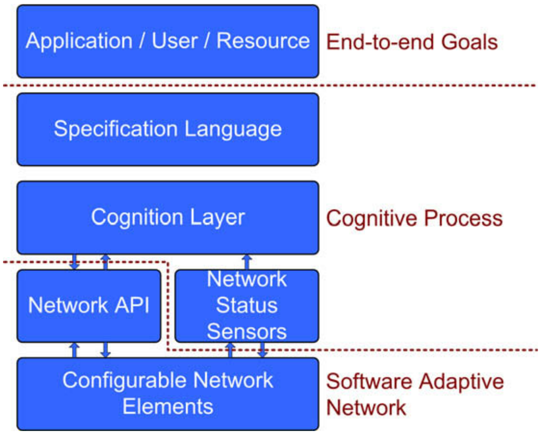


Fig. 5. Cognitive framework [4].

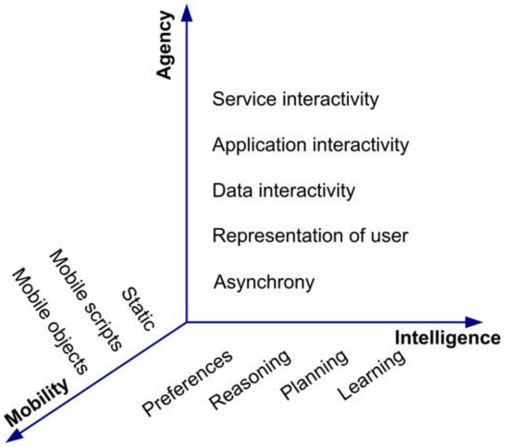


Fig. 6. Space for defining agents [66].

Current networks operate via message passing (i.e. IP packets between two routers or primitives between TCP and IP) where the receiver takes an action as a consequence of the received message. This type of operation is asynchronous and is characteristic to expert systems [66, p. 9,67]. This approach permitted loose coupling of complex systems (e.g. communication networks). However, this approach permits the lowest degree of autonomy according to Fig. 6. On the Intelligence axis, some of the current communication systems do not even reach the lowest level as they do not even allow specification of preferences (e.g. QoS specifications).

In this respect, CNs are expected to enhance the level of intelligence of current communication systems by incorporating so called Intelligent Agents (IAs) in the KP. On the Agency axis, IAs can perform actions on behalf of the user, more specifically they can interact with data, applications or services. On the Intelligence axis, IAs can hold a model (i.e. user, system, environment, etc.), perform reasoning, planning and learning. These actions are exactly the same as the ones desired from CN and can be found in the states of the cognition loop (see Plan, Decide, Act, Learn and Policy Fig. 3). Inside the KP, the IAs can be static, or make use of mobile scripts or objects while running. Mobile IAs seem to be adequate for large networks offering an even larger number of services [68].

Networks of the future will make use of agents to improve their performance with respect to all three axes in Fig. 6. In the case of CNs, the main improvement is achieved with respect to the Intelligence axis. Therefore, in the remainder of the section we focus on describing utility of IAs for these networks. We also emphasize the correspondence between IAs and the states of the cognition loop.

4.3.1. Knowledge and reasoning

From the intelligence point of view, the minimal requirement for an IA in general is to hold a model and be able to reason based on this model. These IAs are also called knowledge-based agents. Reasoning can take place upon two types of knowledge: certain (true, false and unknown) and uncertain.

Reasoning under certain knowledge is accomplished by logical agents. In this respect, agents "can form representations of the world, use a process of [logical] inference to derive new representations about the world, and use these new representations to deduce what to do" [21, p. 194]. Logical agents use symbolic knowledge representations, so called artificial languages, and typically first-order logic to infer new facts. These representations also support semantic querying. The work [3,22,23,28,29,32,33,38–40] surveyed in Section 3.2 use logical agents for knowledge representation and reasoning.

Agents that have incomplete or uncertain information use decision theory and are also called decision theoretic agents. These agents use knowledge representations specific for uncertain domains (i.e. full joint distributions can constitute the knowledge base) to reason. Then they perform probabilistic inference, which is the computation of posterior probabilities from the observed evidence. Examples of CN related work using decision theoretic agents can be found in [4,22,24–26,34,35,37] (also see Section 3.2 and Table 2).

Both types of agents use a knowledge base (KB) to store knowledge. The logical agent's knowledge is stored under the form of sentences expressed via a knowledge representation language. Some of the existing representation languages are propositional logic (also called Boolean logic) and first-order logic (also called first-order predicate calculus). For propositional logic, there are two families of inference algorithms: forward chaining and backward chaining. For first-order logic, there are three such families: forward chaining, backward chaining and logic programming, and resolution-based theorem proving [21, Chapters 8–10].

The decision theoretic agent's knowledge is stored in the form of probability distributions expressed via data structures. Most common data structures for representing uncertain knowledge are Bayesian networks (also called belief networks, probabilistic networks, knowledge maps, etc.). Typical algorithms for inference in Bayesian networks are: variable elimination, polytrees and likelihood weighting. However, if the problem being solved takes into account the time domain, then hidden Markov models, Kalman filters and dynamic Bayesian networks are needed. Other ways for uncertain reasoning are: rule-based methods reasoning, reasoning under ignorance and reasoning under vagueness (fuzzy) [21, Chapters 13–15].

One approach that combines the representational advantages of first-order logic with the ones of Bayesian networks, which are essentially propositional, is the relational probability model. This combination results in a first-order probabilistic KB able to specify probabilities for all possible first-order models [21, p. 519]. To the best of our knowledge there has been no research investigating the suitability of this approach to CNs.

4.3.2. Problem solving and planning

The simplest type of agent is the reflex agent that functions by direct mapping from states to actions. However, complex environments, where mapping is too large to be stored or takes too long to be learned, require some kind of focusing.

Problem solving agents use goals to limit the objectives they are trying to achieve [21, p. 59]. The first thing problem solving agents do is to formulate a goal. Based on this goal, they formulate a problem and search for solutions for this. Search strategies can be classified as uninformed and informed search, according to whether they do or do not use problem specific knowledge (e.g. indication on where to search for solutions). Alternatively, search strategies can be classified as offline (planning beforehand) and online (run time planning) search strategies.

Uninformed search (also called naïve or brute-force search) algorithms use the simplest, most intuitive method of searching through the search space. Conversely informed search algorithms use heuristic functions to apply knowledge about the structure of the search space to try to reduce the amount of time spent for searching. Relevant uninformed search algorithms are: breadth-first, depth-first, deep-limited, iterative deepening and bidirectional. Heuristics for informed search include: greedy best-first search, A^* search, simulated annealing and genetic algorithms [21, Chapters 3 and 4].

When the problem-solving agent has some knowledge of the states of the problem it can formulate and solve a constraint satisfaction problem. In such case, the states and goal conform to a representation and search algorithms can use general purpose (as opposed to problem specific) heuristics such as: backtracking search, minimum remaining value search, forward checking search, arc consistency search, conflict-directed back-jumping and tree decomposition searches [21, Chapter 5].

Sometimes there can be several agents in an environment (also called a multi-agent environment) each trying to solve a problem. When these agents have conflicting goals, they compete against each other resulting in an adversarial search problem. Such problems are also known as games in environments with relatively low number of agents or economies in environments with a large number of agents. Several algorithms (extensively studied in game theory [69]), such as minimax and alpha–beta, are available for adversarial search [21, Chapter 6]. The game theory approach is being used for the problem of dynamic spectrum assignment [62] in CR as defined by FCC.

The search-based problem-solving agents are a subset of planning agents. The drawback of a problem solving agent is that it requires a human to provide a heuristic function for each new problem, thus it lacks autonomy. Logical planning agents try to mitigate this drawback by possessing a representation of the goal as a conjunction of sub-goals (i.e. problem specific knowledge as discussed for informed search-based problem-solving agents) that allows them to use a single domain-independent heuristic [21, Chapter 10].

Complex planning problems use a language for representing planning problems, including actions and states. This logical representation allows planning algorithms to take advantage of the logical structure of the problem. Most commonly, the solution to a planning problem is an action sequence which, when executed, satisfies the goal. Backward and forward search algorithms such as state-space search, partial order planning, GRAPHPLAN and SATPLAN are used by logical planning agents [21, Chapter 11].

Planning for deterministic, fully observable, finite, static and discrete environments is also called classical planning. Non-classical planning is the planning employed when it comes to partially observable and/or stochastic environments such as real-world environments. Real-world planning problems have to take into account time and resources. Problems such as dynamic resource management could be solved using planning and scheduling, taking into account time and resource constraints. Conditional planning deals with uncertainty and checks periodically what is happening in the environment.

Other, more sophisticated planning agents exist as well. The execution monitoring agent checks whether everything is going according to the plan. The action monitoring agent checks the environment to verify that the next action will work. The plan monitoring agent verifies the entire remaining plan. The replanning agent knows what to do when something unexpected happens while the continuous planning agent persists indefinitely in an environment. In a multi-agent environment, multi-agent appropriate planning solutions are used according to whether the agents cooperate, compete or coordinate [21, Chapters 12 and 13].

It can be seen that problem solving and planning combine search and logic. "When a logical agent cannot conclude that any particular course of action achieves its goal, it will be unable to act. Conditional planning can overcome uncertainty to some extent, but only if the agent's sensing actions can obtain the required information and only if there are not too many different contingencies" [21, p. 463]. In real-world problems logical and decision theoretic agents can be combined to make rational decisions.

The choice of the planning agent for use in the CN will obviously depend on the complexity of the problem it needs to solve. While the complexity of the agents needs to be kept at the minimum, CNs are likely requiring complex real-world planning and multi-agent environments. Furthermore action, plan monitoring and/or replanning agents might be needed to make sure the goals are achieved. In the surveyed work, a logical planning agent was used for network security in [39].

4.3.3. Decision making

Decision making [70] combines probability theory with utility theory to allow an agent, called decision theoretic agent, to make decisions in such way as to get what it wants at least on average, thus yielding better than random choice results. Decision theoretic agents have a continuous measure of state quality, while goal based agents only distinguish between good and bad states, thus having a binary measure for state quality. Rational agents choose "the right thing to do" based on the importance of various goals (or their utility) and the likelihood (or probability) that they will be achieved [21, p. 463].

Rational agents make decisions using a model represented with a formal language for knowledge representation. In a pure uncertainty model the decision is taken under ignorance whereas in a deterministic model, the decision is taken under complete knowledge. Somewhere in the middle lies decision making under risk which uses a probabilistic model [71]. Decision networks, an extension of Bayesian networks, are one of the formalisms for dealing with decision making under uncertainty [21, Chapter 16].

Complex decision making problems consider decision making in more than one episode where an agent's utility depends on a sequence of decisions. Sequential decision problems include utilities, uncertainty, and sensing, and they generalize the search and planning problems. These problems are also called Markov decision processes (MDPs) and are defined by a transition model and a reward function. The transition model specifies the probabilistic outcomes of actions and the reward function specifies the reward in each state. The solution of a MDP is a policy that associates a decision with every state the agent might reach. The value iteration algorithm is commonly used for solving MDPs. Partially observable MDPs are suitable for partially observable environments and use a dynamic decision network to represent transitions and observable models, to update its belief state and to project forward possible actions. Rational behavior of agents in a multi-agent environment is described by game theory and solutions of games are called Nash equilibria [21, Chapter 17].

An example of decision theoretic agent for CNs can be found in [35].

4.3.4. Learning

Machine learning [72] is defined as "*computer algorithms that improve automatically through experience*" and stands at the foundation of the learning state of the cognition loop. Learning agents observe their interactions with the world and monitor their decision making process. These observations can be used by the agent to improve its acting in the future. Typically, learning techniques are classified into four classes, depending on the type of feedback available for learning: supervised, unsupervised, semi-supervised and reinforcement learning.

Supervised learning is appropriate for fully observable environments. Typically, with this type of learning the data is labeled or directly observable by the learning agent which learns a function by examples. Inductive learning is a form of supervised learning which enables learning simple theories in propositional logic. Learning decision trees and ensemble learning are examples of inductive learning [21, Chapter 18].

Unsupervised learning typically tries to find patterns in data which is unlabeled and no observations are available to the agent. Such an agent learns a probability model from examples. A purely unsupervised learning agent cannot learn what to do, because it has no information as to what constitutes a correct action or a desirable state. Statistical learning, learning with hidden variables, instance based learning, neural networks and kernel machines are types of unsupervised learning techniques [21, Chapter 20].

Semi-supervised learning falls between supervised and unsupervised. In this case, only some of the training instances are labeled while most of them are unlabeled to reduce the expenses and effort [36].

Reinforcement learning allows an agent to learn based on successes and failures. Reinforcement learning typically includes the sub-problem of learning how the environment works. This type of learning relies on two types of input: the reward and the sensory input. Types of reinforcement learning are passive reinforcement learning, active reinforcement learning and policy search [21, Chapter 21]. Reinforcement learning is the most common learning method used in CN [34,35,4,24].

The learning techniques presented so far assume there is no prior knowledge, thus no model of the problem, which is only built during the learning process. However, learning methods can take also advantage of prior knowledge, thus making use of knowledge representation. Most frequently, prior knowledge is represented as general first-order logical theories. Explanation based learning, learning using relevance information and knowledge-based inductive learning are examples of learning based on prior knowledge [21, Chapter 19].

An alternative view on learning for CNs can be found in [36], classifying learning techniques based on the formulation of the learning task. Therefore, they discuss three types of learning: learning for classification and regression, learning for acting and planning and learning for interpretation and understanding.

4.3.5. Communicating

Communication is particularly important in multi-agent partially observable environments where it can help agents be successful because they can learn information that is observed or inferred by others. Agents can inform each other by making statements or answering questions, they can request other agents to perform an action and they can acknowledge or commit to something [21, Chapter 22].

4.3.6. Discussion

So far, specifications for cognitive networks seem to be in line with the design principles for the future Internet as presented in [73]. By creating a ‘‘playfield” for the tussles that will define the future of networking, cognitive networks open a wide and potentially rich research field. Perhaps, networking technology will follow the same road as the World Wide Web (WWW) did and will become open and dynamic enough to be able to disseminate content in a reliable way [74]. So far, the three driving forces which stand behind communication networks seem to be moving towards offering a seamless user experience using cognitive technologies and a unified business model as can be seen in Fig. 7 [75].

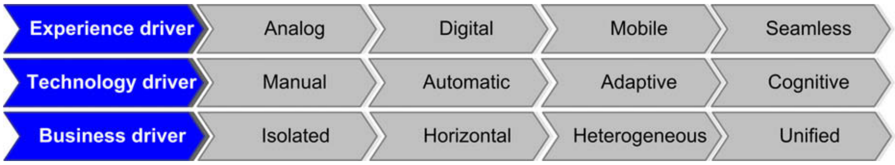
In this section, we mapped AI techniques to the states of the cognition loop, showing the role of these techniques in the development of CNs. Even more, we presented a condensed view on agents and their usability in the cognition loop. It can be noted that there are many possible techniques, algorithms and combinations of these for developing CNs. However, as can be seen in the previous sections, only a limited number of these algorithms and techniques have been used for solving communication networks specific problems. Furthermore, in many cases the motivation for choosing a particular technique is not thoroughly justified and the performance of the system is not compared against other systems employing alternative methods. We see as the main reason for this the cross-domain nature of research in CNs. This requires in depth knowledge in rather disparate areas (i.e. machine learning and network management). However, the development of prototypes for cognitive networks will require overcoming this obstacle so that clear formulations of the problem can be specified and suitable techniques can be applied. As Mitola noted, the research in CR ‘‘is on the organization of cognition tasks” rather than developing algorithms and techniques [17]. We broadly agree with this view which also applies to CRNs and CNs, however we also see some challenges for the domain of AI.

The most researched techniques in AI have been focused on static and centralized environments so far. Computer networks, on the other hand, are dynamic and often distributed. In [36], open issues and challenges for machine learning for cognitive networks are identified. In this respect autonomic, online, distributed and knowledge-rich learning methods that have not been widely investigated are needed for CNs. However, the research for online and distributed methods especially for reasoning [76] and learning [77,78] is under progress so that developers of CNs should be permanently kept updated.

Communication networks have few distinctive features such as dealing with uncertain information and partially observable worlds, but they require high availability and reliability. In this respect methods required for such systems might be different from the ones employed for organizing, searching and structuring large amount of information. Therefore, selecting suitable techniques and building and maintaining coherent knowledge bases which might also influence the learning process to ignore ‘‘unimportant” facts is another important issue in CNs.

5. Architectures for cognitive networks

Several architectures for cognitive networks have been proposed in the literature so far. Thomas [2] classifies them in two categories according to the purpose they serve: architectures for network management and architectures for solving ‘‘hard” problems. We argue in the following that in fact these architectures fall under one or more of the key functional areas of network management [79] and classify



Experience driver	Analog	Digital	Mobile	Seamless
Technology driver	Manual	Automatic	Adaptive	Cognitive
Business driver	Isolated	Horizontal	Heterogeneous	Unified

Fig. 7. Three drivers in communication networks (adapted from [75]).

them as architectures designed for wireless access networks and architectures designed for core networks. We also discuss similarities and differences between these architectures and mention tools they use for achieving their purpose.

5.1. Management architectures for wireless access networks

Network management is a resource consuming task and it heavily relies on human intervention. Future wireless networks will be composed of multiple technologies and a large number of dynamic (reconfigurable) nodes that will pose even greater challenge for management. Network pre-planning will not be a feasible solution in such networks where heterogeneous devices will have to coexist without harmfully interfering each other; bandwidth requirements will have large variations due to vertical handovers and user requirements, etc. In this context the network should make use of the KP. In the following we present several proposals of architectures for wireless access networks which make use of cognition.

The Personal Router (PR) project [80] proposed, to the best of our knowledge, the first architecture for a CRN. The project had two goals: first, to offer customized wireless services in an open market and second, provide users access to these services through a device called personal router. The end-to-end goal of the resulting heterogeneous wireless access network seems to had offered the best and most convenient access service. As part of this project, in [34], the authors investigate a cognitive agent for wireless network selection which is designed to hide the complexity of the wireless environment from the user. The agent interacts with the wireless environment by continuously monitoring it for new access technologies and services, and with the user to retrieve minimal feedback with respect to its satisfaction. The agent learns the user's preferences and selects the network that best fits these preferences in an autonomic and continuous fashion. In [81], the user is modeled using Markov decision processes aided by either reinforcement learning or collaborative filtering as learning methods.

The End-To-End Reconfigurability (E2R) project [26] is an initiative to build an infrastructure that supports reconfigurable heterogeneous wireless systems and pursues end-to-end connectivity between users. The role of cognition in this architecture is to optimize resource utilization while being aware of the end-to-end goals of the network. Dimitrakopoulos et al. present in [24], as part of the work carried out under the E2R project, a management scheme for distributed cross-layer reconfiguration (DCLR) for cognitive Beyond 3G (B3G) architectures, where they introduce a cognitive loop. The end-to-end goal of this scheme is expressed by an objective function that aims to assign users their preferred QoS to the largest extent possible with minimum number of needed reconfigurations. They use Bayesian networks to model the network and reinforcement learning to increase the probability of selection in the future of a well performing scenario.

The DCLR problem takes three inputs: monitoring, discovery, profiles; and renders four outputs: transceiver reconfiguration, traffic distribution, QoS assignment and objective function. The reconfigurations performed by this management scheme affect several layers of the protocol stack: the transceiver reconfigurations affect PHY and MAC layers, the traffic distribution affects the network layer while the QoS assignment affects the application layer. This architecture performs autonomic performance management by aiming to provide end-to-end connectivity to the users at the highest QoS possible and it also performs autonomic network configuration in order to reach this aim. The above described architecture assumes a network of managed elements that do not interact directly among themselves but listen to a master controller (a base station with heterogeneous transceivers).

The m@ANGEL [25] platform's functionality is similar to the goals of the E2R project but the motivating aspect is a business level view of cognitive wireless access networks. The idea behind E2R and m@ANGEL is similar to that in the PR, as they both address the heterogeneous wireless access segment. However, the first two focus on the heterogeneous access point while the third uses a multi-interface personal router delimiting the devices in the personal area network from the provider's access points.

Mahonen et al. propose in [55] a cognitive resource manager (CRM) architecture that interacts with the layered protocol stack and enables autonomic optimization of the communication stack as a whole. Several CRMs residing on different entities can communicate to achieve a globally optimal solution. Thus far, the description of the CRM is similar to the description of the KP, but the authors go into more detail by defining a policy repository and a toolbox that interact with the CRM. The policy repository dictates policies for the CRM to follow, while receiving recommendations and experiences from the CRM. The toolbox is formed of a set of tools such as neural networks, genetic algorithms, simulated annealing, Bayesian reasoning, etc that can be added in a plug and play fashion according to CRM's needs. These tools allow the CRM to efficiently handle large amount of data. Furthermore, the paper presents the way CRM can be used in the context of CRNs. Authors suggest that the CRM can interact with a unified link layer API (ULLA) and determine changes in frequency channels or radio access technology. This idea is not far from the one presented by Dimitrakopoulos et al. [24] and discussed above, but does not assume any master entity in the network. Different CRMs reside on different entities and they can directly communicate between themselves. This architecture promises to perform autonomic resource and performance management similar to the one proposed in [24].

Sutton et al. propose in [54] a reconfigurable platform for cognitive networks that consists of reconfigurable wireless nodes. The platform uses the cognition cycle in order to reconfigure the node; it uses a configuration parser that translates the network's protocol stack configuration to XML and a component manager that maintains an inventory of the components which can be used to build the node's protocol stack. The key component in this framework is the protocol stack manager that takes as inputs the XML and component inventory and constructs the protocol stack of the element and controls its operation. This work discusses device level cognition that performs configuration and resource management.

Thomas [2] applies cognitive network principles to the multicast flow lifetime and the topology control problems in wireless networks. In the first case, the cognitive network has a single objective optimization as its end-to-end goal, namely to maximize the lifetime of a multicast tree. In order to do this, the cognitive process can reconfigure the SAN by adjusting the transmission power, controlling the direction of the transmit antennas and adjusting the network's routing functionality. The information received from the SAN refers to the set of the next hop radios in the multicast tree, k-hop topology and power knowledge. The second problem he approaches is a two objective optimization problem that is based on the first problem but also considers spectral efficiency in addition to lifetime. Game theory, multi-agent systems, multi-objective optimization, machine problem solving and learning tools are used for tackling the two problems and perform resource and configuration management to achieve goals.

Recently, a new architecture called SmartA [82] has been proposed. The architecture conforms to the cognitive framework described in Section 4.2, it integrates the Context Manager [5], the Resource Manager [55], the Stack Manager [54] and it introduces the Service Manager (SM) in the context of heterogeneous wireless networks. The SM takes as input application layer requirements and context information. It holds semantic representation of QoS knowledge and makes use of a control loop. In this setting, it translates application requirements to platform dependent requirements and determines the set of optimal assignments of radio interfaces to services. The Resource Manager decides on the best assignment based on previous experience, thus making use of the cognition loop.

5.2. Management architectures for core networks

Core networks are high capacity fixed networks that carry vast amounts of data. Upon the failure of (part of) such network, a large number of users are affected. However, the network is not aware of what data it was carrying, how this data was affected and what to do in order to fix the failure. Such networks generate a large amount of alarms that have to be analyzed by human operators to determine causes of failures, and then they have to manually reconfigure malfunctioning parts. Trying to make the management of core networks less human dependent and more self-aware is a challenging research topic.

The Foundation–Observation–Comparison–Action–Learn– rEason (FOCALE) architecture is a result of efforts to build an autonomic management platform [5] for core networks. An autonomic management framework is supposed to resemble the human autonomic nervous system that performs unconscious actions. This way, it tries to identify functions that can be performed without human intervention and reach the *recognize-act* state described by Clark et al. [1]. Even though the FOCALE architecture is not explicitly referred to as having cognition, its description contains all the elements required by a cognitive network as described in Section 3. The architecture contains two control loops, one for maintaining the current state and one for reconfigurations. The main idea standing behind this architecture is to be able to specify business requirements in a restricted natural language to the management plane of the network. These requirements would be then translated into a form that can be used to automatically configure network resources. When the context changes, the management plane should be able to sense that and act in such way as to comply with the requirements.

In order to materialize this idea, the business requirements written in natural language are translated into a set of requirements used by the policy manager. The policy manager interacts with a context manager to obtain eventual changes in the state of the managed entity, and with the autonomic manager that is responsible for issuing network configuration commands. The autonomic manager uses policy information, data models and ontologies in order to issue network configuration commands. The data models are associated to corresponding ontologies which exhibit cognitive similarities, all this to hide vendor specific interfaces. The configuration commands issued by the autonomic manager reach the managed entity via a model based translation layer that transforms them into vendor specific commands [5]. The FOCALE architecture was defined for managing fixed networks, but extensions have been proposed to support management of wireless networks as well.

5.3. Discussion

The seven architectures described in this section generally exhibit the requirements posed by a cognitive network as described in Section 4. All of them pursue an end-to-end goal, use some kind of cognition loop and some kind of cognitive framework and can be implemented in the frame of the KP concept. Nevertheless, there are some substantial differences in terms of complexity and functionality between network management architectures designed for wireless access and core networks. The first are mainly dealing with radio interface reconfiguration and radio resource management, whereas the second are predominantly concerned with resilience and configuration of network resources. In other words, network management architectures for wireless access networks are generalizing the resource management functionalities of contemporary wireless networks, which are already now predominantly automated processes. On the other hand, network management architectures for core networks are designed to carry out processes that are at present day executed manually or in semi-automated manner by experienced and trained network operators.

Table 4 summarizes key characteristics of the seven representative architectures and emphasizes the wide range of networks and scenarios where cognition could be used: wireless access or backhaul, core networks, device and network level, etc. All these architectures perform some kind of configurations at several layers of the protocol stack, most of the time using information from other

Table 4
Cognitive architectures.

Architecture	Personal router	E2R/ m@ANGEL	Mahonen et al.	Sutton et al.	Thomas	SmartA	FOCALE
References	[34,35,80,81]	[26,24]/[25]	[55]	[54]	[2]	[3, p. 34]	
Year	2002	2004	2006	2006	2005	2008	2006
End-to-end goal	Customized wireless services with high user satisfaction	High QoS seamless connectivity	Optimization of the communication stack	Device reconfiguration	Maximize multicast flow lifetime, spectral efficiency	User-transparent optimal wireless service selection	Reconfiguration according to business requirements
Cognition loop	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Cognitive framework	PR agent: service evaluator, service change controller, service value predictor	Cognitive process, SAN	Cognitive resource manager, SAN, cross-layer design	Protocol stack manager, SAN, XML	Cognitive specification language, power control, direction control, routing control, topology control, SAN	Service manager interface, reasoner, resource manager	Model based translation and reasoning, policy manager, autonomic manager, context manager, DEN-ng, ontologies
Knowledge representation	Numeric	Numeric	Numeric	Symbolic	Numeric	Symbolic	Symbolic
Reconfig.	Not specified (probably network and application)	PHY, MAC, network, application	PHY, MAC	All layers	PHY, MAC, network	Not specified (PHY, MAC and application)	All layers

layers, thus they could also be seen as a cautious cross-layer design.

The PR project was backed by some of the conceivers of the KP and is meant to develop technology to allow the user of wireless services to connect to the access offering high satisfaction and low cost. The selection would be transparent to the user and would not be bound by technological (i.e. vertical handover) or economical (i.e. contract with a specific service provider) constraints. The shortcoming of wireless access environments standing behind this project are of the utmost actuality in today's wireless world. The difference is that the number of wireless access technologies has increased and that the user terminals became much more complex so that they can be the PRs themselves, not requiring an extra device. The PR looks at the access from the user's perspective but does not investigate the technological challenges from the operator's perspective.

The more recent and much larger scale E2R integrated project, on the other hand, focuses on the reconfigurations needed at the operator's access infrastructure, thus complementing research in the PR. This is an ongoing initiative that investigates reconfigurability starting with the radio interface and ending at the business models, thus upgrading the architecture of existing provider systems. This initiative is also strongly backed up by industry.

The architecture of the CRM proposed by Mahonen et al. mainly targets cognitive reconfiguration of parameters across all the layers of the protocol stack. So far, the implementation of the ULLA allows the CRM to interact with the PHY/MAC layer. The generic architecture of the CRM is applicable to any device featuring a communication protocol stack and deals with managing the resources and optimizing their usage.

The reconfigurable node proposed by Sutton et al. complements the CRM in such way that it automatically and dynamically builds a layer or the entire protocol stack as opposed to tuning and optimization of a static stack. A major shortcoming that we see in this work is the lack of validation. We are not aware of any investigation on how to dynamically connect together software components for a layer for instance.

Thomas investigated distributed cognitive processes on wireless mesh networks, thus his work is complementary to the PR and E2R which focus on traditional centralized scenarios. However, even though he defined a complex framework, including the cognitive specification language, only part of this was investigated so far. This work seems to also be highly theoretical.

SmartA can be seen as a contemporary version of the PR proposal addressing heterogeneous access environment and multimode end devices. One of the specifics of this proposal is that it emphasizes the role of semantic knowledge representation for CRNs.

FOCALE is an extremely ambitious architecture which focuses on automating network management, perhaps the biggest problem communications providers are facing today. Automation would result in increased reliability, less costs for human experts and more rapid repairs. However, this architecture is relatively new and one of its versions is subject to investigation in a project which started recently [86].

Different research groups use different terminologies for similar things. Some groups call their architectures *autonomic*, implying that the network can operate without (or with little) human intervention (see Table 4), i.e. they are self-governing networks. Other groups call their architectures *cognitive* and they refer to the fact that the network is, among others, able to reason and learn, thus being self-aware. However, some proposed autonomic architectures also have reasoning and learning capabilities so that, by the definition, they could also be called cognitive networks. On the other hand, cognitive networks are also able to function without human intervention. Finally, to make it even more confusing, some groups use both terms, autonomic and cognitive, for the same architecture (see Table 4). This confusion seems to have two main roots: first, it is not clear what the difference between *autonomic* and *cognitive* is; and second, it is not clear what level of a network the term refers to: is it radio level, device level, network level?

It seems like the terms *autonomic* and *cognitive* are inspired from biology. In humans, the autonomic nervous system performs tasks that we do not have to think about doing (e.g. breathing). The existence of this system allows us to function normally and allows the brain to perform cognitive tasks that require learning and thinking. For instance, we can learn how to swim and by doing this, we have to think about regulating our breathing. It seems like autonomy allows the luxury of doing cognitive tasks but cognitive tasks can, to a certain degree, control some autonomic tasks. In our view an autonomic network (or system) is more rigid and uses an algorithmic approach, being best suited for simpler, more predictable issues where the space of variables can be handled. However, for large, unpredictable networks or systems, having a large space of variables, a cognitive approach based on learning and reasoning is required. In a network, the cognitive features can be placed at device level to handle radio relates issues leading to the concept of cognitive radios [17], then it can handle the entire protocol stack leading to a cognitive platform [55,54], they can span over a network [24,25,3] or a combination of the two. It is possible to end up having a hierarchy of cognitive entities inside a network.

As discussed throughout this paper, research in cognitive networks has been mainly limited to the architectures summarized above. Even the papers from the previous two years of IEEE Globecom investigate cognitive radios (PHY, MAC) and not cognitive networks. This observation is confirmed by the map in Fig. 1 where the occurrences of the keyword *cognitive* are closer to the lower cluster (channels). However, some recent initiatives [83–87] concerned with defining the future internet [88] are likely to focus some resources also in researching cognitive networks.

The main research challenge in this area seems to consist of finding the right methods from machine learning and related areas that can efficiently be applied, or even adapted if necessary, to solve network specific problems. This is a difficult task, especially because networked environments pose challenges and require techniques and algorithms that are different from the ones typically used in machine learning (i.e. supervised and semi-supervised learning, offline learning) [36]. Some other major challenges address designing and prototyping the KP, creating interfaces between this plane and the existing OSI layers through which sensing/actuating functions can be carried out. In order for the KP to be able to reason, there is a need for network related knowledge to be properly represented, similar to the radio knowledge [17], using ontology based approach.

With respect to cognitive networks we also see a conceptual challenge. Our survey shows that there is still confusion in terminology. We see cognitive networks as having a plane (i.e. KP) spanning over layers, devices and technologies. This plane contains cognition loops and knowledge representations which enable the achievement of end-to-end goals. However, for more concerted work between different research groups, the concepts and terminology should be harmonized and we hope this survey is a step towards this.

6. Standardization

Several standardization bodies have already recognized the need and the potential of introducing cognitive concepts and are working on upgrading existing standards as well as on defining new standards for B3G heterogeneous networks. For now, most of standardization work focuses on cognitive radios (research in this domain is well ahead compared to cognitive networks which is just taking off) and affects PHY and MAC levels [89]. Lately, standardization efforts are extending on other aspects of networks as well. The work in the IEEE Technical Sub-committee on Cognitive Networks (TCCN) [90] addresses a broad range of issues such as agile/dynamic spectrum access networks, related issues from PHY to application layers, security issues, policy issues (e.g. spectrum policy reform by U.S., Canada and European Union), implementation technologies (e.g. software radio, middleware), economic considerations and standardization activities. Some standards currently under development such as IEEE 802.21 [91] and IEEE 802.22 [92] would benefit from research in cognitive networks since they address handover and interoperability in heterogeneous networks and cognitive radio-based PHY/MAC/air_interface for the use by license-exempt devices in spectrum that has been in the past allocated to the analog TV Broadcast Service.

7. Conclusions

The current communication networks composed of different wired and wireless interworking technologies are reaching the point where traditional network management and maintenance will be no longer able to cope with increasing network complexity. The recently emerging CN concept is promising to be the right answer to emerging challenges of the network management.

In this paper we surveyed existing work on CNs. We first analyzed recent research trends in communications, revealing emerging activities in the areas of cross-layer design and CNs. Then we classified research trends with respect to the approaches towards the traditional layered architecture. We emphasized the importance of the knowledge representation and the cognition loop for CNs, arguing that these two elements are crucial for the implementation of the KP. We mapped existing AI techniques to the states of the cognition loop and identified challenges for research in AI from which CNs could benefit. Furthermore, we surveyed proposals for CN architectures and discussed their relative merits. We concluded the paper with identification of standardization activities related to or potentially benefiting from the research in the area of CNs.

The discussions in this paper indicate that the way forward in developing CNs is to bring together the experts from the areas of communication networks and AI. Communication networks are faced with great complexity challenges and several AI techniques proved to handle complexity well. Furthermore, AI is searching for areas of applications, and communication networks are underexploited in this respect.

Acknowledgements

The authors would like to thank B. Fortuna, J. van der Merwe, S. Răpuntean and, last but not least, the anonymous reviewers for helping to improve this paper. This work was partially supported by the Slovenian Research Agency.

References

[1] D.D. Clark, C. Partrige, J.C. Ramming, J.T. Wroclawski, A knowledge plane for the internet, in: Proceedings of the SIGCOMM 2003, Karlsruhe, Germany, August 25–29, 2003.

[2] R.W. Thomas, Cognitive Networks, Ph.D. Dissertation, Virginia Polytechnic and State University, Blacksburg, VA, June 15, 2007.

[3] Q. Mahmoud, Cognitive Networks – Towards Self-Aware Networks, John Wiley and Sons, 2007, ISBN 9780470061961.

[4] R.W. Thomas, L.A. DaSilva, A.B. MacKenzie, Cognitive networks, in: Proceedings of the First IEEE International Symposium on New Frontiers in Dynamic Spectrum Access Networks, Baltimore, MD, USA, November 8–11, 2005.

[5] J. Strassner, The role of autonomic networking in cognitive networks, in: Q.H. Mahmoud (Ed.), Cognitive Networks: Towards Self-Aware Networks, John Wiley and Sons, 2007.

[6] Ontogen. <http://ontogen.ijs.si/> (visited on February 2008).

[7] V. Srivastava, M. Motani, Cross-layer design: a survey and the road ahead, IEEE Communications Magazine 43 (12) (2005).

[8] M. Chiang, S.H. Low, A.R. Calderbank, J.C. Doyle, Layering as optimization decomposition: a mathematical theory of network architectures, Proceedings of the IEEE 95 (1) (2007).

[9] M. Chiang, S.H. Low, A.R. Calderbank, J.C. Doyle, Layering as optimization decomposition: current status and open issues, in: Proceedings of the 40th CISS, Princeton, NJ, USA, March 2006.

[10] M. Chiang, S.H. Low, A.R. Calderbank, J.C. Doyle, Layering as optimization decomposition: framework and examples, in: Proceedings of the IEEE Information Theory Workshop, Puerto Rico, March 13–17, 2006.

[11] G. Giambene, S. Kota, Cross-layer protocol optimization for satellite communication networks: a survey, International Journal of Satellite Communications and Networking 24 (5) (2006).

[12] S.S. Dixit, IP Over WDM: Building the Next Generation Optical Internet, John Wiley and Sons, 2003, ISBN 0471212482. p. 293.

[13] A.S. Tanenbaum, Computer Networks, fourth ed., Prentice Hall, 2002, ISBN 0130661023. p. 34.

[14] G. Carneiro, J. Ruela, M. Ricardo, Cross-layer design in 4G wireless terminals, IEEE Wireless Communications 11 (2) (2004).

[15] V. Kawadia, P.R. Kumar, A cautionary perspective on cross-layer design, IEEE Wireless Communications 12 (1) (2005).

[16] V. Srivastava, M. Motani, Cross-layer design and optimization in wireless networks, in: Q.H. Mahmoud (Ed.), Cognitive Networks: Towards Self-Aware Networks, John Wiley and Sons, 2007.

[17] J. Mitola, Cognitive Radio – An Integrated Agent Architecture for Software Defined Radio, Ph.D. Dissertation, Royal Institute of Technology, Kista, Sweden, May 8, 2000.

[18] P. Balamuralidhar, R. Prasad, A context driven architecture for cognitive nodes, Wireless Personal Communications 45 (2008) 423–434.

[19] G.D. Abowd, A.K. Dey, P.J. Brown, N. Davies, M. Smith, P. Steggles, Towards a better understanding of context and context-awareness, in: Proceedings of the First International Symposium on Handheld and Ubiquitous Computing, Karlsruhe, Germany, 1999, pp. 304–307.

[20] R. Davis, H. Shrobe, P. Szolovits, What is a knowledge representation?, AI Magazine 14 (1) (1993) 17–33

[21] S. Russell, P. Norvig, Artificial Intelligence: A Modern Approach, second ed., Prentice Hall, 2002, ISBN 0137903952.

[22] A. Devitt, B. Danev, K. Matusikova, Constructing Bayesian networks automatically using ontologies, in: Proceedings of the Formal Ontologies Meets Industry, Trento, Italy, December 2006.

[23] E. Lehtihet, J. Strassner, N. Agoulmine, M. O'Foghlu, Ontology-based knowledge representation for self-governing systems, Large Scale Management of Distributed Systems, Springer, Berlin/Heidelberg, 2006, ISBN 9783540476597.

[24] G. Dimitrakopoulos, K. Tsagkaris, K. Demestichas, E. Adamopoulou, P. Demestichas, A management scheme for distributed cross-layer reconfigurations in the context of cognitive B3G infrastructures, Computer Communications 30 (18) (2007).

[25] P. Demestichas, V. Stavroulaki, D. Boskovic, A. Lee, J. Strassner, m@ANGEL: autonomic management platform for seamless cognitive connectivity to the mobile internet, IEEE Communications Magazine 44 (6) (2006).

[26] End-To-End Reconfigurability. <http://e2r2.motlabs.com/project_overview> (visited on February 2008).

[27] M.A.L. Thathachar, P.S. Sastry, Networks of Learning Automata: Techniques for Online Stochastic Optimization, first ed., Springer, 2003, ISBN 1402076916.

[28] D. Lewis, J. Keeney, D. O'Sullivan, S. Guo, Towards a managed extensible control plane for knowledge-based networking, in: Proceedings of the International Workshop on Distributed Systems: Operations and Management, (DSOM 2006), Manweek 2006, Dublin, Ireland, October 2006.

[29] S. Guo, J. Keeney, D. O'Sullivan, D. Lewis, Adaptive semantic interoperability strategies for knowledge based networking, in: Proceedings of the Third International Workshop on Scalable Semantic Web Knowledge Base Systems, Vilamoura, Algarve, Portugal, November 27–29, 2007, pp. 1187–1199.

[30] C. Zhou, L.-T. Chia, B.-S. Lee, Semantics in service discovery and QoS measurement, IEEE IT Professional Magazine 7 (2) (2005) 29–34.

[31] QWL-QoS Ontology. <http://www3.ntu.edu.sg/home5/PG04878518/OWLQoSOntology.html> (visited on August 2008).

[32] G. Dobson, A. Sanchez-Macian, Towards unified QoS/SLA ontologies, in: Proceedings of the IEEE Services Computing Workshops, Chicago, USA, September 18–22, 2006, pp. 169–174.

[33] L. Zhou, H.K. Pung, L.H. Ngoh, Towards semantic modeling for QoS specification, in: Proceedings of the IEEE Conference on Local Computer Networks, Tampa, FL, USA, November 2006, pp. 361–368.

[34] G. Lee, P. Faratin, S. Bauer, J. Wroclawski, A user-guided cognitive agent for network selection in pervasive computing environments, in: Proceedings of the Second IEEE International Conference on Pervasive Computing and Communications, Orlando, FL, USA, March 14–17, 2004, p. 219.

[35] G. Lee, P. Faratin, S. Bauer, J. Wroclawski, Learning user preferences for wireless service provisioning, in: Proceedings of the Third International Joint Conference on Autonomous agents and Multiagent Systems, NY, USA, July 19–23, 2004, pp. 480–487.

[36] T.G. Dietterich, P. Langley, Machine learning for cognitive networks: technology assessment and research challenges, in: Q.H. Mahmoud (Ed.), Cognitive Networks: Towards Self-Aware Networks, John Wiley and Sons, 2007.

[37] E. Gelenbe, R. Lent, A. Nunez, Self-aware networks and QoS, Proceedings of the IEEE 92 (9) (2004) 1478–1489.

[38] S.B. Kodeswaran, O. Ratsimor, A. Joshi, F. Perich, Utilizing semantic tags for policy based networking, in: Proceedings of the IEEE Globecom, Washington, DC, USA, November 26–30, 2007, pp. 1954–1958.

[39] B. Shepard, C. Matuszek, C.B. Fraser, W. Wechtenhiser, D. Crabbe, Z. Gundordu, J. Jantos, T. Hughes, L. Lefkowitz, M. Witbrock, D. Lenat, E. Larson, A knowledge-based approach to network security: applying Cyc in the domain of network risk assessment, in: Proceedings of the Innovative Applications of Artificial Intelligence Conference, Pittsburgh, PA, USA, July 9–13, 2005.

[40] J. Undercoffer, J. Pinkston, Modeling computer attacks: a target-centric ontology for intrusion detection, in: Proceedings of the CADIP Research Symposium, Baltimore, USA, October 25–26, 2002. <http://www.csee.umbc.edu/cadip/2002Symposium/Ont-for-IDS.pdf> (visited on August 2008).

[41] V.I. Gorodetsky, I.V. Kotenko, J.B. Michael, Multi-agent modeling and simulation of distributed denial of service attacks on

computer networks, in: Proceedings of the Third International Conference on Navy and Shipbuilding Nowadays, St. Petersburg, Russia, June 2003.
[42] OWL Web Ontology Language Overview. <http://www.w3.org/TR/owl-features/>.
[43] RDF Resource Description Framework. <http://www.w3.org/RDF/>.
[44] The DARPA Agent Markup Language Homepage. <http://www.daml.org/>.
[45] T. Berners-Lee, J. Hendler, O. Lassila, The semantic web: when the internet gets smart, Scientific American (2001).
[46] OMG Object Management Group. <http://www.omg.org/>.
[47] Jena – A Semantic Web Framework for Java. <http://jena.sourceforge.net/>.
[48] Pellet. <http://pellet.owldl.com/>.
[49] FaCT. <http://owl.man.ac.uk/factplusplus/>.
[50] RacerPro. <http://www.racer-systems.com/>.
[51] Jess, The Rule Engine for Java Platform. <http://herzberg.ca.sandia.gov/>.
[52] Cyc. <http://www.cyc.com/>.
[53] N.J. Nilsson, Probabilistic logic, Artificial Intelligence 28 (1) (1986) 71–87.
[54] P. Sutton, L.E. Doyle, K.E. Nolan, A reconfigurable platform for cognitive networks, in: Proceedings of the CROWNCOM 2006, Mykonos Island, Greece, June 8–10, 2006.
[55] P. Mahonen, M. Petrova, J. Riihijarvi, M. Wellens, Cognitive wireless networks: your network just became a teenager, in: Proceedings of the INFOCOM 2006, Barcelona, Spain, April 23–29, 2006.
[56] J. Xie, I. Howitt, A. Raja, Cognitive resource management using multi-agent systems, in: Proceedings of the Fourth IEEE Consumer Communications and Networking Conference, January 11–13, 2007, pp. 1123–1127.
[57] V. Marojevic, X. Reves, A. Gelonch, Cooperative resource management in cognitive radio, in: Proceedings of the IEEE International Conference on Communications, June 2007, pp. 5939–5944.
[58] D. Raychaudhuri, N.B. Mandayam, J.B. Evans, B.J. Ewy, S. Seshan, P. Steenkiste, CogNet – an architectural foundation for experimental cognitive radio networks within the future internet, in: Proceedings of the First ACM/IEEE International Workshop on Mobility in the Evolving Internet Architecture, San Francisco, CA, 2006, pp. 11–16.
[59] M. Pitchaimani, B.J. Ewy, J.B. Evans, Evaluating techniques for network layer independence in cognitive networks, in: Proceedings of the ICC, June 24–28, 2007, pp. 6527–6531.
[60] The Future of the Internet: A Compendium of European Projects on ICT Research Supported by the EU Seventh Framework Programme for RTD, European Communities, 2008.
[61] FCC, ET Docket No. 03-322, Notice of Proposed Rule Making and Order, December 2003. <http://www.scribd.com/doc/1125166/Federal-Communications-Commission-FCC03322A1>.
[62] I.F. Akyildiz, W.Y. Lee, M.C. Vuran, S. Mohanty, NeXT generation/dynamic spectrum access/cognitive radio wireless networks: a survey, Computer Networks 50 (12) (2006) 2127–2159.
[63] I.F. Akyildiz, Won-Yeol Lee, M.C. Vuran, S. Mohanty, A survey on spectrum management in cognitive radio networks, IEEE Communications Magazine 46 (4) (2008) 40–48.
[64] Cognitive. <http://www.merriam-webster.com/dictionary/cognitive>.
[65] A. Jamalipour, Cognitive heterogeneous mobile networks, IEEE Wireless Communications 15 (3) (2008) 2–3.
[66] J. Bradshaw, Software Agents, AAAI Press/The MIT Press, 1997, ISBN 0262522349.
[67] P. Jackson, Introduction to Expert Systems, Addison-Wesley International Computer Science Series, 1986, ISBN 0201142236.
[68] T. Magedanz, K. Rothermel, S. Krause, Intelligent agents: an emerging technology for next generation telecommunications, in: Proceedings of the Infocom 96, San Francisco, CA, USA, March 24–28, 1996.
[69] N. Nisan, T. Roughgarden, E. Tardos, V.V. Vazirani, Algorithmic Game Theory, Cambridge University Press, 2007, ISBN 0521872820.
[70] E. Turban, J.E. Aronson, T.P. Liang, Decision Support Systems and Intelligent Systems, seventh ed., Prentice Hall, 2004, ISBN 0130461067.
[71] H. Arsham, Leadership Decision Making. <http://home.ubalt.edu/ntsbarsh/opre640/partXIII.htm>.
[72] T. Mitchell, Machine Learning, McGraw Hill, 1997, ISBN 0070428077.
[73] D.D.Clark, J. Wroclawski, K.R. Sollins, R. Braden, Tussle in the cyberspace: defining tomorrow's internet, in: Proceedings of the SIGCOMM 2002, Pittsburgh, PA, USA, August 19–23, 2002.
[74] V. Jacobson, A new way to look at networking, Google Tech Talks, August 30, 2006. <http://video.google.com/videoplay?docid=-6972678839686672840> (visited on January 2008).
[75] D. Boscovic, Topic introduction, in: Proceedings of the Autonomic Communication and Wireless Cognitive Networks Panel, CROWNCOM 2006, Mykonos, Greece, June 8–10, 2006. <http://www.autonomic-communication.org/publications/doc/AC-CognitiveRadio-PanelReport-v2.pdf> (visited on February 2008).
[76] D. Fensel, F. van Harmelen, B. Andersson, P. Brennan, H. Cunningham, E. Della Valle, F. Fischer, Z. Huang, A. Kiryakov, T.K. Lee, L. Schooler, V. Tresp, S. Wesner, M. Witbrok, N. Zhong, Towards LarKC: a platform for web-scale reasoning, in: Proceedings of the Second IEEE International Conference on Semantic Computing, Santa Clara, CA, USA, August 4–7, 2008.
[77] M.M. Gaber, A. Zaslavsky, S. Krishnaswamy, Mining data streams: a review, ACM Sigmod Record 34 (2) (2005) 18–26.
[78] H. Kargupta, Thoughts on human emotions, communication breakthroughs, and the next generation of data mining, in: Proceedings of the NGDM'07, Baltimore, USA, October 10–12, 2007. <http://www.cs.umbc.edu/%7Ehillol/PUBS/Papers/ngdm07_kargupta.pdf> (visited on September 2008).
[79] Management framework for open systems interconnection (OSI) for CCITT applications, ITU-T X.700. <http://www.itu.int/rec/T-REC-X.700/en> (visited on March 2008).
[80] S. Bauer, G. Lee, I. Chakraborty, X. Brucker, X. Yang, B. Leong, J. Wroclawski, The personal router, in: Proceedings of the MOBICOM'02, Atlanta, GA, USA, September 23–28, 2002.
[81] P. Faratin, J. Wroclawski, G. Lee, S. Parsons, The personal router: an agent for wireless access, in: Proceedings of the American Association of Artificial Intelligence Fall Symposium, Falmouth, MA, 2002, pp. 13–21.
[82] C. Fortuna, M. Mohorcic, Advanced access architecture for efficient service delivery in heterogeneous wireless networks, in: Proceedings of the International Workshop on Cognitive Networks and Communications, Hangzhou, China, August 25–27, 2008.
[83] End-to-End Efficiency. <http://www.eurescom.de/activities/EU_Projects/e3.asp> (visited on September 2008).
[84] Exposing the Features in IP version Six Protocols that Can Be Exploited/Extended for the Purposes of Designing/Building Autonomic Networks and Services. <http://www.efipsans.org/> (visited on September 2008).
[85] Self-Optimisation and Self-Configuration in Wireless Networks. <http://www.fp7-socrates.org/> (visited on September 2008).
[86] Autonomic Internet. <http://ist-autoi.eu/autoi/> (visited on September 2008).
[87] ARAGORN – Adaptive, Reconfigurable, Access and Generic Interfaces for Optimization in Radio Networks. <http://www.ict-aragorn.eu/index.html> (visited March 2008).
[88] European Future Internet Portal. <http://www.future-internet.eu/> (visited on March 2008).
[89] M. Sherman, A.N. Mody, R. Martinez, C. Rodriguez, R. Reddy, IEEE standards supporting cognitive radio and networks, dynamic spectrum access, and coexistence, Wireless Communications 46 (7) (2008) 72–79.
[90] Technical Sub-committee on Cognitive Networks. <http://www.eecs.ucf.edu/tccn/index.html> (visited in February 2008).
[91] IEEE 802.21. <http://ieee802.org/21/> (visited on March 2008).
[92] IEEE 802 LAN/MAN Standards Committee. <http://ieee802.org/22/> (visited on March 2008).

Carolina Fortuna received her B.Sc. degree in Electrical Engineering from the University of Cluj-Napoca, Romania, in 2006. Currently, she is a Ph.D. student and a Junior Researcher at the Department of Communication Systems at the Jožef Stefan Institute. Her research interests are in the field of cognitive networks, wireless networks, machine learning, high altitude platforms and intrusion detection systems.

Mihael Mohorcic received B.Sc., M.Sc. and Ph.D. degrees in Electrical Engineering from the University of Ljubljana, Slovenia, in 1994, 1998 and 2002, respectively, and M.Phil. degree in Electrical Engineering from University of Bradford, UK, in 1998. He is a research fellow in the Department of Communication Systems at the Jozef Stefan Institute. In 1996/1997, he spent 12 months as a Visiting Scientist at University of Bradford, Bradford, UK. His research interests include development and performance evaluation of network protocols and architectures for mobile and wireless communication systems, and resource management in terrestrial, stratospheric and satellite networks.