Towards Cognitive Internet: an Evolutionary Vision

F. Granelli, D. Kliazovich, and N. Malheiros

Abstract The requirement to support an always increasing number of networking technologies and services to cope with context uncertainties in heterogeneous network scenarios leads to an increase of operational and management complexity of the Internet. Autonomous communication protocol tuning is then crucial in defining and managing the performance of the Internet. This chapter presents an evolutionary roadmap of communication protocols towards cognitive Internet in which the introduction of self-aware adaptive techniques combined with reasoning and learning mechanisms aims to tackle inefficiency and guarantee satisfactory performance even in complex and dynamic scenarios.

In this survey, we overview and compare existing adaptive protocol stack solutions, review the principles of cross-layer design as well as the agent-based and AI based self-configuration solutions. The fundamental principles of cognitive protocols, such as adaptation, learning, and goal optimization, are presented along with implementation examples. Finally, the chapter discusses future research on the topic.

————————————————

Fabrizio Granelli
DISI – University of Trento, Italy, e-mail: granellil@disi.unitn.it

Dzmitry Kliazovich
University of Luxembourg, Luxembourg, e-mail: dzmitry.kliazovich@uni.lu

Neumar Malheiros
University of Campinas, Brazil, e-mail: ncm@ic.unicamp.br

1 Introduction

Communication networks play a central role in our daily life. Advanced wireless technologies and sophisticated network applications have changed the way we collaborate and share information. Today, the Internet represents the communication infrastructure of the modern society as well as a platform for the delivery of any kind of service (video on demand, e-commerce, social networks, etc.).

However, dealing with the complexity and dynamic of large scale networks is a challenging task. As the Internet represents a complex global interconnection of heterogenous end systems and networks, issues about its performance are rising and becoming increasingly relevant. The Internet architecture was not designed to support the Quality of Service (QoS) requirements of advanced multimedia applications and the varying channel conditions and mobility of wireless communication systems. The lack of adaptability and cross-layer mechanisms imposes limitations on internet protocols. In addition, current network management solutions face scalability and performance issues, as they are incapable of optimize resource utilization over long end-to-end paths and across heterogenous communication environments.

Academic and industry researches have been investigating how to provide adaptation capabilities to network protocols to optimize system-wide performance in a decentralized way. However, given the small and diverse time-scale of events and measurements in today's Internet, a specific requirement is related to autonomous operation (i.e. not requiring direct human intervention). Based on such principle, in this survey, we present a potential roadmap of the evolution toward a Cognitive Internet, in which distributed reasoning and learning mechanisms provide self-configuration and self-optimization capabilities to heterogenous networks elements.

This chapter is organized as follows. In the next section, we briefly describe some limitations of current Internet architecture and discuss the motivation for adaptive solutions. In Section 3, we overview well know techniques that enable design of adaptive protocol stack solutions. In Section 4, we focus on cognitive solutions to provide self-management functionalities. Then, we conclude the paper and discuss future research on the topic in Section 5.

2 Historical Perspective

2.1 Legacy TCP/IP

Core protocols of the TCP/IP stack were designed decades ago based on academic research requirements. The main design guidelines on which the Internet was built include layering, packet switching and the principle of keeping the complexity in the end systems at the edge of the network. These solutions are very elegant and effective, and they are still operational in today's Internet. However, a number of

inherited limitations prevent them to deliver the degree of efficiency, scalability and security dictated by current business and scientific requirements in networking.

The limitations of TCP/IP protocols have been described in the literature from several perspectives [18, 32, 37]. The main issues regard the lack of built-in security and support to mobility. A classic example is the poor performance of the traditional TCP congestion control mechanism in wireless environments. Although QoS solutions have been extensively investigated, there is no reliable and scalable approach to guarantee the stringent requirements of the advanced data-intensive network applications over the Internet. Moreover, the limited interaction among layers and lack of adaptation capabilities impact severely on service performance.

Several mechanisms have been proposed to overcome TCP/IP protocol drawbacks. As can be observed from the discussions in [31, 32, 37], the development of solutions for the next generation Internet addresses various issues such as service- and content-centric converged networks, new addressing schemes to support mobility, spectrum-efficient radio access, built-in security, and context-aware autonomic management capabilities. These solutions have been implemented if the form of extensions of the core architecture, affecting the transparency and simplicity of the original Internet design. As a consequence, ongoing discussion is aimed at the possibility or need for a new "clean" architecture for the Internet.

2.2 Motivation for Adaptation

Current Internet is built through the interconnection of different heterogeneous communication networks. This requires network administrators to deal with always increasing costs of operation and management. The ultimate solution would be to make management not necessary any more. In this context, the systems would be able to automatically manage themselves based on the high level administrative policies. This is called Autonomic Computing paradigm [10]. The self management of autonomic systems is based on the principles of context awareness, self-configuration, self-optimization, self-healing, and self-protection.

Obviously, the idea of autonomous control is far from original. The novelty in this paradigm is the holistic view of "autonomicity", and the focus of what is being automated. The goal is not automate system operation, but to automate its management functions. When the autonomic system is a communication network it is called an Autonomic Network [35]. The autonomic elements should be able to adapt themselves to constantly changing networks conditions in order to avoid performance degradation with minimum human intervention. The autonomic behavior must be guided by high level rules defined accordingly business and administrative policies.

The development of autonomic network management systems is a subject of considerable research and industrial interest [9, 36]. Network protocols can be adjusted during runtime using adaptive mechanisms, which make it a promising approach to deal with management complexity resulting from network heterogeneity and dynamics.

3 Adaptive TCP/IP: Enabling Technologies

In this section, we review the key technologies that enable the introduction of self-adaptation within the TCP/IP protocol stack. Those technologies are envisaged to build the basis on top of which to deploy cognitive TCP/IP solutions. The enabling technologies to support this evolution are:

- Cross-layer design, or cross-layering, providing suitable communication infrastructure for information and commands exchange among layers/protocols and network nodes;
- Distributed and agent-based solutions, enabling to re-allocate functionalities and features within the network;
- AI-based reasoning and learning, enabling the Internet to "think" and adapt;
- Architectures to support adaptive protocols, providing proper management environments.

As described in the following sections, cognitive networking employs cross-layer design, reasoning, and learning algorithms to provide system-wide network optimization through decentralized adaptation mechanisms.

3.1 Cross-layer Design

The large variety of optimization solutions requiring information exchange between two or more layers of the protocol stack raises an important issue concerning implementation of different cross-layer solutions inside TCP/IP protocol reference model, their coexistence and interoperability, requiring the availability of a common cross-layer signaling model. This model defines the implementation principles for the protocol stack entities implementing cross-layer functionalities and provides a standardized way for ease of introduction of cross-layer mechanism inside the protocol stack. In the following, we review several cross-layer signaling paradigms that have been proposed by the research community.

3.1.1 Interlayer Signaling Pipe

One of the first approaches used for implementation of cross-layer signaling is revealed by Wang *et al.* [39] as interlayer signaling pipe, which allows propagation of signaling messages layer-to-layer along with packet data flow inside the protocol stack in bottom-up or top-down manner, as illustrated in Fig. 1. An important property of this signaling method is that signaling information propagates along with the data flow inside the protocol stack and can be associated with a particular packet incoming or outgoing from the protocol stack.

Two methods are considered for encapsulation of signaling information and its propagation along the protocol stack from one layer to another: packet headers or

packet structures. Packet headers can be used as interlayer message carriers. In this case, signaling information included into an optional portion of IPv6 header [7], follow packet processing path and can be accessed by any subsequent layer. One of the main shortcomings of packet headers is in the limitation of signaling to the direction of the packet flow, making it not suitable for cross-layer schemes which require instant communication with the layers located on the opposite direction. Another drawback of packet headers method is in the associated protocol stack processing overhead, which can be reduced with packet structures method.

With packet structures signaling information is inserted into a specific section of the packet structure. Whenever a packet is generated by the protocol stack or successfully received from the network interface, a corresponding packet structure is allocated. This structure includes all the packet related information such as protocol headers and application data as well as internal protocol stack information such as network interface id, socket descriptor, configuration parameters and other. Consequently, cross-layer signaling information added to the packet structure is fully consistent with packet header signaling method but with reduced processing. Moreover, employment of packet structures does not violate existing functionality of separate layers of the protocol stack. In case the cross-layer signaling is not implemented at a certain layer, this layer simply does not fill nor modify the corresponding parts of the packet structure and does not access cross-layer parameters provided by the other layers. Another advantage of packet structure method is that standardization is not required, since the implementation could vary between different solutions.

3.1.2 Direct Interlayer Communication

Direct Interlayer Communication (Fig. 2) proposed in [39] aims at improvement of interlayer signaling pipe method by introducing signaling shortcuts performed out of band. In this way, the proposed Cross-Layer Signaling Shortcuts (CLASS) approach allows non-neighboring layers of the protocol stack to exchange messages, without processing at every adjacent layer, thus allowing fast signaling information delivery to the destination layer. Along with reduced protocol stack processing overhead, CLASS messages are not related to data packets and thus the approach can be used
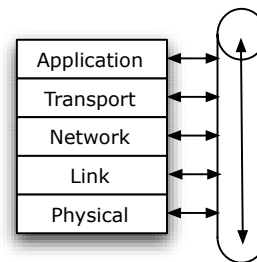

Fig. 1 Interlayer Signaling Pipe.

for bidirectional signaling. Nevertheless, the absence of this association is twofold since many cross-layer optimization approaches operate on per-packet basis, i.e. delivering cross-layer information associated with a specific packet traveling inside the protocol stack.

One of the core signaling protocols considered in direct interlayer communication is Internet Control Message Protocol (ICMP) [33]. Generation of ICMP messages is not constrained by a specific protocol layer and can be performed at any layer of the protocol stack. However, signaling with ICMP messages involves operation with heavy protocol headers (IP and ICMP), checksum calculation, and other procedures which increase processing overhead. This motivates a "lightweight" version of signaling protocol CLASS which uses only destination layer identification, type of event, and related to the event data fields. However, despite the advantages of direct communication between protocol layers and standardized way of signaling, ICMP-based approach is mostly limited by request-response action – while more complicated event-based signaling should be adapted. To this aim, a mechanism which uses callback functions can be employed. This mechanism allows a given protocol layer to register a specific procedure (callback function) with another protocol layer, whose execution is triggered by a specific event at that layer.

3.1.3 Central Cross-layer Plane

Implemented in parallel to the protocol stack, the Central Cross-Layer Plane is probably the most widely known cross-layer signaling architecture. In [4], the authors propose a shared database that can be accessed by all layers for obtaining parameters provided by other layers and providing the values of their internal parameters to other layers, as illustrated in Fig. 3. This database is an example of passive Central Cross-Layer Plane design: it assists in information exchange between layers but does not implement any active control functions such as tuning internal parameters of the protocol layers.

Similar approach is presented by the authors of [12], which introduces a central cross-layer plane called Cross-layer Server able to communicate with protocols at different layers by means of Clients. This interface is bidirectional, allowing Cross-layer server to perform active optimization controlling internal to the layer parameters.
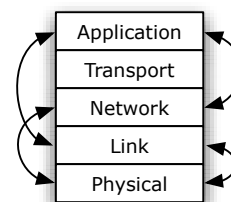

Fig. 2 Direct Interlayer Communication.

3.1.4 Network-wide Cross-Layer Signaling

Most of the above proposals aim at defining cross-layer signaling between different layers belonging to the protocol stack of a single node. However, several optimization proposals exist which perform cross-layer optimization based on the information obtained at different protocol layers of distributed network nodes. This corresponds to network-wide propagation of cross-layer signaling information, which adds another degree of freedom in how cross-layer signaling can be performed, as illustrated in Fig. 4.

Among the methods overviewed above, packet headers and ICMP messages can be considered as good candidates. Their advantages, underlined in the single-node protocol stack scenario, become more significant for network-wide communication. For example, the way of encapsulating cross-layer signaling data into optional fields of the protocol headers almost does not produce any additional overhead and keeps an association of signaling information with a specific packet. However, this method limits propagation of signaling information to packet paths in the network. For that reason, it is desirable to combine packet headers signaling with ICMP messages, which are well suited for explicit communication between network nodes.

One of the early examples of cross-network cross-layering is the Explicit Congestion Notification (ECN) presented in [34]. It realizes in-band signaling approach by marking in-transit TCP data packet with congestion notification bit. However, due to the limitation of signaling propagation to the packet paths this notification need to propagate to the receiver first, which echoes it back in the TCP ACK packet outgoing to the sender node. This unnecessary signaling loop can be avoided with explicit ICMP packets signaling. However, it requires traffic generation capabilities form network routers and it consume bandwidth resources.
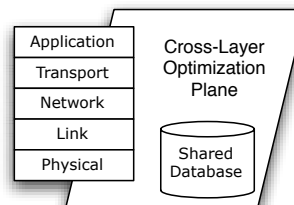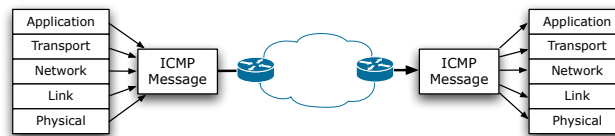

Fig. 3 Central Cross-Layer Plane.

Fig. 4 Network-wide Cross-Layer Signaling.

An example of adaptation of Central Cross-Layer Plane-like architecture to the cross-network cross-layer signaling is presented in [20]. The chapter suggests the use of a network service which collects parameters related the wireless channel located at the link and physical layers, and then provides them to adaptive mobile applications.

A unique combination of local and network-wide cross-layer signaling approaches called Cross-Talk is presented in [40]. CrossTalk architecture consists of two cross-layer optimization planes. One is responsible for organization of cross-layer information exchange between protocol layers of the local protocol stack and their coordination. Another plane is responsible for network-wide coordination: it aggregates cross-layer information provided by the local plane and serves as an interface for cross-layer signaling over the network. Most of the signaling is performed in-band using packet headers method, making it accessible not only at the end host but at the network routers as well. Cross-layer information received from the network is aggregated and then can be considered for optimization of local protocol stack operation based on the global network conditions.

Main problems associated to deployment of cross-layer signaling over the network, also pointed in [23], include security issues, problems with non-conformant routers, and processing efficiency. Security considerations require the design of proper protective mechanism avoiding protocol attacks attempted by non-friendly network nodes by providing incorrect cross-layer information in order to trigger certain behavior. The second problem addresses misbehavior of network routers. It is pointed out that, in 70% of the cases, IP packets with unknown options are dropped in the network or by the receiver protocol stack. Finally, the problem with processing efficiency is related to the additional costs of the routers? hardware associated with cross-layer information processing. While it is not an issue for the low-speed links, it becomes relevant for high speeds where most of the routers perform simple decrement of the TTL field in order to maintain high packet processing speed.

3.2 Distributed and Agent-based Solutions

The possibility to abstract "atomic" functions from a specific protocol layer and execute them in the network is the base for distributed protocol stacks architectures (see Fig. 5). The design process is composed of the following procedures: abstraction, detachment, connection, and execution.

3.2.1 Abstraction

Before a specific function or a set of functions of the protocol stack can be distributed over the network, they should be abstracted and detached from the protocol stack of the host node.

Identification of the functions to be abstracted depends on the optimization goal and is performed on a case-by-case basis. However, as a general recommendation, an abstraction should be performed with non-time critical functions which work on packet basis and do not require continuous access to the internal kernel structures. Ideally, abstracted functions should fit into a single functional block which operates at a packet flow basis and requires minimum or no input from the host protocol stack. The output of the abstracted functional block should be applied to the packet flow (for example, controlling a single bit in a packet header), trying to avoid the requirement for direct communication with the host protocol stack.

Examples of protocol stack functional blocks that could be easily abstracted include TCP ACK generation module, header compression, IP security related functionalities, congestion related packet drop notification, advertise window adjustment in TCP, and many others.

3.2.2 Detachment

Once the identified set of functions is abstracted as a standalone functional block within a protocol layer, it can be detached and moved into the network. This procedure requires a certain level of cooperation from network elements (routers, switches, or gateways). In particular, network element can be considered "friendly" to the proposed Distributed Protocol Stack if they provide an environment able to
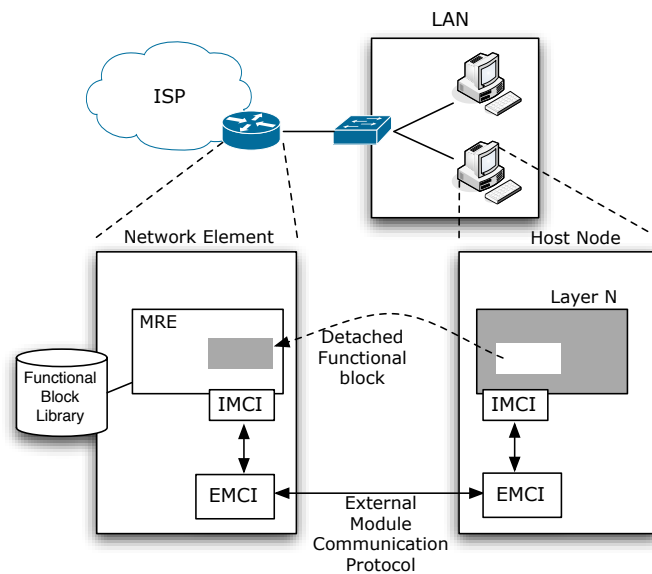

Fig. 5 Distributed protocol stack architecture.

support execution of the detached functional blocks – the Module Running Environment (MRE) – as an extension of their protocol stack.

MRE provides universal ways for registration and execution of different functional blocks. For example, it may provide a set of standard API functions which can be used by the host node to first transfer the abstracted functional block realized in the set of instructions understood by MRE (module description script language), and then register and run the transferred module.

Alternatively, avoiding the need for module transfer and registration procedures, functional blocks could be chosen from functional block library implemented at the network element. Execution of such blocks at the network element could be controlled by the host node or configured by network operator.

3.2.3 Communication

Communication between the detached functional block with the host protocol stack is performed using a Module Connection Interface (MCI), which is designed to provide communication between the detached functional block and the host protocol stack.

MCI is composed of two components:

- Internal Module Connection Interface (IMCI) connects the detached functional block with MRE at the network element side, while at the host node it provides communication interface with the protocol layer the functional block has been detached.
- External Module Connection Interface (EMCI) component provides communication between the detached functional block and the host protocol stack across the network with the use of External Module Communication Protocol (EMCP).

The main idea behind CMI separation into internal and external parts is designed for the purpose of module communication overhead reduction. In particular, EMCI components could be implemented at the lower layers or the protocol stack, leading to fewer header overhead and faster processing. The communication with the IMCI located at the protocol layer where the detachment was performed is performed locally within the protocol stack and it thus does not consume network resources.

3.2.4 Execution

Execution of the detached functional block can be triggered by the host node using MRE module installation primitives, or can be configured and running by the base station - without requiring interaction with the host node. In the latter case, the base station is responsible for notifying its clients with the information related to the list of functional blocks available.

Nevertheless, it is also important to consider the case of operating with clients which are unaware of functional blocks running at the base station or clients which

do not support such operation. Operation of the detached functional blocks should be performed in a transparent way, causing no communication performance degradation.

3.3 AI-based Reasoning and Learning

A set of solutions existing in the literature focus on the complexity of the optimization task, by proposing solutions based on Artificial Intelligence (AI). In this framework, emphasis is on the reasoning and learning processes of a cognitive network, i.e. on identifying suitable algorithms to understand the relationships among the network parameters with minimal a-priori knowledge. Such approaches can be based on fuzzy logic, reinforcement learning, or go beyond traditional AI towards bio-inspired operation [11]. The following paragraphs provide a brief review of AI-based solutions. For a more comprehensive and detailed analysis, the reader should refer to [13].

Expert systems, i.e., systems aiming to store human experts' knowledge in a specific field, represent a useful framework to perform reasoning in cognitive networks – provided the problem to be solved is characterized by a limited number of variables. However, the potentially narrow domain of application, typical of expert systems, clashes with the concept of a cognitive network architecture, which should aim to reason across a variety of diverse domains.

Heuristic optimization algorithms, like simulated annealing, genetic algorithms or swarm intelligence, are often used to automatically identify optimal solutions, and could be employed as alternative reasoning methods. However, such techniques should be preferred when the environment is well-known and the problem is centralized, rather than in distributed scenarios like the Internet.

Neural networks are often considered as a standard artificial intelligence technique and, thus far, have been applied to a wide range of applications including cognitive networks. Their main drawback lies in the fact that they are black boxes: once a neural network reaches a solution, its inner structure does not necessarily reflect the motivation behind that outcome, i.e. the existing relationships among the variables of a system are not reflected by the configuration of the neural network that led to the solution. Therefore, if the purpose is to gain some insights into a networks internals, neural networks can hardly represent the optimal solution.

Bayesian networks are another reasoning tool traditionally associated with artificial intelligence, with the capability of representing causal relationships among variables of a given problem and of being applied where knowledge is not certain. As they are based on directed acyclic graphs, their major limitation lies in their impossibility to deal with causality loops. Similar issues apply to Markov random fields and Markov logic networks, even tough Markov random fields (and all models based on them) suffer less from the limitation peculiar to Bayesian networks about loop-free networks. It is also worth noting that the undirected nature of such structures prevents them from handling induced dependencies.

Fuzzy Cognitive Maps (FCMs) are mathematical structures for modeling dynamical systems. They emphasize the causal relationships among the variables of a system and upon those they base reasoning. An example of the usage of FCMs in networking is presented in [14]. Updating techniques are based on Hebbian learning, according to which connections between concepts that are activated together should be given more weight.

3.4 Architectures to Support Adaptive Protocols

Several approaches for adaptation through autonomic management have been proposed in the literature [30]. In [19], Jennings *et al.* proposed an autonomic network management architecture called FOCALE. This architecture makes use of information and ontological modeling to enable the system to learn and reason about itself and its environment. Such knowledge, embedded within system information models, is used by policy-based management systems to automatically configure network elements in response to changing environmental context. This realizes an autonomic control loop in which the system senses changes and enforces management actions accordingly.

The Autonomic Network Architecture (ANA) project [1] resulted in the design of a novel autonomic network architecture which provides generic networking abstractions and communication primitives to support network adaptability. As described by the authors, ANA was designed as a meta-architecture to support evolution and adaptation of novel networking mechanisms. ANA is a generic development framework and an execution environment for the development and testing of autonomic functionalities. In that work, the authors illustrate key flexibility features of ANA, such as, support to address-agnostic applications and node mobility within the so-called network compartments.

3.5 Discussion

In this section, we described key technologies that enable the design of adaptive mechanisms to optimize network performance. It is important to note that these technologies are complementary to each other and can be used together in the design of sophisticated self-management solutions. For example, a self-managed transport protocol may be designed by the combination of AI algorithms and cross-layer architecture. The AI algorithms can provide reasoning capabilities for optimal protocol tuning, while the cross-layer approach can provide effective system performance monitoring and analysis. Table 1 summarizes the main features provided by those technologies to support self-management capabilities.

Table 1 Enabling technologies for adaptive protocols.

Technology	Provided features
Cross-layer design	Interlayer communication to support improved performance analysis and decision-making.
Distributed and agent-based solutions	Support to dynamic protocol stacks.
AI-based algorithms	Enable the design of learning and reasoning mechanisms.
Autonomic network architectures	Support to knowledge representation, signaling, and performance monitoring.

4 The Evolution to Cognitive Protocols

Although researchers have well described requirements and architectures for the design of autonomic systems, concrete algorithms for the realization of self-management functionality remains a challenge. The design of decentralized adaptive mechanisms able to optimize system wide performance is a quite complex task.

Learning and reasoning techniques have emerged as a promising approach to provide adaptive capabilities to communication protocols. As far as we know, that was first introduced in the construct known as the Knowledge Plane, as proposed by Clark *et al.* [5]. They described a new goal for the next generation Internet: "the ability of the network to *know* what it is being asked to do, so that it can more and more take care of itself".

Another important proposal is the concept of Cognitive Networks [38]. As an evolution of the concept of cognitive radio [17, 29], the paradigm of cognitive networking combines cognitive algorithms, cooperative networking, and cross-layer design for the provisioning of real-time optimization of complex communication systems [22]. Indeed, cognitive radio focuses on the tuning of parameters at the physical and link layers to provide efficient spectrum sharing, while cognitive networking expands the dynamic tuning of parameters to a system-wide scale to improve overall network performance [15].

There are several proposals of cognitive network architectures in the literature [21, 16]. The Software Programmable Intelligent Network (SPIN) presented [26] merges concepts of IP, PSTN, cellular, and ad hoc networks for overcoming the fundamental limitations of IP networks. The SPIN architecture consists of three planes interconnected by layer-2 transport infrastructure: the forwarding plane responsible for switching and monitoring, the control/management plane controlling the forwarding plane devices targeting flow optimization based on the received measurements, and the cognitive plane providing intelligence for and administration of the entire system.

Demestichas *et al.* present in [8] a platform (m@ANGEL) based on autonomic computing principles to provide seamless cognitive connectivity in heterogeneous wireless access networks. That work discusses business level issues and describes the architecture that provides management intelligence like context monitoring, de-

scription of profiles/agreements, resource brokerage, and configuration negotiation and implementation. The focus of m@ANGEL platform is exclusively devoted to bring cognitive functionalities into beyond-3G access networks. Most of the reconfiguration and cognitive functionalities are concentrated at the base stations. The structure of the access network consists of two planes: the infrastructure plane, which includes reconfigurable elements, and the management plane, composed of m@ANGEL entities responsible monitoring and control.

The Cognitive Complete Knowledge Network (CogNet) was proposed by Manoj *et al.* in [28]. It is a cross-layer approach aimed at extracting useful information from large amounts of network observations through inference algorithms and statistical learning techniques. The network state observations can be gathered through direct measurement or from peer nodes. The obtained knowledge enables optimal decisions in controlling network operation in order to improve the efficiency of resource management and the overall network performance.

In [25], Kousaridas *et al.* argue that next generation Internet architecture should be based on cognitive behavior by proposing a novel hierarchical feedback-control cycle in order to provide self-management capabilities. The proposed approach (called Self-NET) encompasses a hierarchical distribution of cognitive cycles in order to address self-organization and dynamic reconfiguration of network elements.

A cognitive network management architecture was proposed by Bouet *et al.* in [2]. The authors use software agents and artificial intelligence algorithms to build a distributed cognitive management framework (called CNM). Such framework includes communication, discovery and topology services, and uses a Fuzzy Logic-based inference system to support decision-making mechanisms. An important contribution of that work was to show the feasibility of embedding cognition into wireless networks. The paper discusses an implementation of the proposed cognitive architecture and its deployment within a heterogeneous wireless access network. The deployed solution was applied to two management functions, namely dynamic coverage control and capacity optimization.

In [27], Malheiros *et al.* present a feasible and effective solution for cognitive self-configuration of communication protocols. They propose a cognitive approach for dynamic reconfiguration of protocol parameters in order to avoid performance degradation as a consequence of changing network conditions. The proposed cognitive framework, called CogProt, provides runtime adjustment of protocol stack configuration parameters. The core of CogProt is a cross-layer cognitive plane, as illustrated in Fig. 6. CogProt periodically reconfigures the parameters of interest based on acquired knowledge to improve system-wide performance. This dynamic reconfiguration process is implemented through a cognitive feedback which includes learning and reasoning mechanisms.

CogProt can be applied to a wide range of protocol parameters in different layers. As a proof of concept, the framework was illustrated for the cognitive configuration of TCP congestion window evolution as presented in [24]. The congestion window increase factor (α) controls the increase of TCP congestion window after each RTT period. Controlling the TCP window evolution allows the adjustment of network utilization, protocol fairness, and the level of network congestion. Higher α values

are desirable in high bandwidth-delay network with low or moderate congestion levels, but should be avoided otherwise. However, there is no effective way for a network node to determine, in advance, available network bandwidth and the level of congestion at the end-to-end path between the sender and the receiver. Therefore, it is not possible to define an optimal value for α.

In that case study, CogProt was applied to the adaptation of the window increase factor which is adjusted in runtime based on the TCP goodput experienced in the immediate past. Both simulation and testbed experiments demonstrate that the proposed cognitive framework is able to improve average TCP performance under changing network conditions. The goal is to improve the performance of the congestion avoidance mechanism of standard TCP New Reno protocol for which the default value of the window increase factor is 1. We compared the TCP NewReno performance with fixed α values and dynamically adjusted α values (dynamically reconfigured by the cognitive mechanism). CogProt was allowed to vary the α parameter in the range $[1,5]$. For fixed value of α, performance degrades under changing conditions. However, CogProt is able to keep performance close to the optimal one for the scenarios with individual flows and it outperforms configurations with any fixed values of α, as shown in Figure 7. These results demonstrate the benefit of CogProt dynamic adaptation capabilities to avoid performance degradation on varying network conditions.

In [3], CogProt was used to design a novel mechanism for rate adaptation in wireless networks. The distributed mechanism enhances network element with self-configuration functionality to dynamically adapt the MAC data rate. It is able to quickly react to changes on channel conditions in order to avoid performance degradation with fair resource sharing among nodes.
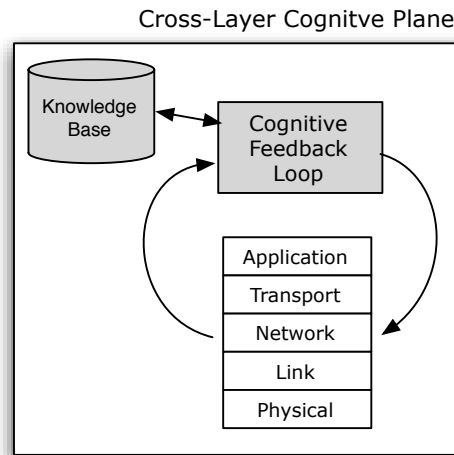

Fig. 6 The Cognitive Plane architecture.

Another CogProt-based solution was proposed in [6]. In that work, the authors present a mechanism for cognitive optimization of multiple link layer parameters in wireless networks. That solution exploits the cognitive adaptation techniques to maintain link layer performance at the optimal level during runtime. Its effectiveness is demonstrated by tuning the parameters of the CSMA-CA protocol. Both simulation and experimental results confirm the benefits of the proposed cognitive adaptation strategy and the ability to maintain optimal configuration of link layer parameters, even under highly dynamic network conditions.

CogProt is a decentralized framework. Each network node is allowed to independently decide on its own protocol setup to best match the current network conditions. However, network nodes may share the knowledge and make collaborative reconfiguration decisions. To support them, the framework includes a centralized Cognitive Information Service (CIS) which fosters the exchange of cognitive information among the nodes belonging to the same network segment. However, when CIS service is not available the nodes can still share cognitive information in a completely decentralized way. CogProt provides both self-configuration during runtime and initial setup of protocol parameters. A cross-layer cognitive plane is in the core of self-configuration process at the network node level. It implements a control feedback loop requiring network nodes to build a knowledge base based on the evidenced average performance. Such performance information is used to periodically adjust the parameter of interest. CogProt does not require modifications of standardized protocol operation and messaging. Thus, it is transparent to the rest of the network and can be deployed incrementally. The proposed framework can be used in the design of self-configuration mechanisms that are able to improve the average performance of network protocols with low degree of complexity and at low computational cost.

Table 2 summarizes the aforementioned cognitive approaches. We identify the target layers for which the solutions were designed and which of them are based on a decentralized approach.
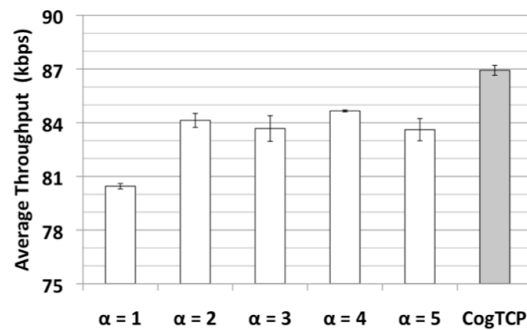

Fig. 7 Average TCP throughput for fixed α values and CogProt.

Table 2 Summary of Cognitive Approaches.

Approach	Target Layers	Decentralized
(SPIN) [26]	All	√
m@ANGEL [8]	Lower	√
CogNet [28]	All	×
Self-NET [25]	All	√
CNM [2]	Lower	√
CogProt [27]	All	√

5 Conclusion

In this work, we presented an overview of several approaches to cope with the complexity of managing dynamic and heterogeneous network environments. We emphasize the need for adaptive mechanism which should allow network elements to reconfigure themselves in order to avoid performance degradation in face of changing network conditions. Such self-management mechanisms must be decentralized and provide system-wide performance optimization.

In this scenario, cognitive techniques represent a promising approach to design the future Internet protocols. Several cognitive frameworks and mechanisms have been proposed to realize distributed self-management functionality. Nevertheless, there are challenges to be overcome in order to fulfill the requirements of scalable and decentralized adaptive network architectures. In this context, it is possible to identify the following issues as opportunities for future research on the topic:

- *Analytical Models*: Cognitive solutions aims at supporting dynamic adjustment of network systems to provide performance optimization. There is a lack of theoretical frameworks and analytical models which demonstrate whether the proposed solutions can efficiently converge to optimal operational points. Cognitive algorithms for learning and reasoning may have various control parameters. Such analytical models would also be very useful to determine optimal values for the control parameters themselves in specific network scenarios or applications.
- *Proactive Self-Management*: As far as we know, the proposed cognitive architectures and protocols provide network systems only with reactive reconfiguration functionalities. Network elements can adapt to changing environment conditions in an attempt to maximize resource utilization, but only after performance degradation has already affected the system. It is worth to investigate novel self-management capabilities that are able to provide pro-active management. This way, network elements would be able to learn how to prevent performance degradation and security attacks.
- *Knowledge Representation*: In order to implement learning and reasoning algorithms in an efficient and scalable way, we need light and flexible knowledge information models. Such models must be independent of specific architectures and technologies, and facilitate cognitive information sharing among managed elements.

References

1. G. Bouabene, C. Jelger, C. Tschudin, S. Schmid, A. Keller, and M. May. The autonomic network architecture (ana). *Selected Areas in Communications, IEEE Journal on*, 28(1):4–14, 2010.
2. M. Bouet, G. Nguengang, V. Conan, A. Kousaridas, P. Spapis, and N. Alonistioti. Embedding cognition in wireless network management: an experimental perspective. *Communications Magazine, IEEE*, 50(12):150–160, 2012.
3. L. Chaves, N. Malheiros, E. Madeira, I. Garcia, and D. Kliazovich. A cognitive mechanism for rate adaptation in wireless networks. In J. Strassner and Y. Ghamri-Doudane, editors, *Modelling Autonomic Communications Environments*, volume 5844 of *Lecture Notes in Computer Science*, pages 58–71. Springer, 2009.
4. Kai Chen, Samarth H. Shah, and Klara Nahrstedt. Cross-layer design for data accessibility in mobile ad hoc networks. *Wireless Personal Communications*, 21(1):49–76, 2002.
5. David D. Clark, Craig Partridge, J. Christopher Ramming, and John T. Wroclawski. A knowledge plane for the internet. In *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, SIGCOMM '03, pages 3–10, New York, NY, USA, 2003. ACM.
6. Jorge Lima de Oliveira Filho, Dzmitry Kliazovich, Fabrizio Granelli, Edmundo Madeira, and NelsonL.S. da Fonseca. Cogmac: a cognitive link layer for wireless local area networks. *Wireless Networks*, pages 1–11, 2013.
7. S. Deering and R. Hinden. Internet Protocol, Version 6 (IPv6) Specification. IETF RFC 2460, 1998.
8. P. Demestichas, V. Stavroulaki, D. Boscovic, A. Lee, and J. Strassner. m@ANGEL: autonomic management platform for seamless cognitive connectivity to the mobile Internet. *IEEE Communications Magazine*, 44(6):118–127, 2006.
9. S. Dobson, S. Denazis, A. Fernández, D. Gaïti, E. Gelenbe, F. Massacci, P. Nixon, F. Saffre, N. Schmidt, and F. Zambonelli. A survey of autonomic communications. *ACM Trans. Auton. Adapt. Syst.*, 1(2):223–259, 2006.
10. Simon Dobson, Roy Sterritt, P. Nixon, and M. Hinchey. Fulfilling the vision of autonomic computing. *Computer*, 43(1):35–41, 2010.
11. Falko Dressler and Ozgur B. Akan. A survey on bio-inspired networking. *Computer Networks*, 54(6):881 – 900, 2010. ¡ce:title¿New Network Paradigms¡/ce:title¿.
12. Karim M. El Defrawy, Magda S. El Zarki, and Mohamed M. Khairy. Proposal for a cross-layer coordination framework for next generation wireless systems. In *Proceedings of the 2006 international conference on Wireless communications and mobile computing*, IWCMC '06, pages 141–146, New York, NY, USA, 2006. ACM.
13. C. Facchini. *Bridging the gap between theory and implementation in cognitive networks: developing reasoning in today's networks*. PhD thesis, University of Trento, Italy, 2011.
14. Christian Facchini, Oliver Holland, Fabrizio Granelli, Nelson L.S. da Fonseca, and Hamid Aghvami. Dynamic green self-configuration of 3G base stations using fuzzy cognitive maps. *Computer Networks*, 57(7):1597 – 1610, 2013.
15. Carolina Fortuna and Mihael Mohorcic. Trends in the development of communication networks: Cognitive networks. *Comput. Netw.*, 53(9):1354–1376, 2009.
16. A. Georgakopoulos, K. Tsagkaris, D. Karvounas, P. Vlacheas, and P. Demestichas. Cognitive networks for future internet: Status and emerging challenges. *Vehicular Technology Magazine, IEEE*, 7(3):48–56, 2012.
17. Simon Haykin. Cognitive radio: brain-empowered wireless communications. *Selected Areas in Communications, IEEE Journal on*, 23(2):201–220, 2005.
18. R. Jain. Internet 3.0: Ten problems with current internet architecture and solutions for the next generation. In *Military Communications Conference, 2006. MILCOM 2006. IEEE*, pages 1–9, 2006.
19. B. Jennings, S. van der Meer, S. Balasubramaniam, D. Botvich, M. O Foghlu, W. Donnelly, and J. Strassner. Towards autonomic management of communications networks. *IEEE Communications Magazine*, 45(10):112–121, 2007.

20. B.-J. Kim. A network service providing wireless channel information for adaptive mobile applications. i. proposal. In *IEEE International Conference on Communications (ICC)*, volume 5, pages 1345–1351, 2001.

21. D. Kliazovich, F. Granelli, and N.L.S. da Fonseca. Survey on signaling techniques for cognitive networks. In *Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), 2011 IEEE 16th International Workshop on*, pages 5–10, 2011.

22. D. Kliazovich, F. Granelli, and N. L. S. Da Fonseca. Architectures and cross-layer design for cognitive networks. In *Handbook on Sensor Networks*, pages 3–24. World Scientific Publishing Co, Inc., 2010.

23. D. Kliazovich, F. Granelli, S. Redana, and N. Riato. Cross-Layer Error Control Optimization in 3G LTE. In *IEEE Global Telecommunications Conference (GLOBECOM '07)*, pages 2525–2529, 2007.

24. D. Kliazovich, N. Malheiros, N. L. S. da Fonseca, F. Granelli, and E. Madeira. CogProt: A Framework for Cognitive Configuration and Optimization of Communication Protocols. In *The 2nd Int'l Conference on Mobile Lightweight Wireless Systems (MOBILIGHT)*, 2010.

25. A. Kousaridas, C. Polychronopoulos, N. Alonistioti, A. Marikar, J. Mödeker, A. Mihailovic, G. Agapiou, I. Chochliouros, and G. Heliotis. Future internet elements: cognition and self-management design issues. In *Proceedings of the 2nd International Conference on Autonomic Computing and Communication Systems*, Autonomics '08, pages 13:1–13:6, ICST, Brussels, Belgium, Belgium, 2008. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

26. S. M. Lake. Cognitive networking with software programmable intelligent networks for wireless and wireline critical communications. In *IEEE Military Communications Conference (MILCOM)*, volume 3, pages 1693–1699, 2005.

27. N. Malheiros, D. Kliazovich, F. Granelli, E. Madeira, and N. L. S. da Fonseca. A cognitive approach for cross-layer performance management. In *IEEE Global Telecommunications Conference (GLOBECOM 2010)*, pages 1–5, 2010.

28. B. S. Manoj, R. R. Rao, and M. Zorzi. Cognet: a cognitive complete knowledge network system. *Wireless Communications, IEEE*, 15(6):81–88, 2008.

29. J. Mitola and Jr. Maguire, G.Q. Cognitive radio: making software radios more personal. *Personal Communications, IEEE*, 6(4):13–18, 1999.

30. Z. Movahedi, M. Ayari, R. Langar, and G. Pujolle. A survey of autonomic network architectures and evaluation criteria. *Communications Surveys Tutorials, IEEE*, 14(2):464–490, 2012.

31. J. Pan, S. Paul, and R. Jain. A survey of the research on future internet architectures. *Communications Magazine, IEEE*, 49(7):26–36, 2011.

32. Subharthi Paul, Jianli Pan, and Raj Jain. Architectures for the future networks and the next generation internet: A survey. *Computer Communications*, 34(1):2 – 42, 2011.

33. J. Postel. Internet control message protocol. IETF RFC 792, 1981.

34. K. Ramakrishnan, S. Floyd, and D. Black. The Addition of Explicit Congestion Notification (ECN) to IP. IETF RFC 3168, 2001.

35. D. Raymer, Sven van der Meer, and J. Strassner. From autonomic computing to autonomic networking: An architectural perspective. In *Engineering of Autonomic and Autonomous Systems, 2008. EASE 2008. Fifth IEEE Workshop on*, pages 174–183, 2008.

36. N. Samaan and A. Karmouch. Towards autonomic network management: an analysis of current and future research directions. *IEEE Communications Surveys & Tutorials*, 11(3):22–36, 2009.

37. P. Stuckmann and R. Zimmermann. European research on future internet design. *Wireless Communications, IEEE*, 16(5):14–22, 2009.

38. R.W. Thomas, D.H. Friend, L.A. DaSilva, and A.B. MacKenzie. Cognitive networks: adaptation and learning to achieve end-to-end performance objectives. *Communications Magazine, IEEE*, 44(12):51–57, 2006.

39. Qi Wang and M.A. Abu-Rgheff. Cross-layer signalling for next-generation wireless systems. In *IEEE Wireless Communications and Networking (WCNC 2003)*, volume 2, pages 1084–1089, 2003.

40. R. Winter, J. H. Schiller, N. Nikaein, and C. Bonnet. Crosstalk: cross-layer decision support based on global knowledge. *Communications Magazine, IEEE*, 44(1):93–99, 2006.