Noname manuscript No.
(will be inserted by the editor)

Can Artificial Intelligence meet the Cognitive Networking Challenge?

Submitted May 2011 to *EURASIP Journal on Wireless Communications and Networking*, Special Issue on Ten Years of Cognitive Radio: State of the Art and Perspectives.

Karen Zita Haigh[1], Craig Partridge[2]

[1] Raytheon BBN Technologies, 5775 Wayzata Blvd, St. Louis Park, MN 55416 e-mail: `khaigh@bbn.com`
[2] Raytheon BBN Technologies, 10 Moulton Street, Cambridge, MA 02138 e-mail: `craig@bbn.com`

The date of receipt and acceptance will be inserted by the editor

Abstract Recent developments in Software Defined Radio technology have created the opportunity to develop networks that are, in principle, highly adaptable and effective under a much wider range of operating conditions than currently possible, but few researchers are able to fully exploit this new flexibility. Cognitive radio networks require multiple interacting capabilities for situation assessment, planning and learning, and are therefore a rich application area for Artificial Intelligence (AI) technology. AI techniques enable real-time, context-aware adaptivity at the core of the cognitive networking vision. This paper briefly discusses some of the AI techniques that can and have been leveraged in this domain. The goal is to encourage further research in this area so that we can overcome the most significant challenges that remain in cognitive networking.

1 Introduction

The demand is increasing for networking technologies that support robust communication and functionality under challenging operating conditions. Traditionally, network configurations are hand-tuned and remain static during operations. However, since user needs and operating conditions both change over time, *cognitive networks* must be designed that are aware of their performance needs, determine if their needs are being met, and revise system configurations to better meet their needs.

Recent developments in software defined radio technology have opened up the opportunity to develop networks that are, in principle, highly adaptable and effective under a much wider range of operating conditions than currently possible [3, 9, 28]. However, while these tools provide new flexibility, few researchers are able to fully exploit it. This paper briefly discusses some of the Artificial Intelligence techniques that can (and should) be leveraged in this domain, and highlights specific cases of successful implementations.

A Mobile Ad hoc NETwork (MANET) is a type of ad hoc network that consists of "mobile platforms... which are free to move about arbitrarily... At a given point in time... wireless connectivity in the form of a random, multi-hop graph or 'ad hoc' network exists between the nodes" [15]. MANETs are characterized by dynamic topologies, bandwidth-constrained, variable capacity links, energy-constrained operation, and limited physical security. A MANET is needed for self-forming, self-configuring, and self-healing operation where the media and communications channels undergo rapid changes (e.g., over free space optical, RF, and underwater acoustic links) and nodes freely enter and leave the network. MANETs are not needed when links are unchanging, e.g., GEO satellite links, LOS microwave tower links, fiber optics, Ethernet and wired infrastructure.

In the cognitive network vision, the network adapts to these continuous changes rapidly, accurately, and automatically. A cognitive network must

- *identify and forecast* network conditions, including communications environment and performance,
- *adapt* to constantly changing conditions, including participants, tasks, and conditions,
- *learn* from prior experiences so that it doesn't make the same mistakes,
- *balance* the needs of many users—military, commercial, civilian, and government—while conforming to official regulations and Policies such as rules-of-engagement.

Intelligent cognitive radios require multiple interacting capabilities for situation assessment, planning and learning. Cognitive networks require those capabilities to operate cooperatively in a distributed, diverse environment. Artificial Intelligence (AI) techniques have addressed these challenges in many domains, including several examples in networking. AI enables the real-time, context-aware adaptivity that is required by cognitive networks.

2 A Brief Intro to Artificial Intelligence

Given that many readers of this paper will come from the communications and networking community, it may be useful to provide a little context.

Artificial Intelligence (AI) is the branch of computer science concerned with the automation of intelligent behaviour [39], usually associated with human thinking such as decision making, problem solving and learning [2]. In 1950, Alan Turing proposed the *Turing Test* [71] which called for a human judge to interact through a terminal to both another human and a computer; if the judge cannot tell which is which, then the machine is said to pass the test and would be considered intelligent. The term *Artificial Intelligence* was coined in 1956 by notable researchers including Herb Simon, Allen Newell, John McCarthy and Marvin Minsky, at the Dartmouth Conference [43]. McCorduck [44] presents a comprehensive history of AI, while Russell and Norvig [61] describe AI techniques appropriate for building decision-making agents that make rational actions for their given context.

AI draws techniques from a broad variety of fields including mathematics, psychology, economics, and control theory. AI has a huge variety of subfields, including planning and scheduling, machine learning, knowledge engineering and fusion, and constraint reasoning.

Natural language processing, speech recognition, machine vision and robotics all had origins in AI. Practical AI successes are often pulled into their own domains, leaving AI researchers to deal with the unsolved problems. Larry Tesler is often misquoted as having said "AI is whatever hasn't been done yet" [29]. Tesler corrects the quote to "Intelligence is whatever machines haven't done yet" [67].

> *The Odd Paradox*
> *Practical AI successes, computational programs that actually achieved intelligent behavior, were soon assimilated into whatever application domain they were found to be useful in, and became silent partners alongside other problem-solving approaches, which left AI researchers to deal only with the "failures," the tough nuts that couldn't yet be cracked.*
> *McCorduck, 2004 [44]*

3 Networking Problems Amenable to AI

Artificial Intelligence techniques could plausibly be used in any Networking problem that involves some form of situation assessment and/or decision making. The following list is a small sample of some of the specific domain problems that AI techniques may be able to help solve:

- Cyber Security[1]
- Network Configuration and Planning
- Network Control and Coordination
- Policy and Constraint Management
- Performance Analysis

[1] Because cyber security is a huge research area unto itself, we do not further address these issues in this paper.

This section describes appropriate roles for AI techniques as taken from the perspective of networking needs.

3.1 Potential AI roles from the perspective of the networking cognition loop

AI systems are often described using the cognition loop of "Sense, Plan, Act, and Learn," similar to the OODA loop of "Observe, Orient, Decide, Act" [6,7] shown in Figure 1. Note that the OODA loop lacks the key functions of Planning and Learning required in a cognition loop: without these capabilities it is hard to argue that a system is "cognitive." Joe Mitola proposed the OOPDAL cognition loop specifically for cognitive radio that effectively merges these: "Observe, Orient, Plan, Decide, Act and Learn" [47]. Figure 2 sketches the interaction of these steps and indicates the tasks they perform.

In the **Observe** step, the system must *collect* the raw sensor data and then cluster that data into hypothesized events. It then *validates* that data by paring the set of

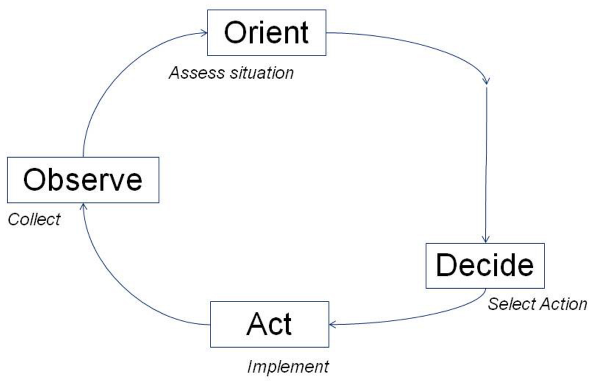

Fig. 1 The OODA cycle does not require cognitive capabilities.

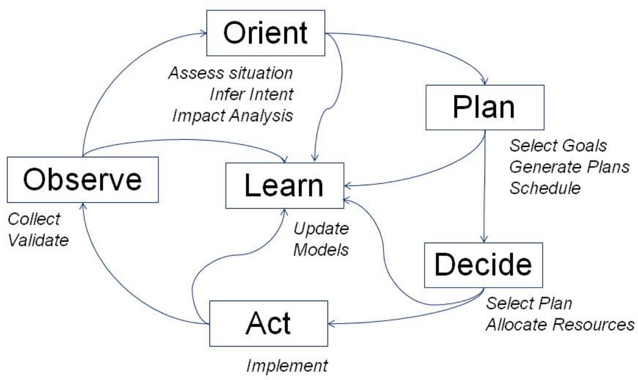

Fig. 2 The OOPDAL cycle highlights the tasks that cognitive systems perform.

hypothesized events down to the set of likely events. Network modules need to expose internal state, including current values of controllable parameters, current values of observations and monitored parameters, and current activity. AI modules need to identify faulty sensors, and integrate values across the layers in the stack. The AI also needs to collect patterns of activity and track performance trends. The AI on a single node may choose to share observations with other nodes, and may need to incorporate indirect observations received from other nodes.

In the **Orient** step, the system must *assess the situation.* To understand the entire situation, this process involves inferring what else might be true if the event has happened. *Intent Inference* infers the goals of other agents based on observations of their actions. This activity may require recognizing that a formerly friendly node has been compromised. Finally, *impact analysis* examines the potential ramifications of the current situation, including predicting future states given possible dynamism in the domain and selected courses of action. Depending on these ramifications, some situations can be safely ignored, others may require information gathering, others may require coordination with teammates, others may require changing activity to adapt to conditions or avoid threats, and others may require explicit threat suppression. The network modules should expose any derived computations, including analyses of performance or network state, and estimates of future conditions if possible. The AI needs to interpret these observations and identify potential factors (or root causes) of situations, and compute progress toward performance goals. The AI then needs to estimate future conditions and the likelihood of achieving goals, so that it can decide on the urgency of responding to problems. Any conclusions should be presented to the network modules where possible, including for example indicating bottlenecks or unreliable nodes.

In the **Plan** step, the system must first *identify goals* to be achieved (and when). This step involves managing the multi-objective performance criteria for all current tasks and upcoming reservations. (Note that network-wide goals are often different from node-specific goals.) The system must then *generate plans* to achieve those goals. Planning involves causality reasoning, conditional planning (or "what-if" analyses), temporal reasoning, constraint reasoning, and resource management. Given the current state of the network, the planner must predict the effect of potential actions on the future state of the network. Plans can be generated at multiple timescales, handling immediate concerns at a fine-granularity, and longer-term issues at a coarse-granularity (potentially allowing negotiation with other nodes). Because cognitive networks operate in a multi-objective space, a planner may generate several plans that tradeoff meeting one objective for another, ideally on a pareto-optimal curve. Finally, the system must *schedule* to allocate specific resources to specific activities over time. Planning & scheduling often operate iteratively, in the sense that tasks cannot be selected for a plan if no schedule exists. Traditionally, network modules contain significant scheduling capabilities, but the planning capabilities are implicit in the software, that is, the human network engineer performs the planning.

The **Decide** step *selects* among the candidate plans and schedules, and then *allocates* computational and radio resources. Given how quickly the domain changes, a potential approach is to select actions that are common at the beginning of "most" of the candidate plans, because the plan is likely to be revised as conditions change.

In the **Act** step, the system implements the chosen activities. This may include setting values of parameters or replacing running modules or waveforms. Note that if the radio or its software has actions that can be selected, these should be exposed to the plan/decide steps otherwise the system will be unable to use the full breadth of system capability.

In the **Learn** step, the system uses experience to update models so that the other steps can make more accurate forecasts. The system can learn human and application-level behaviour, including node mobility and data-access patterns (which applications or humans or roles are accessing the network, and what each of them needs and when). The system can learn environmental conditions, including connectivity patterns and geographical factors. It can also learn capabilities of other nodes, including capacity, reliability, and functionality. The planner can use this understanding of neighbours to bias routing decisions. The learner can use both explicit human feedback (e.g., QoS is below par), or empirical performance data (e.g., statistics mapping parameter settings to QoS).

3.2 Potential AI roles from the perspective of network characteristics

Communications networks have numerous characteristics that challenge network designers, for which AI techniques would be appropriate and effective solutions. These characteristics include:

Dynamic: Very few things in a MANET environment are static. Military missions change, user requirements change, users join or leave the network, hardware fails, and mobility causes continuous fluctuations in connectivity. Machine learning techniques can recognize that the environment has changed, and update models; they can also generalize from previously-seen conditions to infer reasonable solutions for new conditions. This generalization capability is critical because cognitive networks rarely operate under the identical conditions; we can thus collect baseline data in a relatively controlled environment (e.g., the lab or

Adaptation for cognitive networks means that a network trained in a desert can learn how to perform well under water.

a test field), and then expect reasonable operations when the network is actually fielded. Techniques for planning under uncertainty make choices that will be appropriate even as the domain changes.

Partially-observable: Many factors that affect communication cannot be observed. Few radios, for example, have a "fog" sensor. AI techniques are good at inferring missing data and generalizing a situation so that decisions make sense for current conditions.

Ambiguous observations: Detection and understanding of a change in situation is not always simple. For example, how does the system automatically tell the difference between short-term fade versus entering a building? AI techniques are good at recognizing ambiguity or low confidence, and can either gather more information to discriminate or make decisions appropriate for both conditions.

Resource constrained: Cognitive network nodes usually operate under a variety of resource constraints, including bandwidth, compute capabilities or energy. AI techniques are effective at scaling a solution to the platform they are operating on, and designing tasks that manage available resources effectively.

Diverse: Nodes in a MANET have a wide variety of capabilities, from small hand-held radios to large radios with satellite communications (satcom); these vary both in communications and compute power. This heterogeneity requires different solutions on different nodes. AI techniques consider diversity a benefit, as it allows resources to be managed in different ways.

Discrete: As a result of the limited communication and frequent disconnections, nodes have to make decisions locally, considering local requirements and constraints. Using local observations, local learning, and local decision-making simplifies the learning problem without compromising too much optimality. Key information, such as global network performance, can be shared across nodes when required.

Massive scale: There are roughly 600 observable parameters and 400 controllable parameters (possibly continuous-valued) to configure *per node*[2]. We thus have a distributed, heterogeneous, low-communication, partially-observable, high-latency optimization problem of approximately μ^{PN} choices per timestep[3]; one second would be a large timestep. Data mining and ML techniques are effective even on massive datasets; moreover incremental planning and learning techniques incorporate new information efficiently and rapidly.

Complex Access Policies: Due to the heterogeneous nature of the data and the nodes, access policies may restrict the set of nodes that are permitted to hold, transmit or receive specific data. Knowledge engineering techniques can represent policies as constraints, and then constraint reasoning techniques can find satisfying solutions quickly.

Multi-objective performance requirements: Networks are traditionally optimized for one thing, such as throughput, delay, or energy consumption. However, in a realistic cognitive network, multiple users have interacting requirements and policies, thus creating a complex multi-objective function that captures mission, situational and social standpoints [26]; it can include a wide variety of issues including bandwidth, application-level quality of service, energy, network connectivity, and security. Distributed planning and optimization approaches effectively modify behaviour to meet these requirements.

AI techniques are capable of addressing the full richness of these challenges in other domains. Moreover, there are single systems (e.g., robotics) that collectively address many of these challenges together. In the networking domain, however, AI techniques are just beginning to scratch the surface. We need to bring these techniques into the networking domain, and address them in depth.

Networks also have characteristics that have only been lightly addressed by the networking and AI communities.

Complex temporal feedback loops: Within a node, certain activities occur at very rapid speeds (e.g., between the Medium Access Control (MAC) and Physical layers) requiring very a very tight feedback loop to support cognitive control. Other activities (e.g., at the Routing layer) occur on a longer time-scale and cognitive control algorithms may need to take into account a wider range of factors in a slow feedback loop. Between nodes, there is yet a longer feedback loop between changes that are made and the effects that are observed in network-level performance. The variety of temporal loops and their dramatic speed differences means that correlating cause and effect of actions is particularly challenging.

Complex interactions: Networking parameters have deep, poorly-understood interactions with each other and with system performance. In many cases, human net-

[2] We include the ability to dynamically reconfigure the IP stack as control parameters; we model alternate configurations by creating a control parameter x for each available network module, where $x = 1$ when the module has been invoked, and $x = 0$ when the module is not operating [27]. No current system exposes all of these parameters; the highest known is about 100 parameters, of which 30 are controllable.

[3] P = number of parameters, N = number of nodes, and μ is the average number of values that a parameter can take.

work engineers can identify specific pair-wise interactions, such as increased power reduces battery life. However, most of these pair-wise interactions are carefully caveatted by the networking community, with exceptions or conditionals that are rarely observable or computable. AI has the potential to capture and model many more of these interactions than human network engineers could ever be capable of analyzing.

Heterogeneous Intercommunication: There is a very strong norm in the networking community that all nodes must be designed and (statically) configured to interoperate; typical ad hoc networks build a group of homogenous nodes. Cognitive networks break this assumption: each node can have an independent cognitive controller, and thus network nodes *may be heterogeneous, and may fall into in non-interoperable configurations.*[4] Heterogeneous configurations are a key enabler to dramatic improvements in network performance, and thus as AI techniques are slowly given greater access to network configuration, this challenge will be critical to solve.

ADROIT [70], by giving each node its own learning system, represented a *radical* departure from the traditional networking stance that requires homogeneous configurations: ADROIT was the first system to demonstrate how powerful a heterogeneous MANET can be. ADROIT avoided the possibility of catastrophic failure by giving the AI no access to parameters that can cause complete communications failure.

The first two of these challenges have been addressed in point solutions in the AI community, and specific instances in the networking community. The last challenge of automatic heterogeneous intercommunication has never been addressed by the AI community *or* the Networking community. Perhaps a collaborative research effort will give new insights, solutions, and capabilities.

4 AI Techniques in Networking

While almost any AI technique could potentially prove useful in a networking environment, certain techniques are more promising and/or have already produced interesting results. These include Knowledge Engineering, Planning and Scheduling, Machine Learning, Distributed AI and Multi-agent systems, including biologically-inspired approaches, and Game Theory. This section describes possible locations for AI in networks as taken from the perspective of AI techniques.

Knowledge Engineering aims to capture knowledge so that a computer system can solve complex problems [20]. Different knowledge representation approaches are used for different types of knowledge, and the different ways that it will be used. Much knowledge engineering work is concerned with constructing Ontologies. In the networking domain, this knowledge would include models of physics and signal propagation, constraints on the system, analysis of interactions, and rules of thumb (e.g., about how to configure the system). A formal ontology may help a cognitive system reason about how and when capabilities are interchangeable, e.g., recognizing that either of two metrics for computing Quality of Information may be used and that a metric for Quality of Service may be an appropriate replacement under some conditions. Semantics and representations are important considerations for cognitive networks [23,32]. Several researchers have developed knowledge bases and heuristic rules to optimize the network [23,35,56].

Planning and Scheduling techniques are appropriate for decision-making situations, where tasks need to be organized and coordinated to meet performance objectives, under resource constraints. In dynamic environments, the plan needs to be monitored because predictions about performance may have been inaccurate or the conditions have changed such that previously-selected actions are no longer appropriate. In these cases the strategy needs to be revised online. Multi-agent planning, dynamic programming, partially-observable Markov decision processes (POMDPs), constraint satisfaction, and distributed optimization algorithms are common techniques. Planning and scheduling techniques in networks can decide what content to move, where, when, and how, including power-aware computing, node activity and task scheduling, and network management. Scheduling packets and admission control may also benefit from these approaches.

Rathnasabapathy and Gmytrasiewicz show that routing protocols are conditional plans in the AI sense, and formulate routing as a multi-agent decision problem using POMDP's [57]. Chadha [10,11] created a self-organizing network management hierarchy that dynamically updates itself based on changes in connectivity or domain requirements. As an example task-allocation scheme, mobile ad hoc networks can benefit from pre-pulling or pre-pushing data towards the nodes at the edge of the network. Intelligent search mechanisms can similarly decide which nodes to use as resources for information [30, 76]. Chadha et al use machine learning, planning and domain expertise to dynamically select and place servers in MANETs [12]; Tapiador and Clark [66] combine genetic algorithms with policies for the same problem. Lau et al [34] use AI techniques for planning under uncertainty to estimate the best opportunities for communicating with other nodes. PNUTS [73] contains an adaptive scheduler for handling server queries.

Cognitive networks operate under a variety of environmental, operation, and application constraints. Moreover, solutions to traffic admission, scheduling and rout-

[4] The alternative is to have one cognitive controller for several nodes; while coordination issues are reduced, communication overhead increases dramatically and intelligent control is vulnerable to network partitions.

ing should be solved on-line and in a distributed way to cope with mobility and frequent topology changes. The MANET research community has extensively studied traditional constraint optimisation techniques based on Lagragian relaxation, Linear or Mixed Integer Programming. The AI community has developed more flexible and powerful problem solving techniques that hybridize search and constraint propagation, including dynamic choice point selection, decision variable ordering, probe backtracking and constraint-based search control [38,62]. These new techniques solve much broader challenges than traditional approaches, solving much larger problems and giving better adaptation to network mobility, heterogeneous resources, and resilience to spectrum interference, jamming, link breaks or node failures [24,54].

Machine Learning (ML) techniques aim to improve the performance of a system by observing the environment and updating models that describe the interactions of observables [33,45,46]. ML techniques are appropriate in every domain that is imperfectly modelled. Most complex domains (including networking) fall into this category. Moreover, because the set of all possible behaviors is too large to be covered by observed examples, the learner must generalize so that the learned model is useful for new (previously unseen) cases. ML techniques include artificial neural networks, support vector machines, clustering, explanation-based learning, induction, reinforcement learning, genetic algorithms, nearest-neighbour methods, and case-based learning. *Data Mining* techniques are a subset (or close cousin) to ML techniques, in that they identify patterns in large datastores. Data Mining results can be used in a ML system to improve its models. Machine learning differs from *Statistics* in that it generalizes from the observed data.

Dietterich and Langley [17] provide a good overview of ML techniques and how they could be applied to Cognitive Networks, but cite only one concrete example of a realized system in communications networking. Possibly the earliest use of ML in networking, Littman and Boyan [5] introduced a reinforcement-learning approach to routing in networks. Other researchers have extended this work to a wireless environment, to handle dynamic load, to manage energy and to plan node mobility [13,14,36,60,65]. MANET networks are often organized into cluster hierarchies to achieve performance guarantees [77]; ML techniques could be leveraged here. ML techniques could also be used to build patterns of users in forward-deployed enclaves: to understand the relationship between task (or role) and topics of interest, and when those files will be needed [79].

Another rich area for ML is learning how parameters interact with each other and with the domain. Rieser [58] and Rondeau [59] used genetic algorithms to tune parameters and design waveforms. The experiments show no data about how fast it works and moreover the learning appears to operate offline; Rieser states explicitly that it "may not be well suited for the dynamic environment where rapidly deployable communications systems are used." All demos involve one receiver, one transmitter, and one jammer, although in theory the approach should not be limited. Newman et al [51,50] similarly use genetic algorithms to optimize parameters in a simulated network; they also show no time results. Montana et al [49] used a genetic algorithms approach for parameter configuration in a wireline network that can find the 95% optimal solution in "under 10 minutes." Using the ML design of Haigh [27], ADROIT [70] used Artificial Neural Networks to model communications effectiveness of different strategies. They were the first to demonstrate ML in an on-line (real-time) real-world (not simulation) MANET. Each node in their distributed approach learns a different model of global performance based on local observations (i.e. no shared information), thus meeting MANET requirements for rapid local learning and decision making.

Distributed AI and *Multi-agent Systems* are concerned with finding distributed solutions for AI problems [22,53, 75]. Techniques address domains that have the following characteristics:

Discrete: Local goals and constraints
Deprived: Locally resource constrained
Distributed: Embedded in a physical world
Decentralized: Local decisions and local views of the environment (i.e., no centralized decision maker)
Diverse: Different capabilities and different roles
Dynamic: Changing task/mission and domain

DAI and MAS approaches generally decompose centralized techniques to make them appropriate for the decentralized environment, often with some calculation of the tradeoff between optimality and latency. While conceptually appropriate for the communications networking environment [21], these traditional techniques have to-date not acknowledge or address a key requirement for communications networks, namely that *the task being negotiated is the communications itself*. In other words, traditional AI has always assumed that that communication is "safe," negotiating and coordinating only the application-level tasks [42,48,78]; moreover they generally require massive communications with non-neighbours, universally do not support mobility (changing connections or constraints between the nodes), and universally do not support a changing objective function. These drawbacks are so significant that extensive research and redesign are required to make them applicable in this domain.

Biologically-inspired computing approaches are lightweight coordination mechanisms [72], and have been used for a variety of networking problems. AntHocNet [16] uses both proactive and reactive schemes to update the routing tables, and outperforms AODV. Konak et al [31] use particle swarm optimization and agents to improve network connectivity. Sesum-Cavic and Kühn use swarm intelligence for dynamic load balancing [64]. Parunak

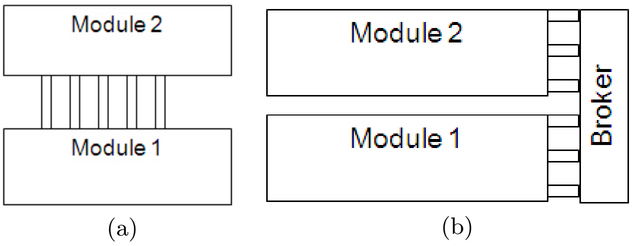

Fig. 3 (a) IP modules are usually tightly coupled, with one function in the API for each exposed parameter, e.g., `setTimeout(x)` (b) If IP modules used a consistent interface, e.g., `set("Timeout", x)`, we solve the $n \times m$ problem and facilitate later addition of cognition.

and Brueckner use a stigmergic approach to decide where to locate services on a MANET [55]. Biologically-inspired methods are often slower in reaction than conventional control systems, and may lose optimality, but can offer greater resilience.

Game theory is a branch of applied mathematics that is used for analyzing the interaction among agents whose decisions affect each other. Game theory is becoming a common formalism for studying strategic and cooperative interaction in multi-agent systems [18]. Applications of game theory to wireless communications have also received significant interest by the research community; Nisan et al [52] and Liu and Wang [36] present good introductions. Previous research includes enforcing fairness and thwarting selfish behavior in shared medium [40], multi-hop packet relaying [1,8], multi-carrier (OFDM) systems [4], MIMO [63], interactions between communicating nodes [37], and overlay networks [74].

5 An Environment for Cognitive Networking

AI techniques can be incorporated directly within traditional modules, or structured as additional modules that directly access IP-module APIs. However, we as a community will obtain the most effective results if we restructure the traditional IP-stack architecture. The critical issue is that networking software architectures rely on APIs that are carefully designed to expose each parameter separately, as sketched in Figure 3a. This approach to network configuration is not maintainable, for example as protocols are redesigned or new parameters are exposed. It is also not amenable to cognitive control. One issue is that there is no way to get a "directory" of the parameters that can be observed or controlled. Another issue is that there is no coordination mechanism: what happens if a cognitive controller wants to set the same parameter that another module wants to set?

To support AI-based control, we need an architecture more like the one sketched in Figure 3b, where each module exposes parameters in a consistent format, with functions such as `expose(paramName)`, `set(paramName,val)` and `get(paramName)`. In this structure, first shown in ADROIT [25,68], a *Broker* serves as a kind of system bus between the modular software and any entity that wishes to change how a running module behaves. So, for instance, if a timer parameter is to be adjusted, that request is sent to the Broker, which passes the request on to the relevant module. Communication goes through the Broker even for module-to-module requests. Anything that wishes to observe, monitor, or change the state of the radio must do so via a command relayed by the Broker. The Broker must provide directory services so that modules can find each other based on capabilities or exposed parameters. To ensure that no real-time network module is blocked while waiting for a response, the architecture design must support asynchronous requests to set or get parameter values, with corresponding upcalls, reporting and alerting functions. The Broker must notify interested parties of any changes in the radio's state or configuration. It is also useful for the design to include capabilities that allow a protocol to adapt within given constraints, particularly to support the rapid MAC/PHY layers.

One concern raised with this architecture is the reliance on a centralized Broker. The Broker is a mandatory part of the new system architecture, with failure of the Broker having grave consequences, and therefore each network module should have a failsafe default operation that work (presumably with degraded performance), even when the Broker is not functional. Another useful failsafe capability is support for a module to reject decisions that don't make sense, including for example constraint management mechanisms that ensure that settings are consistent.

Note that this architecture supports cognitive control, but *does not mandate it.* This software engineering change solves two difficult problems in (traditional) networking architectures. The first is known as the $n \times m$ problem: we wish to have one consistent interface to any and all network modules so that if it changes, or when additional modules are created, none of their controlling applications need to be modified (including other modules in the network stack, applications, cognitive capabilities, or even the user via a command-line interface). The second problem the Broker solves is that of coordination of control. Multiple controllers may be actively seeking to manage modules at the same time, and to avoid control battles, they need to know about each other.

This architecture was designed and successfully demonstrated in ADROIT [25,68], and the Broker software is available on an open-source basis.

6 Corollaries and Implications

Cognitive networks have the potiential to achieve capabilities not previously seen. There are several interesting implications that arise in this environment.

Benefits and scope of cross-layer design. Cognitive networking implies the ability to adapt many parameters across all of the layers in the stack. Traditional networking "cross-layer design" has generally meant two layers, and one or two parameters in each layer. To understand the potential impact of this broad cross-layer optimization, it is worth doing detailed drill-down walkthroughs, each focusing on how certain changes in parameters could produce novel changes in networking protocols and behavior under certain observed conditions [25, 69]. Simulation results for specific scenarios can demonstrate the power of broad cross-layer optimization. These walkthroughs also help the team understand how to make the new approach work better.

Adaptation for cognitive networks requires performing cross-layer optimization over all of the layers, each of which may have many exposed parameters.

Heterogeneous and non-interoperable nodes. A deeply-held tenet in networking is conformance to written protocol specifications; all nodes must always follow the protocol, and from this one can conclude that they will interoperate (but one cannot guarantee maximal performance). Further, most nodes are homogeneous. Cognitive controllers enable the network to become heterogeneous to the point of non-interoperability, resulting in possible failure of the nodes to communicate, but also enabling greater performance when managed correctly. The architecture design must explicitly address the mandate of maintaining connectivity, while allowing for heterogeneity. The ADROIT architecture included an "orderwire" bootstrap channel to be used when a node can not communicate with the rest of the network and an explicit "coordination manager" that decided what information to share with other nodes.

Capability Boundaries. In traditional networking approaches, there is a very clear boundary between application and network module, often corresponding to a user/kernel boundary with a widely known API (e.g., "BSD sockets"). Similarly, there is a clear boundary between controller and controllee. With the generic approach to exposing and controlling parameters, these boundaries blur. Any client of the Broker can choose to expose controllable parameters, and any client can choose to set another module's parameters. Thus, an application can choose to have complete visibility into the stack, or be told to back off by the network. While more flexible, we must take care that the additional complexity does not lead to unreliable systems. We expect that removing traditional conceptual restrictions will result in interesting and significant new ideas.

7 Conclusions

By dynamically changing their communications patterns based on the current conditions, cognitive networks can adapt to changes in infrastructure, optimize performance based on current user needs, and modify behavior to avoid communications difficulties or mitigate threats. They may even be able to communicate with non-cognitive radios or other legacy systems.

There are many powerful AI techniques that address knowledge engineering, situation assessment, planning, scheduling, and learning in distributed environments. AI techniques are ready to be challenged with this complex real-world domain, just as Networking requirements are reaching the limits of what can be done by human experts. We are at a nexus from which interesting ideas and capabilities will develop.

Similar to the gap that existed between the Information Theory and Networking communities 10 years ago [19], we need to consummate the union between AI and communications network research.

References

1. L. Anderegg and S. Eidenbenz. Ad hoc-VCG: a truthful and cost-efficient routing protocol for mobile ad hoc networks with selfish agents. In *Proc. of ACM annual International Conference on Mobile Computing and Networking (MobiCom)*, pages 245–259, 2003.
2. R. E. Bellman. *An Introduction to Artificial Intelligence: Can Computers Think?* (San Francisco, CA: Boyd & Fraser Publishing Company), 1978.
3. E. Blossom. GNU radio: tools for exploring the radio frequency spectrum. *Linux Journal*, 122, June 2004.
4. H. Bogucka. Game theoretic model for the OFDM water-filling algorithm with imperfect channel state information. In *Proc. IEEE International Conference on Communications*, pages 3814–3818, 2008.
5. J. A. Boyan and M. L. Littman. Packet routing in dynamically changing networks: A reinforcement learning approach. In *Advances in Neural Information Processing Systems (NIPS)*, pages 671–678. (San Mateo, CA: Morgan Kaufmann), 1994.
6. J. Boyd. Destruction and Creation, 1976. Unpublished Essay, available at `http://www.goalsys.com/books/documents/DESTRUCTION_AND_CREATION.pdf`.
7. J. Boyd. Patterns of conflict, December 1986. `http://www.ausairpower.net/JRB/patterns.ppt`.
8. L. Buttyan and J.-P. Hubaux. *Security and Cooperation in Wireless Networks: Thwarting Malicious and Selfish Behavior in the Age of Ubiquitous Computing.* (Cambridge, UK: Cambridge University Press), 2007.
9. A. Casimiro, J. Kaiser, and P. Veríssimo. An architectural framework and a middleware for cooperating smart components. In *Proc. of the 1st Conference on Computing Frontiers*, pages 28–39, Ischia, Italy, 2004.
10. R. Chadha and C.-Y. Chiang. Drama: Distributed policy management for MANETs. In *IEEE Workshop on Policies for Distributed Systems and Networks*, pages 235–237, June 2008.
11. R. Chadha and L. Kant. *Policy-Driven Mobile Ad Hoc Network Management.* (Somerset, NJ: John Wiley and Sons), 2007.

12. R. Chadha, A. Poylisher, B. Deb, M. Littman, and B. Sabata. Adaptive dynamic server placement in MANETs. In *IEEE Military Communications Conference (MILCOM 2005)*, pages 1083–1089, 2005.
13. Y.-H. Chang, T. Ho, and L. P. Kaelbling. Mobilized ad-hoc networks: A reinforcement learning approach. In *Proc. International Conference on Autonomic Computing (ICAC)*, 2004.
14. S. P. M. Choi and D.-Y. Yeung. Predictive Q-routing: A memory-based reinforcement learning approach to adaptive traffic control. In *Advances in Neural Information Processing Systems (NIPS)*, 1996.
15. S. Corson and J. Macker. Mobile ad hoc networking (MANET): Routing protocol performance issues and evaluation considerations, January 1999. RFC 2501.
16. G. Di Caro, F. Ducatelle, and L. M. Gambardella. AntHocNet: an adaptive nature-inspired algorithm for routing in mobile ad hoc networks. In *European Transactions on Telecommunications (Special Issue on Self-organiztion in Mobile Networking)*, volume 16, pages 443–455, 2005.
17. T. G. Dietterich and P. Langley. Machine learning for cognitive networks: Technology assessment and research challenges. In Mahmoud [41].
18. E. Elkind and K. Leyton-Brown. Introduction to the ai magazine special issue on algorithmic game theory. *AI Magazine*, Winter, 2010.
19. A. Ephremides and B. Hajek. Information theory and communication networks: An unconsummated union. *IEEE Transactions on Information Theory*, 44(6):2416–2434, October 1998.
20. E. A. Feigenbaum and P. McCorduck. *The fifth generation (1st ed.).* (Reading, MA: Addison-Wesley), 1983.
21. D. H. Friend, R. W. Thomas, A. B. MacKenzie, and L. A. DaSilva. Distributed learning and reasoning in cognitive networks: Methods and design decisions. In Mahmoud [41].
22. L. Gasser and M. N. Huhns. *Distributed Artificial Intelligence, vol 2.* (San Mateo, CA: Morgan Kaufmann), 1990.
23. A. Ginsberg, W. D. Horne, and J. D. Poston. The semantic side of cognitive radio. In Mahmoud [41].
24. C. Guettier, P. Jacquet, L. Viennot, and J. Yelloz. Automatic optimisation of reliable collaborative services in olsr mobile ad hoc networks. In *Proc. of IEEE Military Communications Conference (MilCom)*, October 2007.
25. K. Z. Haigh, T. S. Hussain, C. Partridge, and G. D. Troxel. Rethinking networking architectures for cognitive control. In *Microsoft Research's Cognitive Wireless Networking Summit.* Snoqualmie, WA, 2008.
26. K. Z. Haigh, O. Olofinboba, and C. Y. Tang. Designing an implementable user-oriented objective function for MANETs. In *IEEE International Conference On Networking, Sensing and Control*, London, U.K., April 2007.
27. K. Z. Haigh, S. Varadarajan, and C. Y. Tang. Automatic learning-based MANET cross-layer parameter configuration. In *Workshop on Wireless Ad hoc and Sensor Networks (WWASN2006)*, Lisbon, Portugal, 2006.
28. M. A. Hiltunen and R. D. Schlichting. The Cactus approach to building configurable middleware services. In *Proc. of the Workshop on Dependable System Middleware and Group Communication (DSMGC)*, Nuremberg, Germany, October 2000.
29. D. R. Hofstadter. *Gödel, Escher, Bach: An Eternal Golden Braid.* (New York, NY: Vintage Books), 1979.
30. V. Kalogeraki, D. Gunopulos, and D. Zeinalipour-Yazti. A local search mechanism for peer-to-peer networks. In *Proc. International Conference on Information and Knowledge Management (CIKM)*, pages 300–307. (New York, NY: ACM Press), 2002.
31. A. Konak, O. Dengiz, and A. E. Smith. Improving network connectivity in ad hoc networks using particle swarm optimization and agents. In *Wireless Network Design: Optimization Models.* (New York, NY: Springer), 2011.
32. V. J. Kovarik. Cognitive research: Knowledge representation and learning. In *Cognitive Radio Technology.* 2008.
33. P. Langley. *Elements of Machine Learning.* (San Mateo, CA: Morgan Kaufmann), 1995.
34. R. Lau, S. Demers, Y. Ling, B. Siegell, E. Vollset, K. Birman, R. van Renesse, H. Shrobe, J. Bachrach, and L. Foster. Cognitive adaptive radio teams. In *IEEE Sensor and Ad Hoc Communications and Networks*, pages 842–847, 2006.
35. H. Liu, A. Kershenbaum, and R. Van Slyke. Artificial intelligence applications to communication network design with bulk facilities. In *Proc. of the ACM annual conference on Communications (CSC)*, pages 345–350, Kansas City, Missouri, 1992. (New York, NY: ACM Press).
36. K. J. R. Liu and B. Wang. *Cognitive Radio Networking and Security: A Game-theoretic View.* (Cambridge, UK: Cambridge University Press), 2011.
37. X. Liu, G. Noubir, R. Sundaram, and S. Tan. SPREAD: Foiling smart jammers using multi-layer agility. In *Proc. of InfoCom Mini-Symposium*, 2007.
38. F. Lucas, C. Guettier, P. Siarry, A. de La Fortelle, and A.-M. Milcent. Constrained navigation with mandatory waypoints in uncertain environment. *International Journal of Information Sciences and Computer Engineering (IJISCE)*, 2, 2011. To appear.
39. G. F. Luger and W. A. Stubblefield. *Artificial Intelligence: Structures and Strategies for Complex Problem Solving.* (Palo Alto, CA: Benjamin Cummings), 1993. 2nd edition.
40. A. B. MacKenzie and S. B. Wicker. Stability of multipacket slotted Aloha with selfish users and perfect information. In *Proc. of IEEE International Conference on Computer Communication (InfoCom)*, 2003.
41. Q. Mahmoud, editor. *Cognitive Networks: Towards Self-Aware Networks.* (Somerset, NJ: John Wiley and Sons), 2007.
42. T. W. Malone and K. Crowston. The interdisciplinary study of coordination. *ACM Computing Surveys*, 26(1):87–119, March 1994.
43. J. McCarthy, M. Minsky, N. Rochester, and C. Shannon. A proposal for the Dartmouth summer research project on Artificial Intelligence. See `http://en.wikipedia.org/wiki/Dartmouth_Conferences`, 1955.
44. P. McCorduck. *Machines Who Think.* (Natick, MA: A K Peters, Ltd.), 2004.
45. R. S. Michalski, I. Bratko, and M. Kubat. *Machine Learning and Data Mining.* (Somerset, NJ: John Wiley and Sons), 1998.

46. T. Mitchell. *Machine Learning.* (New York, NY: McGraw-Hill), 1997. ISBN 0070428077.
47. J. Mitola III. Cognitive radio for flexible multimedia communications. In *IEEE International Workshop on Mobile Multimedia Communications (MoMuC)*, pages 3–10, 1999.
48. P. J. Modi, W.-M. Shen, M. Tambe, and M. Yokoo. ADOPT: Asynchronous distributed constraint optimization with quality guarantees. *Artificial Intelligence*, 161(1-2):149–180, 2005.
49. D. Montana, T. Hussain, and T. Saxena. Adaptive reconfiguration of data networks using genetic algorithms. In *Proc. Genetic and Evolutionary Computation Conference (GECCO)*, 2002.
50. T. Newman, J. Evans, and A. Wyglinski. Reconfiguration, adaptation and optimization. In A. M. Wyglinski, M. Nekovee, and T. Hou, editors, *Cognitive Radio Communications and Networks: Principles and Practice.* (New York, NY: Elsevier/North Holland), 2009.
51. T. R. Newman, R. Rajbanshi, A. M. Wyglinski, J. B. Evans, and G. J. Minden. Population adaptation for genetic algorithm-based cognitive radios. In *Proc. of ICST Conference on Cognitive Radio Oriented Wireless Networks and Communications (CROWNCOM).* (New York, NY: IEEE Press), May 2007.
52. N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani, editors. *Algorithmic Game Theory.* (Cambridge, UK: Cambridge University Press), 2007.
53. G. M. P. O'Hare and N. R. Jennings, editors. *Foundations of Distributed Artificial Intelligence (Sixth Generation Computer Technologies).* (Somerset, NJ: John Wiley and Sons), 1996.
54. W. Ouaja and B. Richards. Hybrid Lagrangian relaxation for bandwidth-constrained routing: Knapsack decomposition. In *Proc. ACM symposium on Applied computing*, SAC '05, pages 383–387. (New York, NY: ACM Press), 2005.
55. H. Parunak and S. Brueckner. Stigmergic learning for self-organizing mobile ad-hoc networks (MANETs). In *Proc. International Joint Conference on Autonomous Agents and MultiAgent Aystems (AAMAS).* 2004.
56. S. Pierre and H. Hoang. An Artificial Intelligence approach to improving computer communications network topologies. *Journal of the Operational Research Society*, 41(5):405–418, 1990.
57. B. Rathnasabapathy and P. Gmytrasiewicz. Formalizing multi-agent POMDP's in the context of network routing. In *Proc. IEEE Hawaii International Conference on System Sciences (HICSS)*, 2003.
58. C. J. Rieser. *Biologically Inspired Cognitive Radio Engine Model Utilizing Distributed Genetic Algorithms for Secure and Robust Wireless Communications and Networking.* PhD thesis, Virginia Tech, Blacksburg, VA, 2004.
59. T. W. Rondeau. *Application of Artificial Intelligence to Wireless Communications.* PhD thesis, Department of Electrical Engineering, Virginia Polytechnic Institute, Blacksburg, VA, September 2007.
60. B. K. Russell. *Learning-Based Route Management in Wireless Ad Hoc Networks.* PhD thesis, Department of Computer Science, Rutgers University, New Brunswick, NJ, USA, August 2008.
61. S. J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach.* (Upper Saddle River, NJ: Prentice-Hall), 1995.
62. H. E. Sakkout and M. Wallace. Probe backtrack search for minimal perturbations in dynamic scheduling. *Constraints Journal*, 5(4):359–388, 2000.
63. G. Scutari, D. Palomar, and S. Barbarossa. Competitive design of multiuser MIMO systems based on game theory: A unified view. *IEEE Journal on Selected Areas in Communications*, 26(7):1089–1103, 2008.
64. V. Sesum-Cavic and E. Kühn. Applying swarm intelligence algorithms for dynamic load balancing to a cloud based call center. In *IEEE Self-Adaptive and Self-Organizing Systems Workshop on Self-Adaptive Networking*, pages 42 –49. (New York, NY: IEEE Press), 2010.
65. D. Subramanian, P. Druschel, and J. Chen. Ants and reinforcement learning: A case study in routing in dynamic networks. In *Proc. International Joint Conference on Artificial Intelligence (IJCAI)*, pages 832–838. (San Mateo, CA: Morgan Kaufmann), 1997.
66. J. Tapiador and J. Clark. Learning autonomic security reconfiguration policies. In *Computer and Information Technology (CIT), 2010 IEEE 10th International Conference on*, pages 902–909, 2010.
67. L. Tesler. Adages & Coinages, 2011. `http://www.nomodes.com/Larry_Tesler_Consulting/Adages_and_Coinages.html`, cited 29 March 2011.
68. G. D. Troxel, E. Blossom, S. Boswell, A. Caro, I. Castineyra, A. Colvin, T. Dreier, J. B. Evans, N. Goffee, K. Z. Haigh, T. Hussain, V. Kawadia, D. Lapsley, C. Livadas, A. Medina, J. Mikkelson, G. J. Minden, R. Morris, C. Partridge, V. Raghunathan, R. Ramanathan, P. G. Rubel, C. Santivanez, T. Schmid, D. Sumorok, M. Srivastava, R. S. Vincent, D. Wiggins, A. M. Wyglinski, and S. Zahedi. Enabling open-source cognitively-controlled collaboration among software-defined radio nodes. *Computer Networks*, 52(4):898–911, March 2008.
69. G. D. Troxel, S. Boswell, A. Caro, I. Castineyra, A. Colvin, Y. Gabay, N. Goffee, K. Z. Haigh, T. Hussain, V. Kawadia, D. Lapsley, C. Livadas, A. Medina, J. Mikkelson, C. Partridge, V. Raghunathan, R. Ramanathan, P. Rubel, C. Santivanes, D. Sumorok, B. Vincent, and D. Wiggins. Adaptive dynamic radio open-source intelligent team (ADROIT): Architecture and design. Technical Report BBN-TR-8478, BBN Technologies, 2007.
70. G. D. Troxel, A. Caro, I. Castineyra, N. Goffee, K. Z. Haigh, T. Hussain, V. Kawadia, P. G. Rubel, and D. Wiggins. Cognitive adaptation for teams in ADROIT. In *IEEE Global Communications Conference*, pages 4868–4872, Washington, DC, November 2007. Invited.
71. A. Turing. Computing machinery and intelligence. *Mind*, LIX(236):433–460, October 1950.
72. P. Valckenaers, K. Hadeli, B. Saint Germain, P. Verstraete, and H. Van Brussel. MAS coordination and control based on stigmergy. *Computers in Industry*, 58(7):621–629, September 2007.
73. Y. Vigfusson, A. Silberstein, B. F. Cooper, and R. Fonseca. Adaptively parallelizing distributed range queries. *Proc. VLDB Endowment*, 2:682–693, August 2009.

74. Y. Wang and A. Nakao. On socially-inspired cooperative and efficient overlay network evolution based on group selection pattern. In *Proc. International Conference on Bio-Inspired Models of Network, Information and Computing Systems (BIONETICS)*. (Brussels, Belgium: ICST), 2008.
75. G. Weiss, editor. *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. (Cambridge, MA: MIT Press), 1999.
76. S. J. H. Yang, J. Zhang, L. Lin, and J. J. P. Tsai. Improving peer-to-peer search performance through intelligent social search. *Expert Systems with Applications*, 36(7):10312–10324, September 2009.
77. J. Yu and P. Chong. A survey of clustering schemes for mobile ad hoc networks. *Communications Surveys Tutorials, IEEE*, 7(1):32 –48, 2005.
78. X. Zhang, V. Lesser, and S. Abdallah. Efficient Management of Multi-Linked Negotiation Based on a Formalized Model. *Autonomous Agents and Multi-Agent Systems*, 10(2):165–205, 2005.
79. I. Zukerman, D. Albrecht, and A. Nicholson. Predicting users' requests on the WWW. In *Proc. of the International Conference on User Modeling (UM)*, 1999.