2

Cognitive Networks[1]

Ryan W. Thomas, Daniel H. Friend, Luiz A. DaSilva, and Allen B. MacKenzie

Virginia Tech, Blacksburg VA 24060

2.1 Introduction

Current data networking technology limits a network's ability to adapt, often resulting in sub-optimal performance. Limited in state, scope, and response mechanisms, the network elements (consisting of nodes, protocol layers, policies, and behaviors) are unable to make intelligent adaptations. Communication of network state information is stifled by the layered protocol architecture, making individual elements unaware of the network status experienced by other elements. Any response that an element may make to network stimuli can only be made in the context of its limited scope. The adaptations that are performed are typically reactive, taking place after a problem has occurred. In this chapter, we advance the idea of cognitive networks, which have the promise to remove these limitations by allowing networks to observe, act, and learn in order to optimize their performance.

Cognitive networks are motivated by complexity. Particularly in wireless networks, there has been a trend towards increasingly complex, heterogeneous, and dynamic environments. While wired networks can also take on any of these characteristics (and are not excluded from potential cognitive network applications) wireless networks are a natural target because of their inter-node interactions and the size of their system state space. Previous research into cognitive radio and cross-layer design have addressed some of these issues but have shortcomings from the network perspective. Cognitive networks represent a new scope and approach to dealing with this complexity.

This chapter provides the reader with a primer on the cognitive network concept, as envisioned by the authors. It begins by explaining the need for cognitive networks, how they are defined, and possible applications for the technology. Then the chapter examines how cognitive networks are related

[1] Portions reprinted, with permission, from "Cognitive Networks: Adaptation and Learning to Achieve End-to-end Performance Objectives," *IEEE Communications Magazine*, vol. 44, pp. 51–57, December 2006. ©2006 IEEE.

H. Arslan (Ed.), Cognitive Radio, Software Defined Radio, and Adaptive Wireless Systems, 17–41.

to, but distinct from, previous work in cognitive radios and cross-layer design. A practical discussion of the implementation of a cognitive network and important areas of future work close the chapter.

2.1.1 Definition

Cognitive networks were first described by us in [1] as

> ... a network with a cognitive process that can perceive current network conditions, and then plan, decide and act on those conditions. The network can learn from these adaptations and use them to make future decisions, all while taking into account end-to-end goals.

The cognitive aspect of this definition is similar to that used to describe cognitive radio and broadly encompasses many simple models of cognition and learning. More critical to the definition are the network and end-to-end aspects. Without the network and end-to-end scope, the system is perhaps a cognitive radio or layer, but not a cognitive network. Here, end-to-end denotes all the network elements involved in the transmission of a data flow. For a unicast transmission, this might include such elements as subnets, routers, switches, virtual connections, encryption schemes, mediums, interfaces, and waveforms. The end-to-end goals are what give a cognitive network its network-wide scope, separating it from other adaptation approaches, which have only a local, single element scope.

2.1.2 Motivation and Requirements

The overall goal of any technology is that it meet some need in the best way possible for the least cost. With the first half of this goal in mind, a cognitive network should provide, over an extended period of time, better end-to-end performance than a non-cognitive network. Cognition can be used to improve such end-to-end objectives as resource management, Quality of Service (QoS), security, access control, or throughput. Cognitive networks are only limited in application by the adaptability of the underlying network elements and the flexibility of the cognitive process.

In examining the second half of the goal, the cost must justify the performance. Cognitive network costs are measured in terms of communications and processing overhead, architecture roll-out and maintenance expenses, and operational complexity. These costs must be outweighed by the performance improvement the cognitive network provides. For certain environments, such as static wired networks with predictable behavior, it may not make sense to convert to cognitive operation. Other environments, such as heterogeneous wireless networks, may be ideal candidates for cognition.

Cognitive networks should use observations (or proxy observations) of network performance as input to a decision making process and then provide output in the form of a set of actions that can be implemented in the modifiable elements of the networks. Ideally, a cognitive network should be forward-looking, rather than reactive, and attempt to adjust to problems before they occur. Additionally, the architecture of a cognitive network should be extensible and flexible, supporting future improvements, network elements, and goals.

Cognitive networks require a Software Adaptable Network (SAN) to implement the actual network functionality and allow the cognitive process to adapt the network. Similarly to cognitive radio, which depends on a Software Define Radio (SDR) to modify aspects of radio operation (e.g. time, frequency, bandwidth, code, spatiality, waveform), a SAN depends on a network that has one or more modifiable elements. Practically, this means that a network must be able to modify one or several layers of the network stack in its member nodes. A simple example of a SAN could be a wireless network with directional antennas (antennas with the ability to direct their maximum receive or transmit gain to various points of rotation). A more complex example would incorporate more modifiable aspects at various layers of the protocol stack, such as Medium Access Control (MAC) algorithms or routing control.

2.1.3 A Simple Example

As an example of the need for end-to-end rather than just link adaptations, consider an ad-hoc data session between a source node, S_1, and a destination node, D_1, as shown in Figure 2.1. The source node must route traffic through intermediate nodes R_1 and R_2 acting as regenerative relays. Node S_1 performs a link adaptation by choosing the relay node based on the set of minimum hop routes to D_1 and the probability of link outage. For this simple network, nodes R_1 and R_2 are both in the set of minimum hop relays on routes to D_1. Therefore, node S_1 selects the link on which to transmit by observing the outage probabilities on the links to R_1 and R_2 and selecting the link with the lower outage probability. From the standpoint of the link layer in node S_1, this guarantees that the transmitted packets have the highest probability of

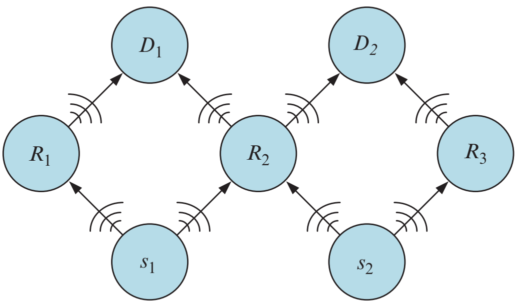

Fig. 2.1. Simple relay network for illustrating the need for cognition with an end-to-end scope.

arriving correctly at the relay node. However, it does not guarantee anything about the end-to-end performance, i.e. the total outage probability from S_1 to D_1.

In contrast to the link adaptation, the cognitive network uses observations from all nodes to compute the total path outage probabilities from S_1 to D_1 through R_1 and R_2. This shows the benefit of a more global view as well as another advantage to the cognitive network, the learning capability. Suppose that the learning mechanism measures throughput from the source to its destination in order to judge the effectiveness of previous decisions, and suppose that nodes S_1 and S_2 are both routing their traffic through R_2 because this satisfies the minimum outage probability objective.

Now suppose that R_2 becomes congested because of a large volume of traffic coming from S_2. This becomes apparent to the cognitive process in the throughput reported by S_1 and S_2, though the cognitive process is not explicitly aware of the congestion. Nevertheless, it is able to infer from the reduced throughput and its past experiences that there may be a problem. The cognitive process is then able to respond to the congestion, perhaps by routing traffic through R_1 and/or R_3. This example illustrates the potential of cognitive networks in optimizing end-to-end performance as well as reacting to unforeseen circumstances. The cognitive network goes beyond the purely algorithmic approach of the underlying routing protocol and finds efficient operating points even when unexpected events occur.

2.2 Foundations and Related Work

Having defined a cognitive network, it is helpful to review some existing research areas that are related to the cognitive network concept. We take a look at two areas in particular, cognitive radio and cross-layer design.

2.2.1 Cognitive Radio

Shared Attributes with Cognitive Networks

The 50% correlation in nomenclature would itself imply some degree of commonality, and it can certainly be argued that research in cognitive radio has sparked the formulation of the cognitive network concept. What cognitive radios and cognitive networks do share is the cognitive process that is the heart of the performance optimizations. An essential part of the cognitive process is the capability to learn from past decisions and use this learning to influence future behavior. Both are goal-driven and rely on observations paired with knowledge of node capabilities to reach decisions. Knowledge in cognitive radio is contained within a modeling language such as Radio Knowledge

Representation Language (RKRL) [2]. A network-level equivalent must exist for the cognitive network to be goal oriented and achieve context awareness, two attributes that it shares with a cognitive radio.

A cognitive radio requires tunable parameters which define the optimization space of the cognitive process. These tunable parameters are ideally provided by an SDR. The concept of the SAN is the cognitive network analog of SDR. Therefore, both technologies employ a software tunable platform that is controlled by the cognitive process.

Differences from Cognitive Networks

Cognitive networks are clearly delineated from cognitive radios by the scope of the controlling goals. Goals in a cognitive network are based on end-to-end network performance, whereas cognitive radio goals are localized only to the radio's user. These end-to-end goals are derived at run-time from operators, users, applications, and resource requirements in addition to any design-time goals. This difference in goal scope from local to end-to-end enables the cognitive network to operate more easily across all layers of the protocol stack. Current research in cognitive radio emphasizes interactions with the physical layer, which limits the direct impact of changes made by the cognitive process to the radio itself and other radios to which it is directly linked or with which it may interfere. Agreement with other radios on parameters that must match for successful link communication is reached through a process of negotiation. Since changes in protocol layers above the physical layer tend to impact more nodes in the network, the cognitive radio negotiation process would have to be expanded to include all nodes impacted by the change. However, because the negotiation process is unable to assign precedence to radios' desires without goals of a broader scope, achieving agreement among multiple nodes may be a slow process. For the same reason, the compromise can be expected to result in sub-optimal network performance. In contrast, whether the network components are acting in a cooperative or selfish manner, all cognitive network actions are referenced back to the end-to-end network goals.

Another significant difference between cognitive radios and cognitive networks is the degree of heterogeneity that is supported. Cognitive networks are applicable to both wired and wireless networks whereas cognitive radios are only used in wireless networks. Since the cognitive network may span wired and wireless mediums, it is useful for optimizing performance for these heterogeneous types of networks, which are generally difficult to integrate.

The fact that a cognitive network is composed of multiple nodes also adds a degree of freedom in how the cognitive processing is performed compared to cognitive radio. A cognitive network has the option to implement a fully distributed, partially distributed, or centralized cognitive process.

2.2.2 Cross-layer Design

Shared Attributes with Cross-layer Design

Designs that violate the traditional layered approach by direct communication between non-adjacent layers or sharing of internal information between layers are called cross-layer designs [3]. Cognitive networks indirectly share information that is not available externally in the strictly layered architecture. Therefore, cognitive networks do implement cross-layer designs.

The common theme between these two concepts is that in both, observations are made available for adaptations at layers other than the layer providing the observation. In a cognitive network, protocol layers provide observations of current conditions to the cognitive process. The cognitive process then determines what is optimal for the network and changes the configurations of network elements' protocol stacks.

Differences from Cross-layer Design

Despite similarities, cognitive networks reach far beyond the scope of cross-layer designs. The cognitive network can support trade-offs between multiple goals and in order to do so performs Multiple Objective Optimization (MOO), whereas cross-layer designs typically perform single objective optimizations. Cross-layer designs perform independent optimizations that do not account for the network-wide performance goals. Trying to achieve each goal independently is likely to be sub-optimal, and as the number of cross-layer designs within a node grows, conflicts between the independent adaptations may lead to adaptation loops [4]. This pitfall is avoided in a cognitive network by jointly considering all goals in the optimization process.

The ability to learn is another significant difference. The cognitive network learns from prior decisions and applies the learning to future decisions. Cross-layer designs are memoryless adaptations that will respond the same way when presented with the same set of inputs, regardless of how poorly the adaptation may have performed in the past. The ability to learn from past behavior is particularly important in light of the fact that our understanding of the interaction between layers is limited.

Finally, like cognitive radio, the scope of the goals and observations sets cognitive networks apart from cross-layer design. The observations used by the cognitive process span multiple nodes and the optimization is performed with the goals of all nodes in mind, whereas cross-layer design is node-centric. This global information allows the cognitive process to adapt in ways that simply are not possible when nodes have limited visibility into the state of other nodes in the network, as is the case with cross-layer design.

2.2.3 Recent Work

The concept of a cognitive networks is an emerging research field. The idea of adding cognition to a network has in the past been reserved for individual aspects of the network, such as "smart" antennas or "smart" packets. All this changed with the introduction of the cognitive radio by Mitola in [2]. His concept of putting intelligence into radio operation caught the imagination and attention of the research community. Eventually the concept worked its way from radios into the larger network.

Recent research can be divided into two categories: cognitive radio networks and cognitive networks. In the first category, we begin with work from Mitola and his original thesis on cognitive radio. Here, he mentions how cognitive radios could interact within the system-level scope of a cognitive network [2]. Neel continues this line of thinking in [5], where he investigates modeling networks of cognitive radios as a large, multiplayer game to determine convergent conditions. This kind of thinking is also observed in Haykin's paper on cognitive radio [6], where he examines multiuser networks of cognitive radios as a game.

The focus of cognitive radio networks, as with cognitive radios, is primarily on MAC and physical (PHY) layer issues, but now operating with some end-to-end objective. In a cognitive radio network, the individual radios still make most of the cognitive decisions, although they may act in a cooperative manner. Currently suggested applications for cognitive radio networks include cooperative spectrum sensing [7,8] and emergency radio networks [9]. From a more general perspective, Raychaudhuri et al. [10] present an architecture for cognitive radio networks.

Perhaps the first mention of a cognitive network rather than a cognitive radio network comes from Clark et al. [11]. Clark proposes a network that can

> assemble itself given high level instructions, reassemble itself as requirements change, automatically discover when something goes wrong, and automatically fix a detected problem or explain why it cannot do so.

According to Clark, this would be accomplished with the use of a Knowledge Plane (KP) that transcends layers and domains to make cognitive decisions about the network. The KP will add intelligence and weight to the edges of the network, and context sensitivity to its core. Saracco also observed these trends in his investigation into the future of information technology [12], postulating that the change from network intelligence controlling resources to having context sensitivity will help "flatten" the network by moving network intelligence into the core and control further out to the edges of the network.

Cognitive networks differ from cognitive radio networks in that the action space of the former extends beyond the MAC and PHY layers and the network may consist of more than just wireless devices. Furthermore, cognitive networks may be less autonomous than a cognitive radio network, with the network elements cooperating to achieve goals, using a centralized cognitive process or a parallelized process that runs across several of the network elements. However, despite these differences, the definition of cognitive networks given in Section 2.1.1 encompasses both cognitive radio networks and cognitive entworks.

More recently, Mähönen discusses cognitive networks in the context of future Internet Protocol (IP) networks and cognitive trends in a series of papers. In his earliest paper, he discusses cognitive networks with respect to future mobile IP networks, arguing that the context sensitivity of these networks could have as interesting an application as cognitive radios [13]. He then examines cognitive networks as part of a larger paper on cognitive trends [14]. He discusses how cognitive radios may be just a logical subset of cognitive networks. He also brings up the idea of a Network Knowledge Representation Language (NKRL) to express and communicate high-level goals and policies.

Several research groups have proposed cognitive network-like architectures. These architectures can be categorized into two objectives: the first centers on using cognition to aid in the operation and maintenance of the network, while the second centers on cognition to solve "hard" problems, problems that do not have a feasible solution other than the use of cognition.

Falling into the first category, the End-to-End Reconfigurability Project II (E^2R II) [15] is designing an architecture that will allow the seamless reconfiguration of a network in order to allow for universal end-to-end connectivity. Although E^2R II is an ambitious project with many facets, the overarching goal is one of maintaining connectivity to the user. This is similar to the goal of the m@ANGEL platform [16], which attempts to provide an cognitive network-like architecture for mobility management in a heterogeneous network. Both of these architectures are focused on the operation and maintenance of 4G cellular and wireless networks.

In contrast, the Center for Telecommunications Value-Chain Research (CTVR) at Trinity College [17] has presented a proposal for a cognitive network platform that consists of reconfigurable wireless nodes. Although focused on wireless operation, these nodes are able to solve a variety of problems by modifying or changing the network stack based on observed network behaviors. The possible objectives of these networks can extend beyond mobility management and connectivity. Similar to the CTVR work but less dependent on the wireless focus, Mähönen proposes a general architecture utilizing a collaborative Cognitive Resource Manager (CRM) that provides cognitive behavior from a toolbox of machine learning tools such as neural networks, clustering, coloring, genetic algorithms, and simulated annealing. The work in this chapter describes also falls under this objective, attempting to provide a general cognitive architecture capable of solving a variety of hard problems, rather than being tied to network operation issues.

2.3 Implementation

In order to synthesize the preceding concepts and components into an actual cognitive network, we investigate how a cognitive network should be implemented. We construct a framework for the cognitive process and identify the critical features of this architecture.

A common model of cognition is the three-level theory [18]. The model is often summarized as consisting of behavioral, functional, and physical layers. The behavioral level determines what observable actions the system produces, the functional layer determines how the system processes the information provided to it, and the physical layer comprises the neuro-physiology of the system.

From this concept, we draw a three-layer framework, with each layer roughly corresponding to the layers in the model described above. At the top layer are the goals of the system and elements in the network that define the behavior of the system. These goals feed into the cognitive process, which computes the actions the system takes. The SAN is the physical control of the system, providing the action space for the cognitive process. This framework is illustrated in Figure 2.2.

In our framework, we consider a cognitive process which consists of one or more cognitive elements, operating in some degree between selfish autonomy and full cooperation. If there is a single cognitive element, it may still be physically distributed over one or more nodes in the network. If there are multiple elements, they may be distributed over a subset of the nodes in the network, on every node in the network, or several cognitive elements may reside on a single node. In this respect, the cognitive elements operate in a manner similar to software agents.

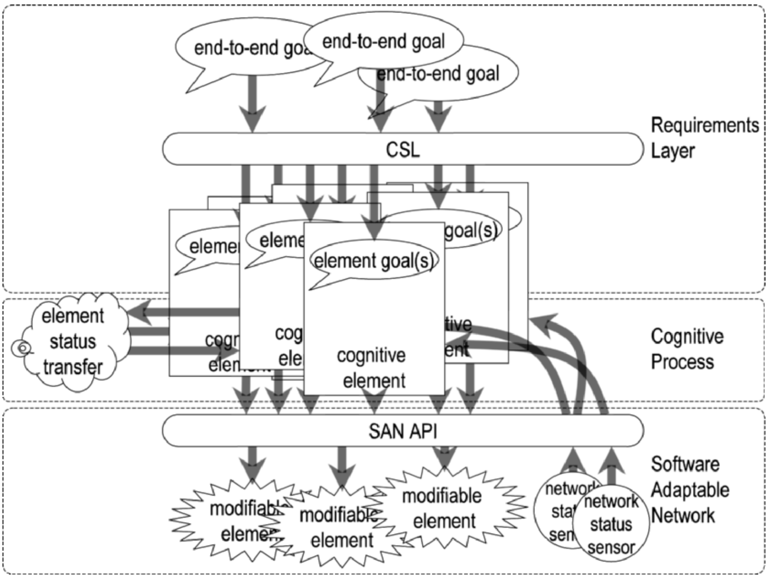

Fig. 2.2. The cognitive network framework.

2.3.1 User/Application/Resource Requirements

The top-level component of the cognitive network framework includes the end-to-end goals, Cognitive Specification Language (CSL) and cognitive element goals. Without end-to-end goals guiding network behavior, undesired consequences may arise. For instance, optimizing a network element without an end-to-end scope can cause a negative effect on the performance elsewhere in the network or node. This is a problem with many cross-layer designs and is explored in some depth in [4], which illustrates unintended end-to-end interactions in a MAC/PHY cross-layer design.

Like most engineering problems, there is likely to be a trade-off for every goal optimized on. When a cognitive network has multiple objectives it will not be able to optimize all metrics indefinitely, eventually reaching a point at which one metric cannot be improved without adversely affecting another. In order to determine this Pareto optimal front (the set of actions from which no goal can be improved without worsening another), each cognitive element must have an understanding of all end-to-end goals and their constraints.

To connect the goals of the top-level users of the network to the cognitive process, an interface layer must be developed. In a cognitive network, this role is performed by the CSL, providing behavioral guidance to the elements by translating the end-to-end goals to local element goals. Less like the RKRL proposed by Mitola for cognitive radio and more like a QoS specification language [19], the CSL maps end-to-end requirements to underlying mechanisms. Unlike a QoS specification language, the mechanisms are adaptive to the network capabilities, as opposed to fixed. Furthermore, a CSL must be able to adapt to new network elements, applications, and goals, some of which may not even be imagined yet. Other requirements may include support for distributed or centralized operation, including the sharing of data between multiple cognitive elements.

The scope of the cognitive network is broader than that of the individual network elements; it operates within the scope of a data flow, which may include many network elements. For a distributed cognitive process, the cognitive elements associated with each flow or network element may act selfishly and independently (in the context of the entire network) to achieve local goals, or act in an altruistic manner to achieve network-wide goals. The job of converting the end-to-end goals to these local element goals is often a difficult problem.

2.3.2 Cognitive Process

There does not seem to be a common, accepted definition of what cognition means when applied to communication technologies. The term cognitive, as used by this chapter, follows closely in the footsteps of the definition used by

Mitola in [2] and the even broader definition of the FCC. The former incorporates a spectrum of cognitive behaviors, from goal-based decisions to proactive adaptation. Here, we associate cognition with *machine learning*, which is broadly defined in [20] as any algorithm that "improves its performance through experience gained over a period of time without complete information about the environment in which it operates." Underneath this definition, many different kinds of artificial intelligence, decision making, and adaptive algorithms can be placed, giving cognitive networks a wide scope of possible mechanisms to use for learning.

Learning serves to complement the objective optimization part of the cognitive process by retaining the effectiveness of past decisions under a given set of conditions. Determining the effectiveness of past decisions requires a feedback loop to measure the success of the chosen solution in meeting the objectives defined. This is retained in memory so that when similar circumstances are encountered in the future, the cognitive process will have some idea of where to start or what to avoid.

The effect of a cognitive process's decisions on the network performance depends on the amount of network state information available to it. In order for a cognitive network to make a decision based on end-to-end goals, the cognitive elements must have some knowledge of the network's current state and other cognitive element states. If a cognitive network has knowledge of the entire network's state, decisions at the cognitive element level should be at least as good, if not better (in terms of the cognitive element goals) than those made in ignorance. For a large, complex system such as a computer network, it is unlikely that the cognitive network would know the total system state. There is often a high cost to communicate this information beyond those network elements requiring it, meaning a cognitive network will have to work with less than a full picture of the network status.

Filtering and abstraction may be used to further reduce the amount of information that must be exchanged and to avoid unnecessary triggering of the cognitive process. Filtering means that observations made by the node may be held back from the cognitive process if they are deemed irrelevant. Thus, the nodes themselves make some determination of what is important to the cognitive process. Filtering rules may be identified at design time with additional rules specified in real-time as the cognitive process determines its sensitivity to various types of observations and disseminates filtering rules accordingly. The goal of abstraction is to reduce the number of bits required to represent an observation. Observations or collections of observations made by a node are reported to the cognitive process at a higher level of abstraction than what is available within the node. Abstractions may also be specified at design time with real-time adaptations by the cognitive process. The reductionism resulting from filtering and abstraction carries risk because it may mask information that the cognitive process needs to operate correctly. Therefore, care should be taken in defining the abstractions or filtering.

2.3.3 Software Adaptable Network

The SAN consists of the Application Programming Interface (API), modifiable network elements, and network status sensors. The SAN is really a separate research area, just as the design of the SDR is separate from the development of the cognitive radio, but at a minimum the cognitive process needs to be aware of the API and the interface it presents to the modifiable elements. Just like the other aspects of the framework, the API should be flexible and extensible. Continuing the analogy with SDKs, an existing system that is analogous to the API is the Software Communications Architecture (SCA) used in the Joint Tactical Radio System (JTRS).

Another responsibility of the SAN is to notify the cognitive process of the status of the network (to what level and detail is a function of the filtering and abstraction being applied). The status of the network is the source of the feedback used by the cognitive process, and is composed of status sensor observations and communication with other cognitive elements. Possible observations may be local, such as bit error rate, battery life or data rate, non-local, such as end-to-end delay and clique size, or compilations of different local observations.

The modifiable elements can include any object or element used in a network, although it is unlikely that all elements in a SAN would be modifiable. Each modifiable element should have public and private interfaces to the API, allowing it to be manipulated by both the SAN and the cognitive process. Modifiable elements are assumed to have a set of states that they can operate in, and a "solution" for a cognitive process consists of a set of these states that, when taken together, meet the end-to-end requirements of the system. At any given instant the set of all possible combinations of states S can be partitioned into two subsets. The first, S', contains all possible combinations of sets that meet the end-to-end requirements and the second, $\bar{S}'$, consists of all combinations that do not meet these requirements. Of those in S', some may meet the requirements better than others, making them preferred solutions.

A cognitive network attempts to reach a set of states S'. This means that, should the network be in a state in $\bar{S}'$, or some sub-optimal state in S', the cognitive process attempts to move the system state to an optimal solution. With cognitive control over every element, the cognitive process can potentially set the system to any state; an ideal cognitive process could set the state to the optimal solution. If the system has only a few points of cognitive control, or chooses not to exercise all its control, then the cognitive process has to use the functionality and interactions of the non-cognitive aspects of the network to set the system state. Like the hole at the bottom of a funnel, certain system states will be basins of attraction, pulling the system towards them from a variety of starting states. If a system has several attractors and some are more optimal than others, then a few points of cognitive control may be enough to draw the system out of one attractor and into another. This is analogous to a watershed, in which moving the source of water a few miles may be enough to change what river the water will finally flow into.

2.4 A Cognitive Network for Multicast Lifetime

To illustrate the effect of these critical design decisions on a network, we present a cognitive network approach to maximizing a multicast flow's lifetime. By investigating even a simple cognitive network for a real-world problem, some of the subtleties of the design process can be explored. In this manner, the following cognitive network problem should be viewed as an illustrative case study.

Many factors may affect the expected lifetime of a network connection in a wireless network. For instance, traffic congestion can cause timeouts in upper layer protocols, interference can cause loss of connectivity at the PHY layer, and mobility can cause unexpected disconnections in traffic routing. However, for mobile and portable devices, one of the chief factors in determining the lifetime of a connection is the energy remaining in the batteries of the mobile nodes.

This example focuses on a cognitive network with control over the transmission power, antenna directionality, and routing tables of the network nodes. This is not the first investigation into lifetime routing in wireless networks; a large body of work on power-efficient routing exists in the literature. Gupta's survey [21] provides an excellent comparison of several power-efficient multicast algorithms for omnidirectional antennas. Weiselthier et al. [22] have examined this problem using directional antennas. A complete review of the related literature and an investigation using Mixed Integer Linear Program (MILPs) for determining the optimal lifetimes can be found in [23]. Although primarily designed to illustrate the cognitive network concept, this work is the first to provide a distributed, cognitive network approach to multicast lifetime routing that incorporates energy efficiency considerations, directional antennas, and a Signal to Interference and Noise Ratio (SINR) sufficiency requirement.

2.4.1 Problem Description

A wireless network is made up of a collection of network elements with varying energy capacity. Some elements may be battery powered, with limited capacity, while others may be less mobile, with large, high capacity batteries. The lifetime of a data path, however, is limited by the radio utilizing the largest fraction of its battery capacity. By minimizing the utilization of this bottleneck radio, the lifetime of the path can be maximized. Furthermore, we consider a network where radios are equipped with directional antennas, which are useful to reduce interference, improve spatial multiplexing, and increase range.

We model a network consisting of a set of radios $N = \{1, 2, \ldots, n\}$, in which the objective is to create a maximum lifetime multicast tree between source S and destination set D. As described earlier, the cognitive network controls three modifiable network parameters: the radio transmission power (contained in the elements of vector $\mathbf{pt}$), the antenna directionality (angles are contained in the elements of vector ϕ), and element routing tables (contained in each node of the multicast tree T). The states of the modifiable elements are part of the action set A, of which the action vector $\mathbf{a}$ contains the current state of each modifiable element.

In the model used here, the lifetime of a radio is inversely proportional to the utilization of the radio's battery,

$$\mu_i = \frac{pt_i}{ca_i} \tag{2.1}$$

where pt_i is radio i's transmission power and ca_i is the remaining energy capacity of its battery. The lifetime of a data path is limited by the radio utilizing the largest fraction of its battery capacity, so over the entire multicast tree T, the lifetime will be inversely proportional to the utilization of the max-utilization radio

$$\mu_T = \max_{j \in T} \{\mu_j\} \tag{2.2}$$

The network consists of radios with fully directional antennas in receive mode[2] (each element transmits omnidirectionally and receives directionally) with a fixed beamwidth θ that can take on a boresight angle $\phi \in [0, 2\pi)$. Figure 2.3 illustrates the operation of an ad-hoc network with directional antennas in receive mode.

When radio i transmits, the signal experiences gain factor gb within the main beam of the antenna [25]

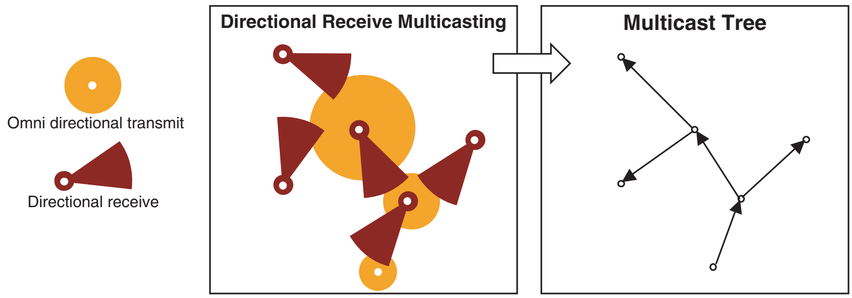

Fig. 2.3. The directional receive multicast operation. The shaded areas extending from the radios represent regions of increased gain.

[2] An argument for using directional reception rather than transmission can be found in [24].

$$gb = \frac{2\pi}{\theta} \tag{2.3}$$

Some energy leaks outside the main beam in sidelobes. The fraction that ends up in the beam is $pct \in (0, 1)$ and the fraction outside the beam is $(1 - pct)$. We also consider a path loss attenuation factor, proportional to

$$gp_{ij} = \frac{1}{d(i,j)^\alpha} \tag{2.4}$$

where $d(i, j)$ is the euclidean distance between source i and destination j and α is the path loss exponent. Combining these gains and attenuations, the overall gain from a transmission by radio i received at radio j is

$$g_{ij}(\phi_j) = \begin{cases} gb \cdot gp_{ij} \cdot pct & \phi_j \in a(i,j) \pm \frac{\theta}{2} \\ gp_{ij} \cdot (1 - pct) & otherwise \end{cases} \tag{2.5}$$

where $a(i, j)$ is the angular function between radios i and j.

A radio j can correctly receive information from radio i if the power received from the desired transmitter is greater than all other power and noise received by some SINR factor. We define the vector **pr** to be the power received at every radio in the tree from their parent radio,

$$pr_j(pt_i, \phi_j) = pt_i \cdot g_{ij}(\phi_j) \tag{2.6}$$

There is an entry in this vector for every radio in the tree, with the exception of the source radio ($|pr| = |T| - 1$). We then define vector **no** to be the minimum required power to overcome the interference and noise received at every element,

$$no_j(\mathbf{pt}, \phi_j, T) = \left(\sum_{k \neq i} pr_k(pt_i, \phi_j) + \sigma_j \right) \gamma_j \tag{2.7}$$

where σ_j is the thermal noise and γ_j is the SINR requirement for a particular radio. The vectors **pr** and **no** combine to give the network constraint,

$$\mathbf{pr} - \mathbf{no} \geq \mathbf{0} \tag{2.8}$$

2.4.2 Cognitive Network Design

The cognitive network framework encompasses a wide spectrum of possible implementations and solutions. This approach allows the framework to be a method for approaching problems in complex networks, rather than a specific solution. The framework sits on top of existing network layers, processes, and protocols, adjusting elements of the SAN to achieve an end-to-end goal. In this section, we show how a cognitive network that solves the multicast lifetime

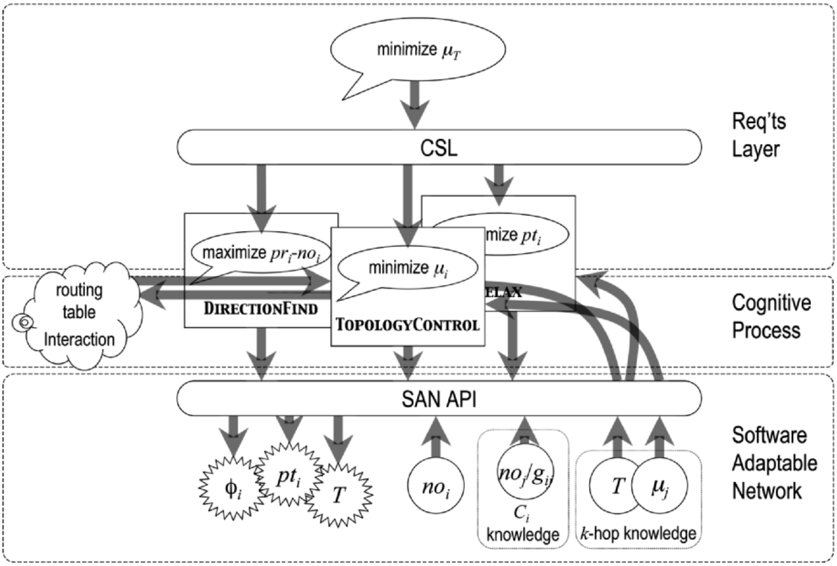

Fig. 2.4. The components of the multicast lifetime cognitive network as they fit into the framework.

problem fits into the framework. We examine each layer, showing how the requirements layer provides goals to the cognitive elements, how the cognitive process performs the feedback loop, and identify the functionality of the SAN. The ideas in this section are illustrated in Figure 2.4.

The cognitive process consists of three cognitive elements that distribute the operation of the cognitive process both functionally and spatially: PowerControl, DirectionControl, and RoutingControl. PowerControl adjusts the PHY transmission power (pt_i), DirectionControl adjusts the MAC spatial operation (ϕ_i), and RoutingControl adjusts the network layer's routing functionality (T). The SAN network status sensors provide each cognitive element with the knowledge of each radio's battery utilization in its *k-hop neighborhood*. The k-hop neighborhood of a radio is defined to be every radio reachable in the routing tree via at most k hops, following the routing tree both up and down branches.

Requirements Layer

The cognitive network investigated here is associated with a single objective optimization as its end-to-end goal. As such, the performance of an action vector is only dependent on the life-span of the multicast flow. The end-to-end objective is defined in Equation 2.9 as a cost function, where the lifetime of a flow is increased as $C(\mathbf{a})$ is minimized.

$$C(\mathbf{pt}, \phi, T) = \begin{cases} \mu_T & \mathbf{pr} - \mathbf{no} \geq \mathbf{0} \\ \infty & \text{otherwise} \end{cases} \tag{2.9}$$

Each of the modifiable elements affects the calculation of this cost: transmission power affects the lifetime directly; antenna orientation and routing table influence the lifetime indirectly by affecting the required transmission power.

The requirements layer transforms the end-to-end objective into a goal for each cognitive element through the CSL. Although these objectives are localized (each element only adapts a single modifiable element) the state of all modifiable elements affects the cognitive element's performance.

PowerControl's objective is to minimize the transmission power on every radio subject to the system constraint. This means that a radio will attempt to transmit at the minimum power that connects it to all of its children through the local control of pt_i. The objective can be represented by the utility function

$$u_i^{\text{PC}}(\mathbf{pt}) = -\left(\max_{j \in \mathcal{C}_i}\left\{\frac{no_k}{g_{ij}}\right\} - pt_i\right)^2 \tag{2.10}$$

which is maximized when the transmitting radio has exactly the power needed to reach the child radio with the greatest noise and least gain factor. $\mathcal{C}_i$ is the set of child radios that receive from radio i in the multicast tree.

The objective of DirectionControl is to maximize the receiving radio's SINR by controlling the directional angle of the antenna beam ϕ_i locally at every antenna. One form that the utility can take is

$$u_i^{\text{DC}}(\mathbf{pt}, \phi_i) = pr_i - no_i \tag{2.11}$$

By rotating the directional antenna, the radio can increase the gain from the parent radio, while attenuating interfering signals.

The objective of RoutingControl is to minimize each radio's battery utilization by manipulating the routing tree (T) used by the network. The utility can be expressed as

$$u_i^{\text{RC}}(pt_i) = \frac{1}{\mu_i} \tag{2.12}$$

By manipulating the children radios that it has to transmit to, a radio can reduce its transmission power and battery utilization.

Cognitive Process

The cognitive process consists of the three cognitive elements described above, each operating on every radio in the network. In this section, we discuss the strategies utilized by these elements to achieve the above objective goals and identify the critical design decisions used by each cognitive element.

Algorithm 1 RELAX($\mathbf{pt}, \phi, T$) $\rightarrow \hat{\mathbf{pt}}$

1: **while** not at $\hat{\mathbf{pt}}$ **do**
2: **for** $i = 1 \ldots n$ **do**
3: $pt_i = \max_{j \in \mathcal{C}_i}\{\frac{no_j}{g_{ij}}\}$
4: **end for**
5: **end while**

PowerControl The PowerControl cognitive element uses a strategy called RELAX. RELAX moves the transmission power of the elements of a tree to a minimum but sufficient power state (referred to as $\hat{\mathbf{pt}}$) for a given tree structure. Nodes do this by increasing or decreasing transmission power until it just meets the SINR sufficiency requirements of all their children. This means that parent node i will iteratively increase or decrease its transmission power according to the amount of interference and noise observed by the child j with the maximum amount of noise and interference. Algorithm 1 describes this process more formally.

RELAX is similar to the asynchronous iterative power control algorithm presented by Yates [26]. That paper proved, for a cellular network consisting of multiple handsets communicating with a base-station, if Equation 2.8 is *feasible* (meaning that there exists a solution), RELAX will find the optimal $\hat{\mathbf{pt}}$. Yates' work is for the reverse of the problem we consider – a cellular network is comprised of many nodes transmitting to a single base station. In contrast, our work considers a multicast wireless network with a set of parent nodes transmitting to many children. However, it is easy to show Yate's results still hold.

DirectionControl The second cognitive element's behavior is DirectionControl. DirectionControl moves the directional antenna to the orientation that maximizes the received SINR from a node's parent node. There are several direction-finding algorithms in the literature [27] and DirectionControl can implement one of these. If a node is a part of the multicast routing tree, it directs its antenna such that the power received from the parent is maximized with respect to the amount of interference and noise. If a node is not part of the multicast tree, it directs the antenna towards any source from which it can receive with the greatest SINR. For clarity, we will delineate these two tree structures: the first, called the *functional tree*, consists of just elements in the multicast routing tree and the second, called the *structural tree*, includes every element in the system that can receive a signal that meets the SINR requirement.

RoutingControl RoutingControl attempts to minimize the utilization of the radio batteries by approximating a Steiner tree for the utilization metric. RoutingControl uses the CHILDSWITCH strategy described in Algorithm 2. CHILDSWITCH begins by determining if it is operating on the *max-utilization* radio (the radio with maximum battery utilization) of its k-hop neighborhood,

Algorithm 2 CHILDSWITCH($\mathbf{pt}, \phi, T$) $\rightarrow$ ($\hat{\mathbf{pt}}, \hat{\phi}, T'$)

if $\mu_i = \max_{n \in N_i}\{\mu_n\}$ **then** $\{$*is on a max-util. node*$\}$
  $\mu_{minmax} = \mu_i$ $\{$*record the config. as the min-max*$\}$
  $minmax = i$
  $j = \text{argmax}_{c \in C_i}\{no_c\}$ $\{$*record the max-power child*$\}$
  for $n \in N_i \quad n \neq j;\ n \notin B_j$ **do** $\{$*every valid neighbor*$\}$
    $C_n = C_n \cup \{j\}$ $\{$*add max-power child*$\}$
    $\hat{\phi} =$ DirectionControl(ϕ)
    $\hat{\mathbf{pt}} =$ RELAX($\mathbf{pt}$)
    $\mu_{max} = \text{argmax}_{n \in N_i}\{\mu_n\}$ $\{$*record max-util.*$\}$
    if $\mu_{max} < \mu_{minmax}$ **then** $\{$*max-util. is least*$\}$
      $\mu_{minmax} = \mu_{max}$ $\{$*record it as the min-max*$\}$
      $minmax = max$
    end if
    $C_n = C_n \backslash \{j\}$ $\{$*remove max-power child*$\}$
  end for
  $C_{minmax} = C_{minmax} \cup \{j\}$ $\{$*change to min-max config.*$\}$
end if

by comparing its battery utilization against every k-hop neighbor's battery utilization. If it is, the radio becomes the *control-radio* and takes control over the routing tables of every element in the k-hop neighborhood. It then identifies which of the children radios in the functional tree requires the greatest amount of power to reach (the *max-power child*). The control-radio then attempts to detach the max-power child from itself and re-attach it as the child of another radio (by changing the routing table of a k-hop neighbor so that it becomes the new parent) in the k-hop neighborhood, in order to reduce the k-hop neighborhood's maximum utilization.

Valid choices for a new parent for the max-power child include all radios in the k-hop neighborhood of the structural tree, except for children of the max-power child. By using the structural tree rather than the functional tree, new radios in the network can be brought sensibly into the functional tree. After assignment, CHILDSWITCH waits until RELAX converges and DirectionControl selects the correct beam angle. When RELAX converges, CHILDSWITCH on the control-radio compares the utilization of all radios in the k-hop neighborhood against its initial utilization. The process is then repeated for the remaining valid radios, with the control-radio remembering the best (minimum) max-utilization configuration, and upon completion setting the routing table to this configuration.

This process repeats indefinitely until the max-utilization control-radios are no longer able to move their max-power children to configurations that lower the max-utilization radio of their k-hop neighborhood. In a synchronous network, in which only one RoutingControl control-radio performs CHILDSWITCH at a time, the network will (except in rare cases) converge to a single set of max-utilization radios.

Software Adaptable Network

The SAN provides an interface to the three modifiable network elements and the status of the network. The reported status is the local noise, maximum transmission power required to reach its children, k-hop battery utilization and k-hop routing tree.

The required transmission power, battery utilization of child radios, and routing tree status can be discovered and reported via a variable power handshaking scheme. In a synchronous manner, radios one by one send a HELLO message addressed to all children. The children each responded with an ACK message to the parent. The parent then decreases its transmission power and sends a new HELLO message until it fails to receive an ACK from some child. The parent radio then stops decreasing its power and returns to the previous power level, which is the maximum transmission power required to reach all its children. These HELLO and ACK messages can also transfer information about each radio's battery utilization and the routing tree within the k-hop neighborhood. In contrast to these non-local measurements, the amount of local noise can be calculated through the local SINR measurement.

2.4.3 Results

To determine the effectiveness of this cognitive network, we developed a simulation of the cognitive network. The simulation was written in Matlab, and consisted of nodes placed with a uniform random distribution in a square 2-D map with density 0.1 nodes/unit^2. There is a single source node and a variable number of receivers. The beam width θ is 30°, the path loss exponent α is 2, and 30% of the transmitted power is assumed to leak out through sidelobes $(1-pct)$. Each wireless node was given a battery with a random capacity (ca_i) uniformly distributed between 0 and 300 units of energy. The SINR sufficiency requirement is set to 1, meaning that the received power must be greater than the noise and interference to satisfy Equation 2.8.

The normalized lifetime of a path is calculated as the ratio of the lifetime obtained by the cognitive network to the optimal lifetime for the same set of source/destinations, capacities, and node positions. The optimal lifetime was determined using Wood's MILP [24]. Knowing the optimal solution is useful, since it allows a true "apples-to-apples" comparison between different scenarios, resulting in an accurate gauge of how effective the cognitive network is.

Underlying the cognitive network, one of two different generic multicast routing algorithms was used. The first, GREEDY, uses a greedy algorithm to create the multicast tree. GREEDY forms the multicast tree from the source node, adding minimum utilization nodes until a spanning tree has been formed. Utilization is estimated for every pair of nodes as the ratio of the (non-interference) transmission power required to reach each node to the

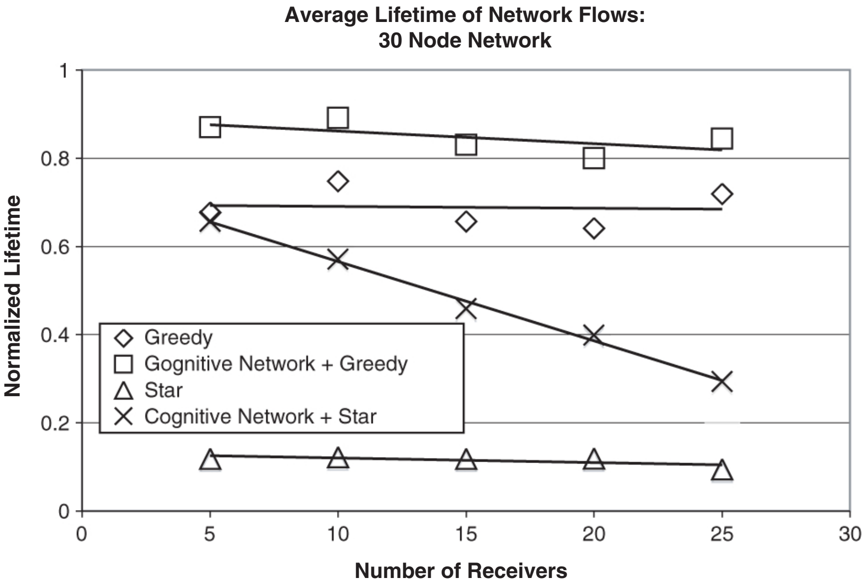

Fig. 2.5. Normalized lifetime of 30 node networks before and after cognitive network improvement. 1-Hop knowledge is used here.

node's battery capacity. Greedy then prunes off branches until it has the minimum tree required to reach every destination. The other multicast algorithm used is Star, which implements a 1-hop broadcast star from the source to every destination.

For a given scenario (consisting of node count, location, and battery capacity) both Greedy or Star were run, individually. The resultant tree topology and parent/child information from each algorithm were handed to Relax and DirectionControl, which determine pt_i and ϕ_i, maximizing the lifetime for this route. The full cognitive process, including RoutingControl was then run on the route determined by Greedy and Star until it converged to a single set of max-utilization nodes. The lifetime of the resultant tree was then calculated. Finally, both the non-cognitive and cognitive lifetimes for a scenario were compared against the optimal lifetime obtained from the MILP, providing a normalized lifetime in $(0, 1)$, where 1 represents an optimal lifetime for that particular scenario.

These performance improvements are illustrated quantitatively in Figure 2.5 and qualitatively in Figure 2.6. Figure 2.5 illustrates the improvement in average lifetime produced by the cognitive process, and Figure 2.6 shows an example multicast tree and corresponding lifetime, both with and without the cognitive process.

These results show that the simplicity of Star leads to sub-optimal performance, with at worst case less than 40% of the average lifetime of Greedy. However, it also confirms that the cognitive network can make a significant

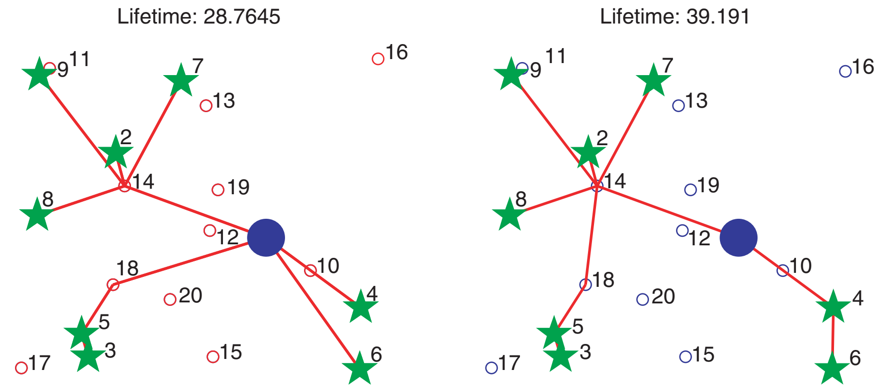

Fig. 2.6. Multicast tree topology and lifetime (source is circle, destinations are stars) for a sample scenario as first chosen by Greedy (left) and then after improvement by cognitive element adaptations (right).

improvement on the average lifetime of the flow by using a 1-hop neighborhood – over 125% improvement in the Star case. The Greedy algorithm alone achieves much longer lifetimes, but the cognitive network is still able to improve it by 5–15%. In both routing algorithms, lifetimes remained steady or decreased as the number of multicast receivers increased. The cognitive network was able to improve the lifetime of the connection for all receiver counts and neighborhood sizes.

2.5 Future Questions and Research Areas

The previous sections make a case for the "what, why, and how" of cognitive networks. We now examine major issues that need to be addressed in order to move from concept to reality.

There is an implicit assumption in this chapter that the cognitive network implements configuration changes synchronously. The details of actually making this happen with high reliability are likely to be complex. The implications of nodes' switching configuration at different times may be worse than if no adaptation had been performed at all. Also, the varying topology of the network means that not all nodes will receive notification of configuration changes at the same time. A possible approach is to require nodes to be synchronized to some common time reference and to issue configuration changes with respect to the time reference. However, this adds complexity to the nodes and still does not guarantee that messages will not be lost, resulting in stranded nodes. It also forces the network to delay its adaptation to the conditions that spawned the configuration change.

Due to the autonomy of each, there is potential conflict between what a cognitive radio and a cognitive network each control if there is not an integrated architecture. One approach is to turn all control over to the cognitive network, but this is probably unwise. The reason is that the cognitive network has to limit its observations as much as possible just to make cognitive processing for a network feasible. This leaves much detailed local information out of the cognitive network picture. This detailed local information may be used by the cognitive radio to further optimize its performance outside the bounds of what is controlled by the cognitive network. To do this, the cognitive radio must know what it is allowed to change and what is in the hands of the cognitive network. A potential solution is to allow the cognitive network to establish regulatory policy for the cognitive radio in real-time, leaving the cognitive radio to perform further optimization under the constraints established by the cognitive network policy.

2.6 Conclusion

This chapter laid the groundwork for the concept of a cognitive network and proposed a definition for the term. Additionally, the cognitive network concept was compared against both cognitive radio and cross-layer design. Finally, a framework for cognitive networks was presented, and critical themes and issues were identified in the design and implementation of a cognitive network. While a significant amount of work remains to be done to make cognitive networks a reality, the rising complexity of networks and the need to manage this complexity makes the concept timely and attractive.

Although computer networks are becoming increasingly ubiquitous, the ability to manage and operate them is not becoming increasingly easier. Cognitive networks, with their promise to self-adapt to meet end-to-end objectives, are an emerging technology that will deal with this increasing complexity. This chapter presented three critical properties that designers need to trade off when architecting a cognitive network. These critical properties will provide design guidelines for future research into and implementation of cognitive networks.

References

1. R. W. Thomas, L. A. DaSilva, and A. B. MacKenzie, “Cognitive networks,” in *Proc. of IEEE DySPAN 2005*, pp. 352–360, November 2005.
2. J. Mitola, *Cognitive Radio: An Integrated Agent Architecture for Software Defined Radio.* PhD thesis, Royal Institute of Technology (KTH), 2000.
3. V. Srivastava and M. Motani, “Cross-layer design: A survey and the road ahead,” *IEEE Communications Magazine*, vol. 43, no. 12, pp. 112–119, 2005.
4. V. Kawadia and P. R. Kumar, “A cautionary perspective on cross-layer design,” *IEEE Wireless Communications*, vol. 12, no. 1, pp. 3–11, 2005.
5. J. O. Neel, J. H. Reed, and R. P. Gilles, “Convergence of cognitive radio networks,” in *Proc. of IEEE WCNC 2004*, vol. 4, pp. 2250–2255, 2004.

6. S. Haykin, “Cognitive radio: Brain-empowered wireless communication,” *IEEE Journal on Selected Areas in Communication*, vol. 23, pp. 201–220, February 2005.
7. G. Ganesan and Y. Li, “Cooperative spectrum sensing in cognitive radio networks,” in *Proc. of IEEE DySPAN 2005*, pp. 137–143, 2005.
8. S. M. Mishra, A. Sahai, and R. W. Brodersen, “Cooperative sensing among cognitive radios,” in *Proc. of IEEE ICC 2006*, vol. 4, pp. 1658–1663, 2006.
9. P. Pawelczak, R. V. Prasad, L. Xia, and I. G. M. M. Niemegeers, “Cognitive radio emergency networks – requirements and design,” in *Proc. of IEEE DySPAN 2005*, pp. 601–606, 2005.
10. D. Raychaudhuri, N. B. Mandayam, J. B. Evans, B. J. Ewy, S. Seshan, and P. Steenkiste, “CogNet: An architectural foundation for experimental cognitive radio networks within the future Internet,” in *Proc. ACM/IEEE MobiArch’06*, pp. 11–16, December 2006.
11. D. D. Clark, C. Partridge, J. C. Ramming, and J. T. Wroclawski, “A knowledge plane for the Internet,” in *Proc. of SIGCOMM ’03*, pp. 3–10, ACM Press, 2003.
12. R. Saracco, “Forecasting the future of information technology: How to make research investment more cost-effective,” *IEEE Communications Magazine*, vol. 41, pp. 38–45, December 2003.
13. P. Mähönen, J. Riihijärvi, M. Petrova, and Z. Shelby, “Hop-by-hop toward future mobile broadband IP,” *IEEE Communications Magazine*, vol. 42, no. 3, pp. 138–146, 2004.
14. P. Mähönen, “Cognitive trends in making: Future of networks,” in *Proc. of IEEE PIMRC 2004*, vol. 2, pp. 1449–1454, 2004.
15. D. Bourse, M. Muck, O. Simon, N. Alonistioti, K. Moessner, E. Nicollet, D. Bateman, E. Buracchini, G. Chengeleroyen, and P. Demestichas, “End-to-end reconfigurability (E^2R II): Management and control of adaptive communication systems.” Presented at IST Mobile Summit 2006, June 2006.
16. 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*, vol. 44, no. 6, pp. 118–127, 2006.
17. P. Sutton, L. E. Doyle, and K. E. Nolan, “A reconfigurable platform for cognitive networks,” in *Proc. of CROWNCOM 2006*, 2006.
18. P. N. Johnson-Laird, *The Computer and the Mind.* Harvard University Press, 1988.
19. J. Jin and K. Nahrstedt, “QoS specification languages for distributed multimedia applications: A survey and taxonomy,” *IEEE Multimedia*, vol. 11, no. 3, pp. 74–87, 2004.
20. M. A. L. Thathachar and P. S. Sastry, *Networks of Learning Automata.* Kluwer Academic Publishers, 2004.
21. S. K. S. Gupta and B. Wang, “Energy-efficient multicast protocols,” in *Resource Management in Wireless Networking* (M. Cardei, I. Cardei, and D.-Z. Du, eds.), Kluwer Academic Publishers, 2004.
22. J. E. Wieselthier, G. D. Nguyen, and A. Ephremides, “Energy-aware wireless networking with directional antennas: The case of session-based broadcasting and multicasting,” *IEEE Transactions on Mobile Computing*, vol. 1, pp. 176–191, July–September 2002.

23. K. N. Wood, *Directional Communications to Improve Multicast Lifetime in Ad Hoc Networks.* Doctoral dissertation, Virginia Polytechnic Institute and State University, Blacksburg, VA, September 2006.
24. K. Wood and L. A. DaSilva, "Optimization of multicast routing for network lifetime with directional listening," in *Proc. of BROADNETS 2005*, pp. 494–500, October 2005.
25. J. E. Wieselthier, G. D. Nguyen, and A. Ephremides, "Energy-limited wireless networking with directional antennas: The case of session-based multicasting," in *Proc. of IEEE INFOCOM 2002*, vol. 1, pp. 190–199, 2002.
26. R. D. Yates, "A framework for uplink power control in cellular radio systems," *IEEE Journal on Selected Areas in Communications*, vol. 13, pp. 1341–1347, September 1995.
27. H. L. V. Trees, *Optimum Array Processing.* No. 4 in Detection Estimation and Modulation Theory, Wiley-Interscience, 2002.