Requirements of an Open Platform for Cognitive Networks Experiments

Luiz A. DaSilva, Allen B. MacKenzie, Claudio R. C. M. da Silva
Bradley Dept. of Electrical and Computer Engineering
Virginia Tech
Blacksburg, VA, USA
{ldasilva, mackenab, cdsilva}@vt.edu

Ryan W. Thomas
Dept. of Electrical and Computer Engineering
Air Force Institute of Technology
WPAFB, OH, USA
ryan.thomas@afit.edu

Abstract— Experimental research on networks of cognitive radios has been hindered by the lack of open, affordable cognitive radios and associated software that are capable of operating with the full network protocol stack. In this paper, we describe our vision of the building blocks needed to create an open platform for cognitive network experimentation and prototyping. These include mechanisms for distributed spectrum sensing, a MAC protocol tailored to dynamic spectrum access, and interface languages for cognitive networks.

Keywords - cognitive networks; hardware platforms; distributed spectrum sensing; DSA MAC; representation languages

I. MOTIVATION

Cognitive radios enable the opportunistic use of under-utilized spectrum and real-time modulation and waveform adaptation to better fit the current wireless environment. Research on cognitive radios has exploded in the past few years, propelled in part by newly-available spectrum created by the move to digital television and by the promise of technical innovation in a more dynamic spectrum allocation regime.

Experimental research on networks of cognitive radios, however, has been hindered by the lack of mature, affordable cognitive radios that are able to operate with the full network protocol stack. This article outlines the requirements of a cognitive radio platform that is flexible enough for fundamental research on cognitive radios and cognitive networks and that is widely available to the entire research community.

The leading candidate for an open cognitive network platform is the Universal Software Radio Peripheral (USRP), with software support provided by GNU Radio, due to its availability to the research community, flexibility in operating with a wide variety of frequencies and waveforms, low cost, and reasonably broad use to date in physical layer communications research. It is worth noting that GNU Radio can work with other hardware platforms, just as the USRP can use other software platforms; we believe the use of both in tandem currently offers the best solution for an open cognitive platform. While this platform has been successfully employed by several research groups (including ours) for research into spectrum sensing and modulation classification (e.g., [1]), it is still limited in its ability to operate in large, multi-hop wireless network testbeds. We describe the current limitations of the platform and our ideas of what would be required to make it more broadly useful for cognitive networks research. Four aspects are of particular importance:

1. Support for distributed signal detection and modulation classification algorithms to allow the assessment of the effectiveness and requirements of distributed methods in detecting the presence of incumbent users and utilizing white spaces;

2. Support for a medium access control (MAC) protocol tailored to the requirements of dynamic spectrum access and in particular one that supports random access;

3. Development and implementation of a Network Knowledge Representation Language (NKRL) to store and communicate information regarding network state to cognitive elements; and

4. Development and implementation of a cognitive specification language (CSL) that describes the interface between policies and objectives and the cognitive engine.

The first two aspects address usability of an open cognitive radio platform in a wireless network, while the last two speak to the reusability of cognitive engines. We discuss each of these in later sections and present our vision for how they should be developed. We start with a brief discussion of the current state of the art in cognitive radio platforms and cognitive networks.

II. COGNITIVE RADIOS AND COGNITIVE NETWORKS

Cognitive radios promise seamless adaptation, opportunistic use of under-utilized spectrum, and selection of the modulation and waveform that best fit the current wireless environment. A simplified block diagram for a cognitive radio is shown in Fig. 1. The cognitive engine is the “brain” of the radio, responsible for reasoning and learning; it strives to
optimize an objective function, while constrained by a pre-defined set of policies. It does so by considering information it
can sense regarding the wireless environment and the state of
the network (the “dials” in the figure) and adapting its
operating parameters, such as frequency channel or type of
modulation (the “knobs” in the figure). The information
available to the cognitive engine may consist of factors the
radio can sense directly, such as channel occupancy in its
vicinity, as well as factors that are reported by other radios,
such as congestion conditions elsewhere in the network. User
requirements, as well as more general radio and network
objectives such as spectral and energy efficiency, are reflected
in the objective function. For more discussion of cognitive
radio concepts, as well as of its most common application,
dynamic spectrum access (DSA), [2] and [3] are among some
of the good tutorial papers that appear in the literature.

Despite the clear potential for efficiency gains from this
type of adaptation, such intelligent radios, when placed in a
network, may bring about unexpected and even undesirable
results (e.g., adaptation cycles, local optimizations that do not
translate into end-to-end performance improvements, or
unreasonable power, bandwidth, and computing requirements
for certain distributed operations such as collaborative
spectrum sensing) unless network considerations are carefully
explored. This has catalyzed recent interest in cognitive
networks.

In [4], we outline the features, objectives and challenges of
a cognitive network, which we define as a network with a
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
End-to-End Reconfigurability Project II (E²R-II) [5],
m@ANGEL platform [6], CTVR at Trinity College [7], and
the Institute for Wireless Networks at RWTH Aachen
University [8] have also proposed architectures at various
degrees of maturity for end-to-end oriented, autonomous
networks of cognitive and/or adaptive radios.

While analytical and simulation research on cognitive
networks has made great strides over the past few years,
experimental work on self-organizing networks of cognitive
radios has lagged, due primarily to the lack of mature
cognitive radio hardware and software platforms that are
capable of supporting a realistic network protocol stack.

It is of note that DARPA has been funding experimental
work on the development of spectrum agile radios [9] and of
large-scale adaptive networks, through the NeXt Generation
(XG) and Wireless Network after Next (WNaN) programs,
respectively. However, these radio platforms are not available
for experimentation by academic researchers outside the scope
of military communications. There seems to be interest in
developing a non-military version of the WNaN radio to be
available to researchers worldwide, but it is still early to
determine how likely this is or how long it will take.

Development of adaptive radios and cognitive radio
testbeds has also progressed in academia, with the Kansas
University Agile Radio (KUAR) [10] and the Berkeley
Emulation Engine 2 (BEE2) at the University of California at
Berkeley [11] examples of particular note. These efforts,
however, have not as yet focused on producing hardware and
software platforms that would enable widespread
experimentation in networks of cognitive radios by the
research community. Rice University’s Wireless open-Access
Research Platform (WARP) [12] is an FPGA-based platform
that supports implementation and testing at the physical and
network layers. It is now being commercialized and starting to
gain traction in the research community, although price and
availability are still obstacles to large-scale network
experimentation.

Similarly, industry is engaged in the development of
spectrum agile radios for specific application scenarios, such
as the opportunistic use of vacant TV spectrum envisioned by
the IEEE 802.22 working group, but is unlikely to produce
open platforms that can be used for general experimental
research. There is critical need for an inexpensive, widely
available, and flexible platform that supports real-time
adaptation of communication parameters and that is able to
operate with a full network protocol stack.

Next, we discuss each of the four aspects outlined in
Section I, needed to achieve such an open platform for
cognitive network experimentation.

III. DISTRIBUTED SPECTRUM SENSING

A cognitive radio’s ability to reliably and quickly perform
spectrum sensing, whether independently or collaboratively, is
an important aspect of opportunistic spectrum usage. If the
spectrum state estimate is unreliable, outdated, or biased,
cognitive radio networks may lead to harmful interference.
Although significant progress has been made in the
development of sensing algorithms (see among others [13-
15]), research and experimental results on the implementation
of such algorithms in radio platforms is still relatively scarce
[10, 16-20]. Additionally, work in this area typically relies
either on the use of platforms developed “in-house,” such as
BEE2 and KUAR, and/or on the use of expensive spectrum
analyzers for signal processing. To enable broader and faster

![Figure 1. Simplified block diagram of a cognitive radio.](image)
development of cognitive systems, open-source “spectrum sensing blocks” that use publicly available hardware and software platforms are needed.

As previously mentioned, the USRP + GNU Radio combination is a proven and viable candidate for research on the physical layer (and higher layers) of cognitive radio systems. This platform has been used for spectrum sensing research at Rutgers, Trinity College Dublin, and the US Department of Defense, for example [17-19]. However, the design and implementation of efficient signal detection and modulation classification algorithms on this platform is still an open problem.

The design of low-complexity and robust (to RF/analog circuit impairments, for example) sensing algorithms is critical for the development of an experimental cognitive radio network. In addition, these algorithms must be able to reliably detect and classify signals with little or no a priori knowledge of the channel conditions and of the received signal (e.g., timing parameters, bandwidth). In this context, cyclic feature-based detectors and classifiers have been widely considered for cognitive systems [14, 16-17, 19-22]. In particular for modulation classification, typical cyclic feature-based methods involve three steps: the estimation of the cyclic spectrum of the received signal; the extraction of features of the cyclic spectrum (which provide unique modulation specific characteristics); and the processing of these features by a pattern matching algorithm. However, the time and computational cost of estimating the received signal’s cyclic spectrum may become prohibitive for practical applications and scenarios (e.g., interference-limited environments or a large number of hypotheses). A possible solution to this problem was presented in [16] for signal detection. This solution states that if the set of signals of interest are known, only certain points (or regions) of the cyclic spectrum need to be estimated (as opposed to the complete spectrum). These points can be chosen in such a way as to provide characteristics that are unique to a given modulation scheme. Signal detection and modulation classification methods based on the use of wavelets are also receiving a great deal of interest for cognitive radio applications [15], [23]. Of note is the development of a wavelet-based spectrum sensing algorithm in CMOS chips at Georgia Tech [23].

Due to the spatial variability of radio signals, it is only natural to perform spectrum sensing in a distributed manner (by the joint work of geographically dispersed radios). Distributed sensing methods have the potential to greatly increase the spectrum estimation reliability, reduce the detection time, and ultimately decrease the probability of interference of cognitive networks to other systems [18, 21-22, 24-26]. For example, Fig. 2 shows the receiver operating characteristics of the distributed detection of a known signal as a function of the number of participating radios. In Table I, distributed modulation classification results are compared with the case in which modulation classification is performed by a single radio. For example, from this table, it can be seen that the average probability of classification error in the single radio case is approximately 5.16% but drops to approximately 0.21% when spectrum sensing is performed collaboratively by four radios. The significant improvement in performance obtained by using a distributed approach seen in both Fig. 2 and Table I is noteworthy.

By using an experimental cognitive network, different aspects that impact distributed spectrum sensing can be analyzed. These include the scheduling of sensing intervals (when, and by what radios, should sensing information be obtained), and the design of low-complexity fusion rules and sensing network architectures (cognitive radio clustering, information overload, and power/bandwidth requirements). The analysis of timing aspects of distributed spectrum sensing is important, as the sensing stage must be able to rapidly detect changes in the spectrum utilization. The performance degradation (e.g., lower probability of signal detection for a given probability of false alarm) due to the use of dated sensing information by the spectrum access protocol can be significant, as shown in [22]. Such experimental analyses have the potential to allow for the characterization of realistic spectrum sensing limits, and ultimately determine the viability of “spectrum sharing” cognitive radio systems.

![Figure 2. Distributed detection of a known signal in AWGN (coherent detector).](image-url)
TABLE I. PROBABILITY OF CORRECT CLASSIFICATION FOR THE: (A) SINGLE RADIO CASE; (B) DISTRIBUTED CASE (4 COLLABORATIVE RADIOS) USING A CYCLIC FEATURE CLASSIFIER. \( E_b/N_0 = -2\text{dB} \). [20]

<table>
<thead>
<tr>
<th>Hypothesis</th>
<th>Noise</th>
<th>BPSK</th>
<th>QPSK</th>
<th>FSK</th>
<th>MSK</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noise</td>
<td>0.9721</td>
<td>0.0020</td>
<td>0.0003</td>
<td>0.0000</td>
<td>0.0150</td>
</tr>
<tr>
<td>BPSK</td>
<td>0.0062</td>
<td>0.9780</td>
<td>0.0015</td>
<td>0.0067</td>
<td>0.0780</td>
</tr>
<tr>
<td>QPSK</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.9357</td>
<td>0.0000</td>
<td>0.0420</td>
</tr>
<tr>
<td>FSK</td>
<td>0.0001</td>
<td>0.0103</td>
<td>0.0001</td>
<td>0.9933</td>
<td>0.0022</td>
</tr>
<tr>
<td>MSK</td>
<td>0.0216</td>
<td>0.0097</td>
<td>0.0624</td>
<td>0.0000</td>
<td>0.8628</td>
</tr>
</tbody>
</table>

(a)

<table>
<thead>
<tr>
<th>Hypothesis</th>
<th>Noise</th>
<th>BPSK</th>
<th>QPSK</th>
<th>FSK</th>
<th>MSK</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noise</td>
<td>0.9985</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>BPSK</td>
<td>0.0001</td>
<td>0.9998</td>
<td>0.0000</td>
<td>0.0008</td>
<td>0.0003</td>
</tr>
<tr>
<td>QPSK</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.9949</td>
<td>0.0000</td>
<td>0.0027</td>
</tr>
<tr>
<td>FSK</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.9992</td>
<td>0.0000</td>
</tr>
<tr>
<td>MSK</td>
<td>0.0014</td>
<td>0.0002</td>
<td>0.0051</td>
<td>0.0000</td>
<td>0.9970</td>
</tr>
</tbody>
</table>

(b)

Let us now move up the protocol stack to identify the needs for medium access control mechanisms in an open cognitive radio platform.

IV. MEDIUM ACCESS CONTROL FOR DYNAMIC SPECTRUM ACCESS

Creating a medium access control protocol for a cognitive network is a difficult problem due to excessive latency involved in baseband signal processing by many software defined radio (SDR) platforms and the need to support dynamic access to multiple channels. We discuss both of these challenges in this section.

A. SDR Framework Latency

Most of the cognitive networks that have been envisioned utilize cognitive radios as network nodes. These cognitive radios, in turn, are almost always built using SDR platforms. A radio transceiver on an SDR platform usually consists of several signal processing units that are connected together in a structure, commonly called a flowgraph, which generates and/or demodulates the desired waveform. Fig. 3 shows an example of a flowgraph for a transceiver implemented for GNU Radio architecture. In many such platforms, particularly those enabling real-time radio reconfiguration, some, if not all, of the baseband signal processing may be performed on general purpose processors (GPPs); in GNU Radio, all baseband signal processing is performed on a GPP. This design choice reflects the fact that GPPs provide maximal flexibility, ease of programming, and availability of tools for reconfiguration. Even if the signal processing blocks themselves are implemented in custom or reconfigurable logic (e.g. ASICs or FPGAs), the middleware used to wire the blocks together may require a GPP; this is the case with the Software Communications Architecture (SCA) (and hence in Joint Tactical Radio System radios) which requires a particularly heavyweight middleware (CORBA) to connect components together. Typically, these signal processing block interconnections are buffered to provide processing efficiencies and to allow the GPP some scheduling flexibility. Unfortunately, the reliance on a GPP for signal processing and/or buffered signal processing block interconnection introduces significant latency to the modulation and demodulation process.

Due to the bursty nature of data traffic, most general-purpose data networking standards (e.g. the entire IEEE 802 family of standards) rely heavily on random access MAC protocols. Specifically, the MAC protocol chosen is usually a derivative of carrier-sense multiple access (CSMA) because this random access protocol typically provides high efficiency. Fig. 4 shows the theoretical efficiency of the CSMA MAC protocol as a function of \( \beta \), the ratio of the sensing time to the average packet time. As one might expect, the performance is heavily influenced by this receive-transmit turnaround time. Namely, CSMA performance falls abruptly if too much time elapses between the time when one stops sensing the channel and the time that one begins transmitting because the probability of a collision grows quite large.

![Flowchart of radio transceiver](figure3.png)

Figure 3. Flowchart of a transceiver implemented for the GNU Radio platform.
to maximize network throughput?

First, how can a set of nodes with disparate views of each other and established communications, several challenges remain:  Even once nodes have located and establish a communication link on a common channel) is challenging in and of itself. With packet sizes of 1000 kB and a data rate of 1 Mbps, this leads to a theoretical maximum efficiency of the CSMA MAC of 59%. While data rates of 1 Mbps are unobtainable with the current USRP, we expect them to be easily obtainable with the soon-to-be-released USRP 2. Moreover, this is a reasonable minimal data rate for broadband wireless communications.)

Unfortunately, the long latencies introduced by the SDR frameworks make it nearly impossible to achieve good MAC efficiency at reasonably high data rates (higher data rates decrease the denominator of $\beta$). We interpret recent work in [26] to suggest that the GNU Radio framework will require a turnaround time of at least 1 ms. With packet sizes of 1000 kB and a data rate of 1 Mbps, this leads to a theoretical maximum efficiency of the CSMA MAC of 59%. While data rates of 1 Mbps are unobtainable with the current USRP, we expect them to be easily obtainable with the soon-to-be-released USRP 2. Moreover, this is a reasonable minimal data rate for broadband wireless communications.)

### B. Dynamic Spectrum Access

Opportunistic spectrum access introduces additional complexity to the MAC problem. Although there has been significant work on signal detection and classification, the rendezvous problem (the ability of two or more radios to meet and establish a communication link on a common channel) is challenging in and of itself. Even once nodes have located each other and established communications, several challenges remain: First, how can a set of nodes with disparate views of spectral availability efficiently exploit the available spectrum to maximize network throughput? It is critical that the MAC protocol perform efficiently no matter how much or how little bandwidth is available (i.e. the protocol must work efficiently even if there is only enough bandwidth for a single, shared, low data rate channel). Second, how can a MAC protocol cope with different node capabilities? Much of the early work on multi-channel MAC has assumed that nodes have multiple transceivers, but a good MAC for dynamic spectrum access should also be able to accommodate nodes with a single transceiver.

Table II summarizes some of the requirements of a MAC that supports DSA and which of those characteristics multi-channel MAC protocols proposed in the literature currently meet. Neighbor discovery and rendezvous are made more complex by two facts: (i) the set of potential neighbors may occupy several among multiple available channels; and (ii) secondary nodes may have to vacate a channel due to the appearance of an incumbent user and reconvene in a different channel. In other words, the set of channels available to a node changes dynamically. Also, the MAC may need to interact with the spectrum sensing module to participate in distributed sensing schemes such as discussed in the previous section.

A MAC protocol that supports random access and addresses the additional requirements brought about by DSA is needed for the USRP + GNU Radio to operate in a cognitive network. Our vision for this is outlined next.

Using distributed spectrum sensing, nodes may be able to identify available portions of the spectrum and detect and rendezvous with other detected cognitive nodes. The rendezvous process, including bootstrapping from common control channels and/or blind rendezvous in one of the currently available channels, should be part of the MAC design. To overcome the SDR framework latency problem while preserving the benefits of random access medium access control, the MAC protocol may rely on a hybrid of random and scheduled access. Namely, nodes can transmit short, random access frames in order to attempt to claim future channel resources for actual data transmission. It is impractical to rely on tight frame synchronization, which would be difficult to achieve, to delineate random and scheduled access periods. Instead, all non-reserved channel time can be used for random access reservations, while data transmission times can be scheduled relative to the time of the reservation request. We anticipate a process similar to the IEEE 802.11 four-way handshake: when a node detects no current or pending reservations on the channel, it transmits an RTS frame indicating a desire to send a packet to a specific neighbor. Upon successful receipt of an RTS, the intended destination replies with a CTS frame, which reserves the channel for a data transmission to start imminently (after a predetermined interval). In order to address the dynamic spectrum access problem, RTS and CTS frames can be transmitted on mutually agreed control channels. (The process of dynamically agreeing on a control channel is also part of the proposed rendezvous process.) RTS frames may include a selection of possible channels and modulation schemes for data transmission, while CTS frames would identify a chosen set of transmission parameters. The MAC must also have appropriate logic for managing multiple transceivers and a variety of channel conditions.

Although a CSMA implementation for GNU Radio currently exists (and is distributed with the GNU Radio software), it is neither robust nor extensible. The current implementation is not robust because it provides no MAC-layer acknowledgements. As a result, the transport layer retransmits any lost packets, incorrectly interprets these losses as reflecting network congestion, and responds by needlessly throttling the connection. The current implementation is not extensible, because it is tightly integrated into a single transceiver flowgraph that implements a single modulation scheme. For modularity and reusability, the MAC control must be separate from the primary transmit and receive flowgraphs, thereby making it possible to use the implemented MAC protocols with any transceiver flowgraph that implements appropriate control interfaces.

![Figure 4. CSMA efficiency versus $\beta$, the ratio of sensing time to average packet time.](image-url)
TABLE II. DSA MAC REQUIREMENTS AND SUMMARY OF MULTI-CHANNEL MAC PROTOCOLS IN THE LITERATURE

<table>
<thead>
<tr>
<th>Protocol/MAC Type</th>
<th>Neighbor discovery</th>
<th>Rendezvous</th>
<th>Random Access</th>
<th>Channel Vacate</th>
<th>Coordinated Sensing</th>
<th>No. Control Channels</th>
<th>Dynamic Set of Channels</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi-channel MAC (McMAC) [28]</td>
<td>Blind, Accidental</td>
<td>Hop and RTS/CTS</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>0</td>
<td>No</td>
</tr>
<tr>
<td>Cognitive MAC (C-MAC) [29]</td>
<td>Blind</td>
<td>Scheduled NAV</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>1</td>
<td>Yes</td>
</tr>
<tr>
<td>Dedicated Control Channel [30]</td>
<td>Assumed</td>
<td>RTS/CTS</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>1</td>
<td>No</td>
</tr>
<tr>
<td>Common Hopping (CHMA [31], HRMA [32], RICHDP [33])</td>
<td>Assumed</td>
<td>RTS/CTS on current hop</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>0</td>
<td>No</td>
</tr>
<tr>
<td>Slotted Seeded Channel Hopping (SSCH) [34]</td>
<td>Assumed</td>
<td>Wait for hop</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>0</td>
<td>No</td>
</tr>
<tr>
<td>Split Phase (MAP) [35]</td>
<td>Assumed</td>
<td>Schedule NAV</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>1</td>
<td>No</td>
</tr>
<tr>
<td>On Demand Channel Switching [36]</td>
<td>Mildly assumed</td>
<td>RTS/CTS (multi-channel)</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>0</td>
<td>No</td>
</tr>
</tbody>
</table>

V. INTERFACE LANGUAGES FOR COGNITIVE RADIOS

The MAC layer and distributed spectrum sensing elements described earlier provide the foundation for the cognitive network. Beyond these physical requirements to sense the RF environment and arbitrate access to shared spectrum, cognitive networks require the ability to share knowledge and interpret end-to-end objectives. These two functions are accomplished via two interfaces: an interface to the other cognitive elements in the network, and an interface to the sources of the network’s end-to-end objectives. These interfaces require languages to pass messages between the cognitive elements and these entities – messages from other cognitive elements and messages from applications, users and processes requiring end-to-end support. In this section we discuss the languages used to communicate and represent information at each interface. In particular, the Network Knowledge Representation Language (NKRL) stores and communicates knowledge between cognitive elements and the Cognitive Specification Language (CSL) bridges the interface between the end-to-end goals and the cognitive elements. These interfaces and the role of the languages in them are illustrated in Fig. 5.

The topic of storing and exchanging knowledge has been examined in the cognitive radio framework, with Mitola describing a knowledge language called Radio Knowledge Representation Language (RKRL) [37]. RKRL is an instantiation of Knowledge Query Markup Language (KQML), an interaction language developed for communication between software agents [38]. KQML itself is based on Standard Generalized Markup Language (SGML) and is designed to express and exchange information between intelligent agents and other entities, such as applications or other agents. In addition to exchanging information, KQML is designed to make and respond to requests for information, as well as locate qualified agents.

RKRL, according to [37], consists of: mappings between the real world and the various models formed by the cognitive process; a syntax defining the statements of the language; models of time, space, entities and communications among entities such as people, places, and things; an initial set of knowledge including initial representation sets, definitions, conceptual models, and radio domain models; and mechanisms for modifying and extending RKRL. KQML, and by association, RKRL, make some demanding assumptions as to the availability and reliability of communication channels. In particular, it is assumed that there are distinguishable connections between all agents and that the connections are reliable and preserve the order of messages [39]. These
assumptions are appropriate for the local, point-to-point scope that the cognitive radio operates with, but are limiting for a dynamic, complex network of connections.

To attempt to address a language for higher level goals, [40] suggests extending RKRL to encompass the high-level goals of the users of the network. Such an extension is called a Network Knowledge Representation Language (NKRL). It is reasonable to take the NKRL suggestion as a starting point for the goal of creating an open, extensible language for representing and transacting knowledge in a cognitive network.

The requirement here is to develop the framework for a NKRL that captures the core components as RKRL, but extends them to handle the complexities of the entire network environment. NKRL must support the greater uncertainty and collaborative potential of the network environment. Whereas RKRL focuses on modeling the knowledge that a radio has, NKRL must be capable of representing the knowledge that a network node does not have, including uncertain, missing and indeterminate information. To this end, NKRL should be able to represent “compressed” versions of the knowledge, to avoid flooding other cognitive elements with too much information, including information that is either superfluous or repetitive. A NKRL can be evaluated according to how effectively it synchronizes, validates, and updates knowledge in dynamic and lossy network environments.

At the other side of the cognitive process, an interface is needed to connect the end-to-end objectives of the network into the cognitive process. In a cognitive network, this role should be performed by a Cognitive Specification Language (CSL), providing behavioral guidance to the cognitive elements by describing, in a formal manner, the end-end goals for eventual use as cognitive element objectives. A CSL can be thought of as the architectural “blueprints” of the network behaviors. The CSL represents the end-to-end goals of the network in a standard, abstract fashion that various cognitive elements can interpret and act on. This is similar to how different construction contractors can use the same set of blueprints to determine their specific tasks, from correctly framing the building to running the venting system.

The CSL is not used to represent the observed network environment, as would be found in the NKRL. Instead, it is more analogous in scope and intention to a Quality of Service (QoS) specification language. These languages are used to represent QoS requirements to the various mechanisms that the network offers to support them. There are already several different QoS specification paradigms in existence and the concept of these languages – mapping requirements to underlying mechanisms – is the same here, except that the mechanisms are adaptive to the network capabilities as opposed to a fixed set of QoS capabilities.

In establishing criteria for a successful CSL, we adapt the criteria for a good QoS specification language, as described in [41]. The following criteria represent design objectives for an effective CSL. A language that does not meet any or all of these requirements may still perform the role of a CSL, albeit less effectively.

- **Expressiveness**: A CSL must be able to specify a wide variety of end-to-end goals. It should be able to express constraints, goals, priorities and behaviors to the cognitive elements that make up the process. It should be able to express new goals without requiring a revision in the language.
- **Cognitive process independence**: The cognitive process architecture and functionality should not dictate the CSL. Instead, the CSL should abstract away as much of the cognitive process as possible to the application, user, or resource. This allows a goal to be used with different cognitive processes with little modification and promotes re-usability.
- **Interface independence**: Whether the cognitive process is distributed or centralized in operation, autonomous or aggregated in architecture, the user should be presented as abstract an interface as possible. Like the previous criteria, this abstraction promotes reusability by allowing the re-use of goals over many different cognitive processes with little effort from the top layer.
- **Extensibility**: The CSL should be extensible enough to adapt to new network elements, applications and goals, some of which may not even be imagined yet.

The effectiveness of an open CSL that accomplishes the design requirements above is determined by selecting several basic end-to-end objectives and cognitive processes (with differing interfaces and functionality), and then having the CSL represent these objectives to the process.

VI. CONCLUSIONS

In this paper, we described our vision of the building blocks needed to create an open platform for cognitive radios that is capable of operating with the full protocol stack. These include mechanisms for distributed spectrum sensing, a MAC protocol tailored to dynamic spectrum access, and interface languages for cognitive networks.

The availability of such a platform will ultimately accelerate the deployment of cognitive networks through experimental work and proof-of-concept demonstrations.

We argued that the USRP + GNU Radio is currently the leading candidate for an open platform for cognitive network experimentation. However, other approaches are fast coming to maturity. One such approach, using FPGAs to realize custom signal processing pipelines, is presented in [42]. In this approach, the signal processing structures are synthesized in advance and dynamically assembled at run-time. A widely-available hardware platform based on this idea is yet to emerge, but it would address many of the current limitations in processing power and latency exhibited by existing ones. The rapid availability and commercialization of Rice’s WARP also positions it well to potentially become a de facto standard platform for cognitive network research.

ACKNOWLEDGMENTS

We gratefully acknowledge the contributions of our graduate students William C. Headley, Jesse D. Reed, and Ryan E. Irwin to some of the results reported here. Headley and Reed conducted the experiments that produced the results
reported in Table I, while Irwin tabulated the comparison of MAC protocols presented in Table II.

REFERENCES


