SAMPLE WORKPLAN
Charles Bostian, "NetS-ProWIN: An Enabling Technology for Wireless Networks"

1. Overview and Objectives

This proposal presents the enabling technology for wireless networks consisting of intelligent nodes sharing a distributed knowledge base and capable of both individual and collective reasoning and learning. The nodes are cognitive radios (CRs): intelligent machines that can determine the best way to operate in a given situation and configure themselves accordingly. Like living creatures, they are aware of their surroundings, their own and their peers' capabilities, and the governing social constraints. Their actions arise from a rational process that predicts probable consequences and remembers past successes and failures.

A CR is a software defined radio with a "cognitive engine" brain. Conceptually, the cognitive engine responds to the operator's commands by configuring the radio for whatever combinations of waveform, protocol, operating frequency, and networking are required. It monitors its own performance continuously, reading the radio's outputs to determine the RF environment, channel conditions, link performance, etc., and adjusting the radio's settings to deliver the needed quality of service subject to an appropriate combination of user requirements, operational limitations, and regulatory constraints. We call these processes "turning the radio's knobs" and "reading the radio's meters" for short.

Our group is developing a distributed cognitive engine that can make any radio with electronically accessible inputs and outputs truly cognitive. A first proof of concept prototype is now operational. In this proposal we seek to apply our technology at the network level, using cognitive GNU Radios as intelligent nodes, and to build and test cognitive networks using the Rutgers University's ORBIT facility [ORBIT] and the University of Utah's Emulab [Emulab] facility. With these tests we will explore the general behavior of cognitive networks and investigate cognitive techniques for spectrum access and sharing and quality of service (QoS) provisioning. Figure 1 presents the planned timeline for the project.

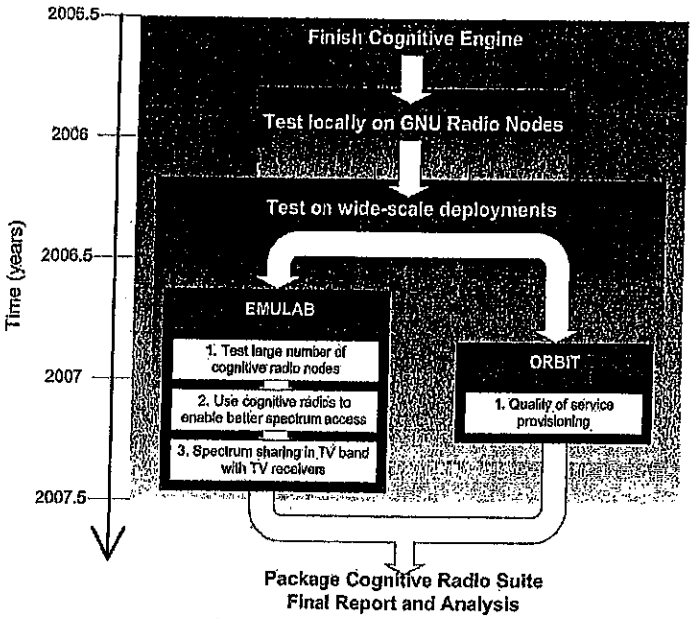

Figure 1 Project Timeline

This work takes programmable wireless networking to new levels (literally) by extending intelligence that, in the past, has at best resided at the application layer, down to the medium access control (MAC) and physical (PHY) layers. It allows nodes cooperatively to set their own data rates, transmitter power levels, modulation formats, etc., rather than limiting these quantities to constant values or to settings dictated by a central authority.

2. Relationship to Current State of Knowledge and Work in Progress

2.1 The Virginia Tech Cognitive Engine in Context

Joseph Mitola invented the basic cognitive radio concept in the late 1990s when he envisioned a CR as a universal and highly intelligent wireless personal digital assistant, operating primarily at the application level. More recently the DARPA XG program extended the concept to allow the CR to operate as an intelligent agent. Our work, based on learning algorithms, extended cognition to the MAC (medium access control) and PHY (physical) layers. This allows the cognitive process to monitor and control

processes (e.g., modulation) and settings (e.g., transmitter power) that are inaccessible from the higher layers. Fig. 2 provides a historical context for our work.

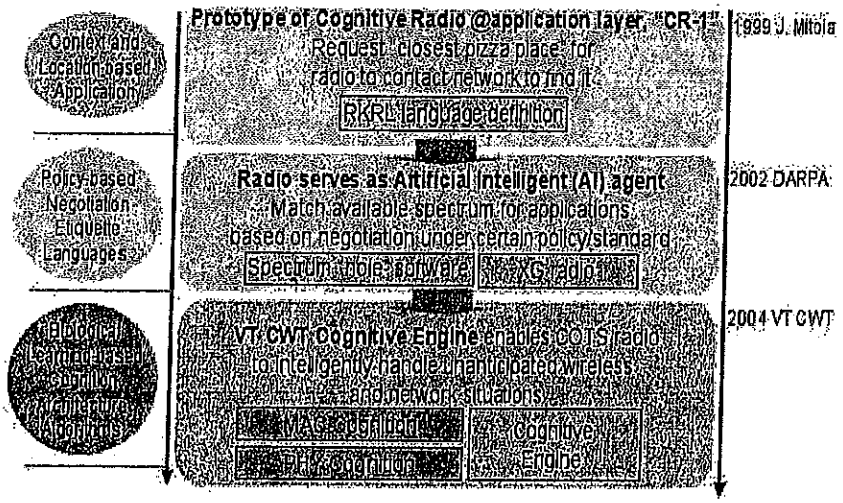

Figure 2. The VT Cognitive Engine in Historical Context

2.2 Brief Description of the VT Cognitive Engine and Its Operation

Our cognitive network research uses a distributed cognitive engine that can make any radio with electronically accessible inputs and outputs truly cognitive. Sketched in Fig. 3, it consists of three software subsystems that (1) model the environment, (2) develop, update, and maintain knowledge, and (3) develop new radio configurations through evolutionary techniques. The CSM (Cognitive System Module) is responsible for learning and the WSGA (Wireless System Genetic Algorithm) adapts the radio behavior based on what the CSM tells the WSGA to do. The CSM contains two main learning blocks: (1) the Evolver and (2) the Decision Maker. The latter takes feedback from the radio and allows the Evolver to update the knowledge base and then respond to and direct the system behavior. The third block in the diagram is the Resource Monitor, which is closely tied to the CSM and provides radio system information like available battery life, computational power, and memory resources. The final subsystem is the Modeling System, which is responsible for taking in external data such as the radio environment, user domain, security restrictions, and the regulatory policy domain and processing and modeling this information.

2.3 Relationship to Prior Results of Principal Investigator's Work

The VT cognitive engine and our work in CR grew directly from NSF award 9987586, Testbed for High-Speed 'End-to-End' Communications in Support of Comprehensive Emergency Management, C.W. Bostian PI and S.F. Midkiff Co-PI. In that project we researched communications networks for disaster response applications. A key problem was to develop radios that can find short-lived paths of opportunity and compensate for shortcomings of these paths to deliver optimum performance. An important characteristic of these paths is a phenomenon called rough-surface scattering that takes short (nanosecond) radio pulses and smears them out in time, introducing "pulse stretching" and serious distortion. This had not previously been observed at the frequencies and time scales we use. To investigate these paths we subsequently built a new type of impulse channel sounder. Inspired by the needs of the public safety community and building upon the developments of the disaster response project, we conceived a CR that would make intelligent decisions based on the sounder output, and developed a cognitive engine that will indeed allow a radio to adapt intelligently to unanticipated situations. That cognitive engine is the basis of this proposal. Subsequently we built a proof-of-concept prototype that allows legacy Proxim Tsunami ® 5 GHz radios to adapt to changing propagation and

interference conditions. This prototype is installed at SAIC's Public Safety Integration Center near Washington, DC, and is available for demonstrations there.

This work under NSF award 9987586 is described in a U.S. patent application for cognitive radio in a network [Patent, 2004], a Ph.D. dissertation, [Rieser, 2004a], and several recent conference presentations and publications [Rondeau, 2004a], [Rieser, 2004b], [Rondeau, 2004b], [Bostian, 2004], [Bostian, 2002].

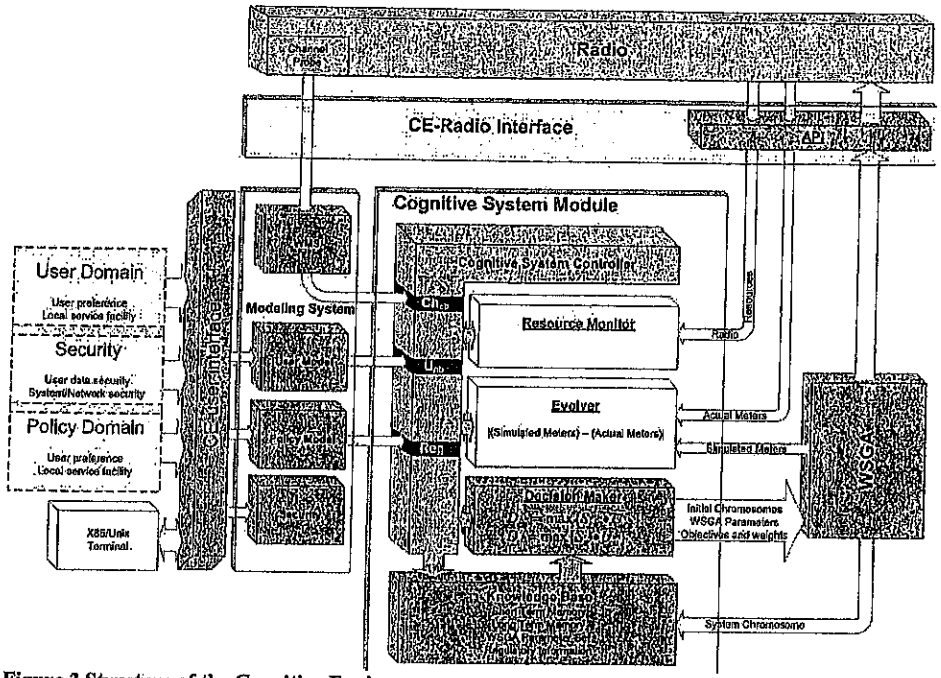

Figure 3 Structure of the Cognitive Engine

2.4 Work in Progress

Currently we are building the complete software for full scale network tests of the cognitive engine with software defined radios, and this proposal is part of that effort. Research topics include (1) monitoring and modeling the user requirements with as little overhead as possible, (2) distinguishing co-channel interferers from ambient noise and propagation impairments, (3) securing proper adaptation by all members of a cognitive network, and (4) improving learning and adaptation to make maximum use of the intelligence available in a network of cognitive radios. We are also considering the related issues of software verification and FCC certification – how do you insure yourself and convince the FCC that a software defined cognitive radio cannot transmit illegally (i.e., with too much power, on the wrong frequency, etc.), even when hacked, damaged, or given an incorrect or illegal command by its operator? We have proposed to investigate these issues for public safety radios in a submission to the National Institutes of Justice (NIJ) and will coordinate that work with the project described in this proposal.

3. Research Tasks, Methods, and Procedures

3.1 Completing the Cognitive Engine

3.1.1 Overview of Major Tasks

Our cognitive engine is already well on the way to completion, and the WSGA block is up and running as part of the disaster response project described in Section 2.3. However, much still remains to be done, and a fully functional cognitive engine is an important outcome of this project. As shown in Figure 1, making the cognitive engine ready for the proposed experiments is our first goal, but we will update and enhance it throughout the life of the project. In this section we provide a broad overview; later sections discuss implementation details and research issues in more depth.

The set of tasks listed below describe the modules we must complete before realizing the full potential of the cognitive engine. Tasks A, B, and C will be done concurrently; subtasks are listed in chronological order as determined by relative importance to the overall project timeline.

Task A: Cognitive System Monitor (CSM) Completion

Subtask A.1 Knowledge Base: The Knowledge Base is a data base representing the models, system performance, and behavior; it is the most important piece of the CSM since everything else relies on it. Many input parameters from the radio hardware, regulatory requirements, and user domains and the WSGA output parameters require access to the Knowledge Base.

Subtask A.2 Decision Maker: The Decision Maker interacts with the Knowledge Base and decides how the WSGA should act and what data it should act upon.

Subtask A.3 Resource Monitor: The Resource Monitor takes information from the radio to determine the impact of the system settings. Resources are system parameters like computational power, memory, and battery life.

Subtask A.4 Evolver: The Evolver monitors both the output of the radio and the WSGA to compare the actual performance to the simulated performance. The Evolver rewards good performance, penalizes poor performance, and uses machine learning techniques to update and evolve the Knowledge Base.

Subtask A.5: Distributed Cognition: The final cognitive engine can be distributed among all radios on the network to provide enhanced learning and faster adaptation. A shared Knowledge Base can provide all radios with knowledge learned by any one machine, thereby reducing the need to redevelop this knowledge within each node. Other benefits come with distributing the computational responsibilities between the nodes or sharing capabilities some radios may not have (like modeling systems).

Task B: Radio Hardware and Interface

Subtask B.1: Radio environment modeling: The radio environment information includes both node data (channel frequency, transmitter power, modulation, etc.) and network data (spectrum occupancy, protocols, capacity, etc.), which is important for the CR to learn and adapt intelligently. Properly representing such knowledge requires a balance between speed and fidelity.

Subtask B.2: Hardware capability knowledge: The cognitive engine needs to be aware of the functional capabilities of the hardware radio platform (including both RF functions and DSP functions) to control its operation. We are developing a universal Application Programmable Interface (API) to serve as middleware between cognitive engine and various radio platforms.

Subtask B.3: Genetic Algorithm enhancements: The cognitive engine should analyze the overall environment information (including physical link constraints, user service demands, and security/policy regulations) and make a performance optimization decision that balances these (multiple) objectives and constraints. Although a regular multi-objective genetic algorithm (MOGA) is capable of solving this problem, it must be designed in both algorithm core and interface for robustness and convergence speed under various situations. Genetic algorithms can be distributed over multiple network nodes for a global system optimization.

Subtask B.4: User interface for CR function control and feedback
The cognitive engine provides different types of user interfaces between system operators and consumers (end users).

(1) Designer interface – parametric functionality control and coordination
(2) Consumer interface – service preference options with selection suggestions
The designer interface provides radio and cognitive engine designers a standard method of communicating radio capability information like transmitter power ranges, modulation formats, coding capabilities, etc. The consumer interface offers higher level details like overall radio configuration or goals (e.g., wireless LAN or mobile phone). The interfaces will be an HTML or simple XML-style webpage with a user friendly interface. [Ray, 2004]

Task C: Regulation, Security, and Performance Guarantees

Subtask C.1: The Policy Domain module will provide abstract of the policy and regulatory compliance information. The policy information will be obtained from the radio's own awareness, taking into account the time, space, and rules that regulate any given operational band under consideration.

Subtask C.2: The Security module will focus on two separate issues: overall network security authentication and data security, given the user's application needs.

Subtask C.3: The User Domain module will focus on how to model the user's preferences against the local service offerings. This will be done from a network standpoint to further enhance the cognitive engine's capabilities to improve network performance. Our current research uses neural networks to learn patterns of user behavior.

3.1.2 Stability, Regulatory Compliance, and FCC Certification Methodology

An issue of paramount importance to practical cognitive networking is FCC certification. Because a malfunction or an erroneous output produced by the CR software can cause serious problems, the need to build correct, secure, and safe software in such a system is a top concern. In particular, with the underlying genetic algorithms' framework for evolving the optimal set of parameters for the CR, we must ensure that the evolved chromosome (set of parameters) does not violate FCC regulations or network protocol.

To successfully tackle this problem, we will address two related but distinct issues: (1) the specification of safety properties and (2) the verification of the software against the specification. For the specification of safety properties, we not only need to define safe and/or illegal properties, we also need to offer an efficient model with which users can easily check to see if any property has been violated. Next, in the problem of verification against the specification, we offer a powerful and effective verification engine to check if the implemented software abides by the specification.

To solve the problem of specifying safety, we will gather an initial constraint set of radio parameters (values that the parameters must not be allowed to have), provided by the designers (with input invited from the FCC). Note that we gather illegal radio parameters rather than legal parameters since the space of all legal inputs can potentially be enormous. Further, illegal parameter settings may be represented by a very small subset of parameters, whereas determining legal setting requires examining all parameters. As we are modeling illegal space, the discussion that follows will be in "negative logic."

We will illustrate the use of negative logic via two simple examples. We first define a predicate to be a logical expression bounding the value range of any parameter, e.g., predicate p1 = (parameter a > 0). Consider a simple condition p1 $\rightarrow$ p2 (p1 implies p2). Logic implications can be converted to Boolean logic expressions. In this particular case, the implication is converted to ($\neg$p1 $\vee$ p2), where $\neg$ denotes the negation operator and $\vee$ denotes the logical OR operator. ($\neg$p1 $\vee$ p2) is called a clause, and at any given time, this clause must evaluate to true. In this example, whenever p1 is true, p2 is implied to be true. On