An Architecture to Support Cognitive-Control of SDR Nodes

Karen Zita Haigh
khaigh@bbn.com

Roles for AI in Networking

- Cyber Security
- Network Configuration (which modules to use)
- Network Control (which parameter settings to use)
- Policy Management
- Traffic Analysis

- Sensor fusion / situation assessment
- Planning
- Coordination
- Optimization
- Constraint reasoning
- Learning (Modelling)
  - Complex Domain
  - Dynamic Domain
  - ➢ Unpredictable by Experts

AI enables real-time, context-aware adaptivity

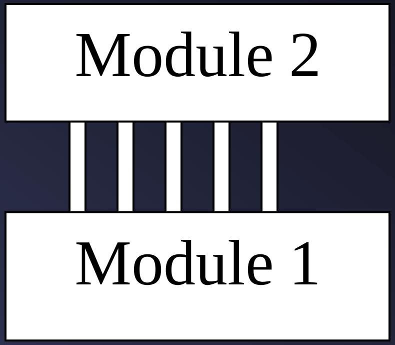
Network Control is ready for AI

- **Massive Scale**: ~600 observables and ~400 controllables *per node.*
- **Distributed:** each node must make its own decisions
- **Complex Domain:**
  - Complex & poorly understood interactions among parameters
  - Complex temporal feedback loops (at least 3: MAC/PHY, within node, across nodes); High-latency
- **Rapid decision cycle:** one second is a *long* time
- **Constrained:** Low-communication: cannot share all knowledge
- **Incomplete Observations:**
  - Partially-observable: some things can not be observed
  - Ambiguous observations: what caused the observed effect?

Human network engineers can't handle this complexity!

A Need for Restructuring

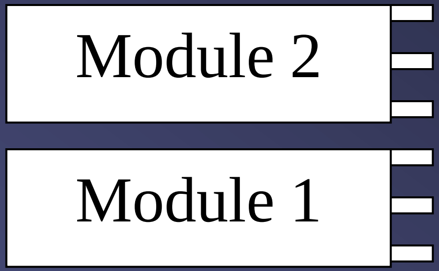
- SDR gives opportunity to create highly-adaptable systems, BUT
  - They usually require network experts to exploit the capabilities!
  - They usually rely on module APIs that are carefully designed to expose each parameter separately.



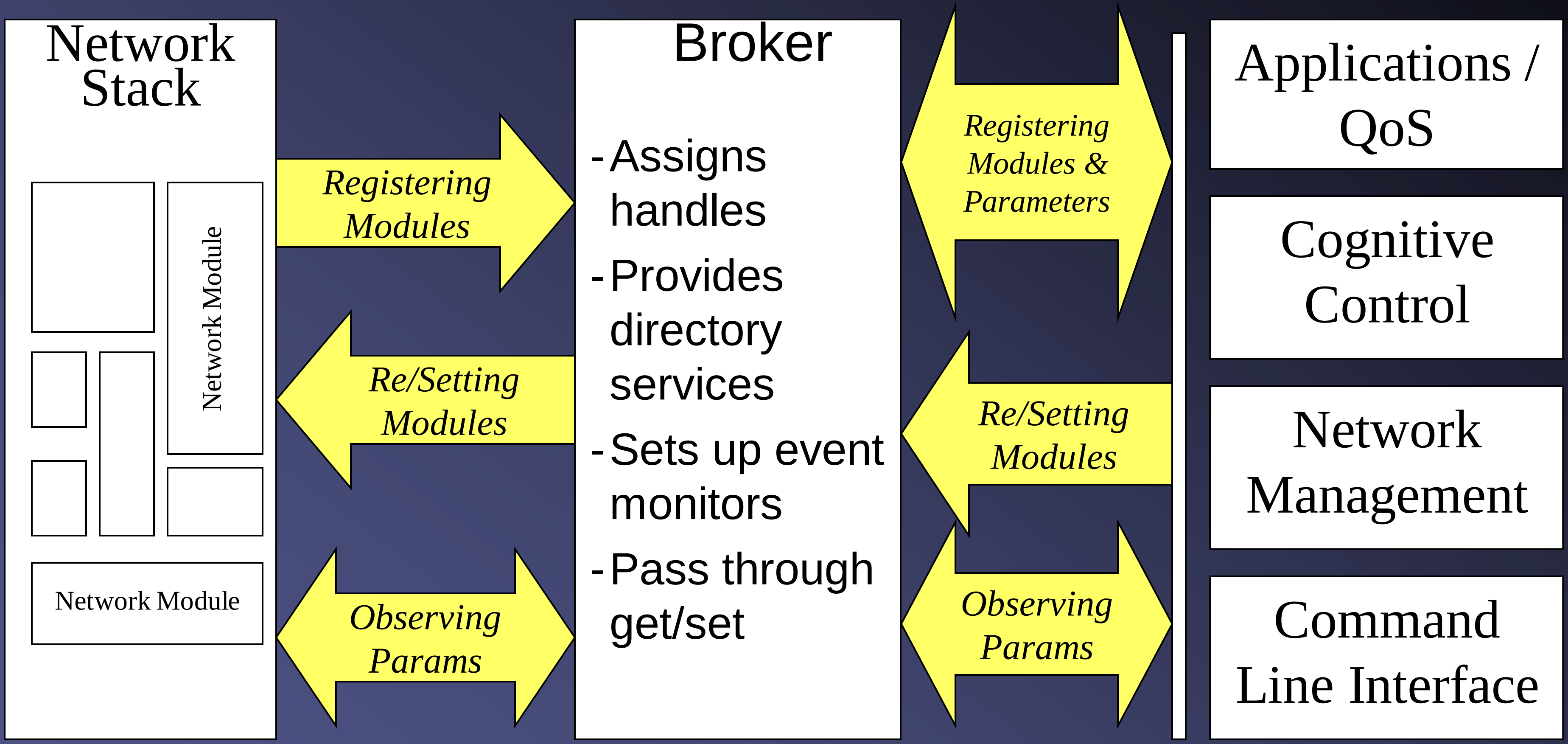
- This approach is not maintainable
  - e.g. as protocols are redesigned or new parameters are exposed.
- This approach is not amenable to real-time cognitive control
  - Hard to upgrade
  - Conflicts between module & AI

A Need for Restructuring

- We need one consistent, generic, interface for all modules to expose their parameters and dependencies.



A Generic Network Architecture



exposeParameter(*parameter_name*, *parameter_properties*)
setValue(*parameter_handle*, *parameter_value*)
getValue(*parameter_handle*)

Benefits of a Generic Architecture

- It supports network architecture design & maintenance
  - Solves the *nxm* problem (upgrades or replacements of network modules)
- It doesn't restrict the form of cognition
  - Open to just about any form of cognition you can imagine
  - Supports *multiple* forms of cognition on each node
  - Supports *different* forms across nodes

An example:

Adaptive Dynamic Radio Open-source Intelligent Team (ADROIT)

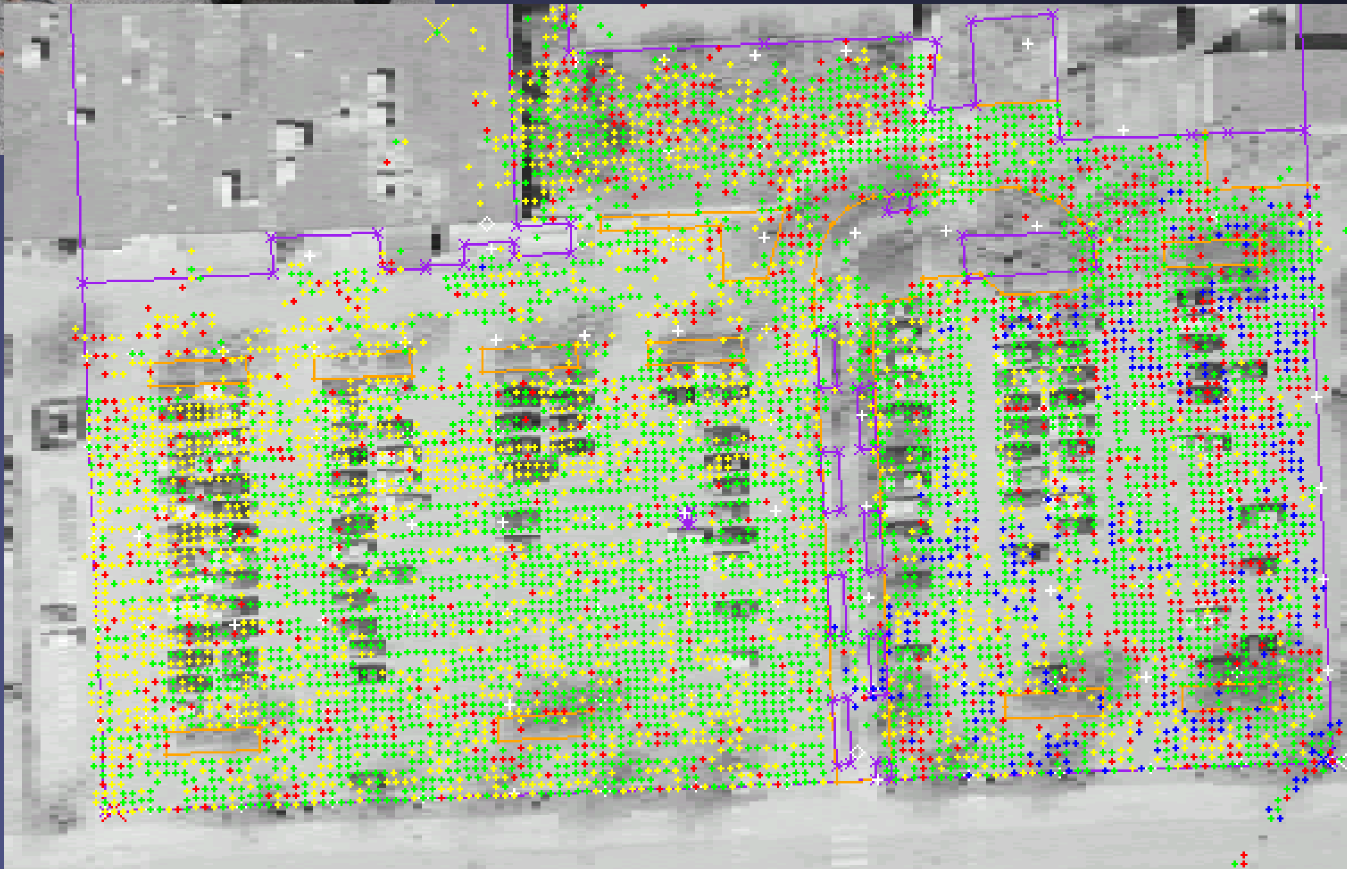
BBN, UKansas, UCLA, MIT

ADROIT's mission

- DARPA project
- Create cognitive radio teams with both *real-time composability* of the stack and *cognitive control* of the network.

- Recognize that the situation has changed
- Anticipates changes in networking needs
- Adapts the network, in real-time, for improved performance
  - Real-time composability of the stack
  - Real-time Control of parameters
  - On one node or across the network

Experimental Testbed



Maximize % of shared map of the environment

Experiment Description

- Maximize % of shared map of the environment
- **Goal:** Choose Strategy to maximize expected outcome given Conditions.
  - Each node chooses independently, so strategies must be interoperable
- Measure conditions
  - signal strength from other nodes
  - location of each node

Strategies:

- 2 binary strategy choices for 4 strategies

1. How to send fills to nodes without data?
   - multicast, unicast
2. When to send fills?
   - always
   - if we are farthest (and data is not ours), refrain from sending

Experimental Results

Training Run:

- In first run nodes learn about environment
- Train neural nets with (C,S)$\rightarrow$P tuples
  - Every 5s, measure and record progress conditions, strategy
  - Observations are local, so each node has different model!

Real-time learning run:

- In second run, nodes adapt behavior to perform better.
- Adapt each minute by changing strategy according to current conditions

Real-time cognitive control of a real-world wireless network

Observations from Learning

System performed better with learning

- Selected configurations explainable but not predictable
  - Farthest-refraining was usually better
    - congestion, not loss dominated
  - Unicast/Multicast was far more complex
    - close: unicast wins (high data rates)
    - medium: multicast wins (sharing gain)
    - far: unicast wins (reliability)

Overcoming Cultural Differences to Get a Good Design

Cultural Issues: But why?

- Benefits and scope of cross-layer design:
  - More than 2 layers!
  - More than 2-3 parameters per layer

  - ➢ Drill-down walkthroughs highlighted benefits to networking folks; explained restrictions to AI folks
  - ➢ Simulation results for specific scenarios demonstrated the power

- Traditional network design includes adaptation
  - But this works against cognition: it is hard to manage *global* scope
  - AI people want to control everything
  - But network module may be better at doing something focussed

  - ➢ Design must include *constraining* how a protocol adapts

Cultural Issues: But how?

- Reliance on centralized Broker:
  - Networking folks don't like the single bottleneck
  - ➢ Design must have fail-safe default operation

- Asynchrony and Threading:
  - AI people tend to like blocking calls.
    - e.g. to ensure that everything is consistent
  - Networking folks outright rejected it.
  - ➢ Design must include reporting and alerting

Cultural Issues: But it'll break!?!

- Relinquishing control outside the stack:
  - Outside controller making decisions scares networking folks
  - AI folks say "give me everything & I'll solve your problem"

  ➢ Architecture includes "failsafe" mechanisms to limit both sides

- Heterogenous and *non-interoperable* nodes
  - Networks usually have homogeneous configurations to maintain communications
  - AI likes heterogeneity because of the benefit
    - But always assumes safe communications!

  ➢ "Orderwire" bootstrap channel as backup

Cultural Issues: New horizons?

- Capability Boundaries
  - –Traditional Networking has very clear boundary between “network” and “application”
  - –Generic architecture blurs that boundary
    - AI folks like the benefit
    - Networking folks have concerns about complexity

➢ Removing this conceptual restriction will result in interesting and significant new ideas.

Conclusion

- Traditional network architectures do *not* support cognition
  - Hardware is doing that now (SDR), but the software needs to do the same thing

- To leverage the power of cognitive networking, both AI folks & Networking folks need to recognize and adapt

Backup

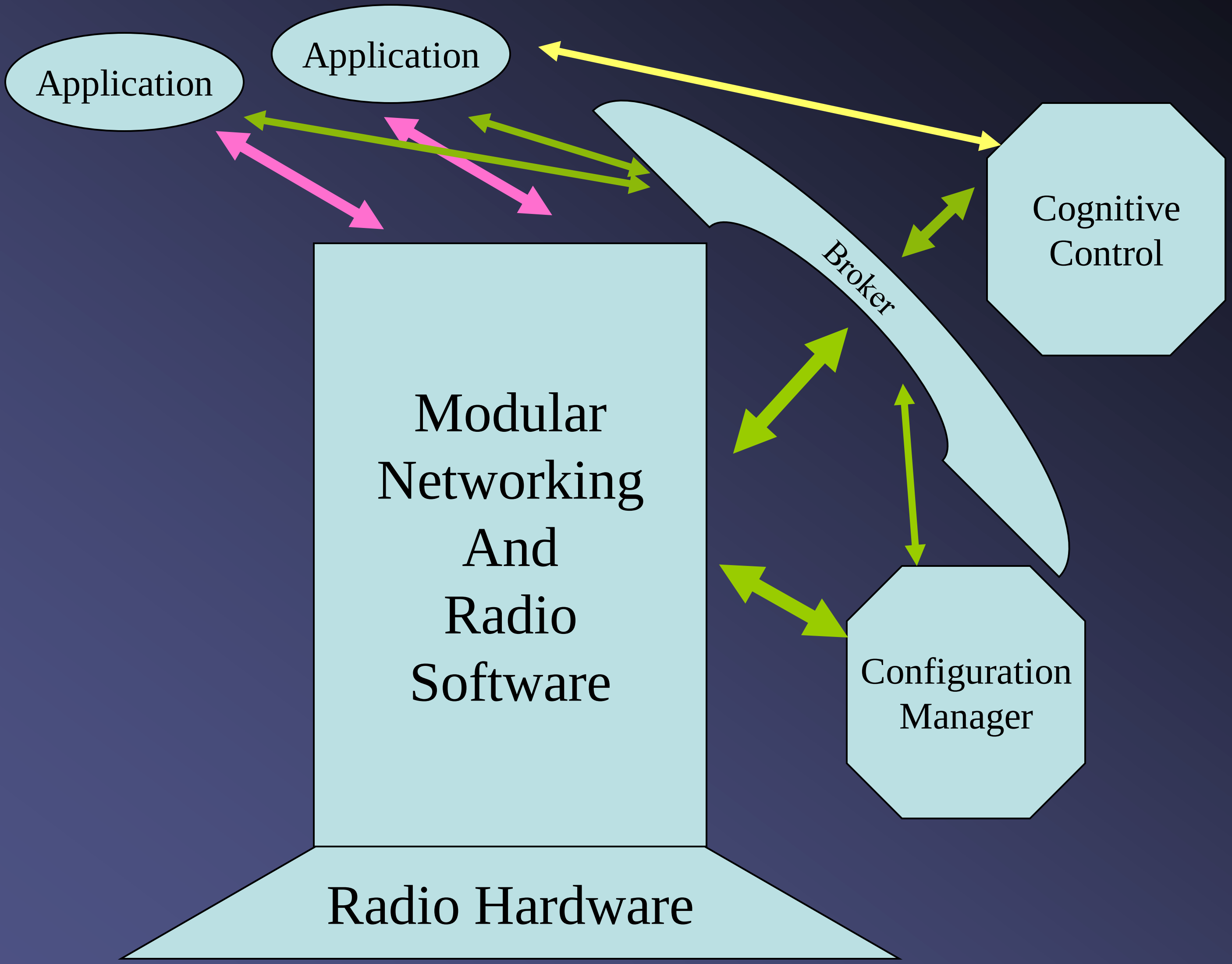
Environment Model

- Signal Strength
  - 12 cart-cart strengths
  - sorted to normalize
    - want to apply learning to similar situations with different cart numbering
- Position
  - seemed like a good idea (“use more information, let neural net sort it out”), but....
  - in testing, seemed more confounding than helpful
- On-line estimate required
  - operation uses environment

Configuration and Adaptation

- Configuration Manager
  - Determines what modules are currently running
  - Tracks what modules exists
  - Manager transitions from one configuration to another
  - Provides basic sanity check before enabling a new configuration
- Broker
  - Changes and monitors the state of active modules
  - Serves as a clearinghouse of information about all the modules in current configuration

ADROIT Big Picture



Managing Cognition

- ADROIT doesn't choose the form
  - Open to just about any form you can imagine
  - Multiple forms on each node, system wide
  - Operate via standard interface (broker)
- Coordination manager
  - Coordinates interactions among radios
  - Chooses local radio's external behavior taking into account needs of other radios in team and in region
  - Manages information sharing (keeps cognitive information exchanges within reasonable limits)

Modelling the Radio

- Need a way to model the radio for cognition
  - A chunk of code (module) is not expressive enough
  - At minimum, cognition needs to know what the chunk of code does
- A basic object model
  - Each module is an object
  - Two implementations of the same functionality are same object type, or inherit characteristics from the same object type
  - Pieces of hardware, etc, also viewed as objects

ADROIT resources

- Troxel et al. "Enabling open-source cognitively-controlled collaboration among software-defined radio nodes." *Computer Networks,* 52(4):898-911, March 2008.
- Troxel et al, "Cognitive Adaptation for Teams in ADROIT," in *IEEE Global Communications Conference*, Nov 2007, Washington, DC. **Invited**.
- Getting the ADROIT Code (Including the Broker)
  - https://acert.ir.bbn.com/
  - checkout instructions
  - GNU Radio changes are in main GNU Radio repository

Learning

- Karen Zita Haigh, Srivatsan Varadarajan, Choon Yik Tang, "Automatic Learning-based MANET Cross-Layer Parameter Configuration," in *IEEE Workshop on Wireless Ad hoc and Sensor Networks (WWASN)*, Lisbon, Portugal 2006.

ADROIT Team

BBN Technologies:

- Greg Troxel (PI), Isidro Castineyra (PM)
- *AI*: Karen Haigh, Talib Hussain
- *Networking*: Steve Boswell, Armando Caro, Alex Colvin, Yarom Gabay, Nick Goffee, Vikas Kawadia, David Lapsley, Janet Leblond, Carl Livadas, Alberto Medina, Joanne Mikkelson, Craig Partridge, Vivek Raghunathan, Ram Ramanathan, Paul Rubel, Cesar Santivanez, Dan Sumorok, Bob Vincent, David Wiggins
- Eric Blossom (GNU Radio consultant)

University of Kansas:

- Gary Minden, Joe Evans

MIT: Robert Morris, Hari Balakrishnan

UCLA: Mani Srivastava