A SOFTWARE DEFINED RADIO MODELING FRAMEWORK FOR ENABLING COGNITIVE APPLICATION

Angelo Sapello (University of Delaware, Newark, DE, USA)
Constantin Serban (Applied Communication Sciences, Piscataway, NJ, USA)
Adarsh Sethi (University of Delaware, Newark, DE, USA)
C. Jason Chiang (Applied Communication Sciences, Piscataway, NJ, USA)
Kimberly Moeltner (CERDEC, Aberdeen, MD, USA)

Outline

- Overview
- Modeling goals and reference device
- Modeling framework
  - MAC module
  - Coder module
  - PHY module
  - Wireless channel
  - Channel end-point
  - Error module
  - Reconfiguration module
- Results

Overview

- Why do we need a simulation framework?
  - Testing in situations not easily produced in real-world
  - Testing scaling behavior beyond real-world resources
  - Expense of real-world spectrum expenses
  - Potentially faster than real-time testing
- Challenges of SDR simulation
  - Large comm delays between software-defined link layer and radio front end (hardware)
  - SDRs are highly configurable
  - Importance of SITL

Overview

- Modeling Challenges
  - Modeling delayed continuous information (CS)
  - Long term simulation evolution requires multiple modeling techniques
    - SITL
    - Obstacle modeling
    - Interference modeling
    - Multichannel modeling
    - Dynamic parameter setting

Overview

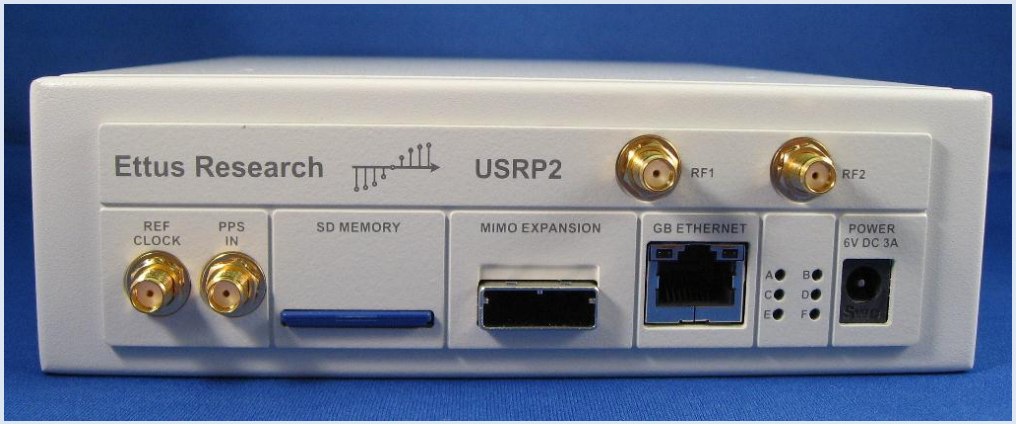
- Model design
  - NS-2: free, open-source, widely used
  - Modeling GNU Radio
  - MAC layer
  - PHY layer
  - Wireless channel
  - Channel end-point

Modeling goals

- Evaluate impact of SDR functions under realistic environment conditions
- Support development and evaluation of runtime adaptation strategies
- Support pure simulation deployment
- Support hybrid SITL deployments

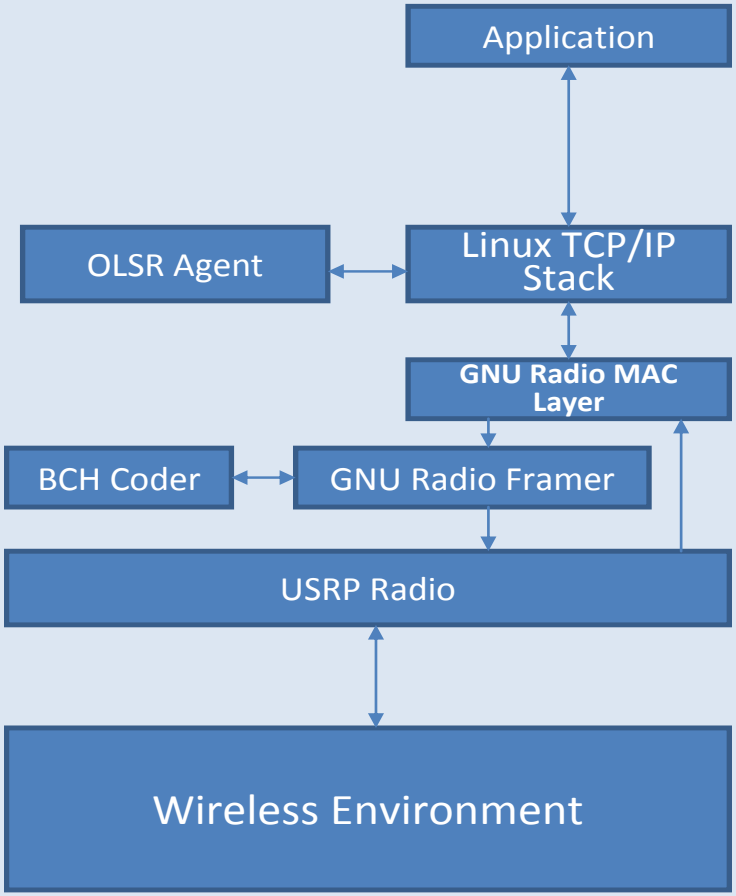
Reference device

- GNU Radio: popular open-source SDR on GPP toolkit
- Packet radio for tactical ad-hoc networks
- USRP II front-end

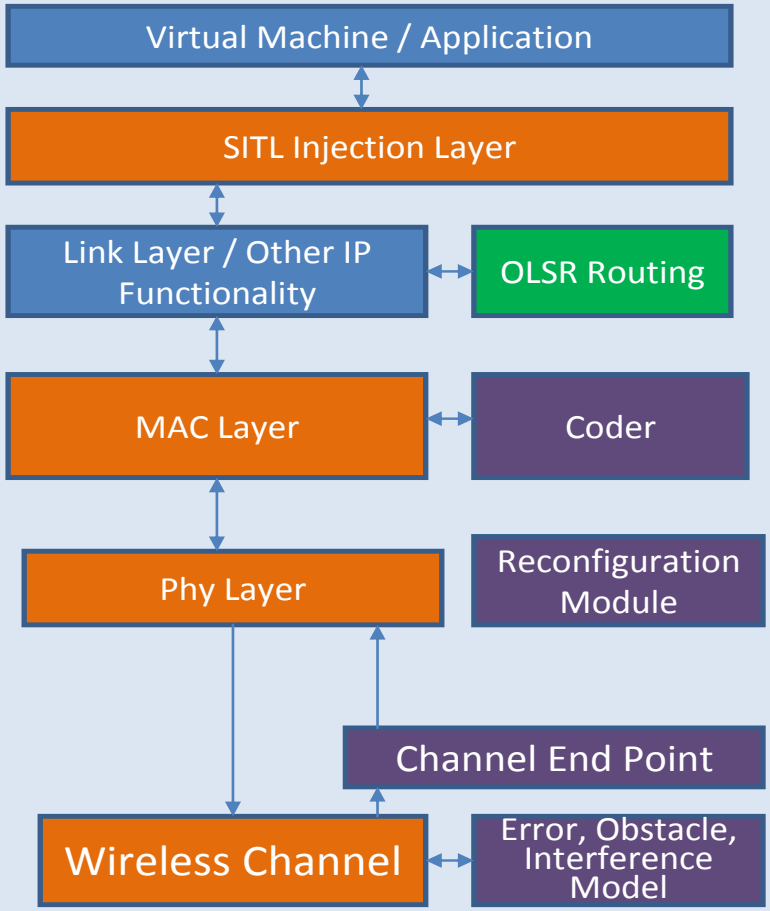


Model Framework

GNU Radio Components



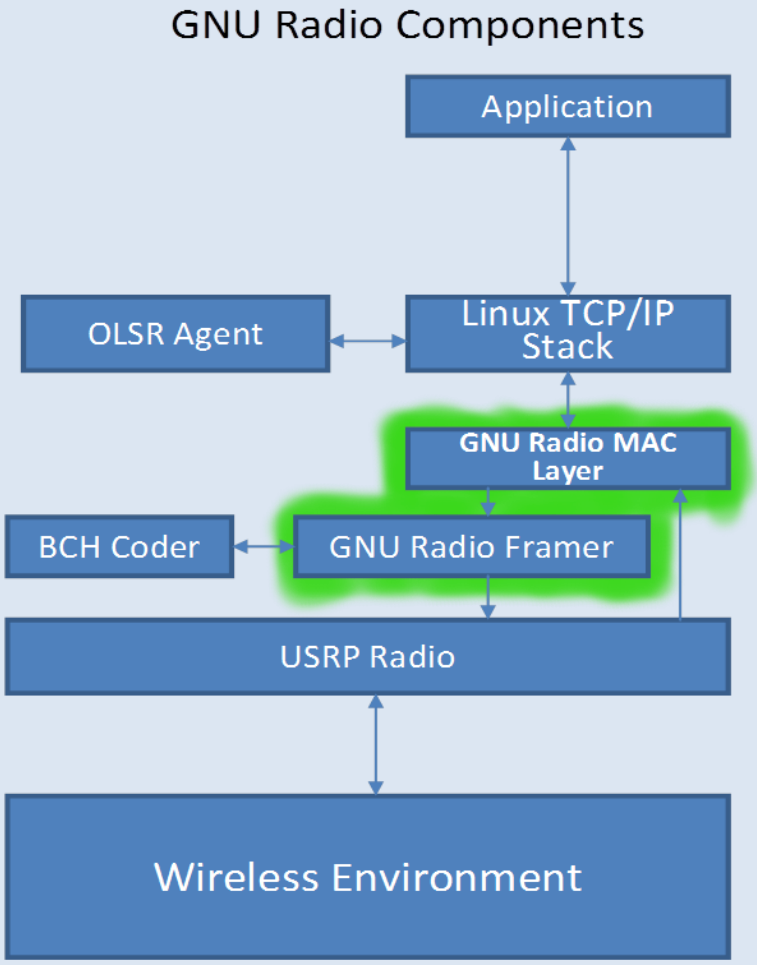
SDR Model Framework

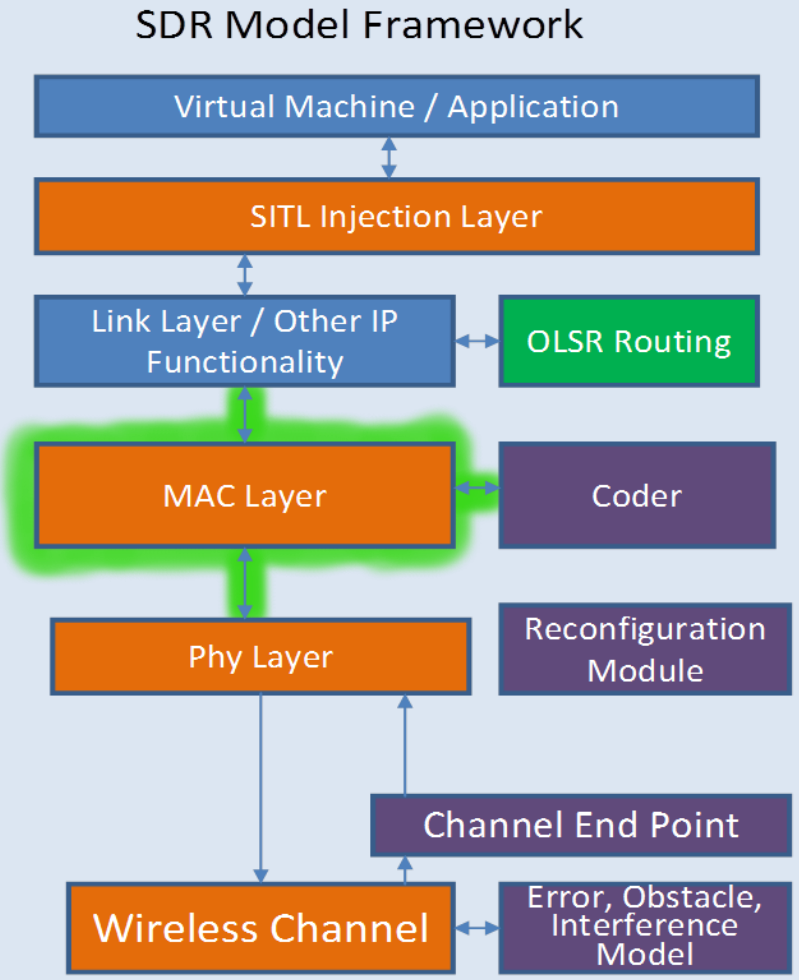


UNIVERSITY OF DELAWARE 1743

APPLIED COMMUNICATION SCIENCES

MAC Module


GNU Radio Components
Application
OLSR Agent
Linux TCP/IP Stack
GNU Radio MAC Layer
BCH Coder
GNU Radio Framer
USRP Radio
Wireless Environment


SDR Model Framework
Virtual Machine / Application
SITL Injection Layer
Link Layer / Other IP Functionality
OLSR Routing
MAC Layer
Coder
Phy Layer
Reconfiguration Module
Channel End Point
Wireless Channel
Error, Obstacle, Interference Model

MAC Module

- Simulates MAC of SDR
- Transmitting
  - Packets received from interface queue
  - Decides whether to transmit, reschedule or drop possibly asking PHY for carrier signal strength
  - Generates frame from packet
  - Passes frame to Coder module to add ECC, interleaving and channel coding
  - Sets transmission frequency of PHY
  - Passes packet to PHY Module
  - Informs interface queue that its ready for next packet

MAC Module

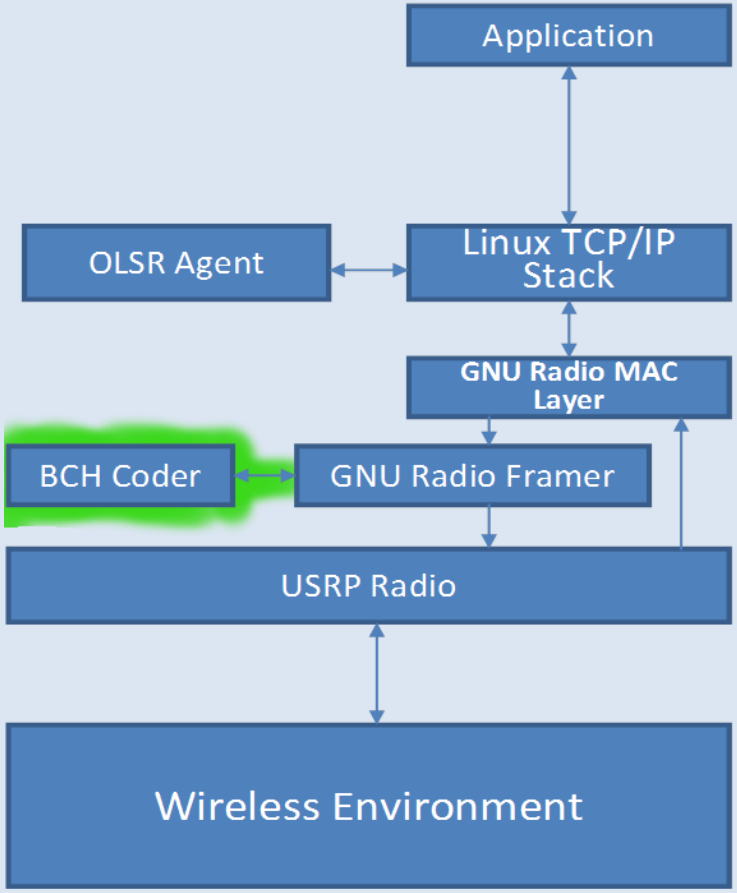
- Receiving
  - Receives coded frame from PHY layer
  - Decides if it can receive the packet (multichannel function) or needs to drop
  - Schedules iterative decoding through the Coder module
  - Checks access code and drops if invalid
  - Checks that it is the intended receiver

MAC Module

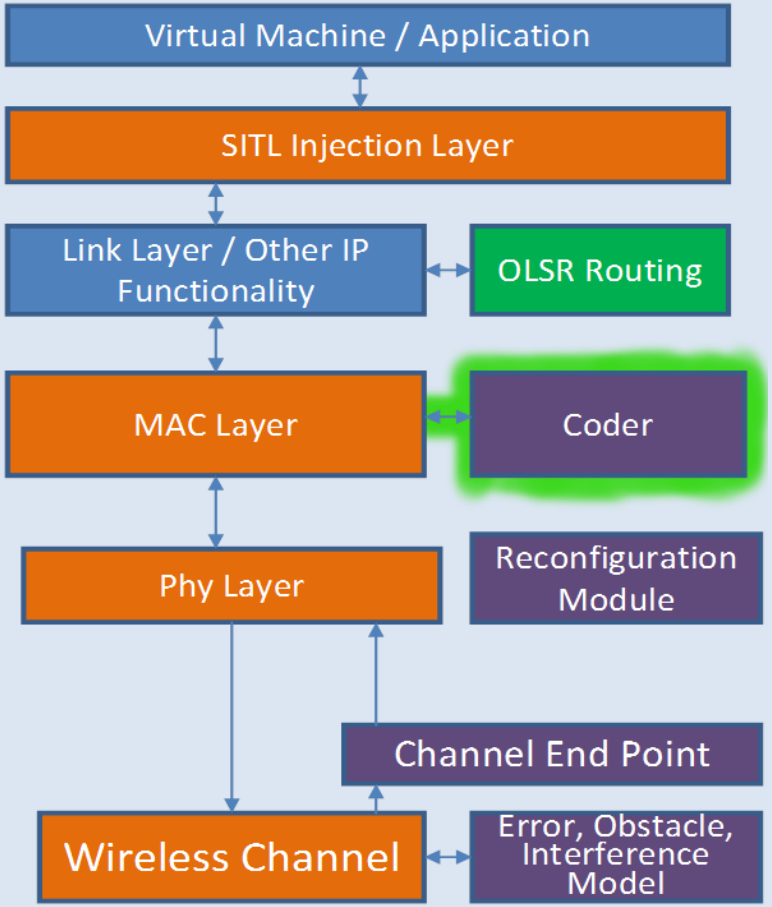
- SDR responsibilities
  - Computation and buffer delays due to signal processing
  - Computation and buffer delays due to carrier sense
- Configurable parameters
  - CSMA (back offs, yield times, CS threshold)
  - Number of channels
  - Data rates

Coder Module

GNU Radio Components



SDR Model Framework



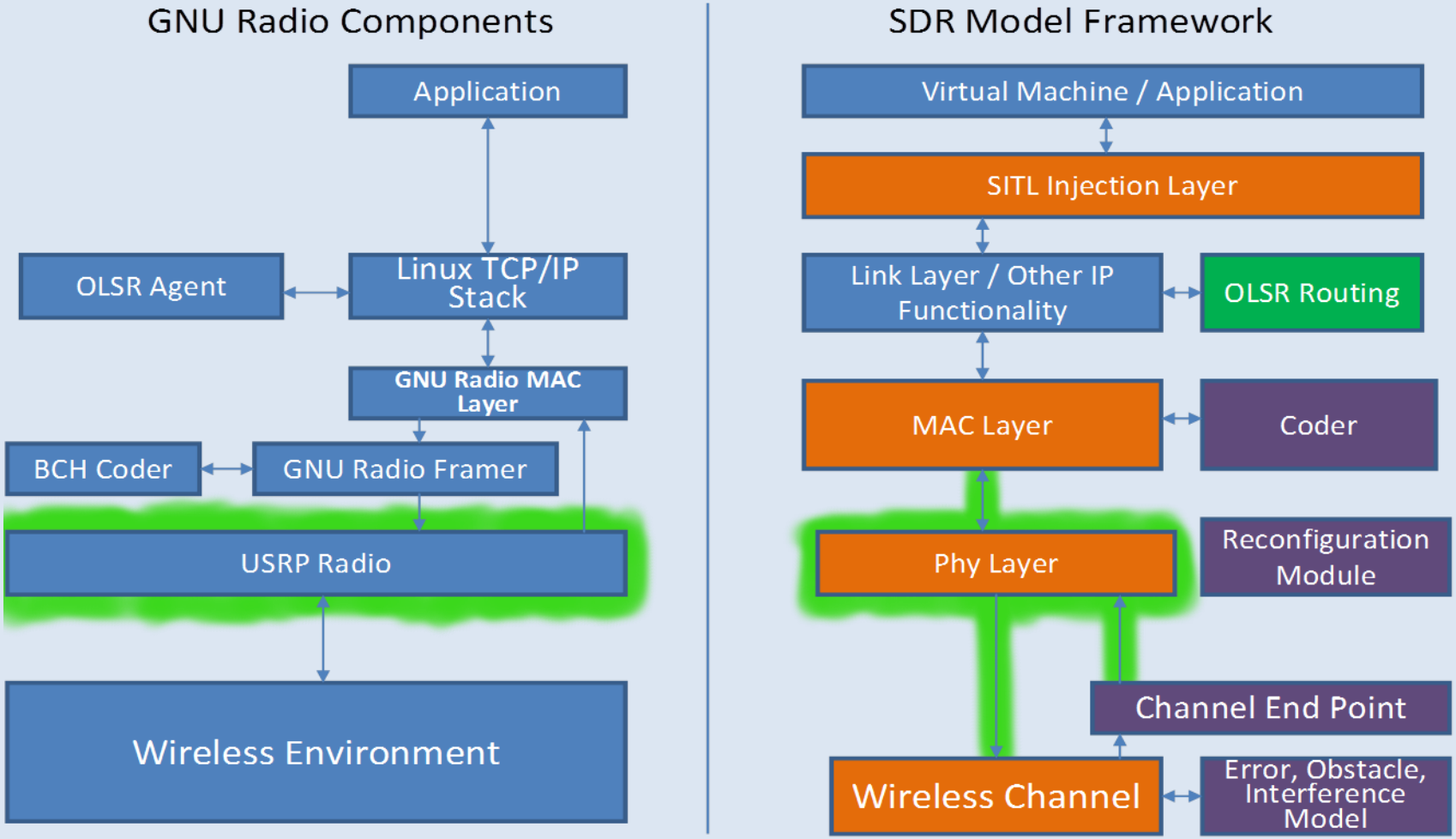
Coder Module

- Simulates encoding and decoding of packets with ECC
- Special dummy coder simulates lack of channel coding
- Highly configurable – coder number (code identification), block length, coding rate, coding data
- Coding data
  - curve representing input error rate to output error rate
  - linear interpolation between points

Coder Module

- Encoding
  - Stores coder type in meta-header
  - Transforms size of packet to reflect coding rate and block size
- Decoding
  - Checks proper coder type (assumes failure with wrong decoder)
  - Restores original packet size
  - Transforms the input error rate from the error meta-header to an output error rate
  - Selects output errors based on Bernoulli distribution
  - Marks packet as error-free, corrected or uncorrectable
- Models computation and buffering delays

PHY Module


GNU Radio Components
Application
OLSR Agent
Linux TCP/IP Stack
GNU Radio MAC Layer
BCH Coder
GNU Radio Framer
USRP Radio
Wireless Environment
SDR Model Framework
Virtual Machine / Application
SITL Injection Layer
Link Layer / Other IP Functionality
OLSR Routing
MAC Layer
Coder
Phy Layer
Reconfiguration Module
Channel End Point
Wireless Channel
Error, Obstacle, Interference Model

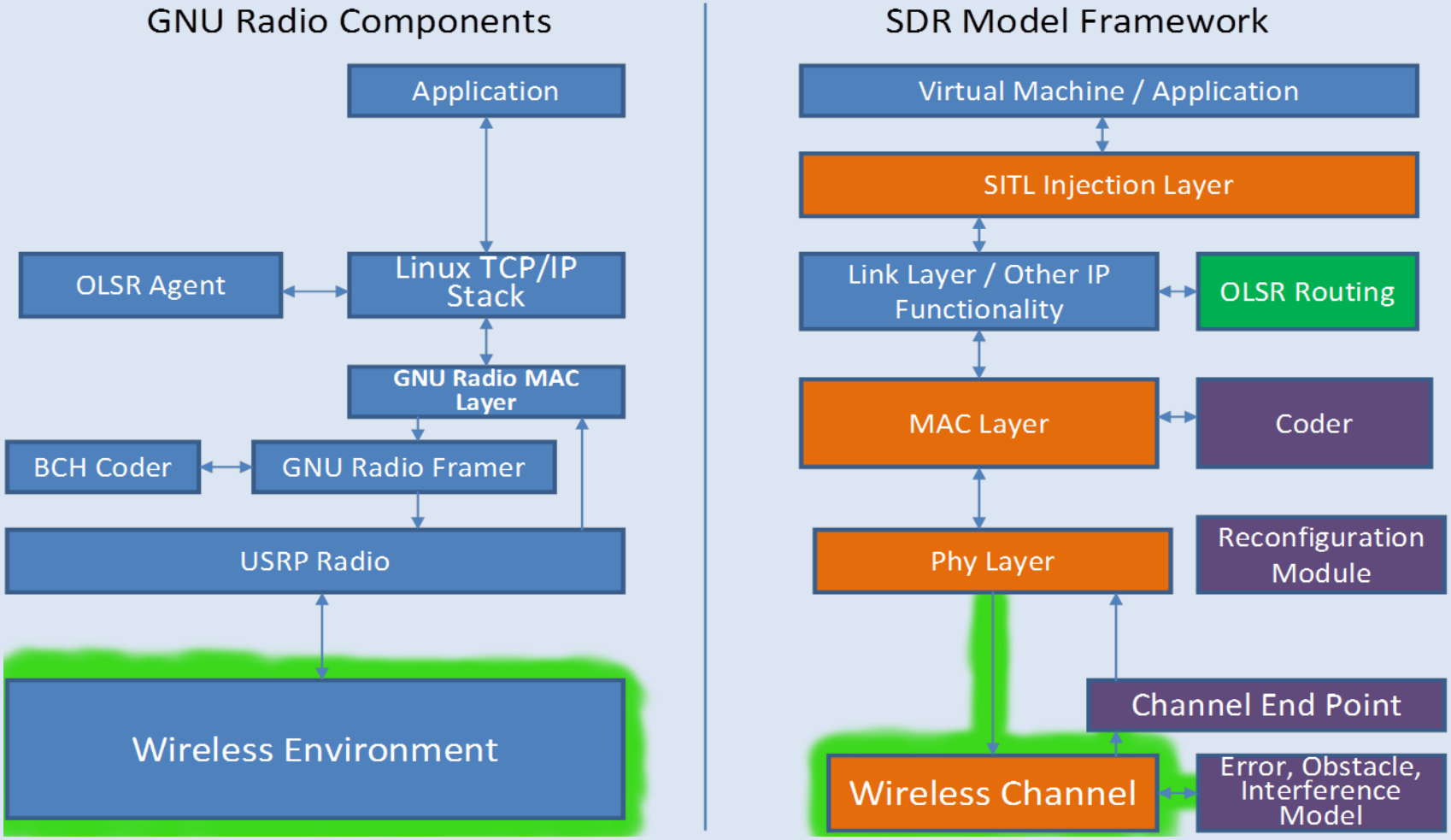
PHY Module

- Simulates the behavior of the radio front end (hardware)
- Very thin module (similar to actual SDR)
- Transmitting
  - Receives packet (frame) from MAC module after a comm delay
  - Requests the MultiChannel object for the Tx frequency and channel width
  - Forwards the packet to the channel

PHY Module

- Receiving
  - Receives packet(frame) from Channel End-Point
  - Adds packet to the received packet list (part of signal strength computation) with a comm delay
  - Forwards packet to the MAC module after a comm delay
- Configurable parameters: bits per symbol, samples per symbol, sample frequency, comm delay, tx frequency, radio system loss, tx power

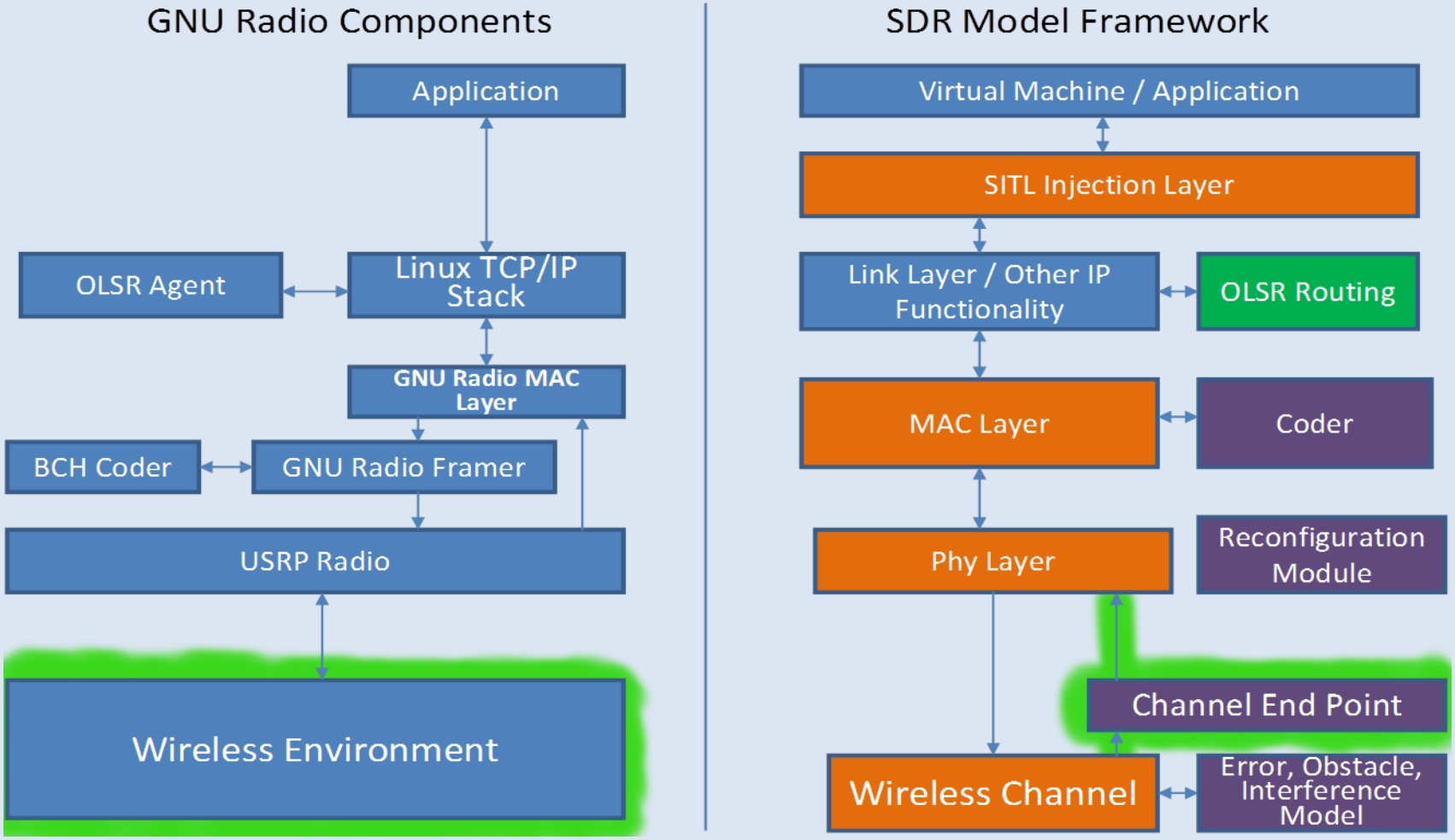
Wireless Channel



Wireless Channel

- Simulate behavior of the wireless channel
- A global instance
  - Maintains list of channels being simulated
  - Provides interface to start and stop listening
- Each channel has a frequency, width and noise floor
- Records sender information (Tx power, location) then duplicates packet to each listening PHY via a Channel End-Point with a propagation delay
- Optimized with a distance table to only forward within a specified range of a sender

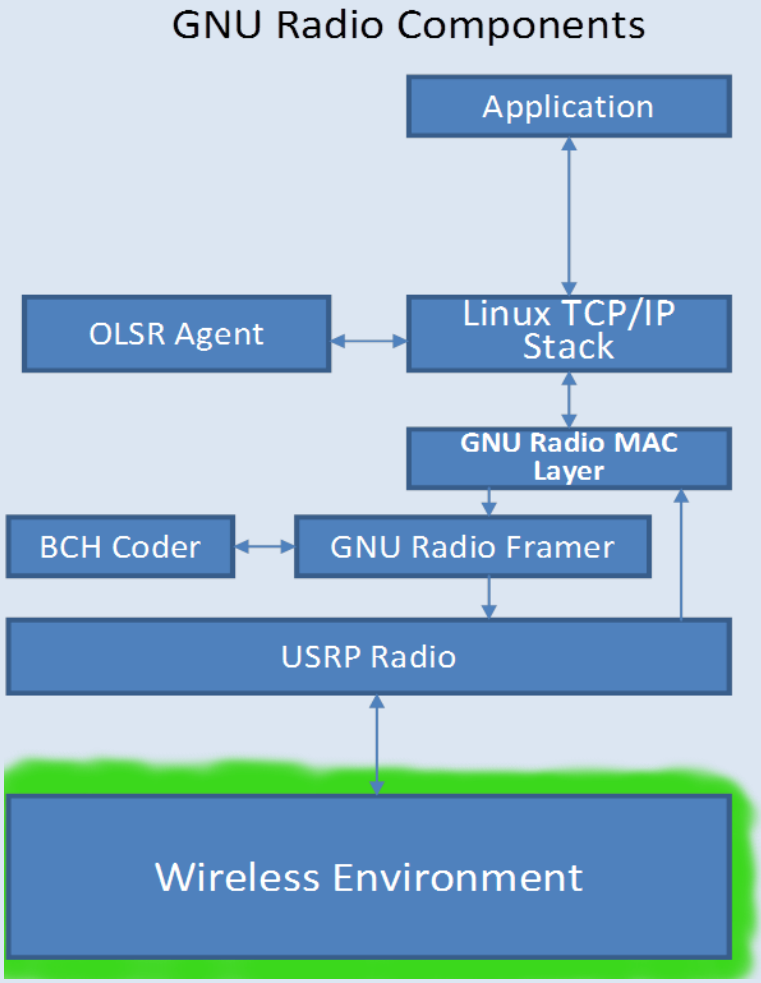
Channel End-Point

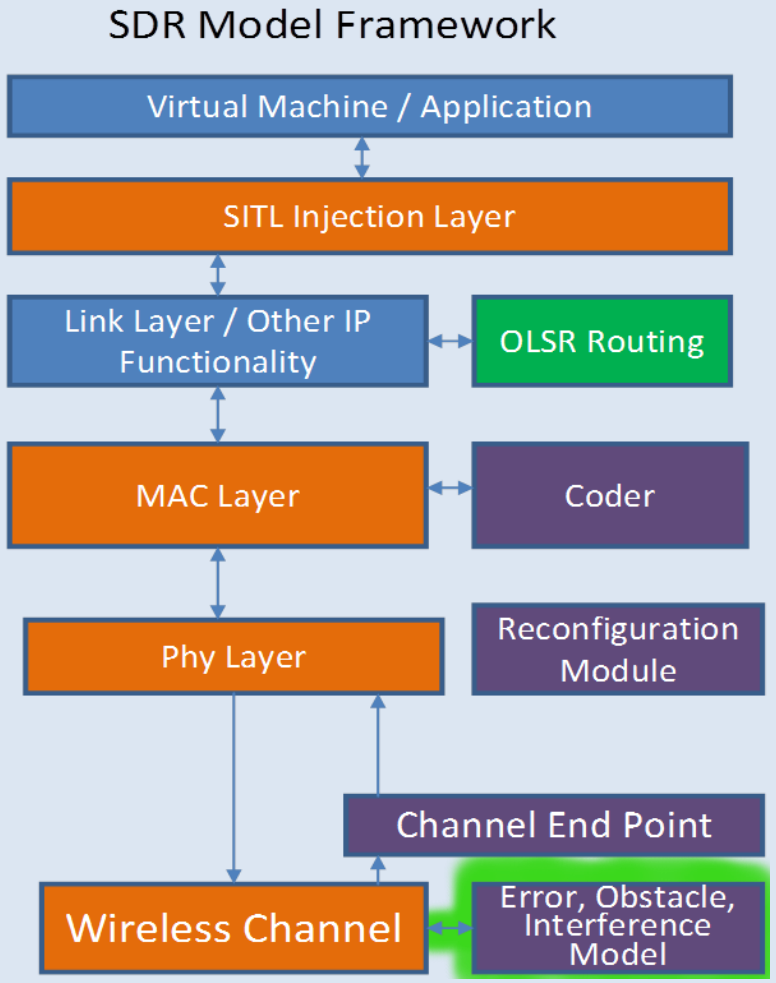


Channel End-Point

- Simulates the channel at a specific location
- Necessary since the channel exists everywhere but certain calculations are location specific
- Maintains a received packet list to compute instantaneous signal strength for the Error module
- Notifies the Error module to update error info on all incoming packets whenever a packet starts or stops transmission
- Forwards packets to the PHY module

Error Module

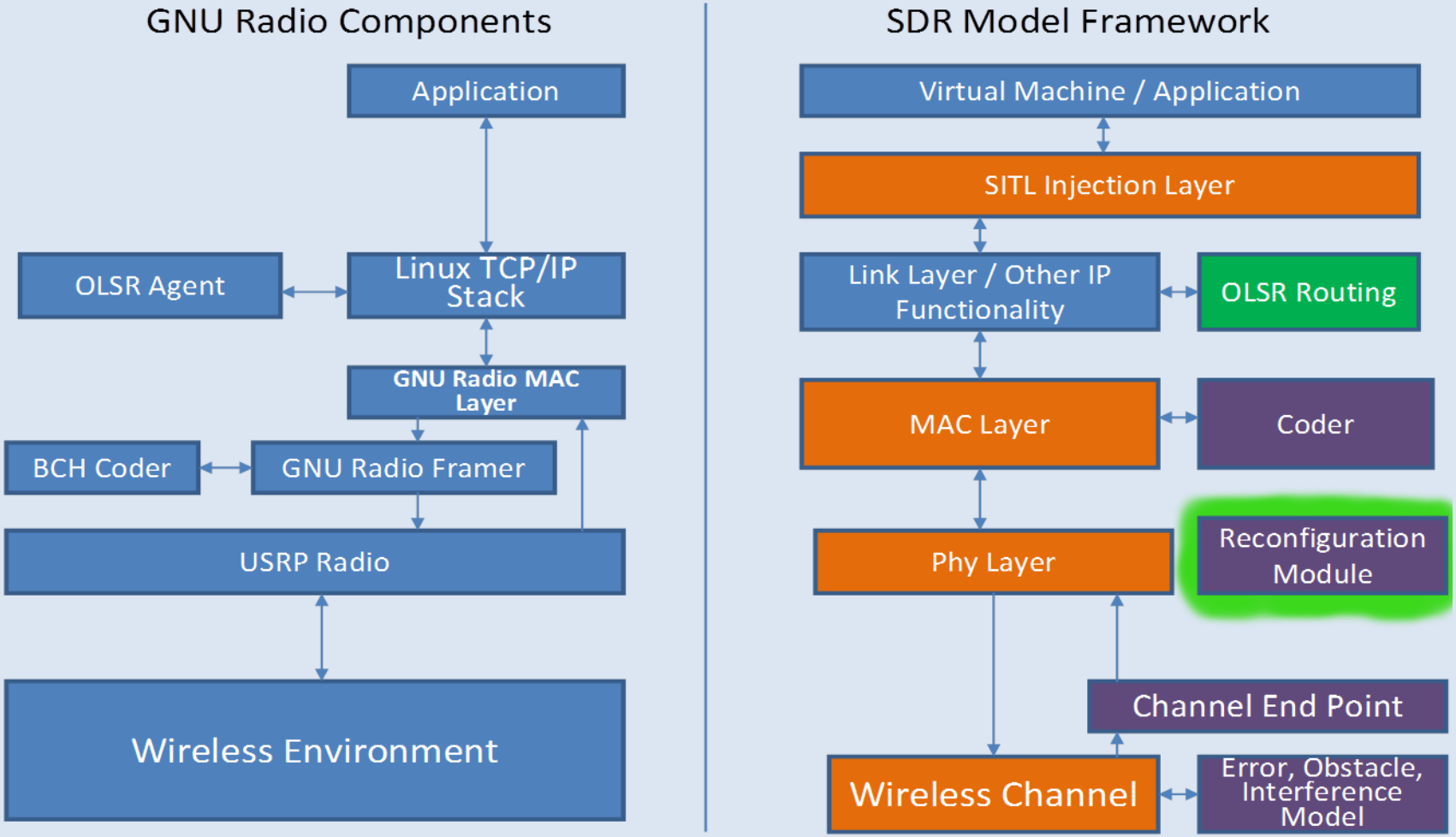




Error Module

- Simulates the errors introduced due to channel conditions
- Models errors by creating a list of “points of interest” (POI) for each packet
- At each POI the channel signal strength is used to compute the SINR for the packet for the upcoming interval and the number of errors is recorded for the previous interval using a Bernoulli distribution with mean:
  - Expected errors = symbols * 0.5 * erfc(sqrt(sinr))
- This models a BSC with hard decision decoding

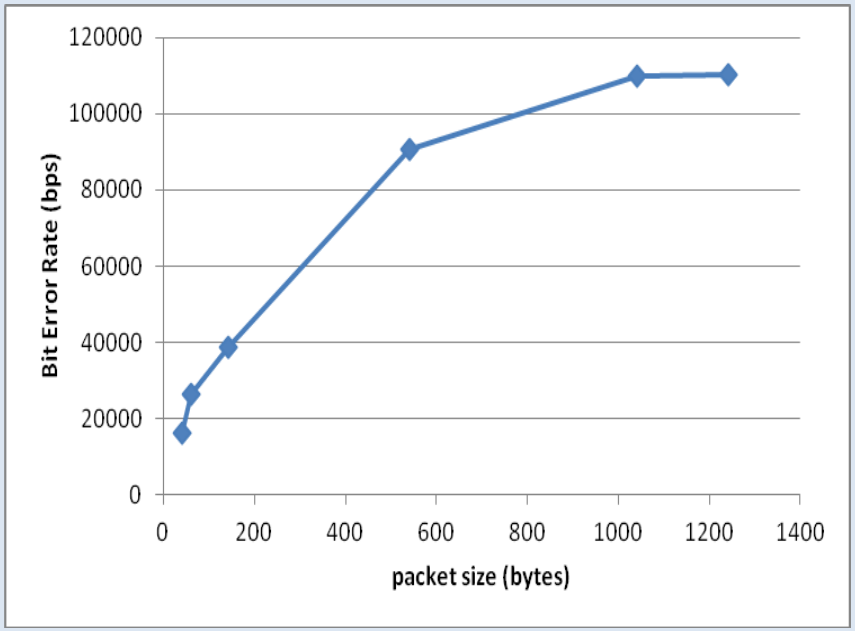
Reconfiguration Module

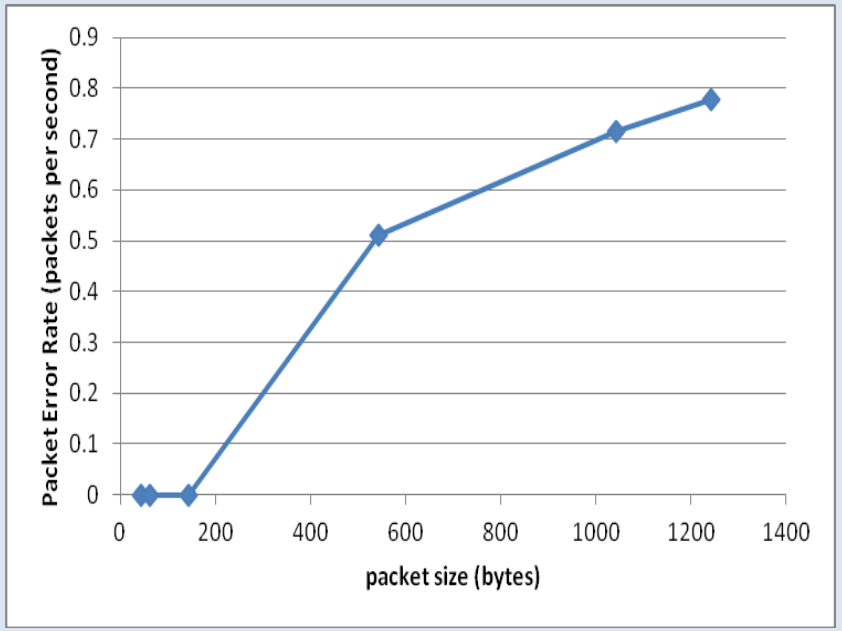


Reconfiguration Module

- Allows an external application to reconfigure simulation parameters for a simulated node and monitor simulation statistics
- Application can be real code on a VM using SITL technology or simulated code in the simulation
- Interacts with all other modules
- Interfaces
  - Applications issue TCL commands to NS-2's TCL interpreter
  - SNMP module with special VM SNMP connector
- Enables testing and development of cognitive adaptation algorithms

Results – Parameter Study

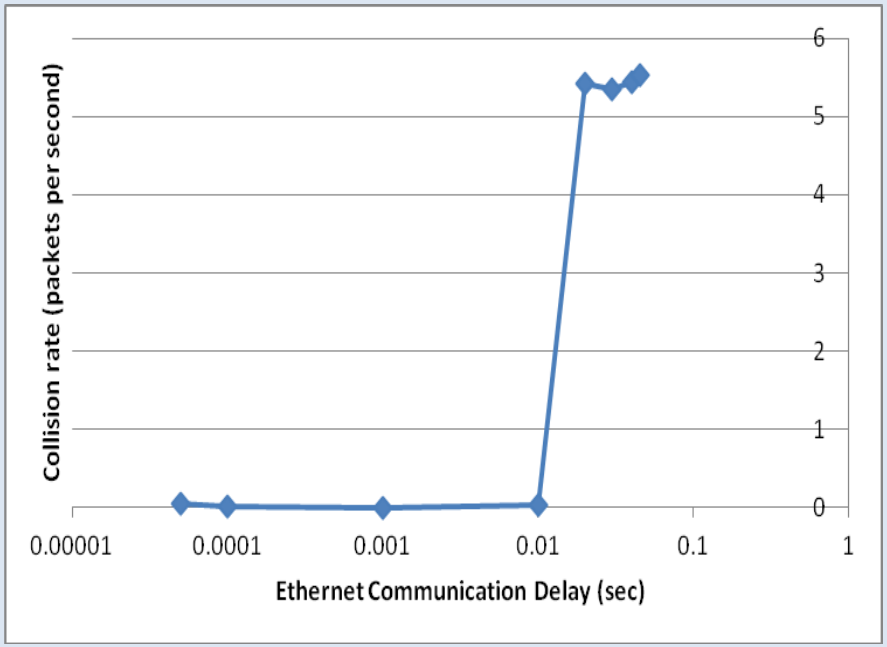


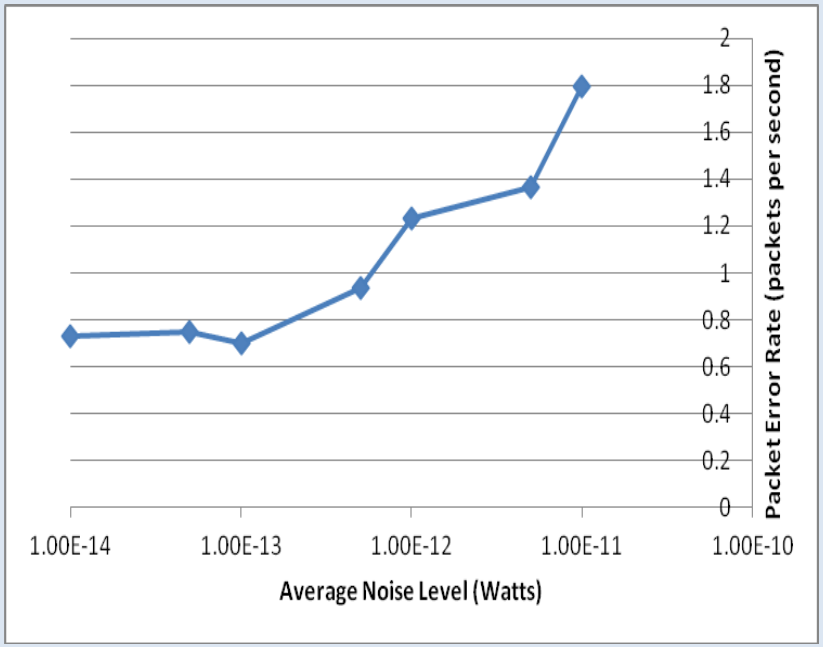


UNIVERSITY OF DELAWARE 1743

APPLIED COMMUNICATION SCIENCES

Results – Parameter Study

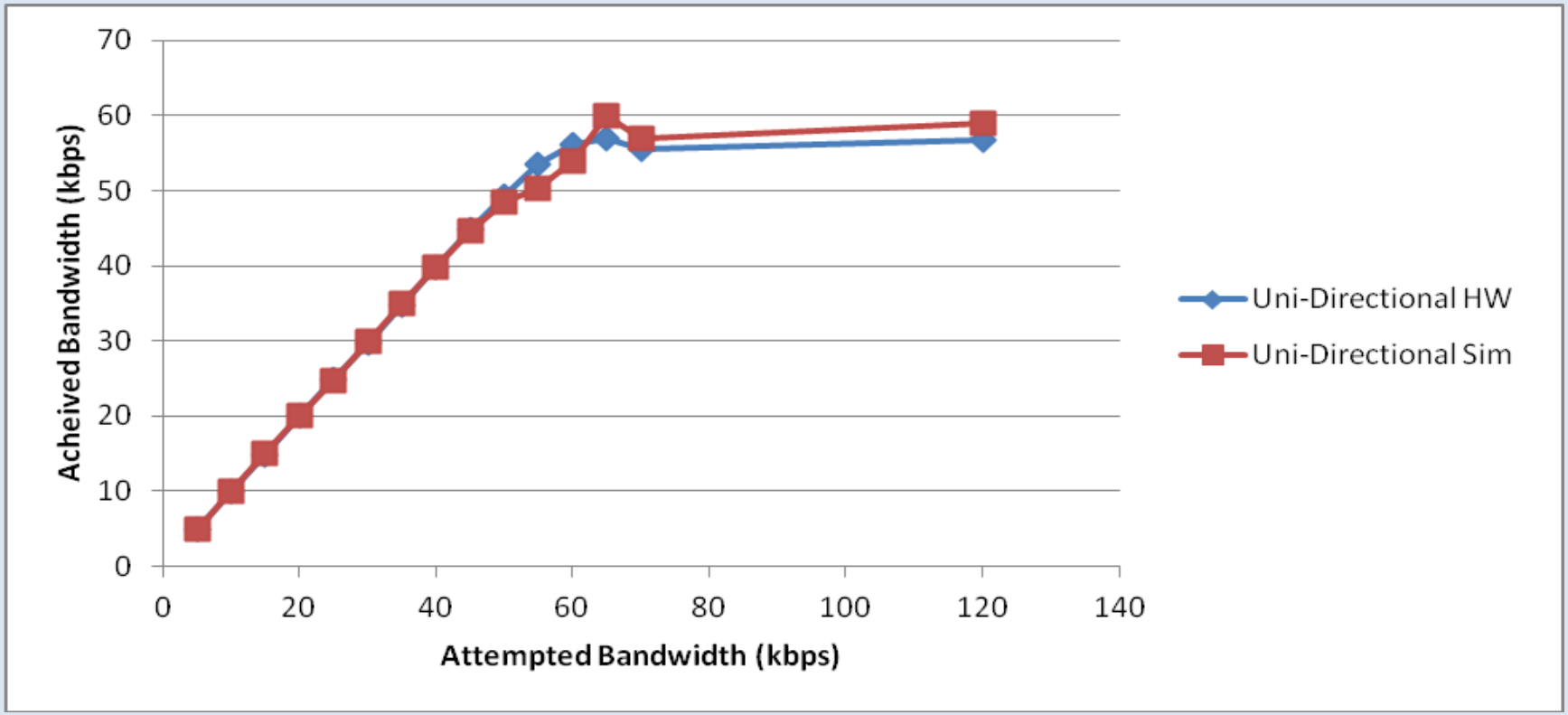




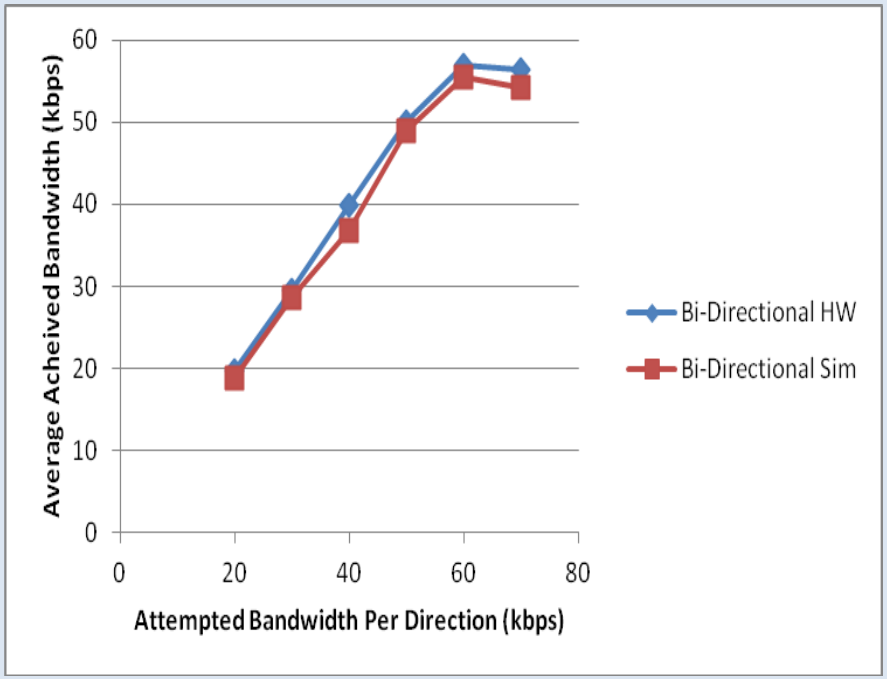
UNIVERSITY OF DELAWARE 1743

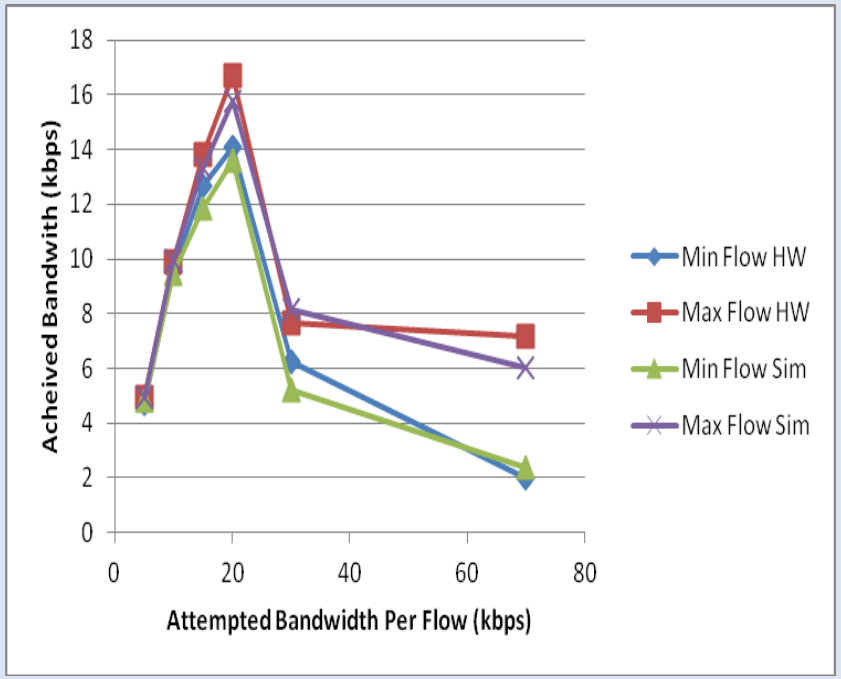
APPLIED COMMUNICATION SCIENCES

Results - Fidelity

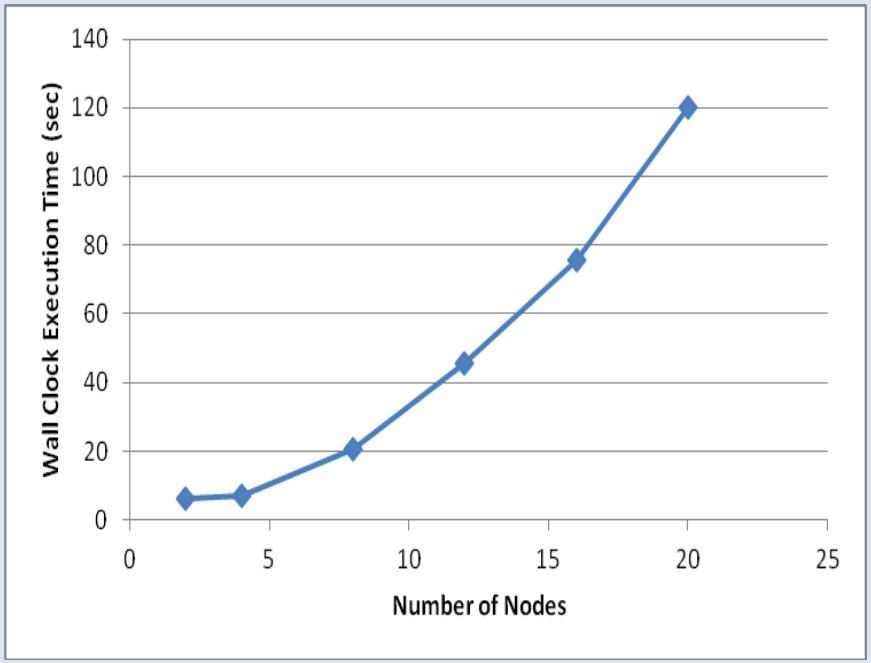


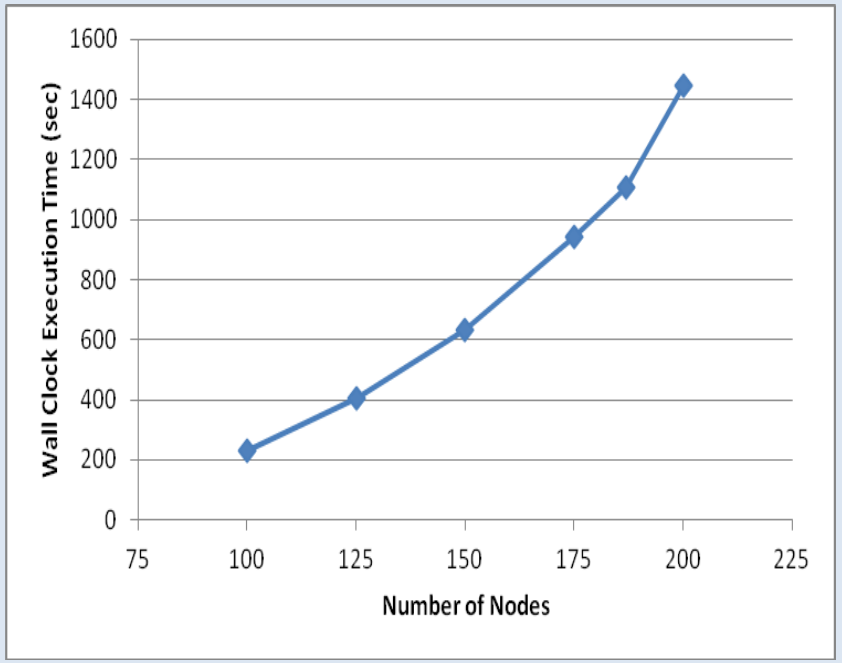
Results - Fidelity





Results - Scalability





UNIVERSITY OF DELAWARE 1743

APPLIED COMMUNICATION SCIENCES

Questions?

sapello@cis.udel.edu