A Unified, Scalable, and Extensible Physical Layer Design for INET
Motivation

- Incompatibility
- Scalability
- Extensibility
- Maintainability
- Duplicated functionality
- Missing functionality
- Parallel hardware support
Goals

• Unify existing INET and MiXiM physical layer functionality into a new physical layer model

• Make the new model scalable in terms of level of detail with respect to signal representation and signal processing

• Make the new model extensible with alternative implementations for meaningful sub-components

• Port existing MiXiM and INET higher layer functionality to use the new physical layer

• Support optimistic parallel execution on multiple CPUs and on highly parallel GPUs
Scaling the Level of Detail

- Performance versus Accuracy
- Statistical Model Emulation
- Flat Architecture Layered
- Scalar Data Dimensional
- Some Message All
Extensible Modules

- Split up radio and medium modules into submodules
- Avoid mixing parameters of different implementations
Layered Data and Processing Architecture

- Further split up transmitter and receiver modules
Domain Specific Signal Representations

- Packet domain
  - packet, packet error rate, packet error
- Bit domain
  - bit length, actual bits, bit rate, FEC, CRC, bit error rate, bit error count, erroneous bits
- Symbol domain
  - number of symbols, symbol rate, actual symbols, modulation, symbol error rate, symbol error count, erroneous symbols
- Sample domain
  - number of samples, sampling rate, actual samples
- Analog domain
  - space-time coordinates, ranges, scalar, dimensional, RSSI, SNIR
Analog Domain Signal Representations

- Range-based
- Communication range
- Interference range
- Detection range
- Single dimensional
- Scalar
- Multi dimensional

- Power
- Bandwidth
- Carrier frequency

- Start
- End

- Time (t[s])
- Power (P[W])
- Frequency (f[Hz])
Message Processing

Radio A

mac frame

transmitPacket()

t[s]

Medium

parallel computation

radio frame

isSynchronizationAttempted()

isReceptionAttempted()

receivePacket()

t[s]

Radio B

mac frame

timer

receivePacket()

t[s]
Optimizing Message Sends

- Range filter
- Radio mode filter
- Listening mode filter
- MAC address filter
Parallel Execution

- Parallelization opportunity: computing receptions
  - they dominate performance
  - they are independent of each other
- Parallel hardware: multi-core, vector instructions, GPU
- Optimistic parallel computing in background threads
- Minimize blocking of main simulation thread
- Efficient only if upcoming receptions are mostly cached
- Discard cached results upon changes on the medium
- Optionally compute arrivals on GPU in main thread
Physical Environment

- 3D geometry
- Physical properties
- Physical objects
- Materials
- Graphical properties
- Initialized from XML
- Efficient object cache
- Used by obstacle loss
Visualization

- Physical objects
- Movement trajectory
- Ongoing transmissions
- Successful receptions
- Obstacle intersections
- Reflection normal vectors
Notable Changes in C++ Source Code

- Extensible classes and data structures
  - polymorphism using subclassing and virtual functions
- Physical quantities have compile-time verified SI units
  - base units, prefixes, operators, arithmetic expressions
  - m, s, mps, W, mW, Hz, MHz, etc.
- Parallel execution needs
  - immutable data structures
  - purely functional code that is free of side effects
Implemented Functionality: Radio

- Radio modes
  - off, sleep, receiver, transmitter, transceiver, switching
- Antenna
  - isotropic, constant gain, dipole, interpolating
- Transceiver
  - range-based, flat statistical scalar and dimensional
- Power consumer
  - based on radio mode, transmitter state and receiver state
- Ported standards
  - IEEE 802.11 from INET
  - IEEE 802.15.4a UWBIR from MiXiM
Implemented Functionality: Medium

- Propagation
  - constant time, constant speed
- Path loss
  - free space, breakpoint, log normal, two-ray ground, Nakagami, Rayleigh, SUI, UWB stochastic
- Obstacle loss
  - straight path based dielectric and reflection loss
- Background noise
  - isotropic
- Neighbor cache
  - neighbor list, spatial grid, quad tree
Implemented Functionality: Geometry

- **3D sets**
  - line segment, axis aligned box, polygon, plane
- **3D shapes**
  - sphere, cuboid, convex prism, convex polyhedron
- **3D orientation**
  - Euler angles, rotation matrix, quaternion
- **Caches**
  - spatial grid, quad tree, BVH tree
- **Algorithms**
  - bounding box, faces and normal vectors, intersection, visible faces, convex hull, 2D projection
Other Implemented Functionality

- **Mobility**
  - stationary orientation, constant speed rotation
- **Power source**
  - ideal power source, voltage regulated battery
- **Physical environment**
  - environment, object, material
- **Physical object cache**
  - spatial grid, BVH tree
Higher Layer Functionality

- Link layers
  - ported IEEE 802.11 from INET
  - ported IEEE 802.15.4a from MiXiM
  - ported CMSA, BMAC, LMAC from MiXiM
  - added Ideal mac

- Network layers
  - ported Flood, Probabilistic Broadcast, Wireless Sensor Network from MiXiM
  - untouched INET network layers
Functionality under Development

- Radio
  - layered bit precise, GNU software-defined radio
- Medium
  - multi-threaded, GPU based scalar, acoustic wireless, wired
- Propagation
  - GPU based, receiver movement approximating
- Path loss
  - Weibull, Jakes from MiXiM
- Stochastic obstacle loss
- Multipath fading
  - UWB stochastic from MiXiM, ray tracing for reflections
Tests and Examples

• All existing wireless fingerprint tests pass in INET
• New tests for reception and interference corner cases
• New examples for
  – various MAC and physical layer combinations
  – indoor and outdoor scenarios with obstacles
  – scaling for parallel execution
  – neighbor cache comparison
  – object cache comparison
Questions and Answers

Thank you for your attention!