Utilization of Modeling and Simulation for Networked Waveform Characterization and Validation

Scott Rediger
Rockwell Collins
ssredige@rockwellcollins.com
(319) 295-1723

Rockwell Collins

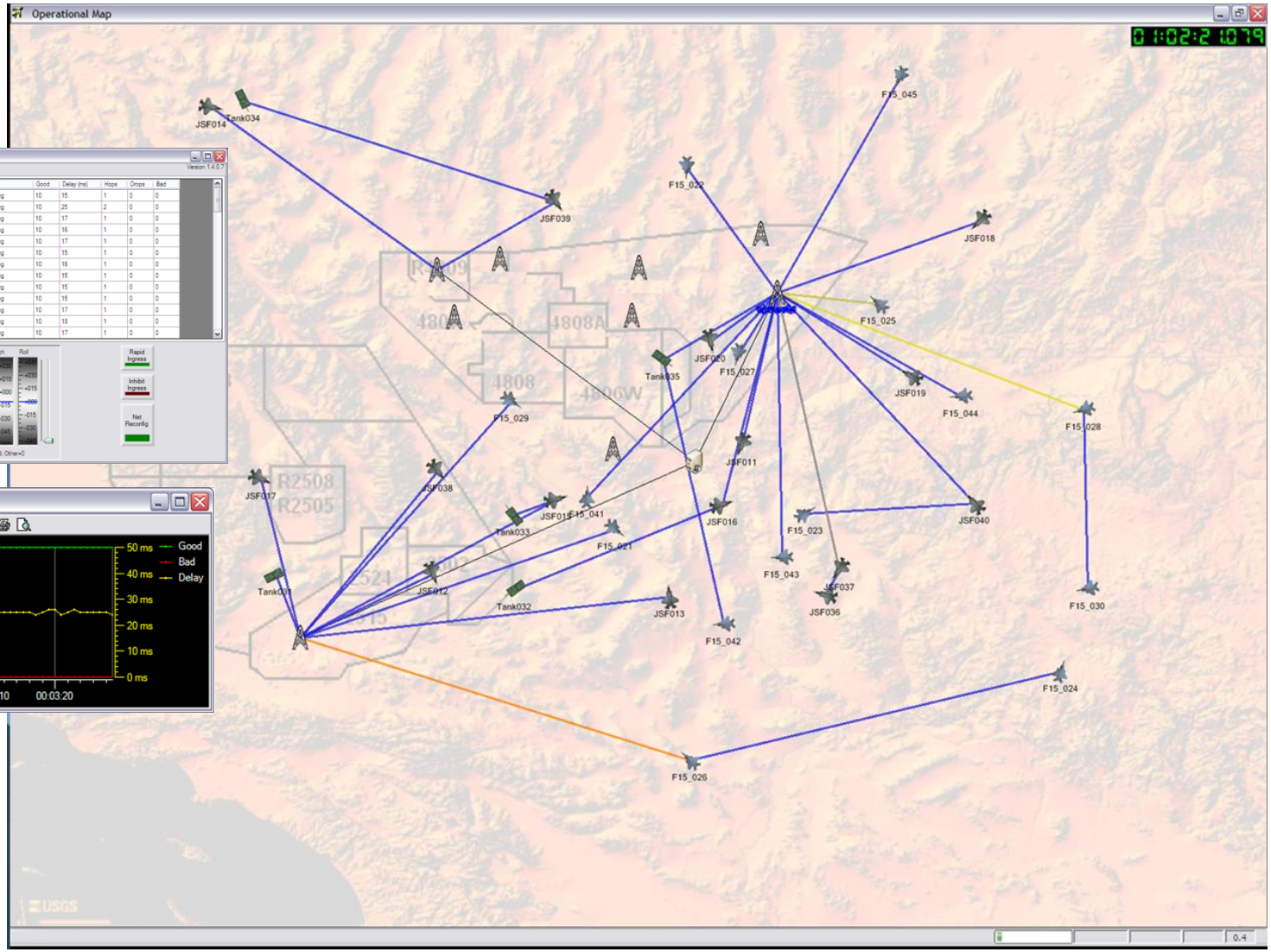
Agenda

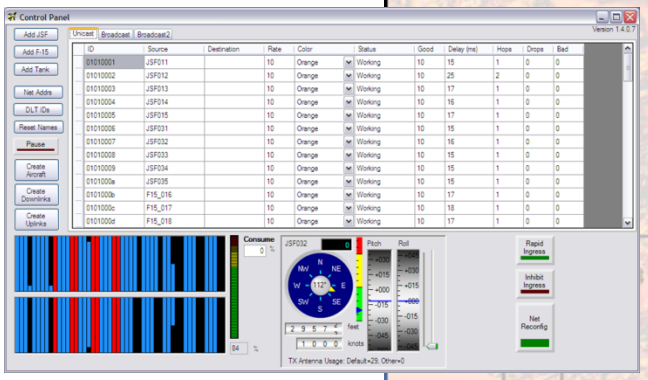
- Discuss Modeling and Simulation used for Networked Waveform Development and Validation
  - What is a Networked Waveform?
  - Why is simulation required for Networked Waveforms
  - How Modeling and Simulation can be applied and utilized through the entire product lifecycle
  - Prerequisites for using Simulation
  - Examples of lessons learned from Networked Waveform Simulation

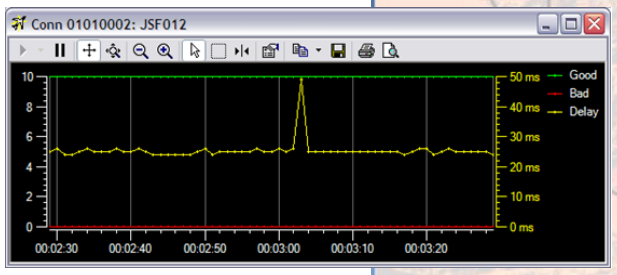
What is a Networked Waveform?

- Sometimes referred to as Mesh Network or Mobile Ad-Hoc Network
- Self-configuring network of nodes connected via wireless data links
  - Each node dynamically adapts to evolving network topologies
    - Network protocols ensure that all nodes are kept abreast of topology updates
    - Data can successfully route through the network with varying numbers of hops depending on the topology
- Nodes are free to physically move about in any direction
  - Nodes can be on land, sea, or air
- Network topology changes over time based on:
  - Each node's physical location
  - Vehicle/Aircraft dynamics
  - Node configuration changes
  - Environmental effects

Example Network Topology

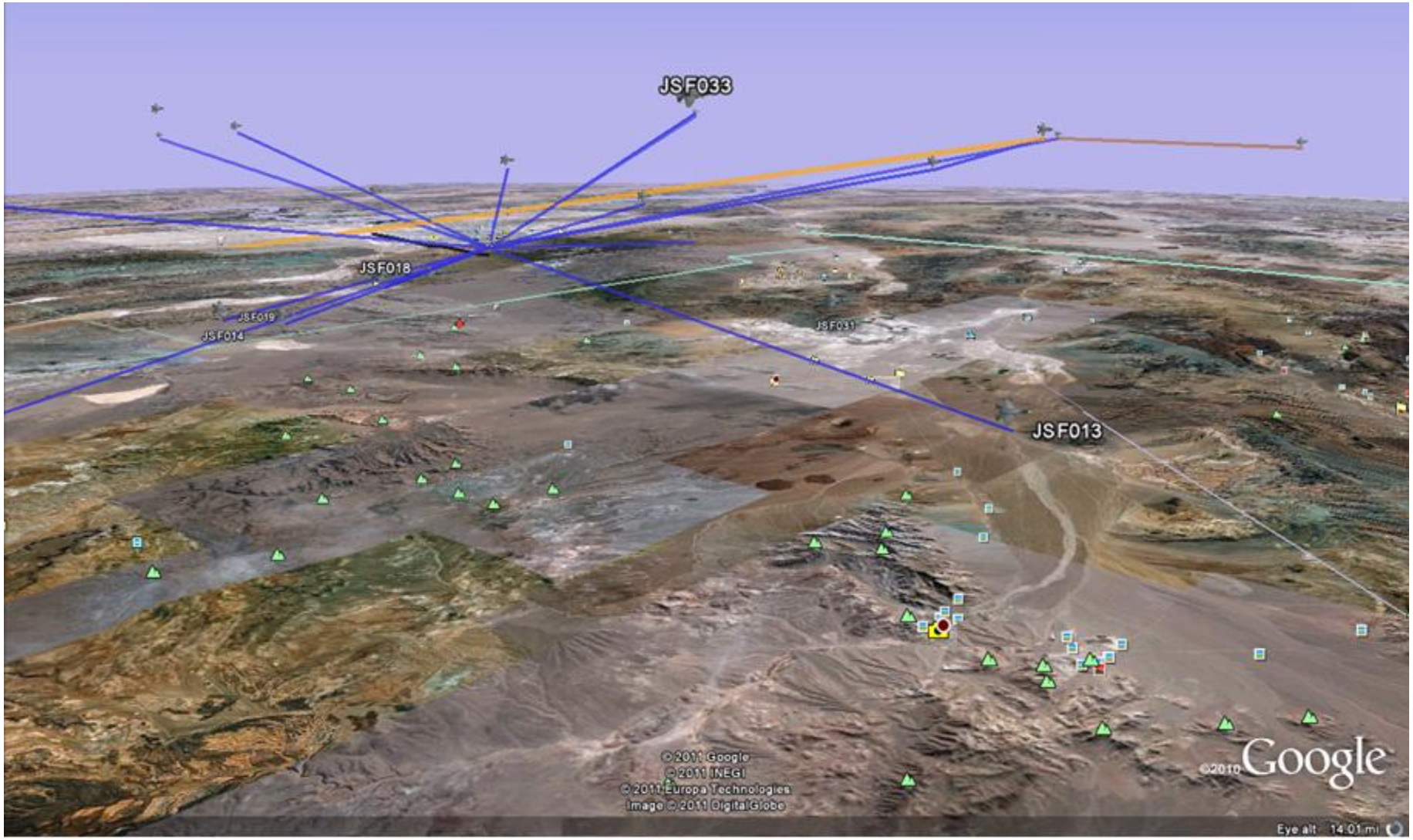






Rockwell Collins

Example Network Topology



Why is Simulation Required?

- The number of variables involved in a networked waveform are far too many for static analysis
  - Vehicle/Aircraft types
  - Vehicle/Aircraft dynamics
  - Antenna patterns per vehicle/aircraft
    - including polarization and shadowing
  - Different network sizes
  - Traffic profiles
  - Different bandwidth usage profiles

- Networking is not about absolute determinism, but rather statistical probability
  - Requires repetitive testing to characterize a network
  - Requires both:
    - controlled sequences of events
    - random sequences of events

Why is Simulation Required?

- Testing network limits is impractical with real hardware
  - Maximum number of nodes (100+)
  - Maximum bandwidth utilization (90% - 100%)
    - With limited assets available, it requires unrealistic loading on individual nodes

- Validating a network design requires different types of testing:
  - Repetitive (Regression) testing *with fixed conditions* to ensure network behavior is deterministic to the desired degree
  - Repetitive (Regression) testing *with injected randomness* to discover hidden corner conditions and network heuristics
  - Human Gremlin - testing with an eye to breaking things
    - Intentionally stressing network in ways it may not be intended to be used
    - Ensuring it ends up in a known state and recovers under all conditions
  - Monte Carlo style testing
    - Automated running of tests with data collection and analysis to determine boundary conditions and statistical probabilities of the network

Testing and Characterization of a Networked Waveform is expensive and requires a comprehensive testing strategy

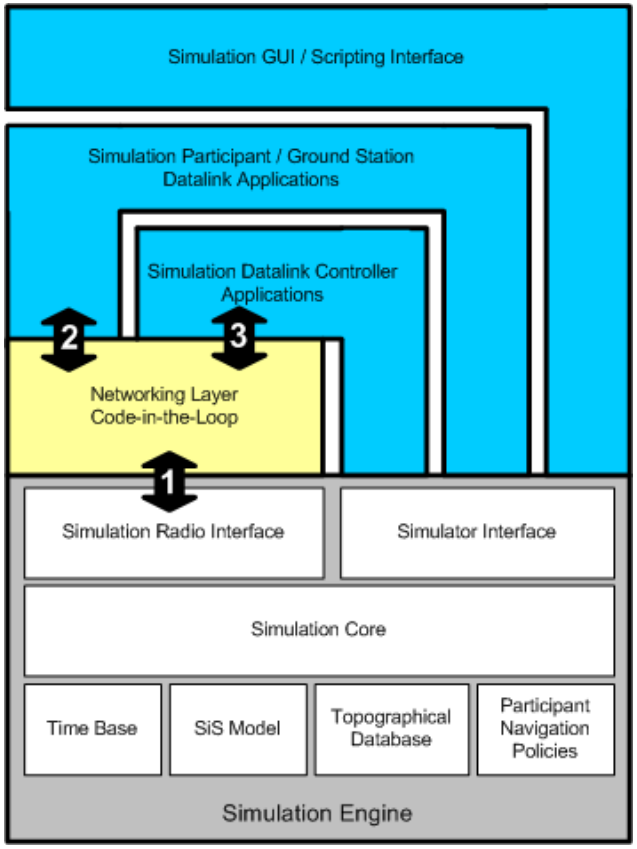
What comprises a Networked Waveform Simulator?

- A network simulator facilitates focused testing on the networking layer of a waveform
  - Simulates targeted non-networking aspects of the overall system

- Well-defined interfaces allow Code-in-the-Loop use of networking layer

  1 Networking to SiS interface

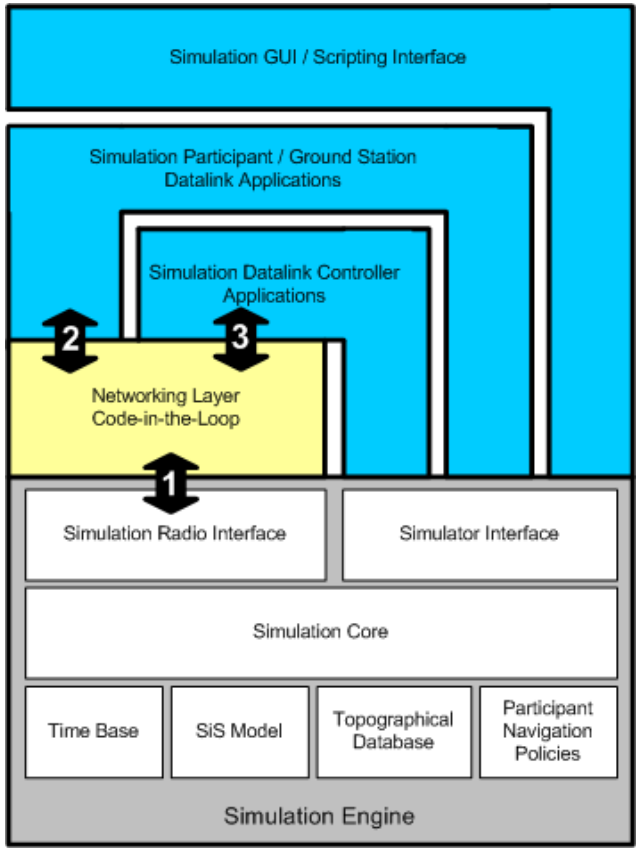
  2 3 Upper Layer Applications



Well defined interfaces are key to facilitating effective Code-in-the-Loop simulation

What comprises a Networked Waveform Simulator?

- Simulator Engine
  - Provides a controlled, synthetic environment
    - Topography (physical terrain, obstructions)
    - Node Navigation / Mobility
    - Signal in Space Model
      - Antenna models
      - Physics of waveform modulation
      - Propagation delays / effects
    - Time
      - Simulation
      - Allows pausing of simulation for inspection
      - Allows simulation to run slower than real-time as model fidelity increases
      - Allows initial simulation to focus on networking algorithms themselves independent of real-time constraints



Considerations for Simulation

- Before you ever start, you should be able to answer these two questions:
  - **What are we trying to measure?**
    - Simulation Architecture Requirements
  - **What do we want to measure in the future?**
    - Refine Simulation Architecture Requirements
    - Tradeoff Criteria Determination

The answers to these two questions have a significant impact on the total cost of Simulation

Considerations for Simulation

- Determining what you *ARE NOT* simulating is almost as important as determining what you *ARE* simulating

- There is a tradeoff between the fidelity of the model and the hardware resources required to keep simulation real-time

What are Some Uses of Simulation through the Development Lifecycle?

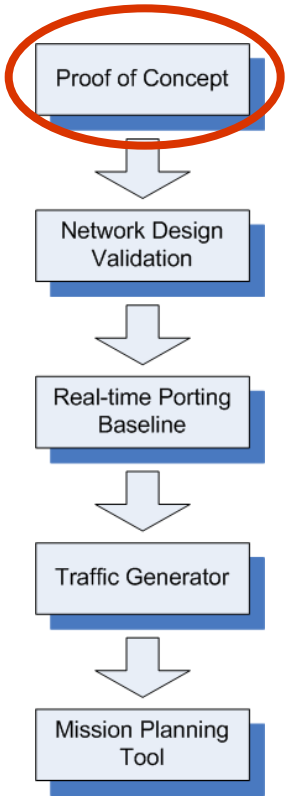
- Proof of Concept
  - Determine if this is a viable network design
  - Determine risks in proposals (RFIs / RFPs)
  - Perform initial trade studies
    - Determine the right thresholds and objectives
- Network Design Validation
  - As the simulation model is matured, better assessments can be made for corner conditions and design constraints, as well as requirements trades
  - Prior to Hardware being built
- Real-time porting baseline and debugging tool
  - Once real hardware is available, INTEGRATION begins
  - Simulation provides a baseline characterization that can be used to diagnose and track down hardware porting issues

What are the Uses of Simulation through the Development Lifecycle?

- Traffic Generator
  - If the Simulation Environment is designed with proper hooks in place, the simulated nodes can generate network traffic
    - Lab bench testing of real hardware in a loaded condition
    - Flight testing can be with a loaded network as well

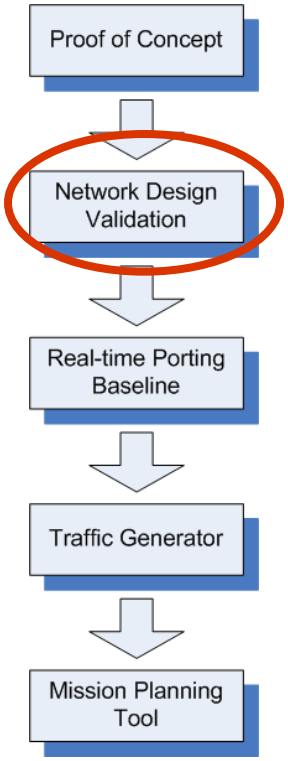
- Mission Planning Tool / Mission Playback Tool
  - Growth opportunity to enhance the simulation model fidelity such that missions can be validated via simulation before any aircraft deploy
    - Identify network choke points in mission plans
    - Ensure adequate network coverage for theater of operations
    - Test logistical aspects of larger networks
  - Playback allows captured flight data to be fed into Simulator
    - Allows refining of simulator model fidelity by comparing actual flight data to simulated flight data

Proof of Concept Phase



- Focus
  - Straw-man fleshing out of network algorithms
  - Discover dynamic aspects and corner conditions
  - Mitigation of High-Risk Items
  - Identification of key strengths and limitations
- Characteristics
  - Low-Fidelity physical environment modeling
    - Basic Signal in Space model
    - Basic antenna models
    - Basic topography models (maybe even 2-D vs. 3-D)
    - Entire Networking solution does not need to be implemented or simulated
      - Only that which is necessary to mitigate high risk items
- Special Considerations
  - Is this throw away code?

Network Design Validation



- Focus
  - Full development and demonstration of networking algorithms and services
  - Code-in-the-Loop simulation
  - High-Fidelity Signal in Space model (with antenna models)
  - High-Fidelity Topographical models
  - Uncovering and fixing any dynamic aspects and corner conditions
  - Mitigation of as many risk items as possible
  - Documentation of Network Design (with trades documented)

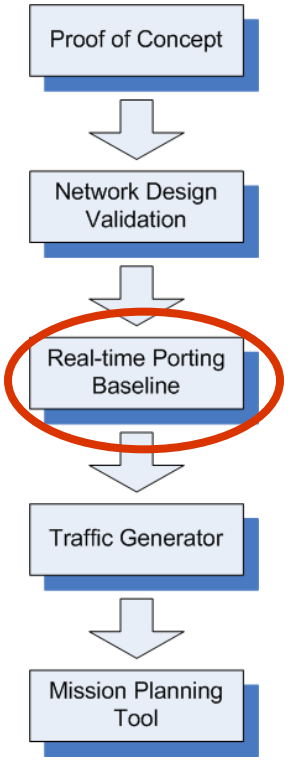
- Characteristics
  - Target hardware is not yet available

- Special Considerations
  - While waiting for real hardware, is there benefit to porting to an evaluation board?
  - Are accurate antenna models required for proving network design?
  - How accurate does our model need to be to adequately validate the network design?

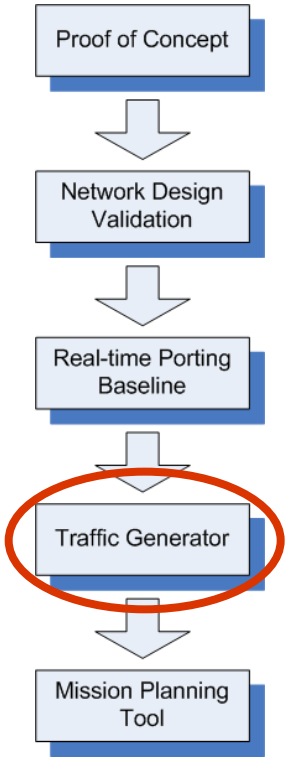
Real-time Porting Baseline

- Focus
  - Simulation is used as a performance baseline to isolate porting issues
  - As porting bugs are fixed, retest fixes in simulation (Code-in-the-Loop)

- Characteristics
  - Target specific simulation scenarios that validate issues found in porting process
  - Simulator capabilities are not further refined or developed, but rather used as a performance reference point
  - Whenever real-time bugs are fixed, the simulator code-in-the-loop must be rebuilt with fixes and retested in the simulation environment



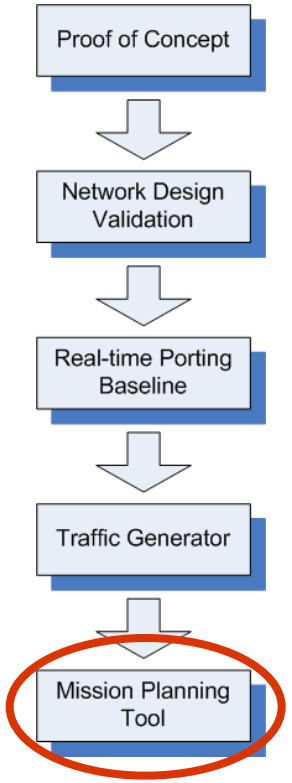
Traffic Generator



- Focus
  - Generating network traffic (network traffic loading)
  - Simulating real network nodes (network tree processing load)
  - Assessing real hardware performance with network loading
  - Make any simulator real-time performance enhancements (if necessary)
    - Must work in conjunction with real hardware
    - It is possible that fidelity must be reduced in certain simulation models in order to perform in real-time
- Characteristics
  - Focus in this phase is not networking algorithms, but rather how the real hardware performs in various loading conditions

Mission Planning Tool

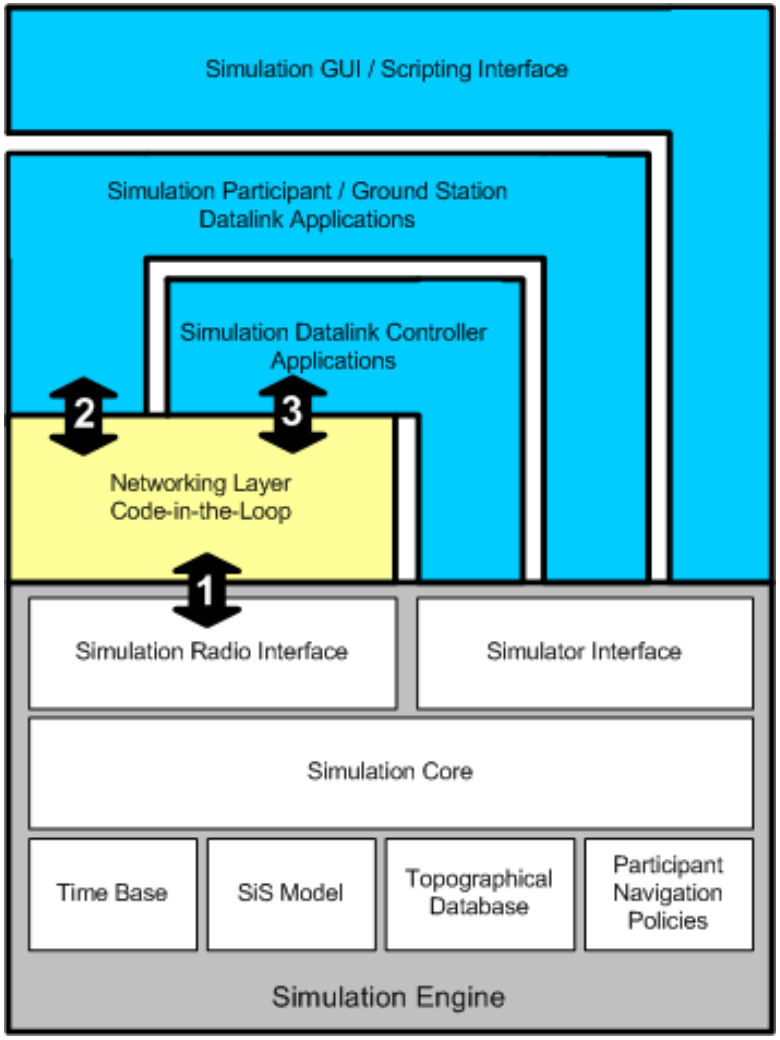
- Focus
  - Enhancing the simulation fidelity to the point where it can be reliably used to predict mission performance
    - Enhanced antenna models
    - Enhanced topography with terrain modeling (will slow simulation way down)
    - Close the loop on Signal-in-Space performance with real flight testing data
  - Real-time performance is not the focus, accuracy is the focus



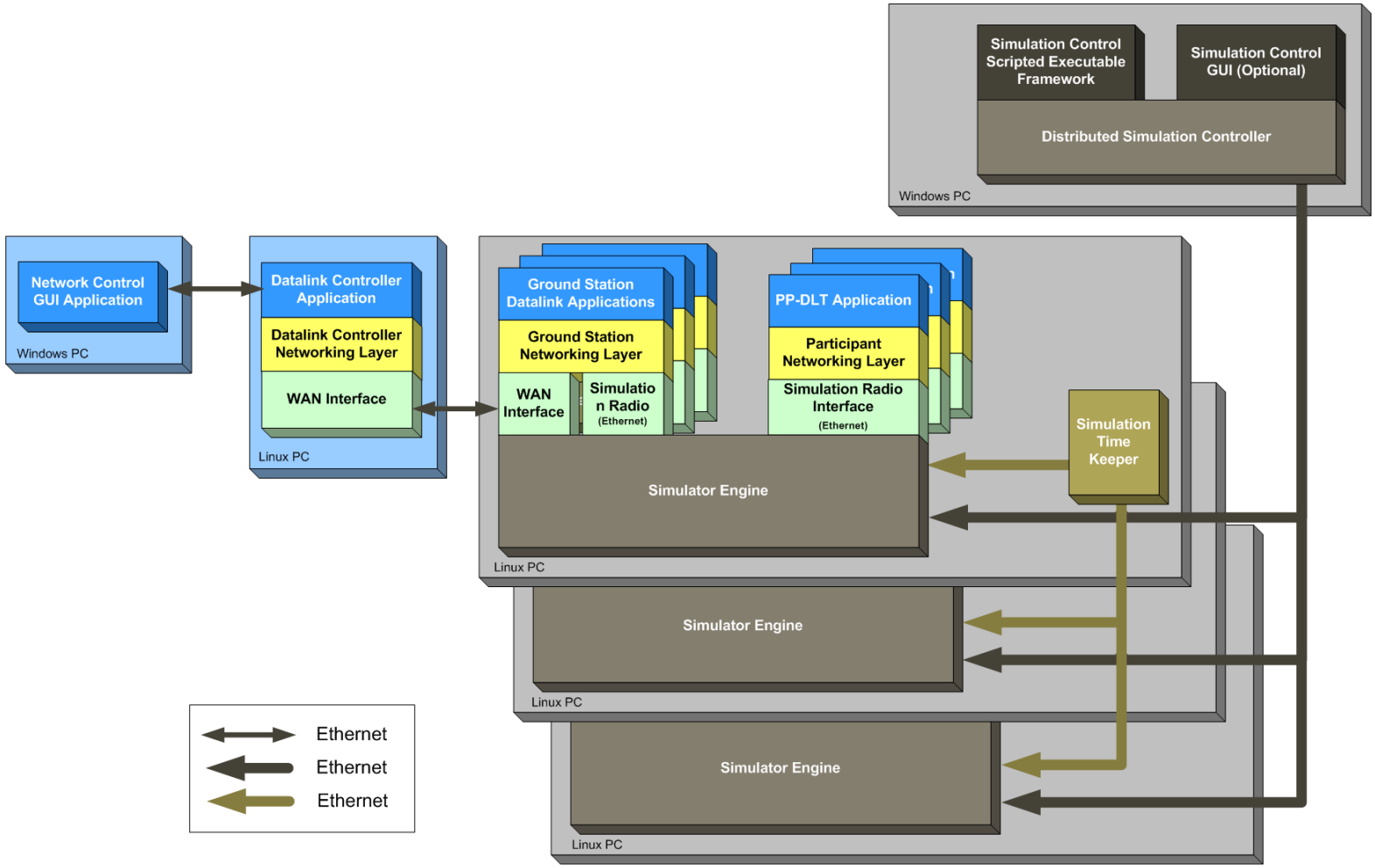
Simulator Architecture

Simulator Architecture and Features

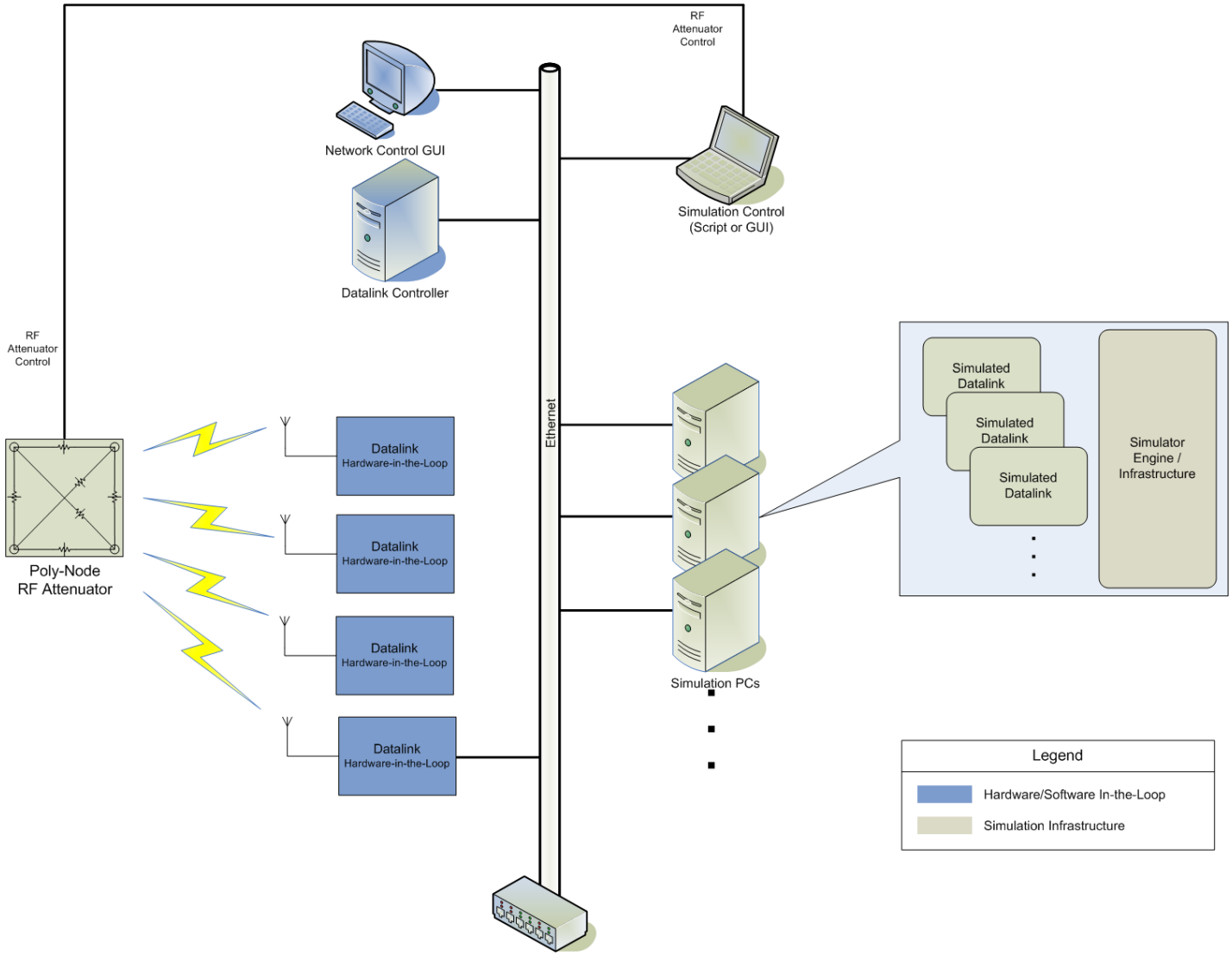
- GUI front-end
- Optional scripting capability
  - Monte Carlo simulations
- Code-in-the-Loop capability
- 3-D physical model
- Earth curvature (WGS-84)
- Simulated Time Base
- Antenna models
- Antenna polarization
- Participant Navigation



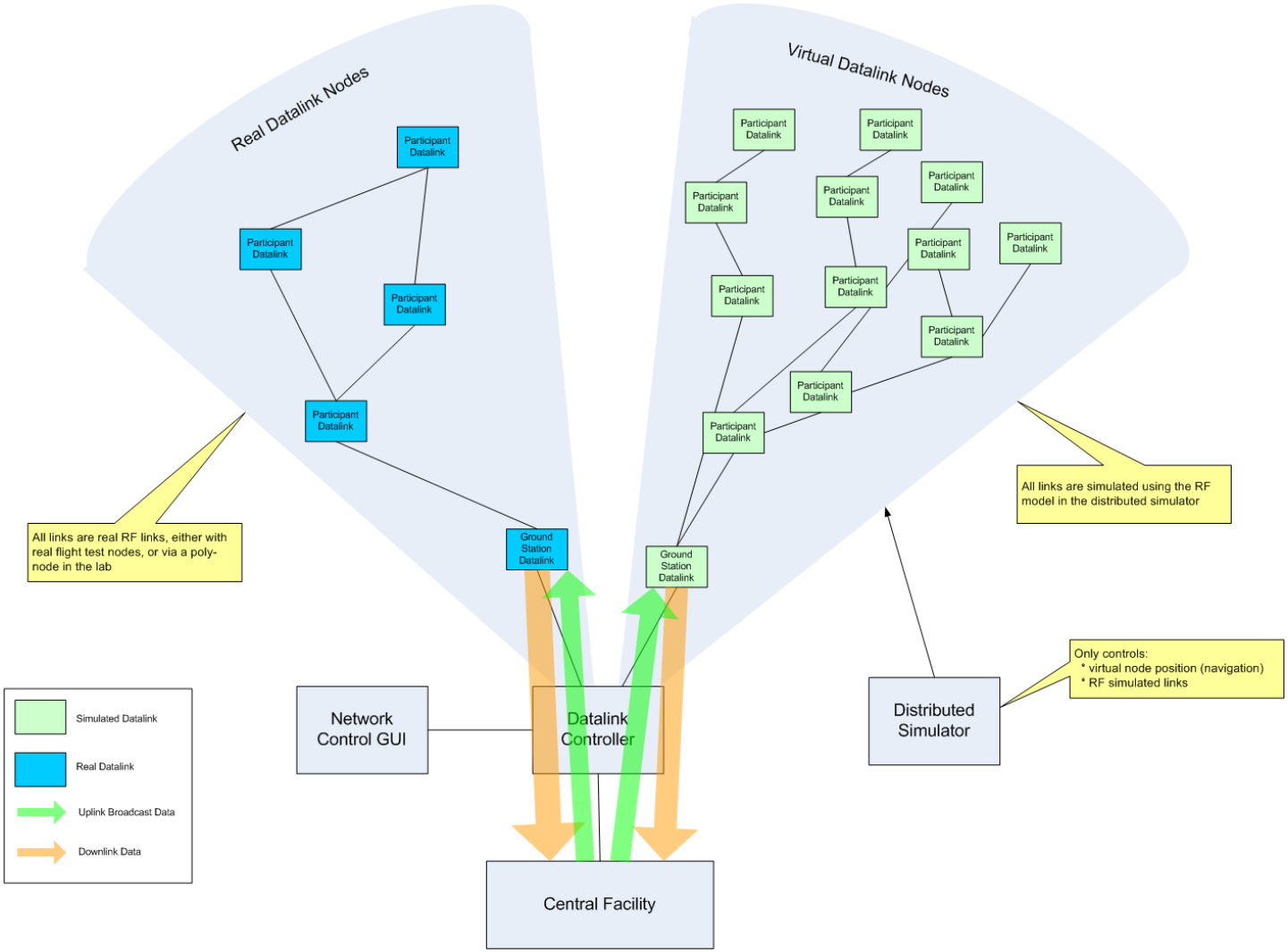
Simulator Physical Architecture



Simulator Physical Architecture – Hardware-in-the-Loop



Traffic Generation / Network Loading



Software considerations with simulation

- Start with simulation in mind
  - Simulator architecture must be compatible with networking code
  - Concurrency model vs. Software Development Plan
    - Global/static variables
    - Threading models
    - Utilization of 3rd party tools
    - Abstraction layers to enable simulation
- Define clear interfaces between layers
- Make sure abstraction layers are efficiently implemented
  - Many of them are high iteration, and if not efficiently done can negatively impact the final code
- Iterative Development Cycles
  - Simulation model is not effective with waterfall development model

Examples of Real Simulation Findings

- Efficient slot allocation is very complex and can lead to computationally intensive calculations
- Early trade study on multiple networking modes uncovered the need for customized link quality thresholds that were dependent on mode of operation
- Three and four hop network spans do not happen easily - like water finding its level, the networking layer finds a shorter path before humans can see it visually
- Make-Before-Break paradigm for healing broken data paths was verified to reduce data loss and was weighed against the temporary increase in bandwidth required