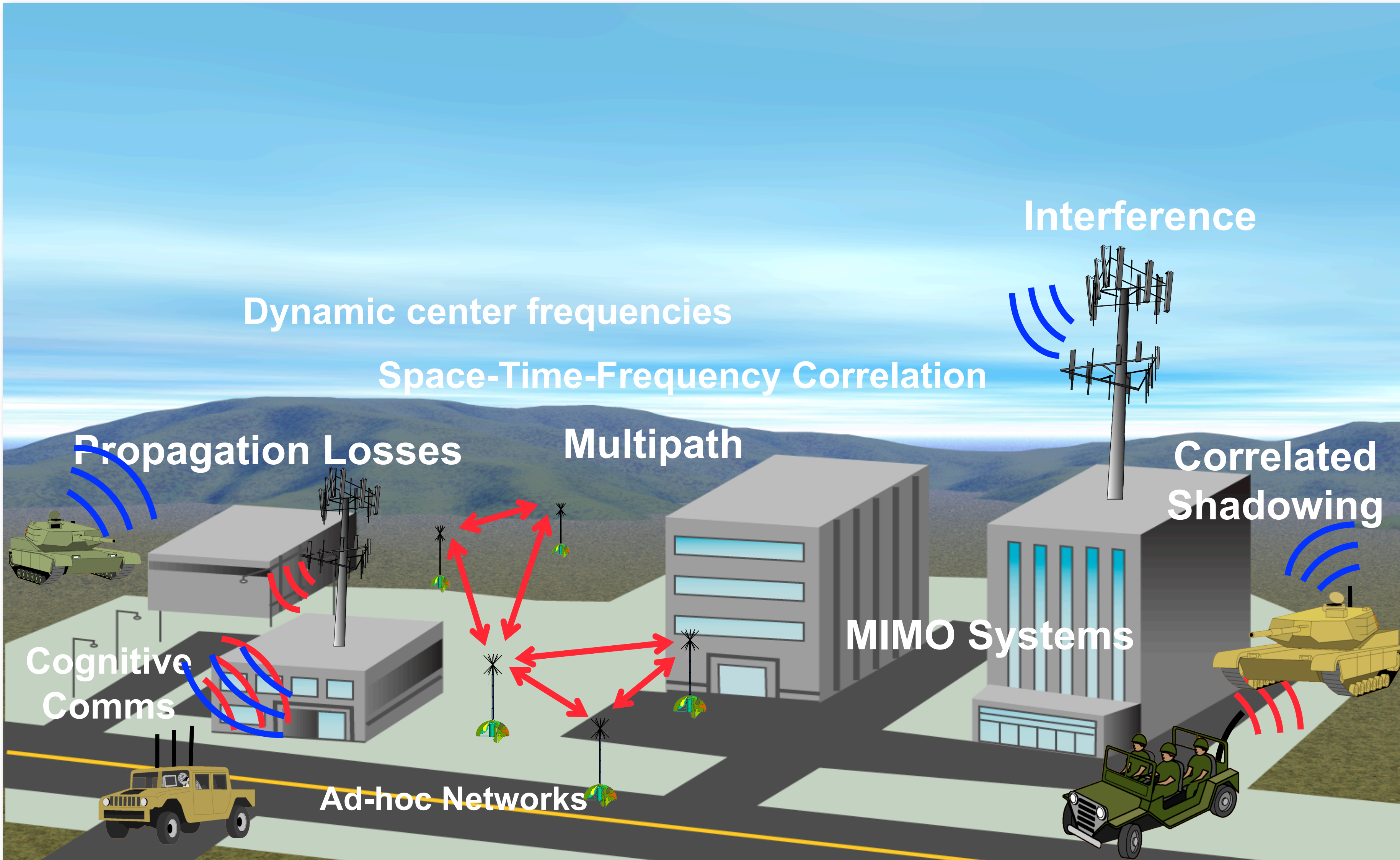
LLAMA Comm

Lincoln Laboratory Ad-hoc MIMO Adaptive Communication Simulation Tool

Adam Margetts, Bruce McGuffin,
Dan Bliss, and Lloyd the Llama

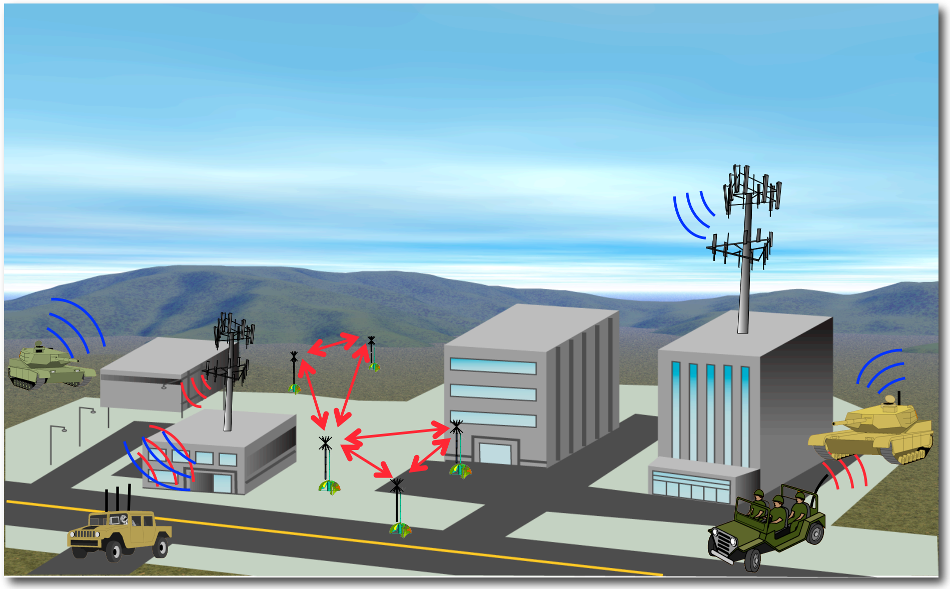
Simulation Environment Overview



Simulation Capabilities

LLAMA Comm

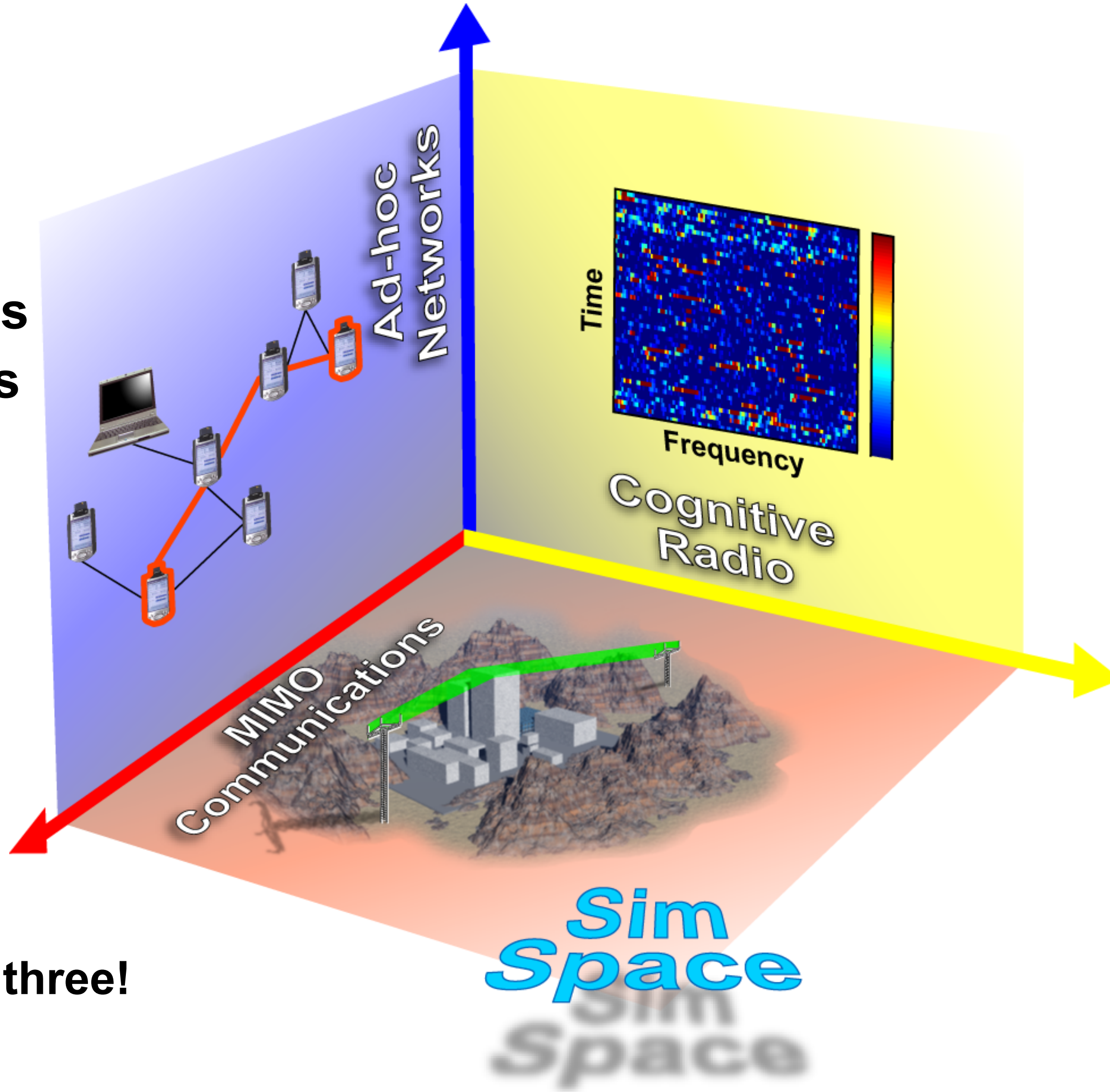
- Operate multiple radio nodes simultaneously and asynchronously
- Generate realistic space-time-frequency MIMO fading channels
- Model propagation over a variety of environments:
  - Urban, suburban, rural
  - Indoors, outdoors, fixed, mobile, or airborne
  - Line-of-sight or non-line-of-sight
  - Link lengths meters to tens of kilometers
  - Frequencies: 10 MHz to 6 GHz



Problem Set

Motivating Scenarios:

1. Cognitive Radios
2. Ad-hoc Radio Networks
3. MIMO Communications

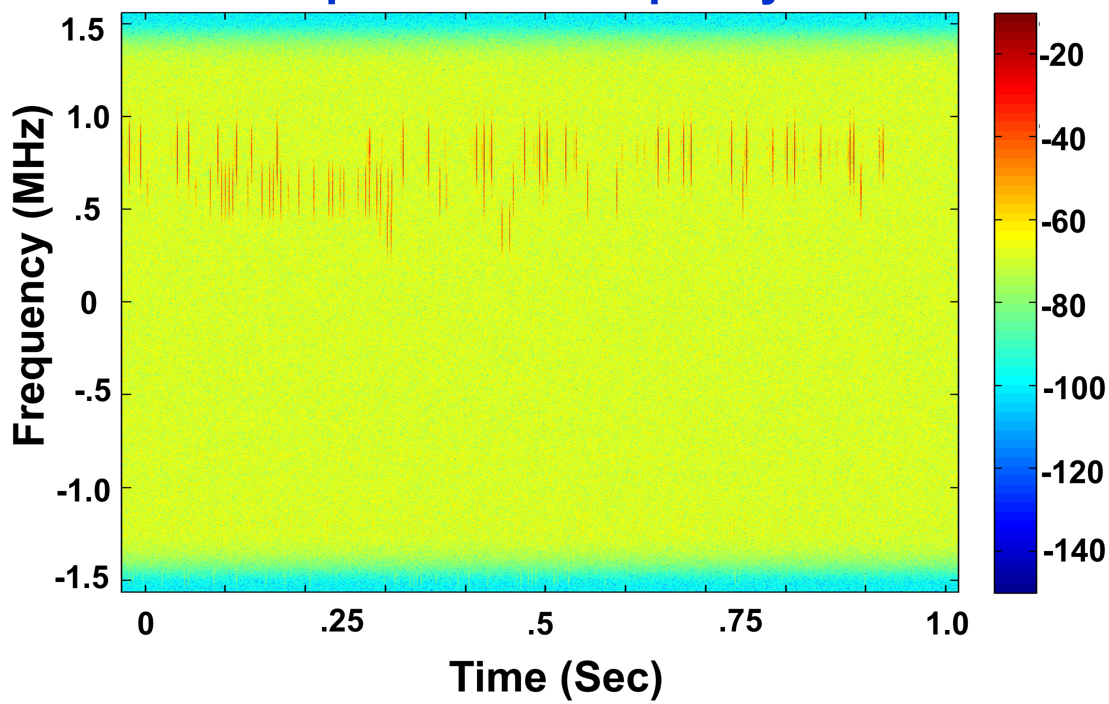


LLAMAComm combines all three!

Adaptive Cognitive Radio

- Radio spectrum is underutilized
- Cognitive radios identify and occupy time-frequency gaps
- Requires simulating a wide range of center frequencies

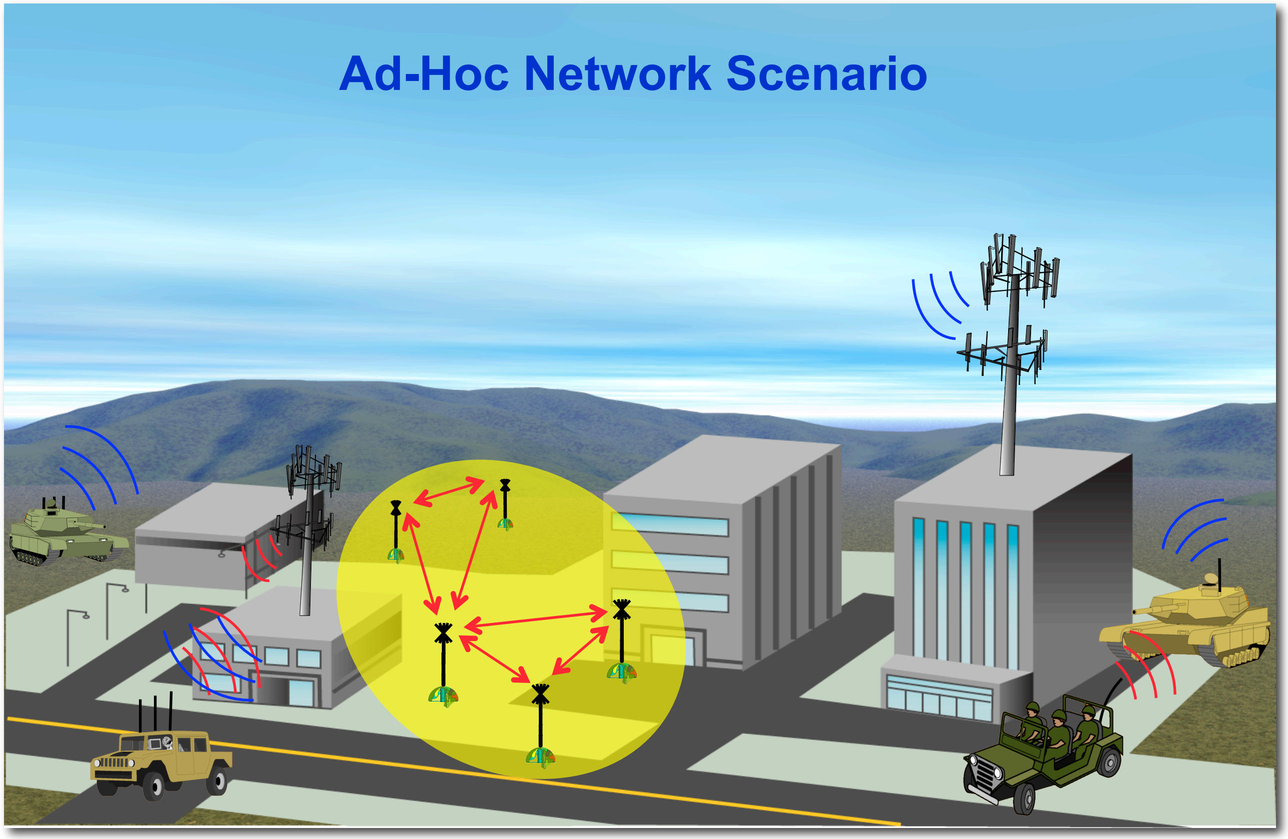
Spectrum Occupancy



Cognitive Radio Scenario

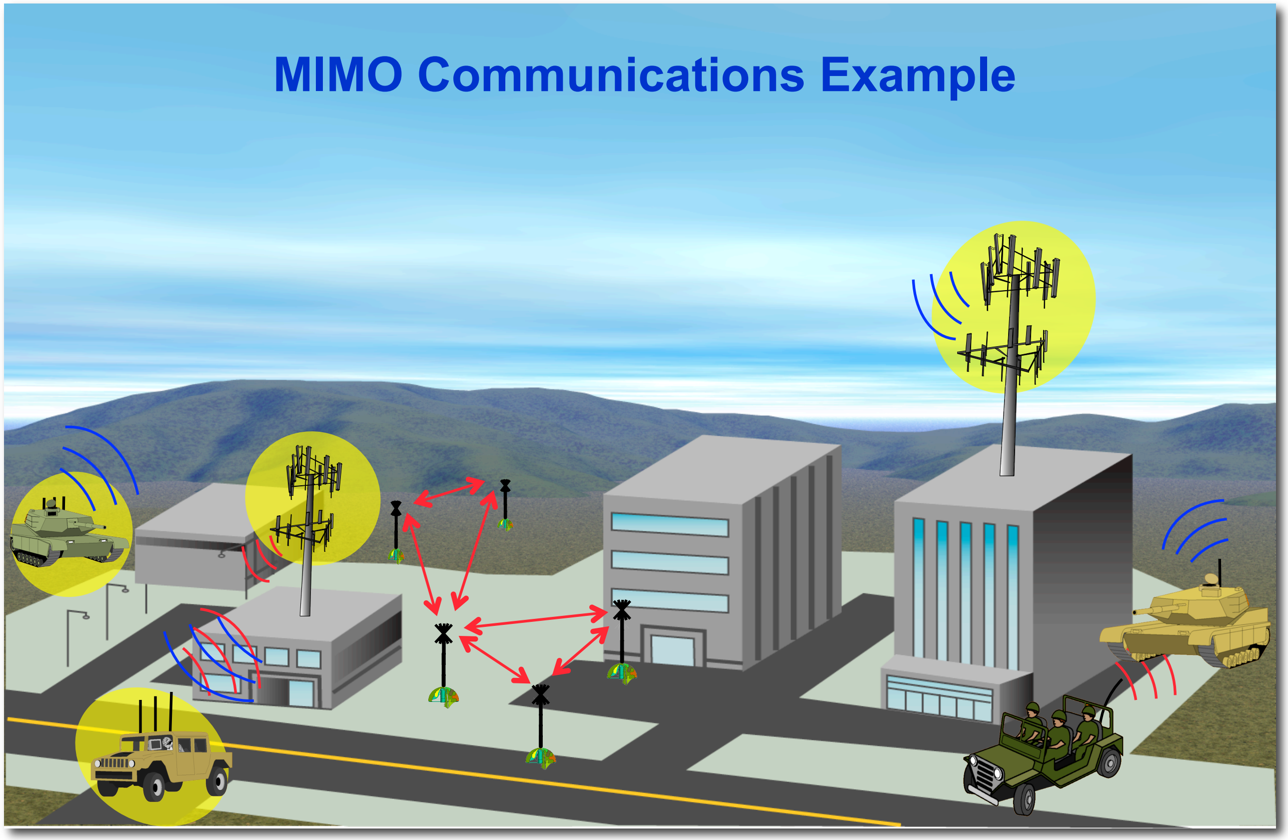
Wireless Ad-hoc Networks

- Formed dynamically
- Multi-hop communications
- Nodes interact asynchronously
- Simulation provides datalink layer control



MIMO Communication
Multiple-Input Multiple-Output

- Arrays of antennas at transmitter and receiver
- Increases reliability and throughput
- Adaptively nulls interference sources
- Simulate space-time-frequency correlations



Available Wireless Network Simulators

- Opnet
  - Commercial
  - Probabilistic physical layer
- Ns2
  - Open source (large research user base)
  - Probabilistic physical layer
- QualNet
  - Commercial
  - Probabilistic physical layer
- ISIS Prowler (Probabilistic Wireless Network Simulator)
  - Gyula Simon (Vanderbilt University)
  - Written in Matlab
  - Probabilistic physical layer
- SWANS (Scalable Wireless Network Simulator) (built on JiST)
  - Claimed 1,000,000 nodes simulated on a PC
  - Probabilistic Physical Layer

Physical Layer Too Simple

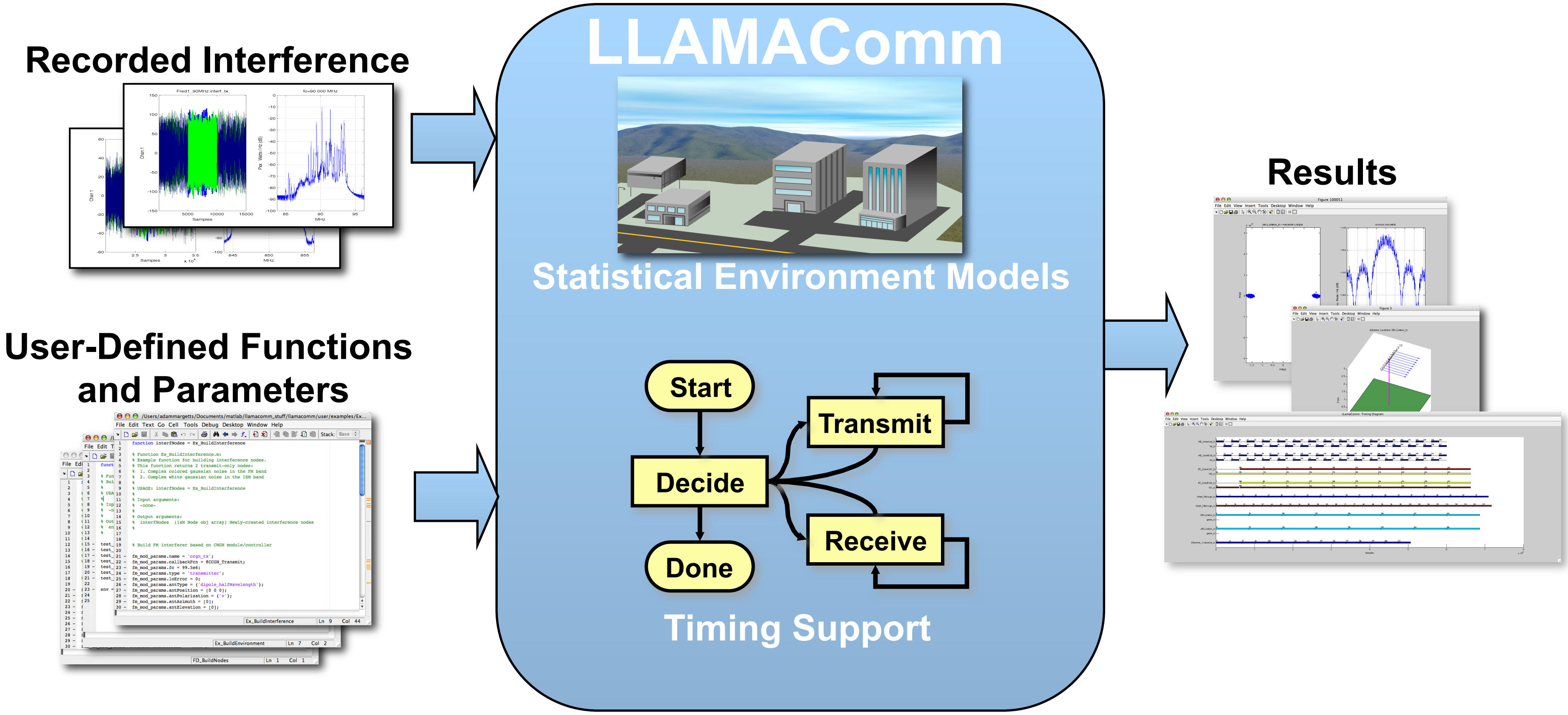
- Numerous cell-site planning tools
- Wireless Insight (cousin to XFDTD)
  - Solves Maxwell Equations

Too Complex

- *Simulation should be as simple as possible, but no simpler*

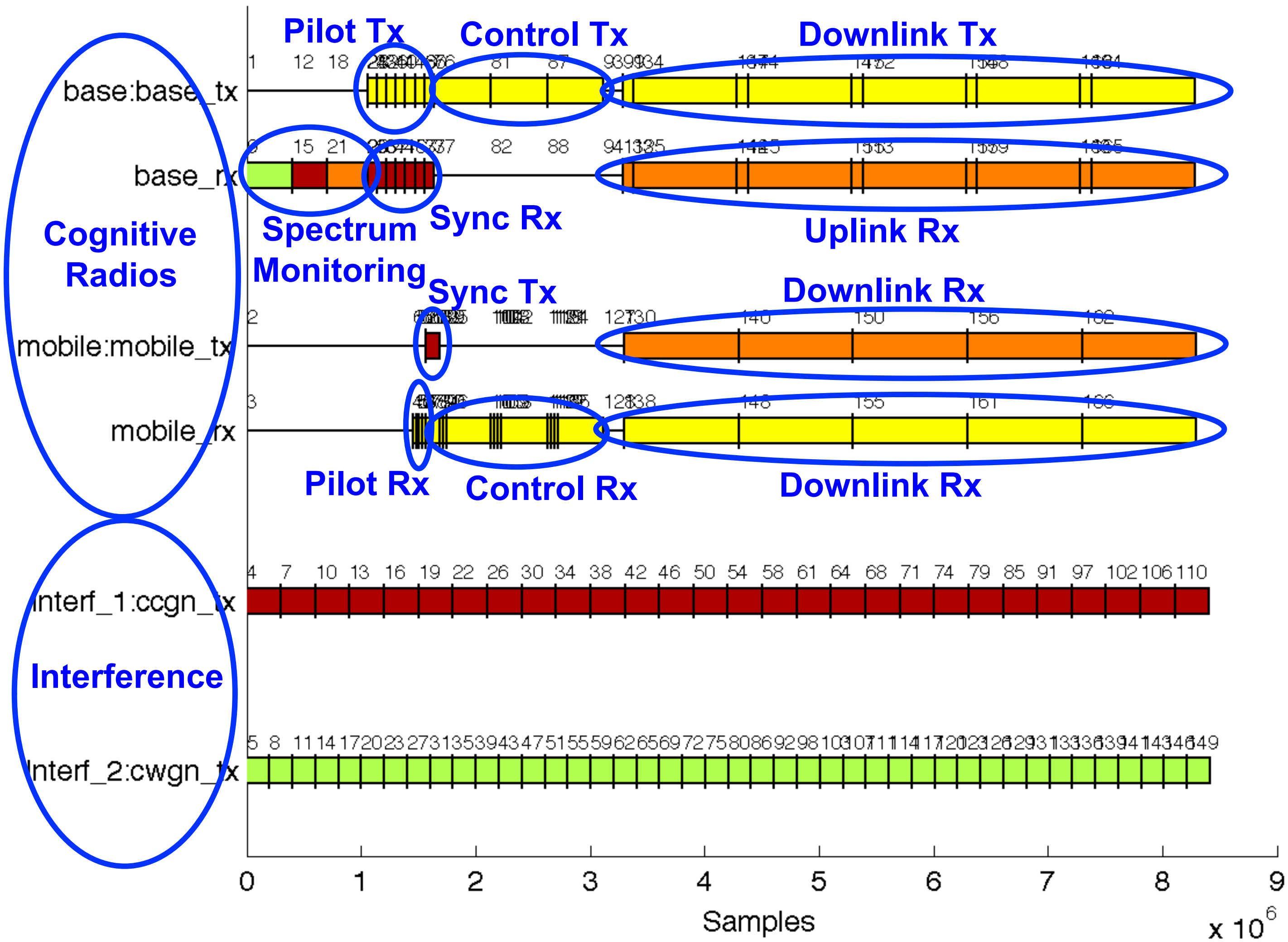
LLAMAComm Overview

- LLAMAComm is a MATLAB simulation tool
- Simulates cognitive Multiple-Input Multiple-Output (MIMO) ad-hoc wireless networks



LLAMAComm Timing Diagram Example

- Complex datalink simulations are possible
  - Colors indicate center frequency



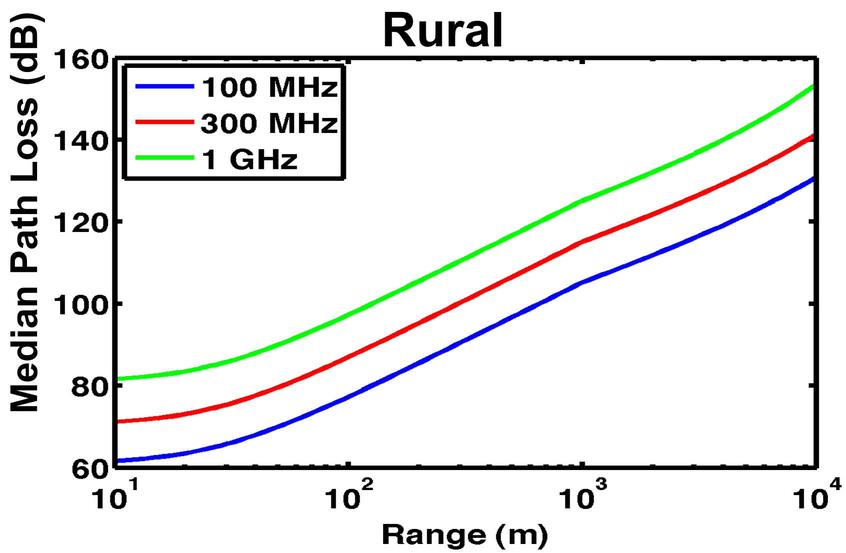
LLAMAComm Requirements

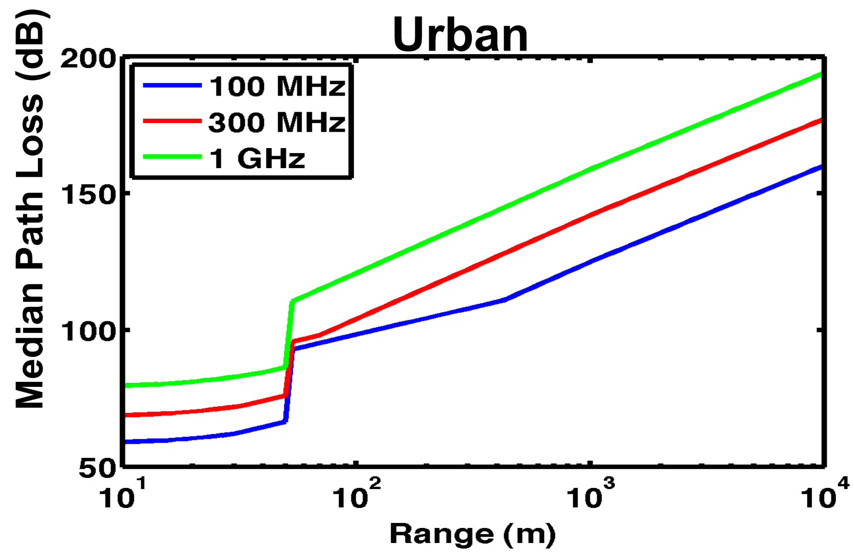
- **MATLAB R2009b or later**
  - Signal processing toolbox
- **Familiarity with MATLAB programming**
- **Fast processor**

Scope of LLAMAComm Path Loss Models

- **Input parameter range**
  - frequency: 10 MHz – 6 GHz
  - range: 1 m – 300 km(1)
  - antenna height: 1 m – 10 km
- **Supported environments**
  - urban/suburban/rural
  - in-building/building penetration/ outdoors
- **Model Outputs**
  - log-normal shadowing loss
    - median power loss
    - standard deviation
    - shadowing correlation with angle
  - multipath fading
    - Rice k-factor
    - delay spread

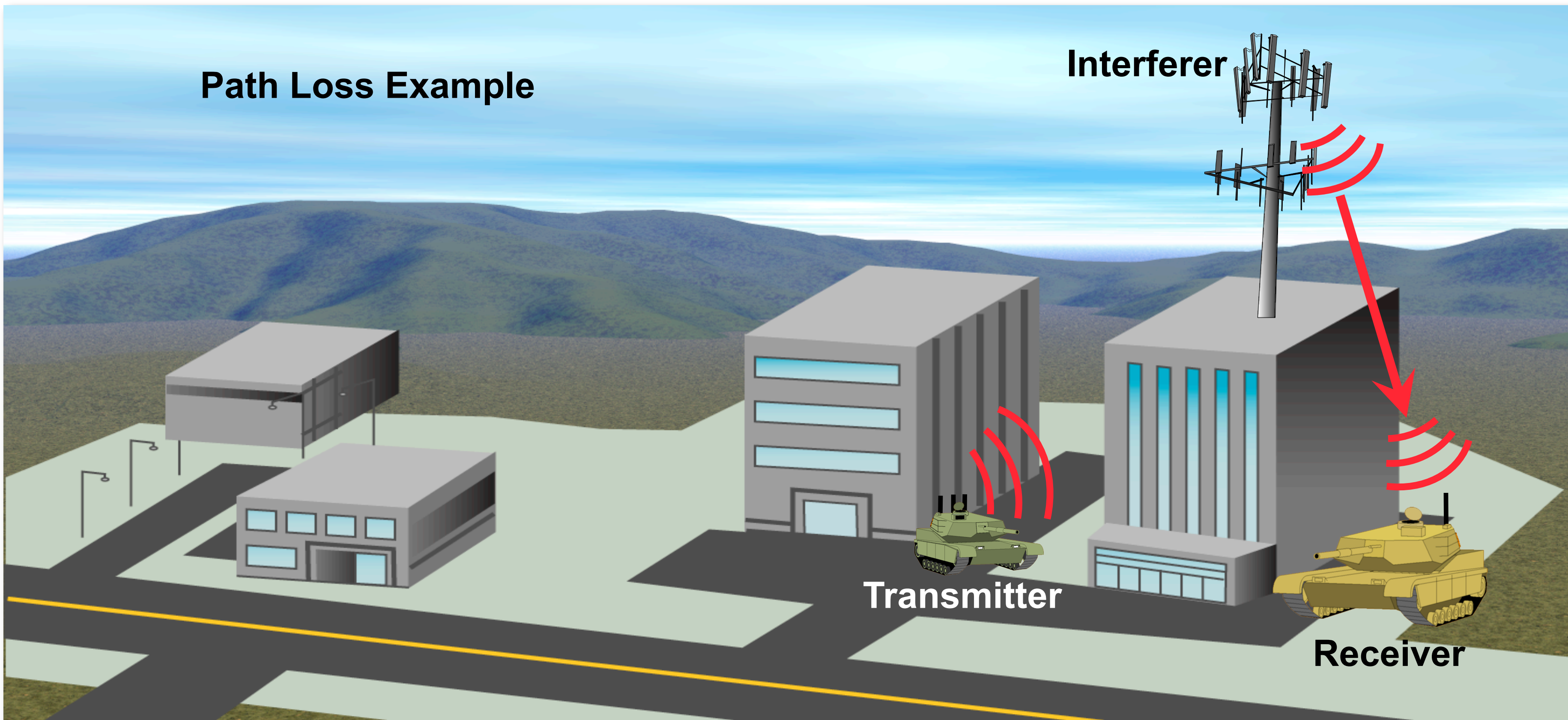
(1) Longer ranges only required for very high receivers





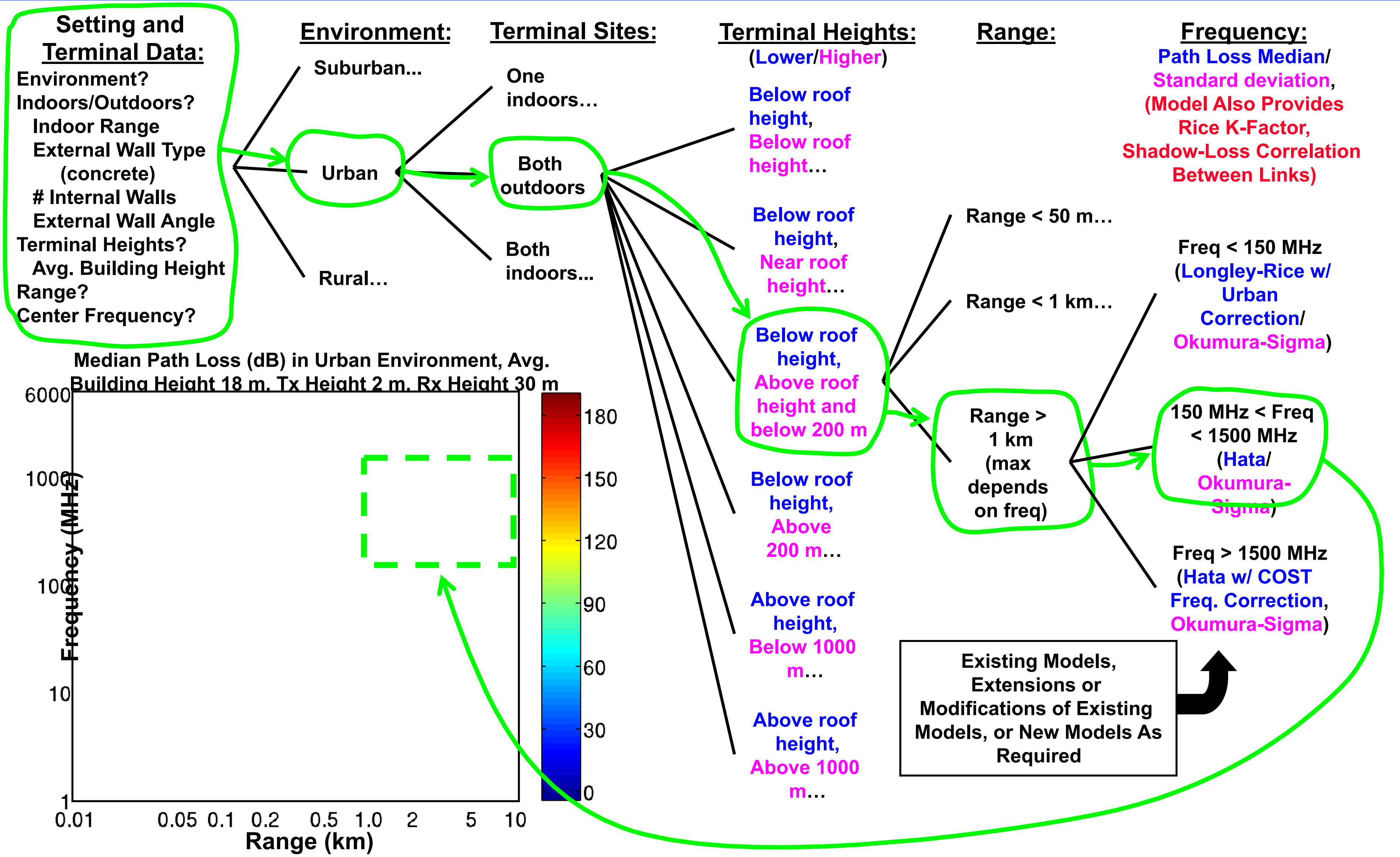
Path Loss Example

- **Examine path loss from Interferer to Receiver**
  - Urban environment
  - Outdoors
  - Receiver below roof height
  - Range > 1 km



Path Loss Model

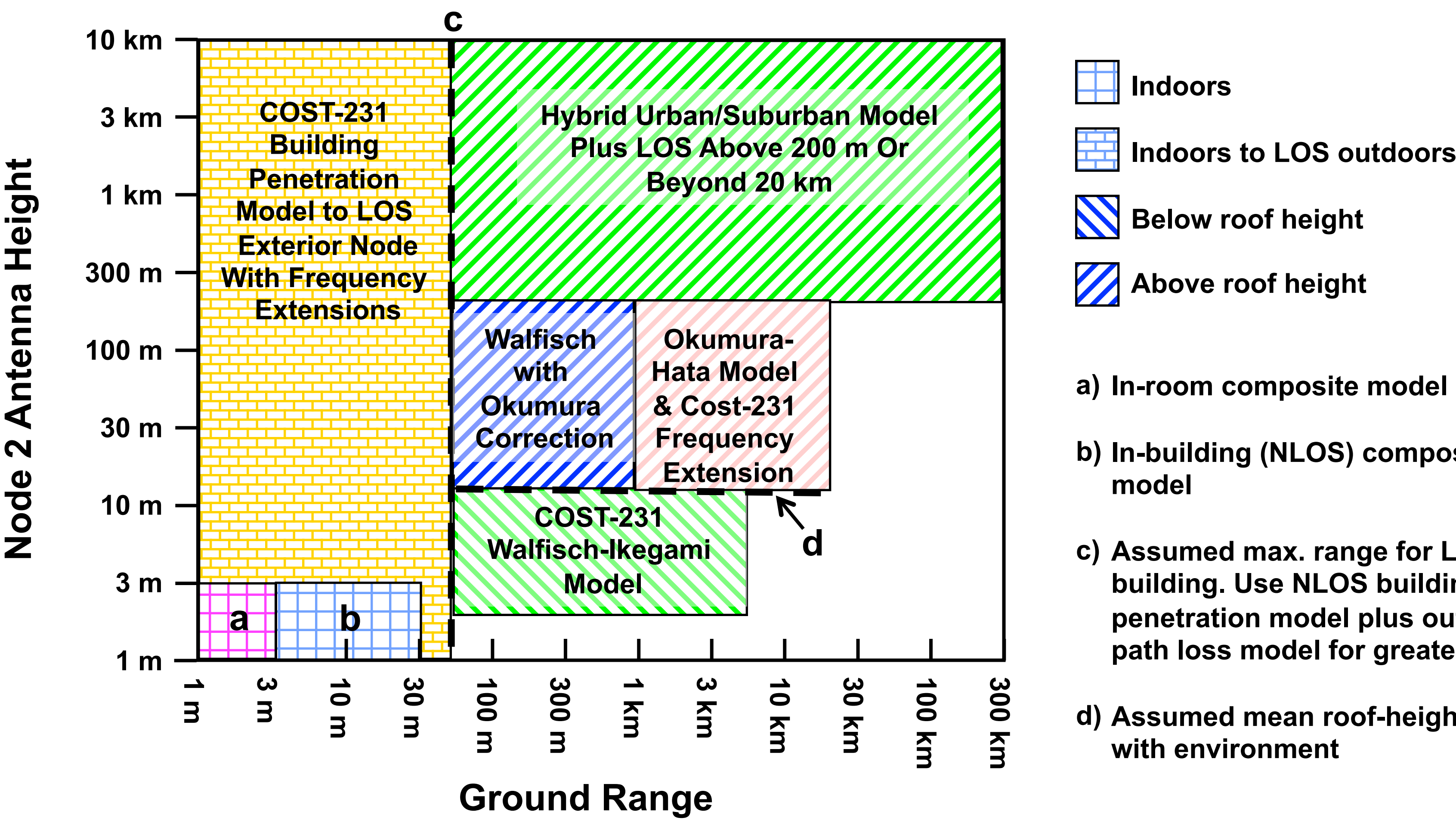
10 MHz-6 GHz, 10 m–20 km, Multiple Environments & Sites



Urban/Suburban Path Loss Models at 2 GHz

Node 1 Indoors

- Path loss model includes indoor-to-outdoor links
  - Loss depends on range, antenna heights, frequency, etc.



a) In-room composite model

b) In-building (NLOS) composite model

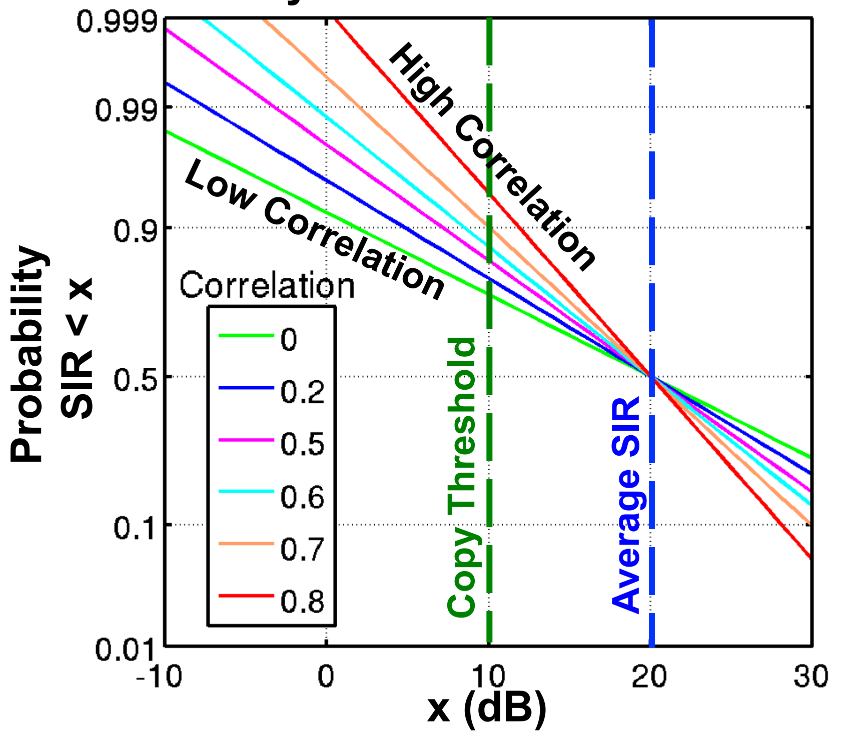
c) Assumed max. range for LOS to building. Use NLOS building penetration model plus outdoor path loss model for greater ranges.

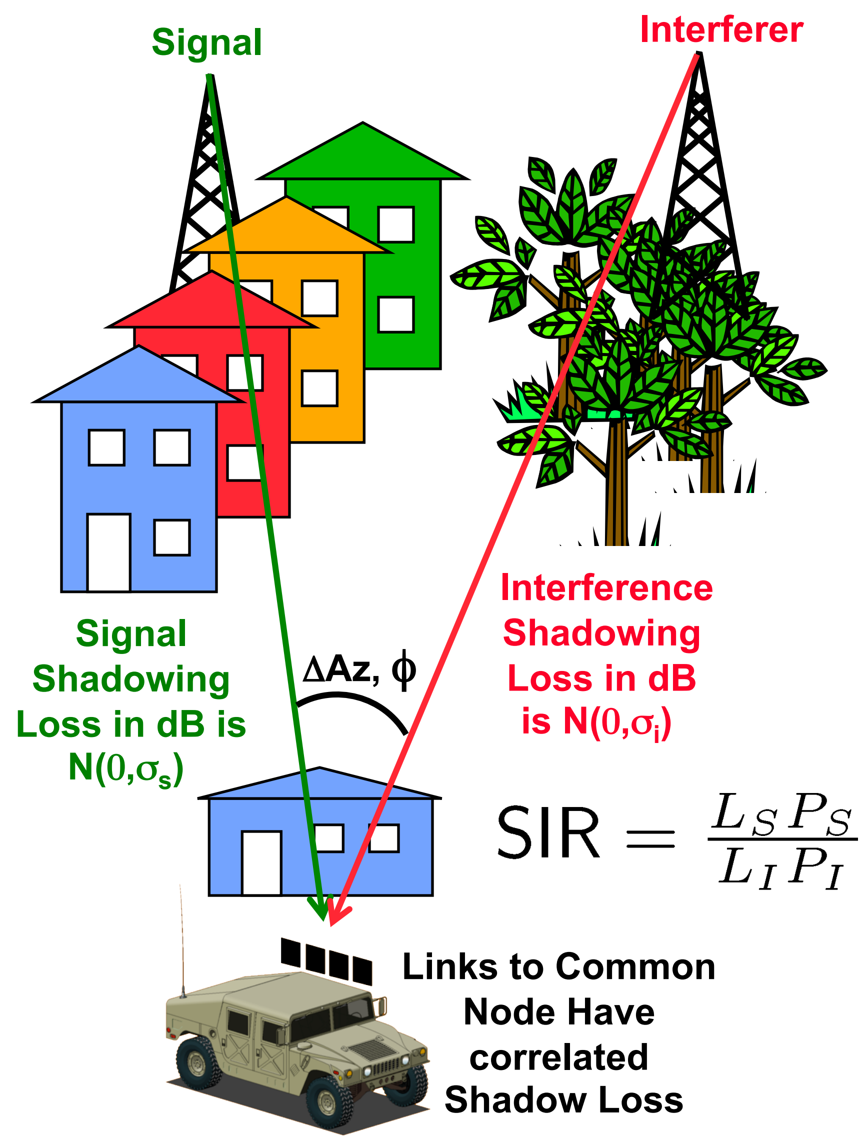
d) Assumed mean roof-height varies with environment

Correlated Shadowing

Shadow loss correlation reduces signal-to-Interferer power variation

Probability SIR < x for correlated links





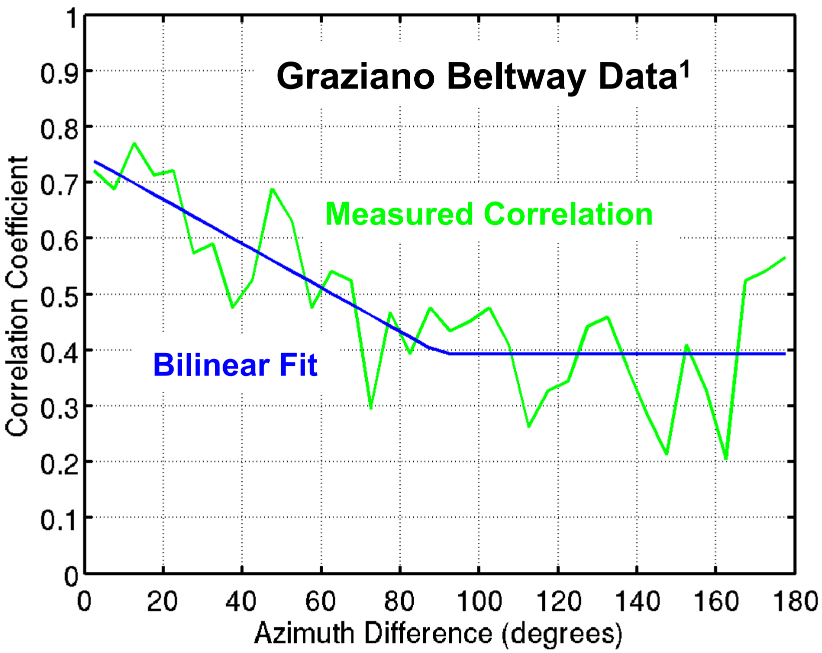
Example: Average SIR = 20 dB
Shadowing Std. Dev. = 10 dB
Copy Threshold = 10 dB
Copy Prob. Varies from 76% to 94%

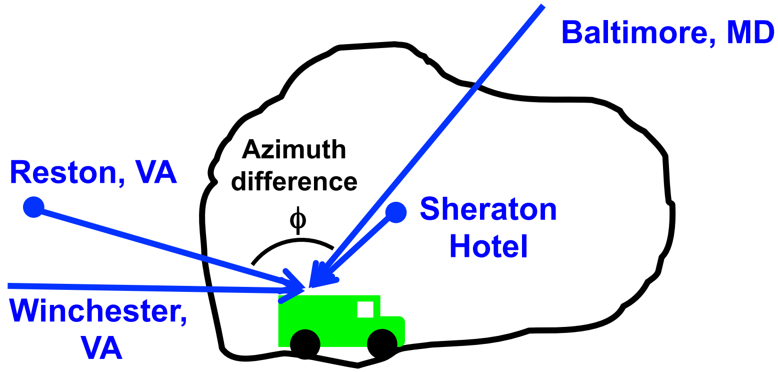
Shadowing Correlation Coefficient

- **Link shadow loss is correlated**
  - Depends on angle separation
- **Limited data available**
  - Graziano measured at 900 MHz around Washington Beltway[1]
  - Similar results by van Rees at 150 MHz, 450 MHz, 900 MHz in The Hague[2]
- **Approximate Graziano's measurements with bilinear fit**

$$\rho = \begin{cases} 0.75 - 0.004\Delta\phi & |\Delta\phi| \le 90^\circ \\ 0.004 & 90^\circ < |\Delta\phi| \le 180^\circ \end{cases}$$

  - Lower MSE than proposed functions found in literature
  - Min > 0 reflects homogeneous environment in different directions
  - Max < 1 reflects uncorrelated shadowing at far end of links

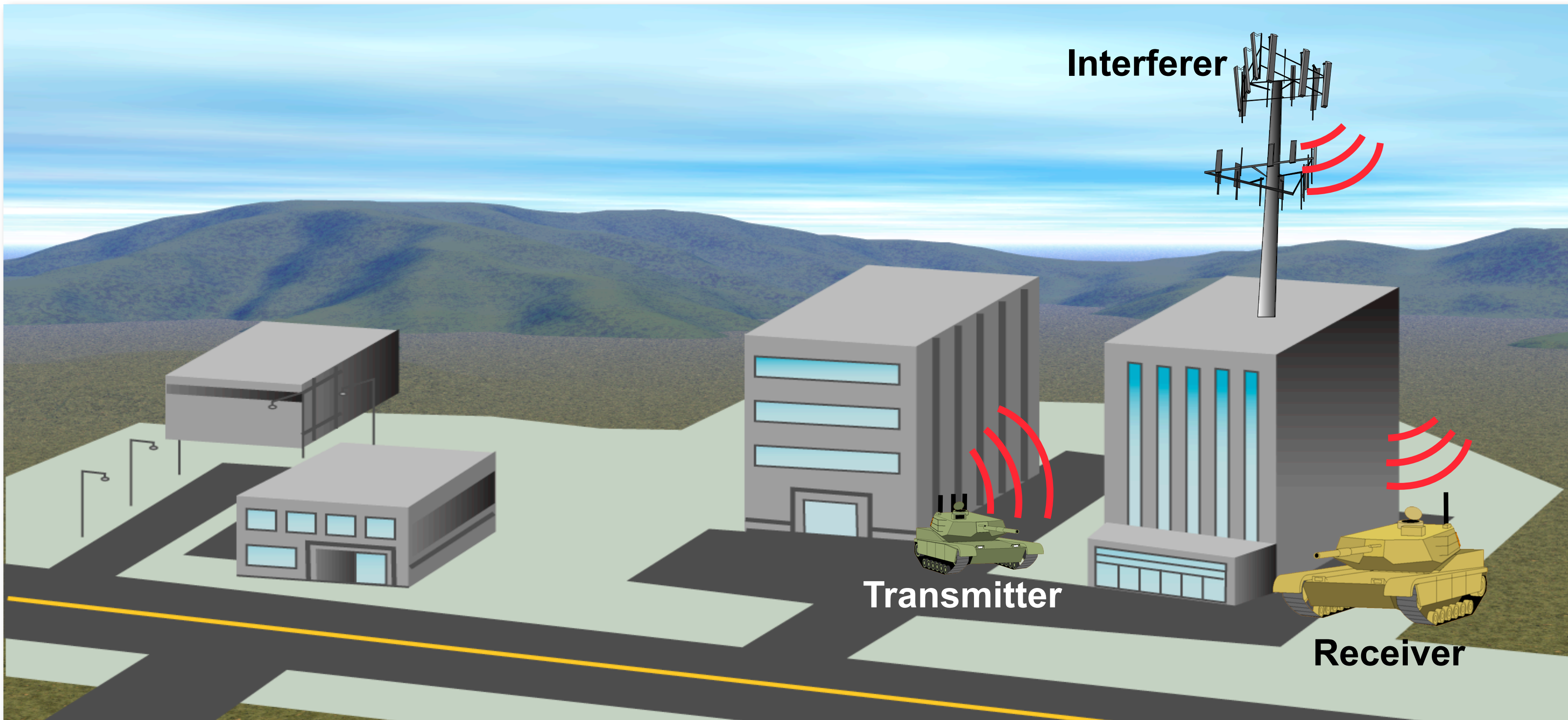




1) Victor Graziano, "Propagation Correlations at 900 MHz", IEEE Trans. Veh. Tech., Nov. 1978
2) Jan van Rees, "Cochannel Measurements for Interference Limited Small-Cell Planning", Archiv Elek. Ubertrag., 1988, pp 318-320

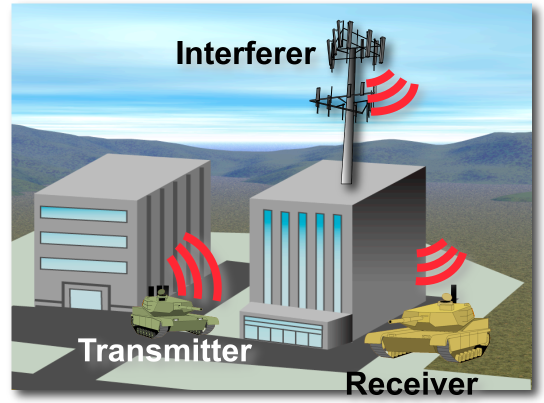
Channel Processing Example

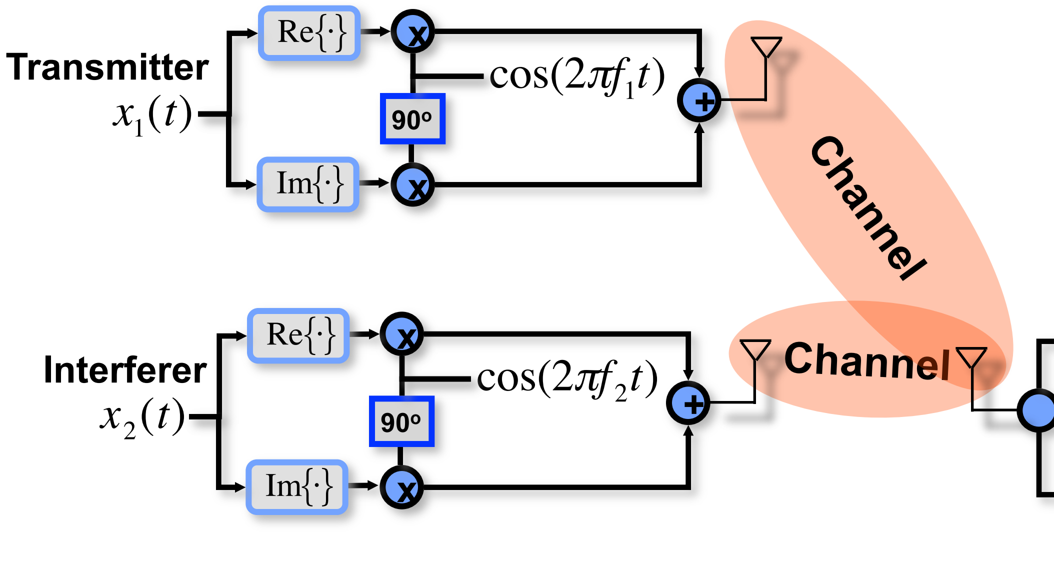
- Interferer operates at nearby center frequency
- Causes interference

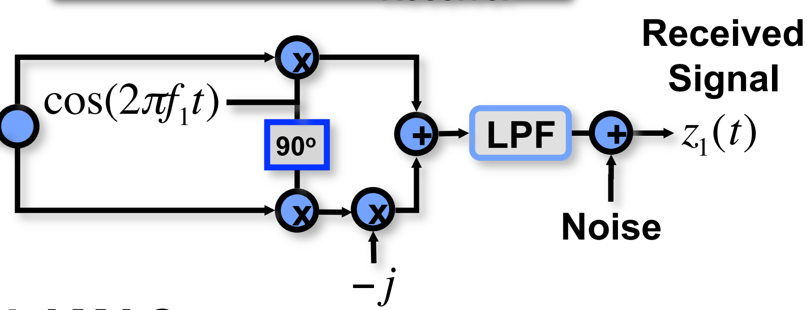


Signal Representation

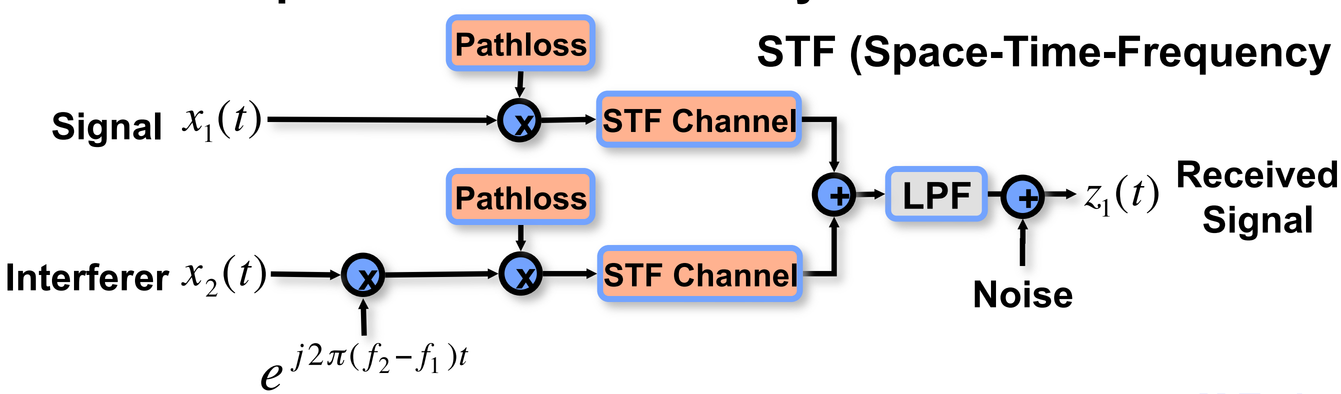
- **Passband Model**





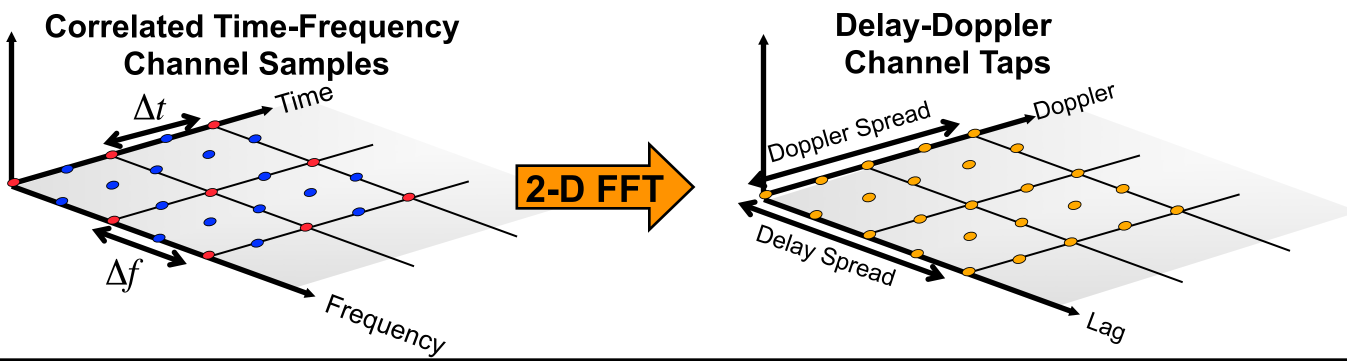


- **Baseband Equivalent Model Used by LLAMAComm**

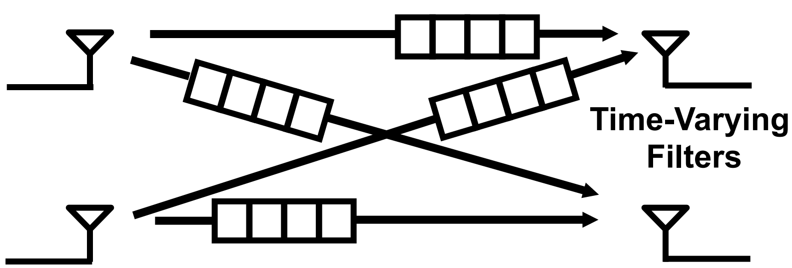


Channel Model Options

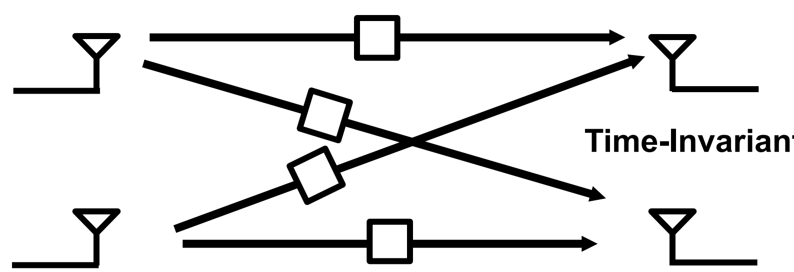
- ‘STFCS’ (Space-time-frequency-correlated scattering)
  - Preserves spatial correlation



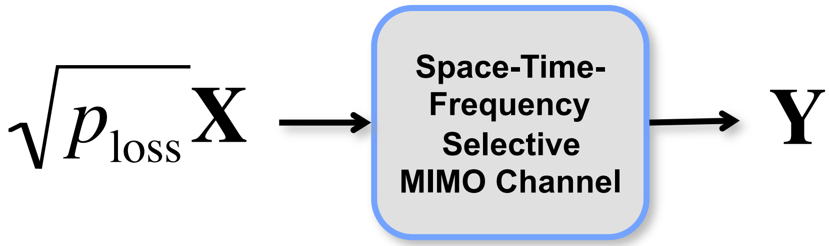
- ‘WSSUS’
  - Wide-sense stationary uncorrelated scattering
  - Exponential power profile



- ‘LOS_AWGN’
  - Line-of-sight path loss in AWGN channel
  - For debugging only



'STFCS' channel model option



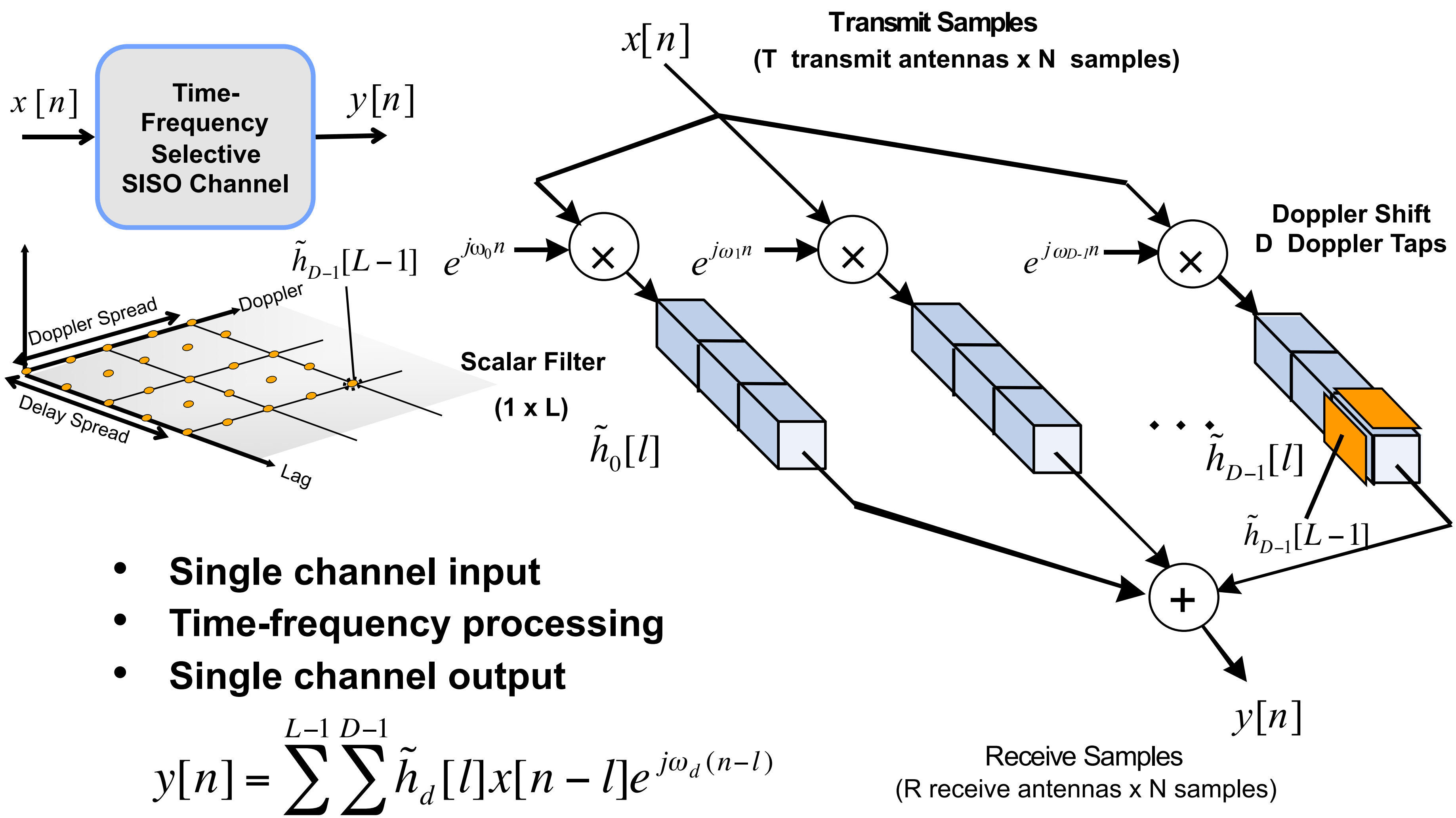
- 'STFCS' channel model was developed at Lincoln
  - Correlates signals over space, time, and frequency

- The STFCS channel tensor is four dimensional:

$$R \times T \times D \times L$$

- R receive antennas
- T transmit antennas
- D Doppler taps
- L lag taps

SISO Channel with Doppler and Multipath



- **Single channel input**
- **Time-frequency processing**
- **Single channel output**

$$y[n] = \sum_{l=0}^{L-1} \sum_{d=0}^{D-1} \tilde{h}_d[l] x[n-l] e^{j\omega_d (n-l)}$$

Spatially Correlated Static MIMO Channel

$$\sqrt{\rho_{\text{loss}}}\,\mathbf{X} \rightarrow \boxed{\mathbf{H}} \rightarrow \mathbf{Y} = \sqrt{\rho_{\text{loss}}}\,\mathbf{HX}$$

- Spatially correlated static MIMO channel:

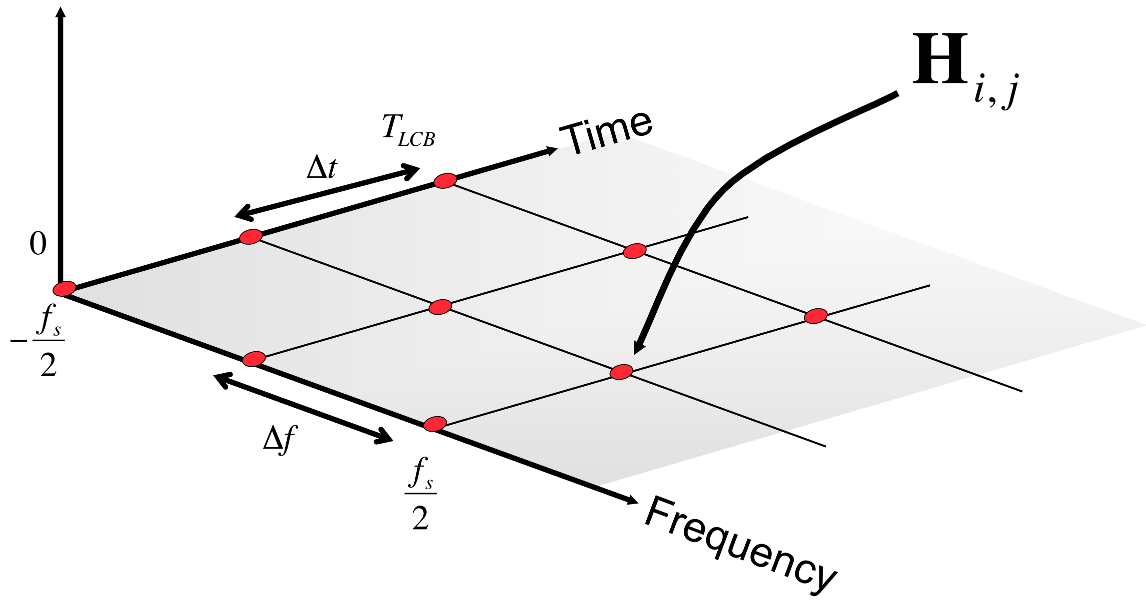
$$\mathbf{H} = \mathbf{U}\mathbf{A}_{\alpha}\mathbf{G}\mathbf{A}_{\alpha}\mathbf{V}^{H}$$

  - $\mathbf{G}$ is an R×T i.i.d. Gaussian matrix
  - $\mathbf{U}$ and $\mathbf{V}$ are random unitary matrices
  - α controls spatial correlation

$$\mathbf{A}_{\alpha} = \begin{pmatrix} \alpha^{0} & & \\ & \ddots & \\ & & \alpha^{n-1} \end{pmatrix} \qquad 0 \le \alpha \le 1$$

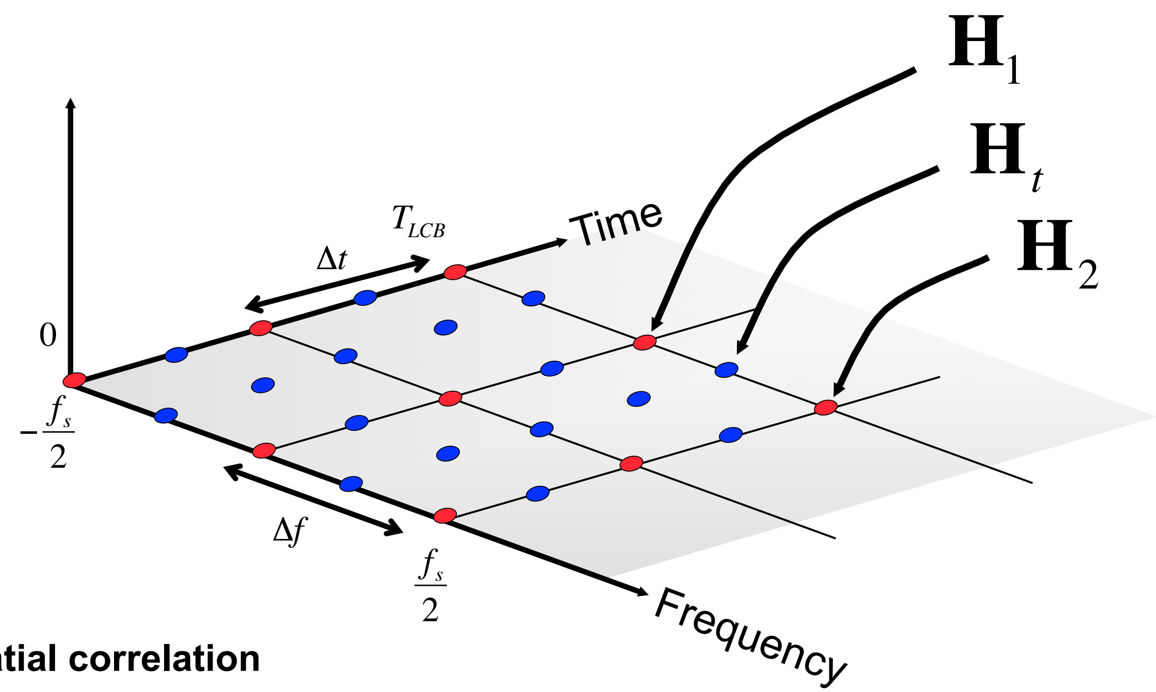
- What about MIMO channels with Doppler and Multipath?
  - Create time-frequency-space correlations

Sample the Time-Frequency Plane



- Δt coherence time
- Δf coherence bandwidth
- Different $\mathrm{H}_{i,j}$'s are uncorrelated with each other
- Each $\mathrm{H}_{i,j}$ is spatially correlated
- T_{LCB} is longest coherent block (environment property)

Interpolate the Time-Frequency Plane



- **Preserve spatial correlation**

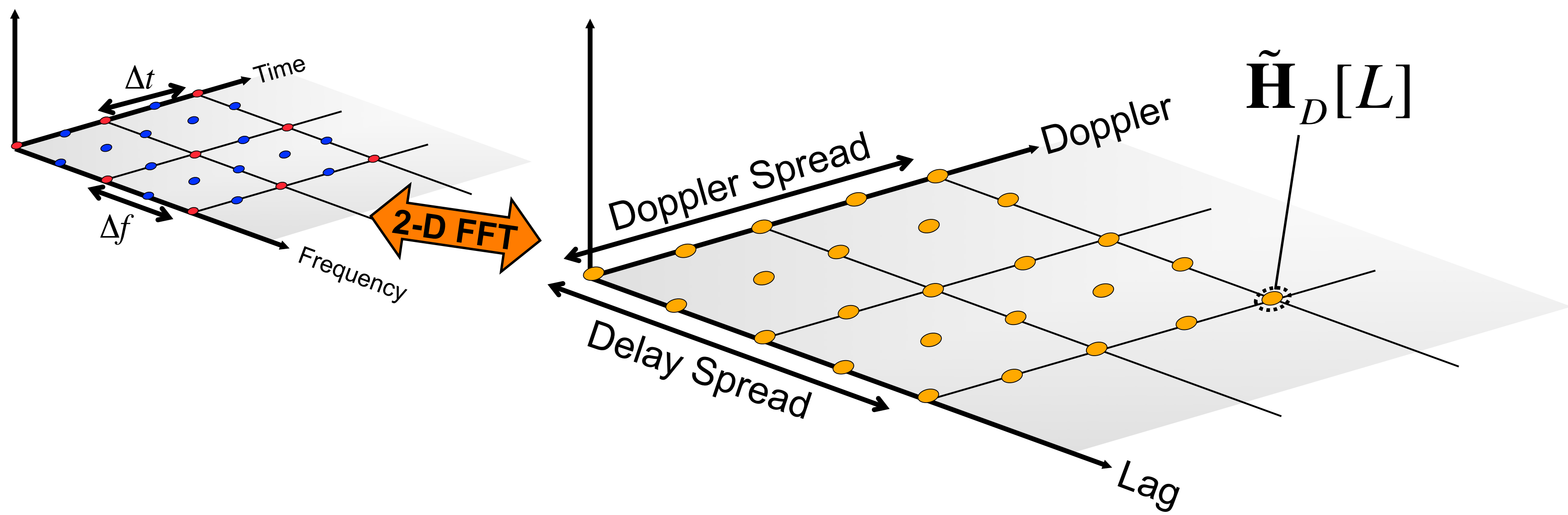
$$\mathbf{H}_1 = \mathbf{U}_1\mathbf{A}_\alpha\mathbf{G}_1\mathbf{A}_\alpha\mathbf{V}_1$$

$$\mathbf{H}_t = \mathbf{U}_t\mathbf{A}_\alpha\mathbf{G}_t\mathbf{A}_\alpha\mathbf{V}_t$$

$$\mathbf{H}_2 = \mathbf{U}_2\mathbf{A}_\alpha\mathbf{G}_2\mathbf{A}_\alpha\mathbf{V}_2$$

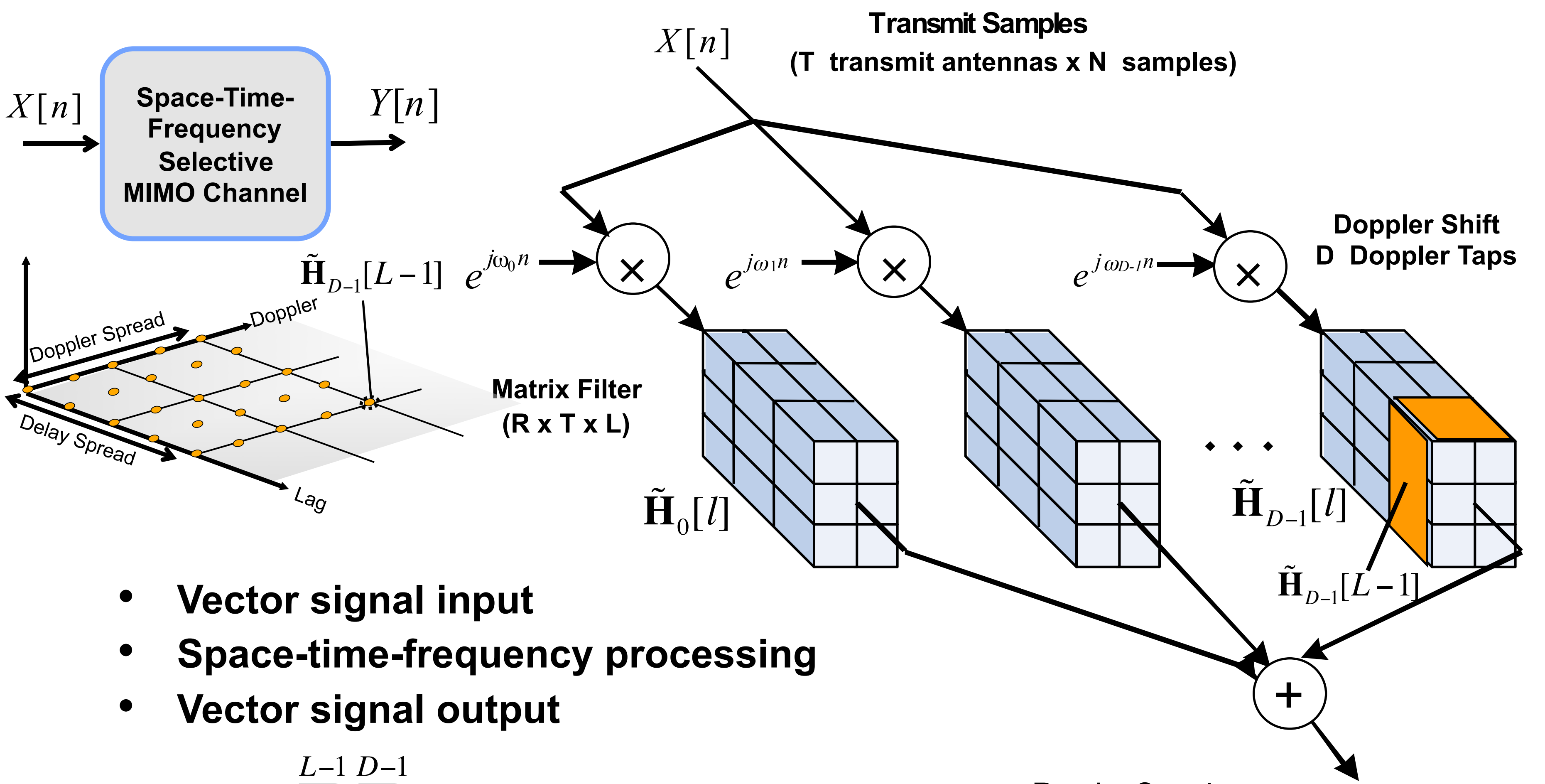
$$\longleftarrow \left\{ \begin{aligned} \mathbf{U}_t &= \mathbf{U}_1 e^{t \log \mathbf{U}_1^H \mathbf{U}_2} \\ \mathbf{G}_t &= \sqrt{1-t^2}\,\mathbf{G}_1 + t\mathbf{G}_2 \\ \mathbf{V}_t &= \mathbf{V}_1 e^{t \log \mathbf{V}_1^H \mathbf{V}_2} \end{aligned} \right.$$

Convert to Doppler-Lag Plane



- **Take 2-D FFT over each spatial component**
  - **Time -> Doppler**
  - **Frequency -> Lag**

MIMO Channel with Doppler and Multipath

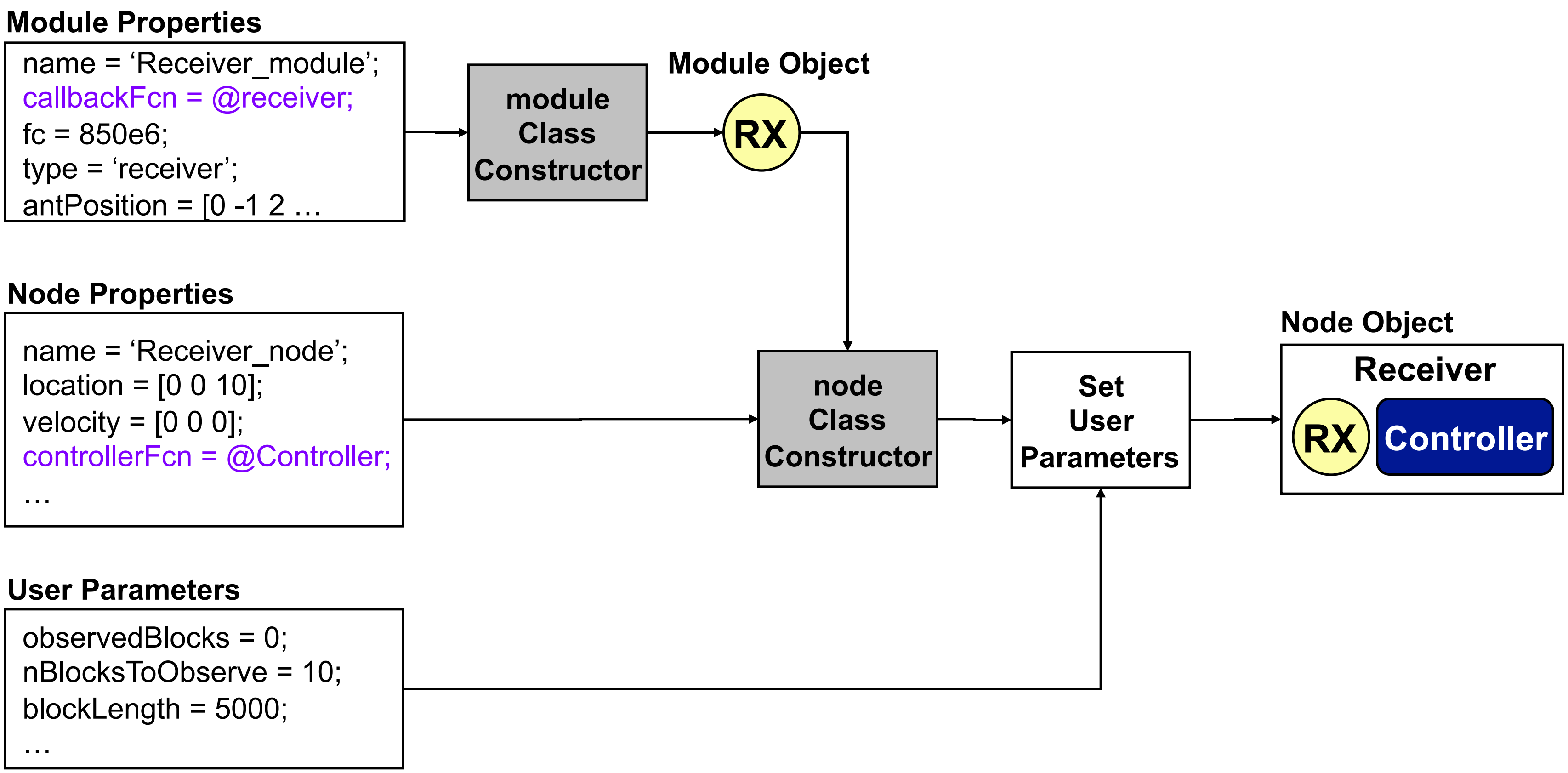


- Vector signal input
- Space-time-frequency processing
- Vector signal output

$$Y[n] = \sum_{l=0}^{L-1} \sum_{d=0}^{D-1} \tilde{\mathbf{H}}_d[l] X[n-l] e^{j\omega_d (n-l)}$$

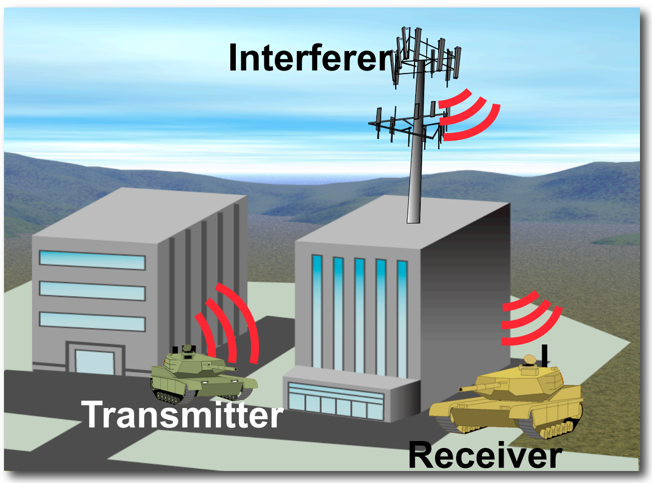
Building a Node Object in LLAMAComm

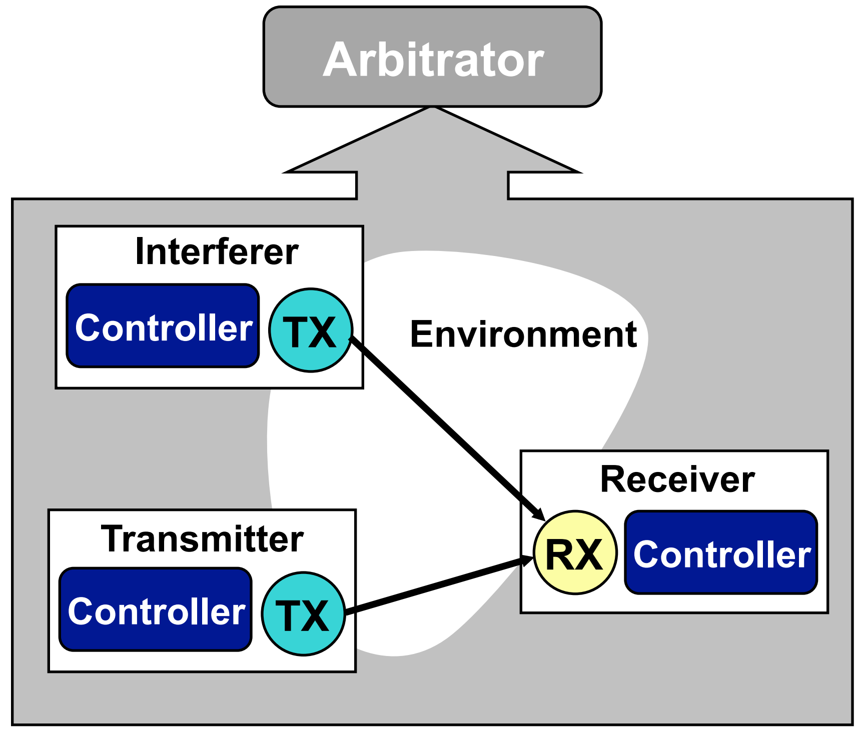
- Example: building a receiver node
- Note: A node can have any number of modules



Simulation Procedure

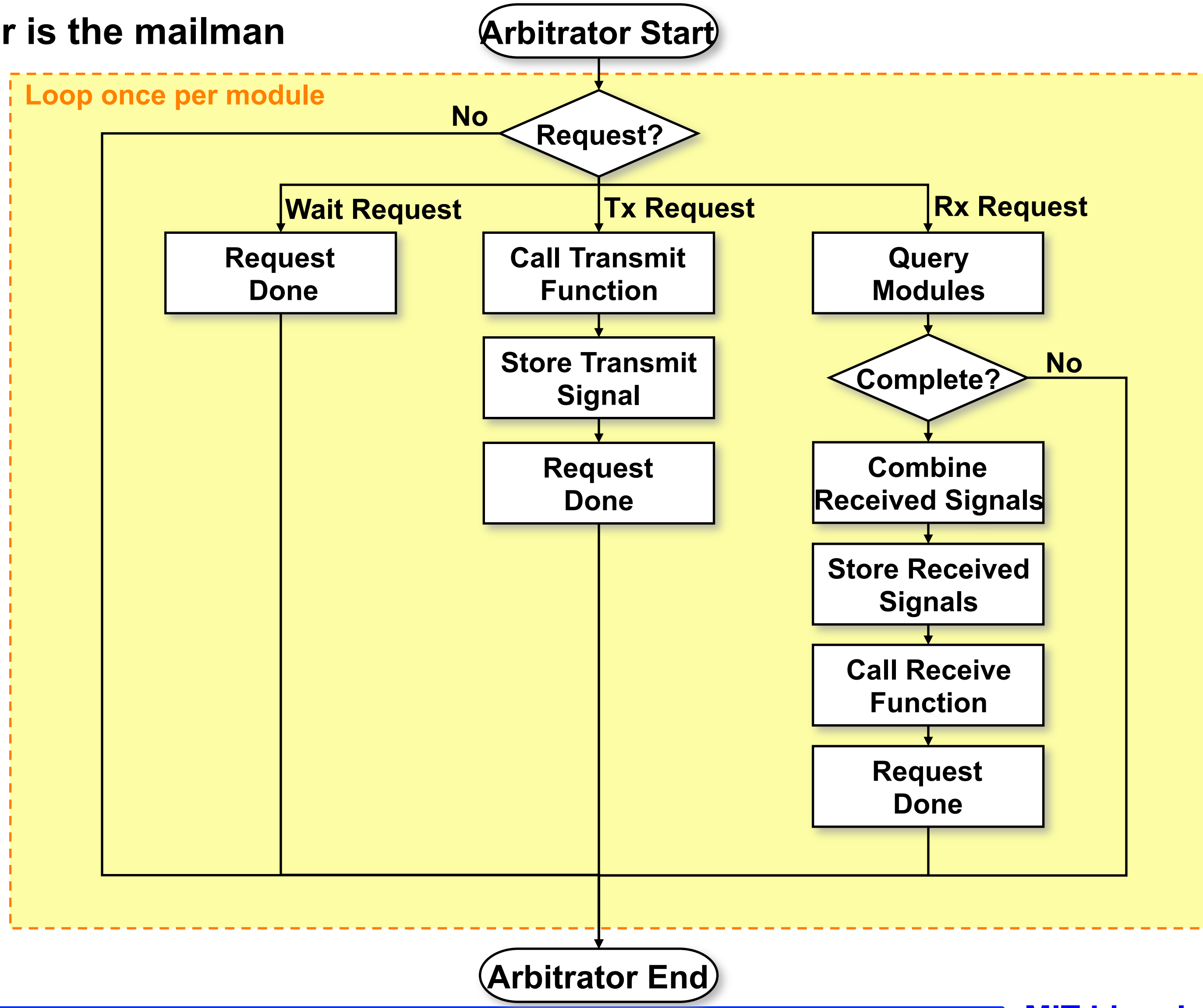
1. Controllers request to transmit and receive
2. Arbitrator enforces causality
3. Go to step 1.





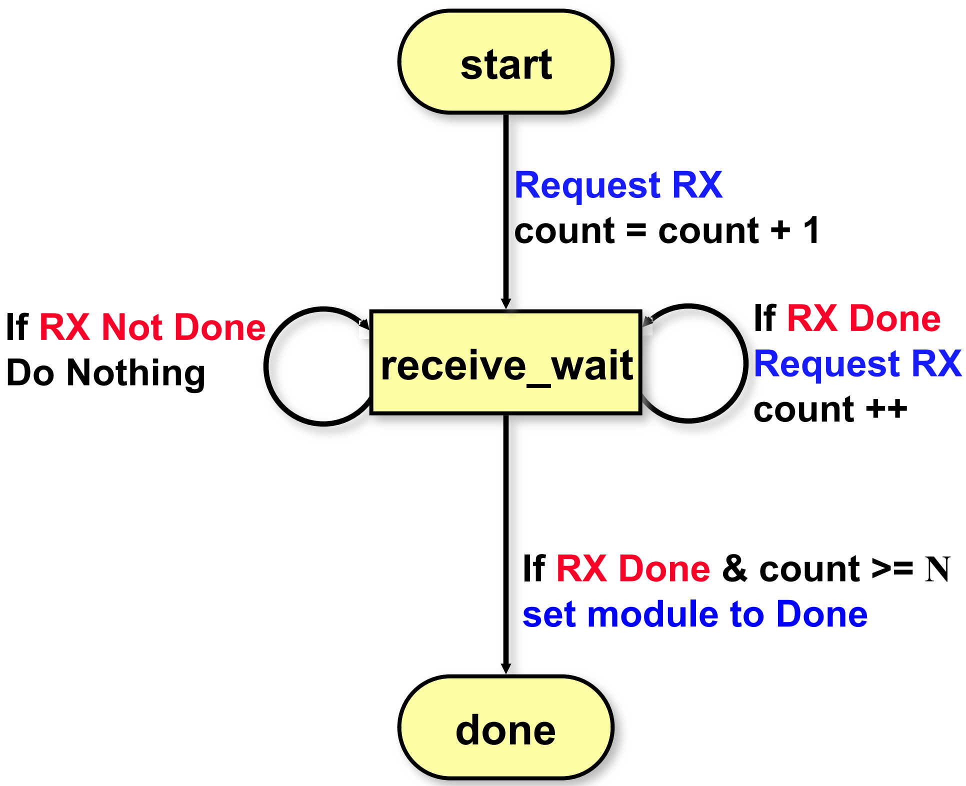
Flowchart of the Arbitrator

Arbitrator is the mailman

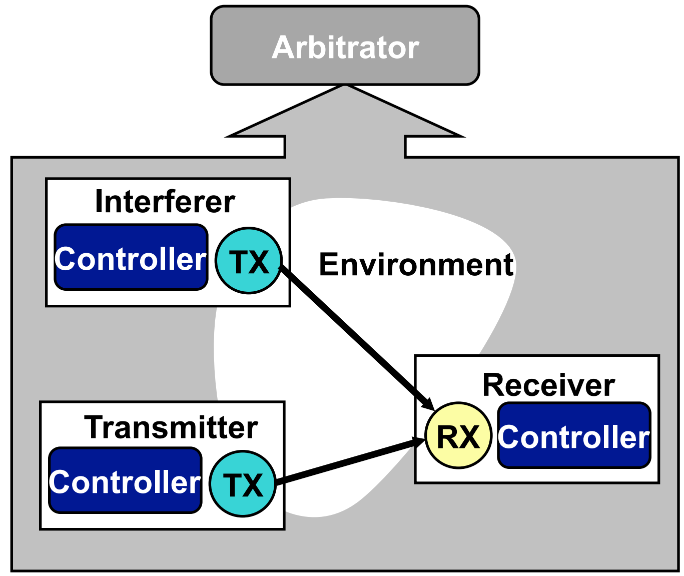


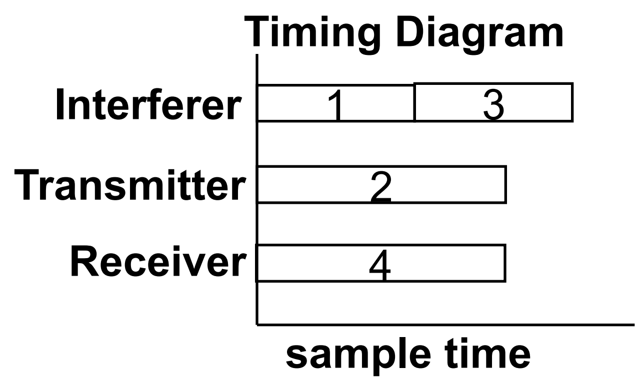
Receiver Controller State Machine

Text Color Key:
Red: Input from Arbitrator
Blue: Request
Black: Inline Code



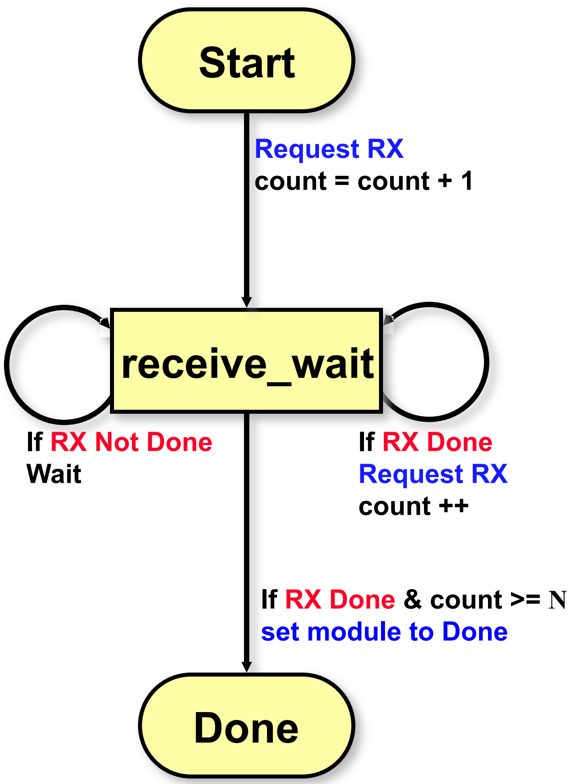
Functionality:
Receive N sample blocks





Receiver Controller Matlab Code

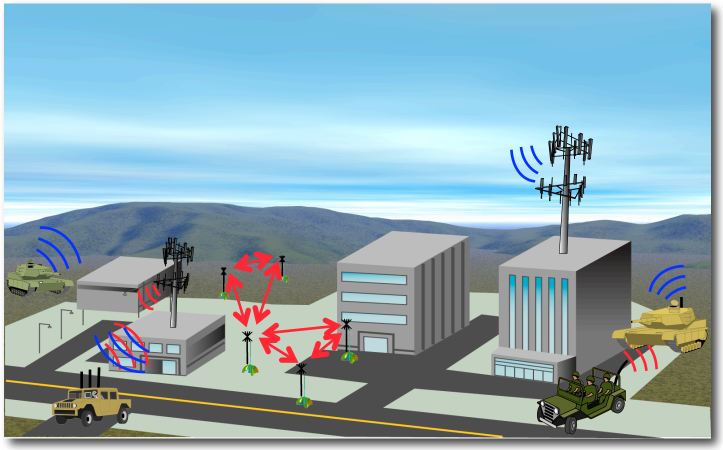
Text Color Key:
Red: Input from Arbitrator
Blue: Request
Black: Inline Code



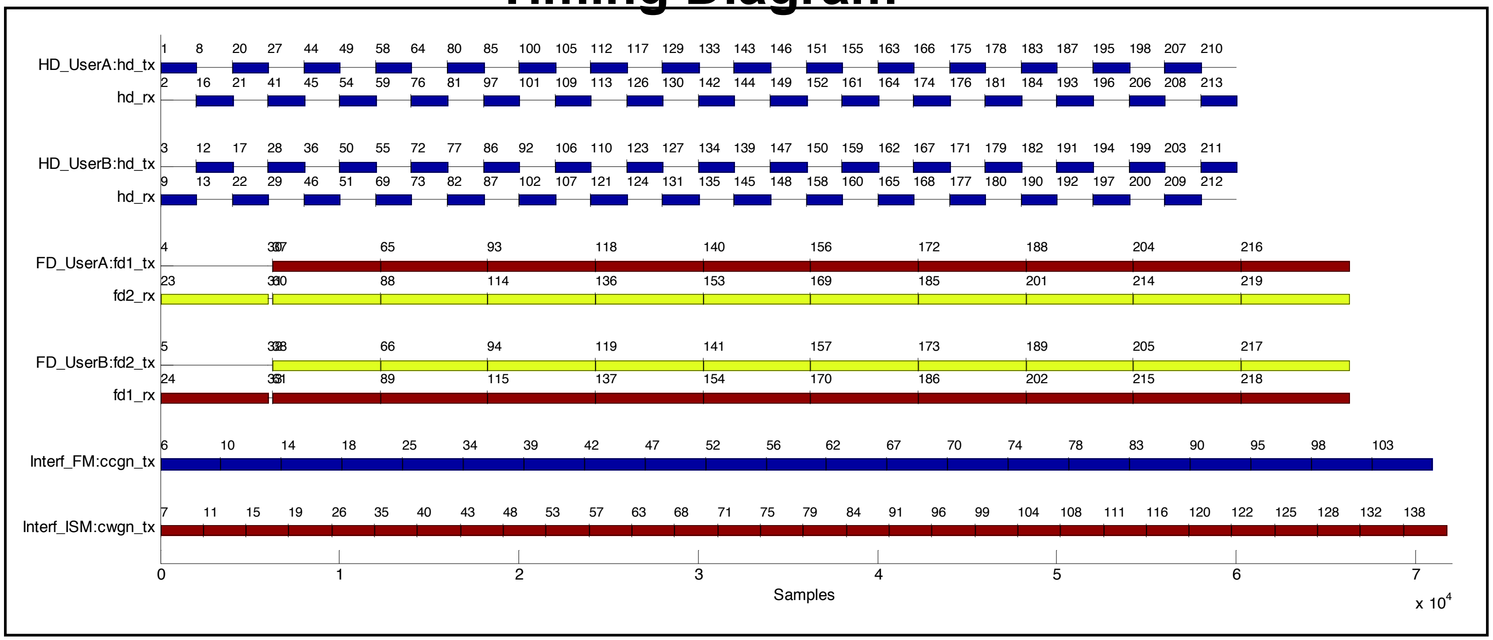

```
p             = GetUserParams(nodeobj);        % Load the user parameters
currState     = GetNodeState(nodeobj);         % Load the current state
switch currState
   case 'start'
      nodeobj = SetModuleRequest(nodeobj,'observer_rx','receive',p.blockLen);
      p.observedBlocks = p.observedBlocks + 1;
      nextState = 'receive_wait';

   case 'receive_wait'
      requests = CheckRequestFlags(nodeobj); % Get request flag
      if requests==0 % if RX Done
         if p.observedBlocks>=p.nBlocksToObserve;
            nodeobj = SetModuleRequest(nodeobj,'observer_rx','done');
            nextState = 'done';
         else % Receive another block and increment counter
            nodeobj ...
               = SetModuleRequest(nodeobj,'observer_rx','receive',p.blockLen);
            p.observedBlocks = p.observedBlocks + 1;
            nextState = 'receive_wait';
         end
      else % Wait until block has been received
         nextState = 'receive_wait';
      end

   case 'done'
      nextState = 'done'; % Continue in Done state
      status = 'done';
end
nodeobj = SetNodeState(nodeobj,nextState);     % Set the next state
nodeobj = SetUserParams(nodeobj,p)             % Set the user parameters
```

Timing Diagram

- **LLAMAComm produces a timing diagram to simplify debugging**
- **Each segment represents a transmit or receive sample block**
- **Thin segments represent ‘wait’ blocks**

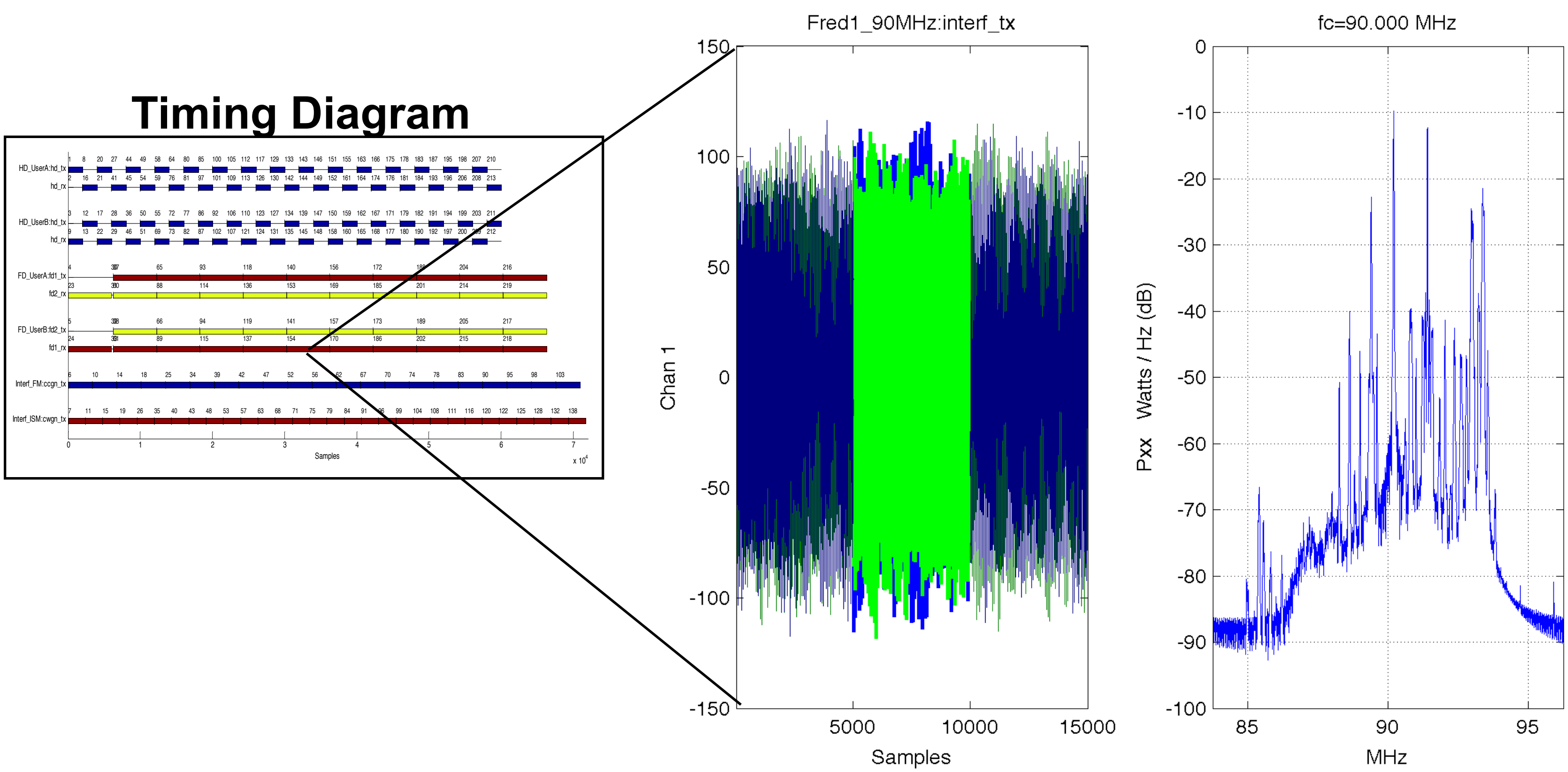


Timing Diagram



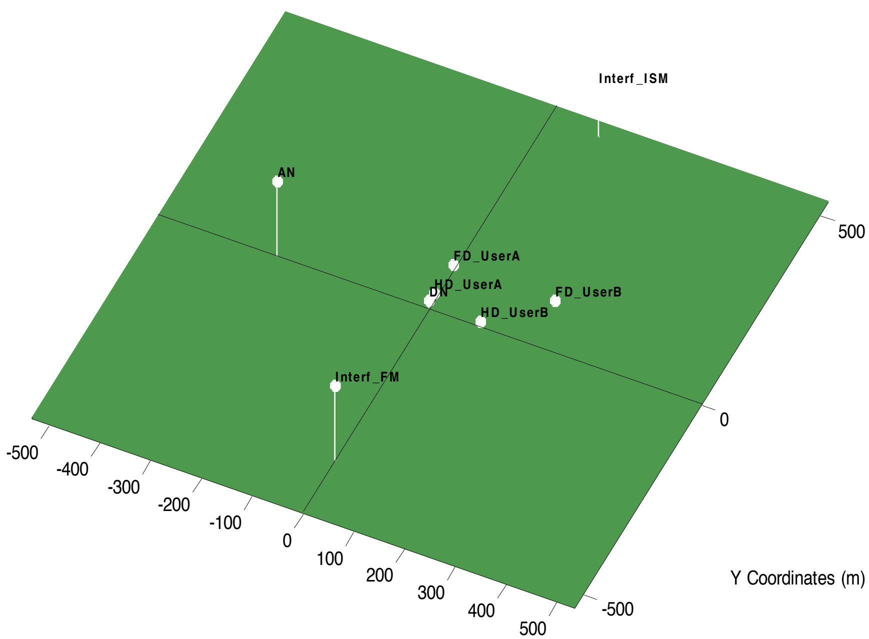
Timing Diagram Callback Plots

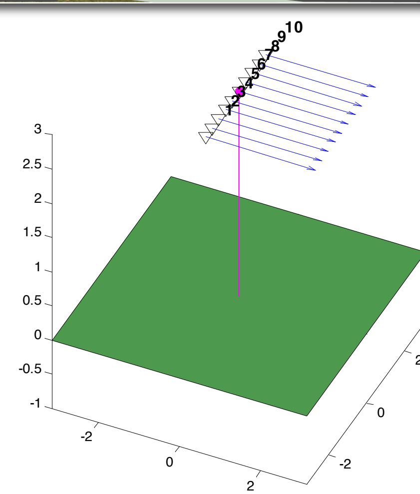
- Clicking on a segment produces a plot
- Plots can be customized by the user



Node Position Map

- **LLAMAComm plots a map of node positions.**
- **Clicking on a node name will produce a plot of the antenna positions and look directions.**





Future for LLAMAComm

- Ground-to-Air link path loss model update
- Nodes in Motion (long time scales)
  - Time-varying path losses
- Enhanced radio library
  - Open source model?
- Parallelism
  - Parallel processing to reduce simulation time
- Special interference modules that generate samples only when overlapping in frequency with a receive request
- Allow dynamic sampling rates