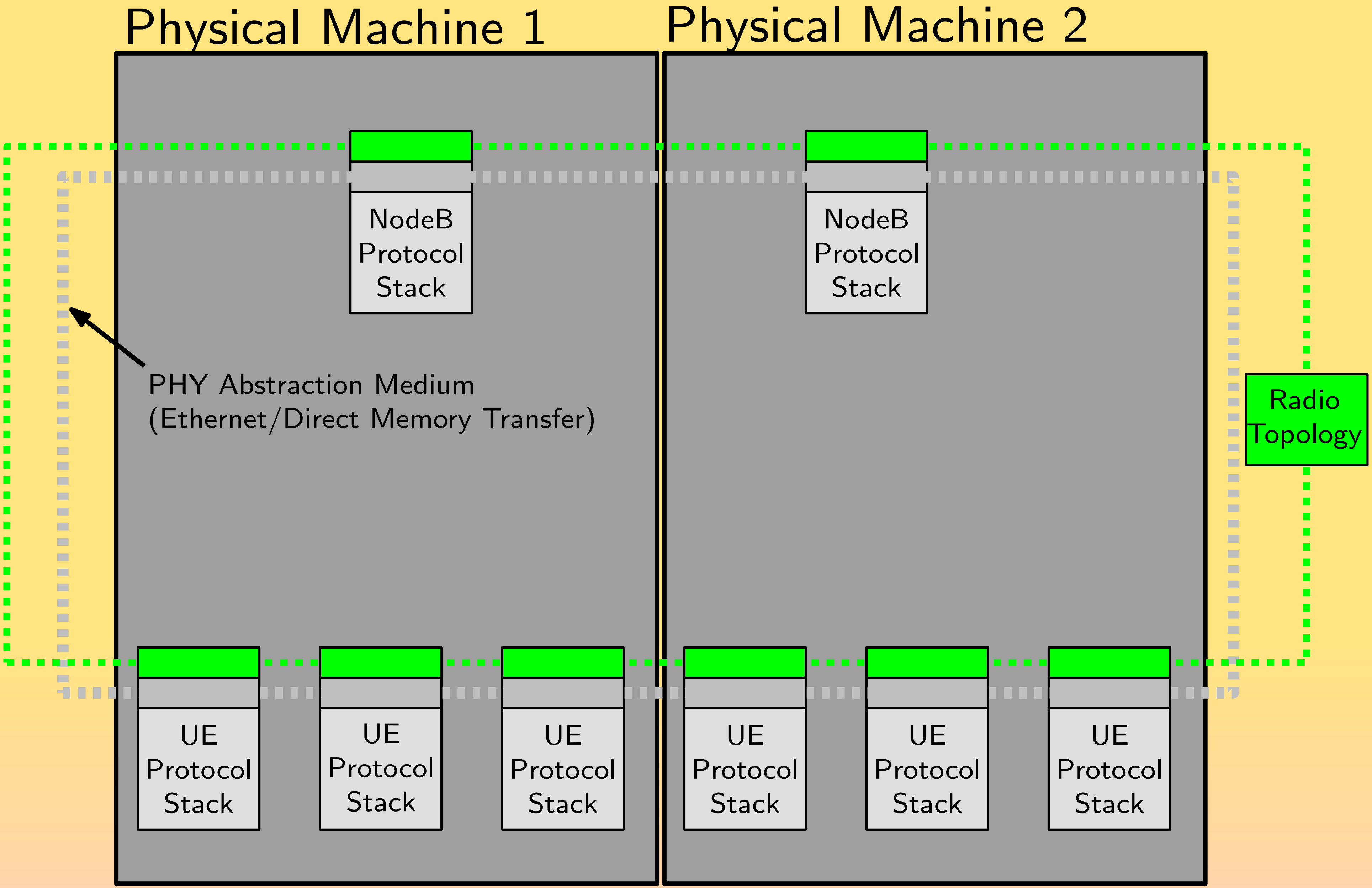
OpenAirInterface Emulation Methodology

- Real-time distributed validation environment comprising
  - IP Multicast over Ethernet: PHY emulation layer
  - PHY behavioural abstraction models
  - OpenAirInteface Layer two real-time protocol stack (**openair2**) potentially virtualized into N_{inst} instances in the same physical machine.
- Virtualization of protocol stacks (L2 + L3) in one machine

- Two modes
  - Hard Real-time with virtualization (RTAI/Linux)
  - Soft Real-time (maintain frame timing on average) in Linux user-space

OpenAirInterface Protocol Validation Environment

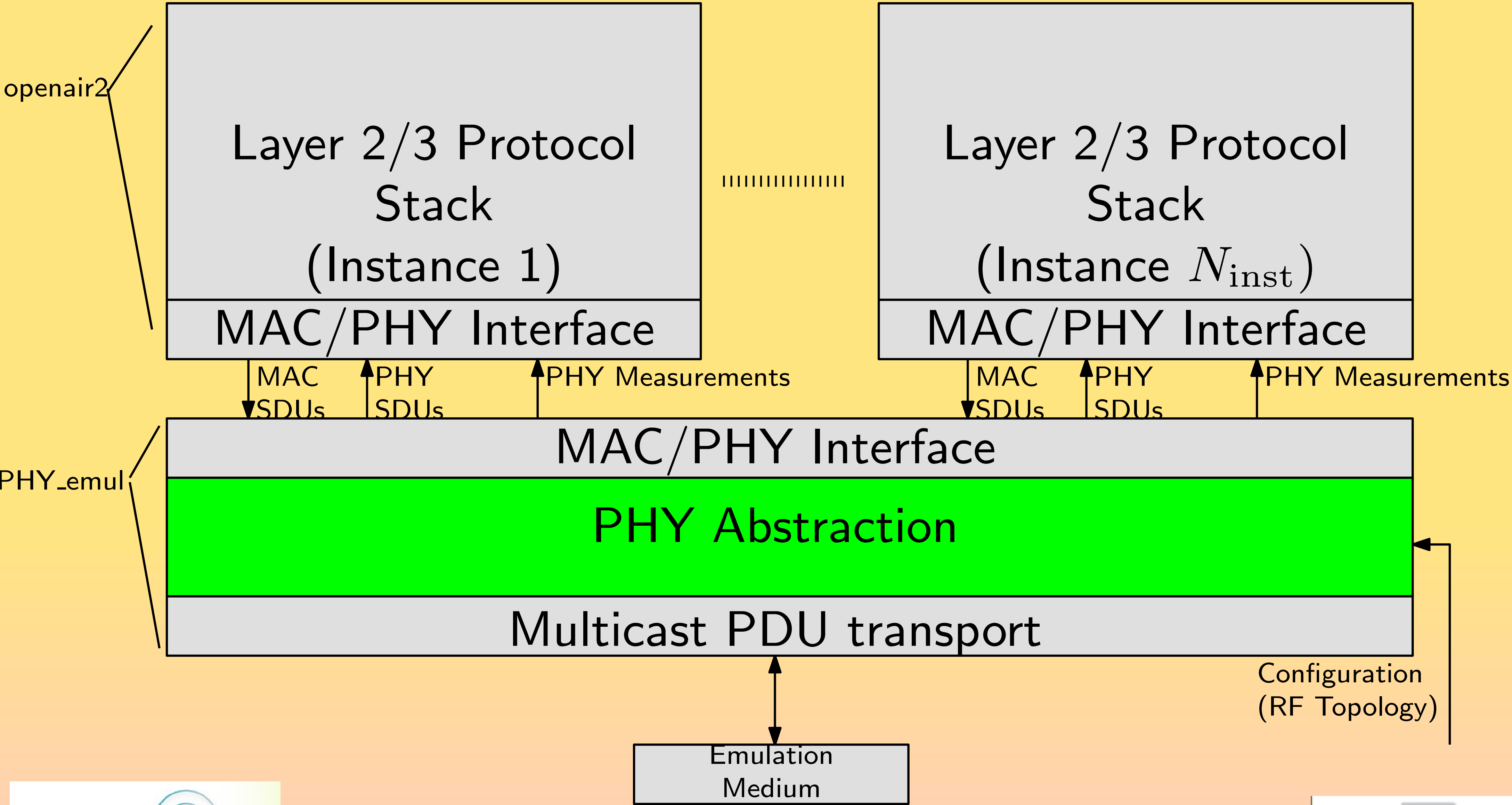


Uses

- Protocol Implementation Validation
  - This enables developers of L2/L3 and applications to test their implementation in a real-time setting without the need for stable RF equipment
  - This can include completely repeatable experiments
- System Performance Evaluation
  - For L2/L3 protocol assessment using accurate PHY abstraction models
  - Possibility of using real channel measurement traces as stimulus
  - Measurement collection for modelling of L2

- Each node in the emulated network (either on a distinct PC or in a different MAC instance within one PC) has a kernel module responsible for PHY emulation
  - Fully implements the MAC/PHY interface (Cellular or AdHoc)
  - Raw MAC PDUs are transported via IP Multicast between correspondent nodes
  - Generates measurements as would the real PHY
  - An PHY abstraction simulation module can inject simulated error patterns in the different MAC-layer streams. This is configured dynamically by a radio topology server or locally

Components



openair2
Layer 2/3 Protocol Stack (Instance 1)
MAC/PHY Interface
Layer 2/3 Protocol Stack (Instance N_{inst})
MAC/PHY Interface
MAC SDUs
PHY SDUs
PHY Measurements
MAC SDUs
PHY SDUs
PHY Measurements
PHY_emul
MAC/PHY Interface
PHY Abstraction
Multicast PDU transport
Configuration (RF Topology)
Emulation Medium

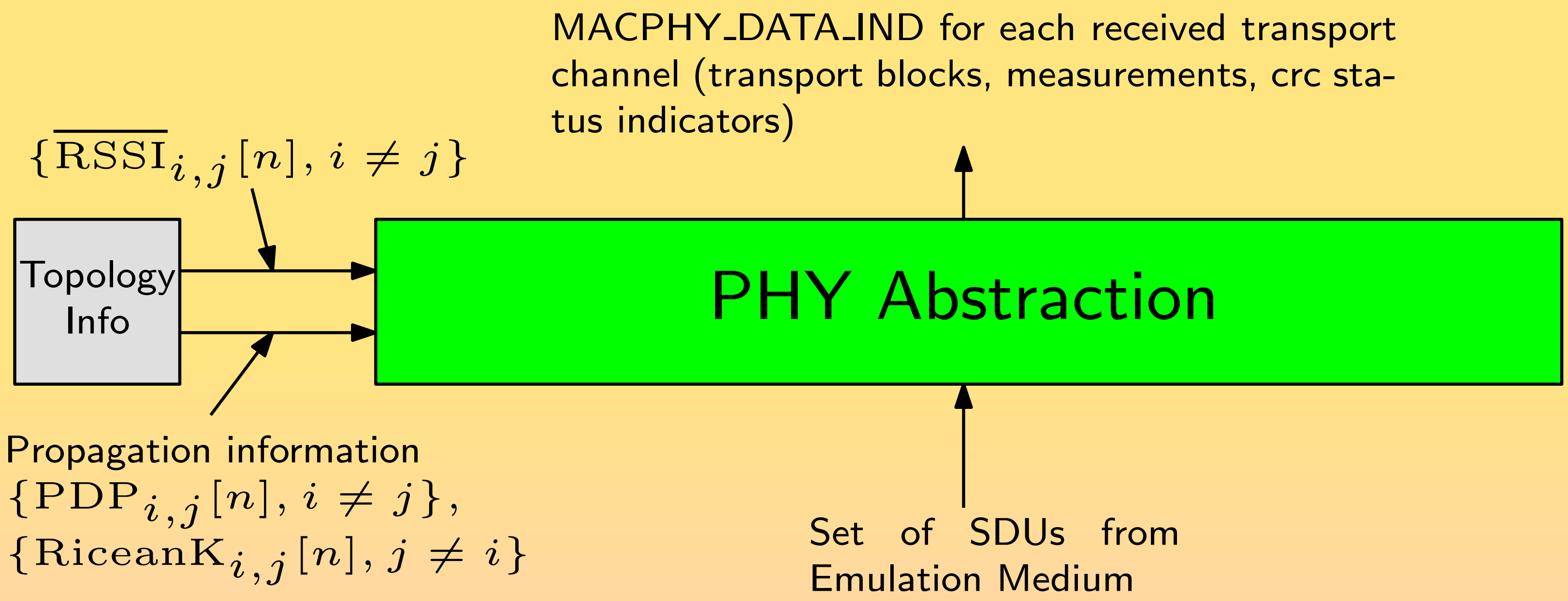
PHY Abstraction

- PHY simulation is done at the receiver of each node. Wide-band SINR are computed every transmission frame based on the RF topology and pre-defined propagation models
  - The function can be system dependent (i.e. based on precomputed probability of error simulations for specific modulation and coding formats) or generic
- The output of the radio simulation is random packet loss indicators for each transport channel block traversing the PHY/MAC interface. Alternatively, if erroneous packets are to be passed to the higher layers (in the spirit of UDP lite-type protocols), bit errors must be generated in the MAC PDUs.

- In each radio frame (TTI), the PHY_Abstraction unit analyzes the set of received SDUs from the emulation medium and determines those which are sources of information and those which represent interference
  - The targets to be received are the programmed by the MAC in the MACPHY_DATA_REQ primitive as with the true PHY.
  - The interferers are naturally present with the true PHY and thus their impact must be simulated in the abstraction unit. Since a particular node in the network is not aware of all sources of interference *a priori*, this is done by adding a PHY_RESOURCES description to each transport block in the emulation medium which is not present in the real PHY.

PHY Abstraction : Example

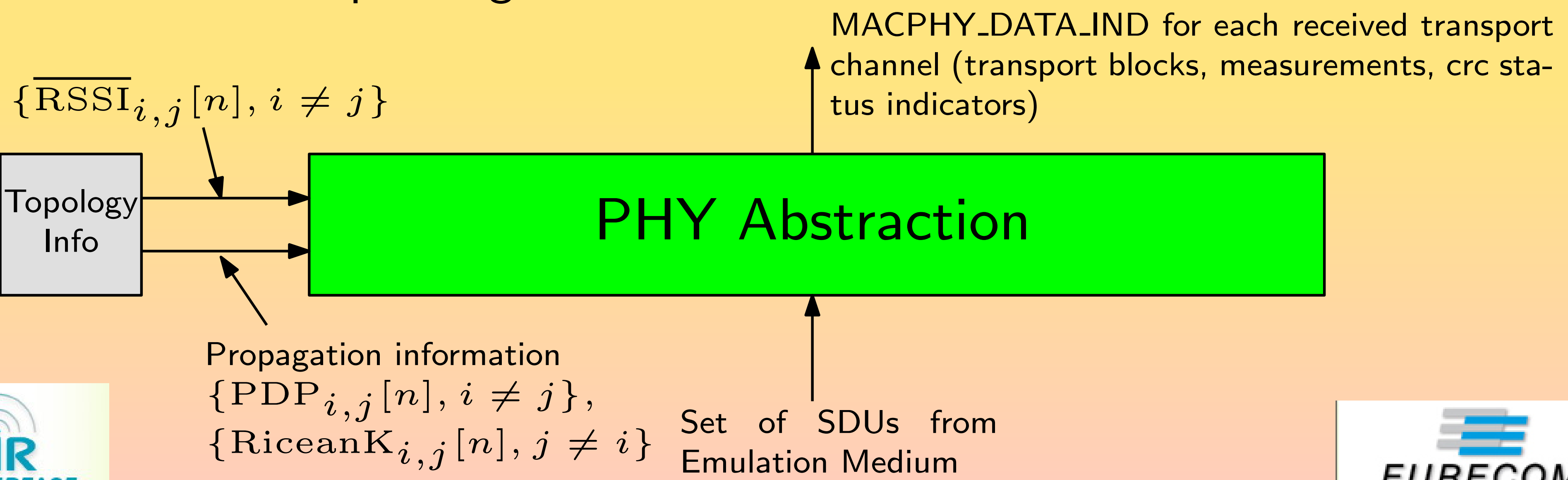
- Consider the following example for PHY abstraction at node j in the network
- Let $\overline{\mathrm{RSSI}}_{i,j}[n]$ be the average received strength in frame (TTI) n between node i and node j. This can be generated locally in each node based on a model for mobility or can be signalled by a topology server dynamically.



PHY Abstraction : Example

- The goal of the PHY_abstraction entity is to simulate the block error rate process (BLER) of each transport block of a particular received resource

- This is fundamentally related to the statistics of the received signal and interference vectors at node j. Let $\sqrt{\overline{\mathrm{RSSI}}_{i,j}[n]}\mathbf{H}_{i,j}[n,k]$ be the spatial channel in frequency band k for the signal from node i to j and $K_{\mathrm{I},i,j,m}[n,k] = \sigma^2 + \sum_{i' \neq i, i' \neq j} \sum_{m=0}^{M(i')-1} \sqrt{\overline{\mathrm{RSSI}}_{i',j}}[n]\mathbf{h}_{i',j}[n,m,k]\mathbf{h}_{i',j}[n,m,k]^*$, where $\mathbf{h}_{i',j}[n,m,k]$ is the spatial channel for transmit antenna m in band k corresponding to interferer i'. $M(i')$ is the number of transmit antennas corresponding to interferer i'.



- Generation of the random variables $\mathbf{H}_{i,j}[n,k]$ and $\mathbf{h}_{i',j}[n,m,k]$ depend on the space/time/frequency description of the propagation environment (PDP,Ricean factor,antenna correlation, mobility)
- A description of the BLER as a function of $\sqrt{\mathrm{RSSI}_{i,j}[n]}\mathbf{H}_{i,j}[n,k]$ and $K_{\mathrm{I},i,j,m}[n,k]$ is required.

PHY Abstraction : Example

- Description of the conditional BLER is the key issue. This can be done in a variety of ways

  1. for non-ARQ based coding and no transmit filtering (beamforming, dirty-paper, etc.)
     - for a particular receiver structure SINR expressions can be derived from the above and used for BLER lookup based on tabulated performance of a particular code.
     - In addition, PHY agnostic information-theoretic bounds can also be derived
  2. For ARQ-based schemes, errors at a particular time n also depend on past values of the signal and interference components. Here additional protocol information from the MAC signaling header are required, but similar semi-analytical models can be used.
  3. For transmit filtering (beamforming, dirty-paper, etc.) additional PHY layer information must be transported in the emulation process along with MAC information, namely the linear/non-linear spatial filtering description at the sending nodes.

PHY Abstraction : Example

- Type 1 Generic Model for BLER
  - Under the assumption of Gaussian transmit signals and a particular receiver structure (i.e. MMSE which is optimal for Gaussian statistics) a model for achievable BLER (on each transport block sent by PHY to MAC) could be bounded

$$P_e\left(\left\{\sqrt{\mathrm{RSSI}_{i,j}[n]}\mathbf{H}_{i,j}[n,k]\right\},\quad\left\{K_{\mathrm{I},i,j,m}[n,k]\right\}\right)\leq K_{\mathrm{impl}}e^{-N_{\mathrm{TB}}(R(n)-R_{\mathrm{MAC}}(n))}$$

where K_{impl} is an implementation degradation factor, $R_{\mathrm{MAC}}(n)$ is the allocated rate by the MAC layer scheduler in TTI n and

$$R(n)=\frac{1}{|\mathcal{A}(n)|}\sum_{f\in\mathcal{A}(n)}\ln\left(1+\mathrm{SINR}_j[n,f]\right)$$