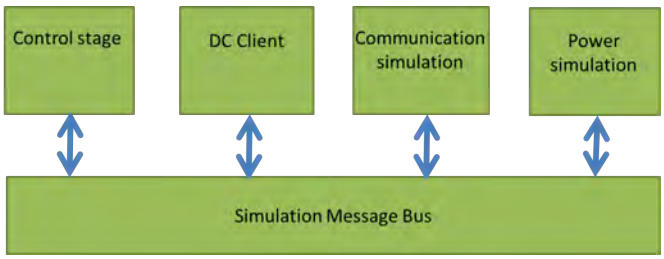
# Implementation of Communication Simulation in Co-simulation of Power- and Communication-Networks

Pavlos Dimitriou, dimitriou@ict.tuwien.ac.at
Mario Faschang, faschang@ict.tuwien.ac.at

## Problem Statement

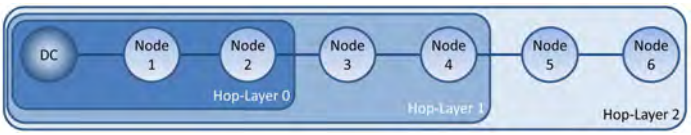
- Challenge: Simulation of Power Line Carrier (PLC) communication using the Automated Metering and Information System (AMIS) protocol.
- Implementation of the communication simulation as a component in a co-simulation schema with power simulation.

## Co-Simulation Overview



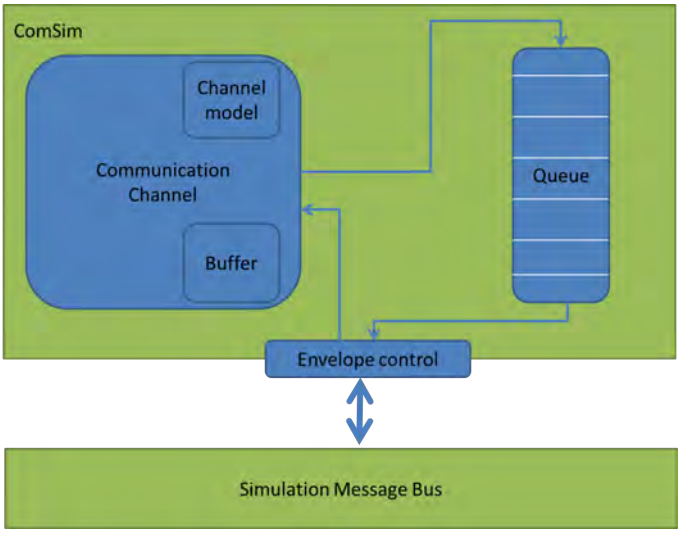

- Simulation Message Bus: co-simulation middleware, responsible for packet routing, synchronization, configuration loading, error logging and much more.
- Power simulation: Digsilent PowerFactory performs numerical simulation of power distribution network, steady state and transient analysis.
- Communication simulation: responsible to simulate the communication protocol AMIS.
- AMIS Data Concentrator (DC) :
  - Express Grid Data Acquisition (EGDA) client: simulates the DC. Sends cyclically queries to nodes in the power simulation network and receives their voltage/power values.
  - Modbus: simulates the communication between the DC and the on load tab changer which uses different modbus actuators.
- Control Stage : responsible to analyse data from DC and based on different active control algorithms to react on the tab changer.

## AMIS Protocol




- Automated Metering and Information System (AMIS).
- Master / Slave protocol. DC is the master, smart meters the slaves.
- Every node receiving a packet repeats it simultaneously for a defined number of times. Retransmissions create hop layers.
- Two parameters are used:
  - Delay time ($T_{delay}$).is the time needed for every packet to do the route DC-Node-DC. Recorded in milliseconds.
  - Loss probability ($P_{loss}$), decides if a packet is lost. When lost, a maximum $T_{delay}$ is used depending on transmission mode.

## Communication Simulation Implementation




❖ Communication Channel:

- EGDA client sends cyclically "Load Request" packets to the power simulation and receives "Load Updates". Both packet types pass first through the communication simulation
- When "Load Request" 2 pairs of values is created: $T_{delay}$ and $P_{loss}$. First pair is added to envelope payload, second pair is saved in an internal queue. $T_{delay}$ is added to the general co-simulation time called "delay_until".
- In emulation the ComSim actually stops and waits until the delay_until time whereas in simulation the delay_until time is written in payload and the ComSim process next envelope as soon as possible.
- When "Load Answer" the already produced $T_{delay}$ and $P_{loss}$ pair is read from internal queue, added in payload and forwarded to SMB.

❖ Channel Model:

- Creates $T_{delay}$ and $P_{loss}$ pairs. With use of statistical methods actual AMIS data from smart meters are recreated,
- The more actual AMIS data used for the statistical methods, the more accurate the model.

❖ Queue:

- Ensures that envelopes leave ComSim in an ascending "delay_until" time.
- Delays the envelopes/packets of the co-simulator when in emulation mode by actually stopping and waiting until delay_until time is reached.

## Acknowledgment

The research project "DG DemoNet - Smart LV Grid" is funded within the program "Neue Energien 2020" by the Austrian "Klima- und Energiefonds".