



Performance Evaluation of Wireless Networks Using the *open***WNS**

Sebastian Max

ComNets Research Group @ RWTH Aachen University
Germany



Open Wireless Network Simulator (*open*WNS)



- Wireless
 - Models for the wireless channel:
 - Pathloss
 - Shadowing
 - Fading
 - SINR computation
 - Models mobile nodes
- Network
 - Build a network of communicating nodes
 - Support multiple different node types
 - Support different network protocols
- Simulator
 - Build models to evaluate real systems (present and future)
 - Event-driven
 - Stochastic

What it is and what it's not

What **openWNS** is

- Dynamic Event Driven System Level Simulation Platform (C++, Python)
 - Investigations of dynamic protocol behavior
 - Cross-layer effects
 - Online calculation of interference
- Full fledged protocol stacks (FUN-based)
 - **UDP/TCP/IP**
 - **IEEE 802.11**
 - **IEEE 802.16**
 - **Future releases: 3GPP LTE, WiMedia**
- Typical Results
 - Protocol level results
 - e2e Packet Delay, Throughput
 - Buffer Fill Levels
 - Retransmissions
 - BER, PER, FER
 - Physical layer results
 - SINR distributions (over area, per terminal, per cell)

What **openWNS** is **NOT**

- Radio planning tool with ray tracing capabilities covering large scenarios of several 100 km²
 - Tool to design and run protocol stacks on an FPGA
 - Monte-Carlo Simulator
- But since it is open source, you never know ...

Challenges for the **openWNS** (and other simulators)

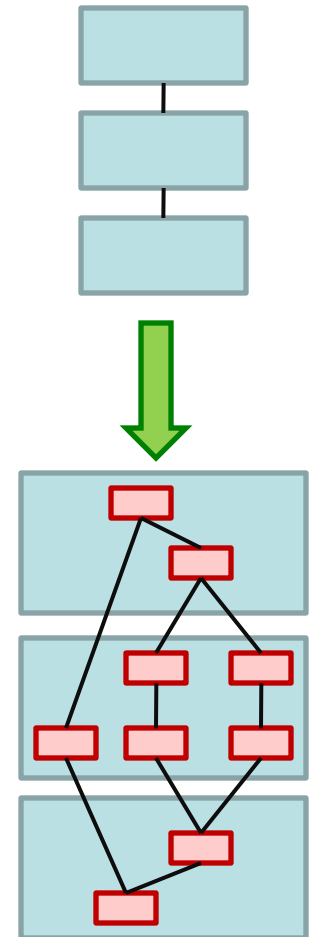
1. Future radio access technologies
 - Model new physical layer enhancements
 - Implement Medium Access Control improvements
 - Systematic protocol stack development with improvement- and extension-friendly design

2. Complexity
 - Large evaluation guidelines (e.g. IMT-Advanced, WiMAX)
 - Huge parameter space
 - Channel, physical layer, traffic models
 - Continuous & comprehensive validation

3. Scenarios
 - Large-scale simulations (e.g. 20.000 user terminals)
 - Precise protocol and wireless channel modeling
 - User-friendly management of large simulation campaigns

Protocol Stack Development

- **Classic**
 - Monolithic implementations of protocol stacks
 - Coarse grained selection of protocol functionality
 - Distinct implementations of the same functionality in different protocol stacks (reinvented wheel)
- **Required**
 - Fine grained selection of protocol functionality
 - Reusable elements
 - Flexible composition



Protocol Stack Development cont'd

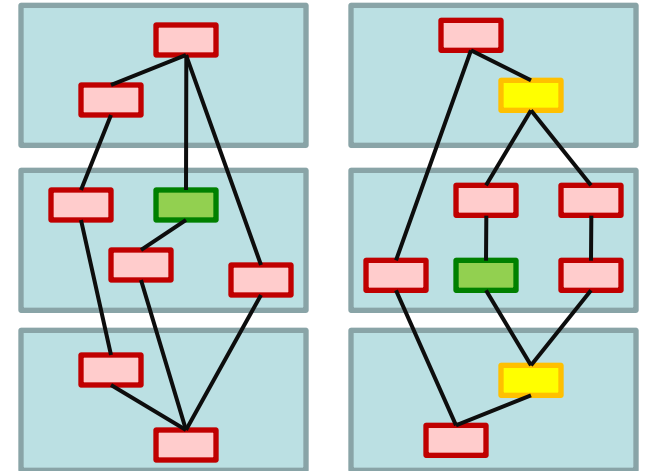
- A lot of commonalities can be observed
 - In the different layers of the same protocol stack
 - Between different protocol implementations

For example:

- Automatic Repeat Request (ARQ)
- Scheduling
- Flow Control
- Flow Separation (e.g. per user)
- Multiplexing and Demultiplexing

➔ Share a common code base of protocol functionality to ...

- ... ease / speed up the implementation of new protocols
- ... increase the quality of protocol implementations

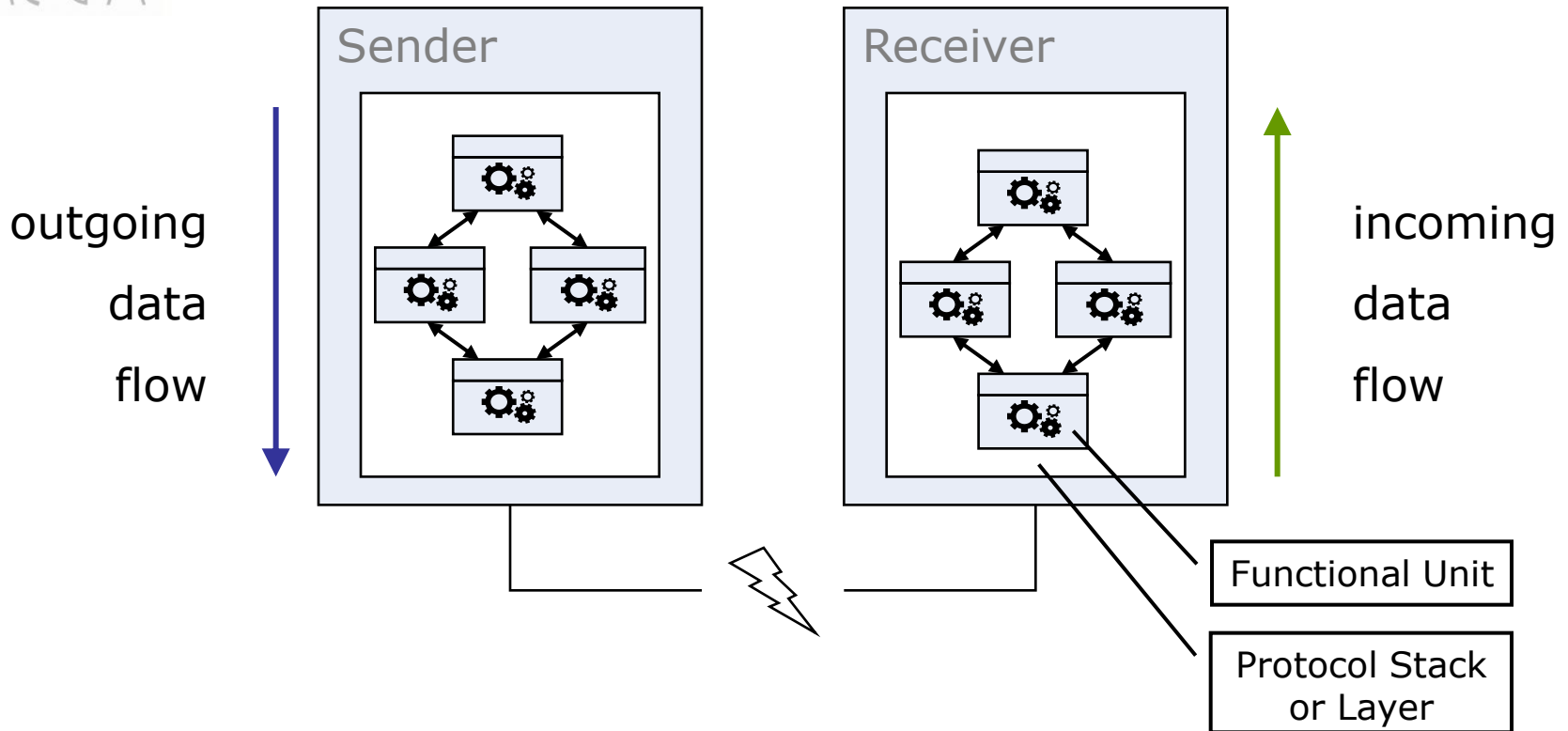


Protocol Stack A Protocol Stack B

Requirements

- High reusability of protocol functionality
 - Atomic and highly decoupled: Design patterns to decouple components (Observer, Visitor, Proxy, ...)
 - Reliable: High quality by testing, model checking, ...
- Framework which allows for the functional composition of protocol stacks
 - Interfaces between components
 - Wiring of components
 - Dynamic creation of components
 - Dynamic composition of protocol headers
 - Insertion/Removal of components
 - Suspending of components

Component-Based Protocol Stack





Functional Unit Networks - FUNs

Component: Functional Unit (FU)

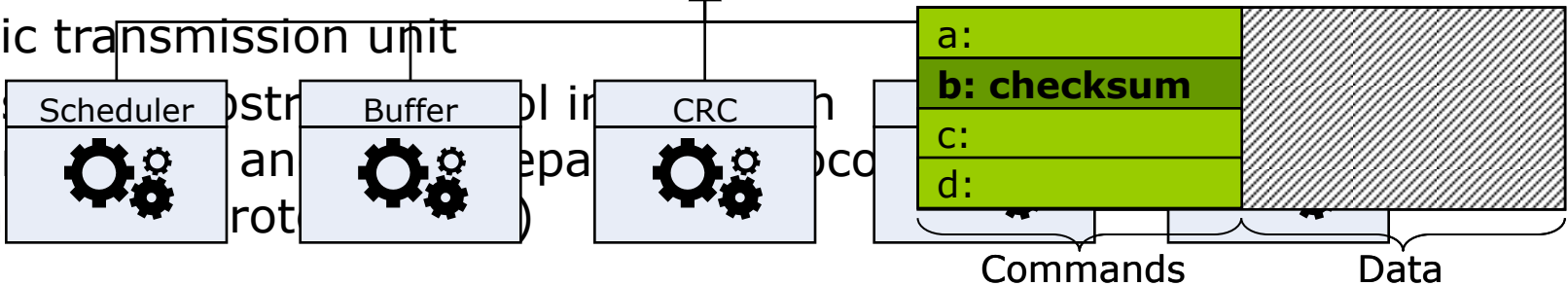
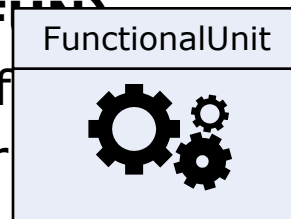
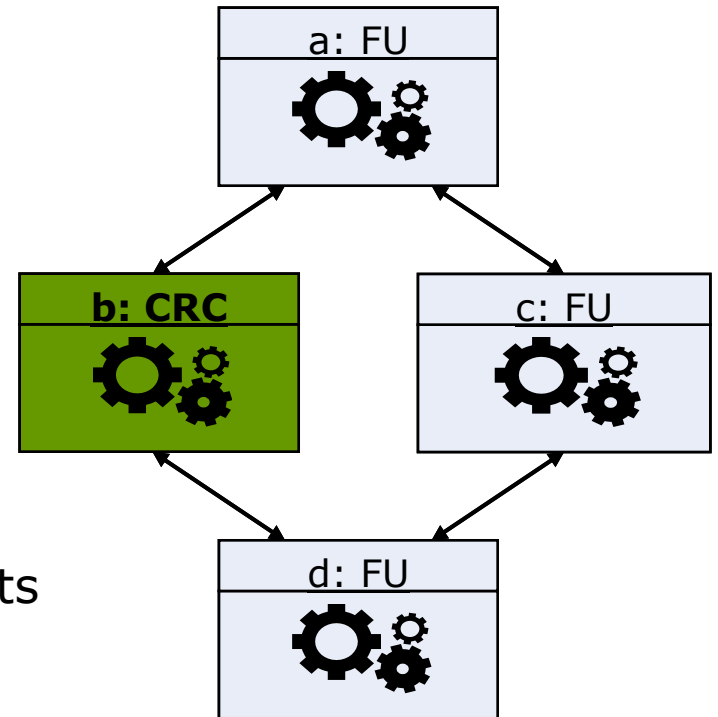
- basic building-block
- atomic protocol functionality
- implements information transformation
- basic narrow interface for communication and management
- interface customization possible
- support multiplexing and demultiplexing

Functional Unit Network (FUN)

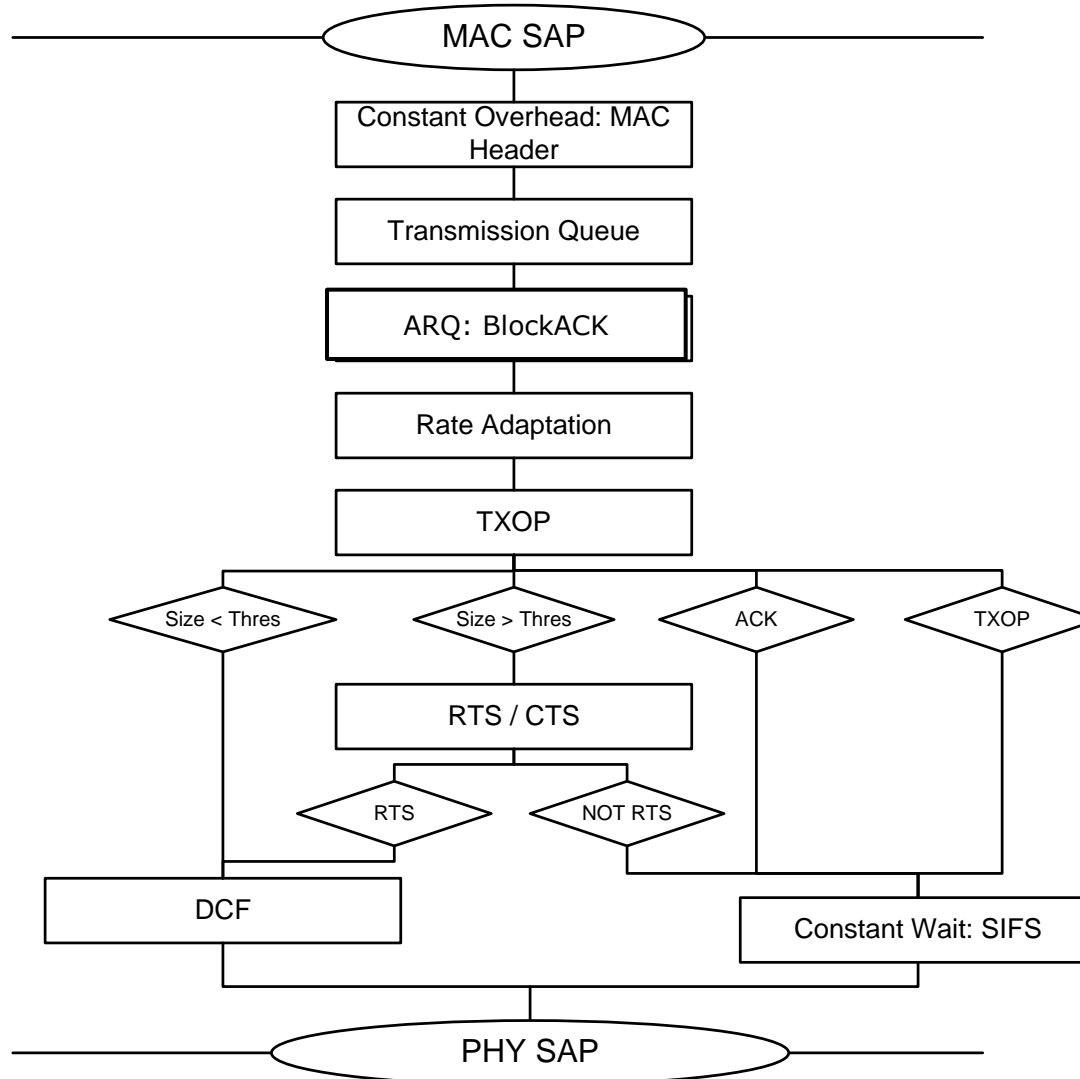
- consists of interconnected functional units
- connections represent information flow

Compound

- basic transmission unit
- compound units (Scheduler, Buffer, CRC)



Example: IEEE 802.11 FUN



Challenges for the **openWNS** (and other simulators)

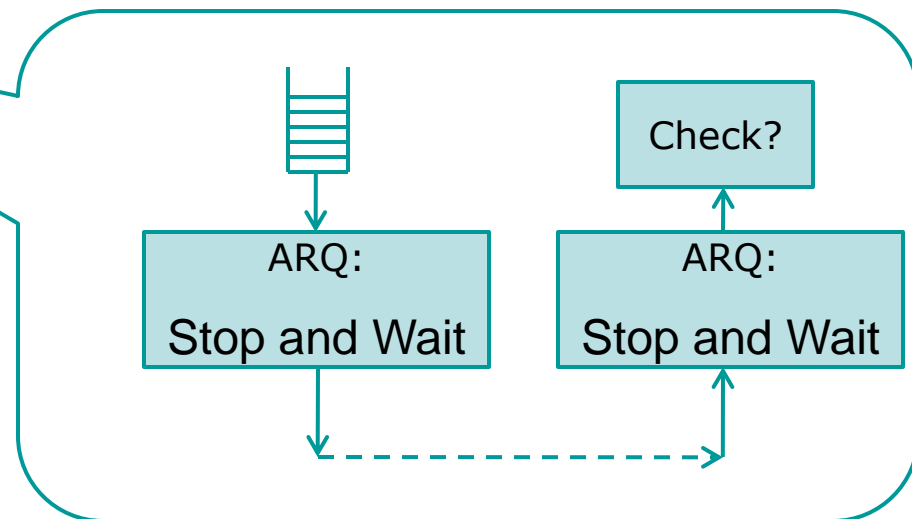
1. Future radio access technologies
 - Model new physical layer enhancements
 - Implement Medium Access Control improvements
 - Systematic protocol stack development with improvement- and extension-friendly design

2. Complexity
 - Large evaluation guidelines (e.g. IMT-Advanced, WiMAX)
 - Huge parameter space
 - Channel, physical layer, traffic models
 - Continuous & comprehensive validation

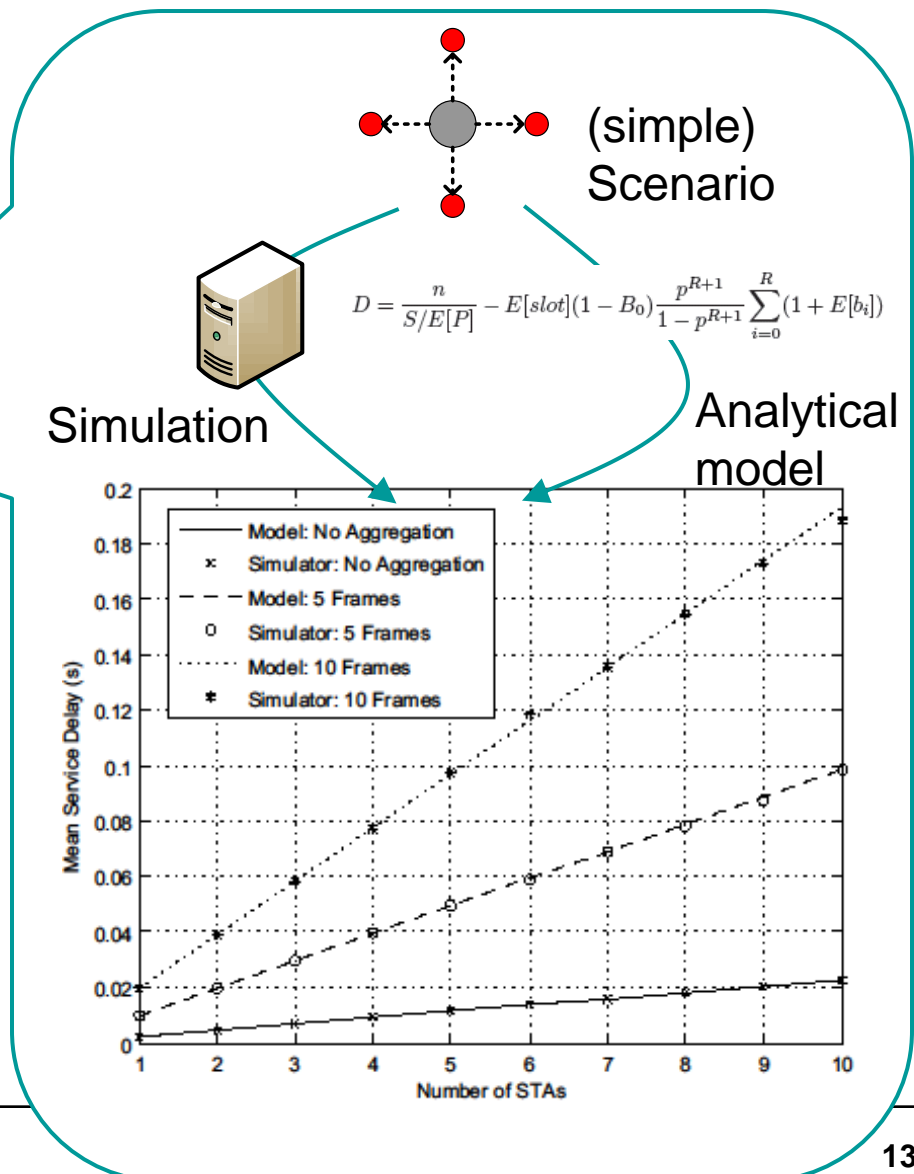
3. Scenarios
 - Large-scale simulations (e.g. 20.000 user terminals)
 - Precise protocol and wireless channel modeling
 - User-friendly management of large simulation campaigns



- Validation = Quality Assurance
 - Unit tests
 - System tests
 - Comparison with independent partners
 - Publication

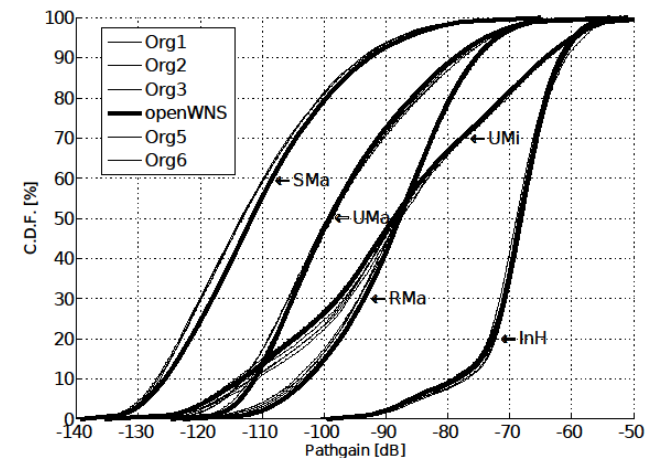
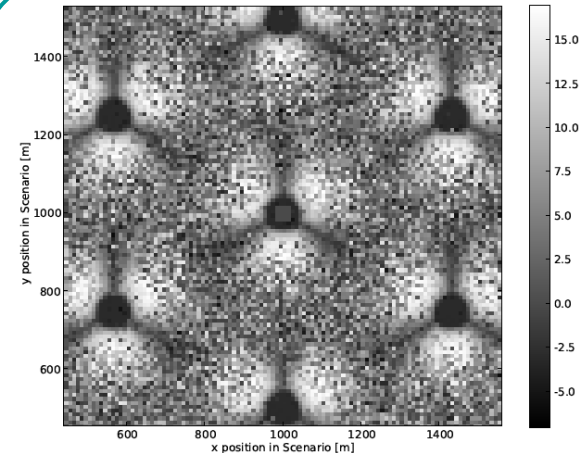


- Validation = Quality Assurance
 - Unit tests
 - System tests
 - Comparison with independent partners
 - Publication



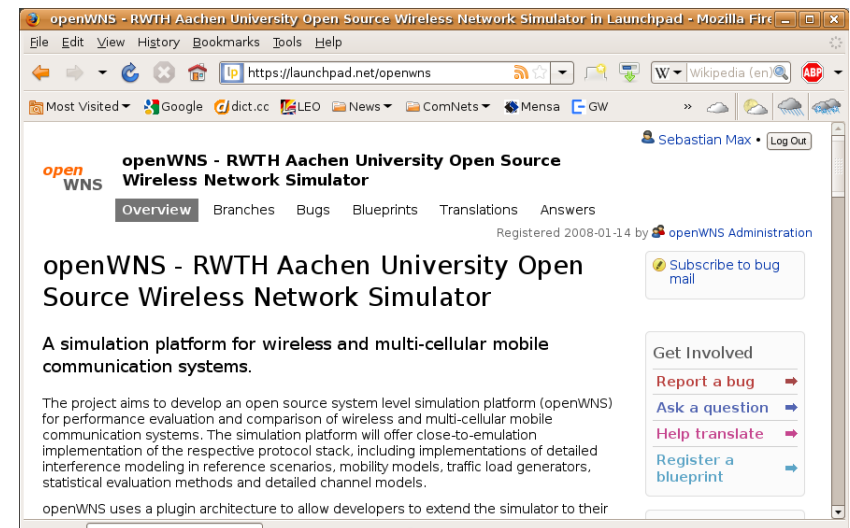
- Validation = Quality Assurance
 - Unit tests
 - System tests
 - Comparison with independent partners
 - Publication

SINR of Serving eNB for Urban Macro Scenario [dB]



- Validation = Quality Assurance
 - Unit tests
 - System tests
 - Comparison with independent partners
 - Publication

- Open source
- Lesser Gnu Public License (LGPL)
 - All modifications to libraries must be open source
 - Closed source extensions allowed
- www.openwns.org
- launchpad.net/openwns



Challenges for the **openWNS** (and other simulators)

1. Future radio access technologies

- Model new physical layer enhancements
- Implement Medium Access Control improvements
- Systematic protocol stack development with improvement- and extension-friendly design



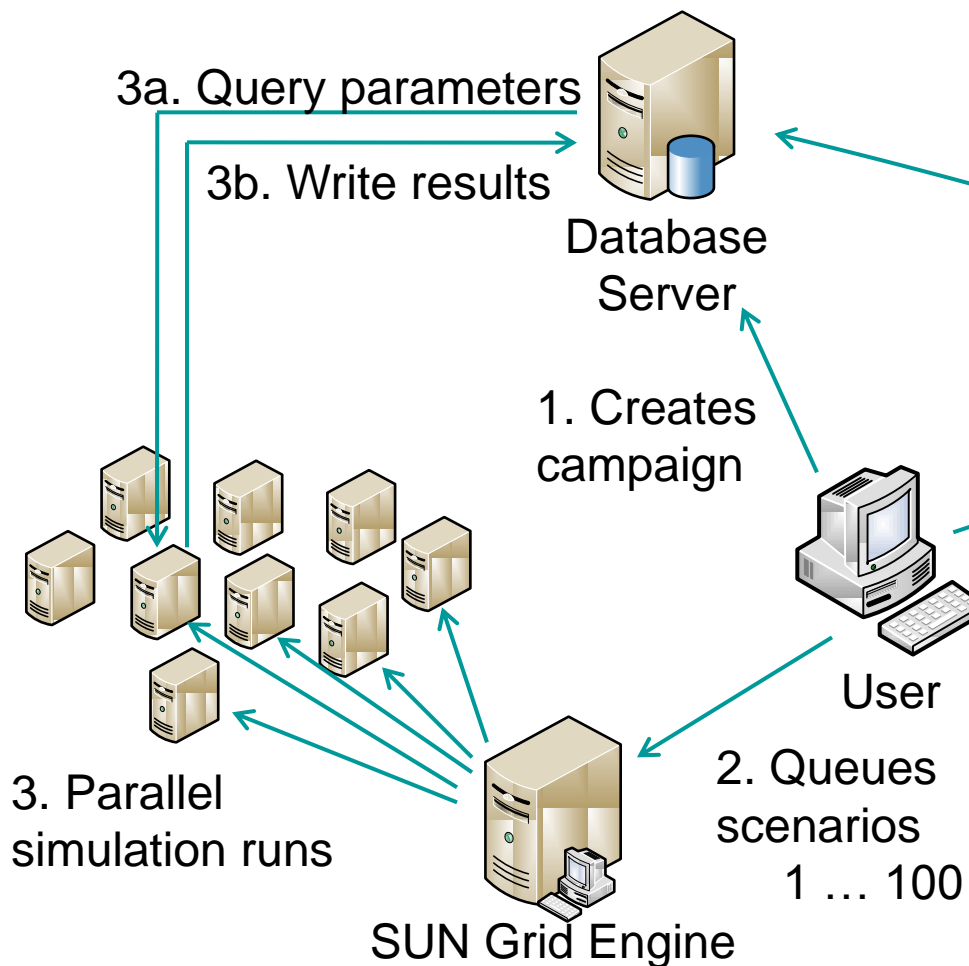
2. Complexity

- Large evaluation guidelines (e.g. IMT-Advanced, WiMAX)
- Huge parameter space
- Channel, physical layer, traffic models
- Continuous & comprehensive validation

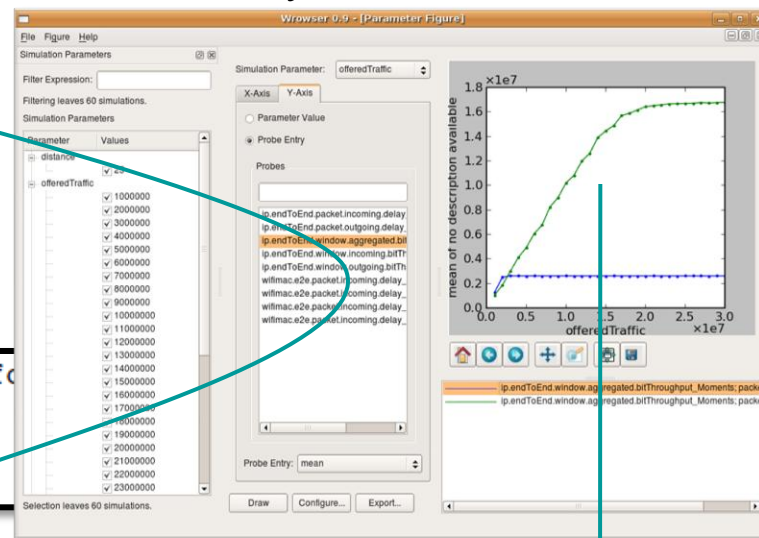


3. Scenarios

- Large-scale simulations (e.g. 20.000 user terminals)
- Precise protocol and wireless channel modeling
- User-friendly management of large simulation campaigns



4. Analyze results



5. Publication



Challenges for the **openWNS** (and other simulators)

1. Future radio access technologies

- Model new physical layer enhancements
- Implement Medium Access Control improvements
- Systematic protocol stack development with improvement- and extension-friendly design



2. Complexity

- Large evaluation guidelines (e.g. IMT-Advanced, WiMAX)
- Huge parameter space
- Channel, physical layer, traffic models
- Continuous & comprehensive validation



3. Scenarios

- Large-scale simulations (e.g. 20.000 user terminals)
- Precise protocol and wireless channel modeling
- User-friendly management of large simulation campaigns





Thank you for your attention!

smx@comnets.rwth-aachen.de

Credits to the openWNS developer team

D. Bültmann, M. Mühleisen, K. Klagges, M. Schinnenburg, K. Sambale,
R. Jennen, R. Papst, H. Rosier and countless master thesis students