



Simulating Sensor Networks with SystemC

Alban Rrustemi,

Computer Architecture Group,
Computer Laboratory



September 19, 2006

14th ESCUGM,
Darmstadt



Sensor Networks: overview

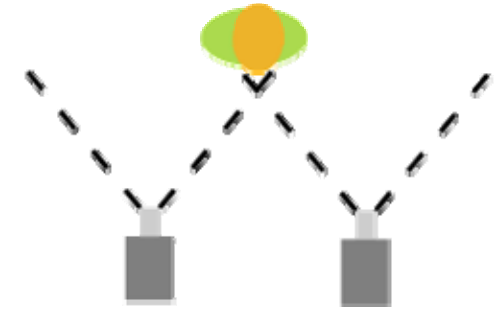
- Networks of autonomous devices using sensors to cooperatively monitor physical or environmental conditions
- Applications
 - Environmental monitoring
 - Acoustic detection
 - Seismic detection
 - Military surveillance
 - Inventory tracking
 - Medical monitoring
 - Process monitoring
 - Smart spaces



Sensor Networks: implementations

○ Implementations

- ZebraNet
- Smart Camera Networks
- ISIS Shooter Localization Systems
- Intel Process Monitoring





Sensor Networks: challenges

- Research challenges
 - Networking
 - Resource management
 - Fault tolerance
 - Low power consumption
 - In-network computation



Simulation tasks

- Need for simulation environments
 - Stimulus propagation model
 - Sensor model
 - Communication model
 - Radio model (in WSN)
 - Routing algorithms
 - Power model
 - Zero configurations

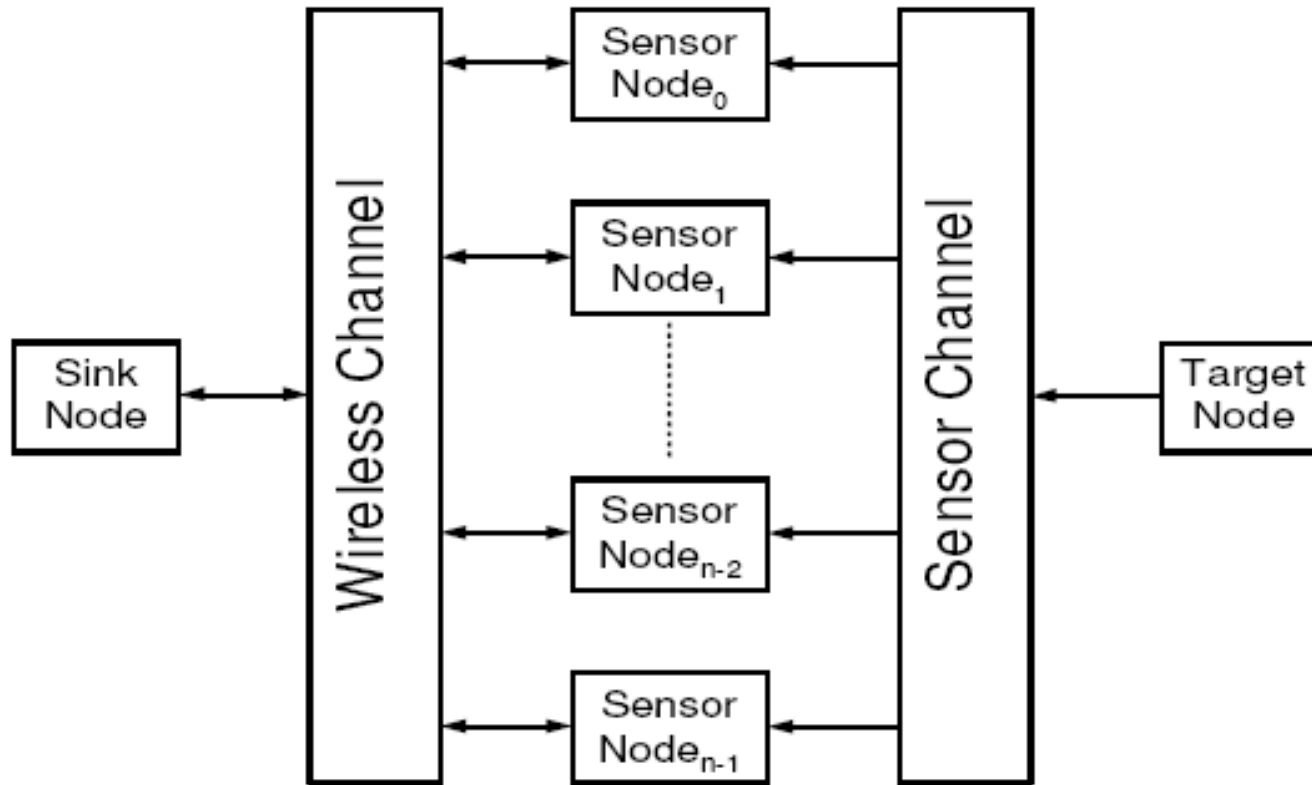


Simulating Sensor Networks

- Existing NW simulators:
 - GloMoSim
 - OPNET
 - Ns-2
 - Ptolemy
 - J-Sim
 - SensorSim
 - MagicWeaver



Implementation example

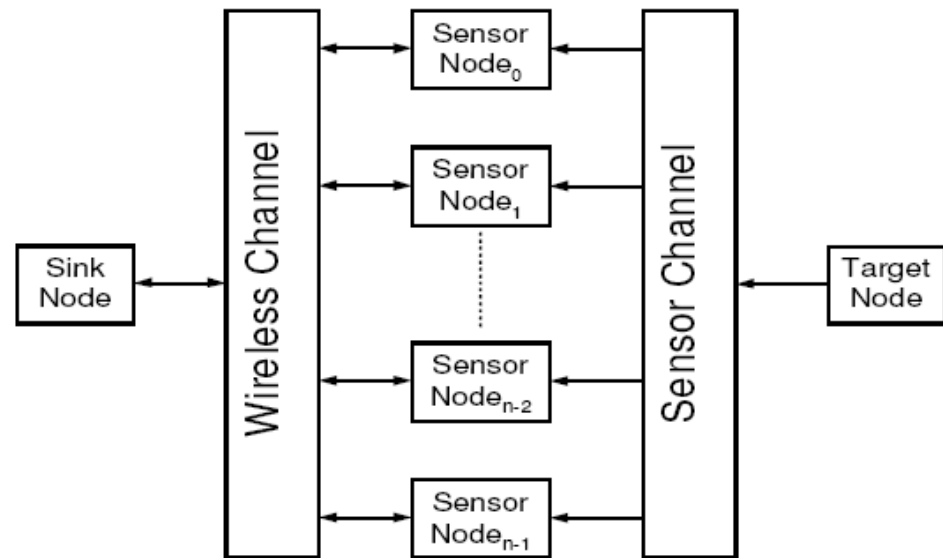




Implementation example

- J-Sim

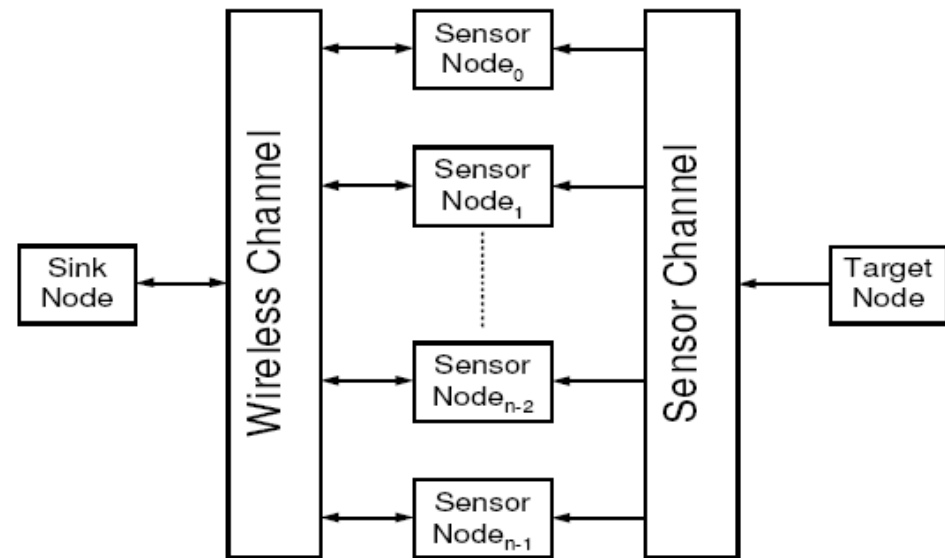
- Built upon autonomous component architecture (ACA)
 - Components
 - Ports
 - Contracts





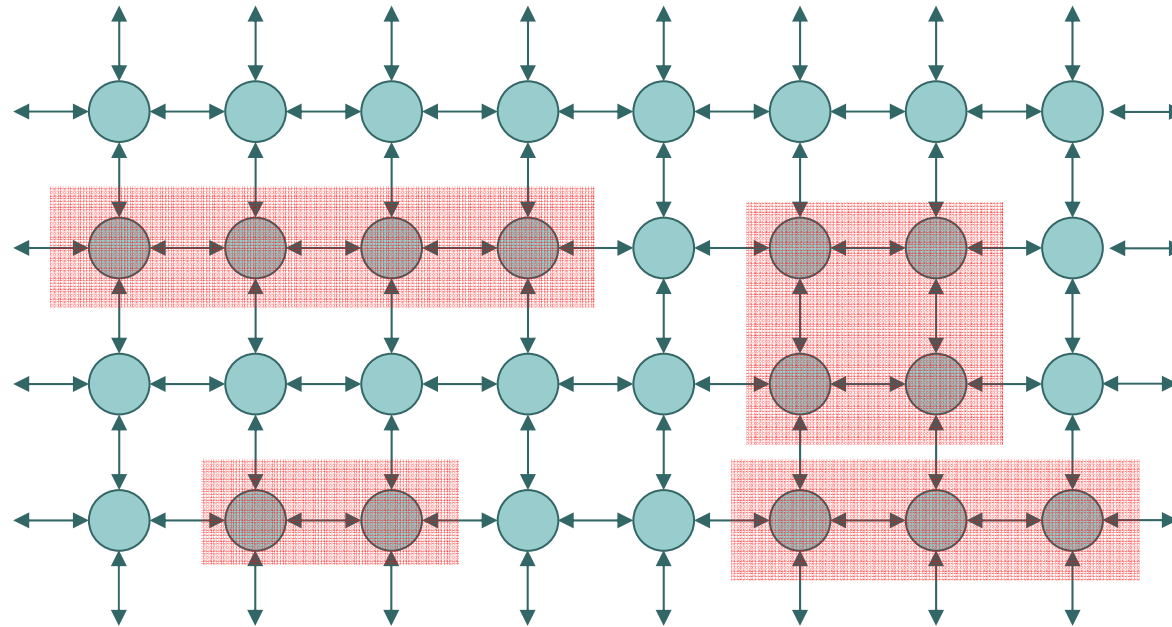
Modelling with SystemC

- Ideal abstraction support
 - Modules
 - Ports
 - Interfaces
 - Channels

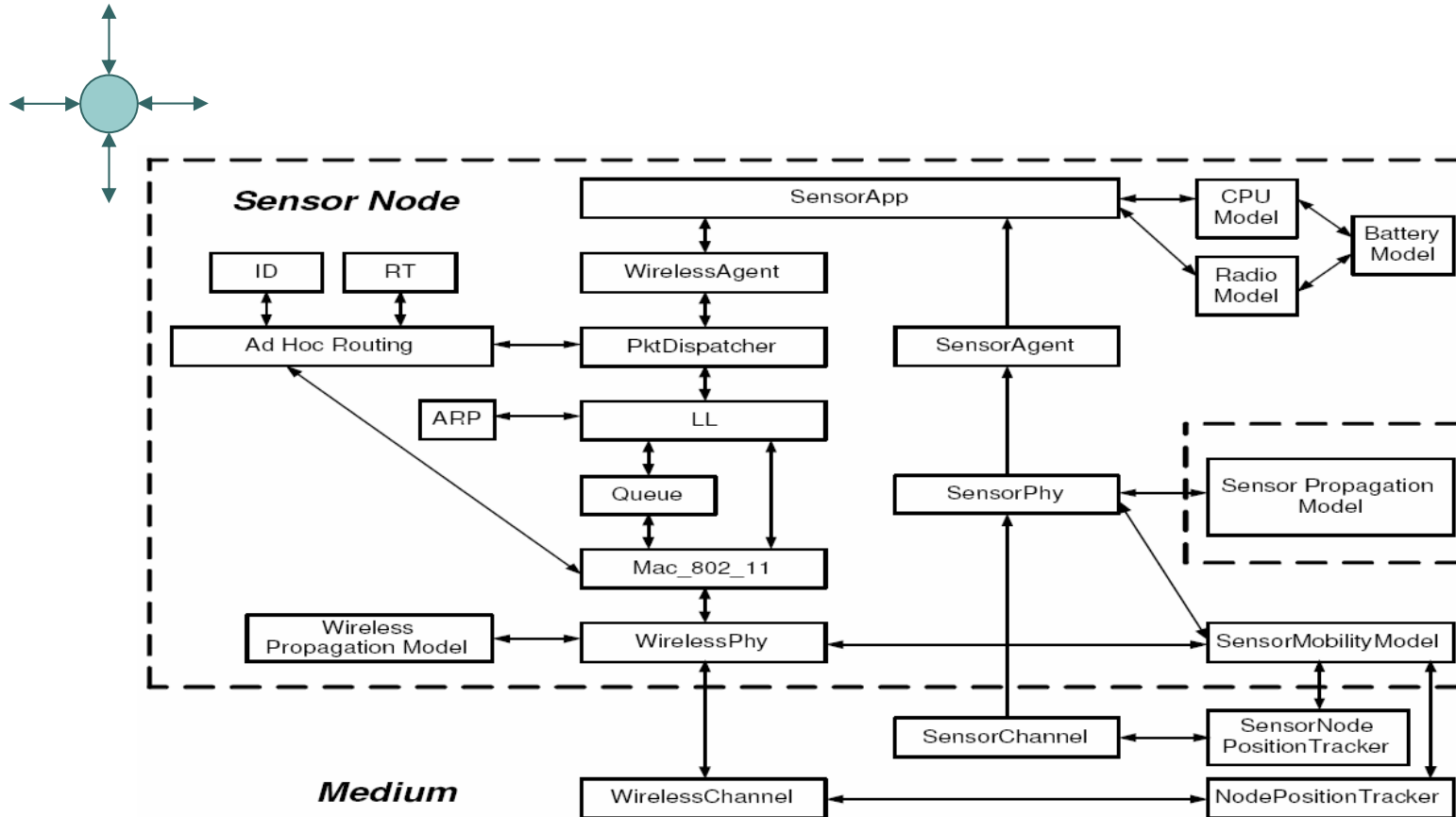




SystemC simulation with the OSCI simulation kernel

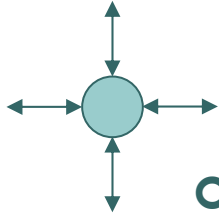


SystemC simulation with the OSCl simulation kernel





SystemC simulation with the OSCI simulation kernel



- Fundamental assumption: nodes will be in “sleep” mode most of the time
- Process implementations with user level threads
 - Fibers, QuickThread, Pthread
- Each thread allocates resources (which may not be used)
- Static hierarchy



IEEE 1666 standard

Elaboration | Simulation

4.1.1 Instantiation

Instances of the following classes (or classes derived from these classes) may be created during elaboration and only during elaboration. Such instances shall not be deleted before the destruction of the module hierarchy at the end of simulation.

- sc_module (see 5.2)

- sc_port (see 5.11)

- sc_export (see 5.12)

- sc_prim_channel (see 5.14)

An implementation shall permit an application to have zero or one top-level modules and may permit more than one top-level module (see 4.3.4.1 and 4.3.5).



Static hierarchy in OSCI simulator

- Pros
 - optimization and/or transformation of internal data structures for simulation speed or capacity
 - synthesis
 - formal verification
- Cons
 - scalability
 - running different test cases



Other systems

- Same problems could be faced when simulating large:
 - Homogenous SoCs
 - NoC platforms



Ways around

- Analyse the test case
 - filter out unneeded modules before elaboration
- Use `SC_METHOD` instead of `SC_THREAD` whenever possible
- Use dynamic processes

- Not elegant!



Conclusion

- SystemC proved very suitable for modelling SN!
- Simulating: challenging!
- Thank you!