Generic Role Assignment for Wireless Sensor Networks

RESEARCH GROUP FOR
Distributed Systems

Christian Frank, Kay Römer
ETH Zürich

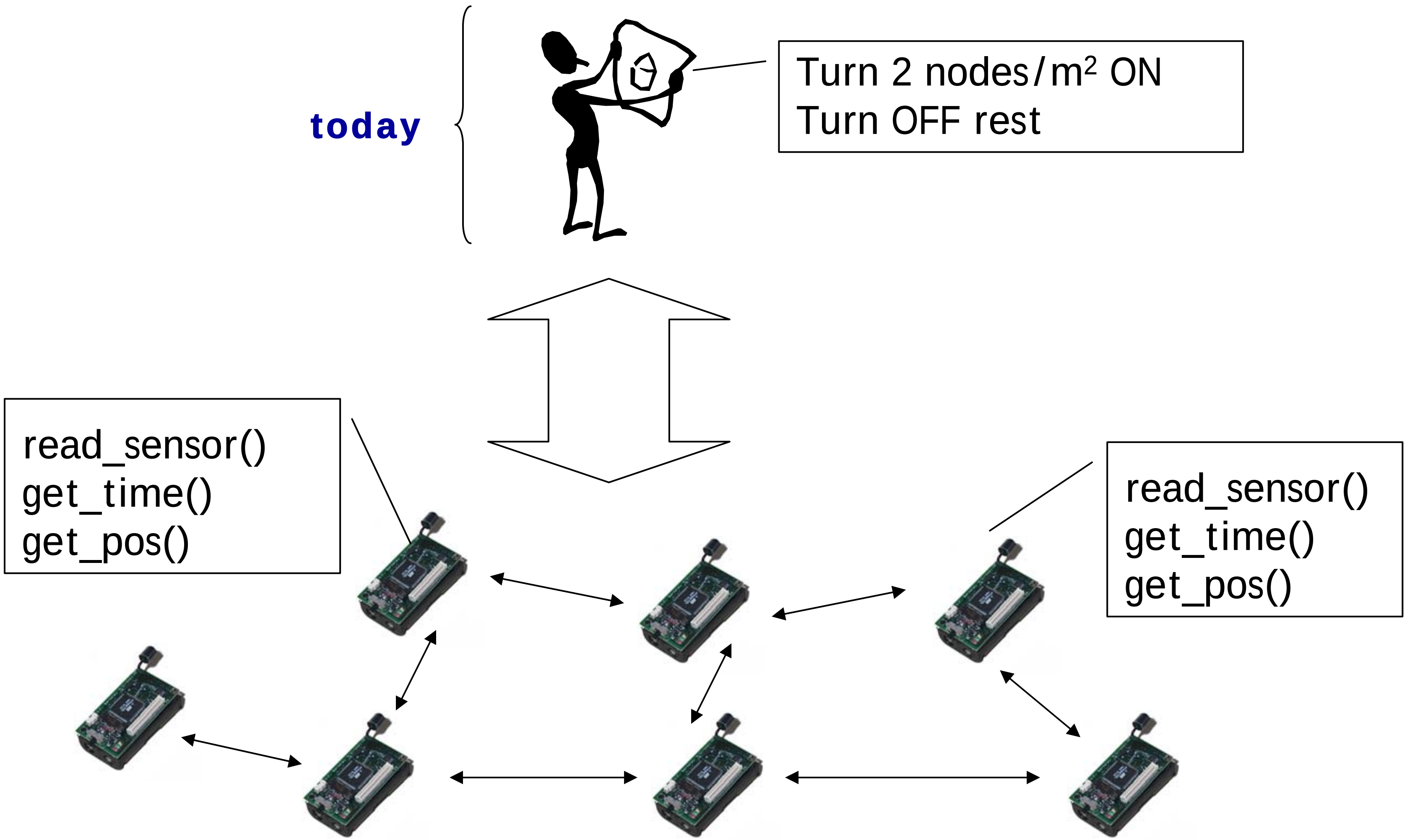
Christian Becker, Pedro José Marrón
Universität Stuttgart

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

The Gap

- Past research focussed on energy efficient:
  - Operating system and hardware abstraction layers (TinyOS, etc.)
  - Services: routing, medium access, localization, time synchronization
- Sensor networks are mostly programmed as a distributed system
- Current research:
  - Abstract from distributed-system details, e.g., message passing → provide higher level abstractions
  - Programmability key for sensor network usability
- This talk:
  - Role assignment (programmer describes network heterogeneity)

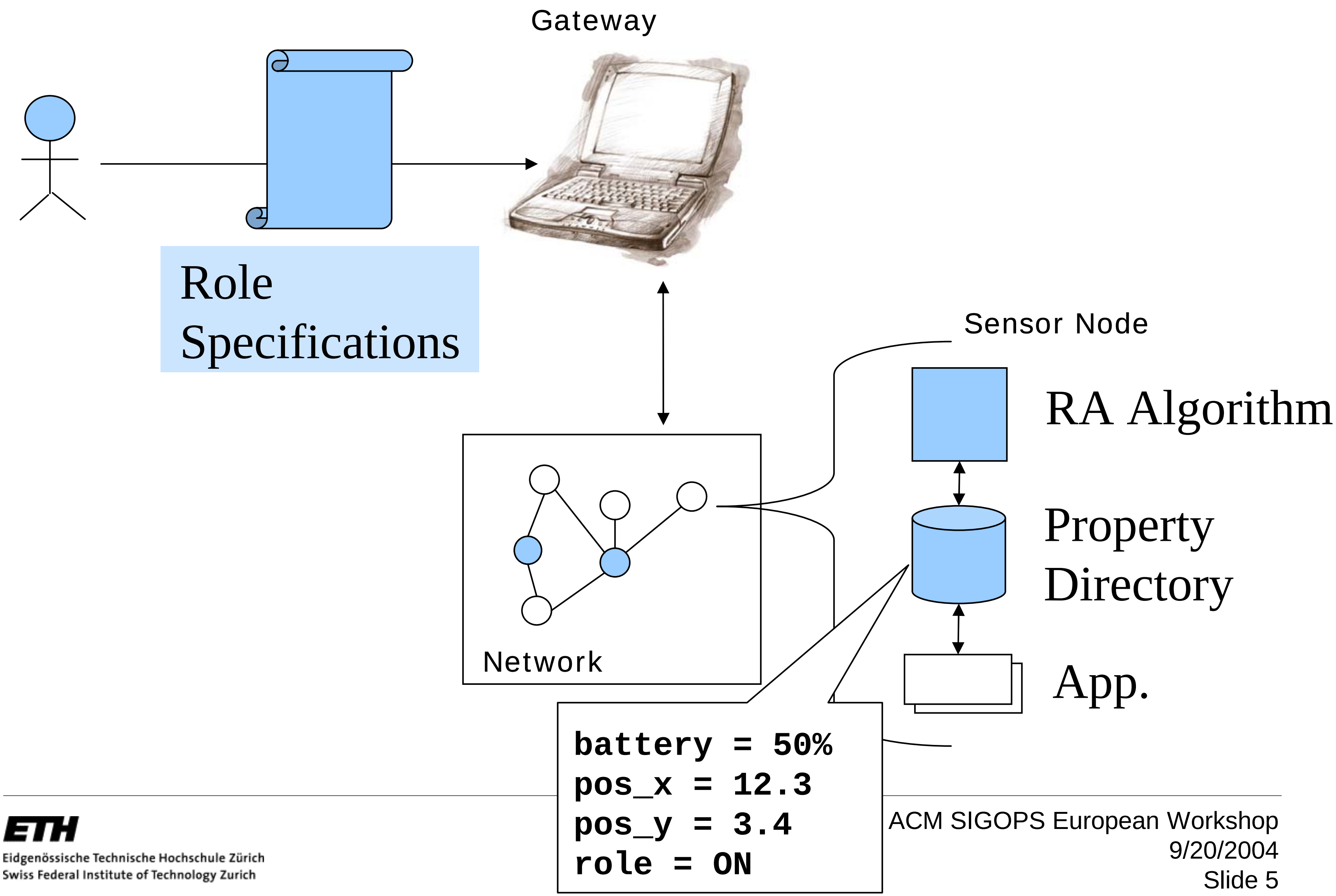
The Gap


today
Turn 2 nodes/m² ON
Turn OFF rest
read_sensor()
get_time()
get_pos()
read_sensor()
get_time()
get_pos()

Examples for Role Assignment

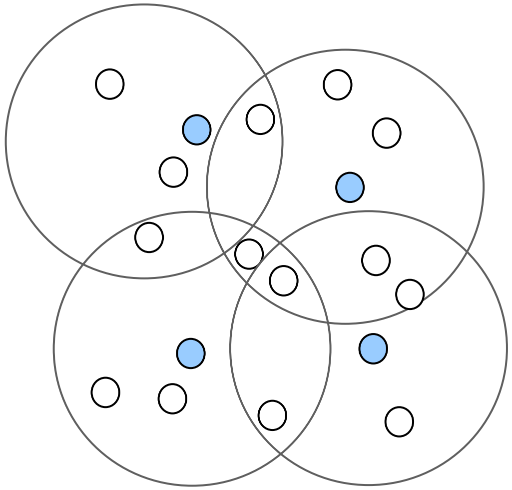
- Coverage
  - Roles ON, OFF
  - ON nodes cover every geographic spot
- Clustering
  - Roles: Clusterhead, Gateway, Slave
  - Connected Subgraph
- Data Aggregation
  - Roles: Data Source, Aggregator
  - Close(Src, Agg)
  - Dist(Sink, Agg) < Dist(Sink, Source)

Use Case / Architecture



Coverage Appl.

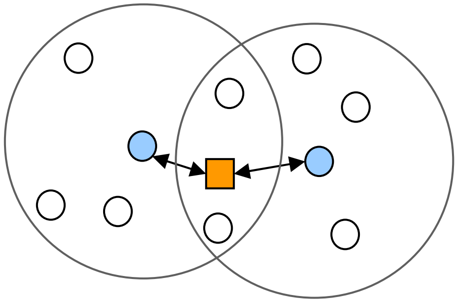
```
ON :: {
    temp-sensor == true &&
    battery >= threshold &&
    count(2 meters) {
        role == ON
    } == 0
}
OFF :: else
```



- count(scope) { pred }:
  – Counts nodes matching *pred* within *scope*

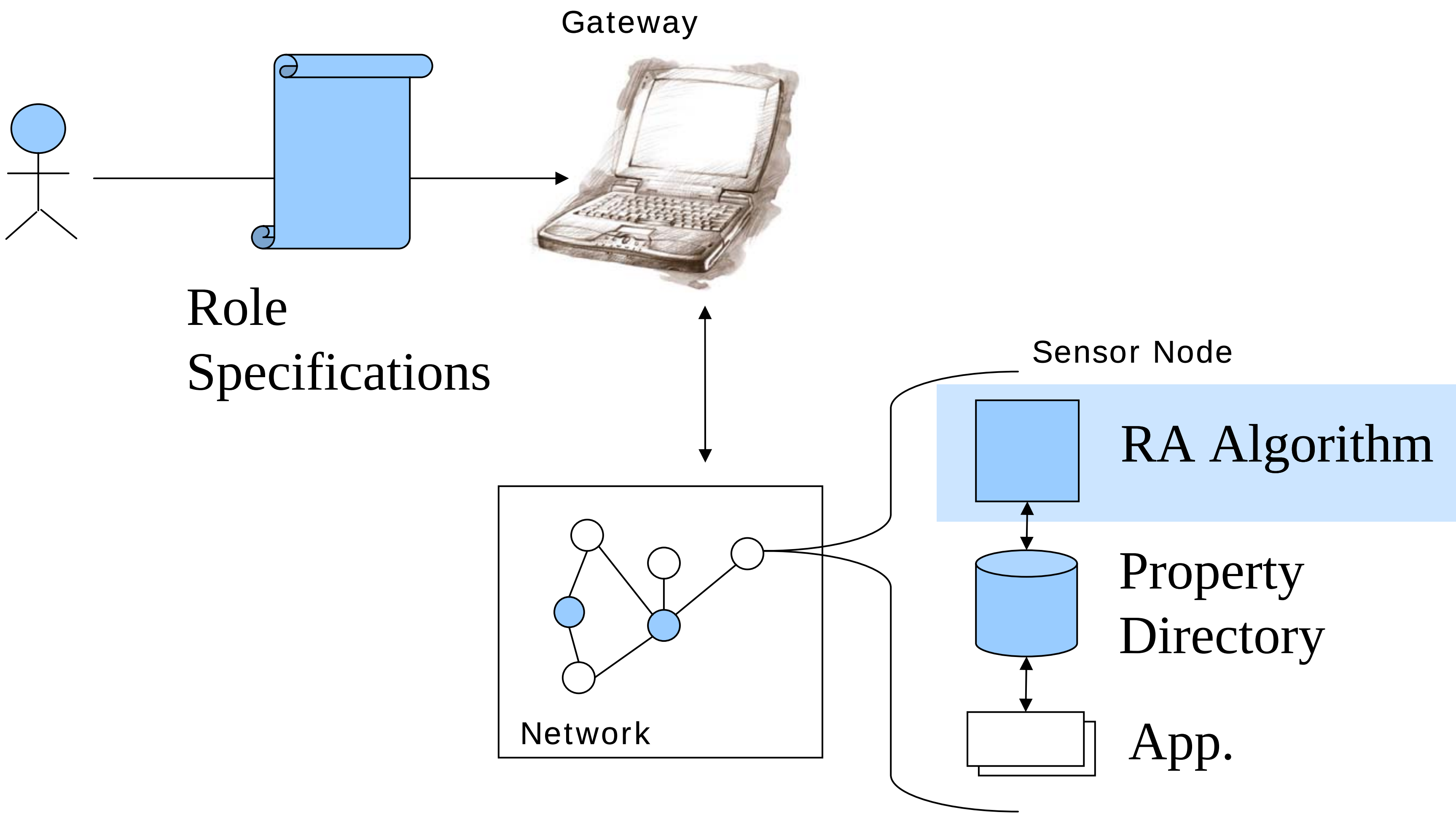
Clustering Appl.

```
CLUSTERHEAD :: {
    count(1 hop) {
        role == CLUSTERHEAD
    } == 0 }
GATEWAY(c1,c2) :: {
    retrieve(1 hop, 2) {
        role == CLUSTERHEAD
    } == (c1,c2) &&
    count(2 hops) {
        role == GATEWAY(c1,c2)
    } == 0 }
SLAVE :: else
```



- retrieve(scope, num) { pred } == (c1,c2) :
  - At least *num* nodes in *scope* must fulfil *pred*
  - Bind the 2 nodes to params (c1,c2)

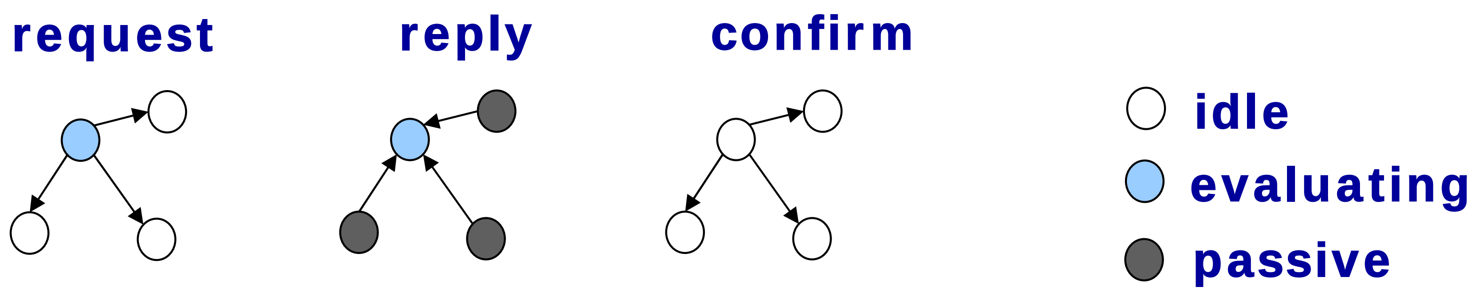
Use Case / Architecture



Distributed Algorithm

- Preliminary approach
- Local neighbourhood queries (request/reply)
- Ensure atomicity of rule evaluation
- Queries triggered:
  - After deployment
  - Changes of neighbour properties

Distributed Algorithm



Node sends request message

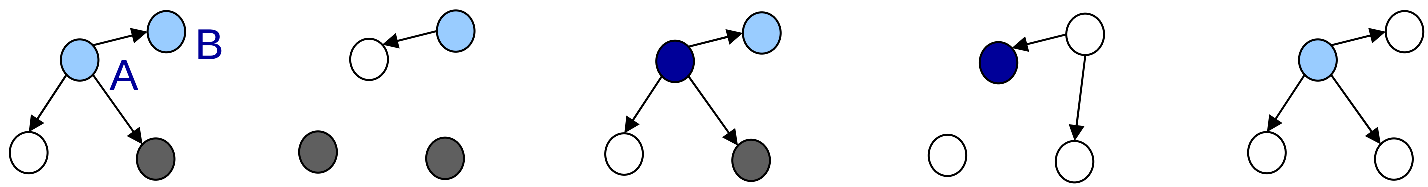
Nb. reply. and assume passive state

Node confirms, now nb. may evaluate

- One query evaluates complete RA specification for one node

Distributed Algorithm

request | **abort** | **abort** | **confirm** | **re-request**



A sends request although B already evaluating

B sends abort

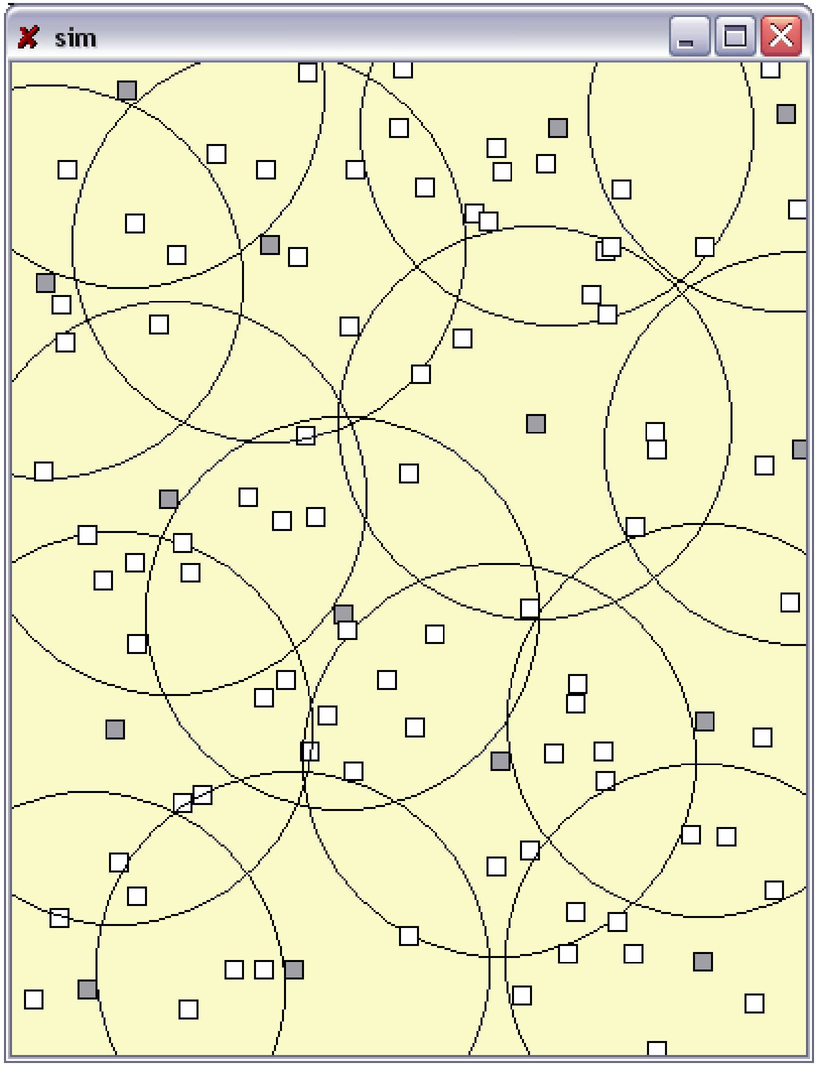
A yields, abort allows neighbors to act

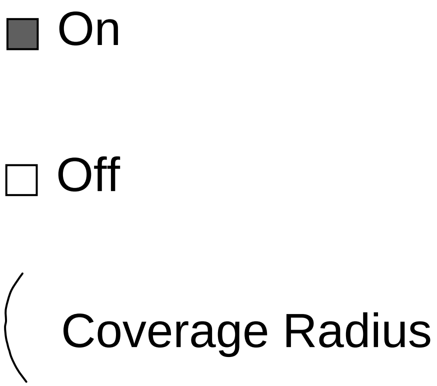
B confirms, eval. over

A starts eval. sends request

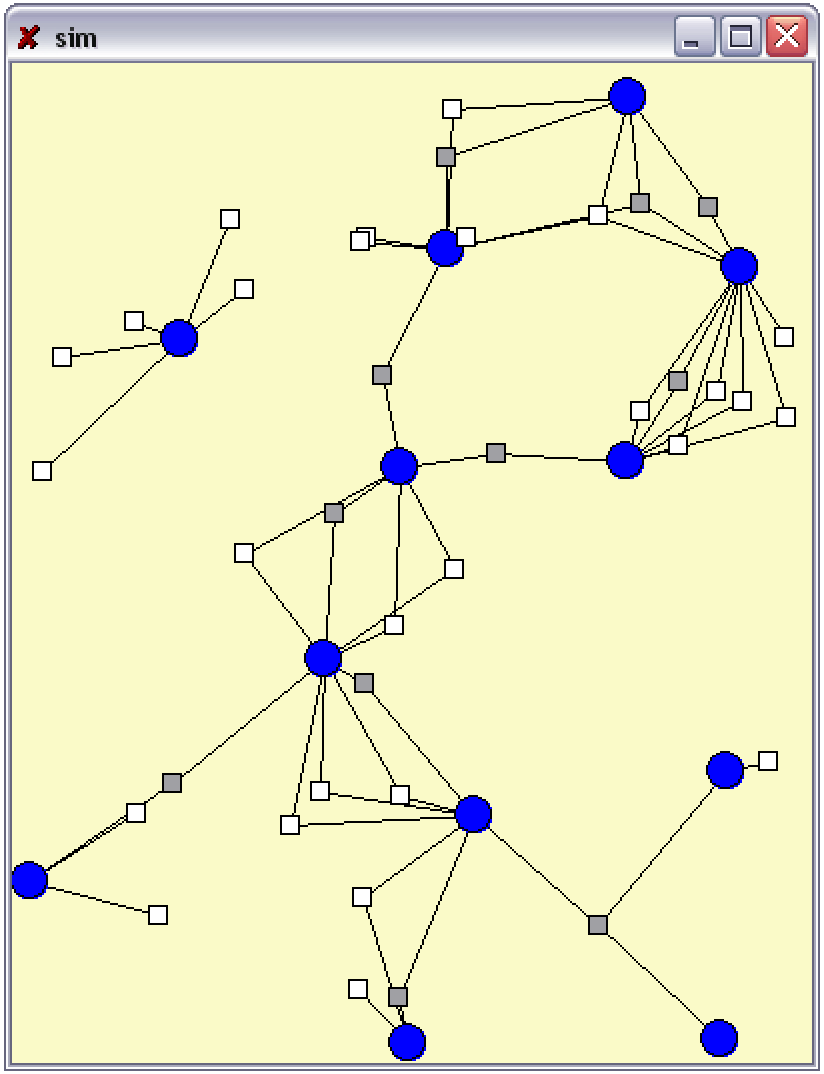
- **idle**
- **evaluating**
- **passive**
- **awaiting ev.**

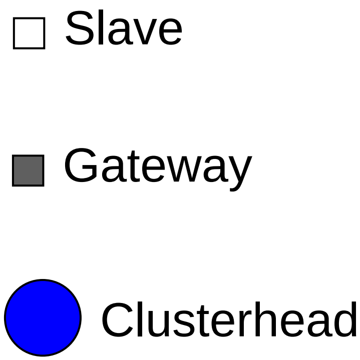
Coverage Simulation



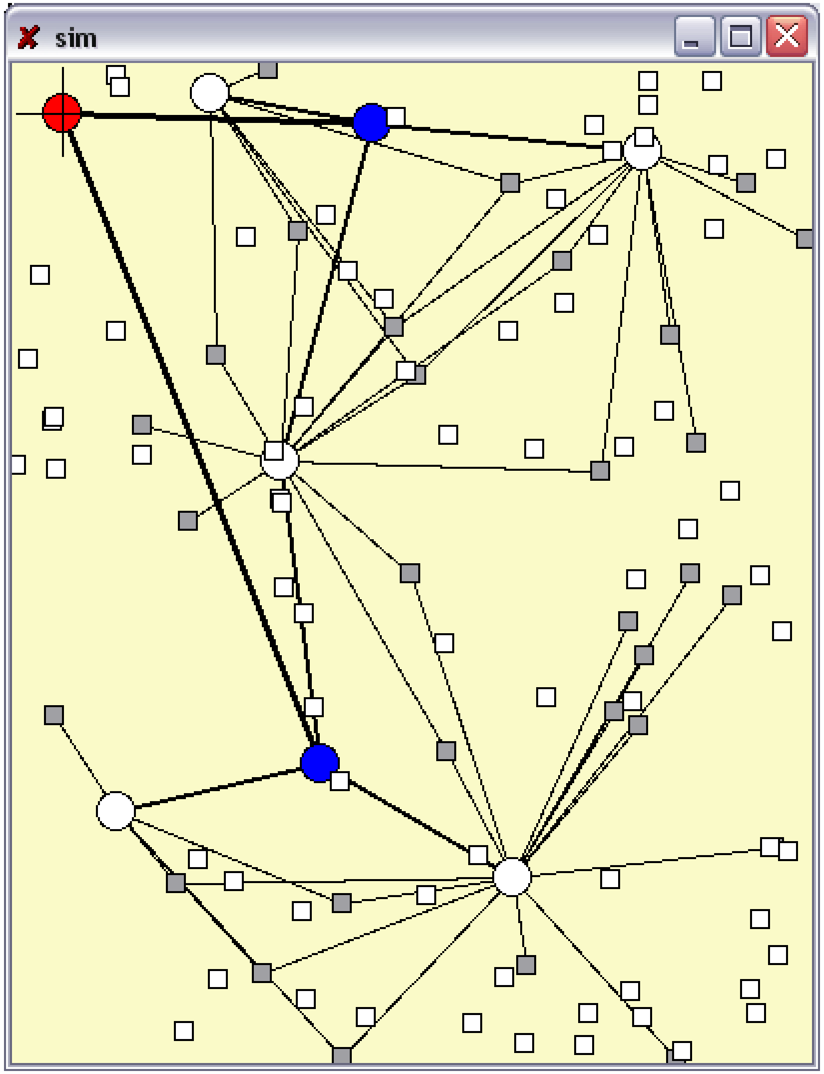


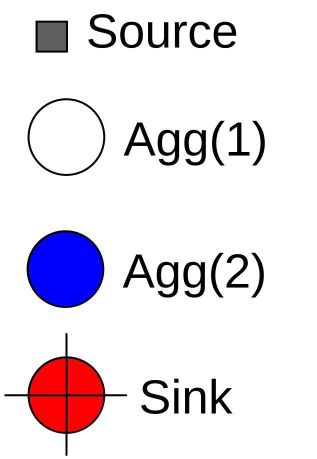
Clustering Simulation





Aggregation Simulation





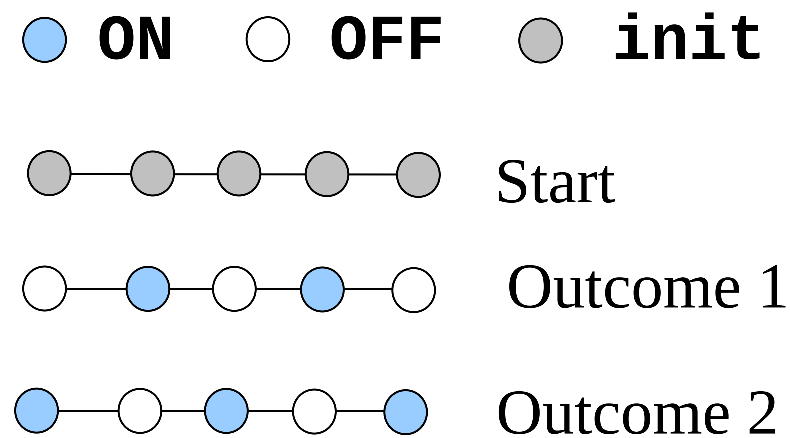
Limitations / Discussion

- Efficiency
  - Limited scope of count/retrieve
  - Possible improvements:
    - Nodes with changes proactively send
    - Precompilation
- Some specifications may not terminate
  - Practical relevance?
  - Support user to detect non-terminating specifications?

Additional Specifications?

- Coverage Example:

```
ON :: {
  count(1hop) {
    role == ON
  } == 0
}
OFF:: else
```



- Current version is non-deterministic (outcomes 1+2)
- Coverage example would require *few* ON nodes
- Additionally: One could tolerate breaching some rules but not others → weighting of different rule clauses

Current Work

- Centralized Algorithm
  - Suitable for simulation/experiments with various
    - role specifications
    - topology types
- Returns:
  - Feasible solution or infeasible
  - Possibly helps to detect termination
    - some infeasible specifications don't terminate?
  - Optimal solution (minimize certain role)
- Derive Integer Program from
  - Role specifications
  - Network topology
  - Node properties

Conclusion/Outlook

- Role Assignment powerful programming abstraction
- Initial approach promising
- Open questions
  - Computational overhead?
  - Termination?
  - Optimality?

Rel. Work

- Hood: Whitehouse et al. (Mobisys04)
  - Data sharing among neighbors
  - Broadcast/filter approach
- Abstract Regions: Welsh/Mainland, (NSDI04):
  - Share state in an arbitrary multi-hop region
    - N-radio hop / add. geo-filter, spanning tree
- Amorphous Computing: Abelson et al. (Comm. of ACM, May 2000)

Generic Role Assignment for Wireless Sensor Networks

Christian Frank, Kay Römer
ETH Zürich

Christian Becker, Pedro José Marrón
Universität Stuttgart

RESEARCH GROUP FOR
Distributed Systems

Thank you!
Questions?

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich