

# ARTiS: Design and Implementation of an Adaptive Middleware for Parallel and Distributed Simulation

---

**Gabriele D'Angelo**

joint work with

**Dr. Luciano Bononi  
Dott. Michele Bracuto  
Prof. Lorenzo Donatiello**



**University of Bologna  
Department of Computer Science**

May, 2005

# Presentation outline

- Computer Simulation, Parallel and Distributed Simulation (PADS)
- IEEE 1516 standard – High Level Architecture (HLA)
- The ARTiS middleware: design and implementation
- Example: Ad-Hoc network simulation
  - a migration based approach (GAIA)
  - performance analysis
- Example: Sensor network simulation
  - simulation of a new energy-aware MAC protocol
  - performance analysis
- Concurrent Replication of PADS (CR-PADS)
- Conclusions and future work

# Computer simulation

*"A simulation is a system that represents or emulates the behavior of another system over time. In a computer simulation the system doing the emulating is a computer program"*

In a very simplified form a computer simulation can be seen as a set of **evolving variables**

The evolution of state variables over time can be managed with different timings (continuous or discrete time flow)

In our case state variables will be updated at **discrete** points in the simulation time (**discrete event simulation**)

# Monolithic sequential approach

The traditional sequential monolithic approach is often inadequate to complete the simulation execution or to fulfill the requirements

Limitations due to:

- limited resources (i.e. system memory, computational power)
- excessive amount of time required to complete the runs (i.e. real time simulation)
- fault-tolerance requirements
- distributed nature of the simulation
- intellectual property (IP) issues

# Parallel and distributed simulation (PADS)

**Parallel simulation:** two or more Physical Execution Units (PEUs) are interconnected by a *low latency communication bus* (i.e. SMP shared memory systems)

**Distributed simulation:** an architecture based on a set of loosely-coupled PEUs, interconnected by a *high latency network* (i.e. LAN, WAN, Internet)

Real world system are often a mix of both: most of High Performance Computing (HPC) architectures

# Simulation decomposition and distributed environment

The simulation is decomposed in a set of **Logical Processes** (LPs), each one allocated on a single CPU. Each LP is responsible to manage the evolution of a subset of **entities** or **state variables**

The communication between LPs is based on **message-passing**

- In a centralized system the notion of time is strictly defined, generated events can be processed following a total ordering thanks to the CPU clock, assumed as main reference of timing
- A distributed environment has not a single reference clock
- Synchronization algorithms: conservative and optimistic approaches

# The shared state and load balancing problems

- In a distributed simulation of highly interacting system models, the main bottleneck may become the communication and synchronization required to maintain the *causality constraints* between distributed model components  
**(communication latency and bandwidth)**
- Many approaches have been investigated in order to reduce the *overhead* effects of distributed synchronization and communication **(i.e. Data Distribution Management)**
- The simulated entities allocation (on the LPs composing the simulation) is subjected to **load balancing** constraints

# IEEE 1516 standard: High Level Architecture

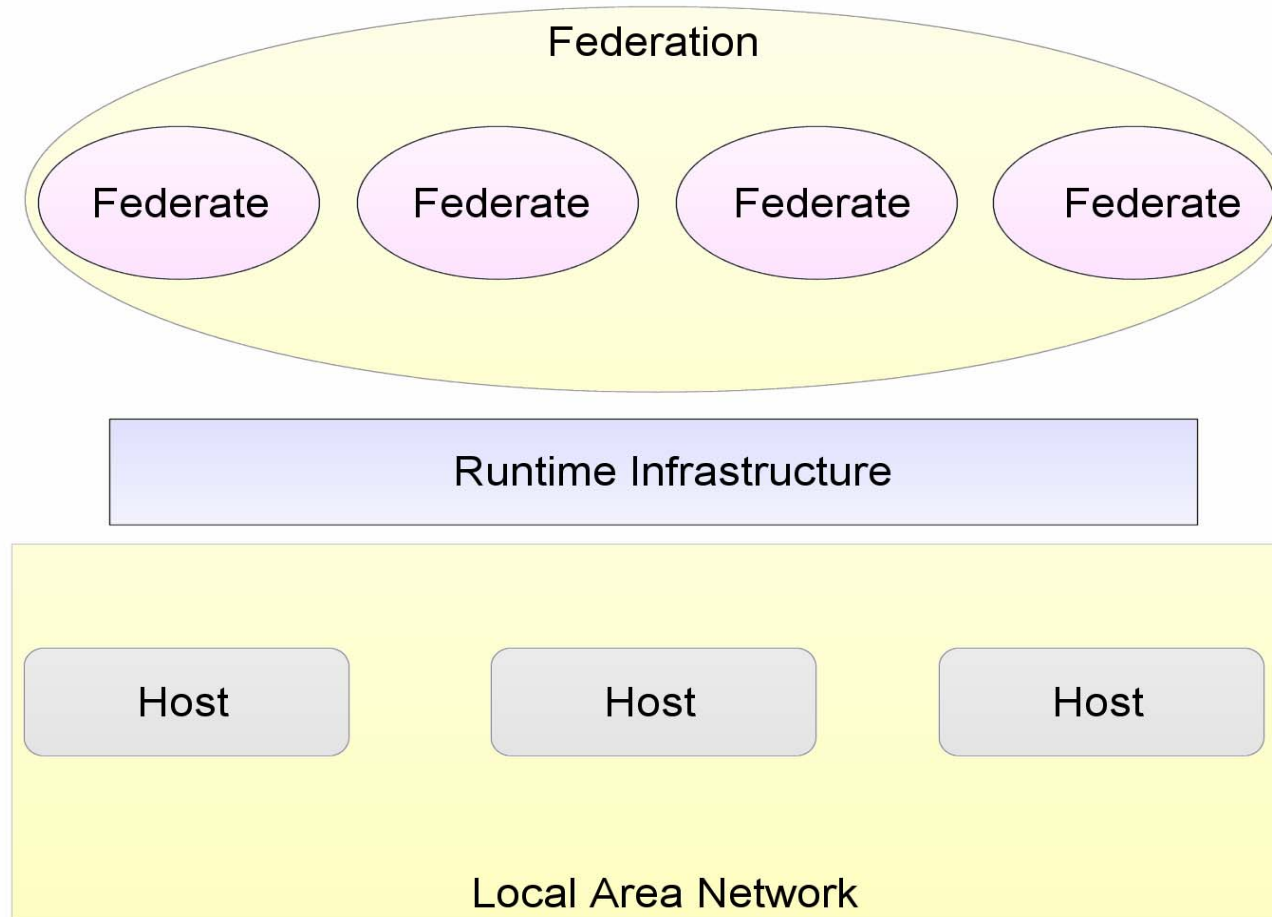
The **High Level Architecture (HLA)** is a general purpose architecture for simulation reuse and interoperability. It was approved as an open standard in September 2000 (**IEEE 1516**)

The standard is composed by:

- Object Model Template (OMT)
- Interface specification
- A set of rules

The standard defines an architecture and **NOT** an implementation

# IEEE 1516 standard: High Level Architecture

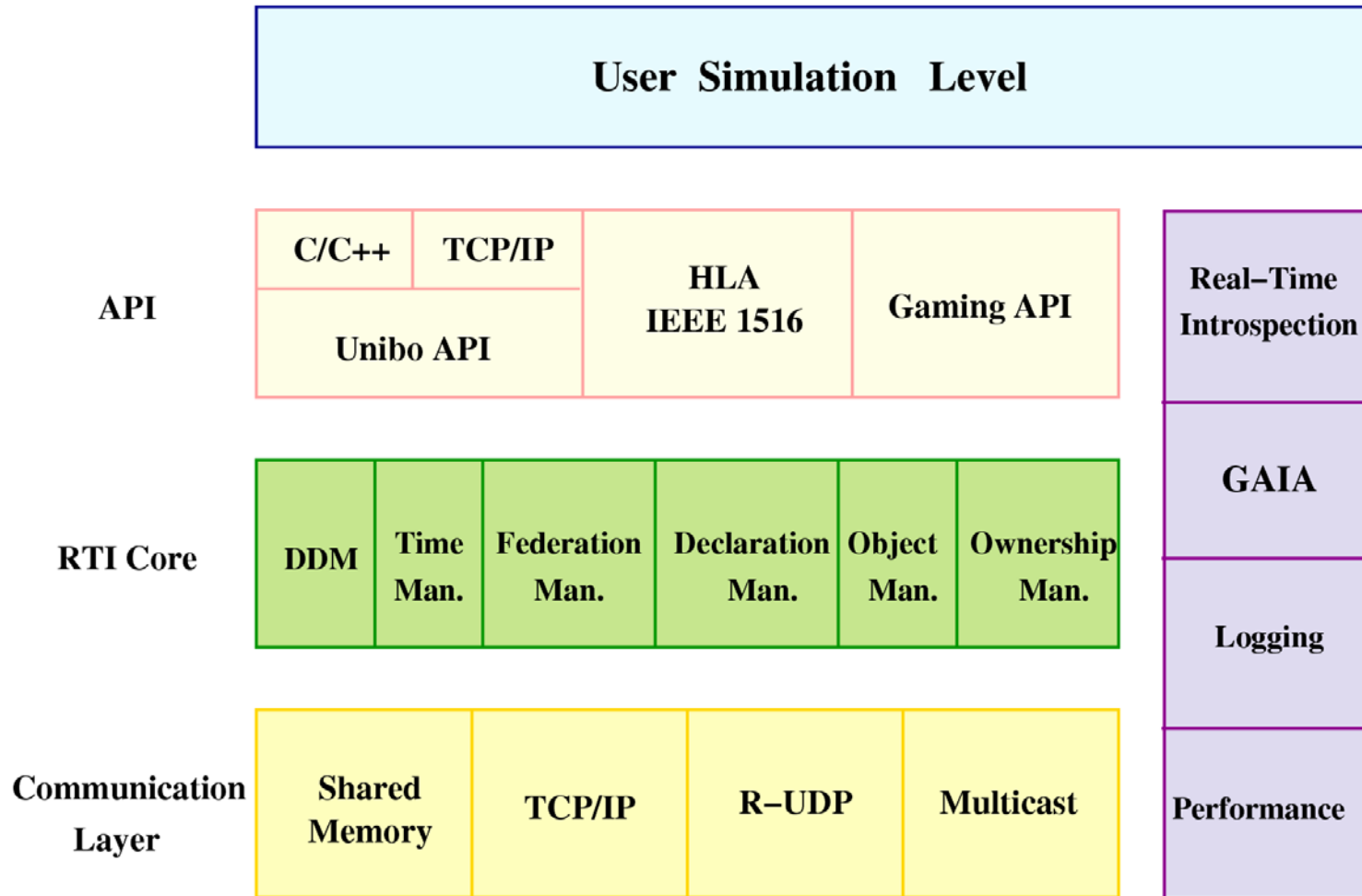


# ARTiS: Advanced RTI System

ARTiS is a new **middleware** for parallel and distributed simulation, it aims to support the model components heterogeneity, distribution and reuse, and to increase the simulation **performances, scalability** and **speed-up**

- Dynamic adaptation of Inter-Process Communication (IPC)
- Simulated entities migration support (GAIA)
- HLA-"inspired"
- Advanced simulation techniques support (replication, cloning)

# ARTiS: Advanced RTI System



# Simulation of Wireless and Mobile Systems

- The simulation of Ad-Hoc and Sensor networks often requires a **large amount of computation, memory and time** to obtain significant results
- The parallel and distributed simulation approach can be a valuable solution to **reduce the computation time** and to support model components' modularity and **reuse**
- Distributed synchronization implies communication overheads

# Mobile and Wireless Networks' model characteristics

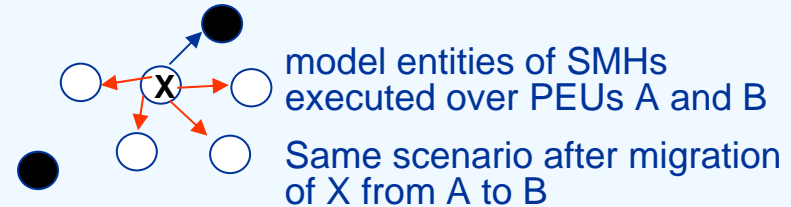
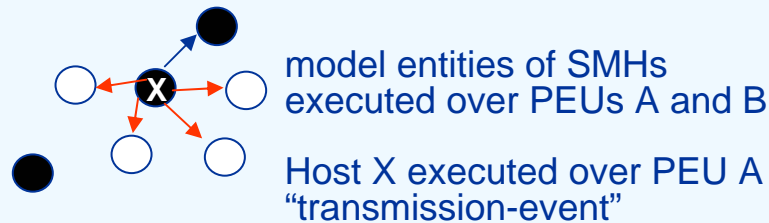
- **“Open broadcast” nature of the wireless transmissions**
  - “space-locality” of causality between neighbor-hosts
  - neighbor-hosts should be notified about transmission events anyway, e.g. to model interference, detection, MAC, etc.
- **Wireless devices can be mobile**
  - the set of neighbor-hosts change as simulated time elapses
- **Communication between hosts is “session-based”**
  - determines a “time-locality” effect
  - the set of neighbor-hosts is interested by transmission events, for a significant time-window

The group of model entities in the shared causal-domain can be highly dynamic:

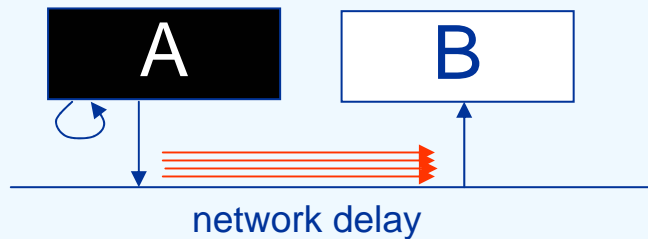
high degree of communication is required to maintain full synchronization

# Simulated Mobile Hosts (SMHs) migration

Wireless ad hoc network scenario: (evaluating migration of SMH x)



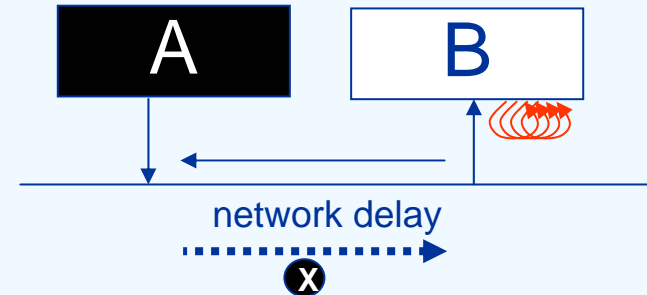
Physical Execution Units for the simulation



X's "transmission-event" must be notified to the 4 model entities executed over B

SMH = Simulated Mobile Host  
PEU = Physical Execution Unit

Physical Execution Units for the simulation



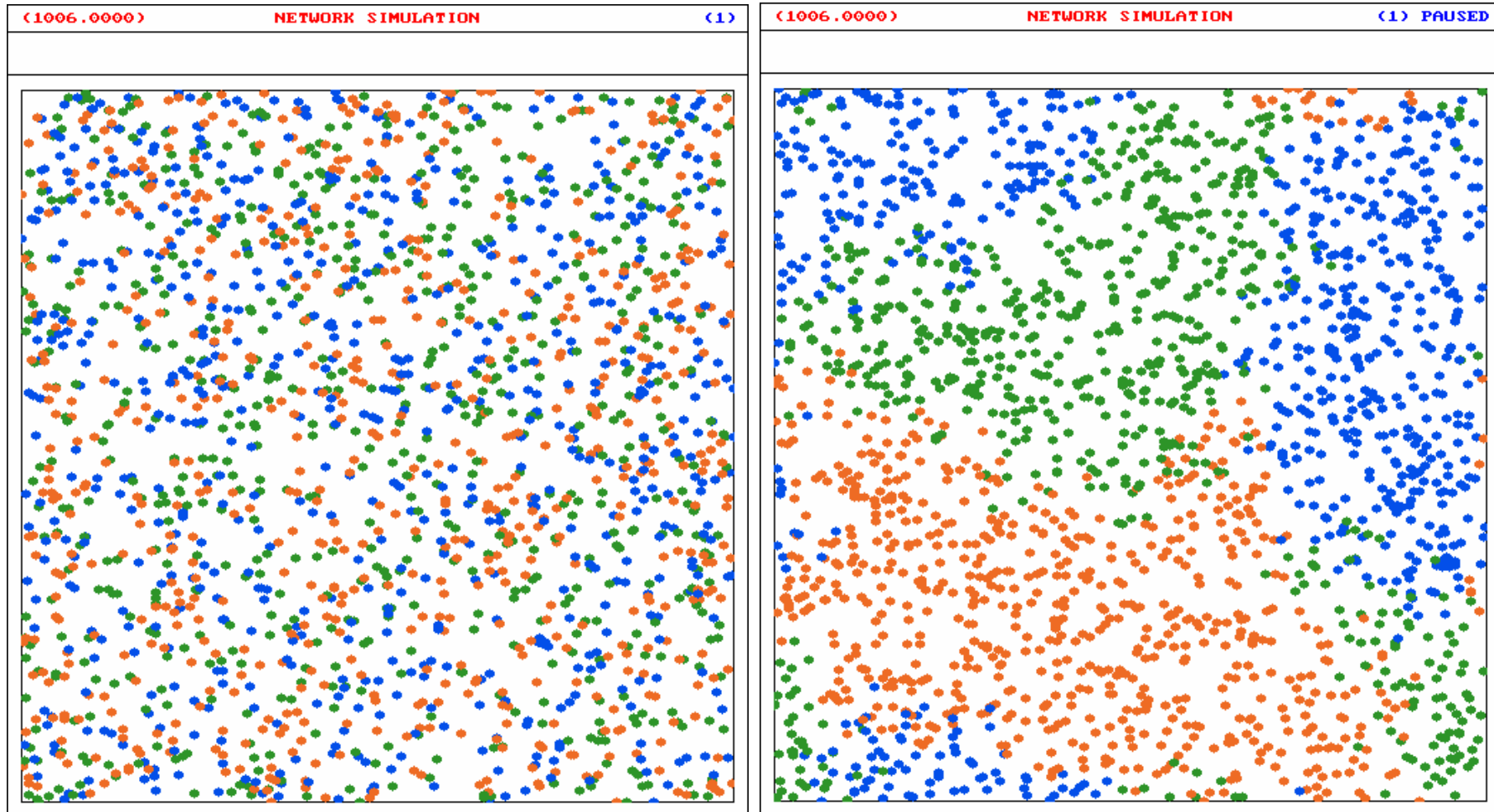
After X's migration, X's "transmission-event" must be notified to one model entity executed over A

# Example: Ad-Hoc network model implementation

## Modeling issues:

- A set of Simulated Mobile Hosts (SMHs)
- Mobility model:
  - Random Mobility Motion model (RMM)
  - Slow- and Fast-RMM (10, 25, 100 m/s)
  - uncorrelated SMHs' mobility (worst case)
- Traffic model:
  - ping messages (CBR) by every SMH to all neighbors within the wireless communication range (250 m)
  - low local computation model (worst case)
- Propagation model
  - open space (neighbor-SMHs within detection range)

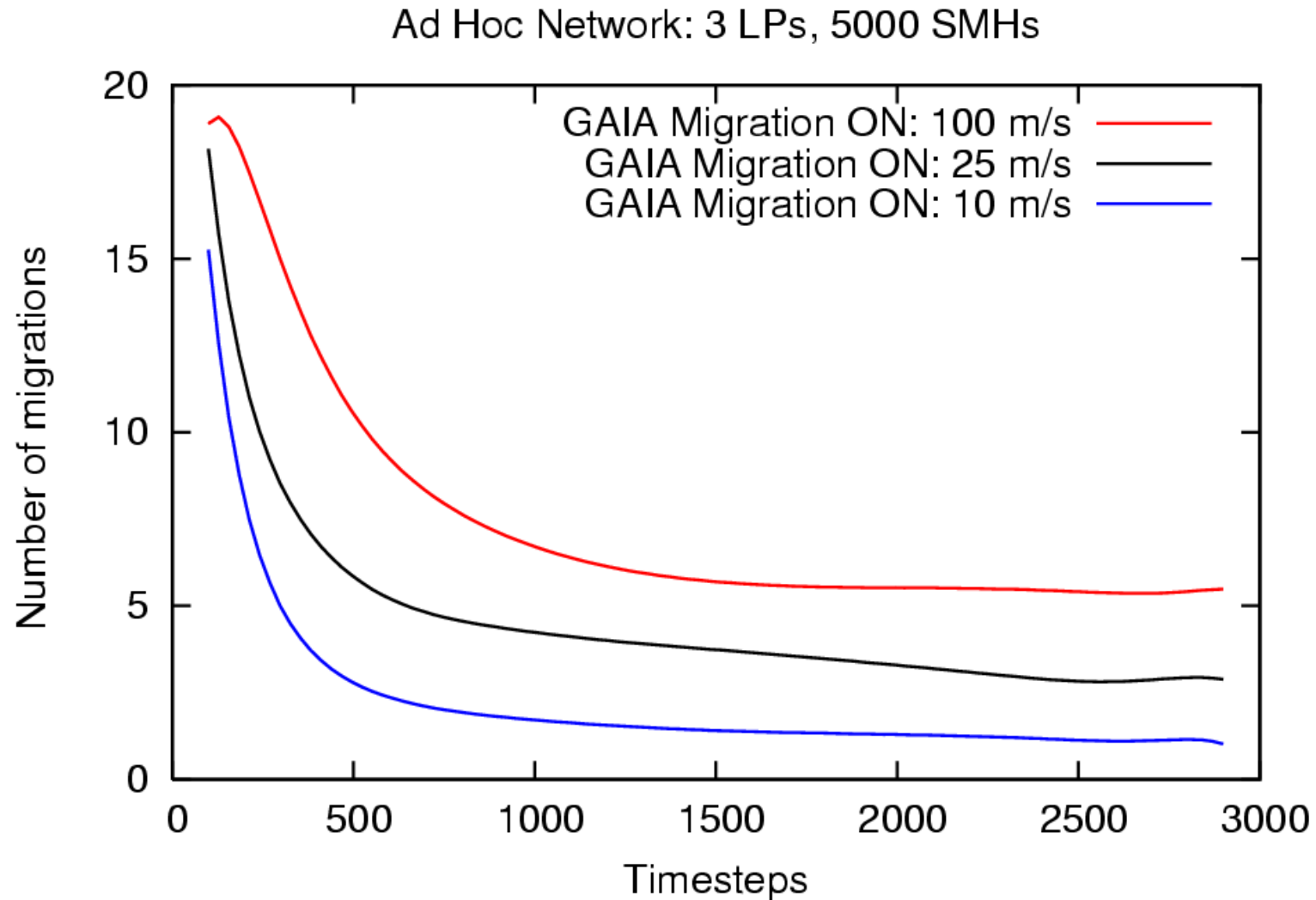
# Ad hoc network: migration mechanism “off” and “on”



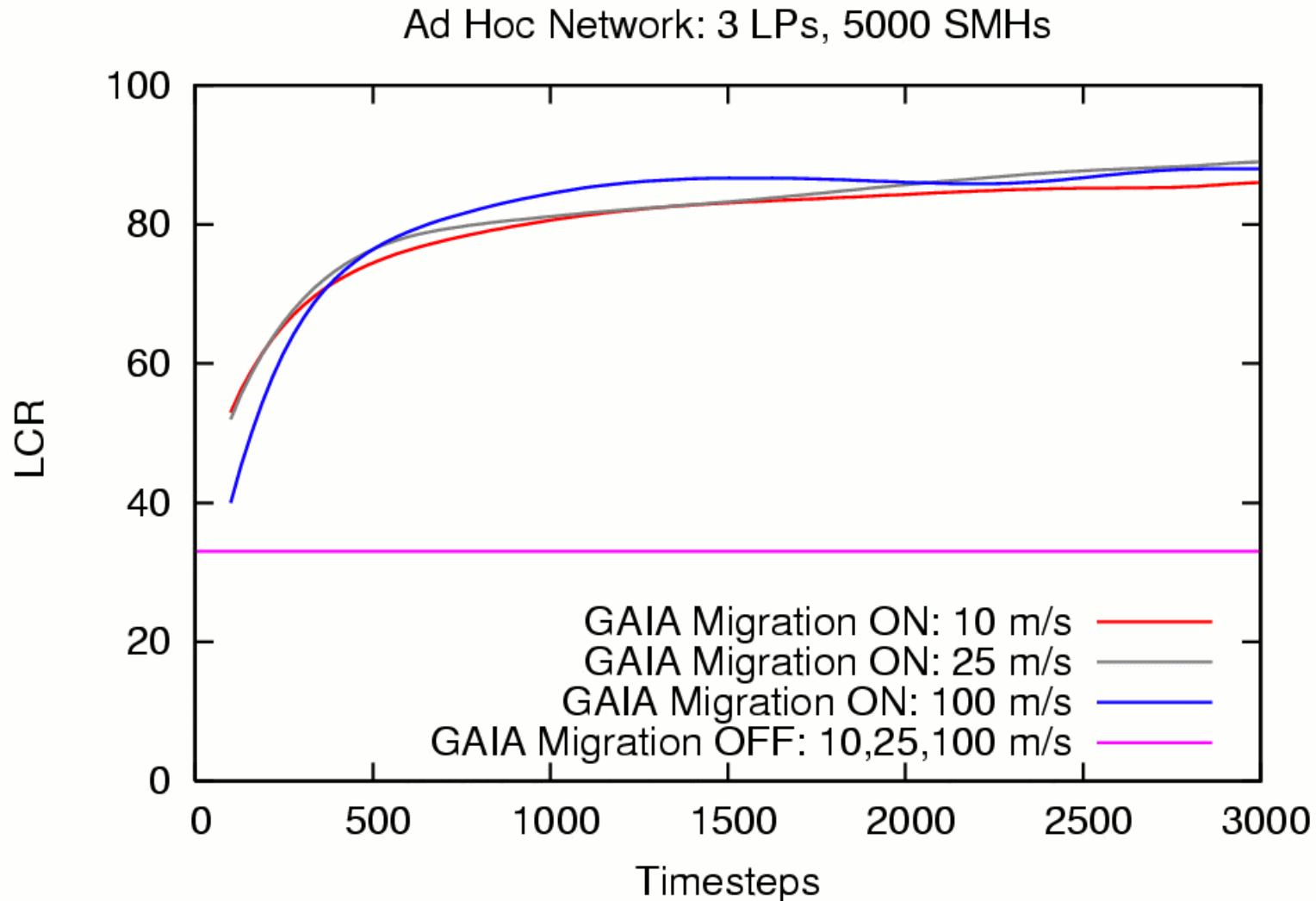
Migration “OFF”

Migration “ON”

# Performance analysis: number of migrations / timestep

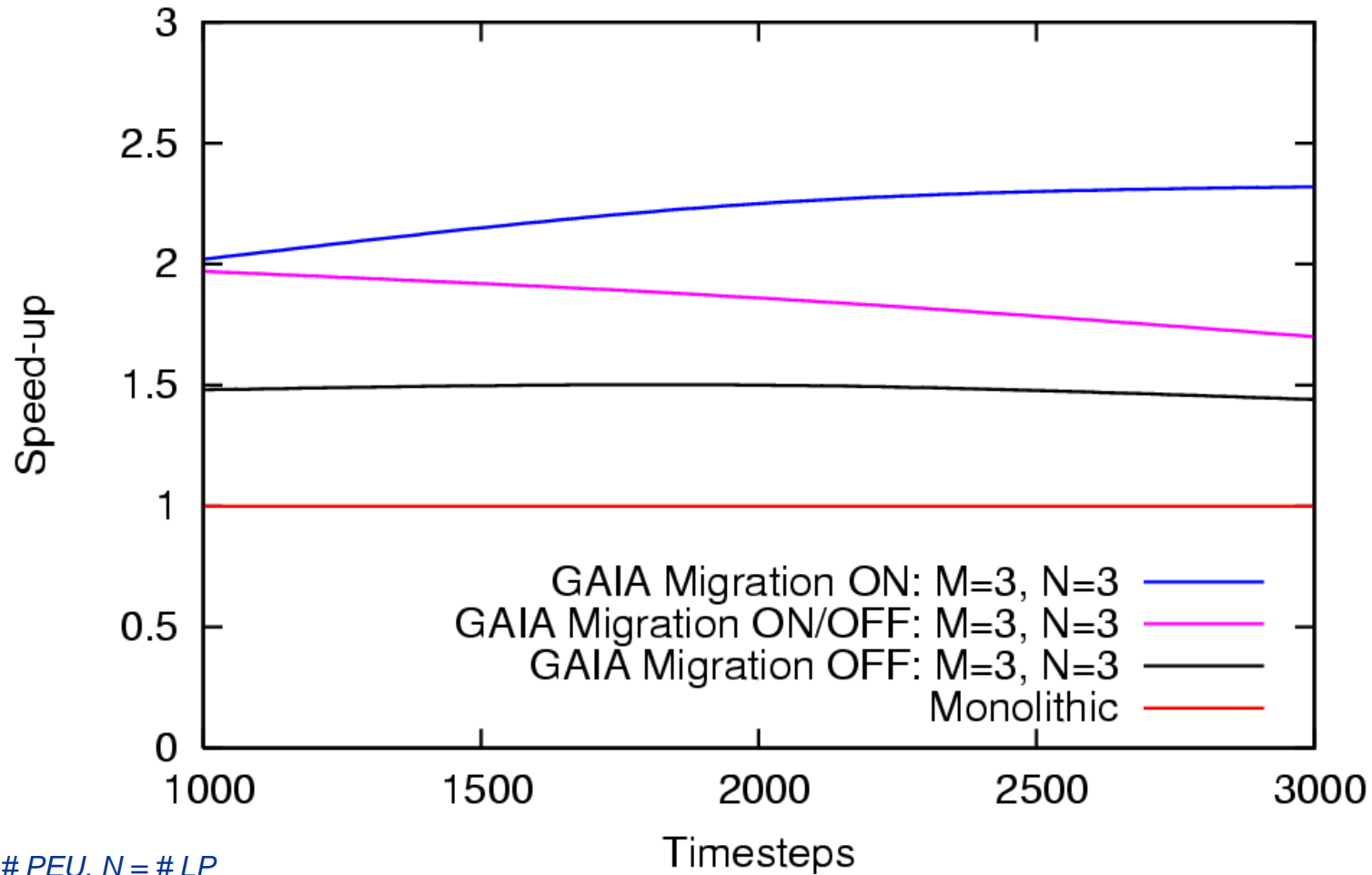


# Performance analysis: Local Communication Ratio



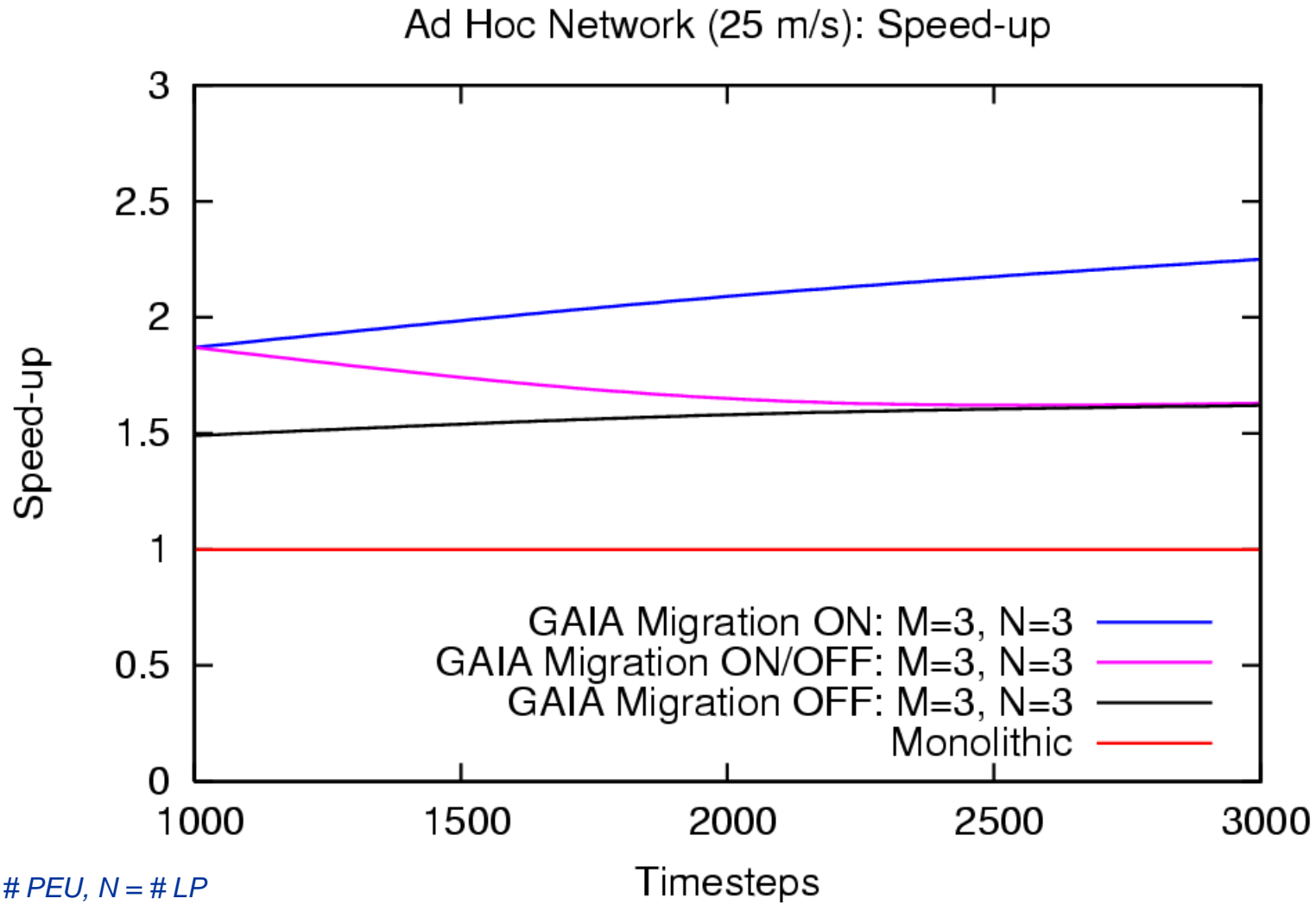
# Performance analysis: speed-up

Ad Hoc Network (10 m/s): Speed-up



$M = \# PEU, N = \# LP$

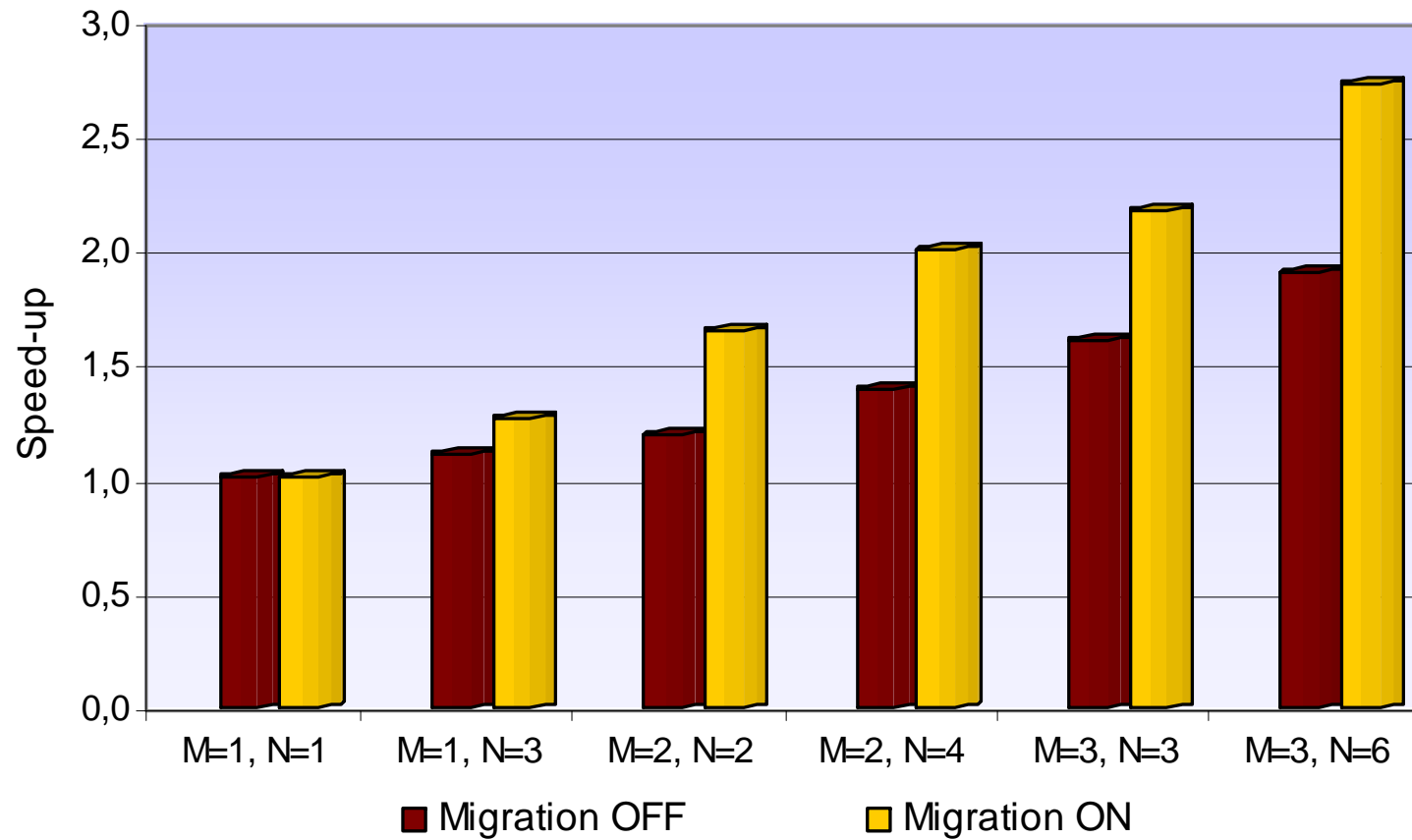
# Performance analysis: speed-up



$M = \# PEU, N = \# LP$

# Performance analysis: speed-up

Ad Hoc (25 m/s): Speed-up



$M = \# PEU, N = \# LP$

# Example: sensor network model implementation

Design of a new energy aware Medium Access Control protocol:

- Up to 40.000 sensors, limited battery resources
- Sensors are **static** (no mobility model)
- Every sensor implements the MAC protocol, no centralization
- 4 different sensor states (active, power saving, listening, died)
- Each sensor implements a “pressure variation” detector and sends broadcast alerts flooding toward a set of detection points
- The energy aware MAC protocol increases the network lifetime managing the sensors’ state (dynamically switching to power-save state the sensors within areas covered by other active sensors)

# Sensor network simulation example

1000 sensors (for this example),  
Limited battery resources

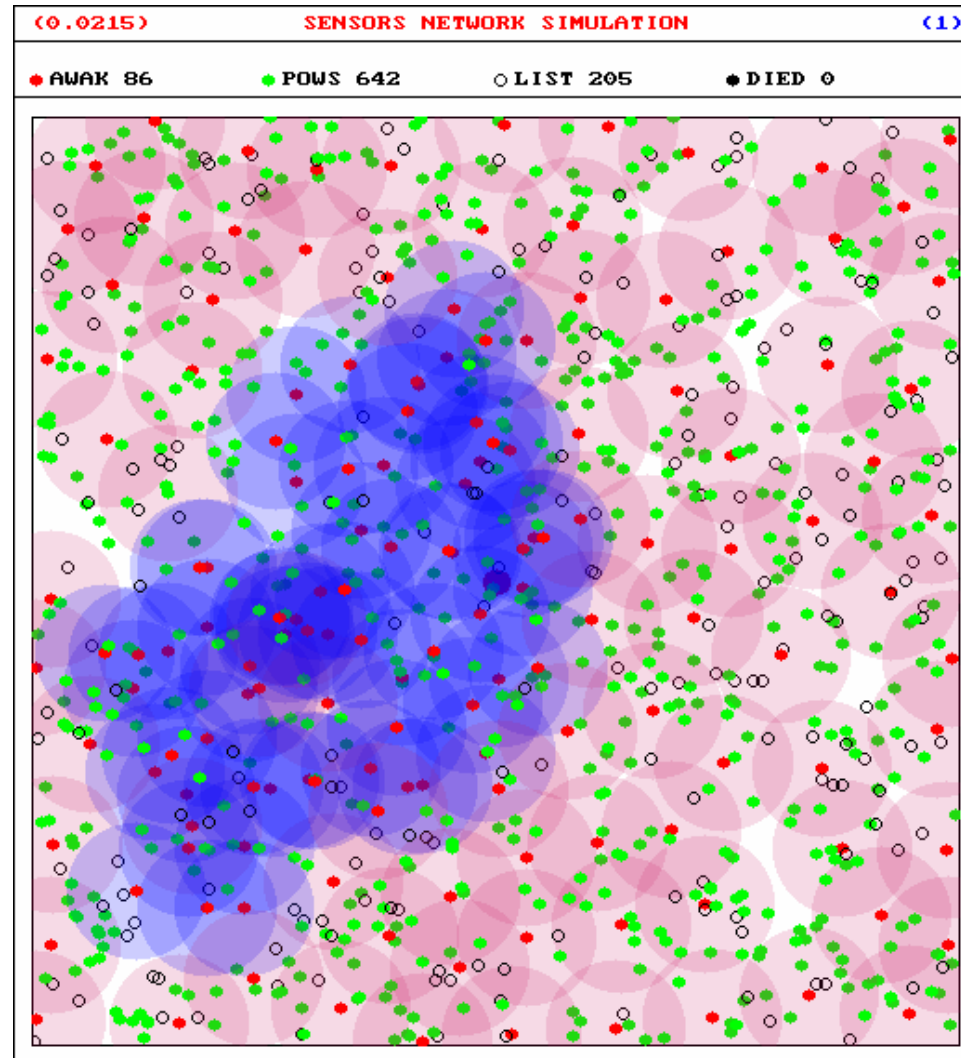
4 different states:

- red = awake
- green = power saving
- white = listening
- black = died

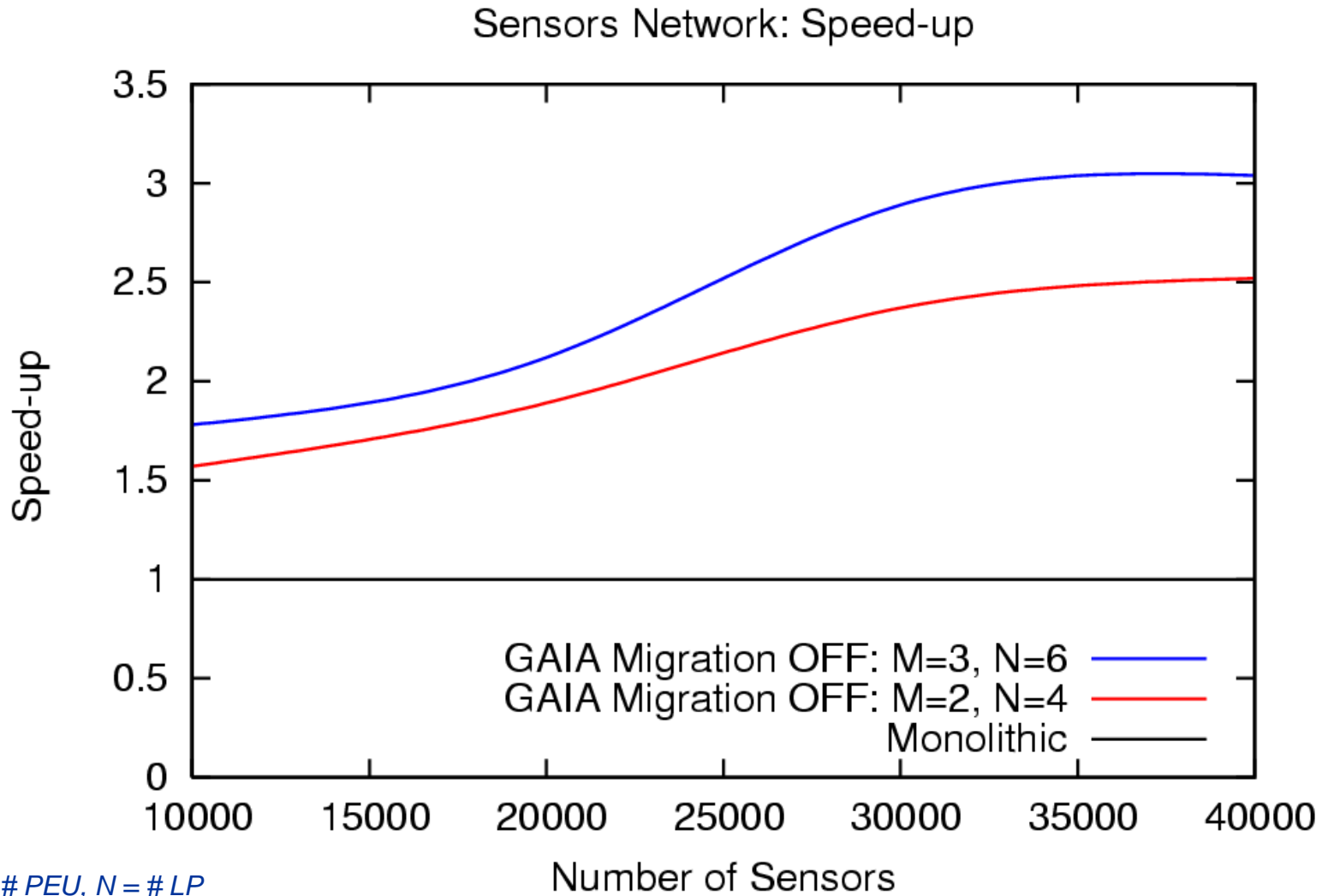
Pink circle =  
active transmitting range

Blue circle =  
alert-message transmission

**GOAL:** extend the network lifetime  
maintaining network connectivity



# Performance analysis: speed-up

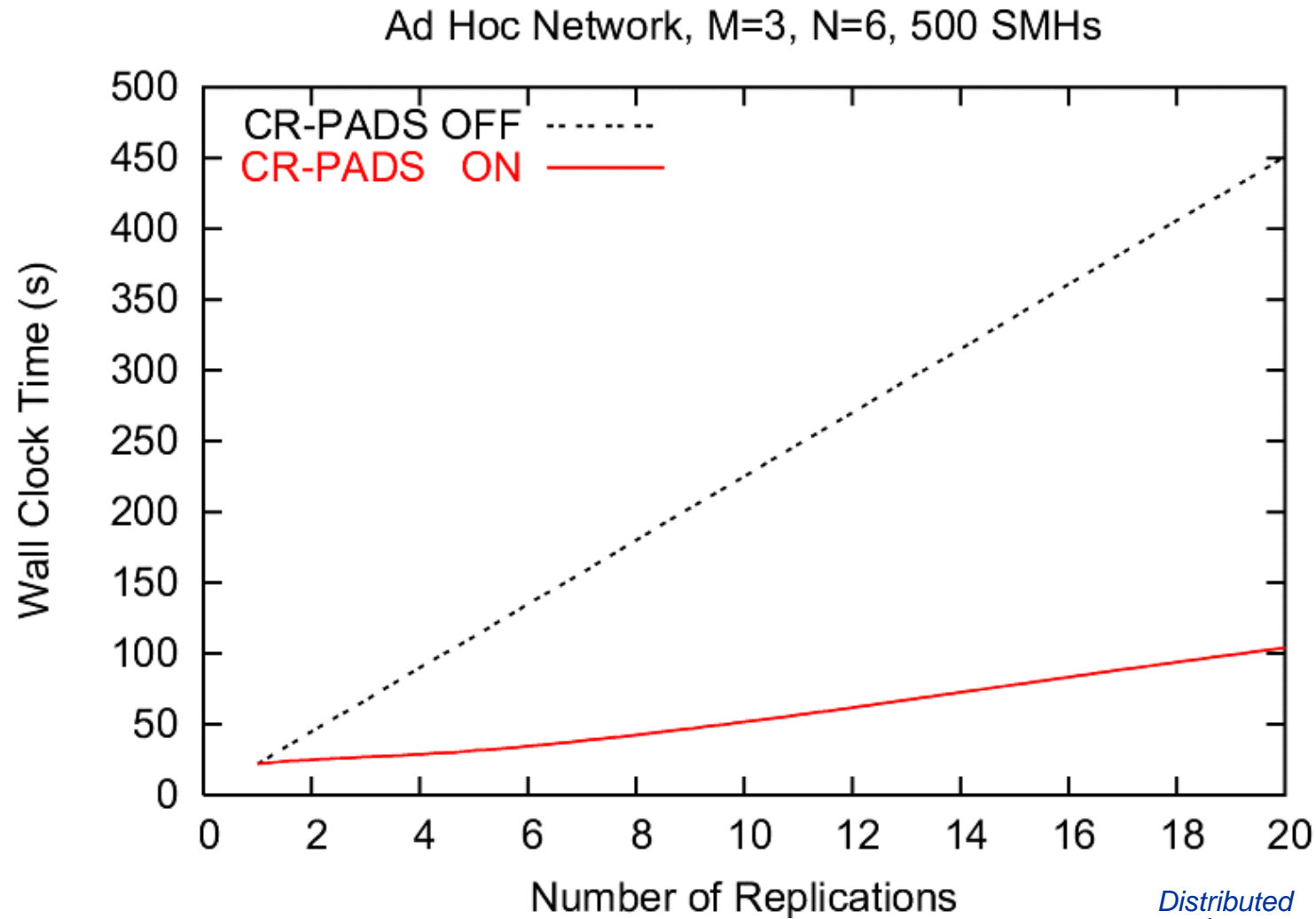


$M = \# PEU, N = \# LP$

# Concurrent Replication of PADS

- Under the Parallel (or Distributed) Discrete Event Simulation (PDES) frequent **synchronizations** are required among the model components
- Every **CPU** swings between computation and **idle**, while the underlying **communication infrastructure** swings between idle and communication periods
- A typical simulation-based investigation requires to collect many independent observations for a correct and significant statistical analysis of results
- **Proposed solution:** launching multiple, independent and concurrent PDES runs over the system, with the goal to reduce idle computation and communication time

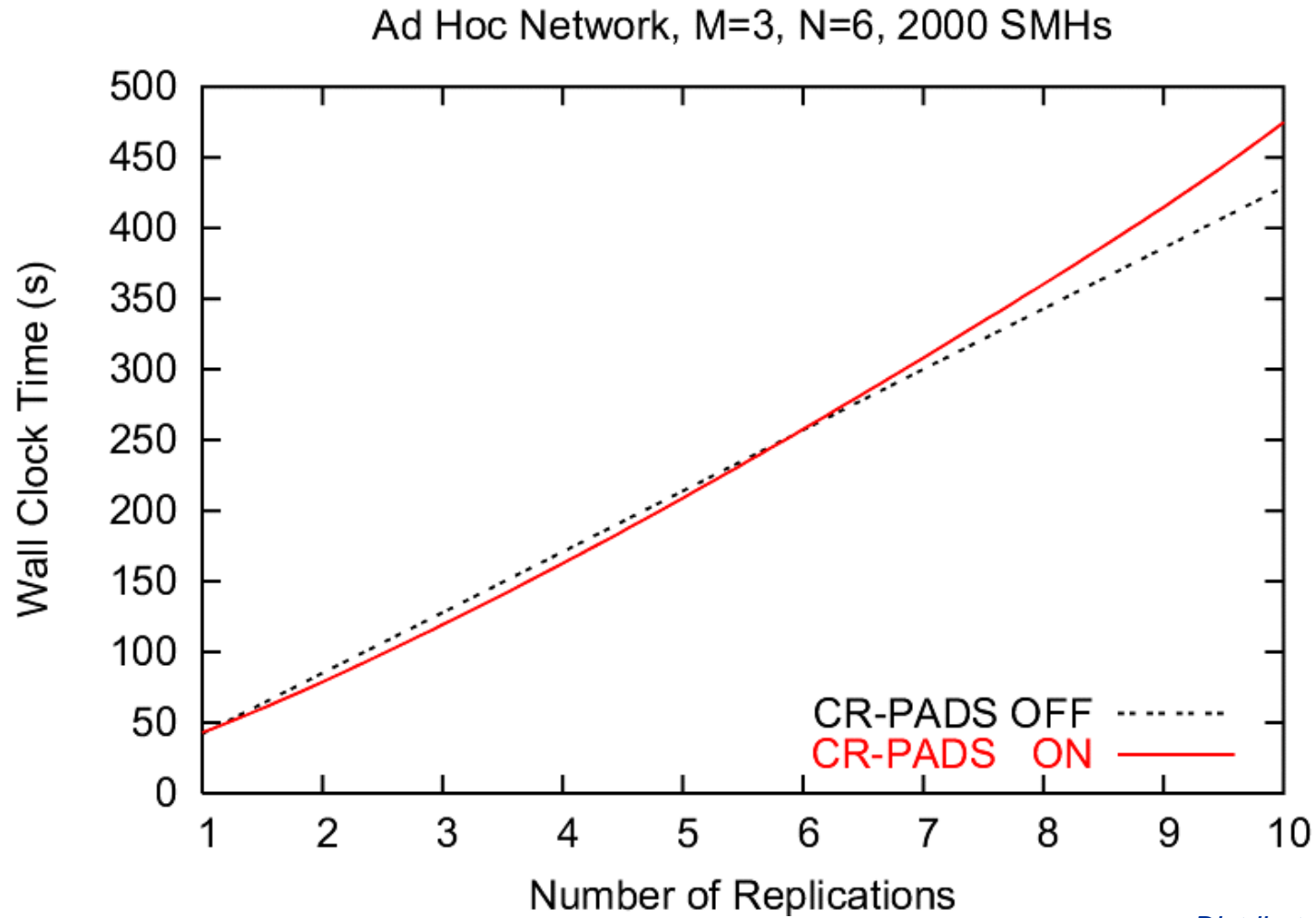
# CR-PADS Performance evaluation



$M = \# PEU$ ,  $N = \# LP$

*Distributed environment*

# CR-PADS Performance Evaluation



$M = \# PEU$ ,  $N = \# LP$

*Distributed environment*

# Internet Games

- In recent years many popular interactive computer games have gained online remote multiplayer functionalities
- Massive Multiplayer Online Role-Playing Games (MMORPGs)
- MMORPGS gaming experience is highly subjected to network performances (communication latency)
- Leveraging from our work in the PADS field we have proposed a new migration based architecture to support massively populated Internet games
- A simulation based performance evaluation of the proposed architecture has demonstrated an increased game **fairness** significantly reducing the latency experienced by gamers

# Conclusions and future work

---

- Design and implementation of the ARTiS middleware
- GAIA: adaptive allocation of model entities in a parallel and distributed simulation
- Examples: ad hoc and sensor network
  
- Current activities and future work:
  - Advanced simulation techniques (simulation replication & cloning)
  - GAIA load balancing and migration heuristics improvement
  - High Performance Computing (HPC) support
  - Design and implementation of new models (scale free-networks, P2P models, detailed 802.11 MAC protocol)
  - IEEE 1516 compatibility

# References

1. L.Bononi, M.Bracuto, G.D'Angelo, L.Donatiello. **Concurrent Replication of Parallel and Distributed Simulations**. To appear in Proceedings of *ACM/IEEE/SCS PADS 2005*
2. L.Gardenghi, S.Pifferi, G.D'Angelo, L.Bononi. **Design and Simulation of a Migration-based Architecture for Massively Populated Internet Games**. Proceedings of *IEEE NIME 2004* (GLOBECOM 2004 Satellite Workshop)
3. L.Bononi, M.Bracuto, G.D'Angelo, L.Donatiello. **ARTIS: a Parallel and Distributed Simulation Middleware for Performance Evaluation**. Proceedings of *ISCIS 2004*
4. L.Bononi, M.Bracuto, G.D'Angelo, L.Donatiello. **A New Middleware for Parallel and Distributed Simulation of Dynamically Interacting Systems**. Proceedings of *IEEE DS-RT 2004*
5. L.Bononi, M.Bracuto, G.D'Angelo, L.Donatiello. **Performance Analysis of a Parallel and Distributed Framework for Large Scale Systems' Simulation**. Proceedings of *ACM/IEEE MSWiM 2004*
6. L.Bononi, G.D'Angelo. **A Novel Approach for Distributed Simulation of Wireless Mobile Systems**. Proceedings of *IFIP-TC6 PWC 2003*
7. L.Bononi, G.D'Angelo, L.Donatiello. **HLA-based Adaptive Distributed Simulation of Wireless Mobile Systems**. Proceedings of *ACM/IEEE/SCS PADS 2003*

# ARTiS: Design and Implementation of an Adaptive Middleware for Parallel and Distributed Simulation

---

**Gabriele D'Angelo**

*[gdangelo@cs.unibo.it](mailto:gdangelo@cs.unibo.it)*

*<http://www.cs.unibo.it/~gdangelo>*



**University of Bologna**  
**Department of Computer Science**

May, 2005