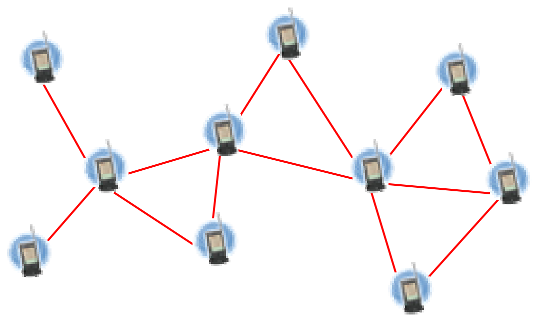
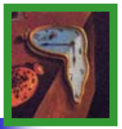
JiST – Java in Simulation Time

- *Transparent* Parallel and *Optimistic* Execution of Discrete Event *Simulations* of MANETs

- discrete event simulations are useful and needed
- but, most published ad hoc network simulations
  - lack *scalability* ~250 nodes; or
  - compromise *detail* packet level; or
  - are of short *duration* few minutes



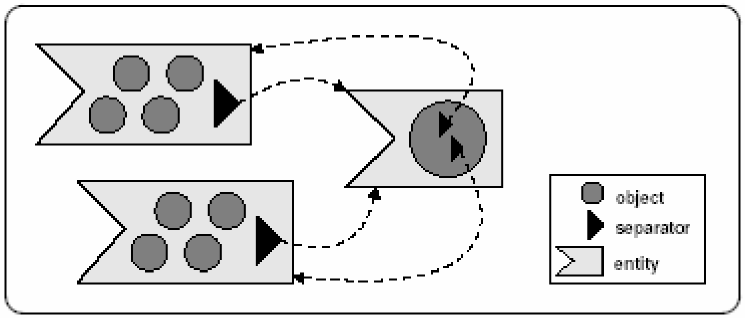
JiST: existing alternatives



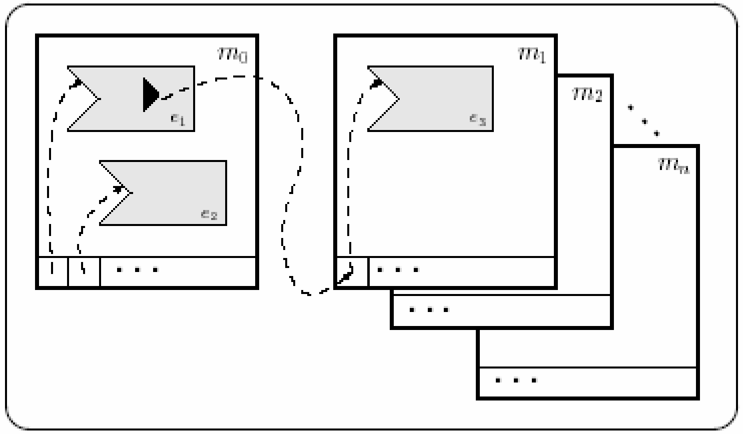
- ns2 is the *gold standard*
  - Tcl-based, with C++ bindings
  - used extensively within research community
  - initially developed for detailed TCP simulations
  - modified to support ad hoc networks
  - processor and memory intensive, sequential
  - max. ~ 250 nodes, $O(n^3)$
- PDNS – parallel distributed ns2
  - perform event loop over Georgia Tech. RTI-KIT
  - requires fast inter-connect
  - helps with memory limits
- OPNet
- Glomosim
  - written in Parsec, a custom C-like language
  - entities map to processes, messages to IPCs
  - “node aggregation” requirement imposes conservative parallelism
  - max. ~10,000 nodes, but on NUMA: Sun SPARCserver 1000, est. $300,000
- custom-made simulators
  - fast, specialized computation
  - lack sophisticated execution, parallelism, *credibility*

JiST: in a nutshell

- achieve *scalability* through
  - parallelism, optimism: maximize execution concurrency
  - state partitioning: split simulation into fine-grained entities
  - transparency: automatic binary rewrite of serial programs
  - genericity: use general-purpose systems language
  - COTS hardware: inexpensive PC clusters



Automatic simulation partitioning



Optimistic parallel execution

JiST: results thus far

- **the "hello world" of event simulations**

```
class MySim implements JistAPI.Entity
{
  private int data = 0;
  public void myEvent()
  {
    JistAPI.sleep(1);
    myEvent();
    System.out.println("myEvent, \
         sim-time="+JistAPI.getTime()+
         " data="+(data++));
  }
}
```

# events	JiST	GlomoSim	Ratio
10^5	0.22s	0.48s	45%
10^6	1.44s	3.18s	45%
10^7	13.55s	30.46s	44%
10^8	130.6s	292.5s	45%

serial throughput increase of 2.2x

- **currently building SWANS atop JiST**
  - **Scalable Wireless Ad hoc Network Simulator**
  - **Java application running in *simulation time***