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

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

<table>
<thead>
<tr>
<th># events</th>
<th>JiST</th>
<th>GlomoSim</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>10^5</td>
<td>0.22s</td>
<td>0.48s</td>
<td>45%</td>
</tr>
<tr>
<td>10^6</td>
<td>1.44s</td>
<td>3.18s</td>
<td>45%</td>
</tr>
<tr>
<td>10^7</td>
<td>13.55s</td>
<td>30.46s</td>
<td>44%</td>
</tr>
<tr>
<td>10^8</td>
<td>130.6s</td>
<td>292.5s</td>
<td>45%</td>
</tr>
</tbody>
</table>

serial throughput increase of 2.2x

- currently building **SWANS** atop JiST
  - **S**calable **W**ireless **A**d hoc **N**etwork **S**imulator
  - Java application running in *simulation time*