

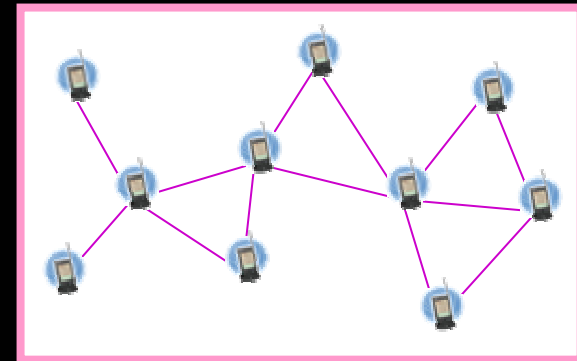


JiST – Java in Simulation Time

Rimon Barr

JiST – Java in Simulation Time

- *Transparent, efficient, scalable, optimistic* execution of discrete event MANET *simulations* over a *standard* virtual machine
- 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 short *duration* few minutes



JiST: existing alternatives

ns2 is the *gold standard*

- Tcl-based, with C++ bindings
- used extensively within community
- written for detailed TCP simulation
- modified to support ad hoc networks
- processor and memory intensive
- sequential; max. ~ 250 nodes, $O(n^3)$

PDNS – parallel distributed ns2

- event loop uses RTI-KIT
- requires fast inter-connect
- helps with memory limits

OPNet

GloMoSim

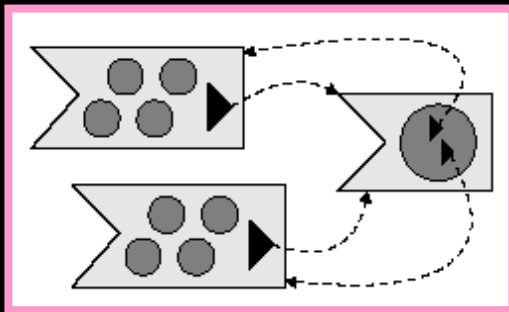
- Parsec-based, a custom C-like language
- entities are memory intensive
- requires “node aggregation,” which imposes conservative parallelism
- high event throughput per processor, but requires shared memory for scale
- shown $\sim 10,000$ nodes, but on NUMA (SPARCserver 1000, est. \$300,000)

custom-made simulators

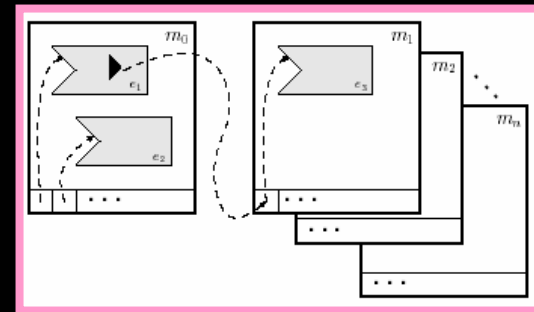
- fast, specialized computation
- lack sophisticated execution, parallelism
- lack *credibility*

The gist of JiST

- achieve **scalability** through
 - **parallelism, optimism**: maximize execution concurrency
 - **state partitioning**: simulation split into fine-grained entities
 - **transparency**: automated binary rewrite of serial programs
 - **genericity**: general-purpose, optimized systems language and runtime
 - **COTS hardware**: cluster of inexpensive PCs



Automatic simulation partitioning



Optimistic parallel execution

JiST: status

- the “hello world” of event simulations

```
class MySim implements JistAPI.Entity {  
    public void myEvent() {  
        JistAPI.sleep(1);  
        myEvent();  
        System.out.println("hello world, \\  
            time="+JistAPI.getTime());  
    }  
}
```

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

	Memory	Limit
JiST	40 bytes	> 10 ⁶ entities
Parsec	27800 bytes	~ 10 ⁴ entities

JiST scales to more entities per process

# events	JiST	GloMoSim	Ratio
10 ⁵	0.044s	0.435s	10%
10 ⁶	0.262s	2.938s	9%
10 ⁷	2.301s	28.04s	8%
10 ⁸	22.48s	278.4s	8%

serial throughput increase of 12x