

**JiST:
Java in Simulation Time**

for

**Scalable Simulation of
Mobile Ad hoc Networks**



Rimon Barr and Zygmunt J. Haas

<barr@cs.cornell.edu> <haas@ece.cornell.edu>

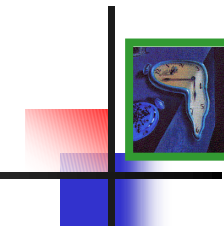
Wireless Network Laboratory

Cornell University

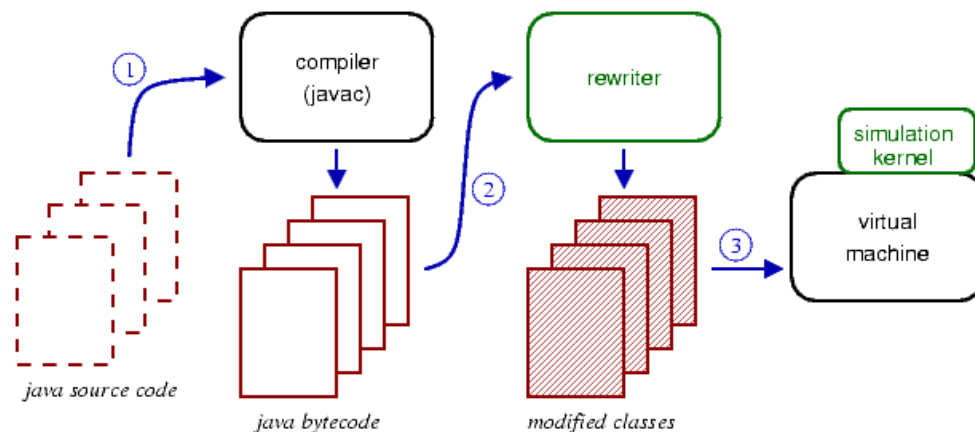
3rd IRTF Ad hoc Network Scalability Meeting

12 November 2003

JiST – Java in Simulation Time



- Simulation **scalability** is important
- JiST is a new approach to building simulators: it **extends** the Java object model and execution semantics, and leverages the Java virtual machine to run simulations **efficiently** and **transparently**.
- Bring simulation semantics to modern, popular language
 - run plain-Java programs in **simulation time**
 - merges systems and languages approaches to simulator construction



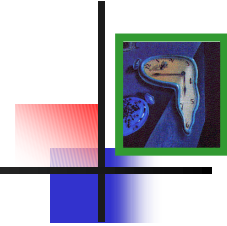
Simulation throughput

5x10 ⁶ events	time (sec)	vs. JiST
JiST	0.97	-
Parsec	1.91	2.0x
ns2-C	3.26	3.4x
GloMoSim	9.54	9.8x
ns2-Tcl	76.56	79.0x

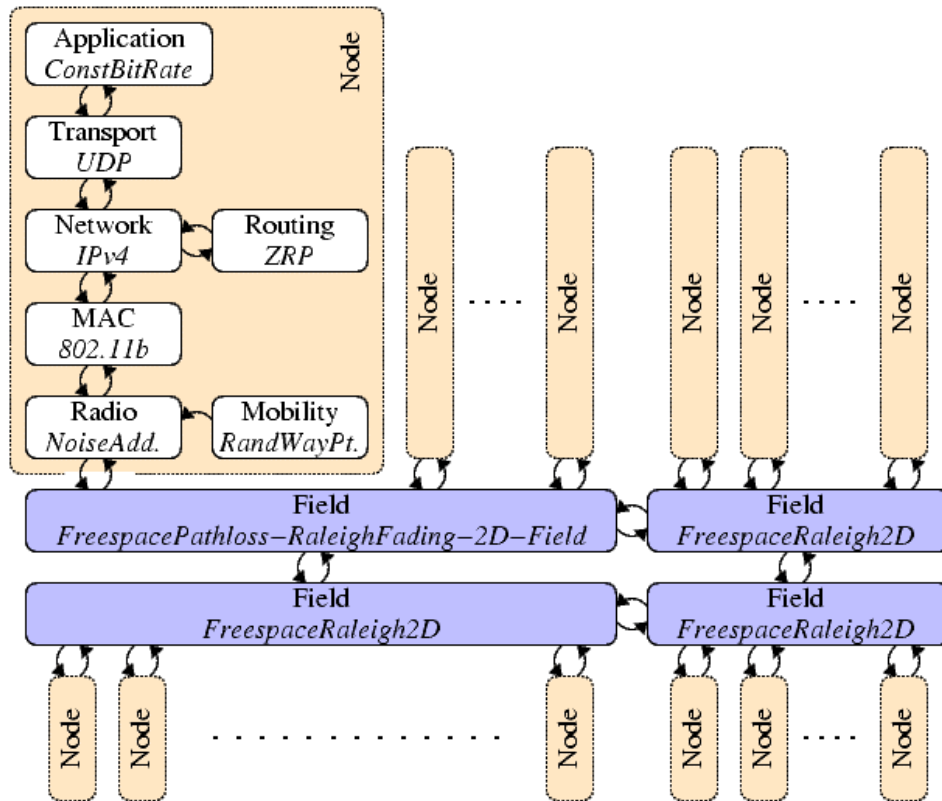
Simulation footprint

memory	per entity per event	
JiST	36 B	36 B
GloMoSim	36 B	64 B
ns2 *	544 B	40 B
Parsec	28536 B	64 B

SWANS



- Scalable **W**ireless **A**d hoc **N**etwork **S**imulator
 - runs **standard Java network applications** over simulated networks
 - uses **hierarchical binning** for efficient signal propagation



sim. stack

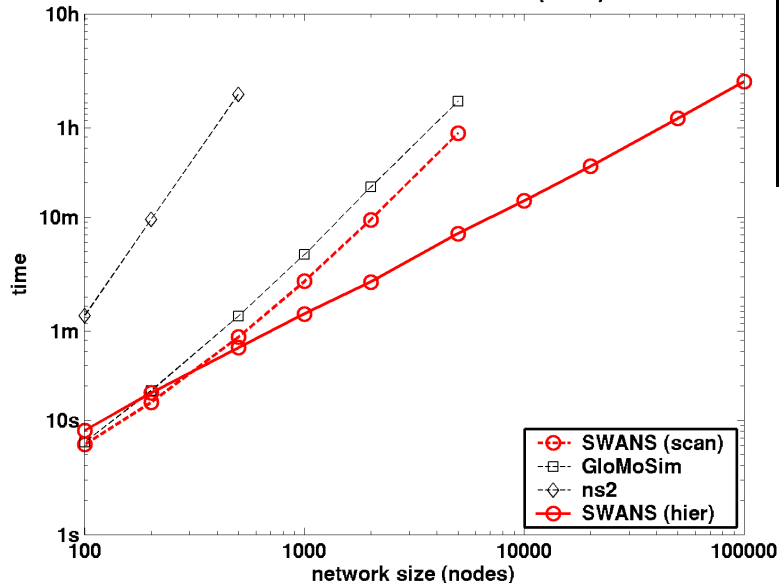
App	files	classes	lines
SWANS			
JiST	25	92	11019
SWANS	65	145	17087
Other	25	52	3808
	115	289	31914

function	implementation
application	- heartbeat; any Java network application
transport	- UDP; TCP [Tamtoro]
network	- IPv4
routing	- ZRP; DSR [Viglietta]; AODV [Lin]
link	- 802.11b; naïve; wired
placement	- random; input file
mobility	- static; random waypoint; input file
interference	- independent, ns2; additive, GloMoSim
fading	- zero; Raleigh; Rician
pathloss	- free-space; two-ray
propagation	- linear scan, ns2;
algorithm	- flat binning, GloMoSim; hierarchical binning

SWANS performance



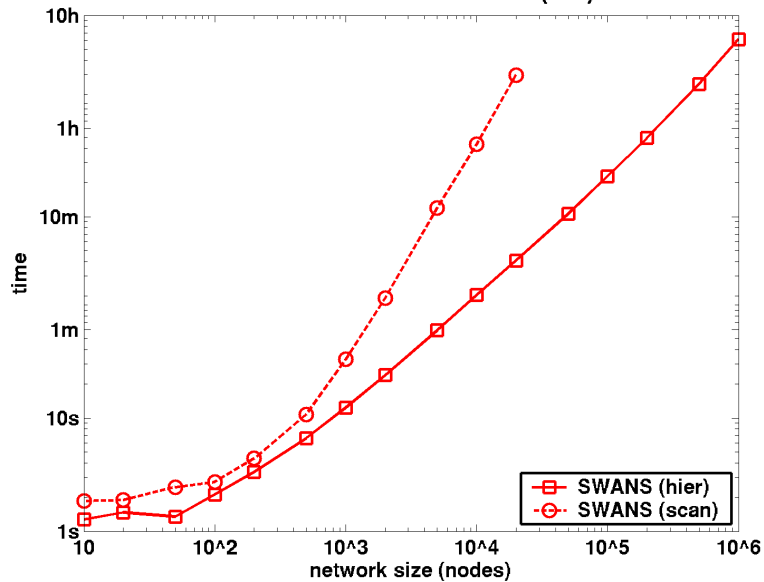
Time for NDP simulation (15m)



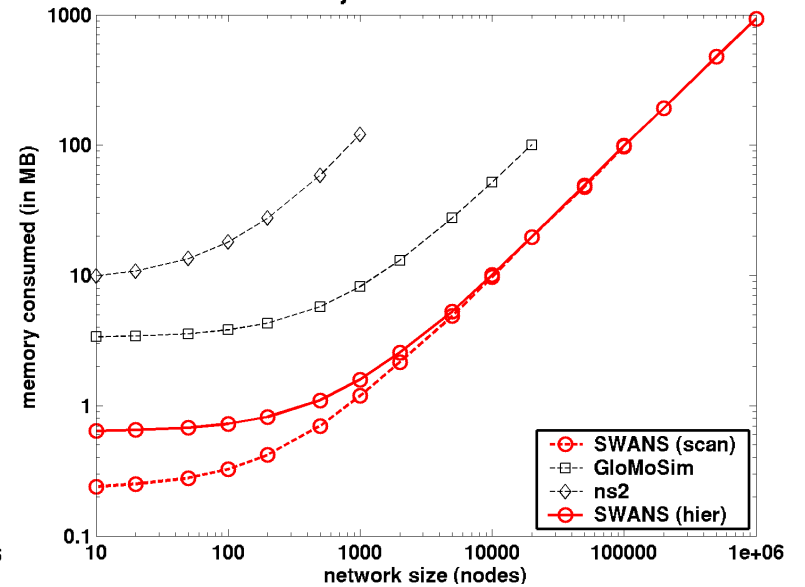
15m nodes	ns2		GloMoSim		SWANS		SWANS-hier	
	time	memory	time	memory	time	memory	time	memory
500	7136.3 s	58761 KB	81.6 s	5759 KB	53.5 s	700 KB	43.1 s	1101 KB
5000			6191.4 s	27570 KB	3249.6 s	4887 KB	430.0 s	5284 KB
50000						47717 KB	4377.0 s	49262 KB

2m nodes	SWANS-hier			
	10,000	100,000	1 million	1
initial memory	13 MB	100 MB	1000 MB	1.0 KB
avg. memory	45 MB	160 MB	1200 MB	1.2 KB
time	2 min	25 min	6 hours	20 ms

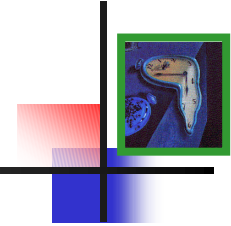
Time for NDP simulation (2m)



Memory for NDP simulation



benefits of the jist approach



- more than just performance...
- **application-oriented** benefits
 - **type safety** source and target statically checked
 - **event types** not required (implicit)
 - **event structures** not required (implicit)
 - **debugging** dispatch source location and state available
- **language-oriented** benefits
 - **Java** standard language, compiler, runtime
 - **garbage collection** cleaner code, memory savings
 - **reflection** script-based simulation configuration
 - **safety** fine grained isolation
 - **robustness** no memory leaks, no crashes
- **system-oriented** benefits
 - **IPC** no context switch, no serialization, zero-copy
 - **Java kernel** cross-layer optimization
 - **rewriting** no source-code access required
 - **distribution** provides a single system image abstraction
 - **concurrency** model supports parallel and speculative execution
- **hardware-oriented** benefits
 - **cost** COTS hardware and clusters
 - **portability** runs on everything
- **simulation research platform**