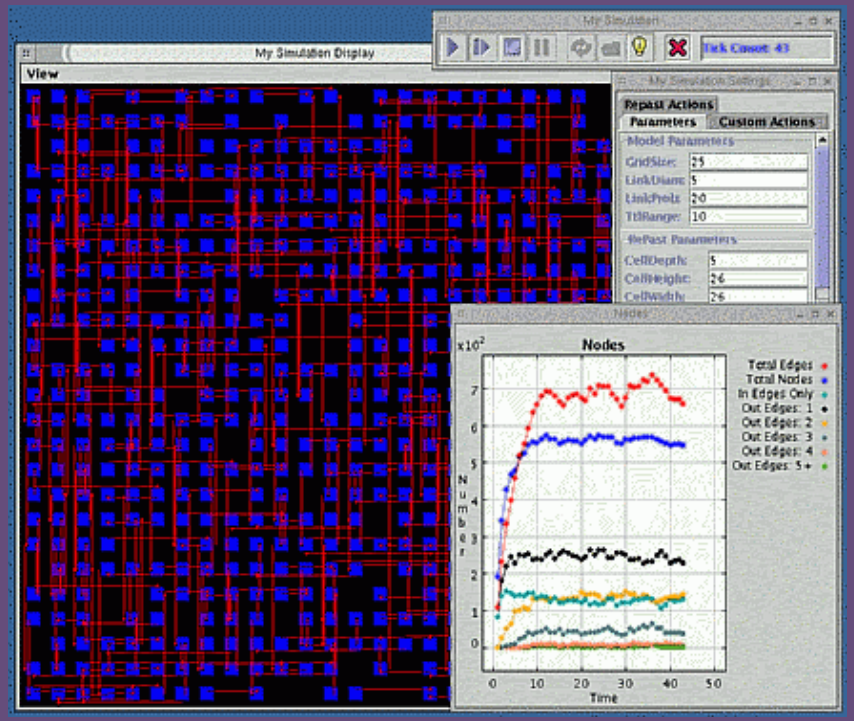
Local Knowledge Networking

Owen Densmore

Sun Labs

http://sunlabs.eng/~owen/proj/

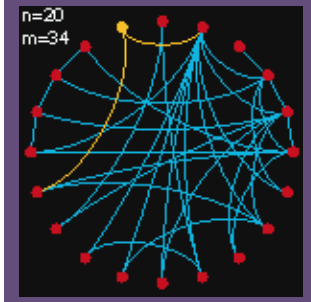


March 26 2001

Background: Two Forces

1) P2P: Very active but poorly understood

2) Complex Systems: Whole >> Sum of Parts

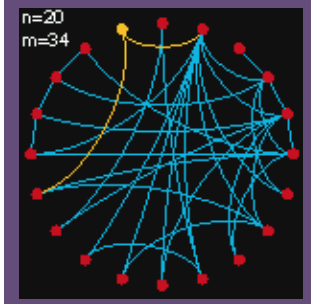


Background: Two Forces

1) P2P: Very active but poorly understood

- OReilly P2P Conference: Currently
  - Server: Napster
  - Serverless: Gnutella, Freenet
  - Infrastructure: Clip2 analysis & "Super Peer"
- Sun: SunLabs-Jxta Peer Intrest Group
- Beyond Filesharing: Emerging New Infrastructure

2) Complex Systems: Whole >> Sum of Parts

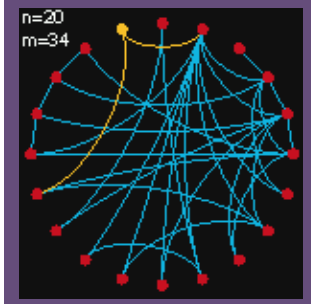


Background: Two Forces

1) P2P: Very active but poorly understood

- OReilly P2P Conference: Curently
  - Server: Napster
  - Serverless: Gnutella, Freenet
  - Infrastructure: Clip2 analysis & "Super Peer"
- Sun: SunLabs-Jxta Peer Intrest Group
- Beyond Filesharing: Emerging New Infrastructure

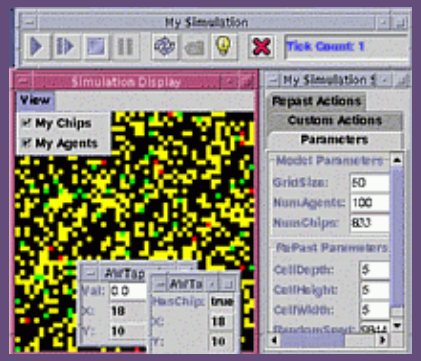
2) Complex Systems: Whole >> Sum of Parts

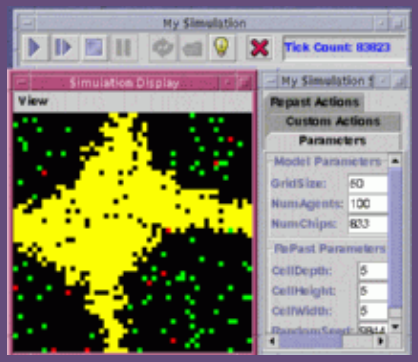


- **Quote: Collective behavior of large number of individuals drastically different from small scale counterpart due to "interesting" interactions among components**
- Statistical vs Deterministic
- Multi-Agent Simulation Plays Major Role

Proposal: Local Knowledge Networking

1) Implement a Serverless Peer with Jxta
   - Session Establishment
   - Short Search Path
2) Analyse using Agent Simulation
   - Session Statistics
   - Search Path Length

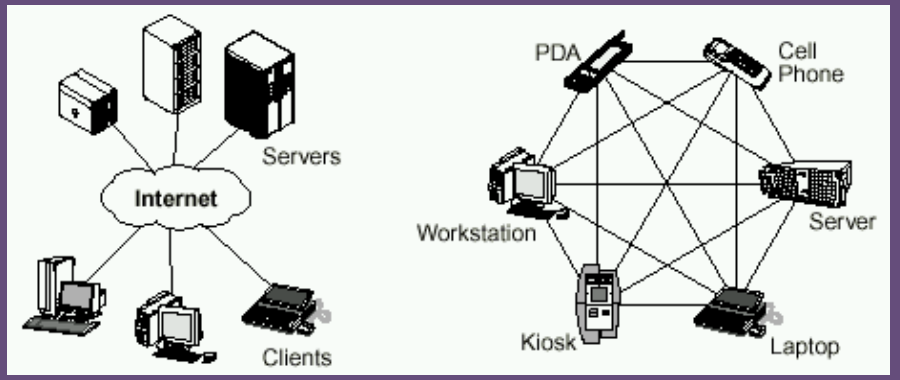




[Demo: Termites]

Why Does Sun Care

- Peer/Small = Disruptive Technology [Innovator's Dilemma]
  - Ex: Appliance Servers, Grid Software, Peer Servers & Clients
  - Sun: Cobalt, Sun GridEngine, LOCKSS, InfraSearch
  - Jxta: Interested in Analysis and Prototypes



- Network: Metcalf's Law
  - Web Imbalance: Hub/Spoke
  - Network Broken:
    - DHCP/DynDNS, MCast, IPSec, IPv6
  - Peer Technology: Return to Combinatorial

Questions to Answer

1) Is Local Knowledge session establishment feasible?
2) If so, how? If not, how fix?
3) Can Peer dynamics create connected network?
4) Can Peer session management scale?
5) How visualize such a vast Peer network?
6) How simulate before building?
7) How model & implement Trust & Reputation?

Project Approach

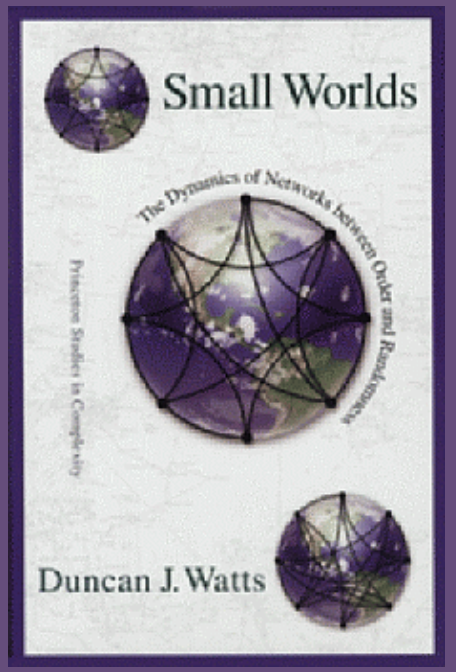
- Java Prototype for Deployability
  - PeerCasting: Rings of Neighbors, optional MCast.
  - Prototype: Me2Me or Serverless Server.
- Java RePast Simulation Framework
  - "Rings of Peers" w/ Frequent Changes in Topology
  - Test via Small Worlds Diameter/Clustering
  - Two Simulations: Session Scalability & Search Path Length

Risks and Issues

- Peer Modeling: Can a Peer system be effectively simulated?
- Size: Is the net too massive for these techniques?
- Local Knowledge: Are local knowledge systems realistic?
- Fixable: If not, can they be fixed.
- Jxta: Is jxta the right platform? Are there fallbacks?
- Multicast: Is project still valid if multicast widely available?

Theory Overview

- Cartography & Measurement
- Small Worlds: Six Degrees of Separation
  - Sparse, Clustered, Slighly Random
  - Simulations: Clustering vs Diameter
- Dynamic Network Lifecycle
  - From Exponential to Power-Law
  - Authorities, Hubs
- Local Knowledge
  - The Maze from Inside



[Books & Papers]

Theory Detail: Data - Cartography

- CAIDA: Cooperative Association for Internet Data Analysis
  - Java Cartographic Tools for Network Analysis

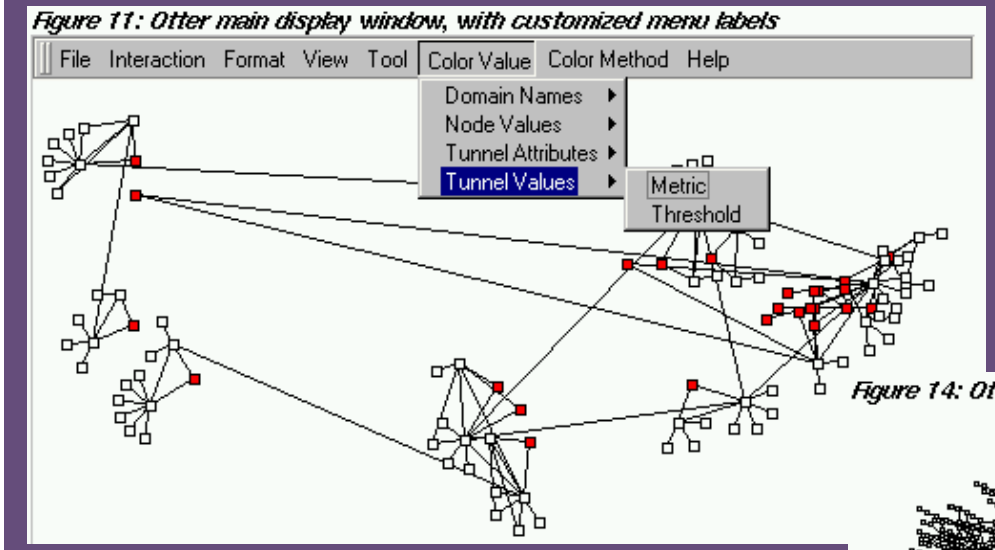


Figure 11: Otter main display window, with customized menu labels

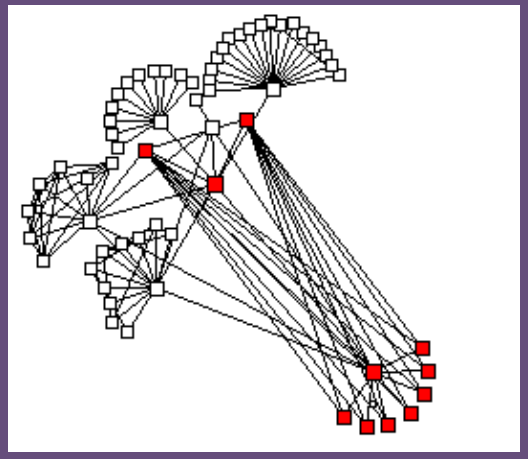
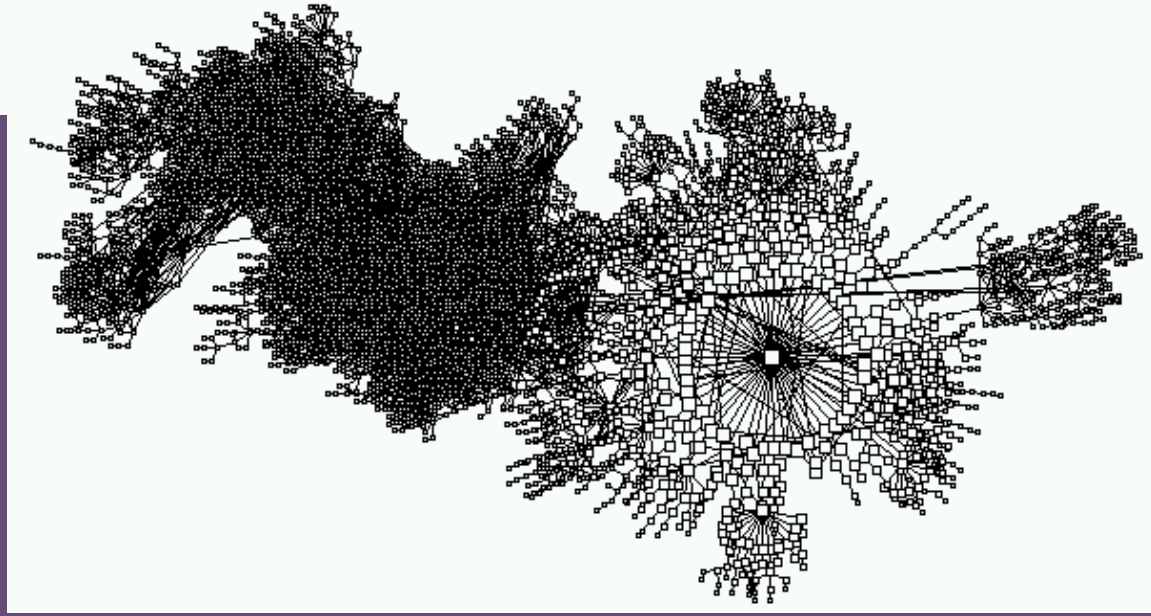


Figure 14: Otter visualizing skitter-gathered topology (RTT) data (approximately 30,000 nodes)



Theory Detail: Small Worlds

- Six Degrees: Milgram 1967 paper and experiment
- Watts/Strogatz 1998 Paper, 1999 Book
- Theory
  - Sparse, Clustered, Slightly Random
  - Yields Small Diameter

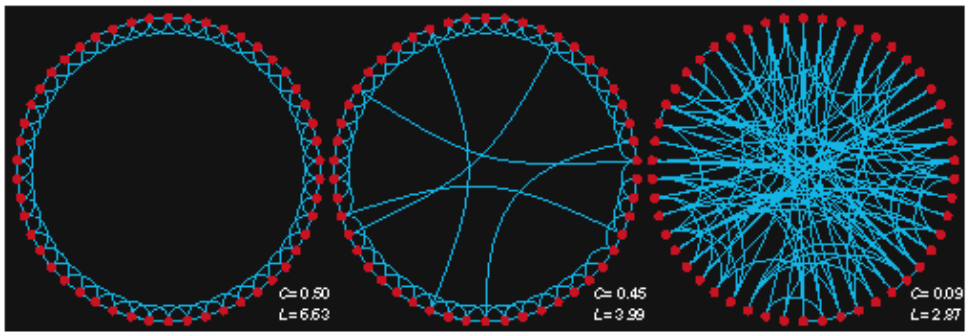


Figure 1. Watts–Strogatz model interpolates between a regular lattice *(left)* and a random graph *(right)*. Randomly rewiring just a few edges *(center)* reduces the average distance between nodes, *L*, but has little effect on the clustering coefficient, *C*. The result is a "small-world" graph.

Theory Detail: Small Worlds [Cont.]

- Simulation Model
  - Clustering & Diameter vs P(k)
  - Yields Early Small Diameter

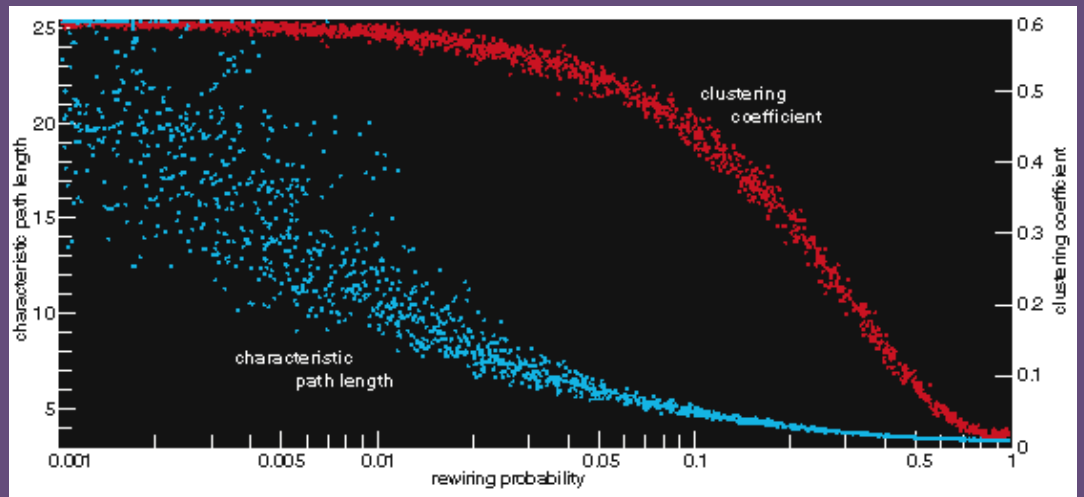


Figure 2. As the probability of rewiring increases in the Watts–Strogatz model, the characteristic path length falls off long before the clustering coefficient drops. Results are from 2,000 random graphs, each with 300 vertices and 900 edges.

Theory Detail: Network Dynamics

- Network Growth
  - Barabasi, Albert, Jeong
  - Start w/ M Nodes, Grow to N.
  - Probabilistic Linking w/ Existing Nodes
  - Produces Realistic Net
    - Power Law
    - Hubs & Authorities

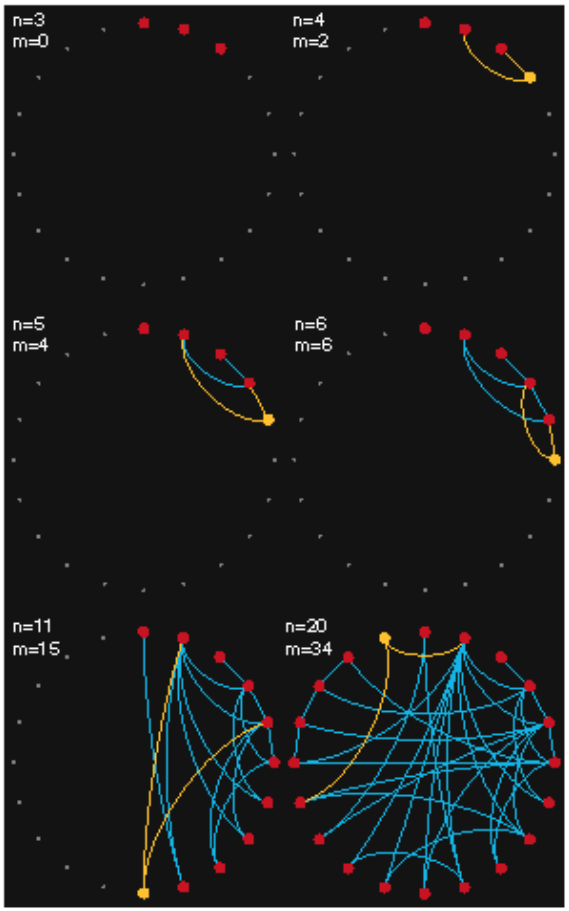
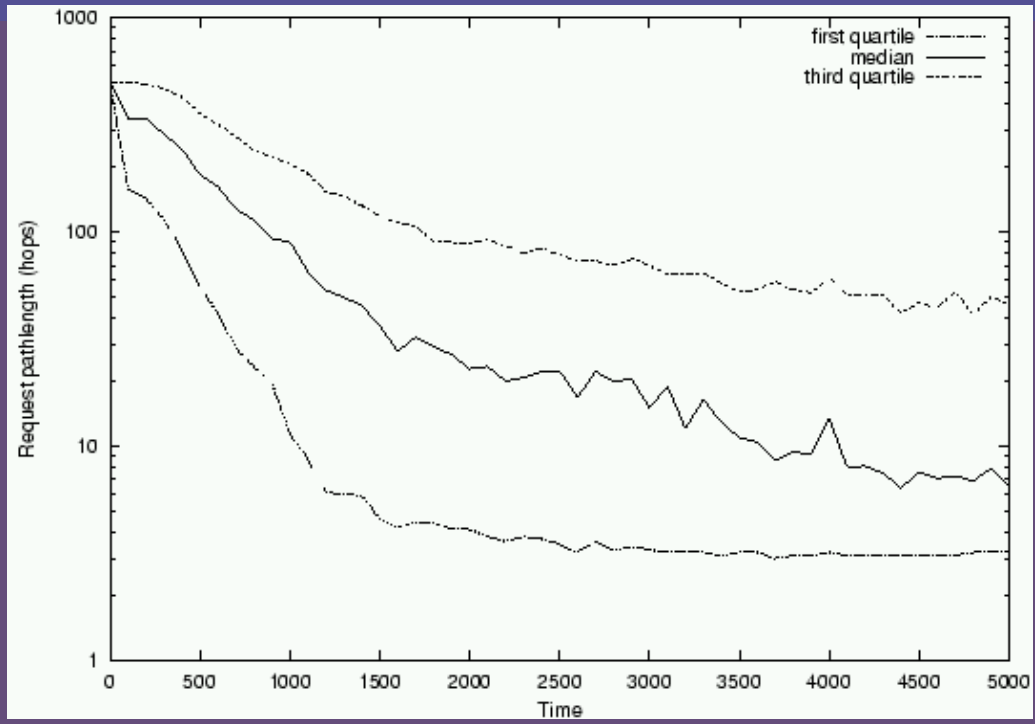


Figure 3. Model of Barabási, Albert and Jeong grows a graph by adding both vertices and edges. In the example shown here, each stage adds one new vertex and two new edges *(yellow)*.

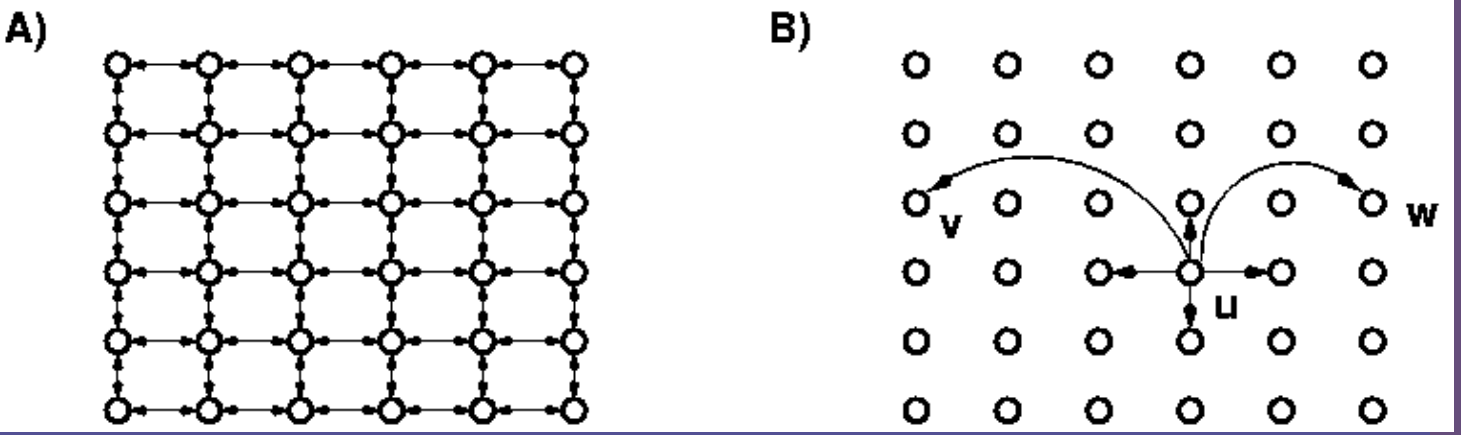
Theory Detail: Finding Sort Paths

- Navigation in a Small World
  - T. Hong: Freenet



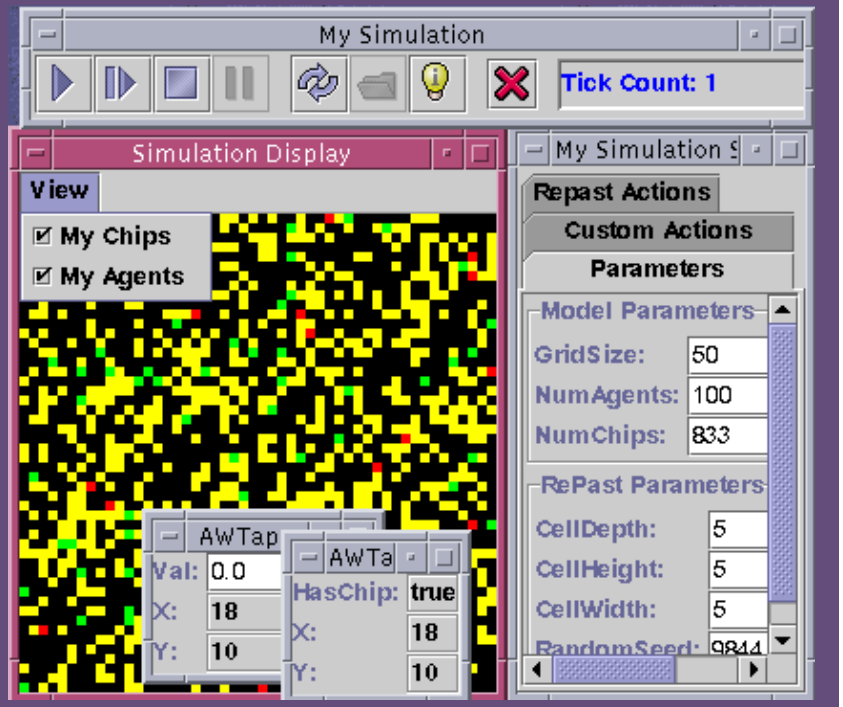
  - Kleinberg: Local Knowledge Routing

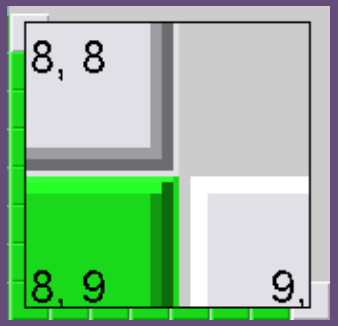
Figure 1: (A) A two-dimensional grid network with $n = 6$, $p = 1$, and $q = 0$. (B) The contacts of a node u with $p = 1$ and $q = 2$. v and w are the two long-range contacts.



Practice: RePast Simulation

- Models: Agents + Spaces
- Agents: Active Elements
- Spaces: Geometry [Torus vs 2D]
- Displays: Layers
- Analysis: Graphs & Histograms
- Pattern Oriented:
  - Swarm "Standard" APIs
  - Imports Several Libraries
  - Lens Visualization Library?





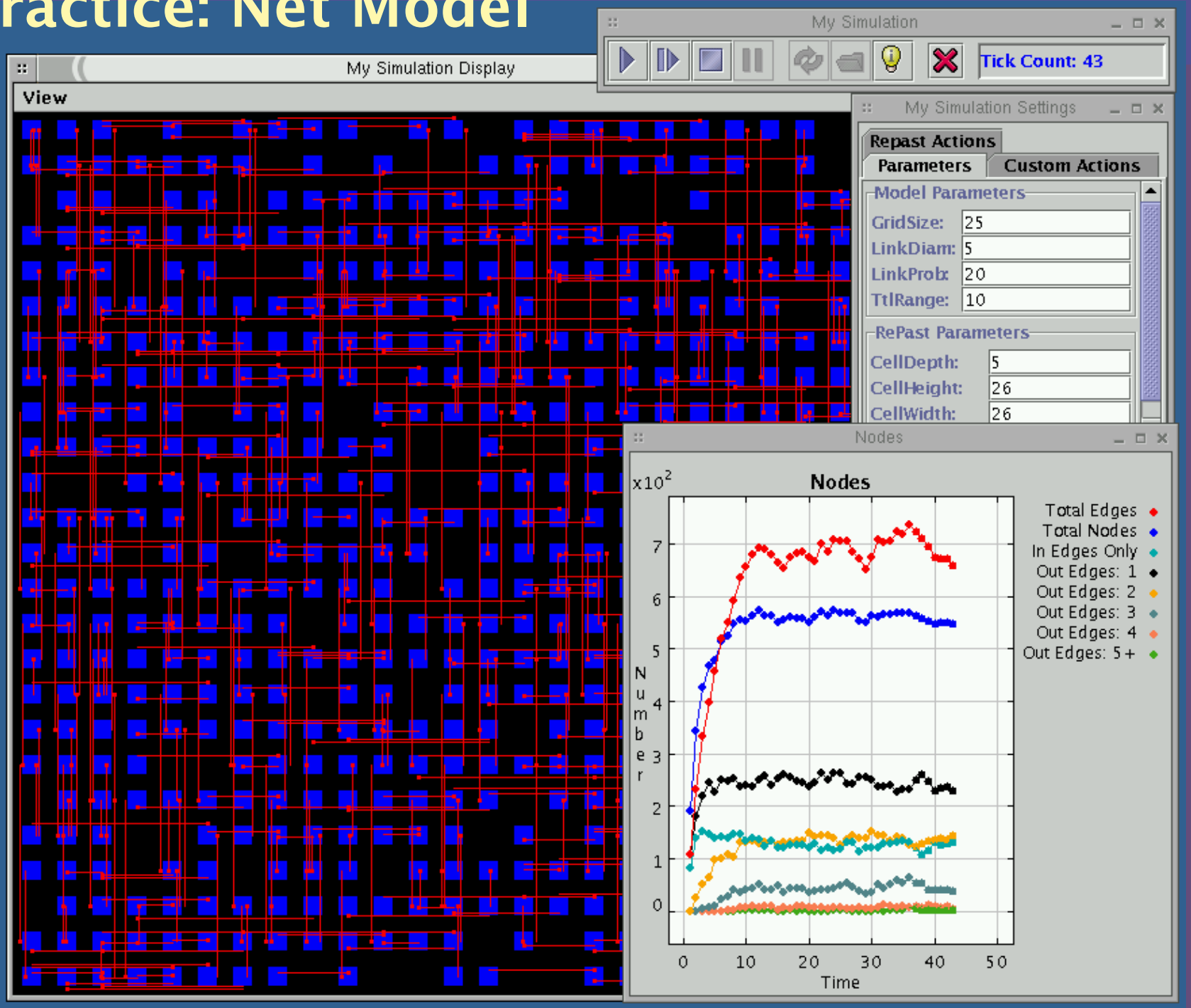
[Demo: Net]

Practice: RePast - Model & Agent Classes

```
import java.awt.Color;
import uchicago.src.sim.engine.*;
import uchicago.src.sim.space.Object2DTorus;
import uchicago.src.sim.gui.DisplaySurface;
import uchicago.src.sim.gui.Value2DDisplay;
import uchicago.src.sim.gui.ColorMap;
import uchicago.src.sim.gui.Object2DDisplay;
import uchicago.src.sim.util.Random;

public class MyModel extends SimModelImpl {
    private Schedule schedule;
    private DisplaySurface dsurf;
    private Object2DTorus space;
    private Object2DTorus chips;
    private int numChips;

    private int numAgents;
    private MyAgent[] agents;
    private int gridSize;

    private void buildModel() {
            space = new Object2DTorus(gridSize, gridSize);
            chips = new Object2DTorus(gridSize, gridSize);
            agents = new MyAgent[numAgents];

            for (int i = 0; i < numAgents; i++) {
                int x, y;
```

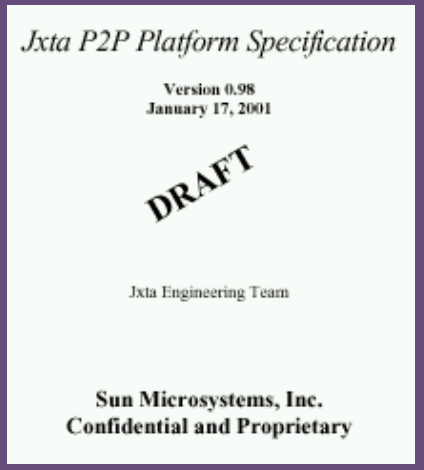
```
Import java.awt.Color;
import java.util.Vector;
import java.awt.Point;
import uchicago.src.sim.gui.Drawable;
import uchicago.src.sim.gui.SimGraphics;
import uchicago.src.sim.space.Object2DTorus;
import uchicago.src.sim.util.Random;

public class MyAgent implements Drawable {
    private int x, y;
    private Object2DTorus chips;
    private Object2DTorus space;
    private boolean haveChip = false;

    public MyAgent(Object2DTorus chips, Object2DTorus spa
            this.chips = chips;
            this.space = space;
    }

    public void execute() {
            wiggle();
            if (onChip())
                if (haveChip) {
                        do {wiggle();} while (onChip());
                        dropChip();
                        jump();
```

Practice: Net Model



Deliverables

- Prototype Application Using Jxta Framework
- Two Java Simulations for Application:
  - Session Scaling
  - Search Path Length
- Critique/Experiences Paper & Talk on Early Jxta Use.
- Extra Credit: Trust & Reputation



Project Status

- Jxta: Team Interactions, 0.98 Spec
- Application: P2P Interest Group Formation
- Simulation:
  - Initial Network Simulation
  - Visualization Issues
- Folks: Complexity Lunch, LOCKSS, SFI BusNet
- SFI: School, BOF, Panel, Workshop
- Project Page:

  http://sunlabs.eng/~owen/proj/