Stress Testing Plan and Network Resilience

Leonard Kleinrock

Professor, Computer Science, UCLA

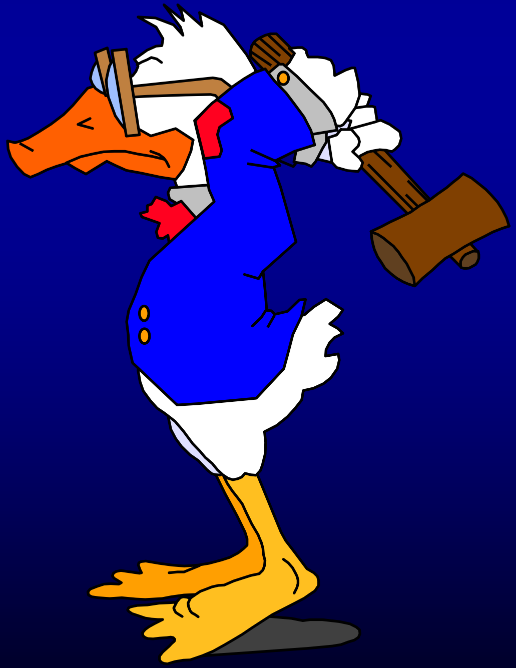
lk@cs.ucla.edu

ONR Program Review

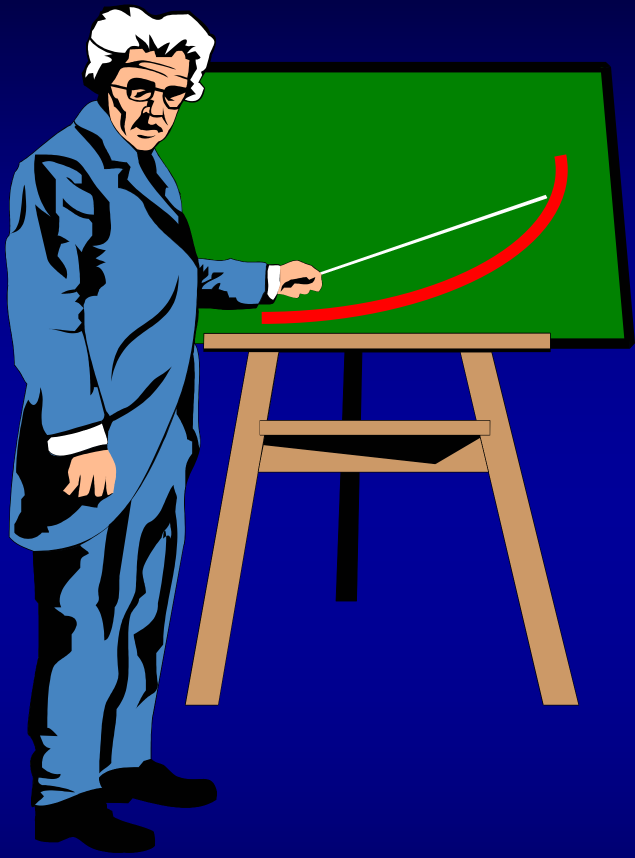
August 4, 2005

Network Stress Testing

- Stress all of the UCLA-developed networks.
  - Individually
  - As an integrated networked system
- Stress each of these networks to the limit:
  - Traffic load
  - Traffic matrix
  - Channel error rate
  - Node failure rate
  - Number of nodes
  - Number of clusters
  - Nodal density
  - Nodal separation
  - Nodal and cluster mobility
  - Malicious attacks



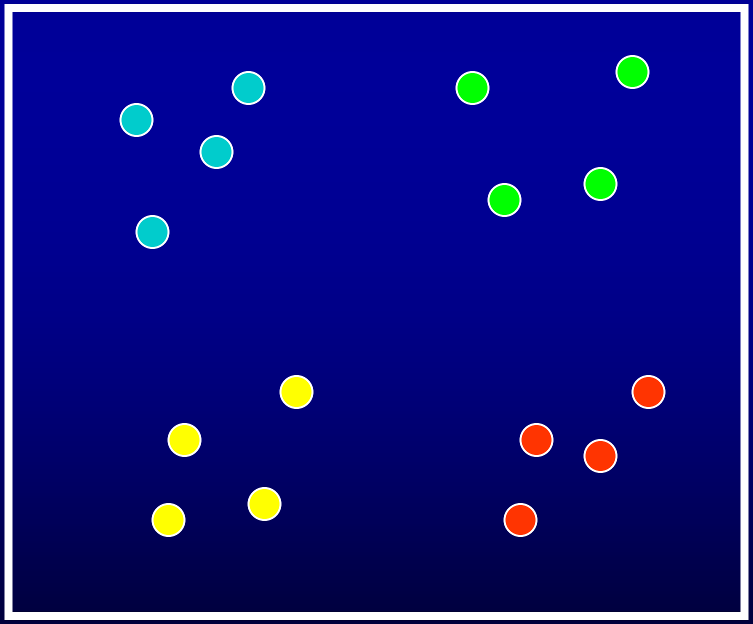
Network Stress Testing



- We measure
  - Response time
  - Throughput
  - QoS
  - Fraction of delivered packets
  - Jitter
  - Recovery time
  - Deadlocks
- Use of simulation as well as actual hardware is used for these tests.

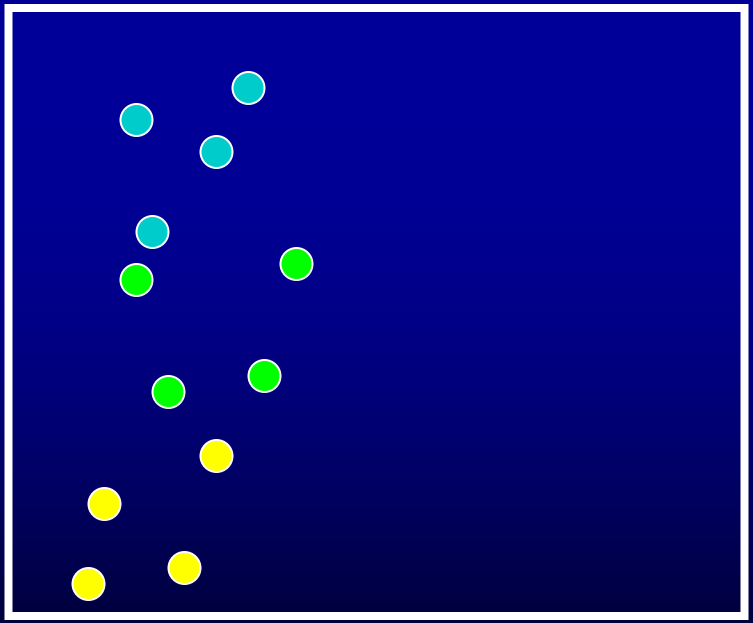
AdHoc Network Scenarios

1. Clusters with Random Search



AdHoc Network Scenarios

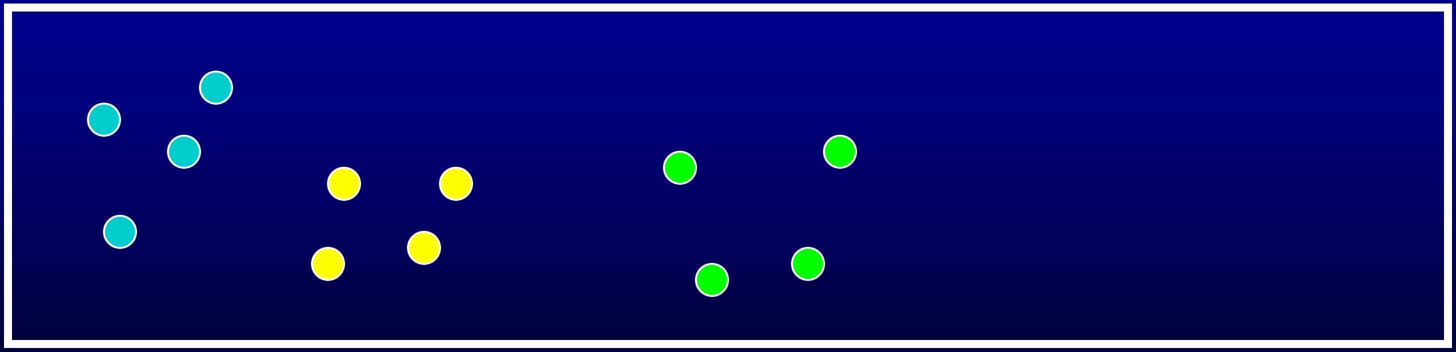
2. Clusters with Patrolling



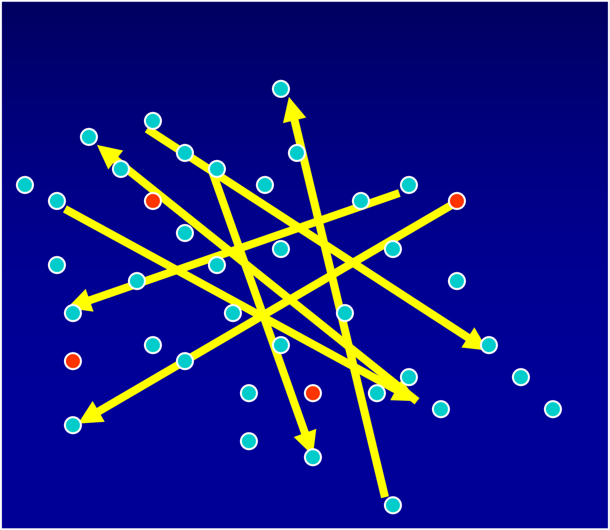
AdHoc Network Scenarios

3. Sneaking

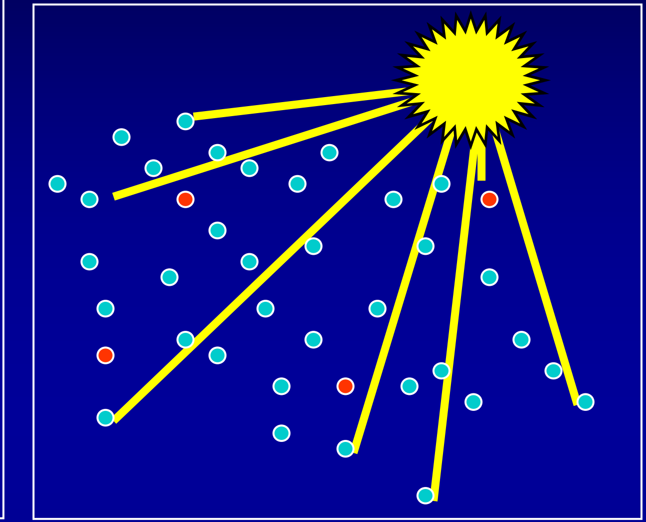
(back and forth after the examination of some zone)



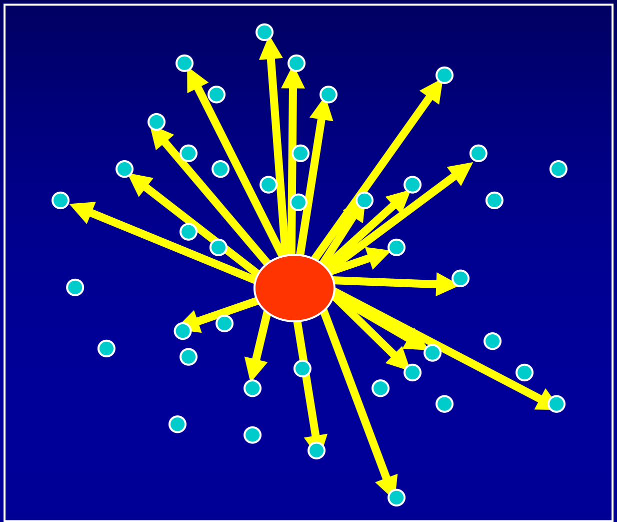
Traffic Matrices



Random destinations



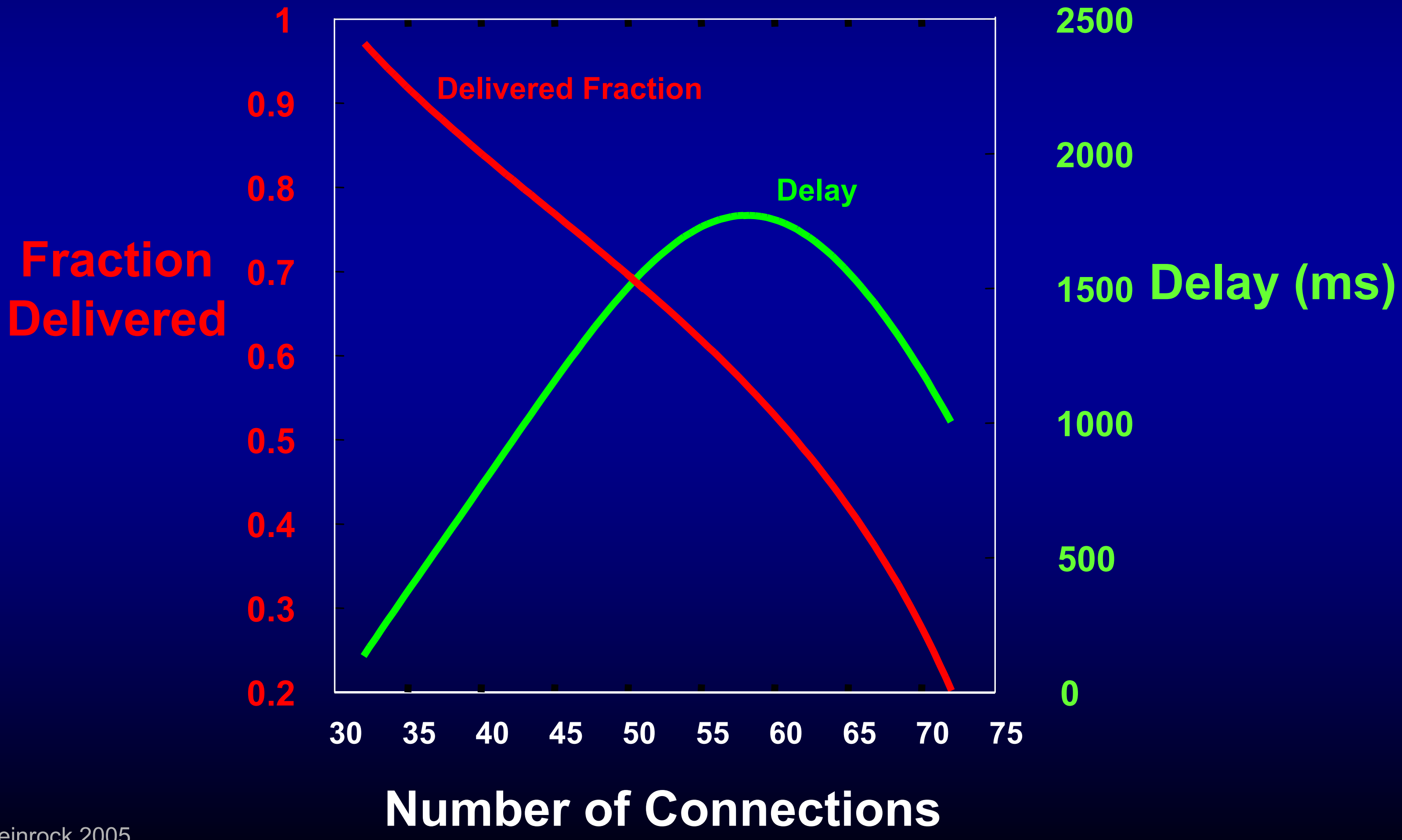
Hot Spot traffic



Multicast traffic

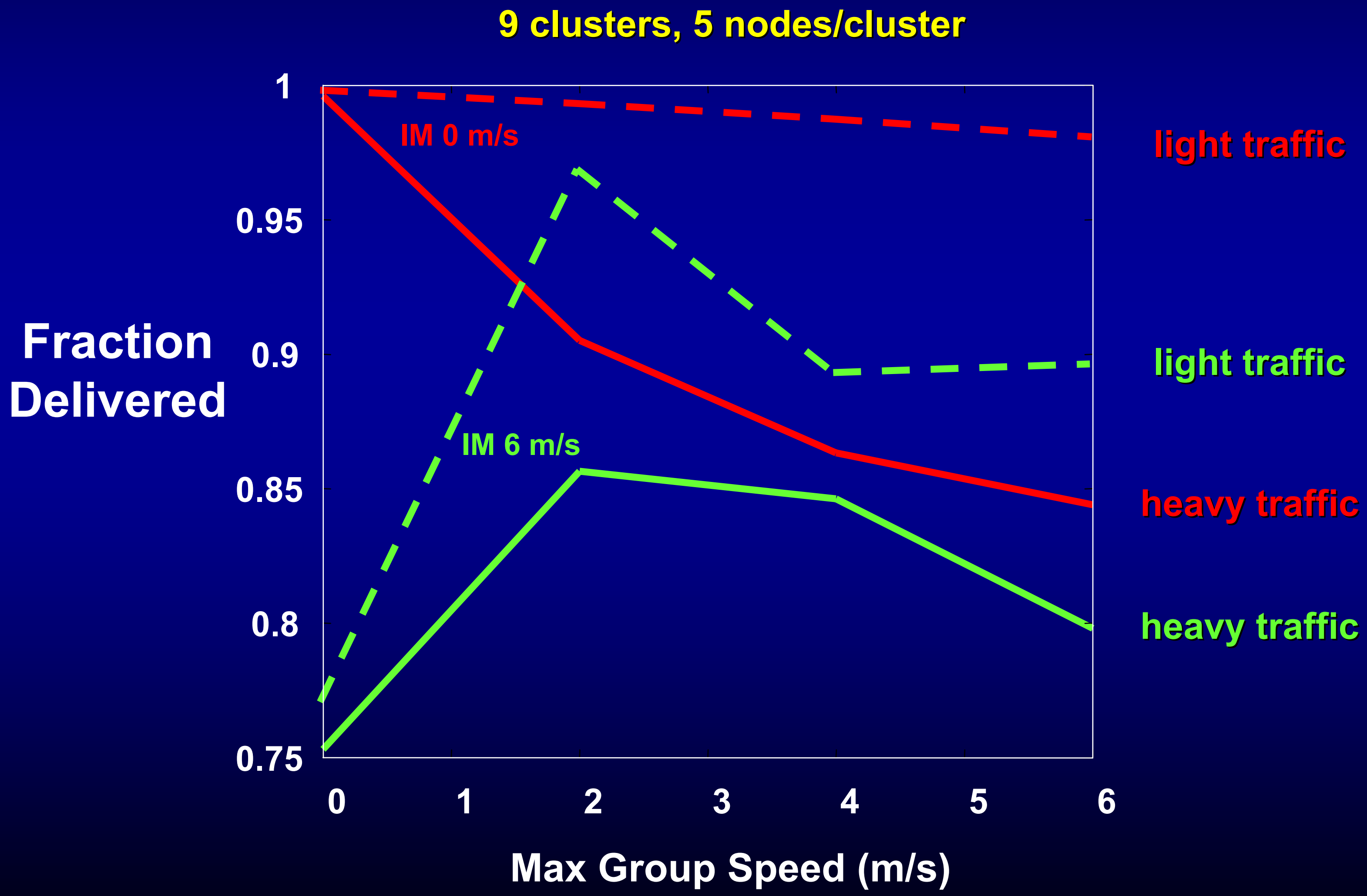
LANMAR Network Results

No Mobility



LANMAR Network Results

Internal and Group Mobility



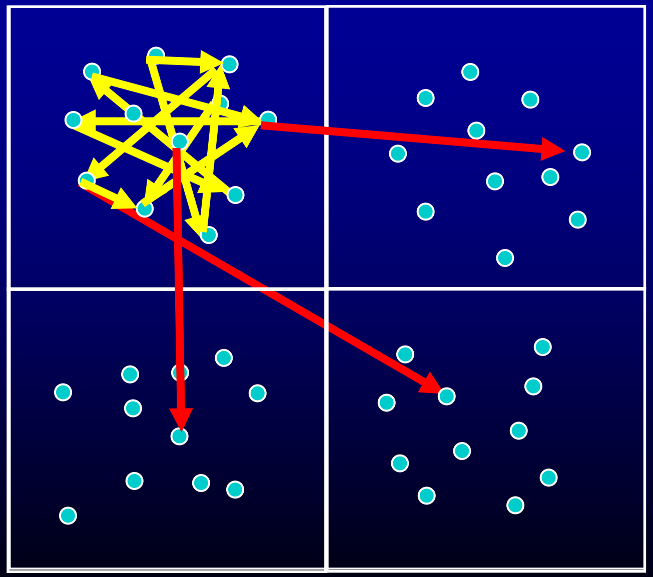
©Leonard Kleinrock 2005

Hilly Landscape Model

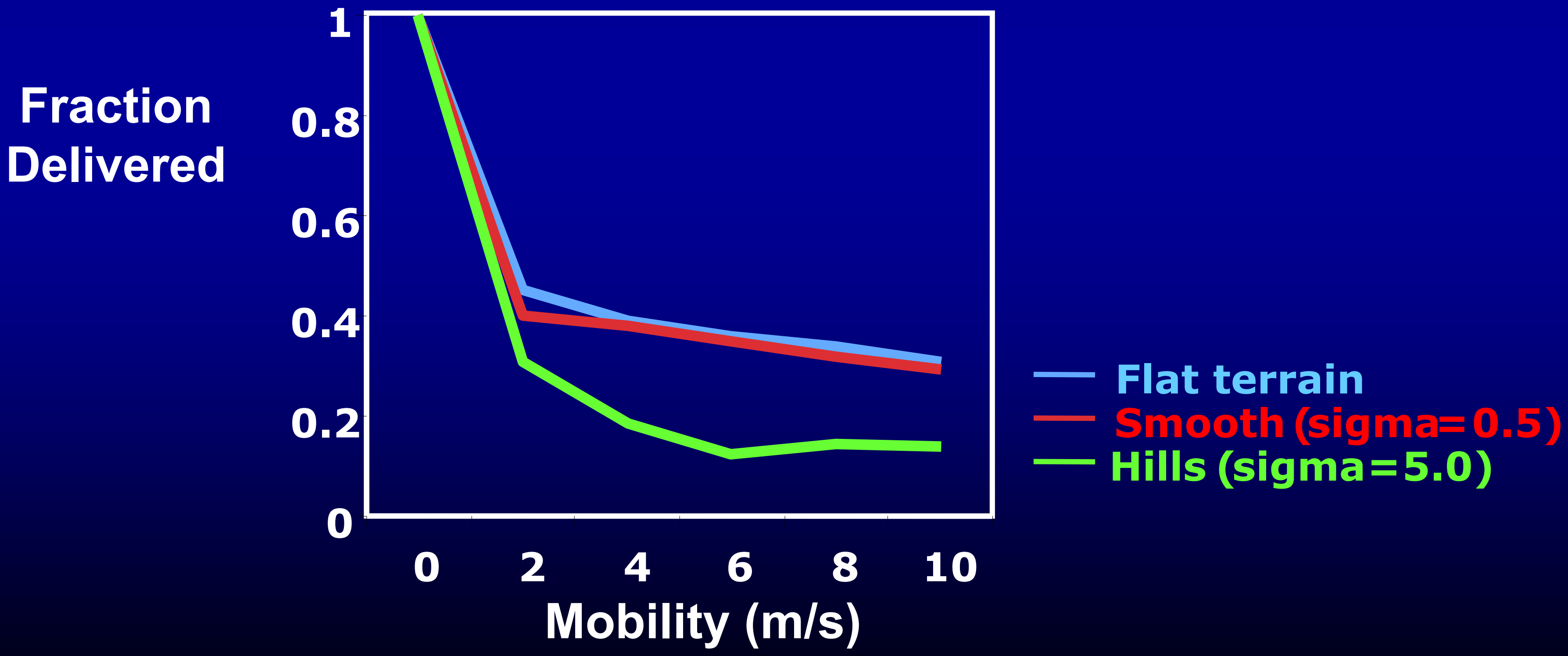
- Nodes are randomly distributed in a square area (1000m x 1000m) with antennae height at 0.5 m.
- “Hills” are located, one in each 50m x 50m. (400 hills in total)
- Hill height has a lognormal distribution
  - sigma 0.5 => 0.5 m average height
  - sigma 5.0 => 5.0 m average height
- Propagation model: Path Loss
  - Slightly pessimistic for high hills since a path loss model assumes that the radio signal does not reach the destination in presence of an obstacle whereas in the real world, the radio signal would go around the hill.

LANMAR Scenario

- Nodes: 45
  - In motion: 30 (Random Waypoint)
  - 4 clusters (simulation area split into 4 squares)
- Connections: 54
  - 45 intracluster
  - 9 intercluster

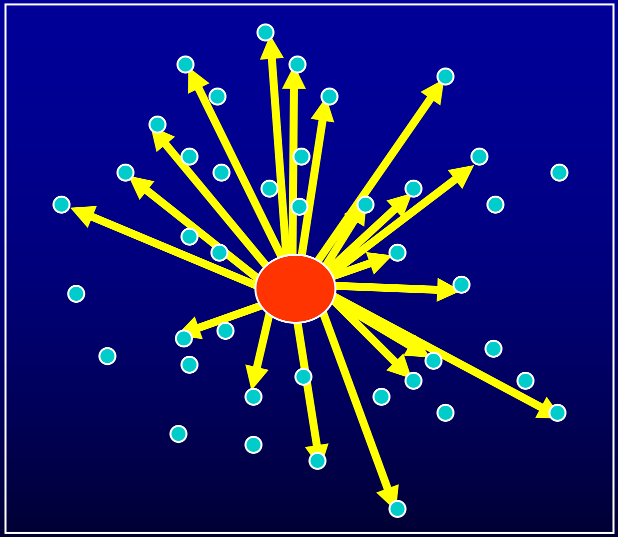


LANMAR Landscape Results



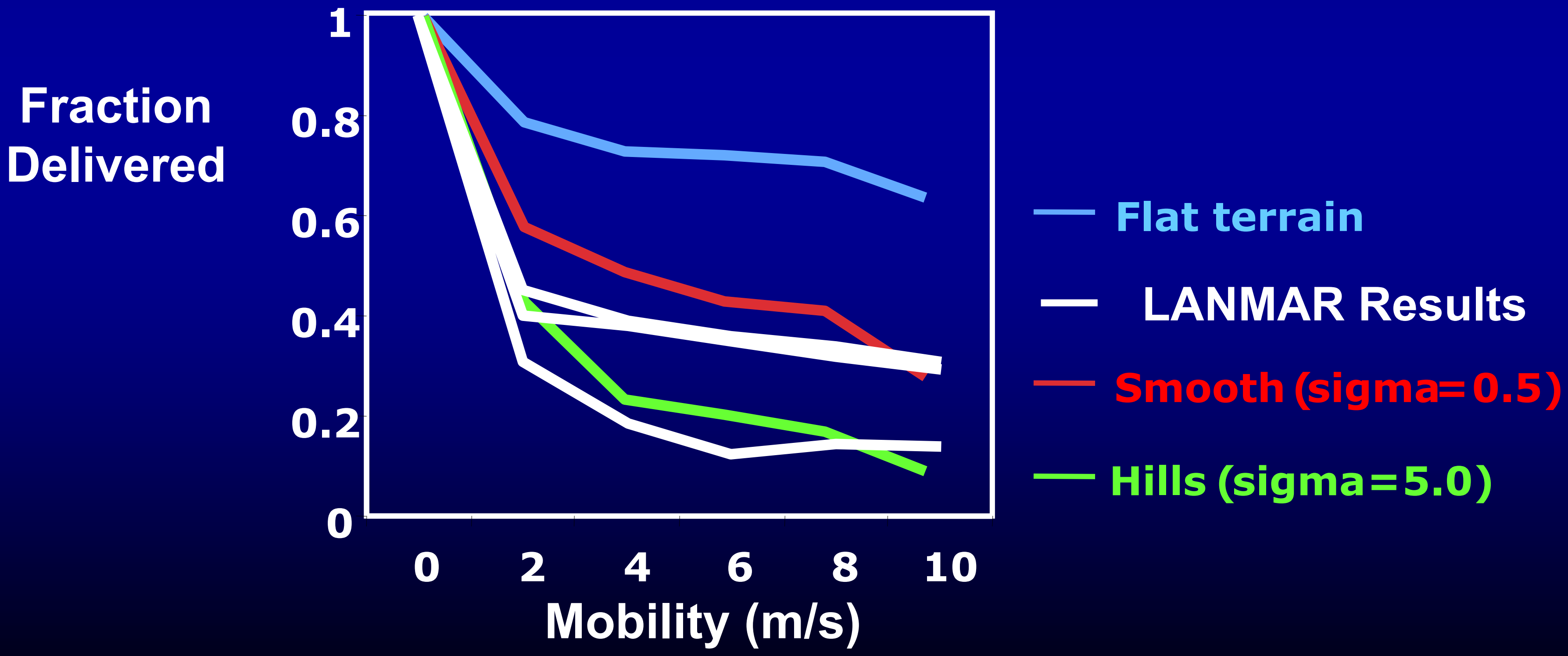
ODMRP Scenario

- Nodes: 45
  - In motion: 30 (Random Waypoint)
- Connections:
  - Source: 1
  - Destinations: 20



©Leonard Kleinrock 2005

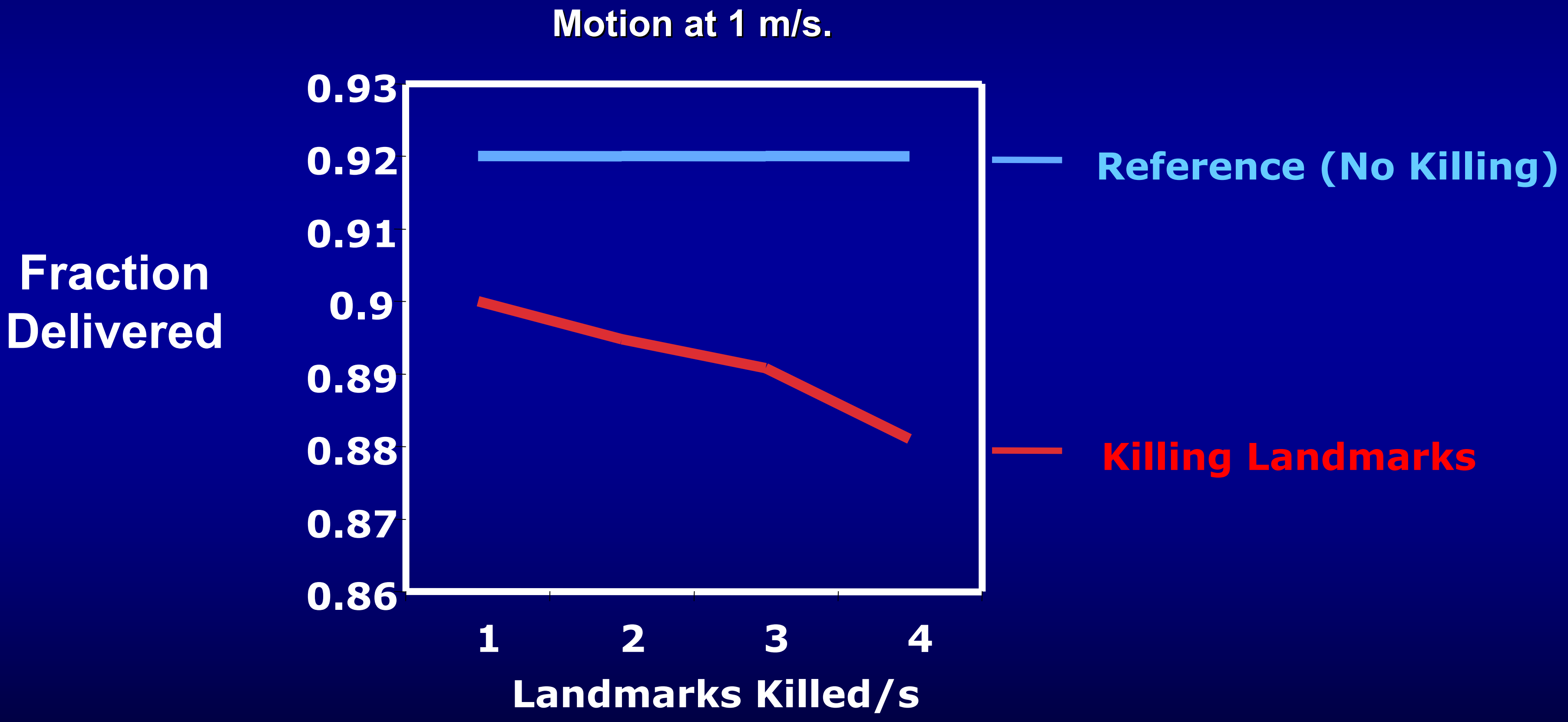
ODMRP Landscape Results



Malicious Attack on LANMAR

- Nodes: 45
  - In motion: 30 (Random Waypoint).
  - 4 clusters
  - Motion at 1 m/s.
- Connections: 54
- Worst case scenario
  - The enemy knows always which the landmarks are
  - For a fair comparison, in the simulation, landmarks are not killed but they are forced to drop their role as landmarks.

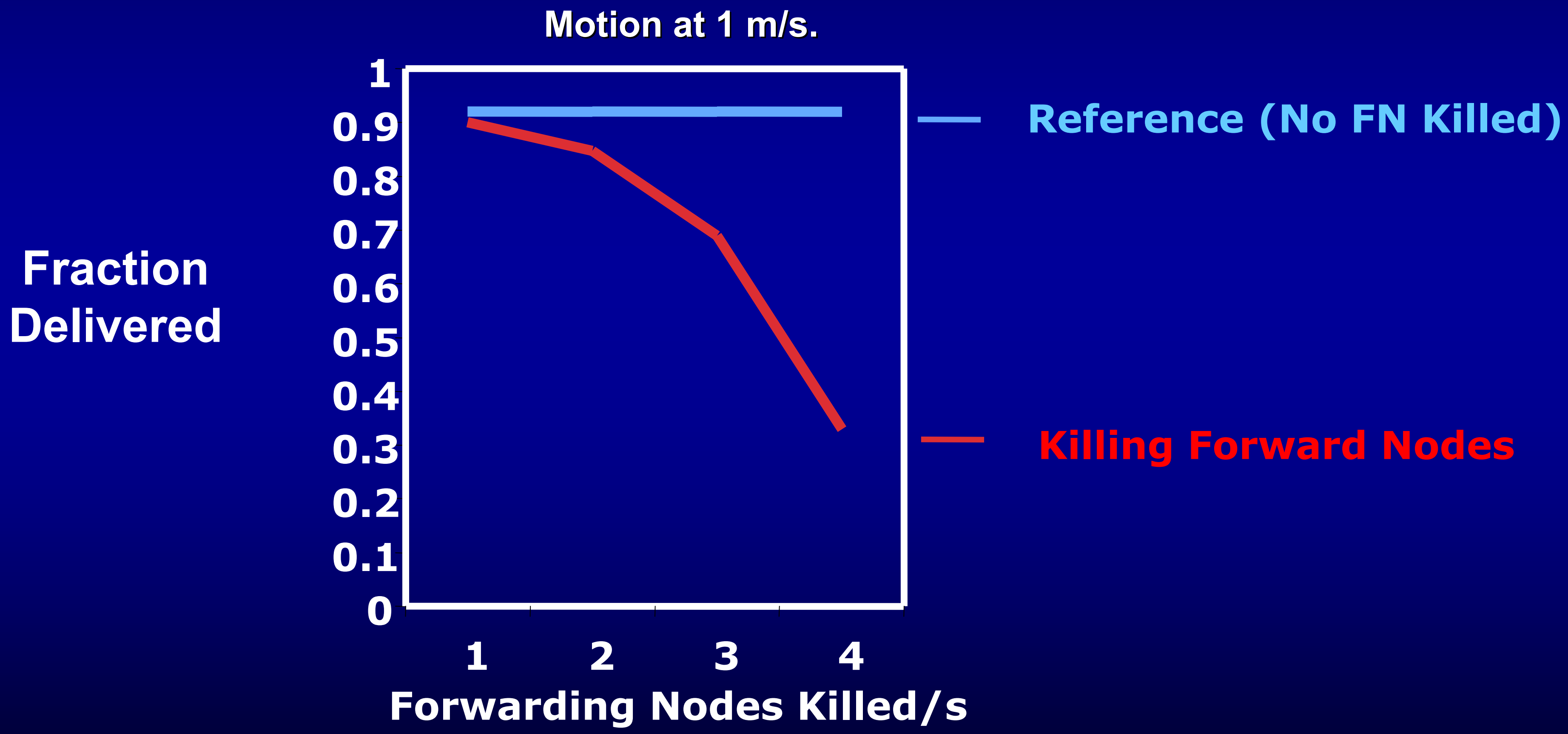
Malicious Attack: LANMAR



Malicious Attack on ODMRP

- Nodes: 45
  - In motion: 30 (Random Waypoint).
  - Motion at 1 m/s.
- Connections:
  - Source: 1
  - Destinations: 20
- Worst case scenario
  - The enemy knows always which the forwarding nodes are;
  - For a fair comparison, in the simulation, forwarding nodes are not killed but they are forced to drop their role as forwarding nodes.

Malicious Attack: ODMRP



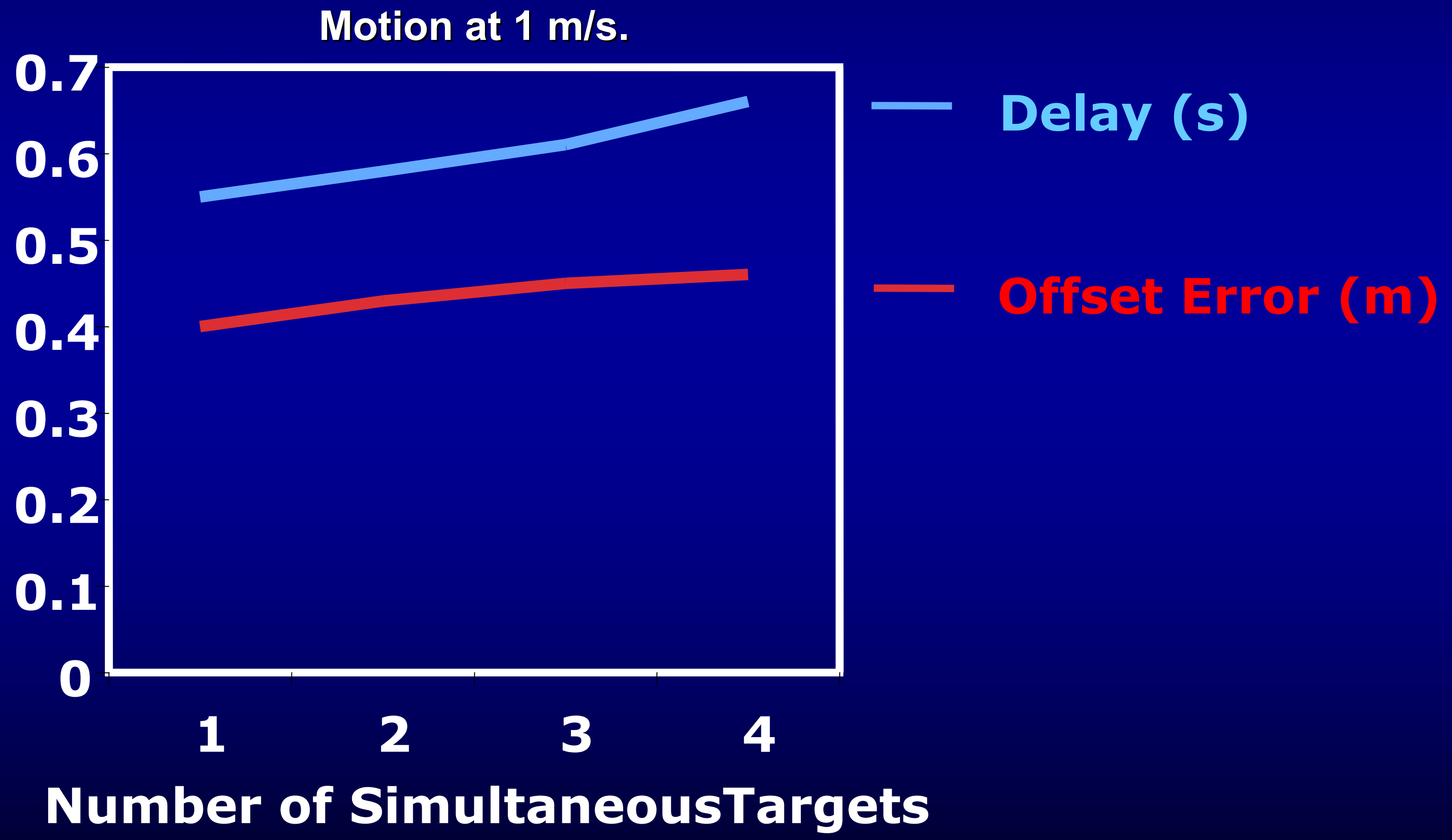
The Sensor Net

- The network is a grid of sensors (1 m. distance).
- The system cannot locate targets if two targets are in range of the same sensor.
- Total time of computation and transmission back to the gateway node is computed.
- No errors on the channel are considered.
- The network is stationary.
- GeoRouting.

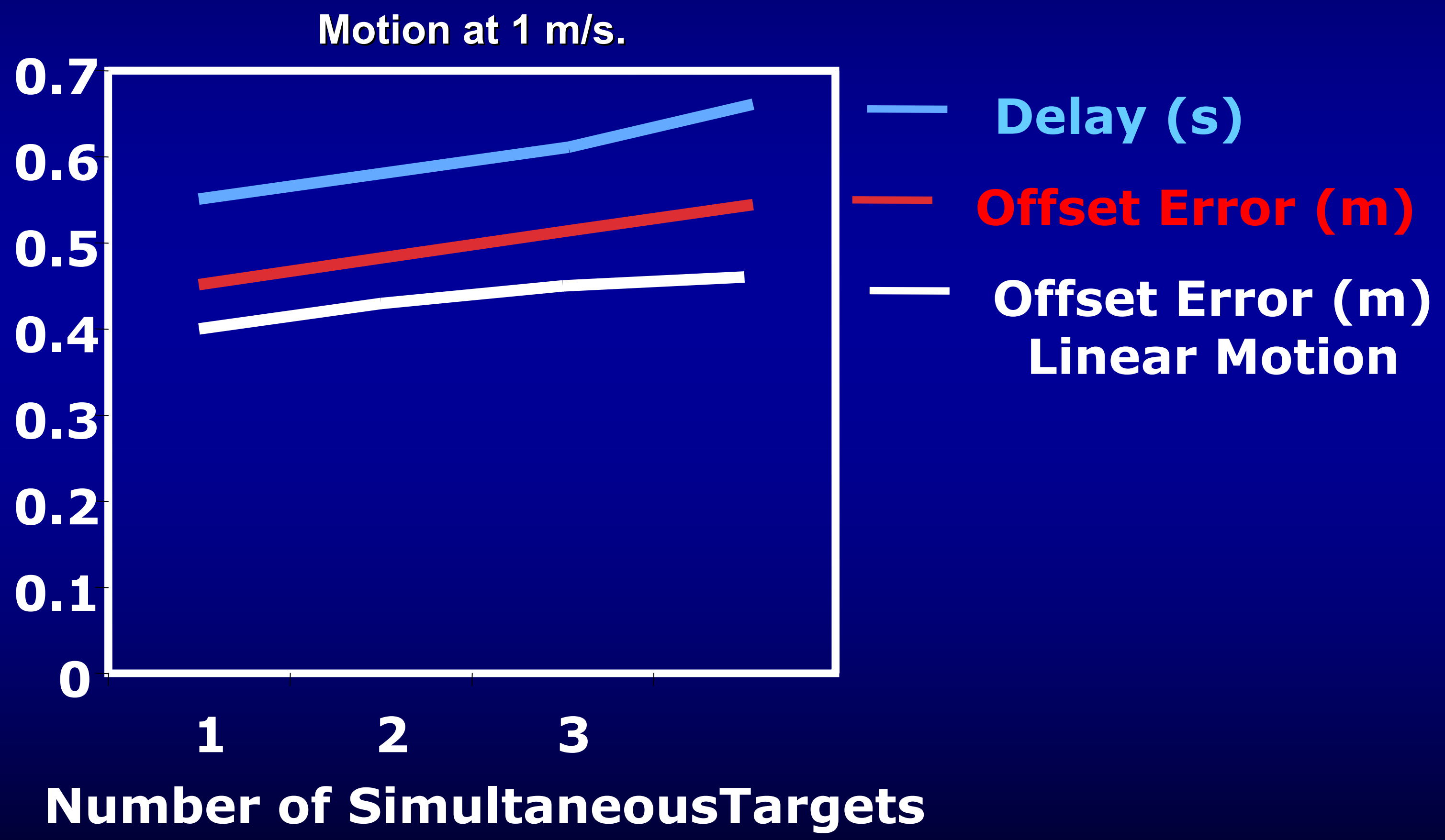
Metrics for the Sensor Net

1. Error in localizing the targets
   - The error is the distance between:
     - the actual location of the target when the backbone receives the information, and
     - the location contained in the information that the backbone receives.
2. Time required to get the information back to the backbone.

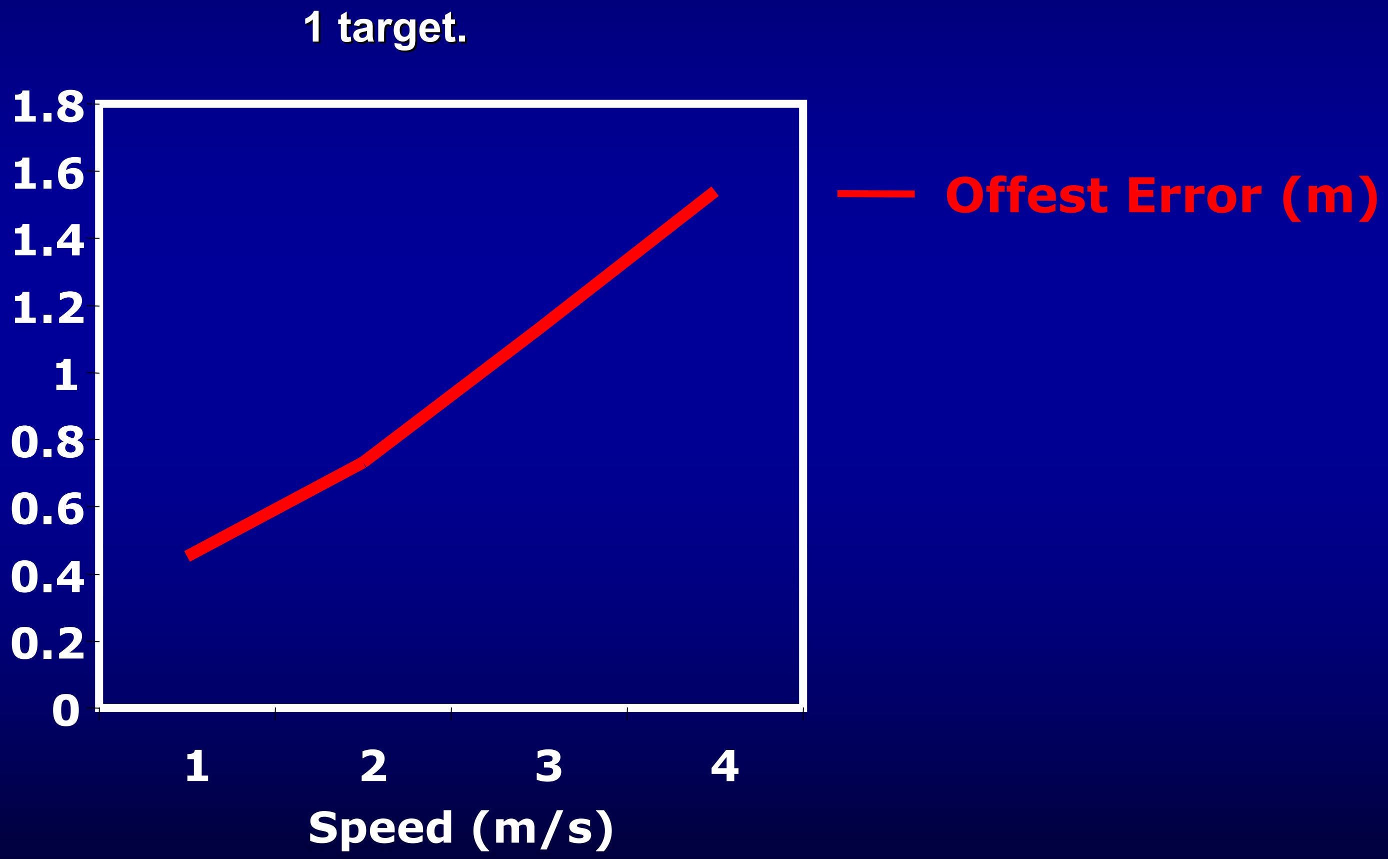
Sensor Net: Linear Motion



Sensor Net: Random Motion



Sensor Net: Linear Motion



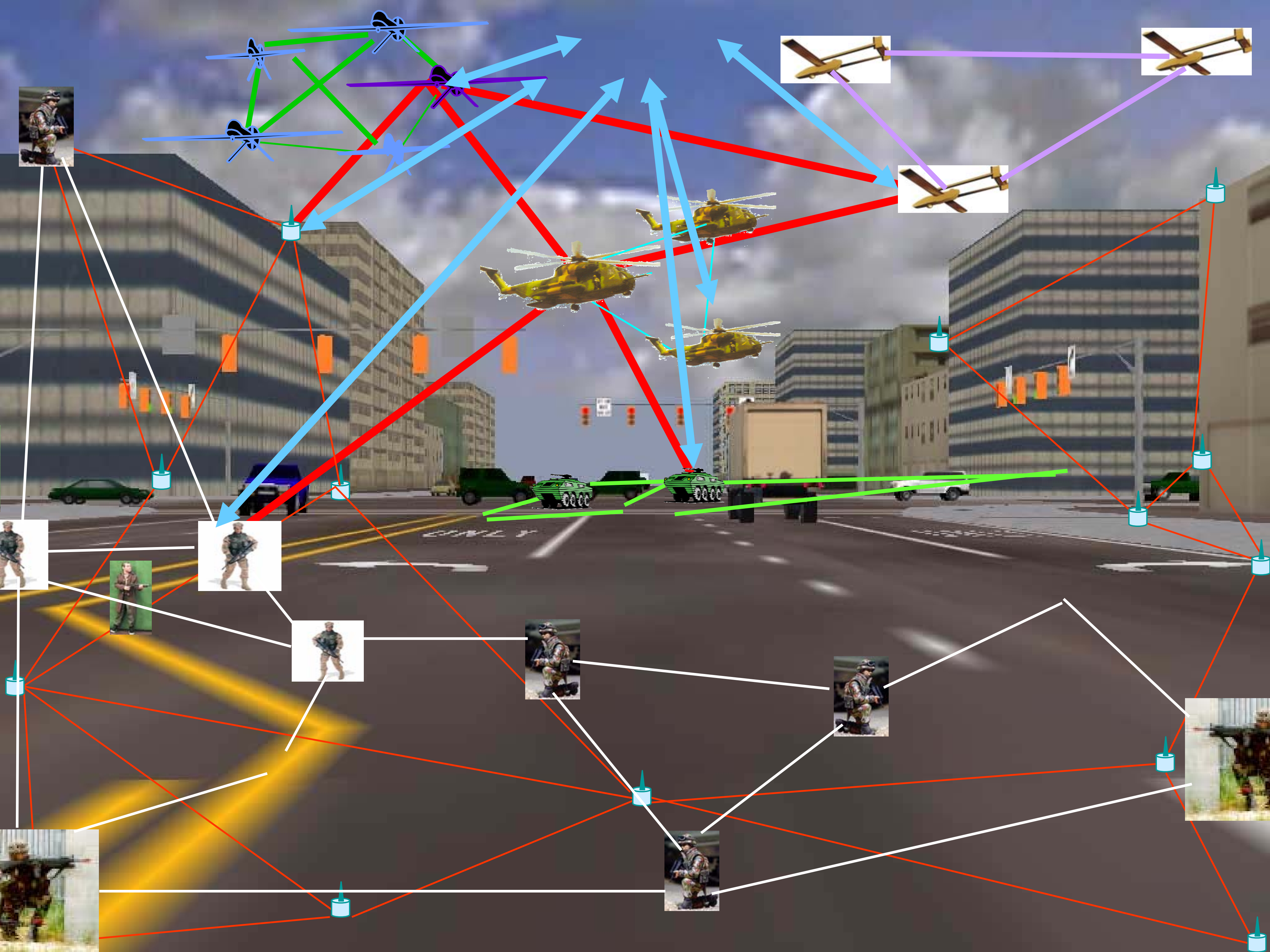
Network Resilience to Stress and Attack

- We have developed a distributed approach to improve the performance of the AINS program, with special focus on the networking elements.
- It provides a mechanism for implementing improved resilience and provides a greater degree of robustness to dynamic conditions.
- The main application is to provide a level of increased network resilience to stress and attack.

Adaptive Autonomous Networks

- We must develop systems whose components are capable of independent cooperative adaptive autonomous action in unpredictable environments.
- Large collections of these entities will be deployed in a distributed environment
- Scalable solutions should be designed with
  - Shared awareness
  - Collaboration
  - Synchronization
  - Understanding

An Example Scenario of Autonomous Operation

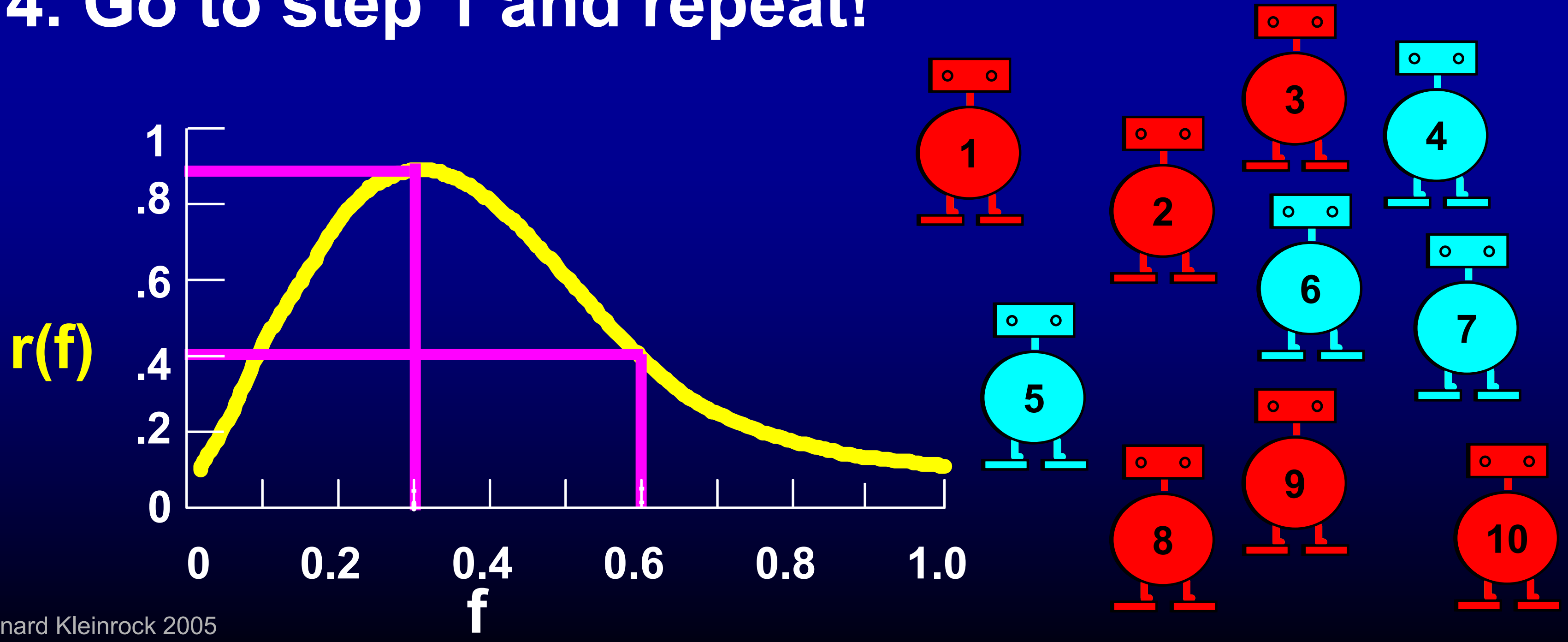


Properties of Our Algorithm

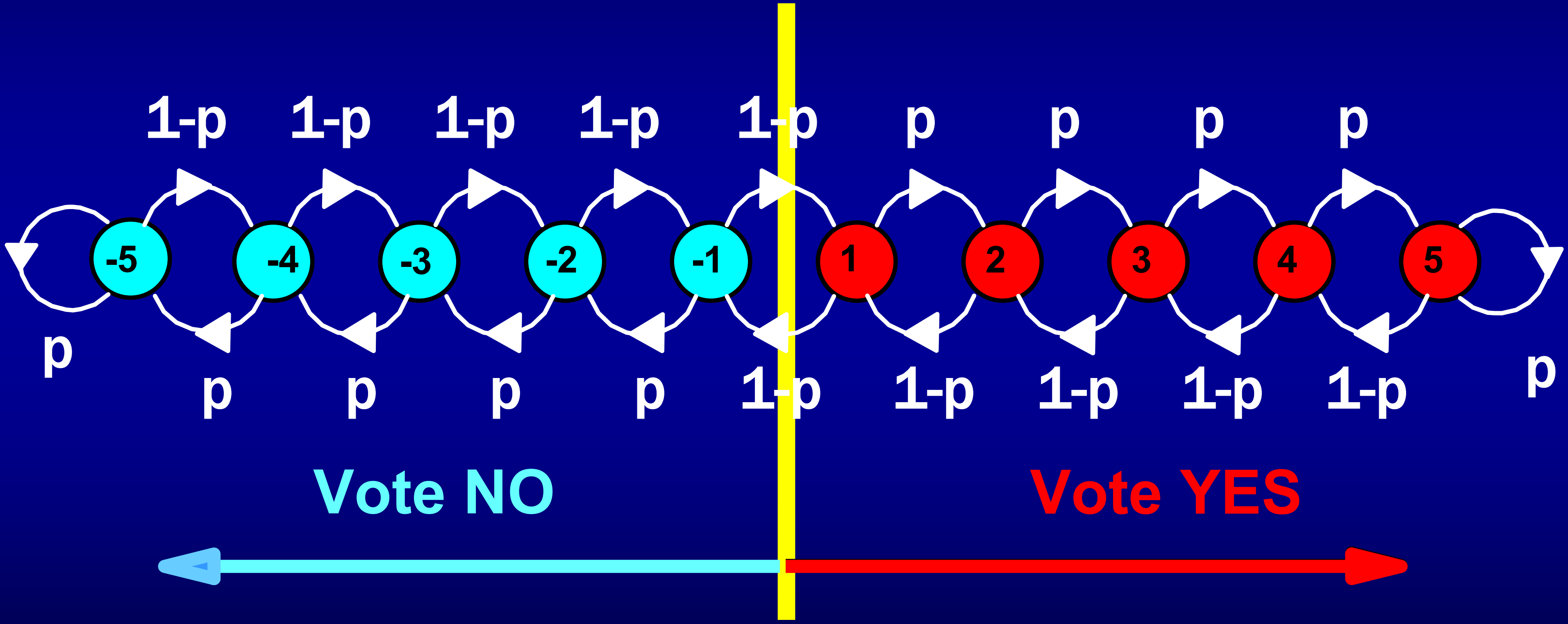
- The approach is based on a distributed algorithm called the Gur Algorithm.
- This is a basic research effort at its core.
- The basic properties of this algorithm are:
  - Highly distributed
  - Each node operates almost independently and autonomously
  - Uses only a minimal amount of global knowledge
  - Uses a non-procedural approach, i.e., one describes a goal and leaves the methodology up to the algorithm; i.e., it provides for ease of control.
  - Extremely robust
  - Highly dynamic
  - Highly scalable
  - Highly flexible

The Gur Algorithm

1. Each Agent votes YES or NO
2. A fraction f votes YES
3. Using a function r(f) which is unknown to them, GOD gives (takes) $1 from each independently with probability p
4. Go to step 1 and repeat!



How It Works

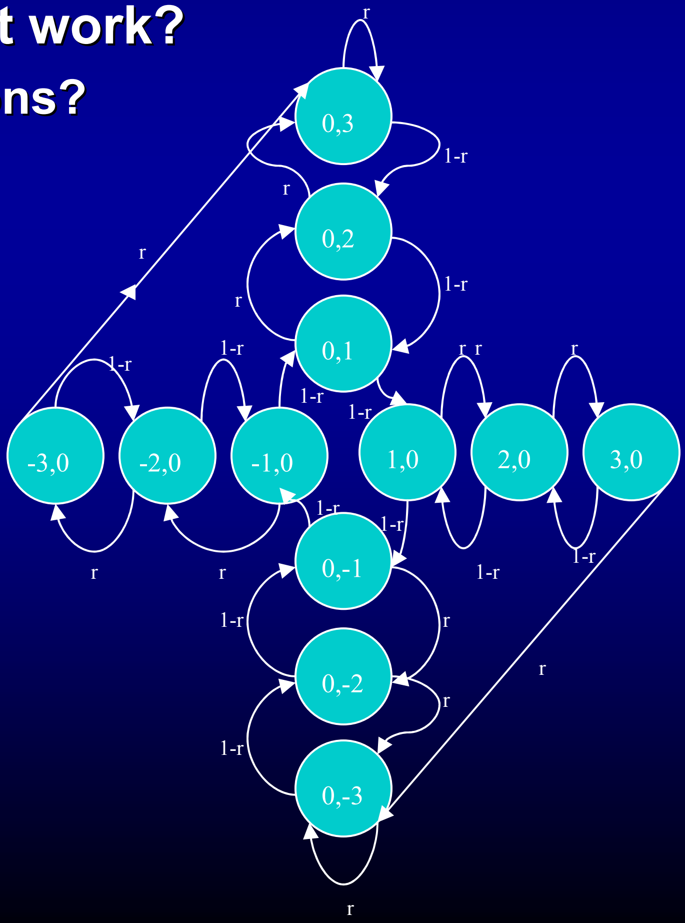

1-p
1-p
1-p
1-p
1-p
p
p
p
p
-5
-4
-3
-2
-1
1
2
3
4
5
p
p
p
p
p
1-p
1-p
1-p
1-p
1-p
p
Vote NO
Vote YES

Applications

- Power control
- Uniform coverage in sensor nets
- Uniform energy expenditure in sensor nets
- Redeployment of scarce resources
- Dynamic configuration of swarms
- Multi-access communications
- Maintenance of zero acceleration bodies subject to arbitrary forces

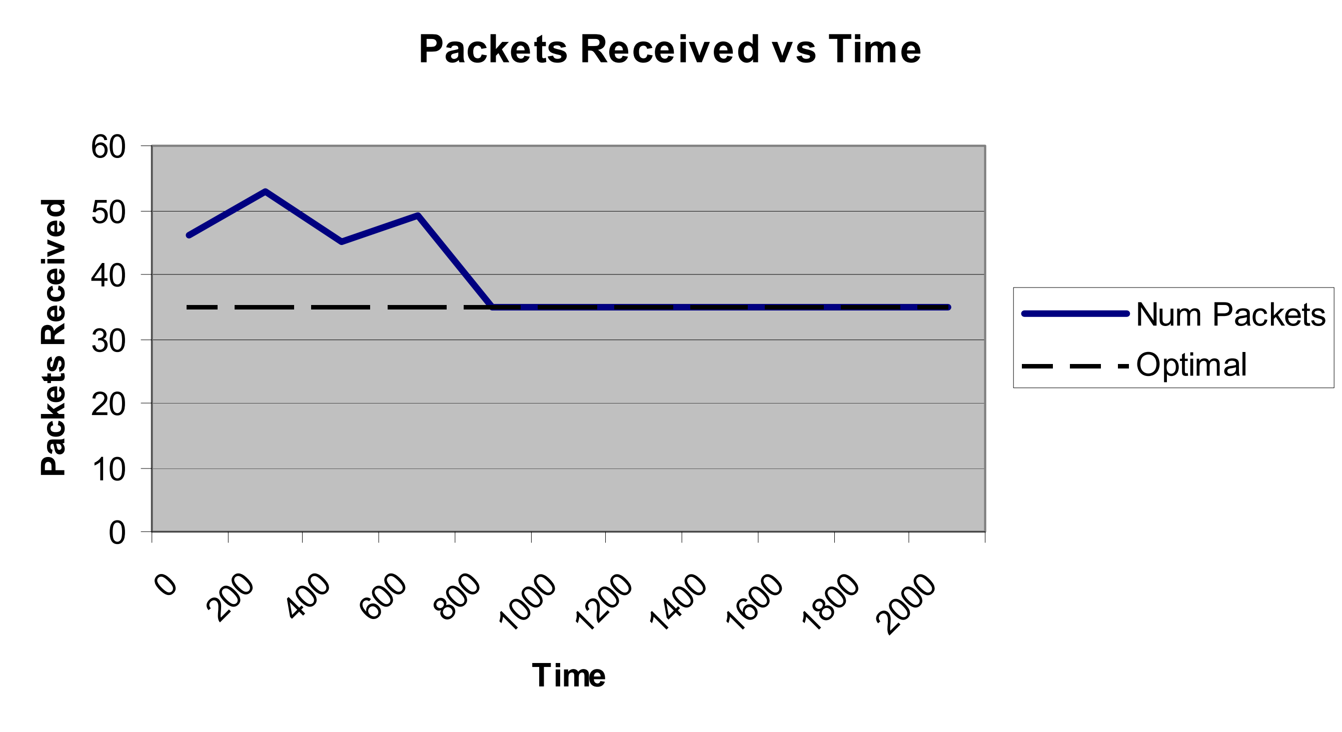
The Gur Game: Modifications

- Different memory structures
- How does it work?
  - Transitions?
  - Voting?



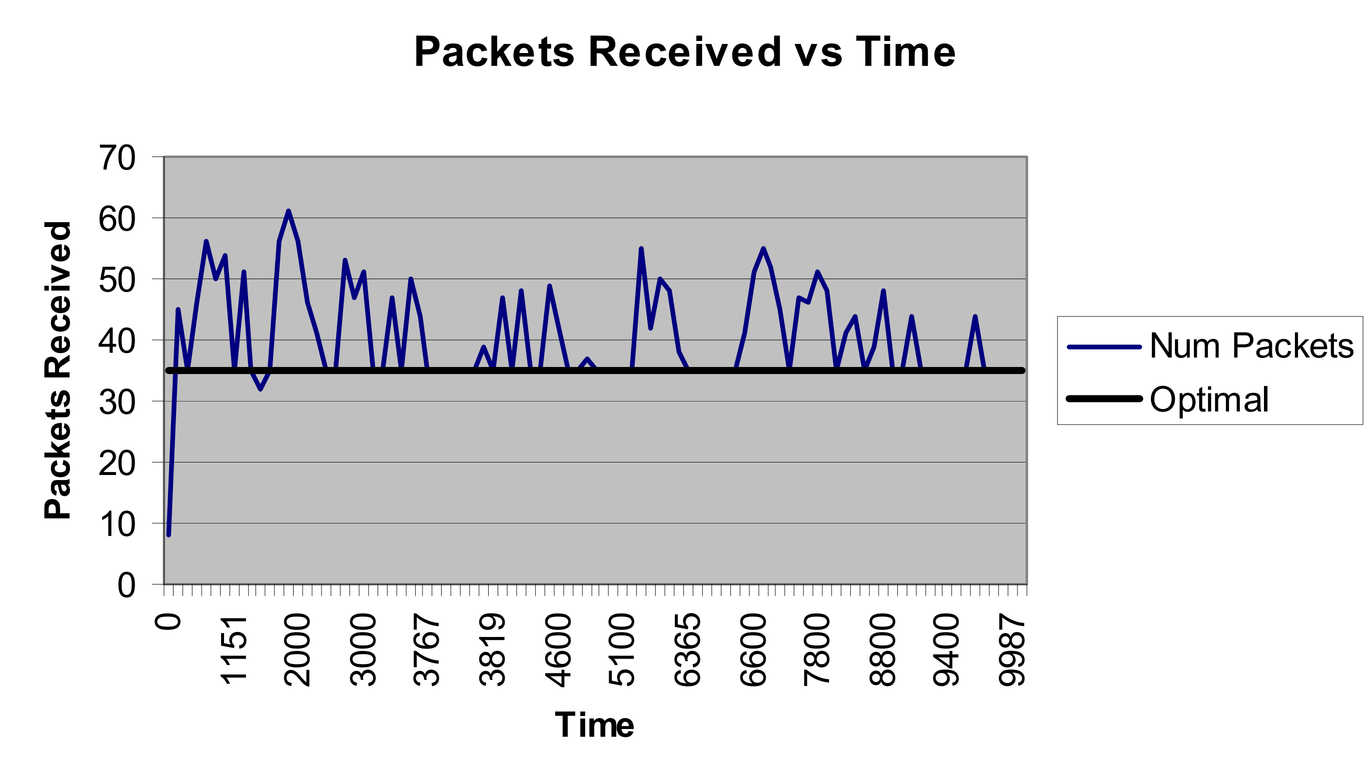
QoS Control For Sensor Networks

- Simulation 1: no birth/death, no delay, run for 2000 secs



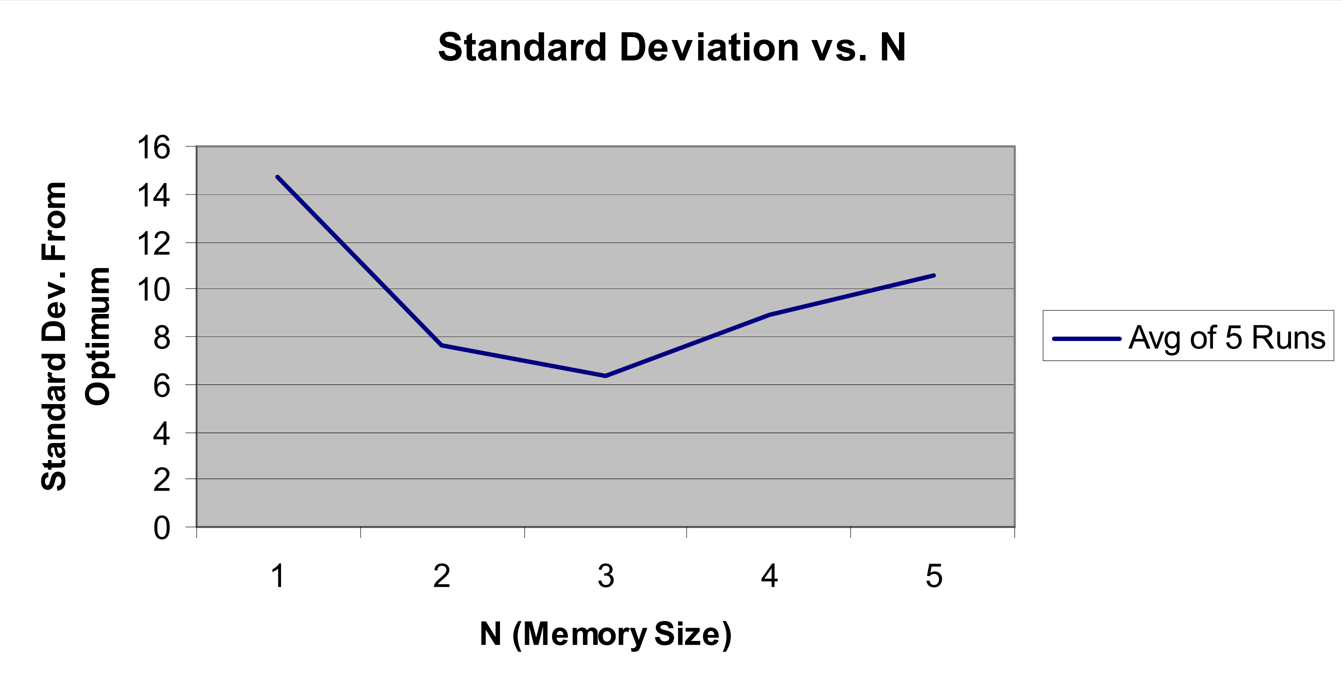
QoS Control For Sensor Networks

- **Simulation 2: delays uniformly distributed between 0-5, mean exp time between births = 101 secs, mean time between deaths = 100 secs, run for 10000 secs**



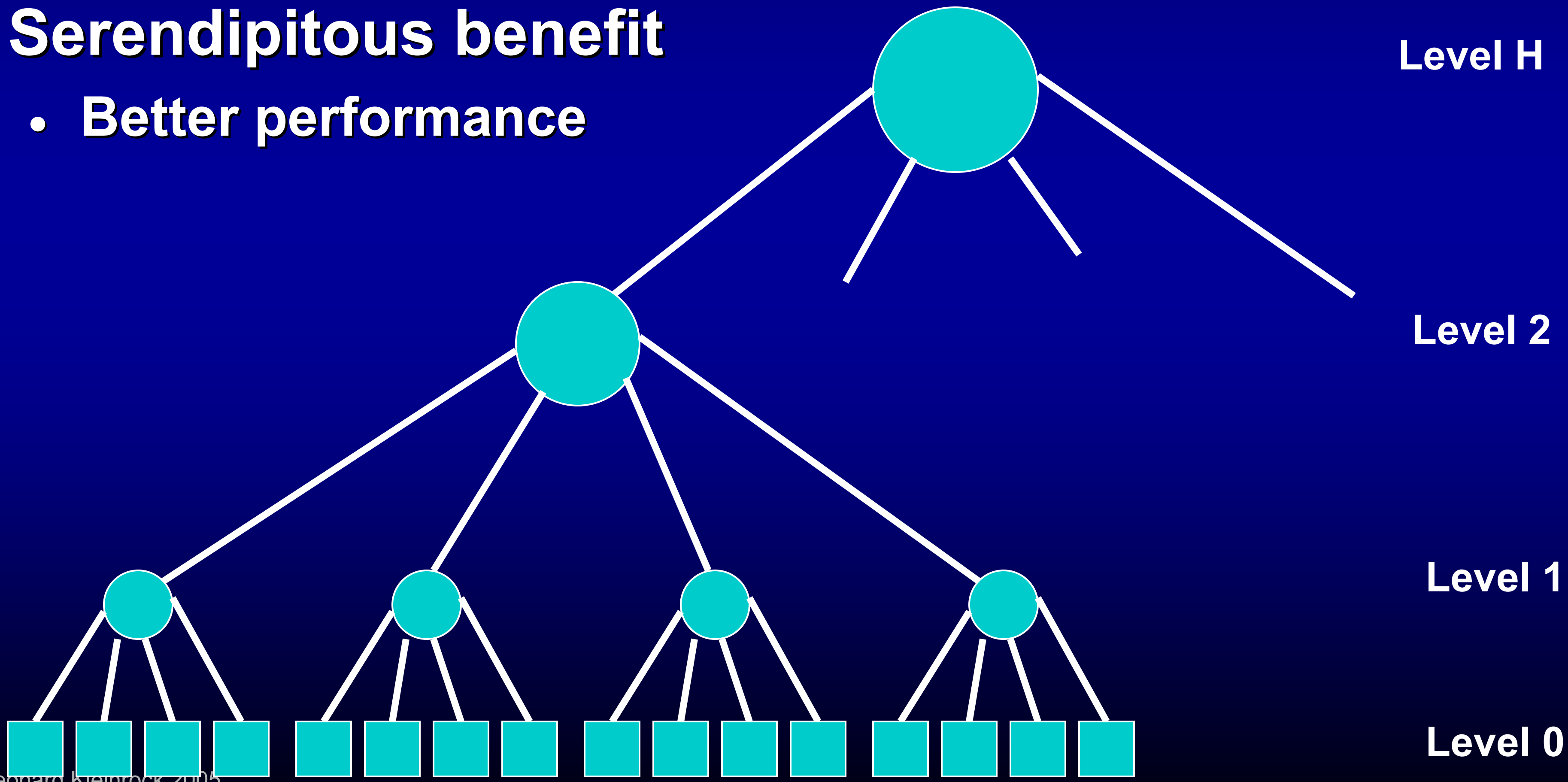
QoS Control For Sensor Networks

- Simulation 3: Study the effect of the memory size N, measure standard deviation from optimal, all parameters the same

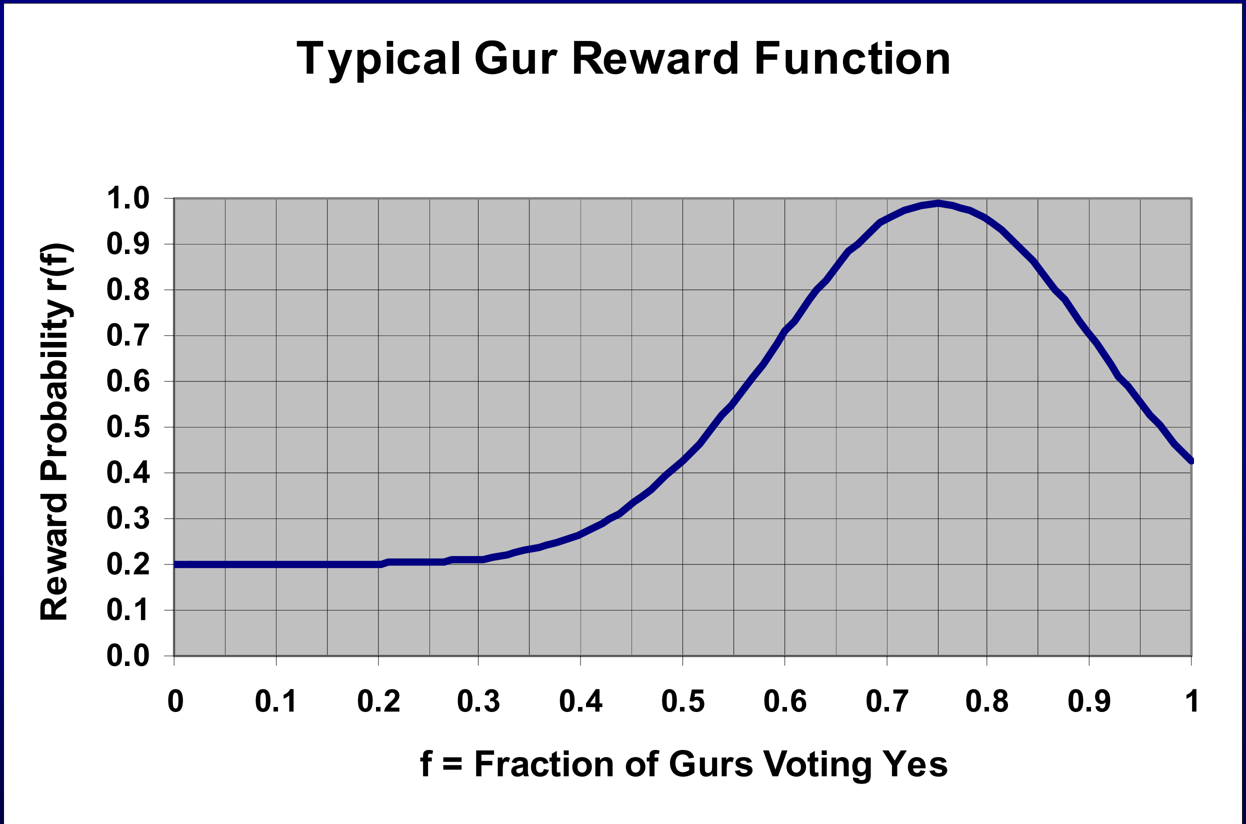


Scalable Sensor Network Resolution

- Hierarchical structure gives us scalability
- Serendipitous benefit
  - Better performance

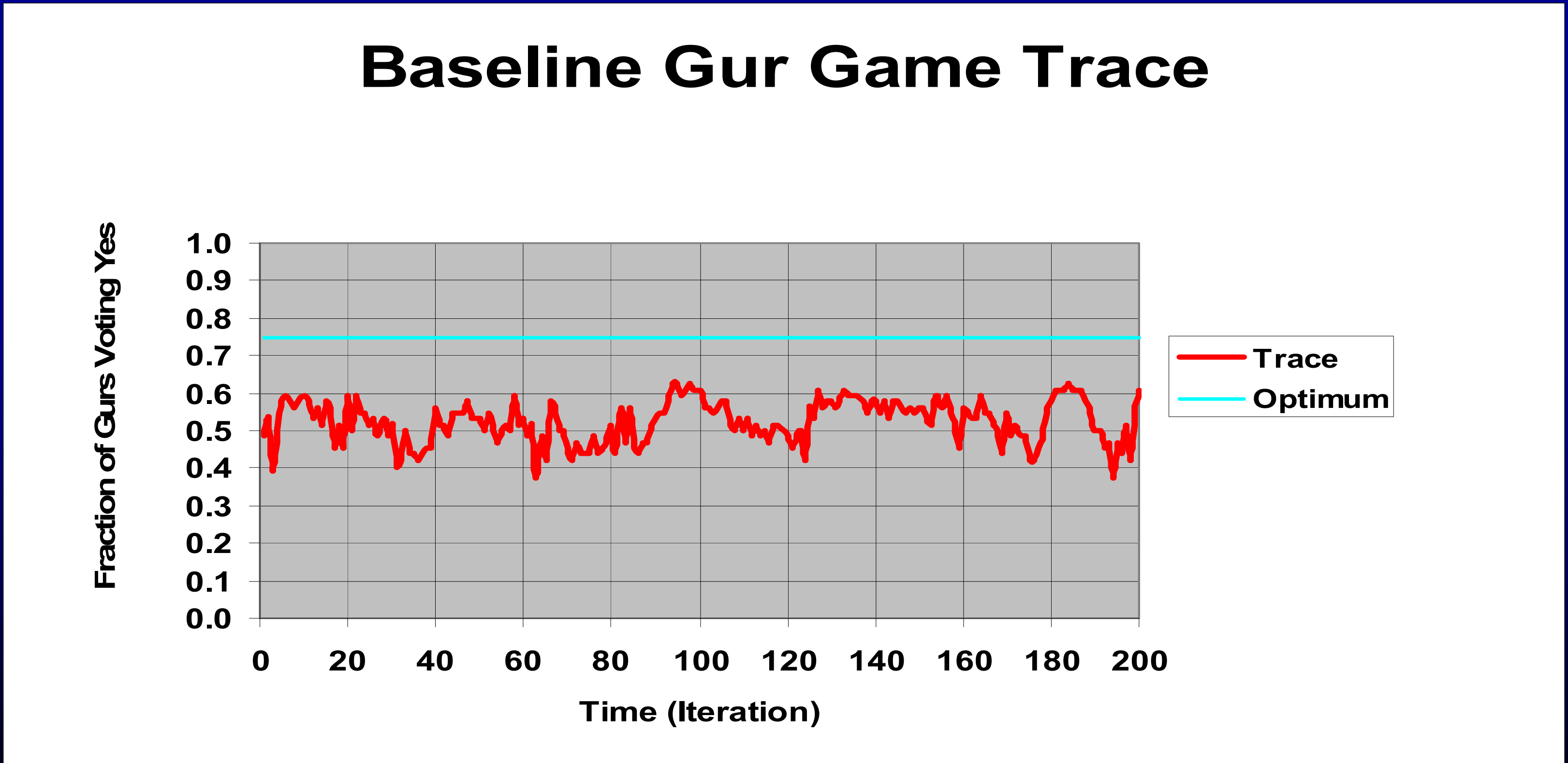


Scalable Sensor Network Resolution



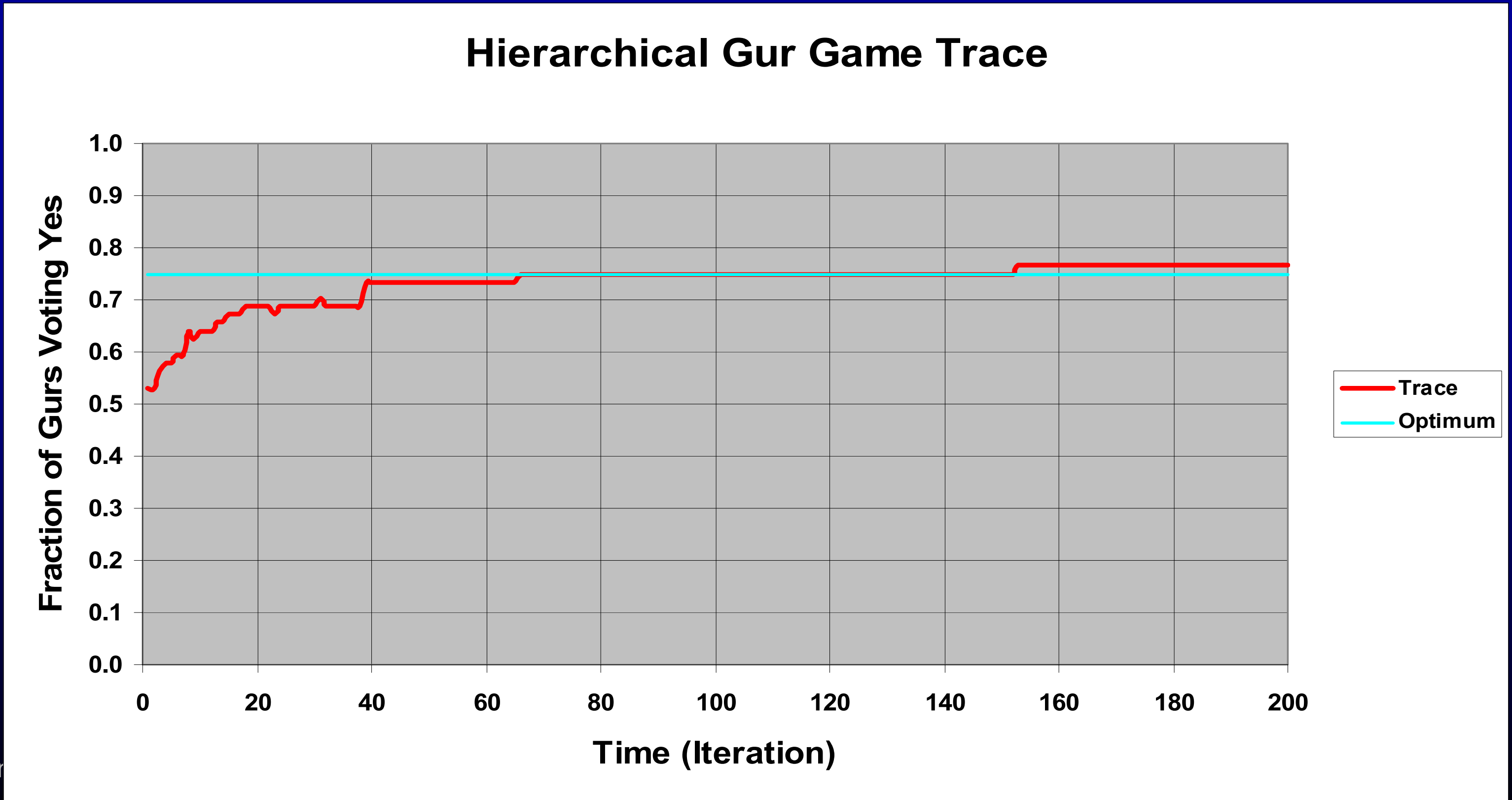
Scalable Sensor Network Resolution

- Simulation 1: Baseline time to convergence

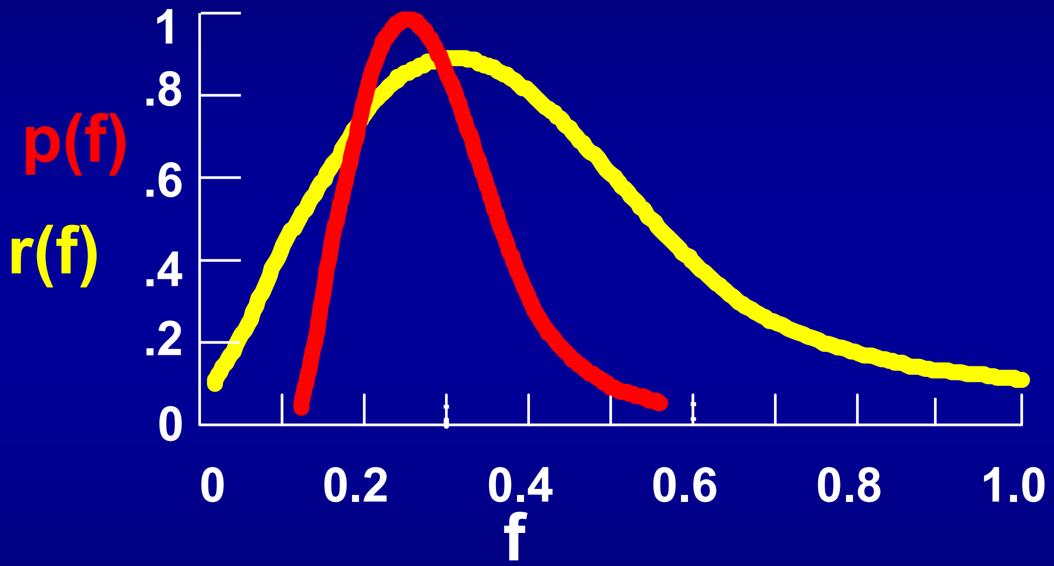


Scalable Sensor Network Resolution

- Simulation 2: Hierarchical Time to convergence

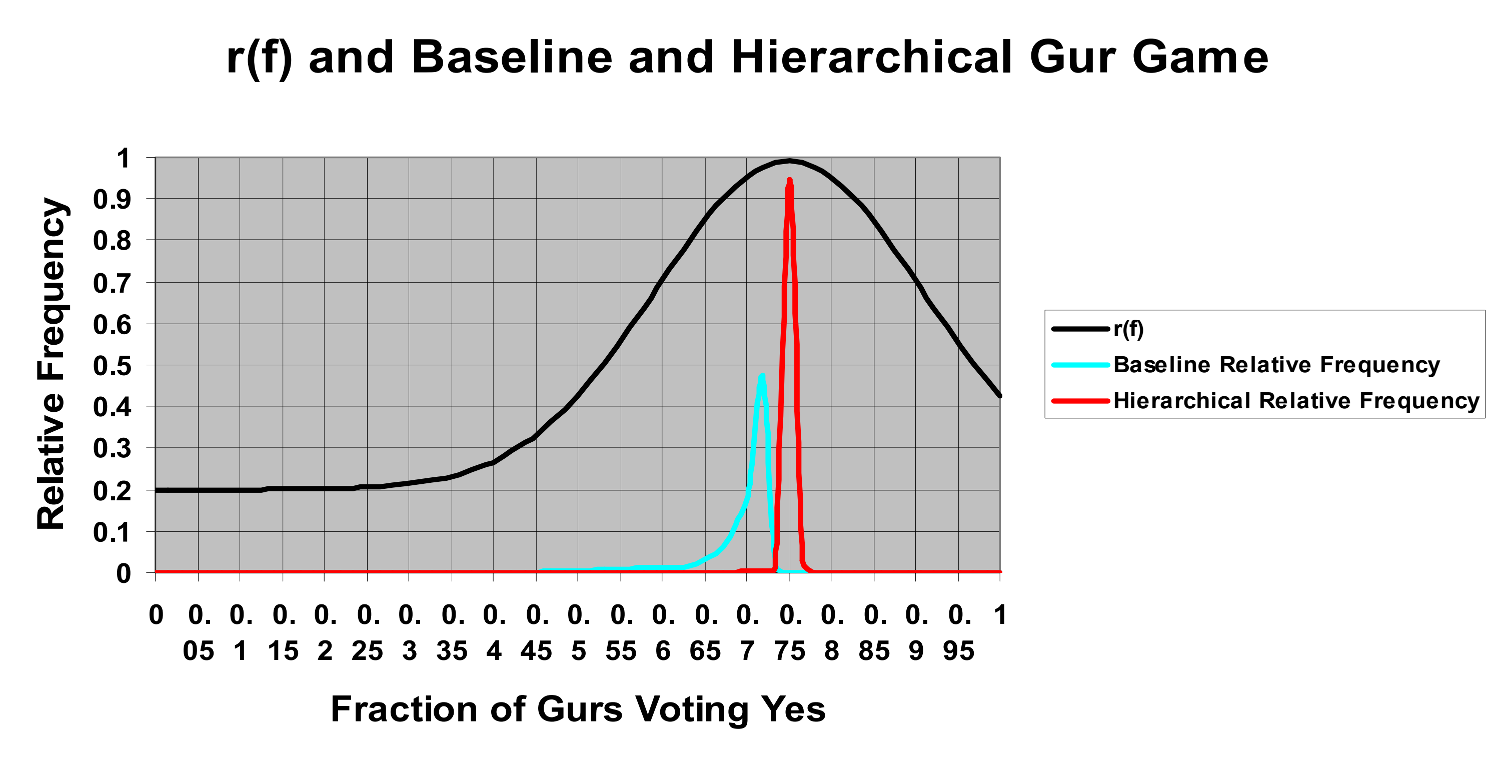


The Gur Algorithm



Scalable Sensor Network Resolution

- Simulation 3: Show time spent in optimum or near optimum states is much higher in Hierarchical than Baseline

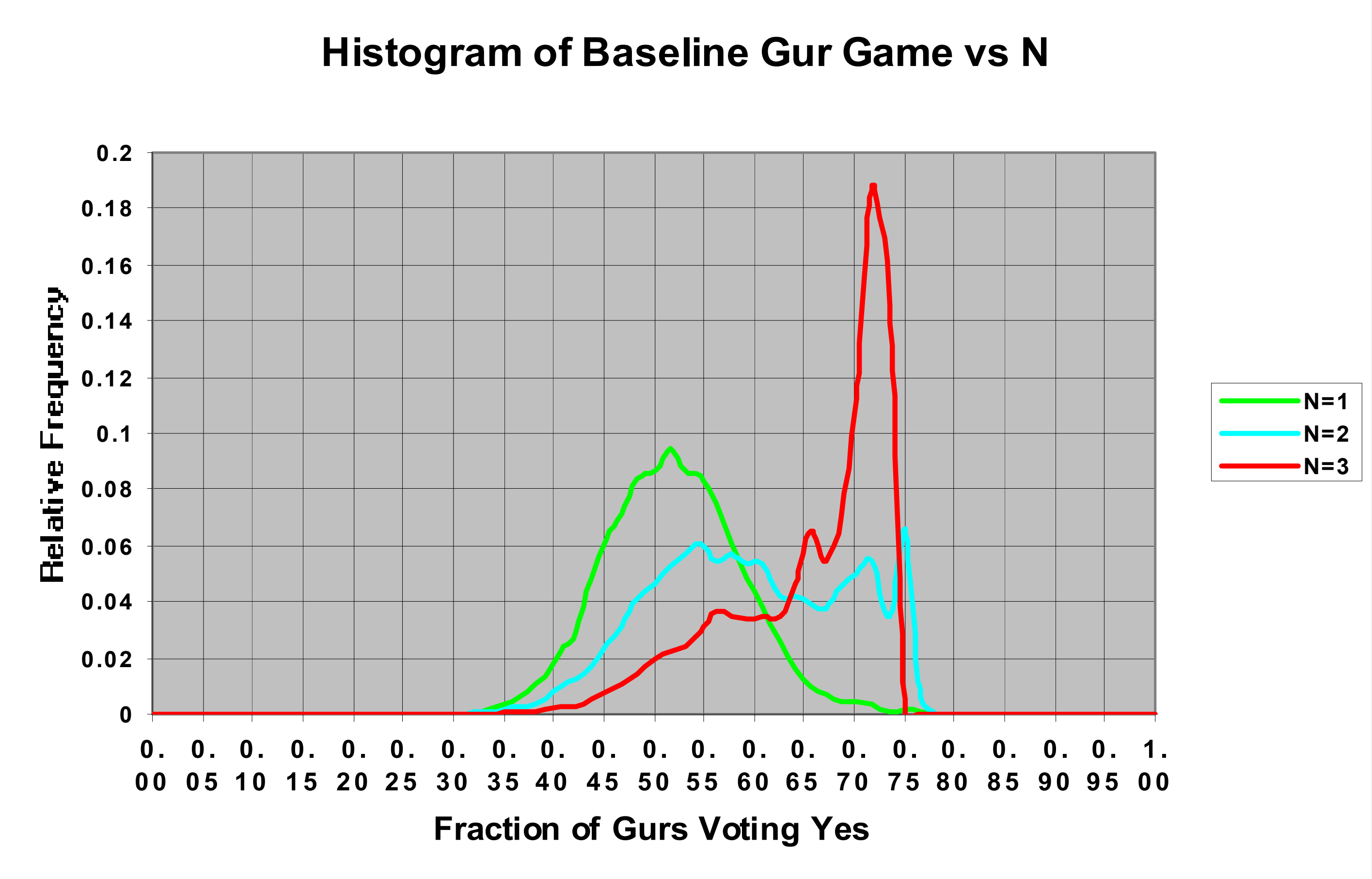


Scalable Sensor Network Resolution

- Simulation 4 and 5: Study the parameter N (memory size) and its effect on the relative time the system stays in a particular state
- run for 10000 seconds
- Study both Hierarchical and Baseline

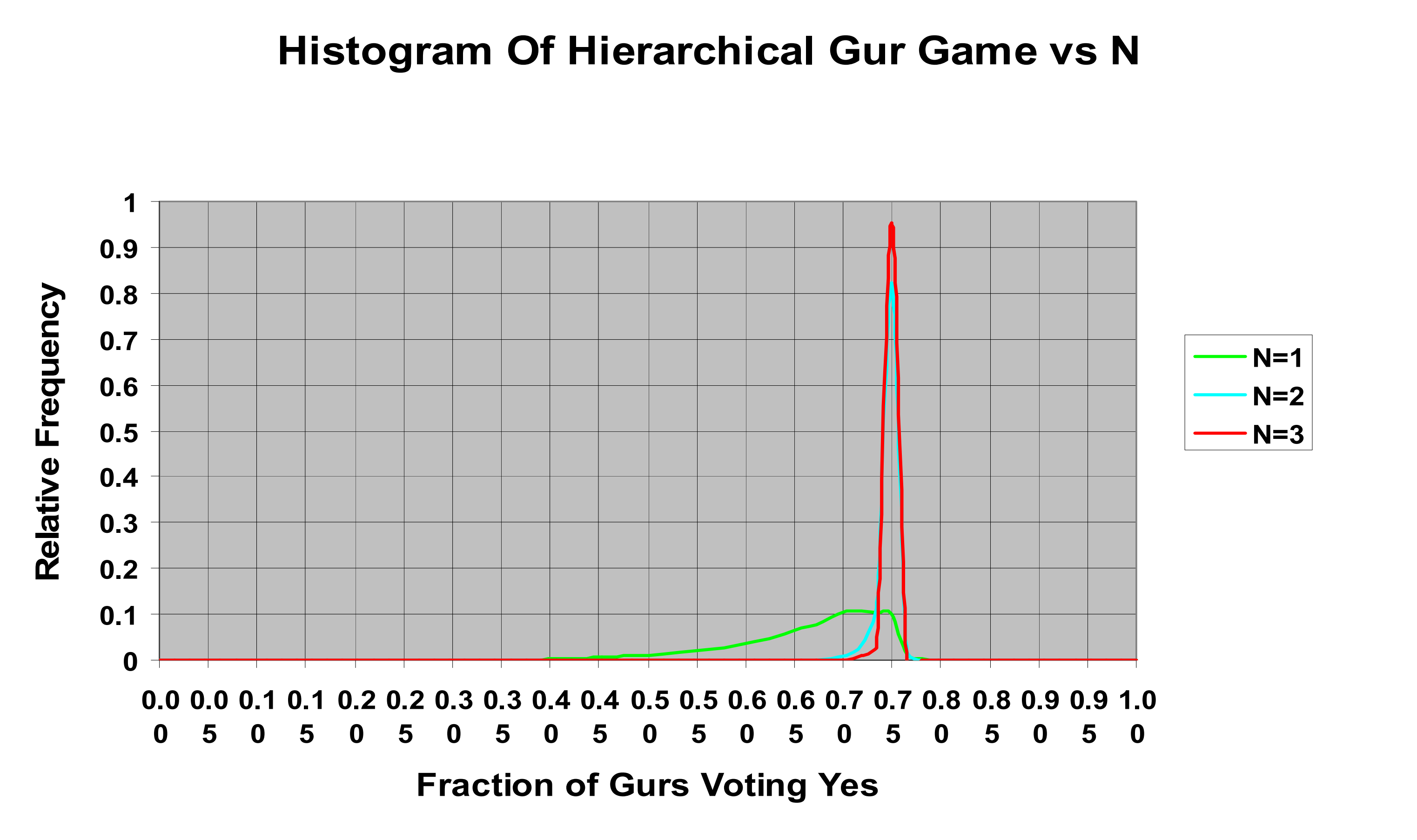
Scalable Sensor Network Resolution

- Simulation 4: Baseline



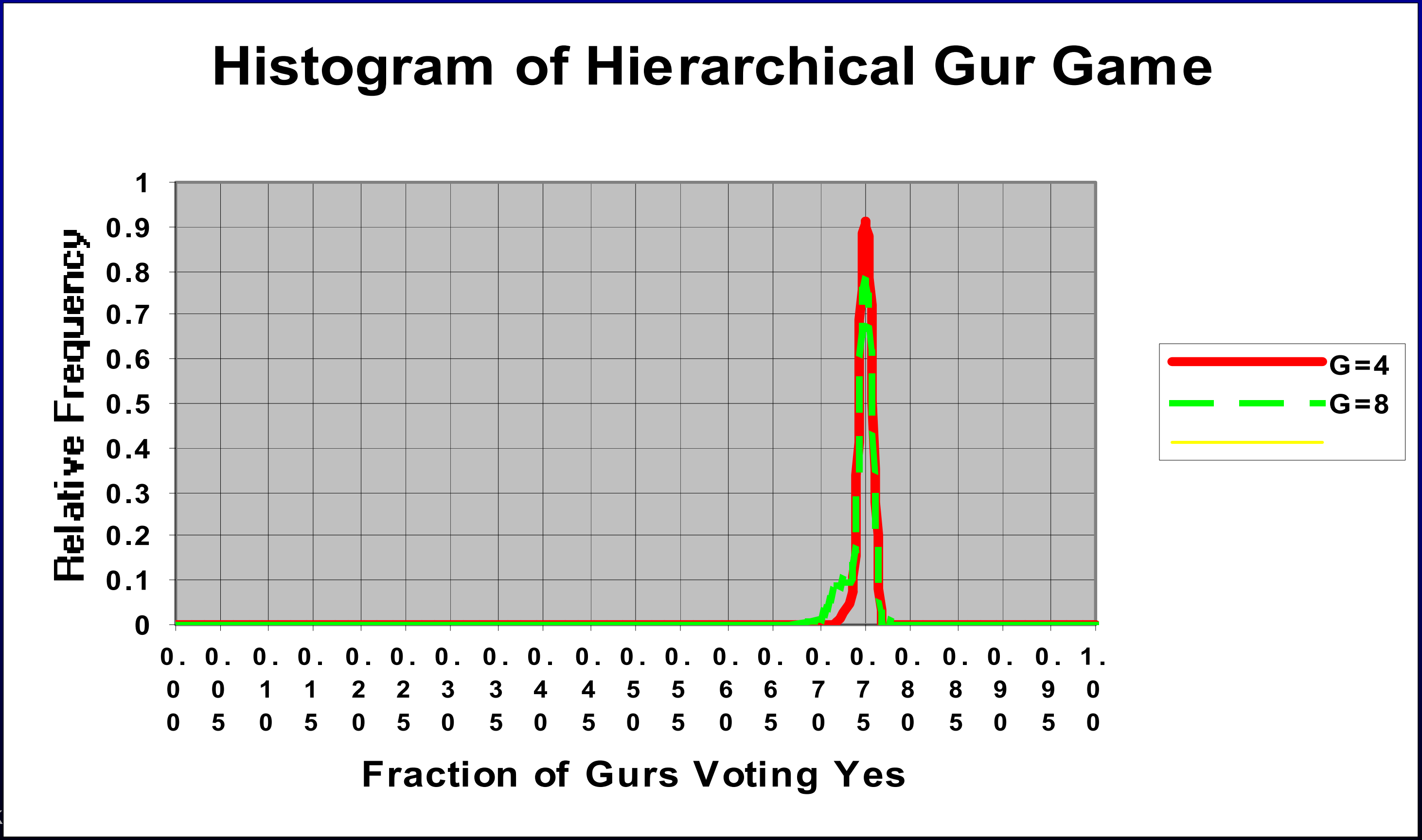
Scalable Sensor Network Resolution

- Simulation 5: Hierarchical



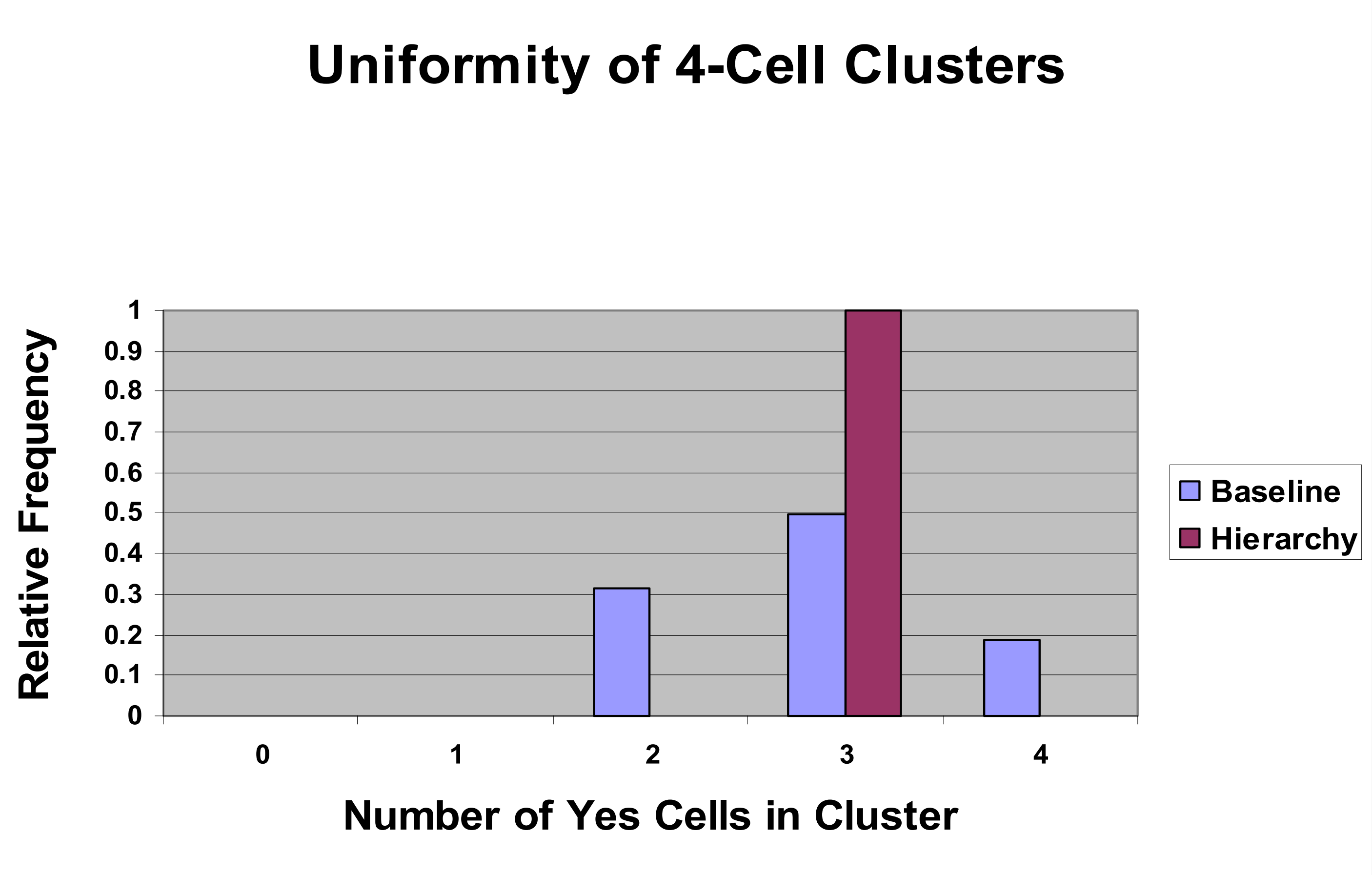
Scalable Sensor Network Resolution

- Simulation 6: Number of hierarchical levels



Scalable Sensor Network Resolution

- Simulation 7: Uniform coverage



Sweet Spot for Parameters

Binomial Effect (Noise, i.e., no locking) ---1
Good locking and Good Duty Cycle---------2
Good locking and Bad Duty Cycle-----------3
Bad locking ---4

alpha = 1000 narrow peak

N →	1	2	3	4	5	6	7	8	9	10
↓ Pk Height										
0.4	1	1	1	1	1	1	1	1	1	1
0.5	1	1	1	1	1	1	1	1	1	1
0.6	1	1	1	1	1	1	1	1	1	1
0.8	1	1	1	2	2	3	3	3	3	3
0.99	1	3	3	3	3	3	3	3	3	3

Sweet Spot for Parameters

Binomial Effect (Noise, i.e., no locking) ---1
Good locking and Good Duty Cycle---------2
Good locking and Bad Duty Cycle-----------3
Bad locking ---4

alpha = 20 moderate peak

N →	1	2	3	4	5	6	7	8	9	10
↓ Pk Height										
0.4	1	1	1	1	1	1	1	1	1	1
0.5	1	1	1	1	1	1	1	1	1	1
0.6	1	1	1	1	1	1	1	2	2	2
0.8	1	1	1	2	2	2	3	3	3	3
0.99	1	3	3	3	3	3	3	4	4	4

Sweet Spot for Parameters

Binomial Effect (Noise, i.e., no locking) ---1
Good locking and Good Duty Cycle---------2
Good locking and Bad Duty Cycle-----------3
Bad locking ---------------------------------------4

alpha = 5 broad peak

N →	1	2	3	4	5	6	7	8	9	10
↓ Pk Height										
0.4	1	1	1	1	1	1	1	1	1	1
0.5	1	1	1	1	1	1	1	1	1	1
0.6	1	1	1	1	1	2	2	2	2	2
0.8	1	1	1	2	2	2	3	3	4	3
0.99	1	2	3	4	4	2	4	4	4	4

The Mathematics

- For peak = 1, guaranteed convergence
- For peak < 1, binomial vs bias by r(f)
  - Find p(f), the probability (i.e., fraction of time) of having fraction f vote yes.
  - Use State Compression

Optimal Update Times for Out-of-Date Information

Problem:

When and how often should a user update a given piece of information as it goes further and further out-of-date?

Assumptions:

1) There is a cost $C>0$ of updating a given piece of information.

2) There is an expected value per unit time associated with having a piece of information that was updated t time units ago. This value is $f(t)$.

3) $f(t)$ is monotonically non-increasing.

Example:

A user is accessing a file.

This file is being modified by the system at a poisson rate of λ modifications per unit time.

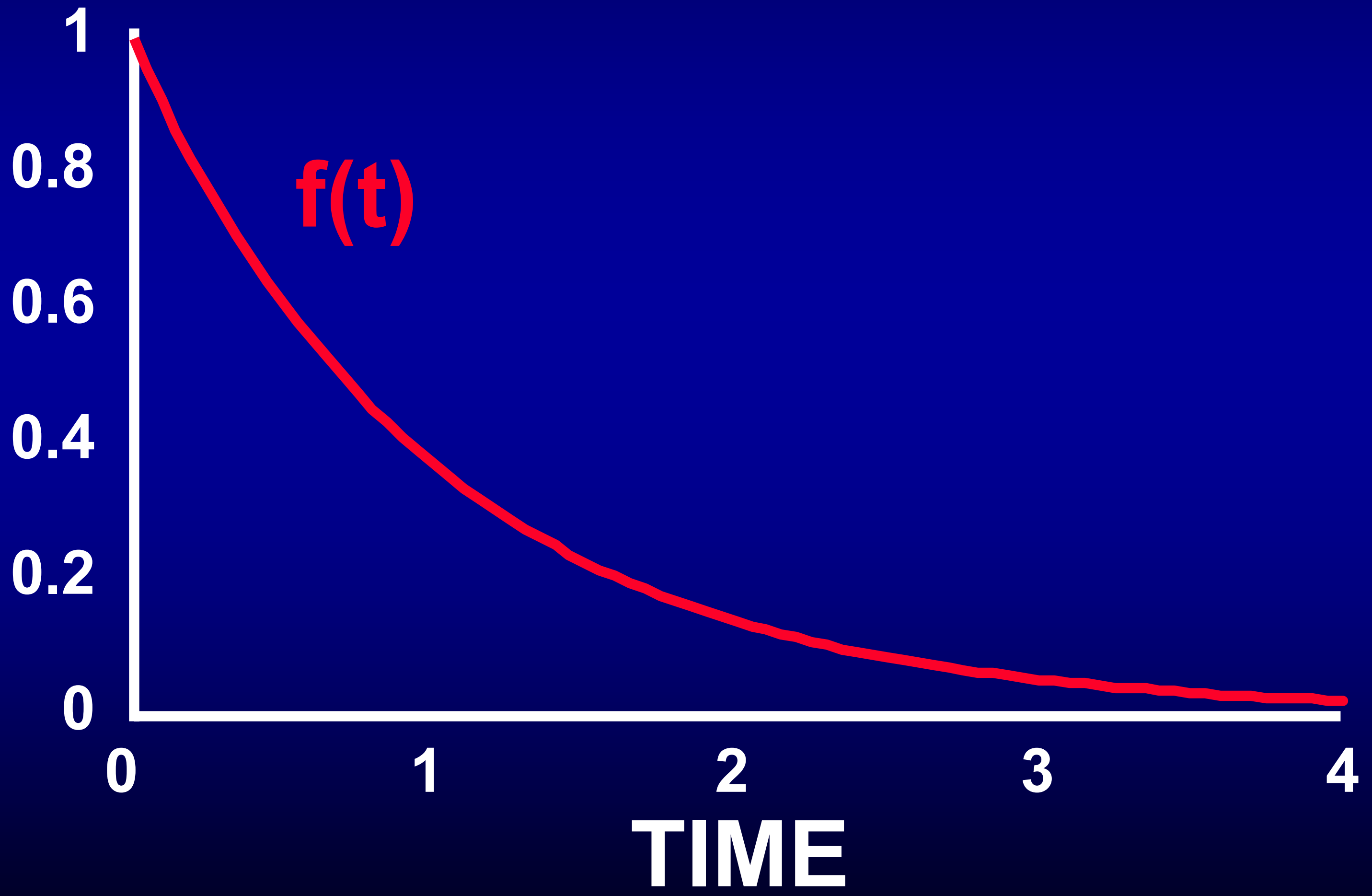
The user's expected value for having the file at t time units since his last update equals the probability that no modifications have been made to the file since his copy was sent.

Then $f(t)=e^{-\lambda t}$.

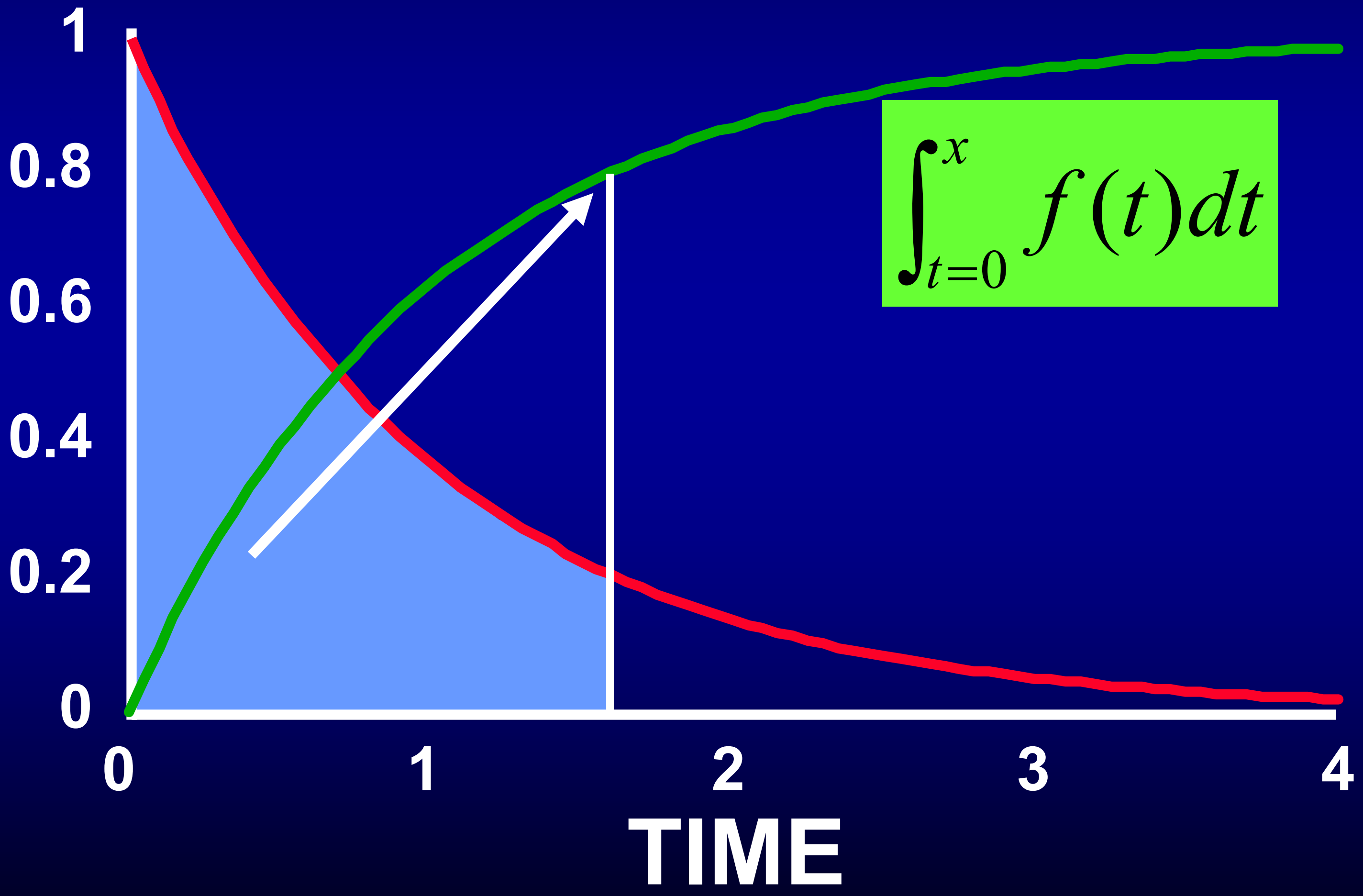
Question:

Given f(t) and C, When and how often should a user update a given piece of information?

Value of Out-of-Date Information

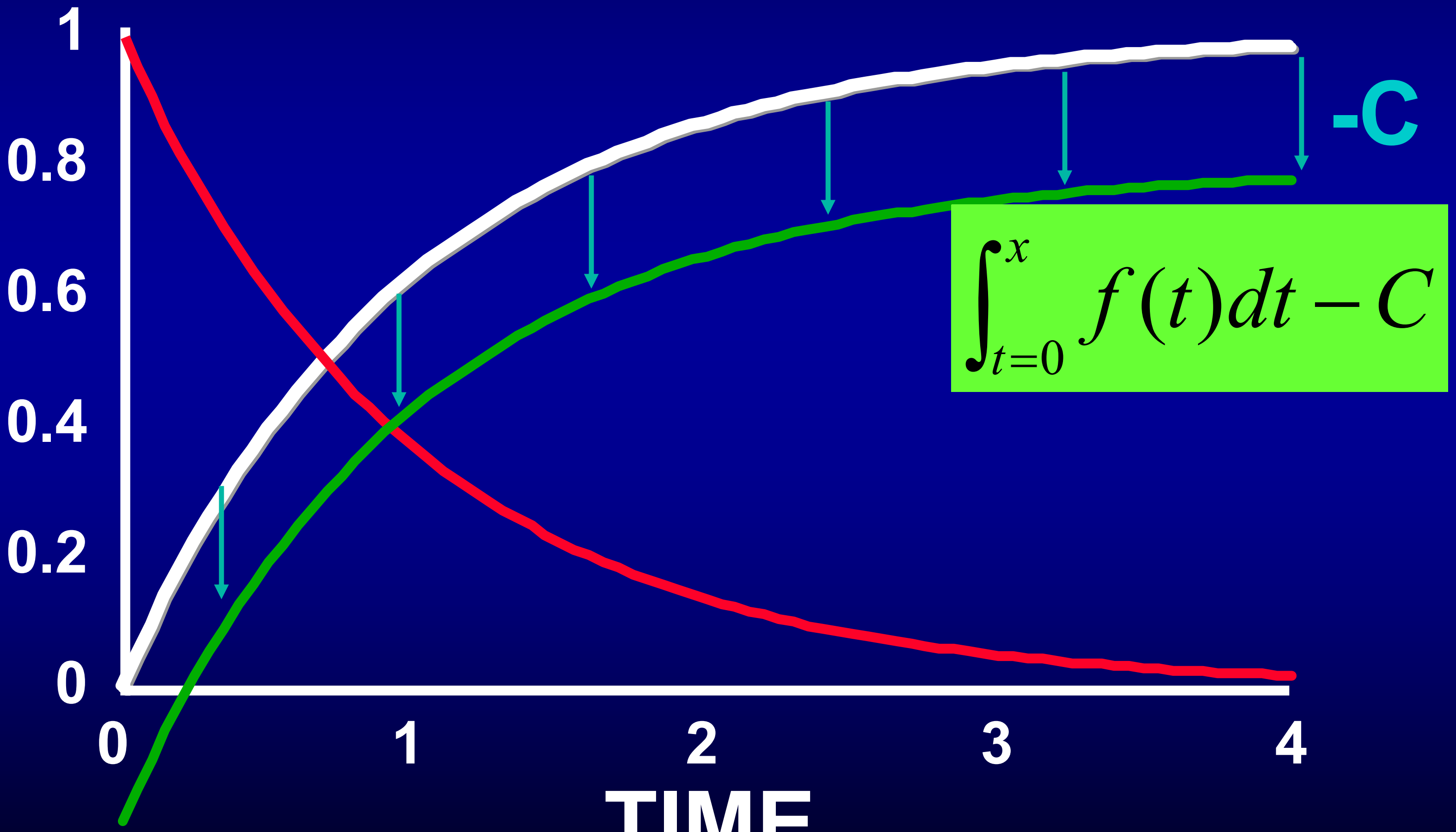


Total Value Gained Through Time x



$$\int_{t=0}^{x} f(t)dt$$

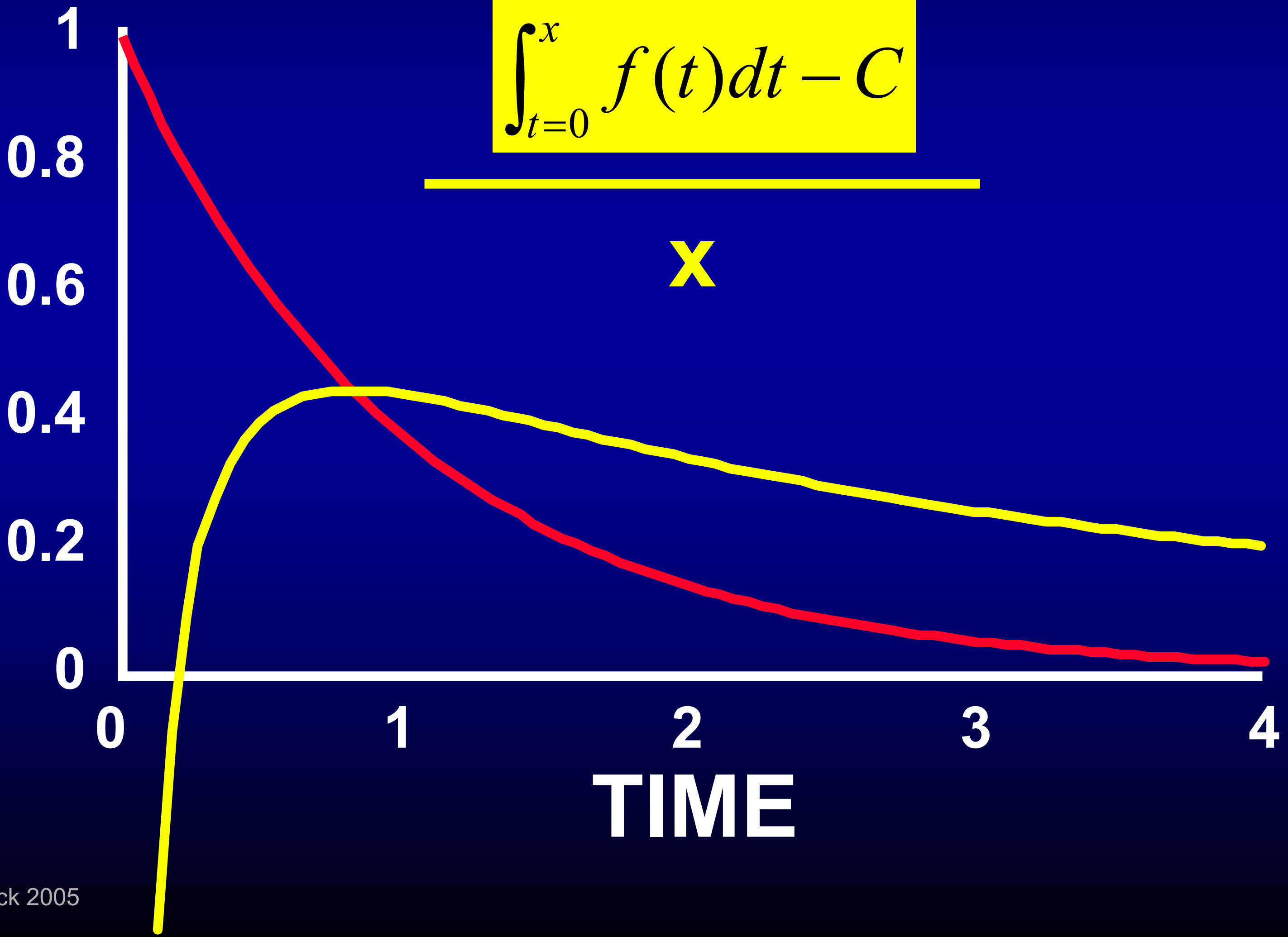
Total Value Gained Including Update Cost Through Time x



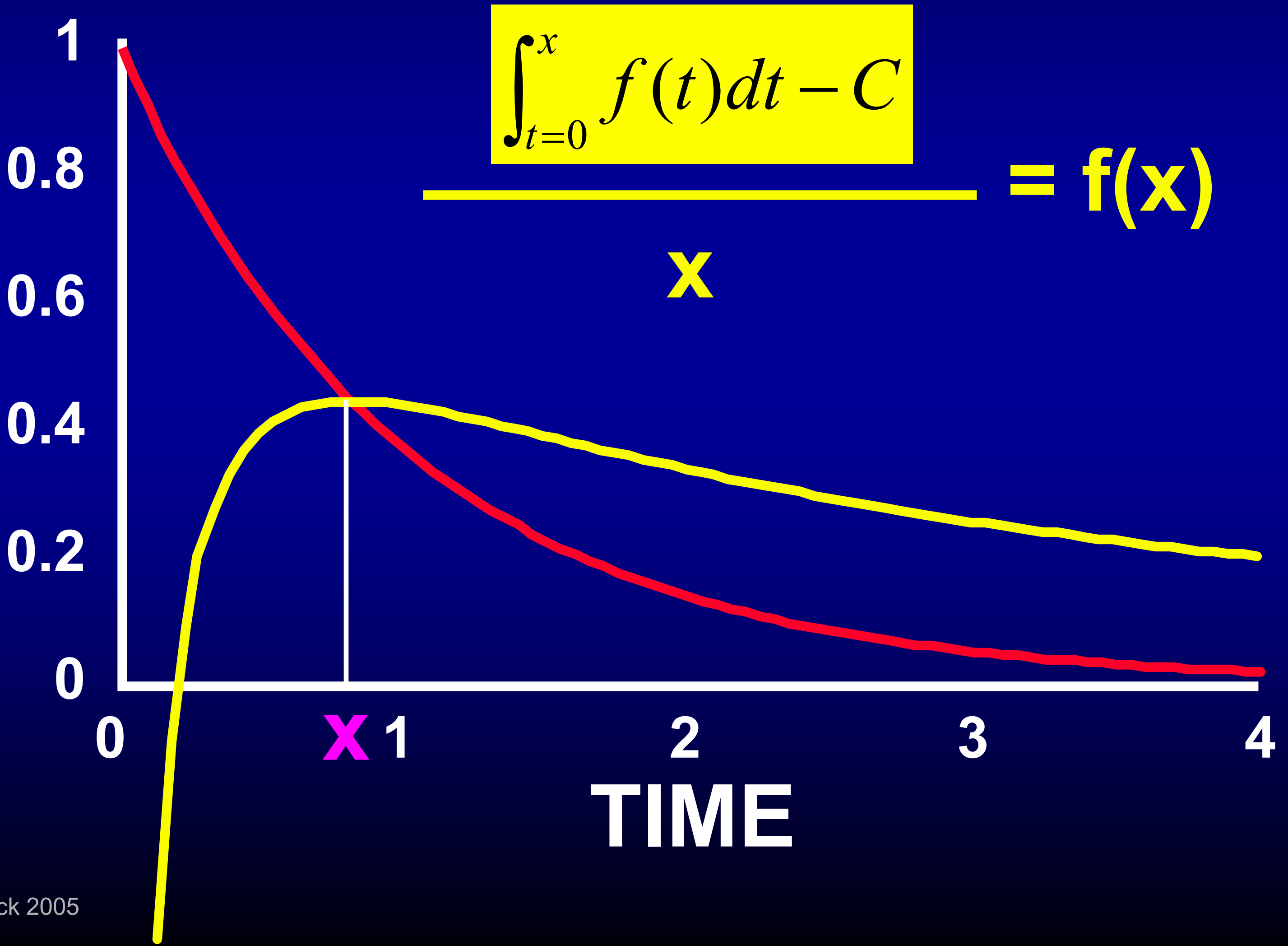
$$\int_{t=0}^{x} f(t)dt - C$$

Average Value Gained per Unit Time

$$\frac{\int_{t=0}^{x} f(t)dt - C}{x}$$

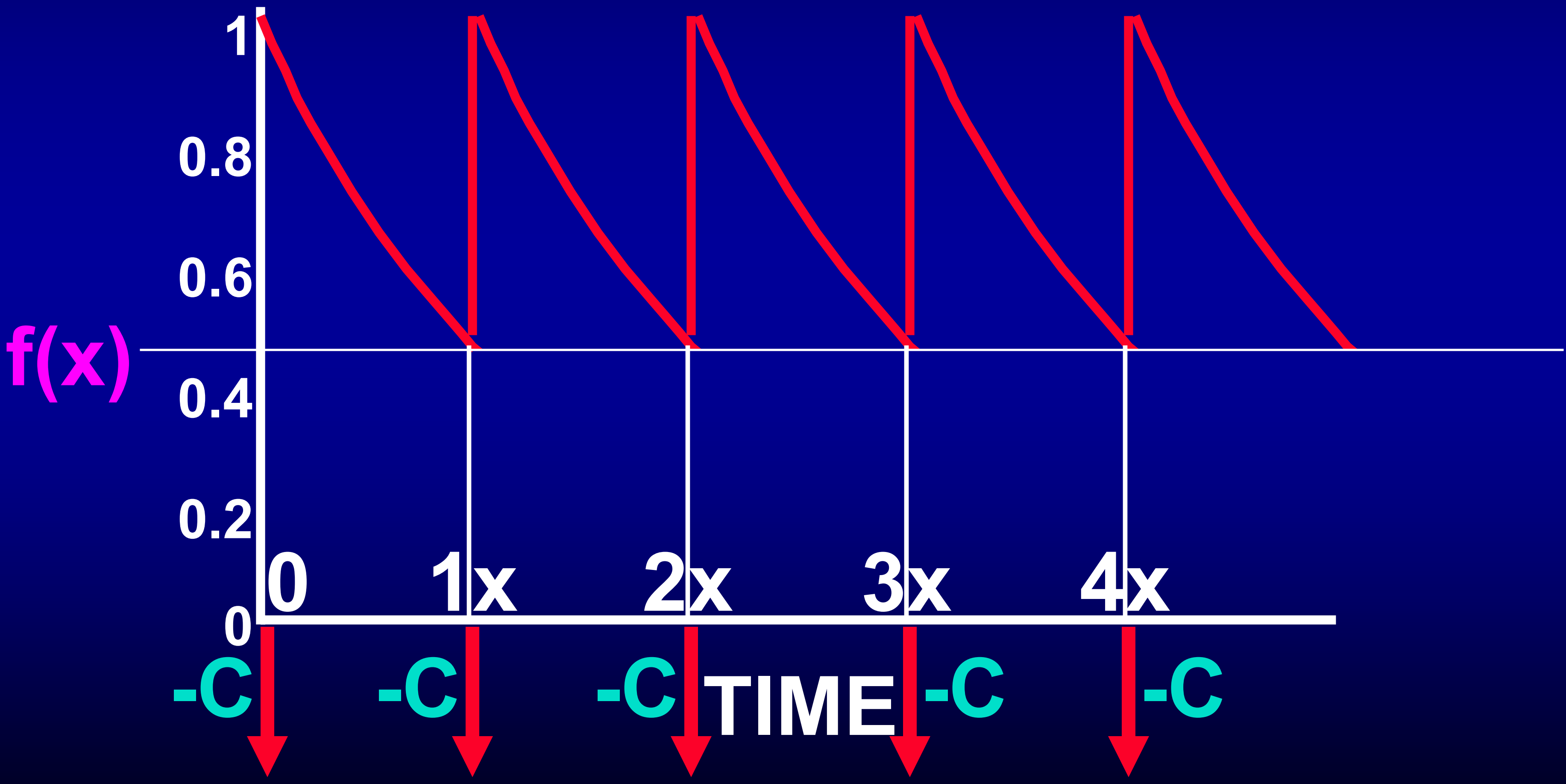


Average Value Gained per Unit Time is a maximum when:



$$\frac{\int_{t=0}^{x} f(t)dt - C}{x} = f(x)$$

Value Gained Over Multiple Updates



Funding, etc

- PostDoc Gone: Job security an issue
- GSR working on Gur algorithm at near-zero funding

Budget Issues

Case	12 Mo Extens	One Post Doc	One GSR	Funds Avail 6/30/05	Extra Funds Needed	Total Funds
1	No	No	No	$35,745	0	$35,745
2	Yes	No	No	$35,745	0	$35,745
3	Yes	Yes	No	$35,745	$153,054	$188,799
4	Yes	No	Yes	$35,745	$128,174	$163,919
5	Yes	Yes	Yes	$35,745	$221,187	$257,572

Thank You

www.lk.cs.ucla.edu