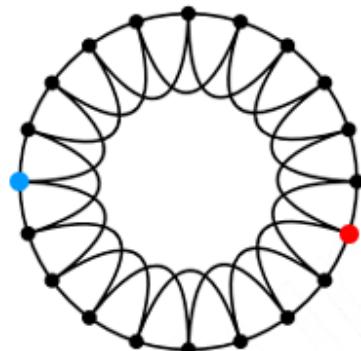


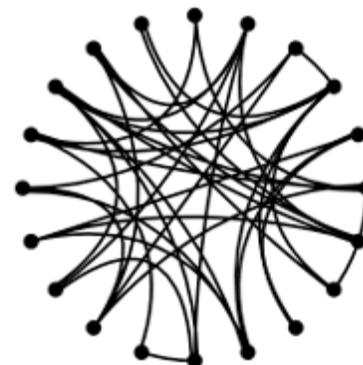
# **Small-World Phenomena and Decentralized Search**

# Recap: Small-World:

- Real networks: low diameter, high clustering
- But  $G_{np}$  is low diameter, no clustering
- How can we at the same time have high clustering and small diameter?



High clustering  
High diameter



Low clustering  
Low diameter

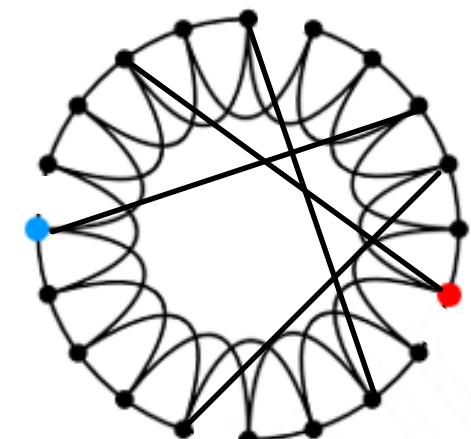
- Clustering implies edge “locality”
- Randomness enables “shortcuts”

# Solution: The Small-World Model

## Small-World Model [Watts-Strogatz '98]

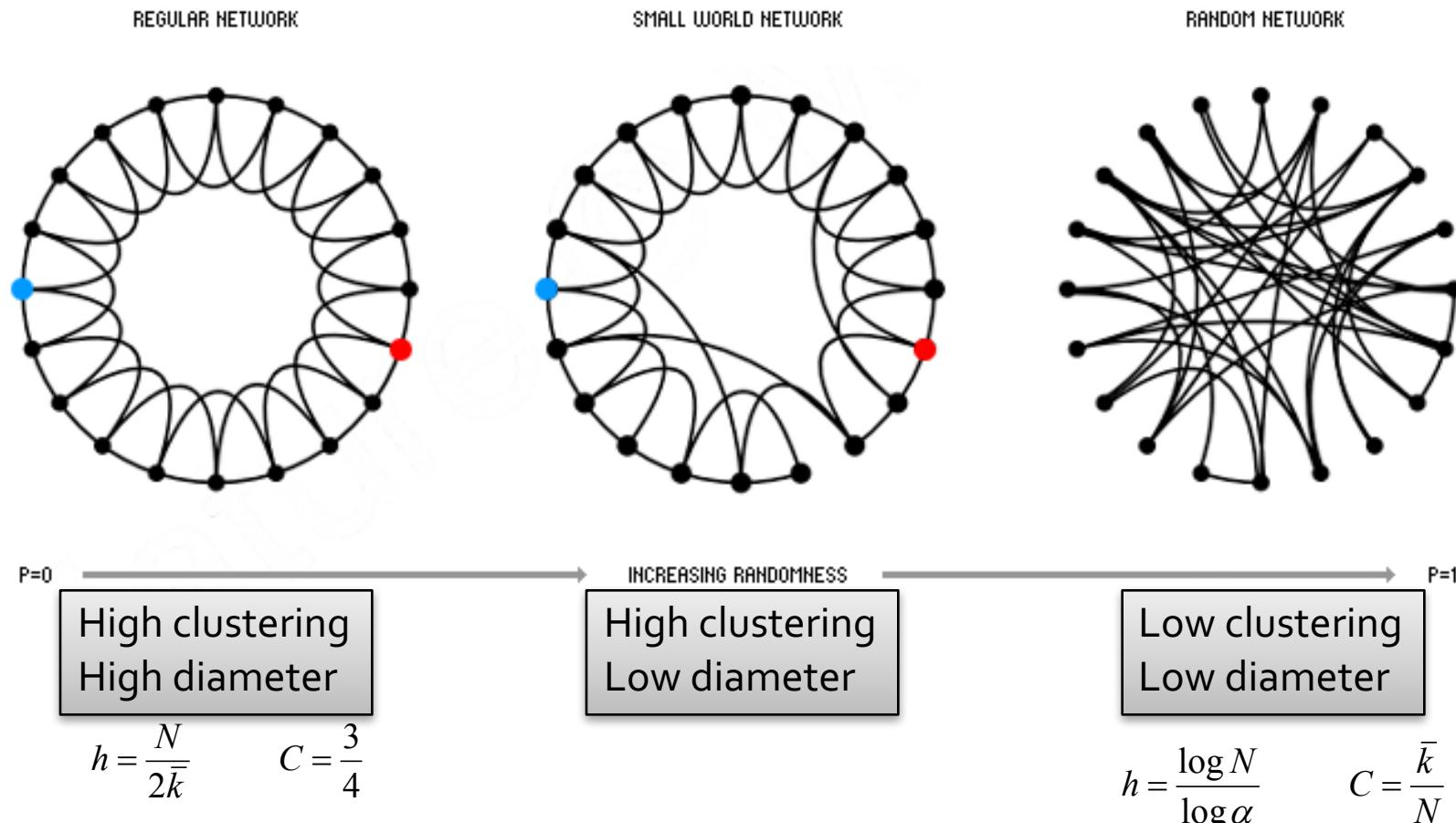
Two components to the model:

- **(1) Start with a low-dimensional regular lattice**
  - (In our case we are using a ring as a lattice)
  - Has high clustering coefficient
- **Now introduce randomness (“shortcuts”)**
- **(2) Rewire:**
  - Add/remove edges to create shortcuts to join remote parts of the lattice
  - For each edge with prob.  $p$  move the other end to a random node

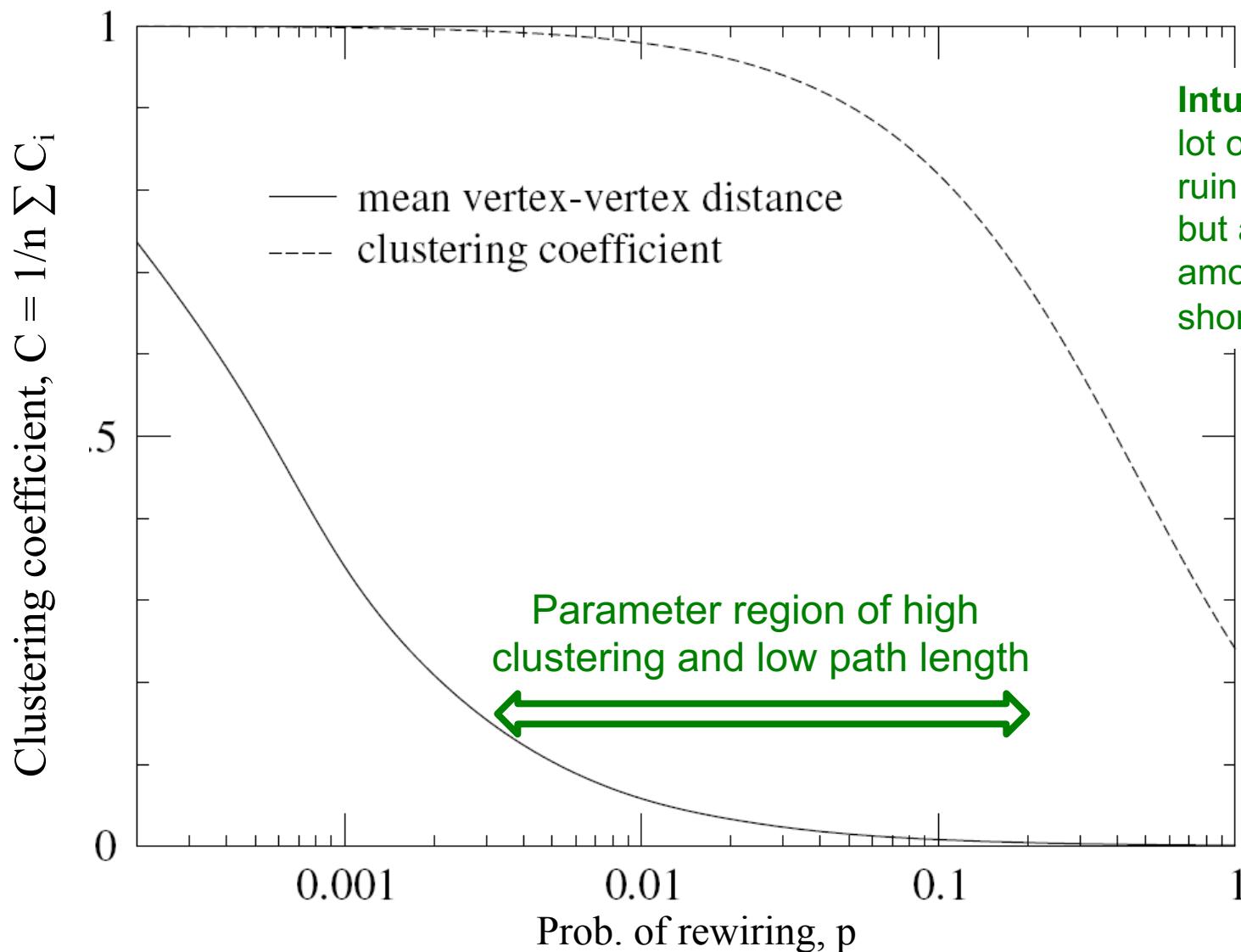


# Small-World - How?

- Could a network with high clustering be at the same time a small world?



# The Small-World Model

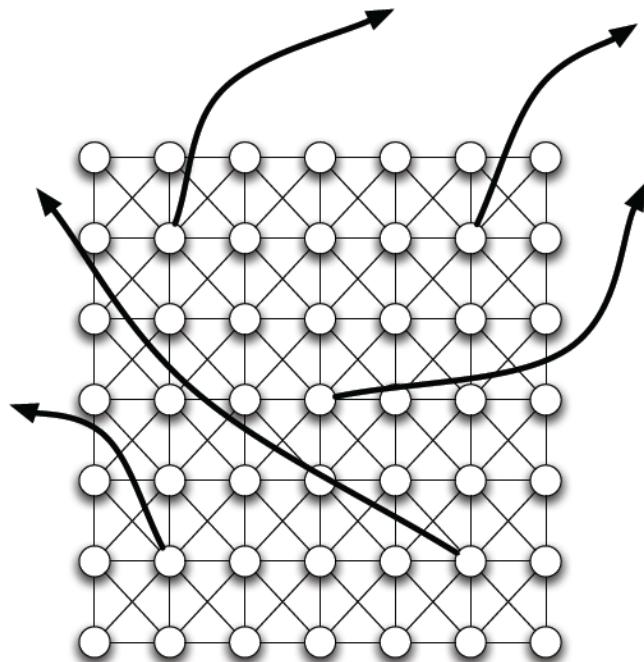


**Intuition:** It takes a lot of randomness to ruin the clustering, but a very small amount to create shortcuts.

# Diameter of the Watts-Strogatz

## ■ Alternative formulation of the model:

- Start with a square grid
- Each node has 1 random long-range edge
  - Each node has 1 spoke. Then randomly connect them.



$$C_i = \frac{2 \cdot e_i}{k_i(k_i - 1)} \geq \frac{2 \cdot 12}{9 \cdot 8} \geq 0.33$$

There are already 12 triangles in the grid and the long-range edge can only close more.

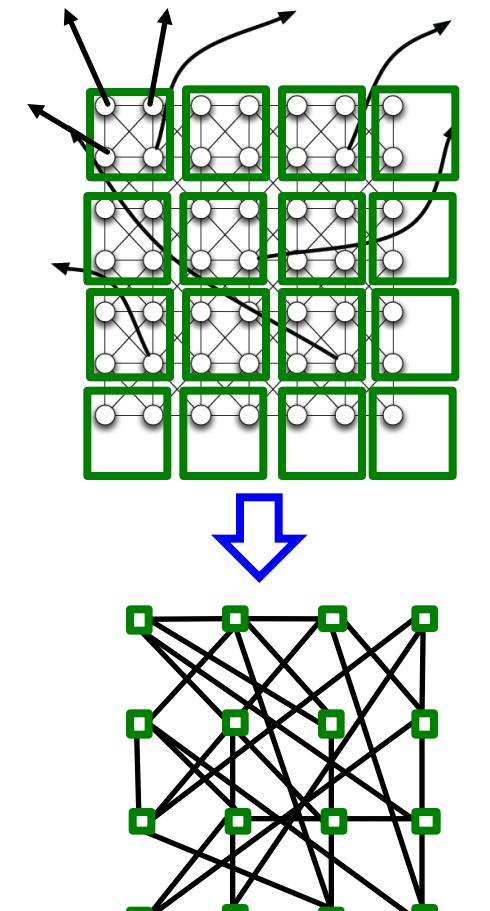
**What's the diameter?**  
**It is  $O(\log(n))$**   
**Why?**

# Diameter of the Watts-Strogatz

## ■ Proof:

- Consider a graph where we contract  $2 \times 2$  subgraphs into supernodes
- Now we have 4 long-range edges sticking out of each supernode
  - **4-regular random graph!**
- Thm. about  $G_{np}$  tell us we have short paths between super nodes.
- We can turn this into a path in the original graph by adding at most 2 steps per long range edge (by having to traverse internal nodes)

⇒ **Diameter of the model is  $O(2 \log n)$**



**4-regular random graph**

Note that this analysis ignores edges between neighbors of super-nodes, but this does not matter since those edges would make the diameter only go further down.

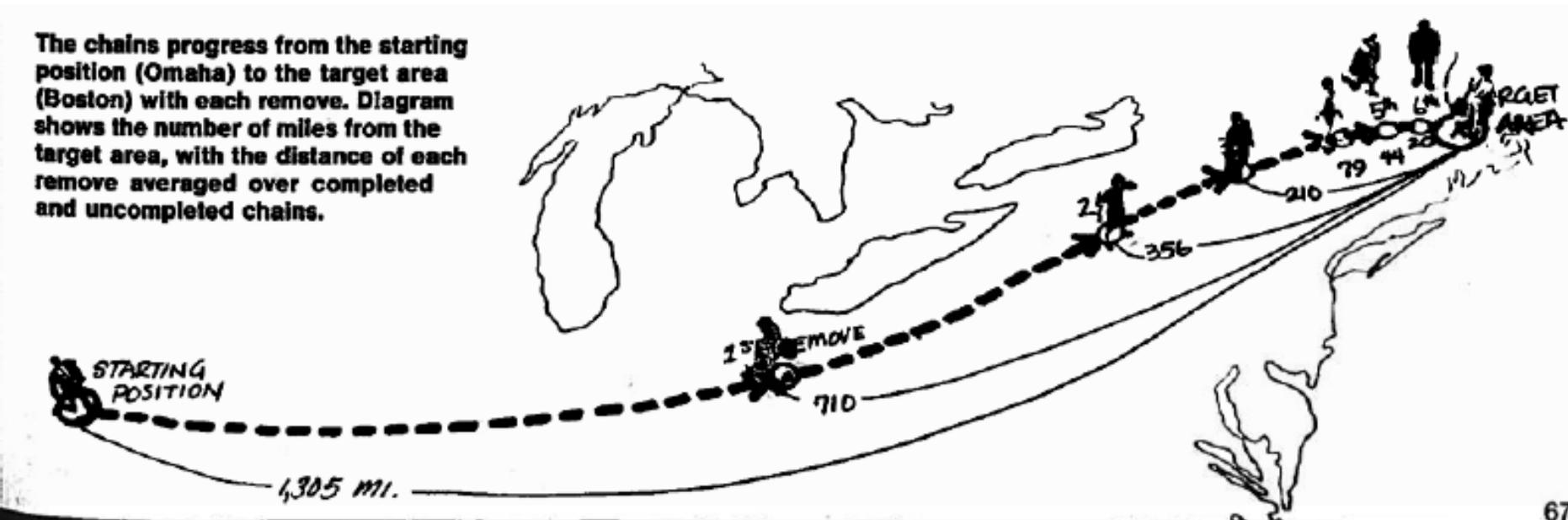
# Small-World: Summary

- Could a network with high clustering be at the same time a small world?
  - Yes! You don't need more than a few random links
- The Watts Strogatz Model:
  - Provides insight on the interplay between clustering and the small-world
  - Captures the structure of many realistic networks
  - Accounts for the high clustering of real networks
  - Does not lead to the correct degree distribution
  - Does not enable navigation (next)

# How to Navigate the Network?

- (1) What is the structure of a social network?
- **(Today) What strategies do people use to route and find the target?**

The chains progress from the starting position (Omaha) to the target area (Boston) with each remove. Diagram shows the number of miles from the target area, with the distance of each remove averaged over completed and uncompleted chains.

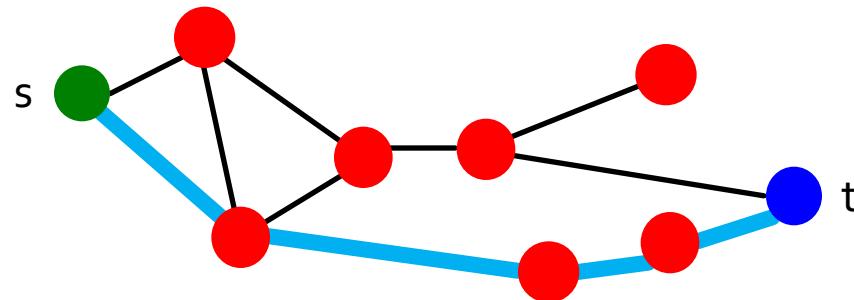


How would you go about finding the path?

# Decentralized Search

## The setting:

- $s$  only knows **locations** of its friends and location of the **target  $t$**
- $s$  does not know links of anyone else but itself
- **Geographic Navigation:**  $s$  “navigates” to a node geographically closest to  $t$
- **Search time T:** Number of steps to reach  $t$



# Overview of the Results

## Searchable

Search time T:

$$O((\log n)^\beta)$$

### Kleinberg's model

$$O((\log n)^2)$$

**Note:** We know these graphs have diameter  $O(\log n)$ .  
So in Kleinberg's model search time is polynomial in  $\log n$ ,  
while in Watts-Strogatz it is exponential (in  $\log n$ ).

## Not searchable

Search time T:

$$O(n^\alpha)$$

### Watts-Strogatz

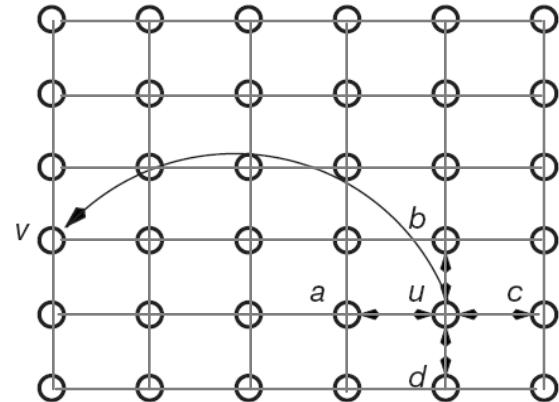
$$O(n^{\frac{2}{3}})$$

### Erdős–Rényi

$$O(n)$$

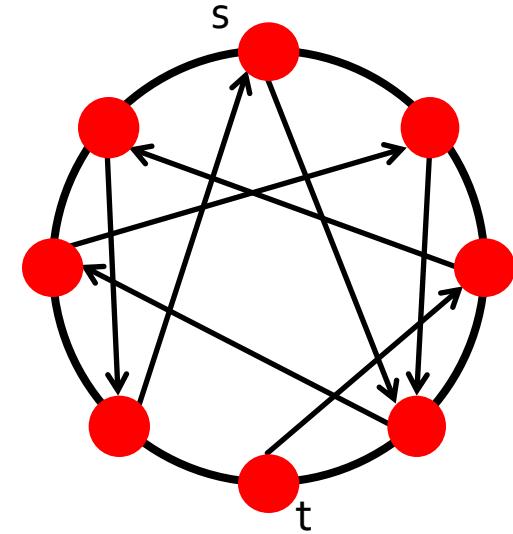
# Navigation in Watts-Strogatz

- **Model:** 2-dim grid where each node has 1 random edge
  - This is a small-world!
  - (Small-world = diameter  $O(\log n)$ )
- **Fact:** A decentralized search algorithm in Watts-Strogatz model needs  $n^{2/3}$  steps to reach  $t$  in expectation
  - Note: Even though paths of  $O(\log n)$  steps exist
- Note: All our calculations are asymptotic, i.e., we are interested in what happens as  $n \rightarrow \infty$



# Navigation in Watts-Strogatz

- Let's do the proof for 1-dimensional case
- Want to show Watts-Strogatz is NOT searchable
  - Bound the search time from below
- About the proof:
  - Setting:  $n$  nodes on a ring plus one random directed edge per node.
  - Search time is  $T \geq O(\sqrt{n})$ 
    - For  $d$ -dim. lattice:  $T \geq O(n^{d/(d+1)})$
  - Proof strategy: Principle of deferred decision
    - Doesn't matter when a random decision is made if you haven't seen it yet
    - Assume random long range link is only created once you get to the node



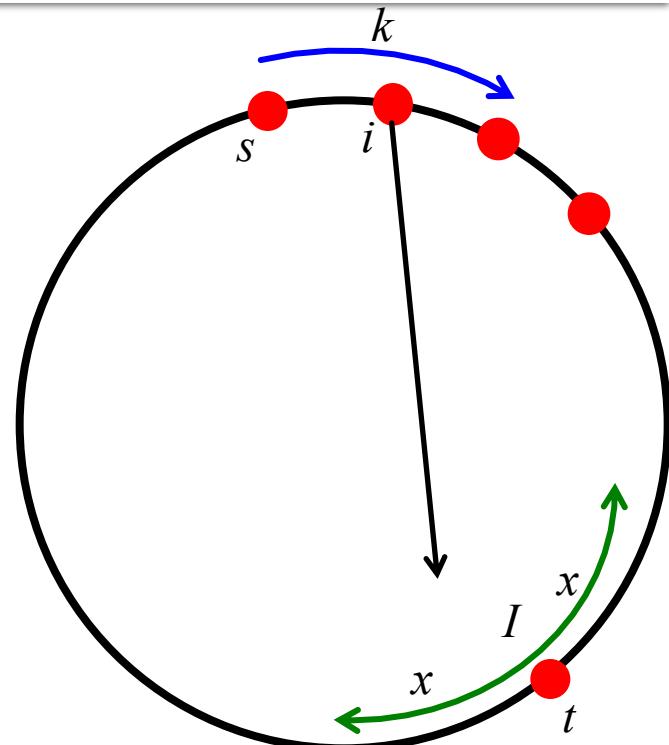
# Proof Sketch: Search time is $\geq O(n^{1/2})$

How long we have to walk before we jump? Overview of the proof:

- **Reason about event  $E$**

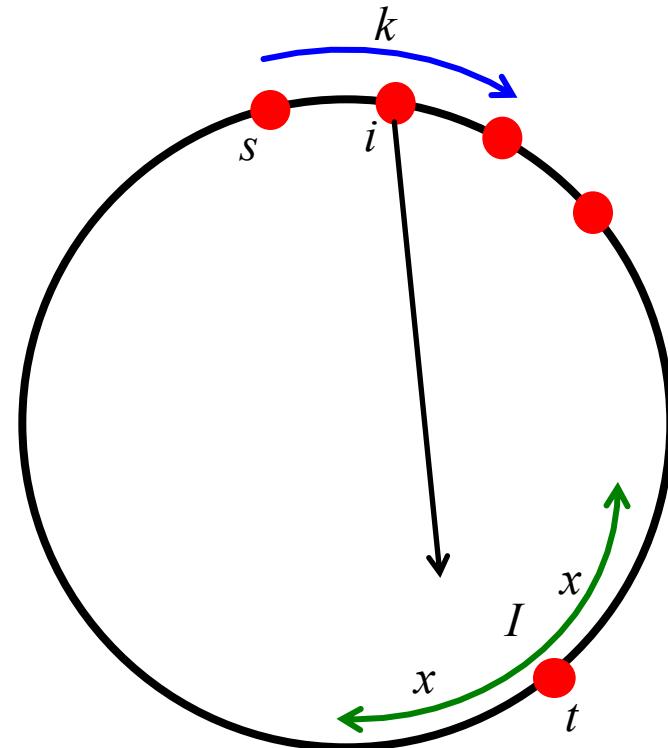
- $E$  = event that any of the first  $k$  nodes visited by the alg. has a link to  $I$  of width  $2 \cdot x$  nodes (for some  $x$ ) around target  $t$
- **We obtain:**  $P(E) \leq \frac{2kx}{n}$

- If  $E$  does not occur, then we walked at least  $k$  steps
- $E[\text{Search time}] \geq P(\text{not } E) * k$
- **So let's pick**  $k = x = \frac{1}{2} \sqrt{n}$  **then**  $P(E) \leq \frac{1}{2}$
- $E[\text{Search time}] \geq \frac{1}{2} * k = \frac{1}{2} * \frac{1}{2} \sqrt{n} = O(\sqrt{n})$



# Proof: Search time is $\geq O(n^{1/2})$

- We reason about the time needed to get into interval  $I$
- Let:  $E_i$  = event that long link out of node  $i$  points to some node in interval  $I$  of width  $2 \cdot x$  nodes (for some  $x$ ) around target  $t$
- Then:  $P(E_i) = \frac{2x}{n-1} \approx \frac{2x}{n}$  (in the limit of large  $n$ )  
 (haven't seen node  $i$  yet, but can assume random edge generation)



# Proof: Search time is $\geq O(n^{1/2})$

- $E$  = event that any of the first  $k$  nodes search algorithm visits has a link to  $I$
- **Then:**  $P(E) = P\left(\bigcup_i^k E_i\right) \leq \sum_i^k P(E_i) = k \frac{2x}{n}$
- **Let's choose**  $k = x = \frac{1}{2} \sqrt{n}$

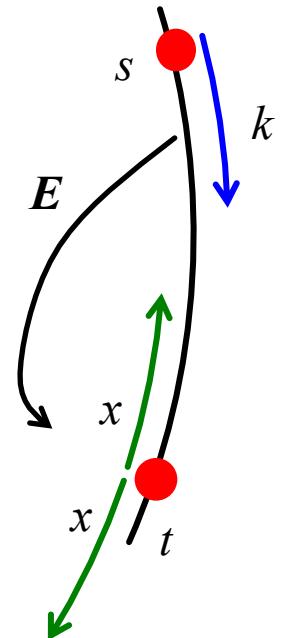
**Then:**

$$P(E) \leq 2 \frac{\left(\frac{1}{2} \sqrt{n}\right)^2}{n} = \frac{1}{2}$$

**Note:** Our alg. is deterministic and will choose to travel via a long- or short-range links using some deterministic rule.

The principle of deferred decision tells us that it does not really matter how we reached node  $i$ .

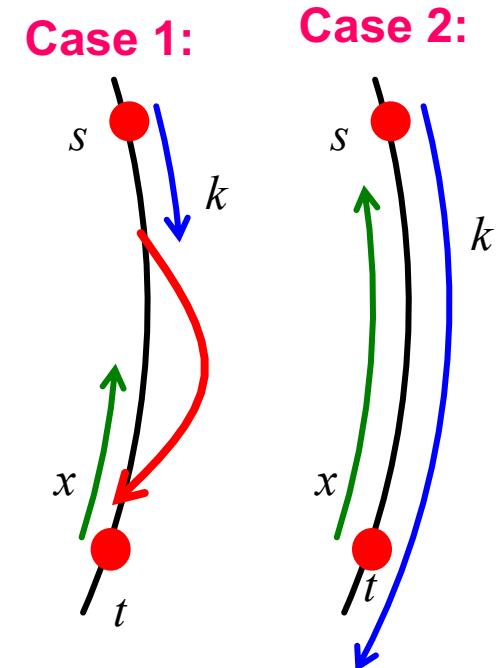
Its prob. of linking to interval  $I$  is:  $2x/n$ .



# Proof: Search time is $\geq O(n^{1/2})$

$$P(E) = P(\text{in } \frac{1}{2}\sqrt{n} \text{ steps we jump inside } \frac{1}{2}\sqrt{n} \text{ of } t) \leq \frac{1}{2}$$

- **Suppose** initial  $s$  is outside  $I$  and event  $E$  does not happen (first  $k$  visited nodes don't point to  $I$ )
- **Then** the search algorithm must take  $T \geq \min(k, x)$  steps to get to  $t$ 
  - (1) Right after we visit  $k$  nodes a good long-range link may occur
  - (2)  $x$  and  $k$  “overlap”, due to  $E$  not happening we have to walk at least  $x$  steps



# Proof: Search time is $\geq O(n^{1/2})$

Details

- **Claim:** Getting from  $s$  to  $t$  takes  $\geq \frac{1}{4}\sqrt{n}$  steps
- $\text{Search time} \geq P(E) * (\#\text{steps}) + P(\text{not } E) * \min(x, k)$
- **Proof:** We just need to put together the facts

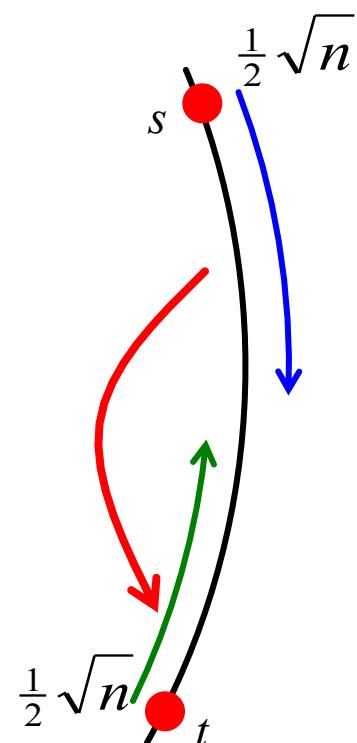
■ We already showed that for  $x = k = \frac{1}{2}\sqrt{n}$

- $E$  does not happen with prob.  $\frac{1}{2}$
- If  $E$  does not happen,  
we must traverse  $\geq \frac{1}{2}\sqrt{n}$  steps to get to  $t$

■ The expected time to get to  $t$  is then

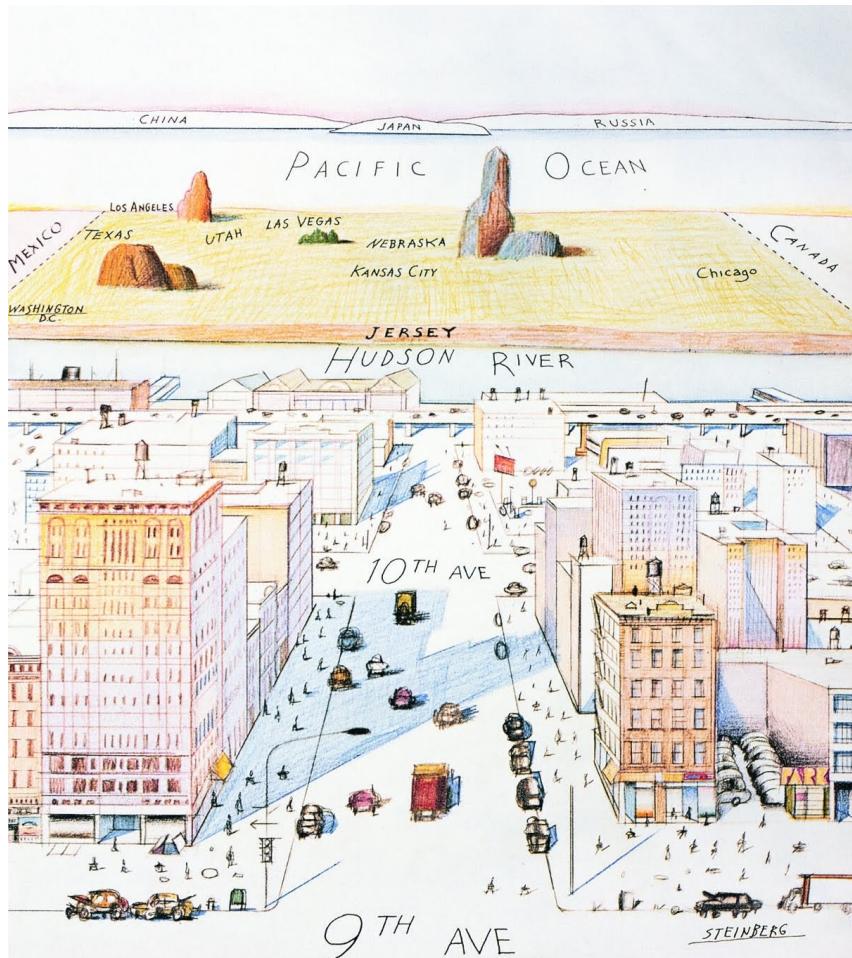
$$\geq P(E \text{ doesn't occur}) \cdot \min\{x, k\} =$$

$$= \frac{1}{2} \frac{1}{2} \sqrt{n} = \frac{1}{4} \sqrt{n}$$



# Navigable Small-World Graph?

- Watts-Strogatz graphs are **not searchable**
- **How do we make a searchable small-world graph?**
- **Intuition:**
  - Our long range links are not random
  - **They follow geography!**



Saul Steinberg, "View of the World from 9th Avenue"

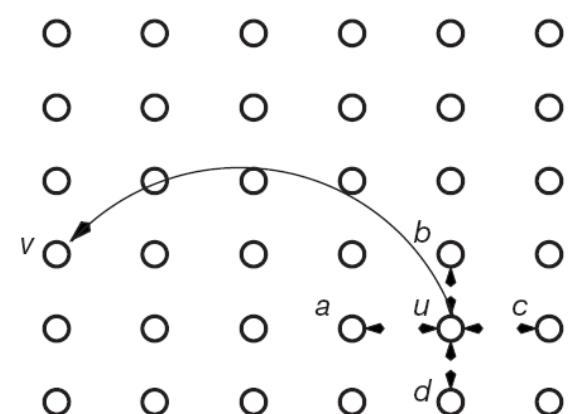
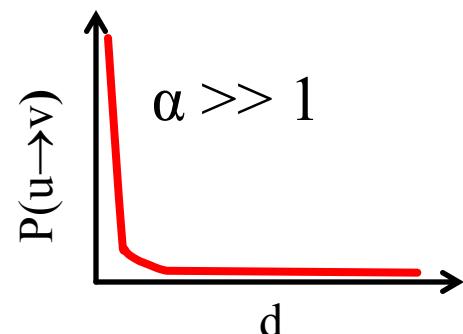
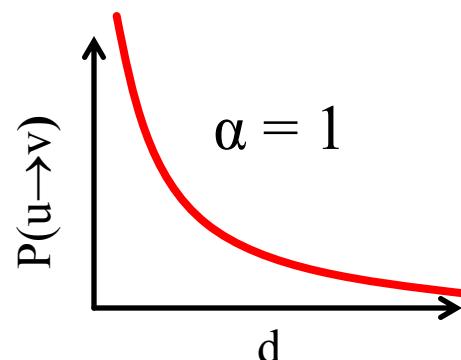
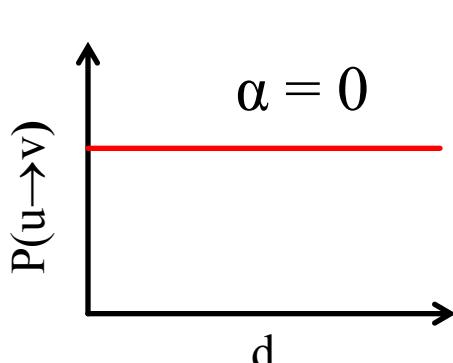
# Variation of the Model

## ■ Model [Kleinberg, Nature '01]

- Nodes still on a grid
- Node has one long range link
- Prob. of long link to node  $v$ :

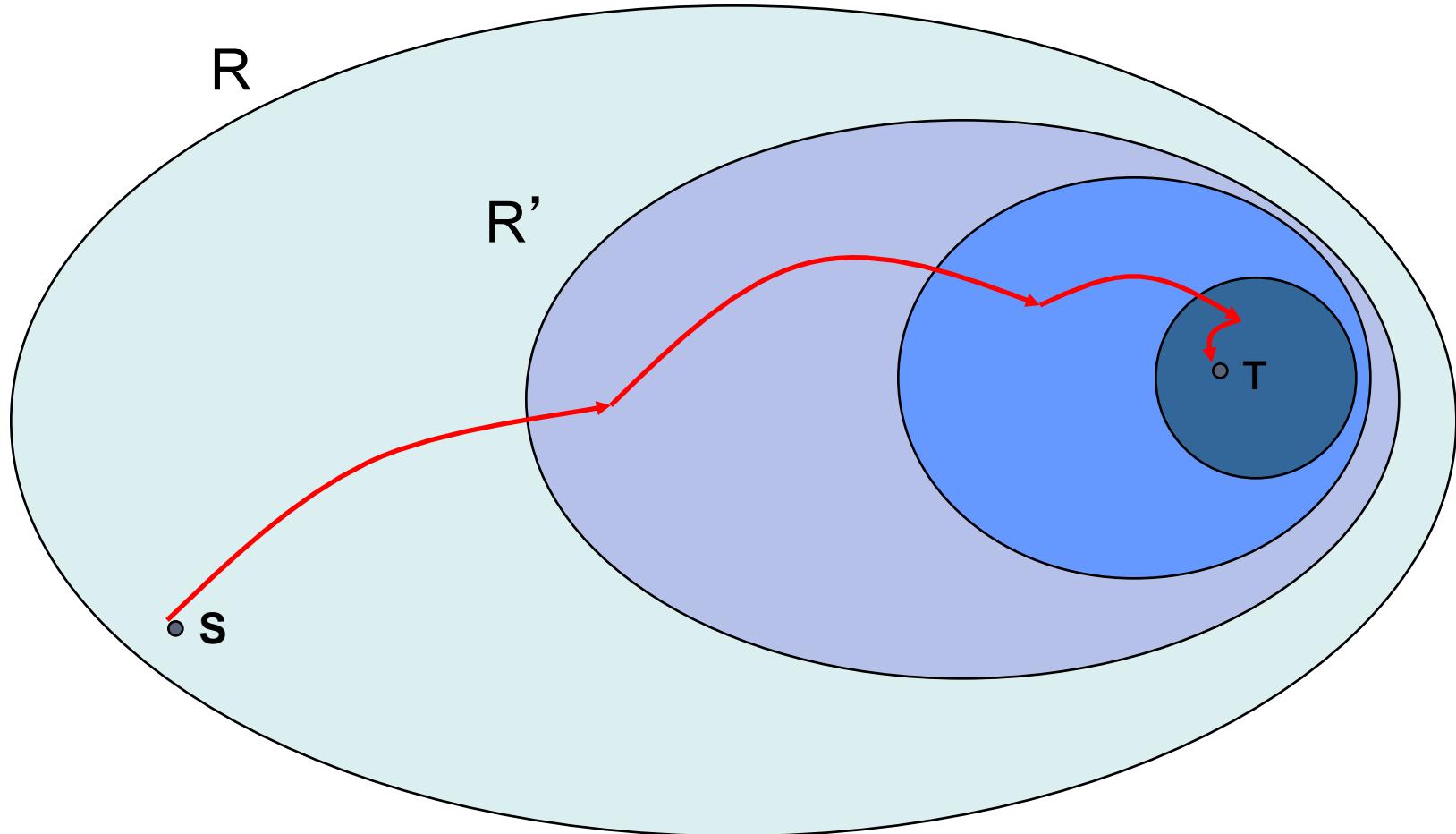
$$P(u \rightarrow v) \sim d(u,v)^{-\alpha}$$

- $d(u,v)$  ... grid distance between  $u$  and  $v$
- $\alpha$  ... parameter  $\geq 0$



$$P(u \rightarrow v) = \frac{d(u,v)^{-\alpha}}{\sum_{w \neq u} d(u,w)^{-\alpha}}$$

# Why Does It Work?



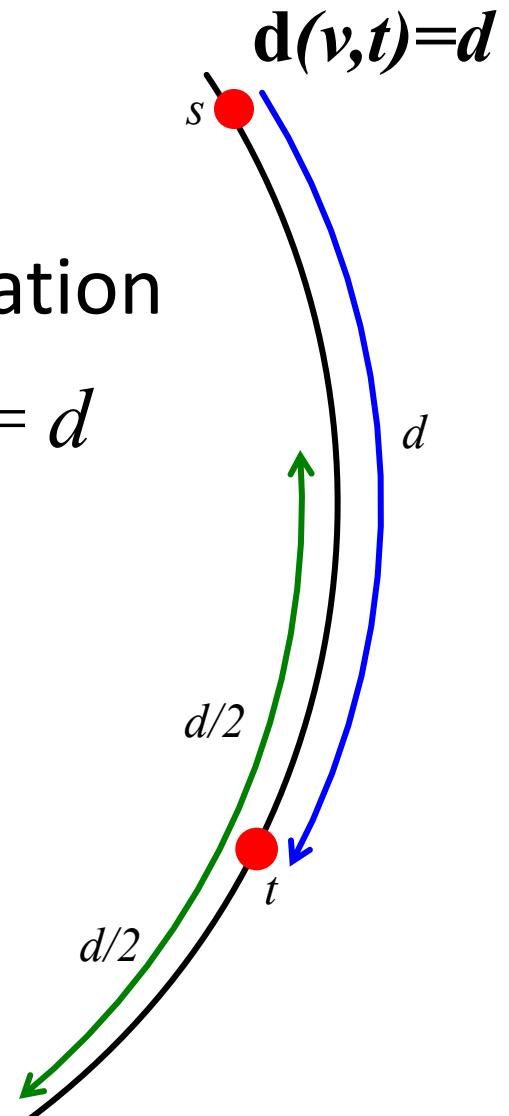
# Kleinberg's Model in 1-Dimension

We analyze 1-dim case:

- **Claim:** For  $\alpha = 1$  we can get from  $s$  to  $t$  in  $O(\log(n)^2)$  steps in expectation
- **Assume:** For some node  $v$ :  $d(v, t) = d$
- **Set interval:**  $I = d$
- **Fact:** (next two slides give a proof of this fact)

$$P\left(\begin{array}{l} \text{Long range} \\ \text{link from } v \\ \text{points to a} \\ \text{node in } I \end{array}\right) = O\left(\frac{1}{\ln n}\right)$$

Why is this cool? As  $d$  gets bigger,  $I$  gets wider, but the prob. is independent of  $d$ .



# Kleinberg's Model in 1-D

Details

- First we need:  $P(v \text{ points to } w) =$

$$P(v \rightarrow w) = \frac{d(v, w)^{-1}}{\sum_{u \neq v} d(v, u)^{-1}}$$

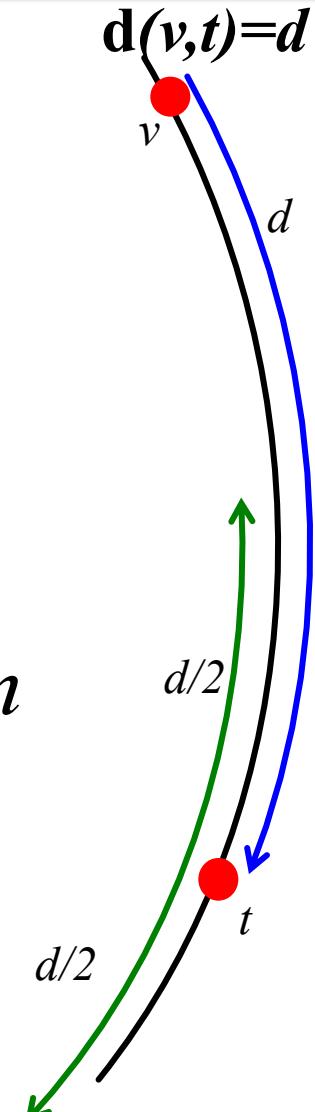
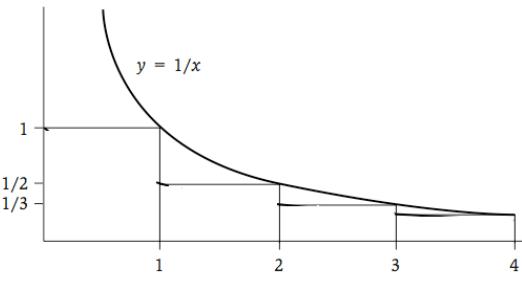
- What is the normalizing const?

$$\sum_{u \neq v} d(u, v)^{-1} = \sum_{\substack{\text{all possible} \\ \text{distances } d \\ \text{from } 1 \rightarrow n/2}} 2 \frac{1}{d} = 2 \sum_{d=1}^{n/2} \frac{1}{d} \leq 2 \ln n$$

At every distance  $d$  there are 2 nodes.  
Prob. of linking to one is  $1/d$ .

**Note:**

$$\sum_{d=1}^{n/2} \frac{1}{d} \leq 1 + \int_1^{n/2} \frac{dx}{x} = 1 + \ln\left(\frac{n}{2}\right) = \ln n$$



# Kleinberg's Model in 1-D

Details

- Next we need:  $P(v \text{ points to } I) =$

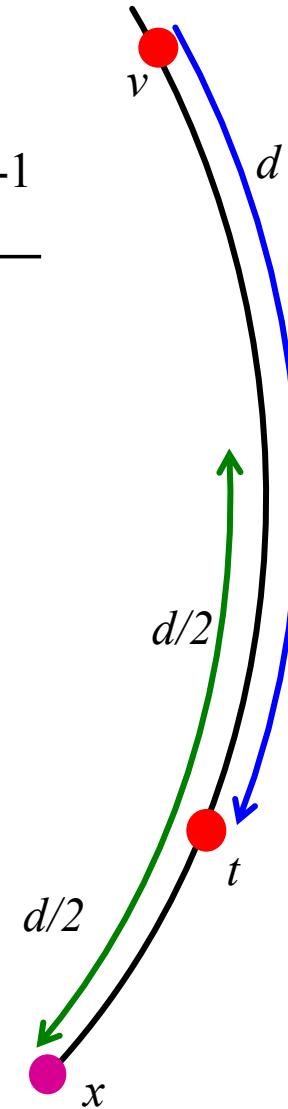
$$P(v \text{ points to } I) = \sum_{w \in I} P(v \rightarrow w) \geq \sum_{w \in I} \frac{d(v, w)^{-1}}{2 \ln n}$$

$$= \frac{1}{2 \ln n} \sum_{w \in I} \frac{1}{d(v, w)} \geq \frac{1}{2 \ln n} d \frac{2}{3d} = \frac{1}{3 \ln n}$$

What's the  
smallest of  
these terms?  
**All terms**  
 $\geq 2/(3d)$

$$= O\left(\frac{1}{\ln n}\right)$$

Note:  
 $d(v, x) = 3d/2$



# Kleinberg's Model in 1-D

- **So, we have:**

- $I$  ... interval of  $d/2$  around  $t$  (where  $d = d(v, t)$ )

- $P(\text{long link of } v \text{ points to } I) = 1/\ln(n)$

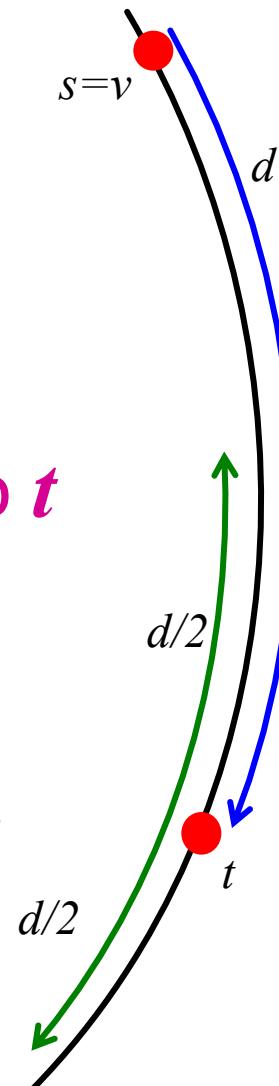
- **In expected # of steps  $\leq \ln(n)$  you get into  $I$ , and thus you halve the distance to  $t$**

- **How many times do we have to walk  $\ln(n)$  steps?**

- Distance can be halved at most  $\log_2(n)$  times

- So expected time to reach  $t$ :

- $O(\log_2(n)^2)$**



# Overview of the Results

## Searchable

Search time T:

$$O((\log n)^\beta)$$

### Kleinberg's model

$$O((\log n)^2)$$

**Note:** We know these graphs have diameter  $O(\log n)$ .  
So in Kleinberg's model search time is polynomial in  $\log n$ ,  
while in Watts-Strogatz it is exponential (in  $\log n$ ).

## Not searchable

Search time T:

$$O(n^\alpha)$$

### Watts-Strogatz

$$O(n^{\frac{2}{3}})$$

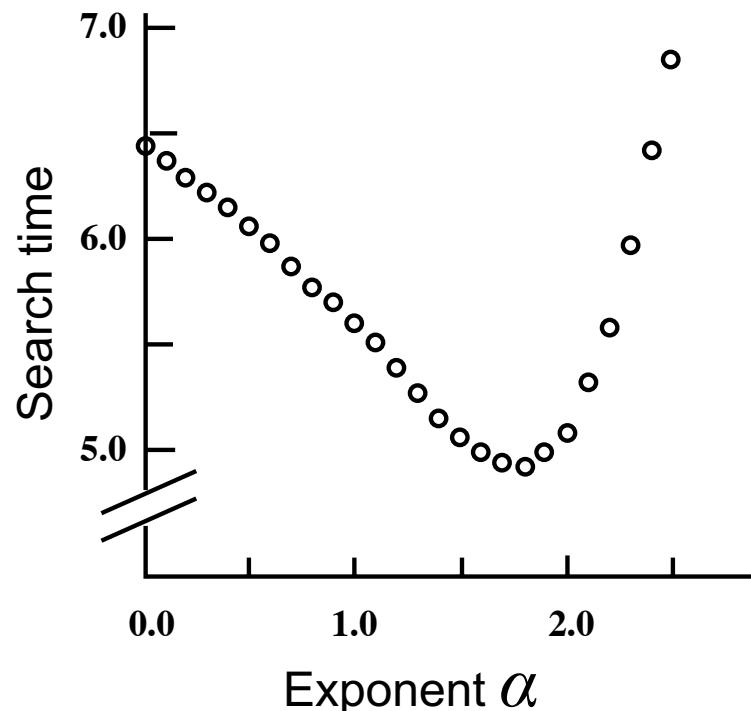
### Erdős–Rényi

$$O(n)$$

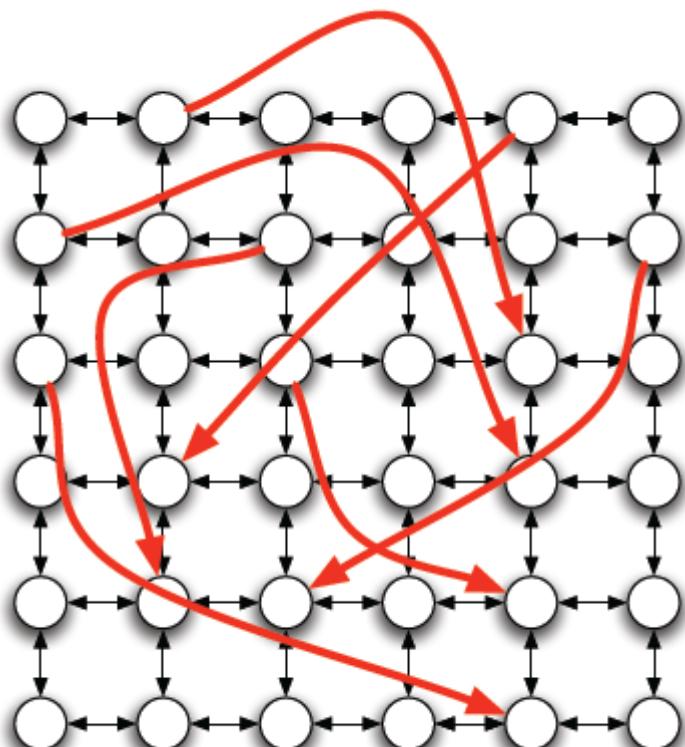
# Kleinberg's Model: Search Time

## ■ We know:

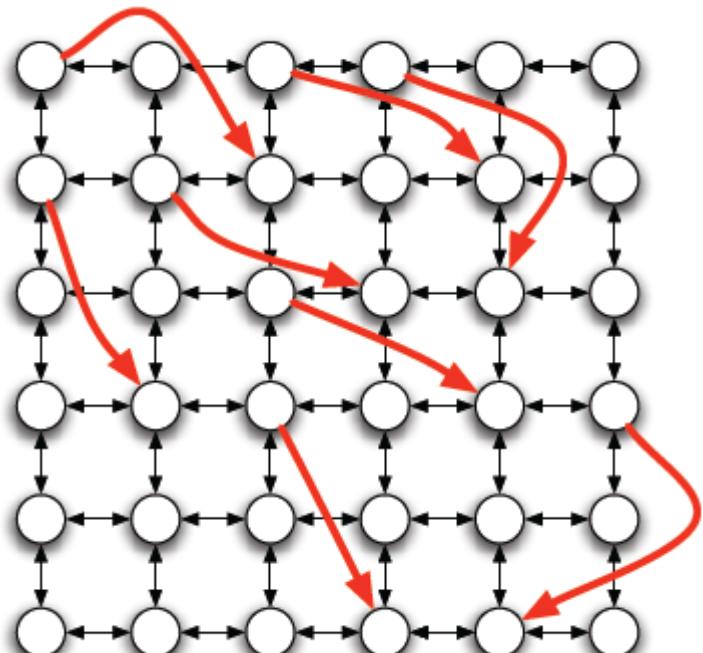
- $\alpha = 0$  (i.e., Watts-Strogatz): We need  $O(\sqrt{n})$  steps
- $\alpha = 1$ : We need  $O(\log(n)^2)$  steps



# Intuition: Why Search Takes Long



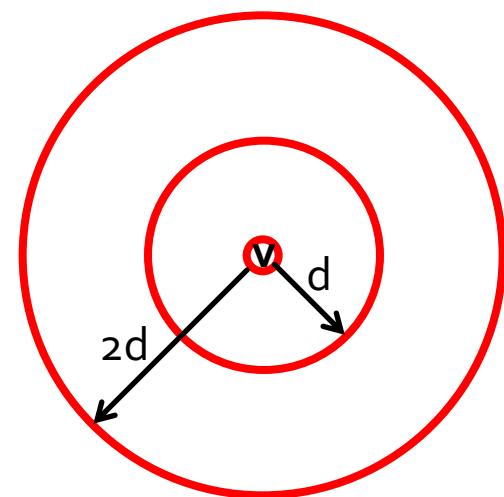
Small  $\alpha$ : too many long links



## Big $\alpha$ : too many short links

# Why Does It Work?

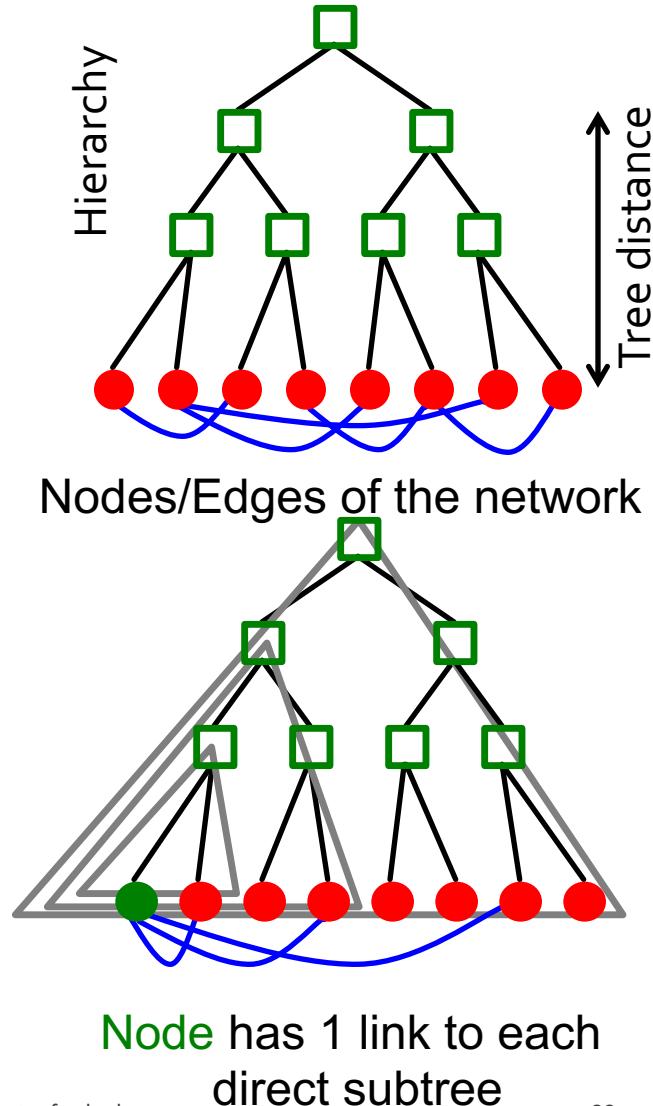
- How does the argument change for 2-d grid:
  - $P(u \text{ points to } I) > 1/Z \cdot \#nodes(I) \cdot P(u \rightarrow v)$   
 $\ln n$                $d^2$                $d^{-2} \Rightarrow \alpha = 2$
- Why  $P(u \rightarrow v) \sim d(u, v)^{-\dim}$  works?
  - Approx uniform over all “scales of resolution”
  - # nodes at distance  $d$  grows as  $d^{\dim}$ , prob.  $d^{-\dim}$  of each edge  
→ const. prob. of a link, independent of  $d$



Number of nodes is  $\propto d^2$   
Prob. of linking each is  $\propto d^{-2}$

# Different Model: Hierarchies

- Nodes are in the leaves of a tree:
  - Departments, topics, ...
- Create  $k$  edges out of every node  $v$ 
  - Create each edge out of  $v$  by choosing  $v \rightarrow w$  with prob.  $\sim b^{-h(v,w)}$ 
    - $h(u,v)$  = tree-distance (height of the least common ancestor)
- Start at  $s$ , want to go to  $t$ 
  - Only see out links of the current node
  - But you know the hierarchy
- Claim 1:
  - For any direct subtree  $T'$  one of  $v$ 's links points to  $T'$
- Claim 2:
  - Claim 1 guarantees efficient search



# Different Model: Hierarchies

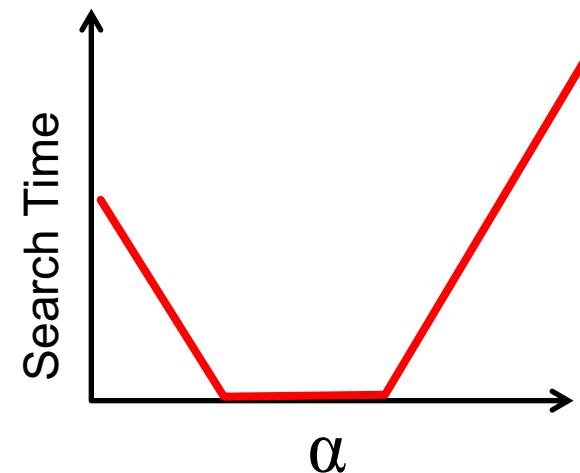
## ■ Extension:

- Multiple hierarchies – geography, profession, ...
- Generate separate random graph in each hierarchy
- Superimpose the graphs
- Search algorithm:
  - Choose a link that gets closest **in any hierarchy**

## ■ Q: How to analyze the model?

### ■ Simulations:

- Search works for a range of alphas
- Biggest range of searchable alphas for 2 or 3 hierarchies
- Too many hierarchies hurts

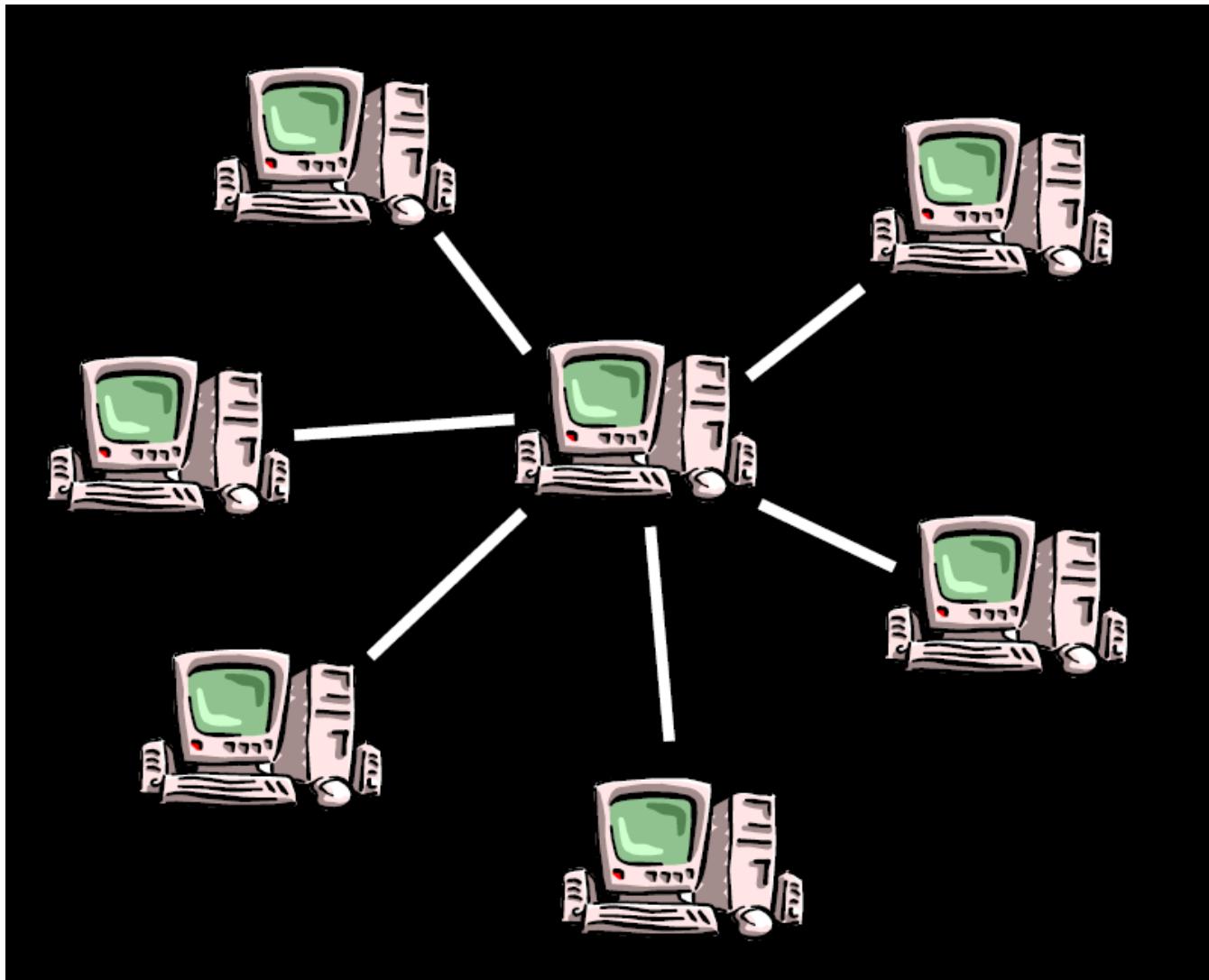


# Search in P2P Networks

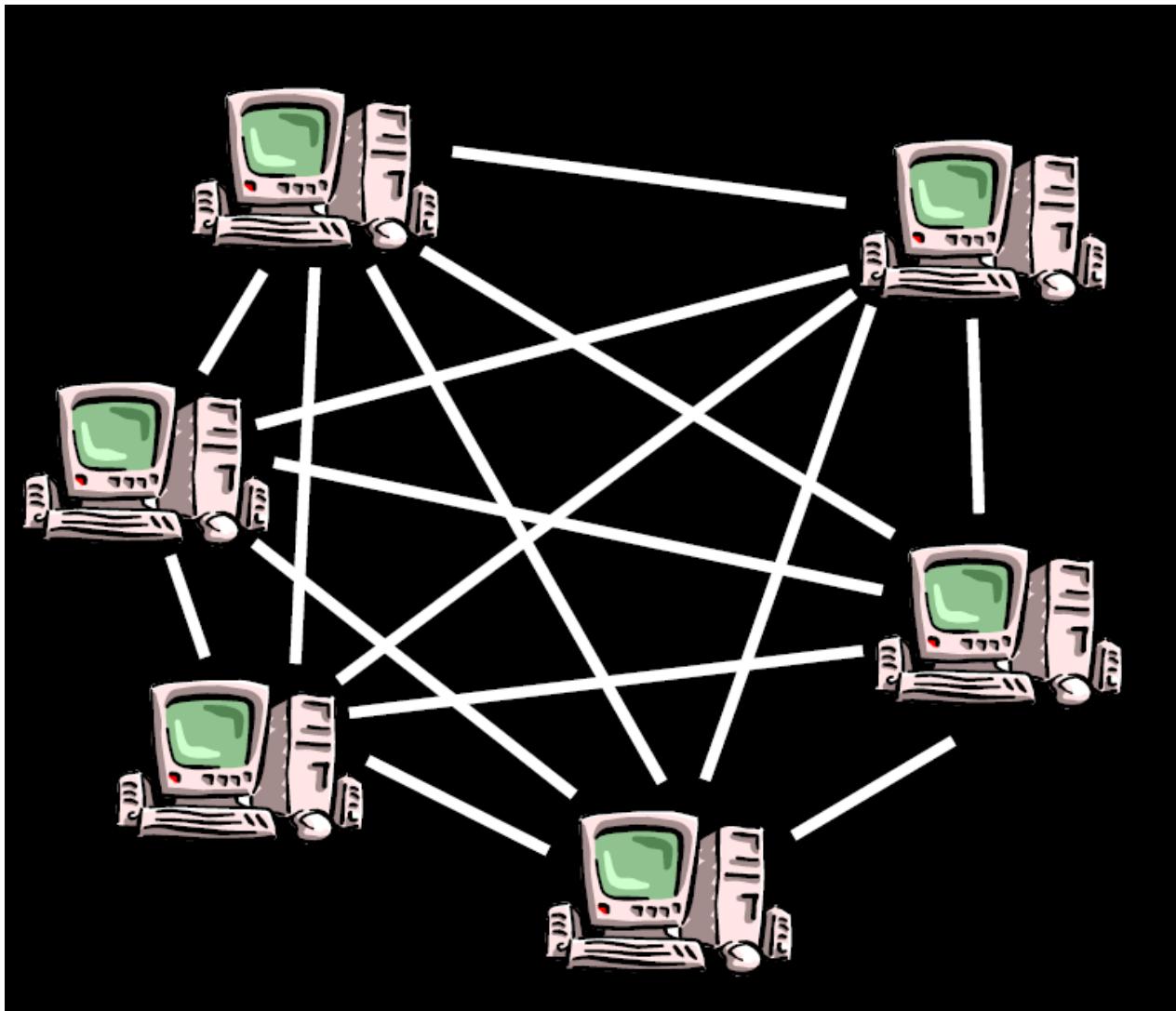
# Algorithmic consequence of small-world:

How to find files in  
Peer-to-Peer networks?

# Client – Server



# P2P: Only Clients



# Napster



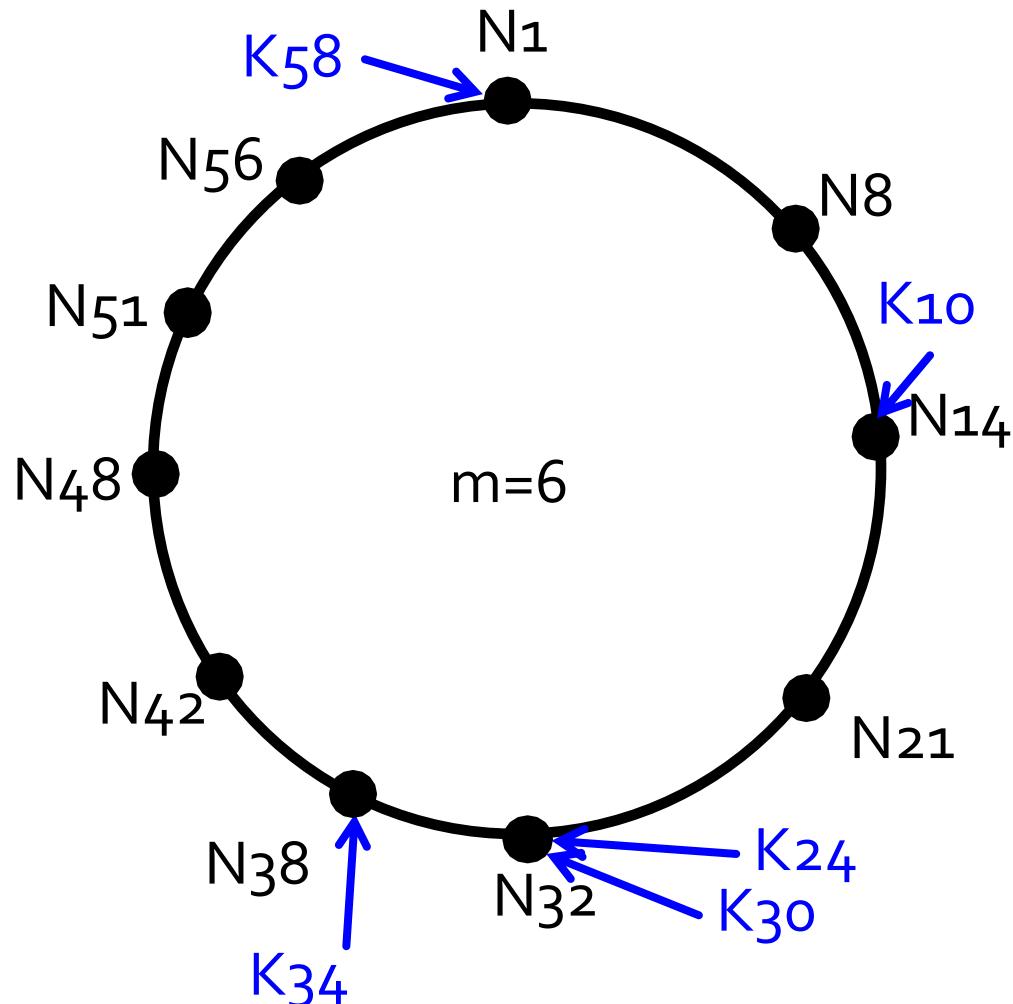
- Napster existed from June '99 and July '01
- Hybrid between P2P and a centralized network
- Once lawyers got the central server to shut down, the network fell apart

# P2P Protocol Chord

- **Protocol Chord maps key (filename) to a node:**
  - Keys are files we are searching for
  - Computer that keeps the **key** can then point to the true location of the file
- **Keys and nodes have  $m$ -bit IDs assigned to them:**
  - Node ID is a hash-code of the IP address (32-bit)
  - Key ID is a hash-code of the file

# Example: Chord on a Cycle

- Cycle with node ids 0 to  $2^{m-1}$
- File (key)  $k$  is assigned to a node  $a(k)$  with ID  $\geq k$

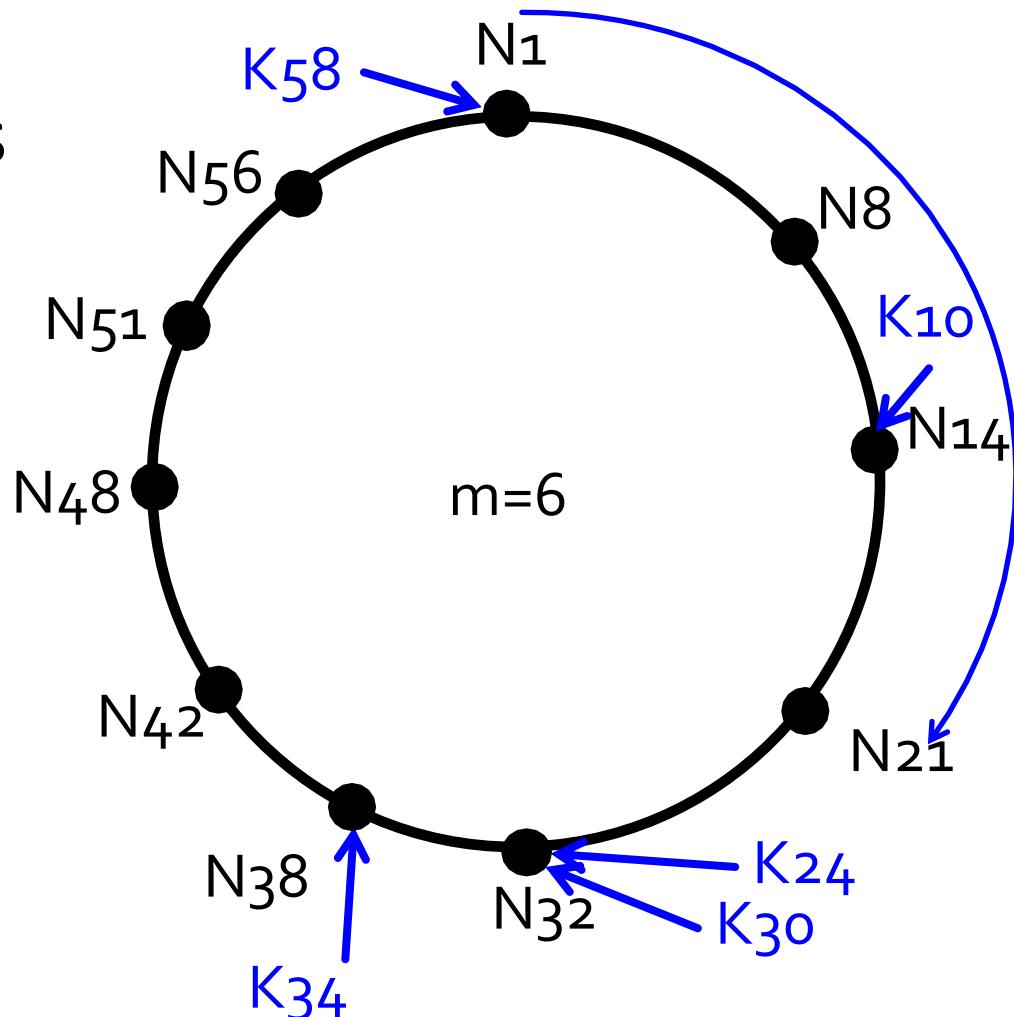


# Chord: Basics

- Assume we have  $N$  nodes and  $K$  keys (files)
- **How many keys does each node have?**
- When a node joins/leaves the system it only needs to talk to its immediate neighbors
  - When node  $N+1$  joins or leaves, then only  $O(K/N)$  keys need to be rearranged
- Each node knows the IP address of its immediate neighbors

# Searching the Network

- If every node knows its immediate neighbor then use sequential search
- Search time is  $O(N)!$



# Faster Search

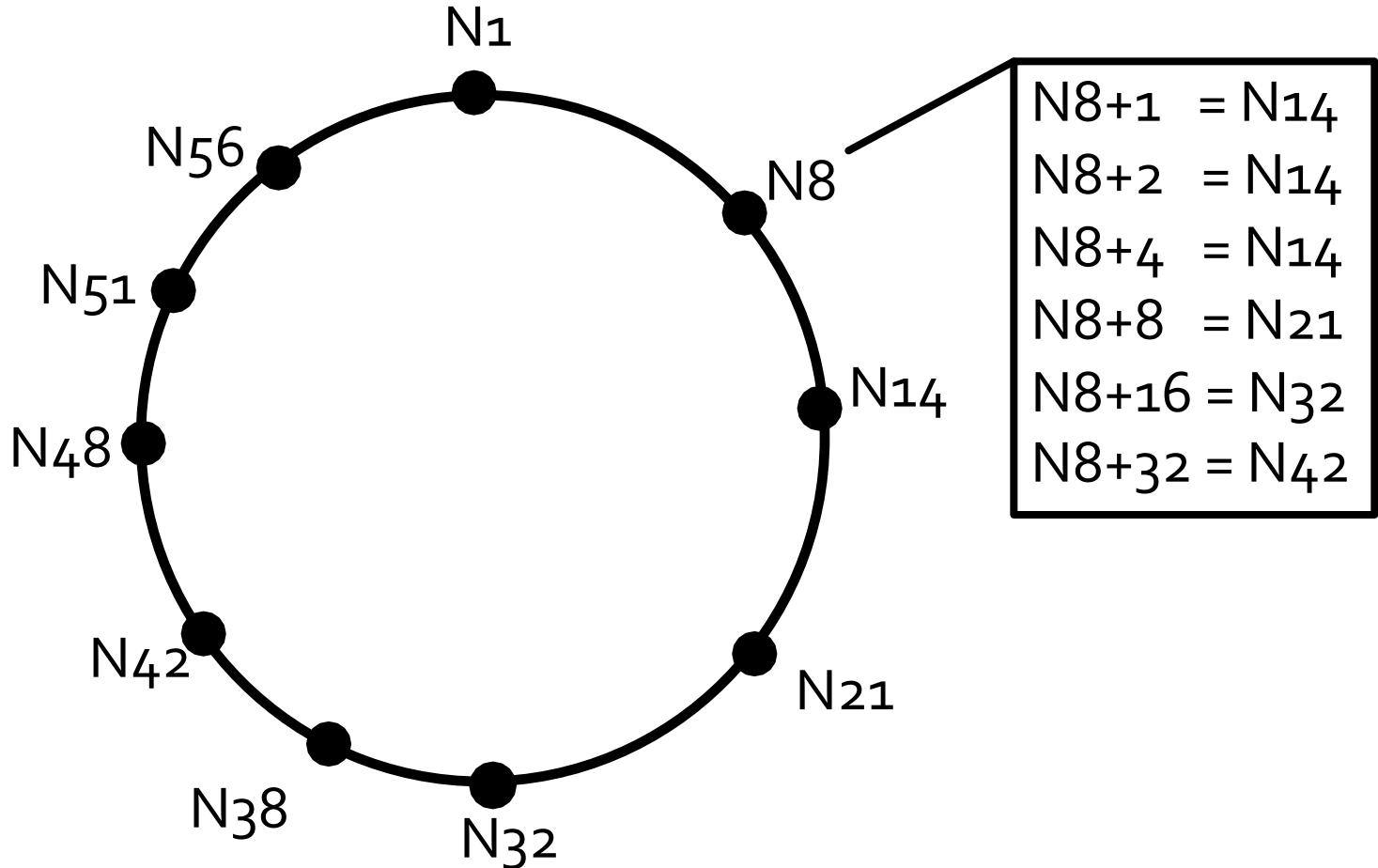
## Faster Search:

- A node maintains a table of  $m=\log(N)$  entries
- $i$ -th entry of a node  $n$  contains the address of  $(2^i)$ -th neighbor
  - $i$ -th entry points to first node with ID  $\geq n+2^i$
  - **Problem:** When a node joins we violate long range pointers of all other nodes
    - Many papers about how to make this work

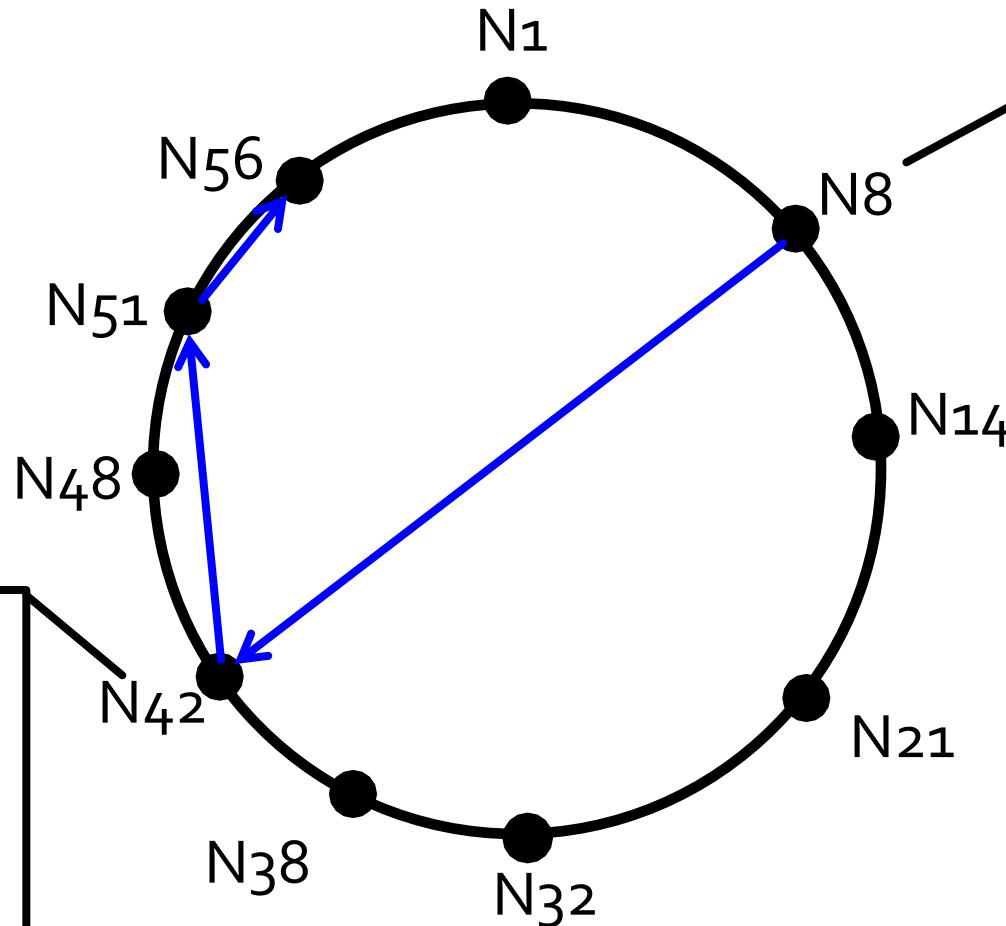
## Search algorithm:

- Take the longest link that does not overshoot
  - With each step we **halve** the distance to the target!

i-th entry of N has the address of  
 $(N+2^i)$ -th node



# Start at N8, find key with ID 54



N<sub>8+1</sub> = N<sub>14</sub>  
N<sub>8+2</sub> = N<sub>14</sub>  
N<sub>8+4</sub> = N<sub>14</sub>  
N<sub>8+8</sub> = N<sub>21</sub>  
N<sub>8+16</sub> = N<sub>32</sub>  
N<sub>8+32</sub> = N<sub>42</sub>

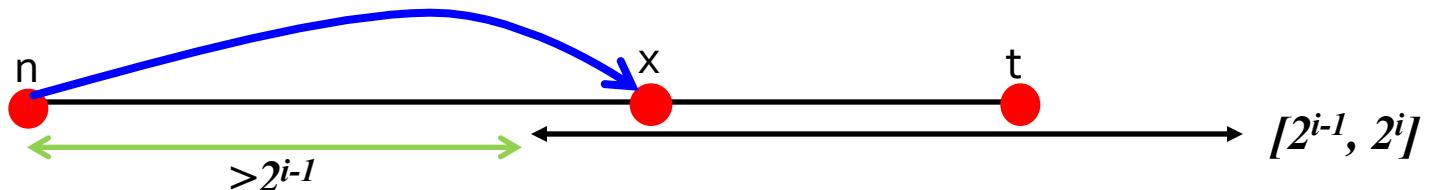
N<sub>42+1</sub> = N<sub>48</sub>  
N<sub>42+2</sub> = N<sub>48</sub>  
N<sub>42+4</sub> = N<sub>48</sub>  
N<sub>42+8</sub> = N<sub>51</sub>  
N<sub>42+16</sub> = N<sub>1</sub>  
N<sub>42+32</sub> = N<sub>14</sub>

# How Long Does It Take to Find a Key?

- Claim: Search for any key in the network of  $N$  nodes visits  $\mathbf{O}(\log N)$  nodes
- Assume that node  $n$  queries for key  $k$
- Let the key  $k$  reside at node  $t$
- **How many steps do we need to reach  $t$ ?**

# $O(\log N)$ steps. Proof:

- We start the search at node  $n$
- Let  $i$  be a number such that  $t$  is contained in interval  $[n+2^{i-1}, n+2^i]$  (for some  $i$ )
- Then the table at node  $n$  contains a pointer to node  $x$  that is the first node past node id  $n+2^{i-1}$
- **Claim:** Node  $x$  is closer to  $t$  than  $n$



- So, in one step we **halved** the distance to  $t$
- We can do this at most  $\log_2 N$  times
- Thus, we find  $t$  in  $O(\log_2 N)$  steps

# **Empirical Studies of Navigation in Small-World Networks**

# Small-World in HP Labs

- **Adamic-Adar 2005:**

- HP Labs email logs (436 people)
- Link if  $u, v$  exchanged >5 emails each way

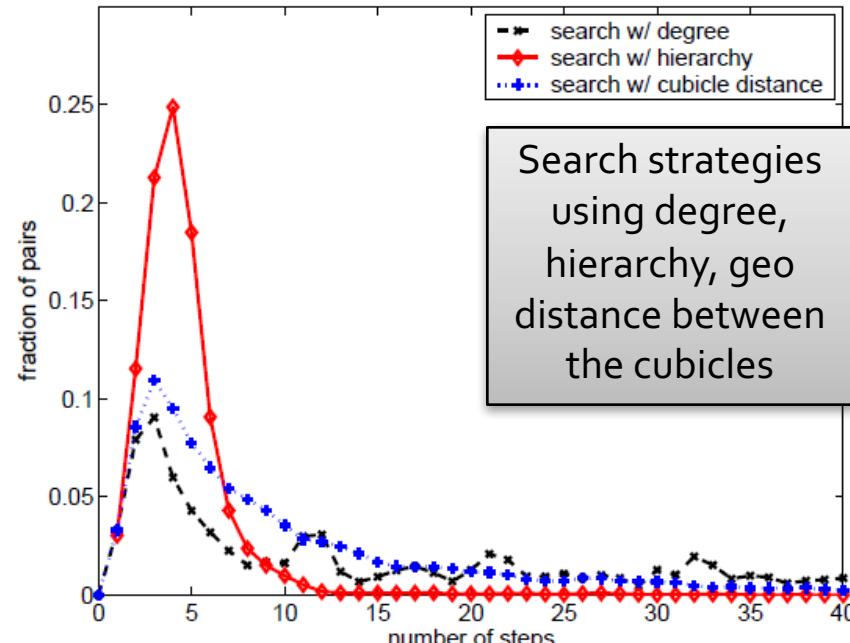
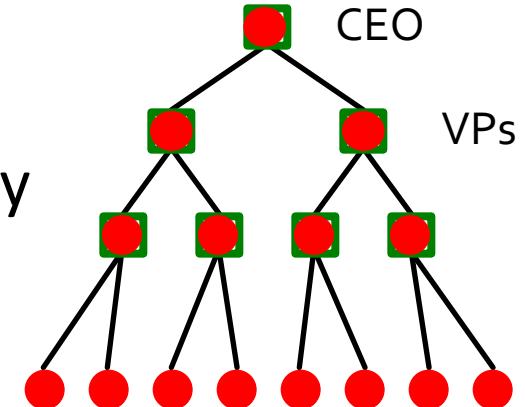
- **Map of the organization hierarchy**

- How many edges cross groups?
- Finding:

$P(u \rightarrow v) \sim 1 / (\text{size of the smallest group containing } u, v)$

- **Differences from the hierarchical model:**

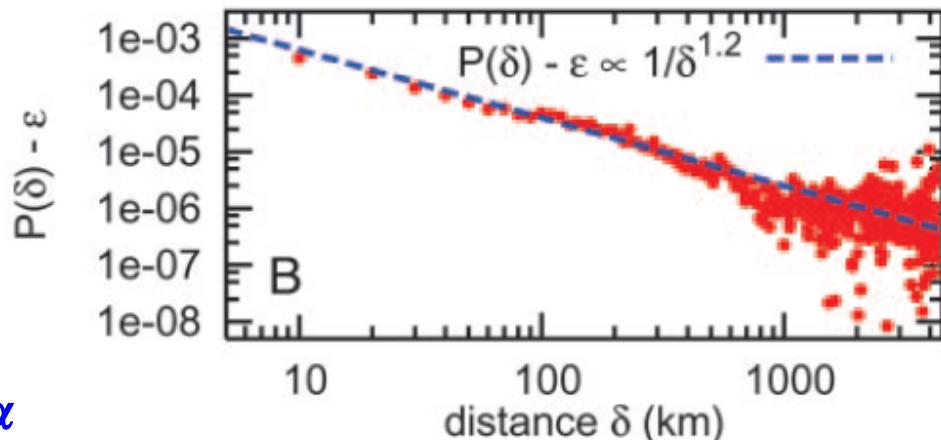
- Weighted edges
- People on non-leaf nodes
- Not  $b$ -ary or uniform depth



# Small-World in LiveJournal

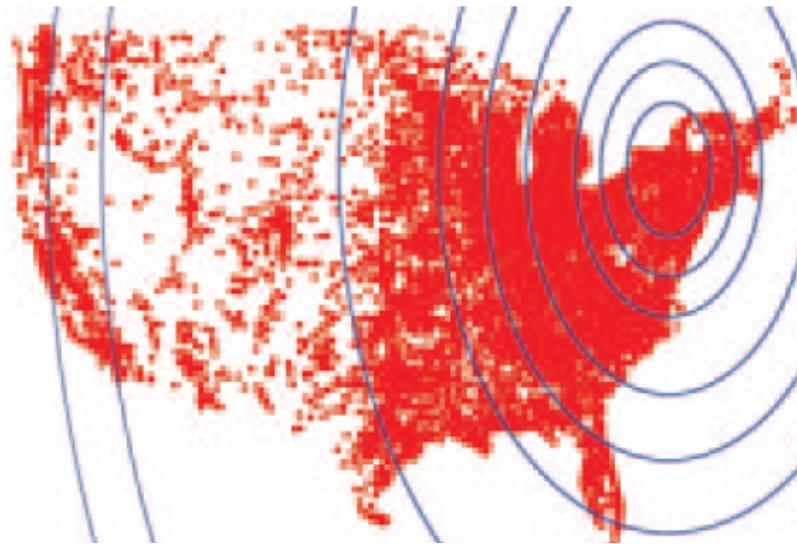
## Liben-Nowell et al. '05:

- LiveJournal data
  - Bloggers + zip codes
- Link prob.:  $P(u,v) = \delta^{-\alpha}$
- $\alpha = ?$
- Problem:
  - Non-uniform population density
- Solution: Rank based friendship



Link length in a network of bloggers  
(0.5 million bloggers, 4 million links)

# Improved Model

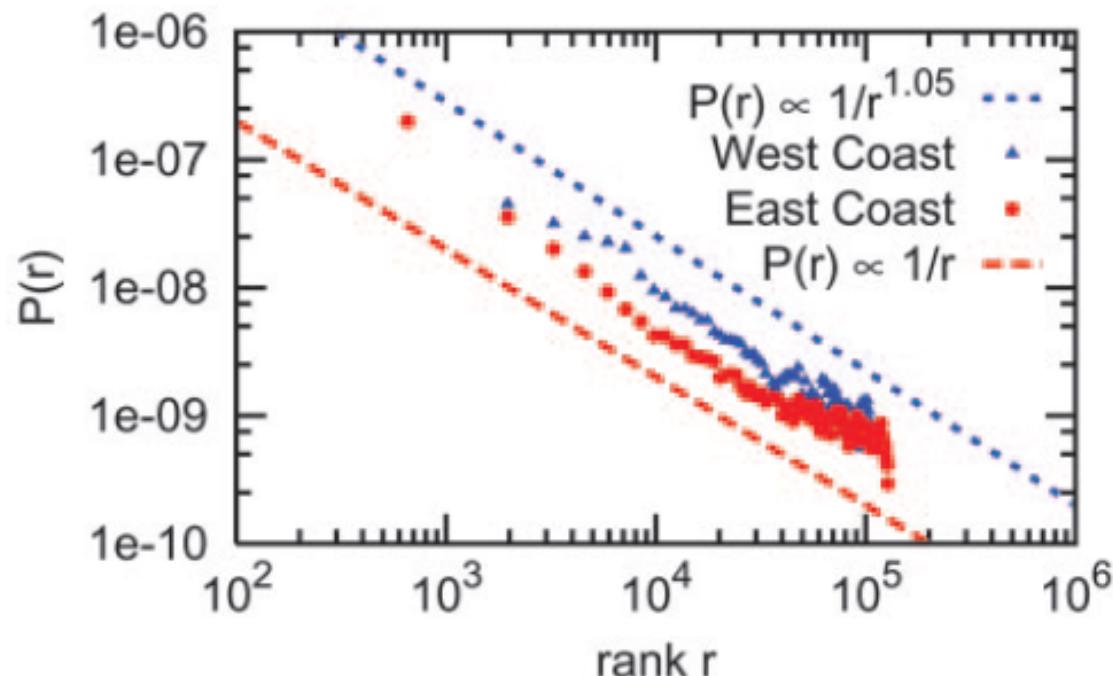


$$\text{rank}_u(v) := |\{w : d(u, w) < d(u, v)\}|$$

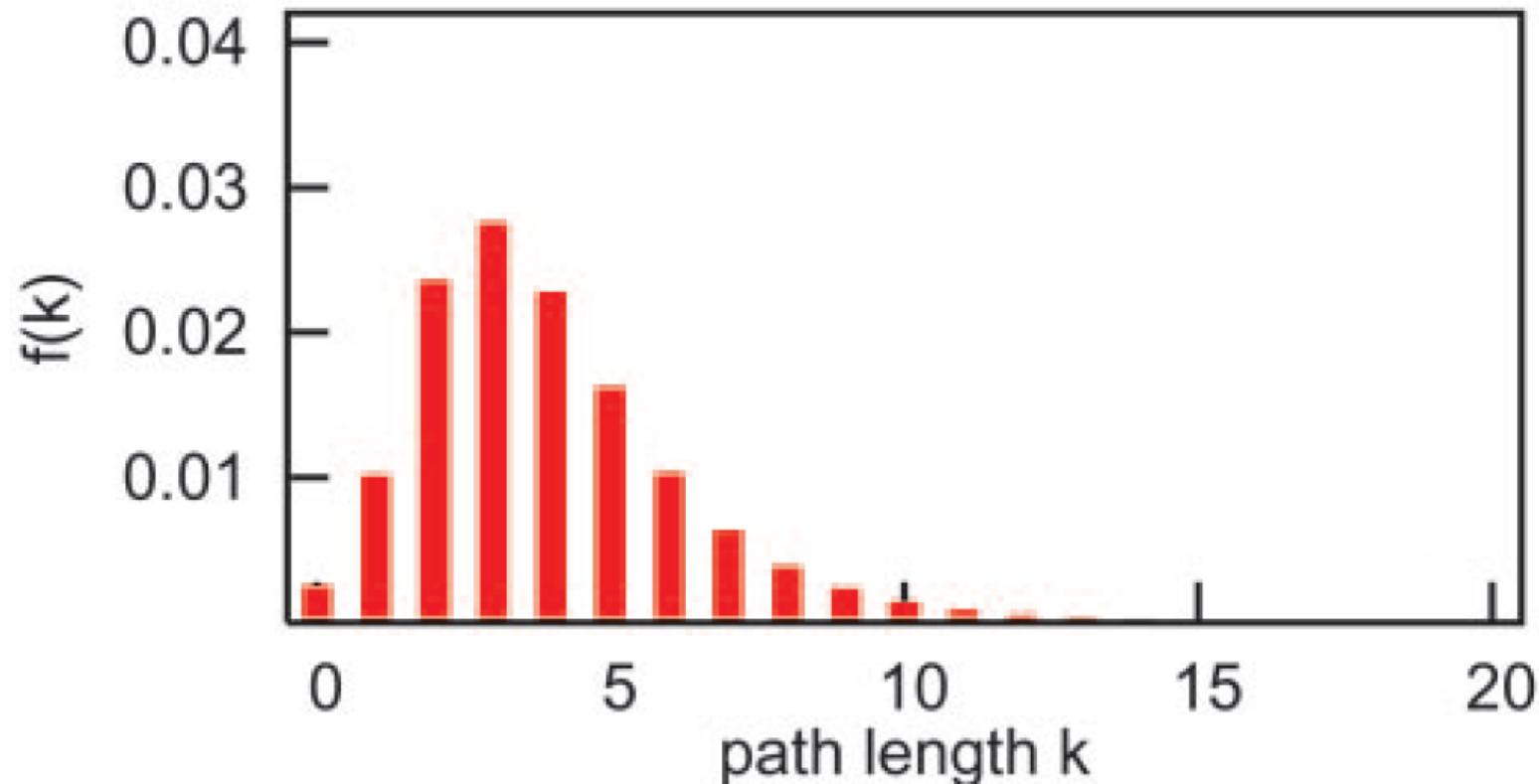
- $P(u \rightarrow v) = \text{rank}_u(v)^{-\alpha}$
- **What is best  $\alpha$ ?**
  - For equally spaced pairs:  $\alpha = \text{dim. of the space}$
  - In this special case  $\alpha = 1$  is best for search

# Rank Based Friendships

- Close to theoretical optimum of  $\alpha = -1$



# Geographic Navigation



- **Decentralized search in a LiveJournal network**
  - 12% searches finish, average 4.12 hops

# Q: Why do searchable networks arise?

- **Why is rank exponent close to -1?**
  - Why in any network? Why online?
  - How robust/reproducible?
- Mechanisms that get  $\alpha = 1$  purely through local “rearrangements” of links
- **Conjecture** [Sandbeng-Clark]
  - Nodes on a ring with random edges
  - Process of morphing links:
    - **Update step:** Randomly choose  $s, t$ , run decentr. search alg.
    - **Path compression:** each node on path updates long range link to go directly to  $t$  with some small prob.
  - **Conjecture from simulation:**  $P(u \rightarrow v) \sim dist^{-1}$

# How the Class Fits Together

## Observations

Small diameter,  
Edge clustering

Patterns of signed  
edge creation

Viral Marketing, Blogosphere,  
Memetracking

Scale-Free

Densification power law,  
Shrinking diameters

Strength of weak ties,  
Core-periphery

## Models

Erdős-Renyi model,  
Small-world model

Structural balance,  
Theory of status

Independent cascade model,  
Game theoretic model

Preferential attachment,  
Copying model

Microscopic model of  
evolving networks

Kronecker Graphs

## Algorithms

Decentralized search

Models for predicting  
edge signs

Influence maximization,  
Outbreak detection, LIM

PageRank, Hubs and  
authorities

Link prediction,  
Supervised random walks

Community detection:  
Girvan-Newman, Modularity