Small-World Phenomena and Decentralized Search

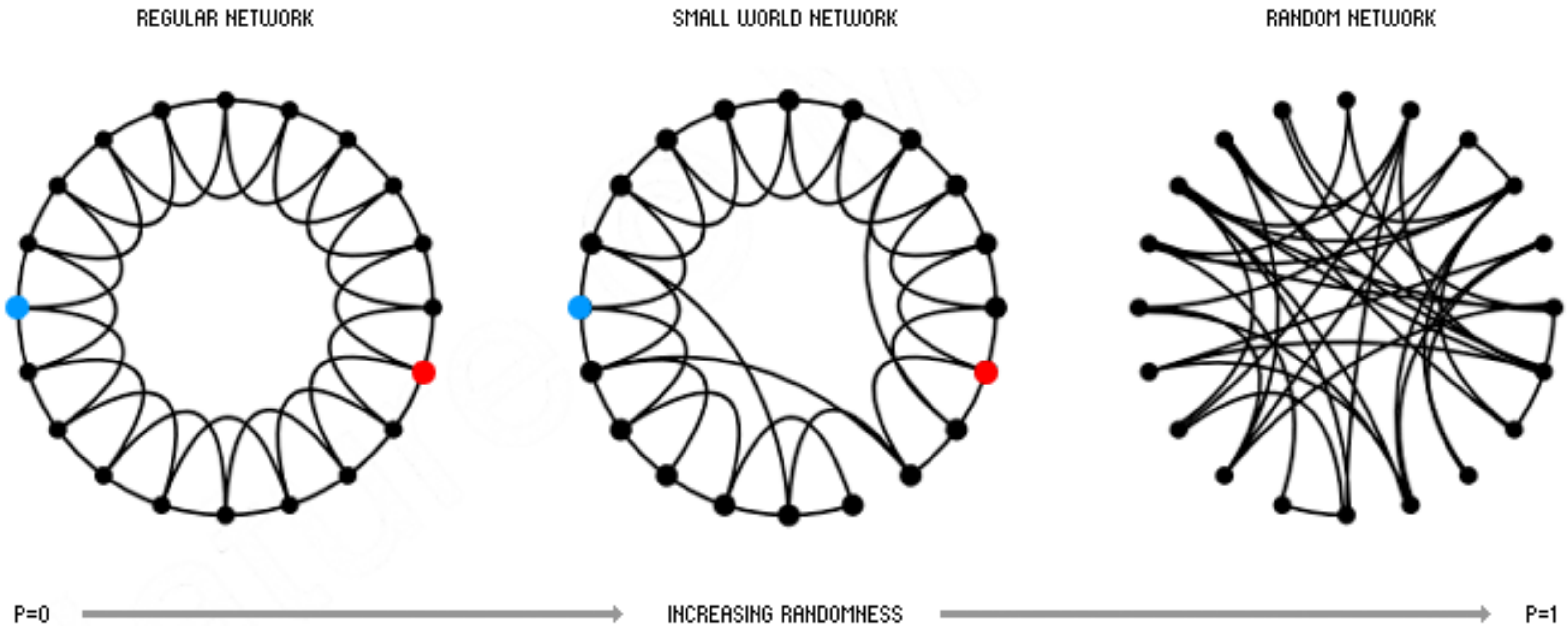
CS224W: Social and Information Network Analysis

Jure Leskovec, Stanford University

http://cs224w.stanford.edu

Recap: Small-World - How?

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

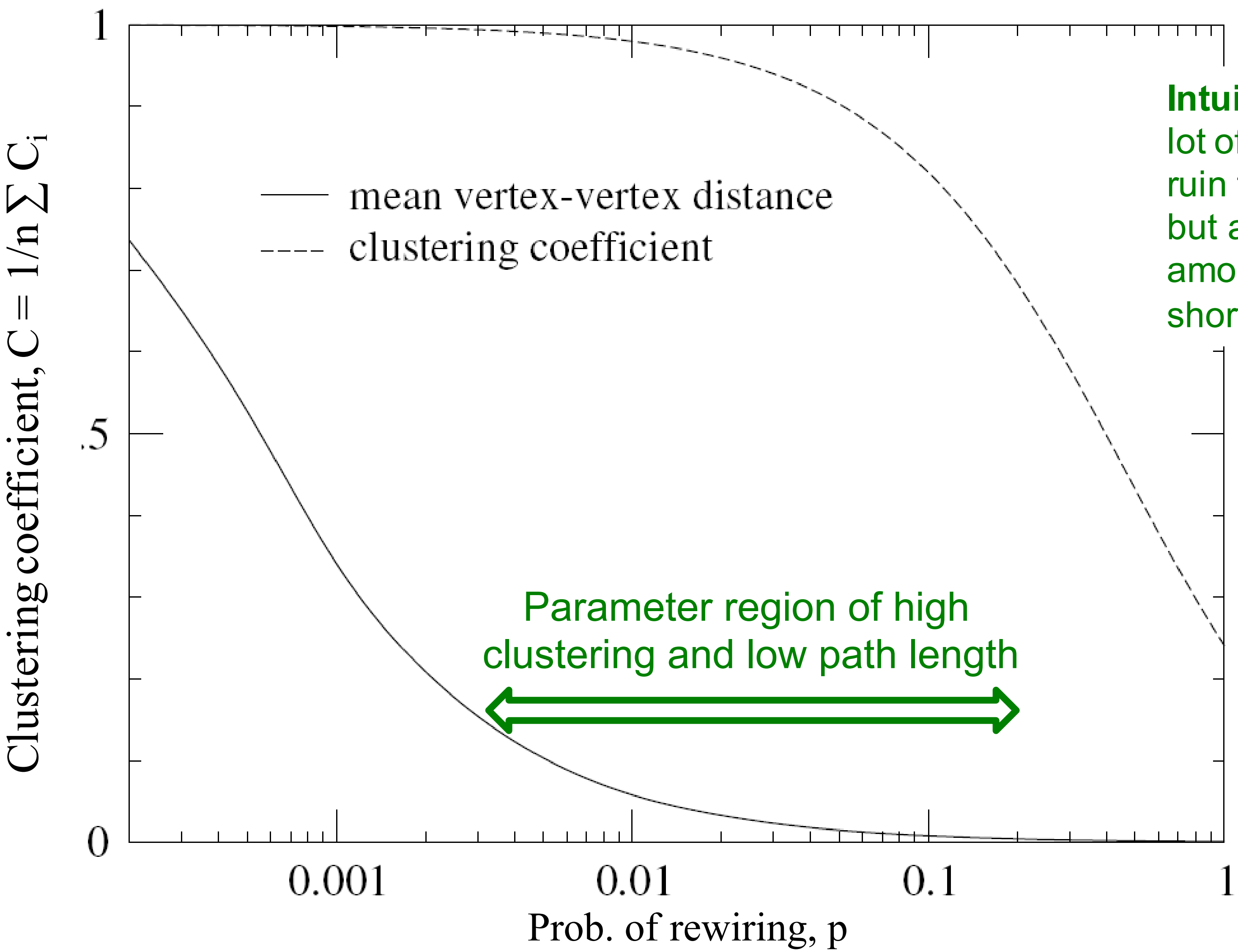


High clustering High diameter	High clustering Low diameter	Low clustering Low diameter

$$h = \frac{N}{2\bar{k}} \qquad C = \frac{3}{4}$$

$$h = \frac{\log N}{\log \alpha} \qquad C = \frac{\bar{k}}{N}$$

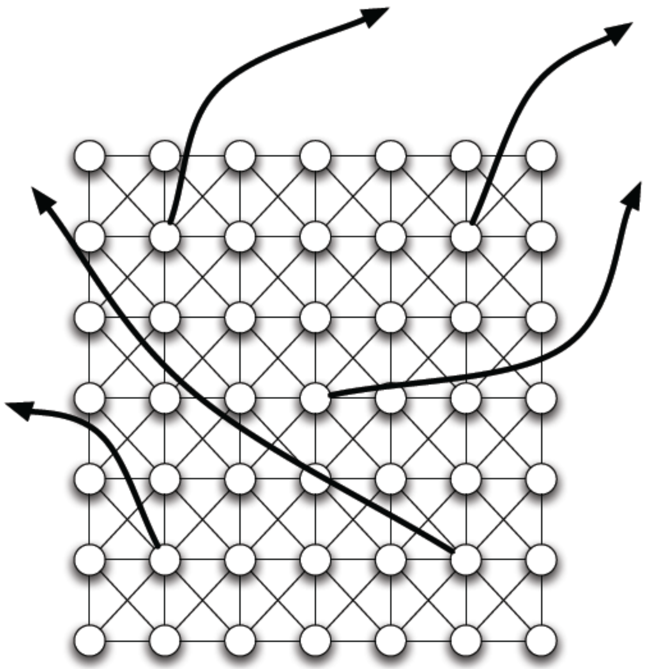
The Small-World Model



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?

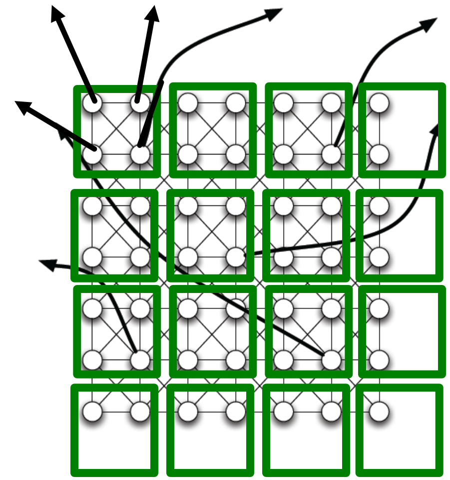
It is *O(log(n))*

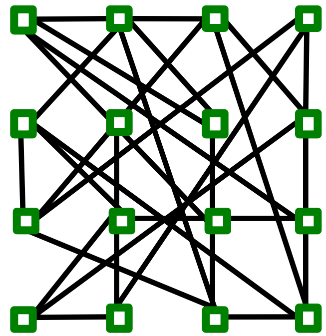
Why?

Diameter of the Watts-Strogatz

- **Proof:**
  - Consider a graph where we contract 2x2 subgraphs into supernodes
  - Now we have 4 random edges sticking out of each supernode
    - **4-regular random graph!**
  - From Thm. we have short paths between super nodes (due to 4 random edges)
  - We can turn this into a path in a real graph by adding at most 2 steps per long range edge (by having to traverse internal nodes)

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





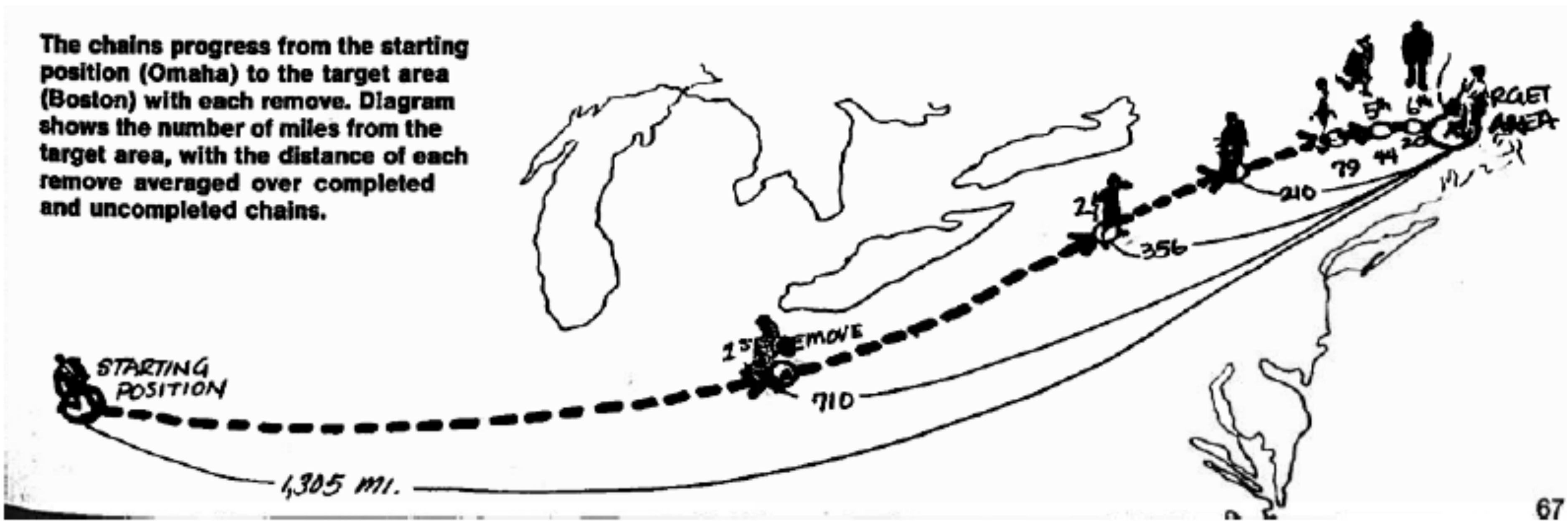
4-regular random graph

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?**

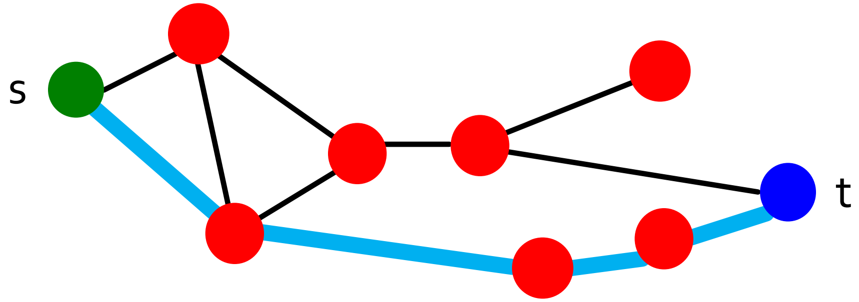


How would you go about finding the path?

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

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

Not searchable

Search time T:

$$O(n^{\alpha})$$

Watts-Strogatz

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

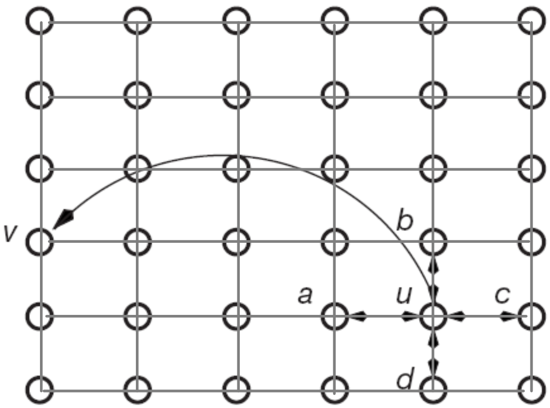
Erdős–Rényi

$$O(n)$$

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*).

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))**

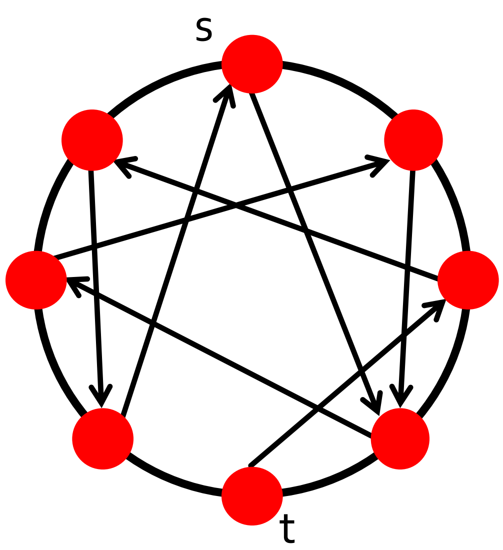


- **Fact:** A decentralized search algorithm in Watts-Strogatz model needs $n^{2/3}$ steps to reach t in expectation
  - **Note:** Even though paths of $O(\log n)$ steps exist

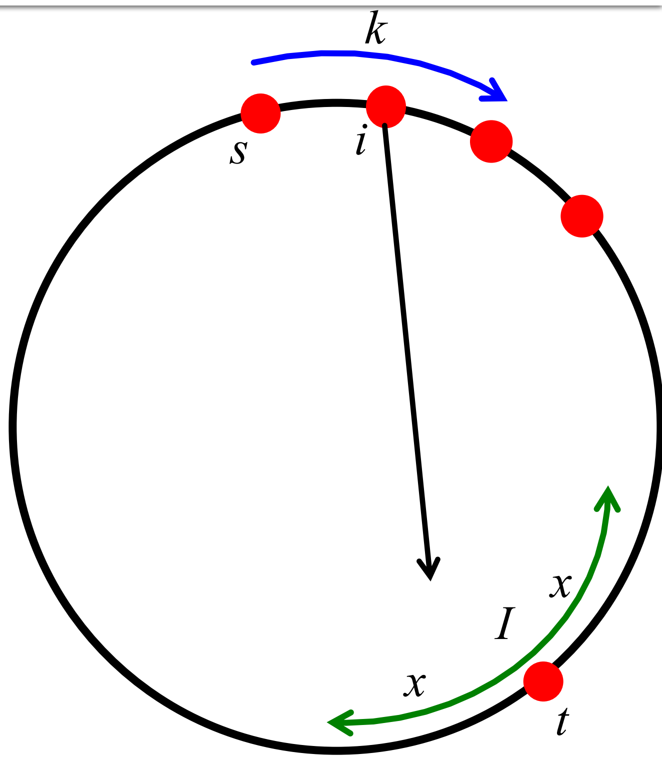
- **Note:** All our calculations are asymptotic, i.e., we are interested in what happens as $n \to \infty$

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 ≥ O($n^{1/2}$)



<u>**Overview of the proof:**</u>

- **Reason about event $\boldsymbol{E}$**
  - $\boldsymbol{E}$ = event that any of the first $\boldsymbol{k}$ nodes search algorithm visits has a link to $\boldsymbol{I}$ of width $\boldsymbol{2 \cdot x}$ nodes (for some $\boldsymbol{x}$) around target $\boldsymbol{t}$
- **If** $k = x = \frac{1}{2}\sqrt{n}$ **then**

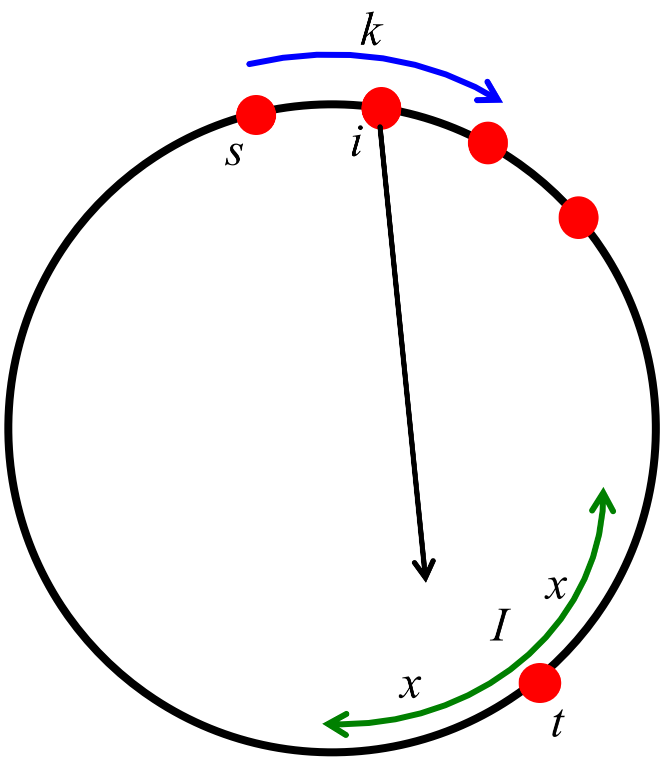
P(E) = P(in $\frac{1}{2}\sqrt{n}$ steps we jump inside $\frac{1}{2}\sqrt{n}$ of t) ≤ $\frac{1}{2}$

- $Search\ time \geq P(E) * \#steps + \underline{P(not\ E) * k}$
- $\geq \frac{1}{2} * k = \frac{1}{2} * \frac{1}{2}\sqrt{n} = O(\sqrt{n})$

(next 4 slides give a detailed proof)

Proof: Search time is ≥ O($n^{1/2}$) Details

- **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)

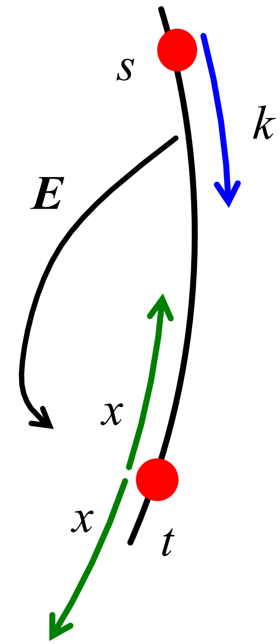
(haven't seen node i yet, but can assume random edge generation)

Proof: Search time is ≥ O($n^{1/2}$) Details

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

Then:

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



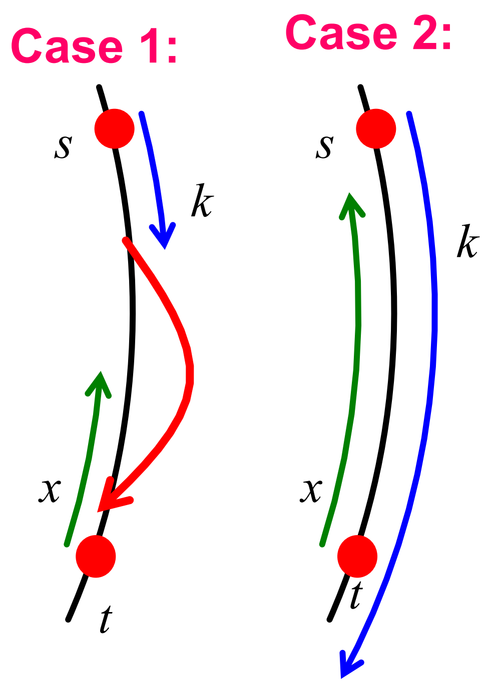
Note: Our alg. is deterministic and will choose to travel via a long- or short-range links using some deterministic rule.

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 ≥ $O(n^{1/2})$ Details

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

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

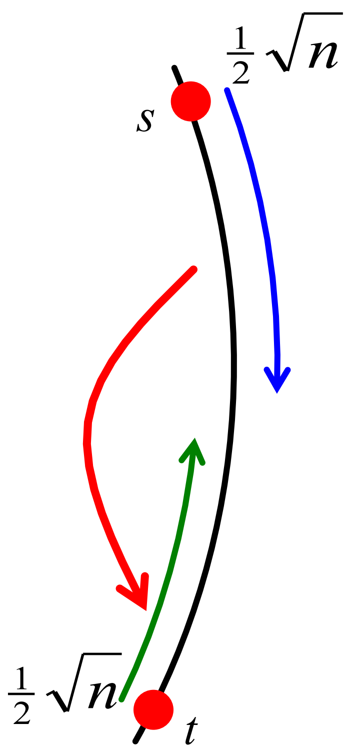


Proof: Search time is ≥ O($n^{1/2}$) Details

- **Claim:** Getting from s to t takes $\geq \frac{1}{4}\sqrt{n}$ steps
- *Search time ≥ P(E)*(#steps) + P(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. ½
    - 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**

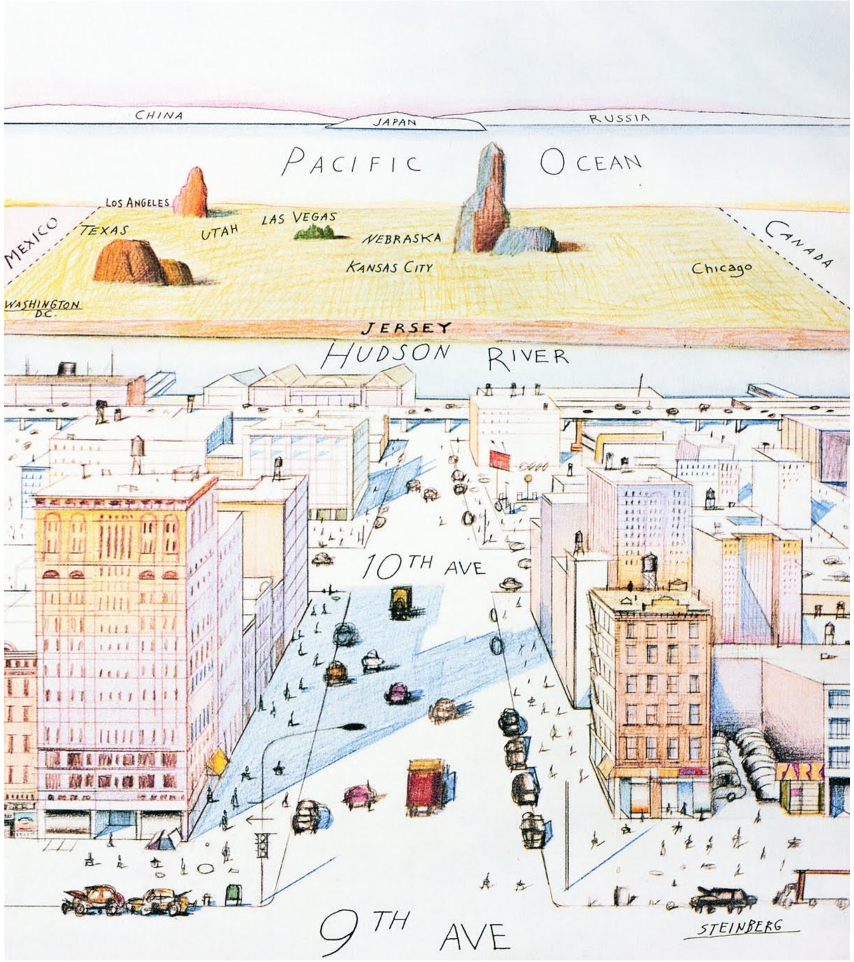
$$\geq P(E\ doesn't\ occur) \cdot \min\{x,k\} =$$

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



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!**



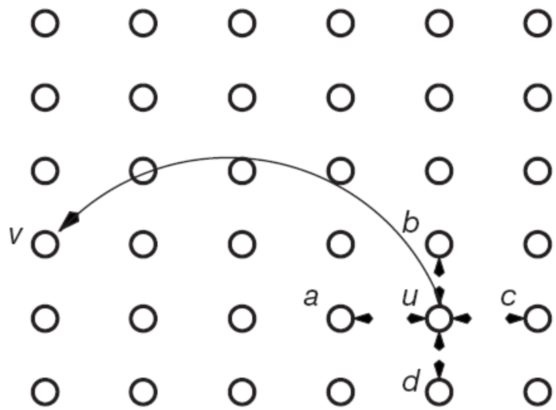
Saul Steinberg, “View of the World from 9th Avenue”

Variation of the Model

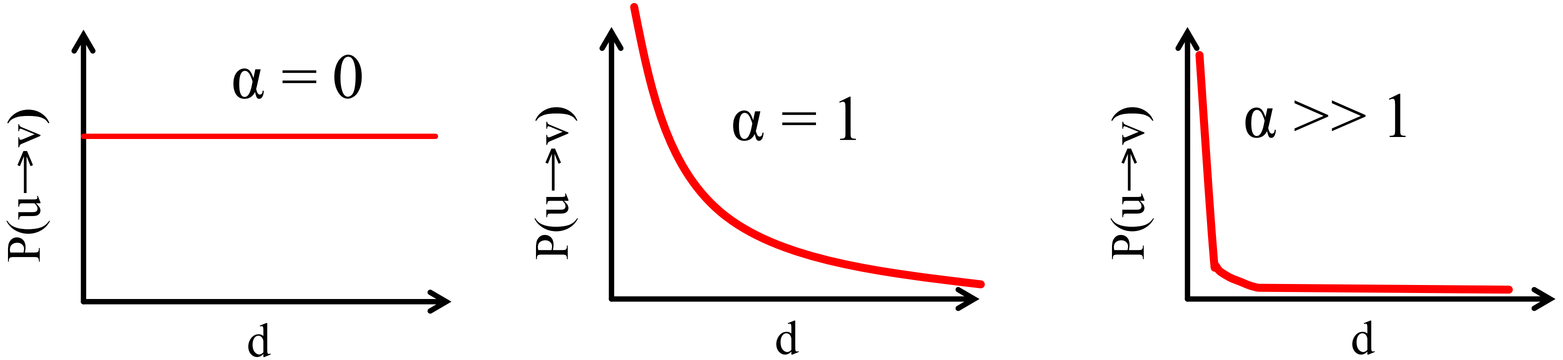
- **Model** [Kleinberg, Nature '01]
  - **Nodes still on a grid**
  - Node has one long range link
  - Prob. of long link to node v:

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

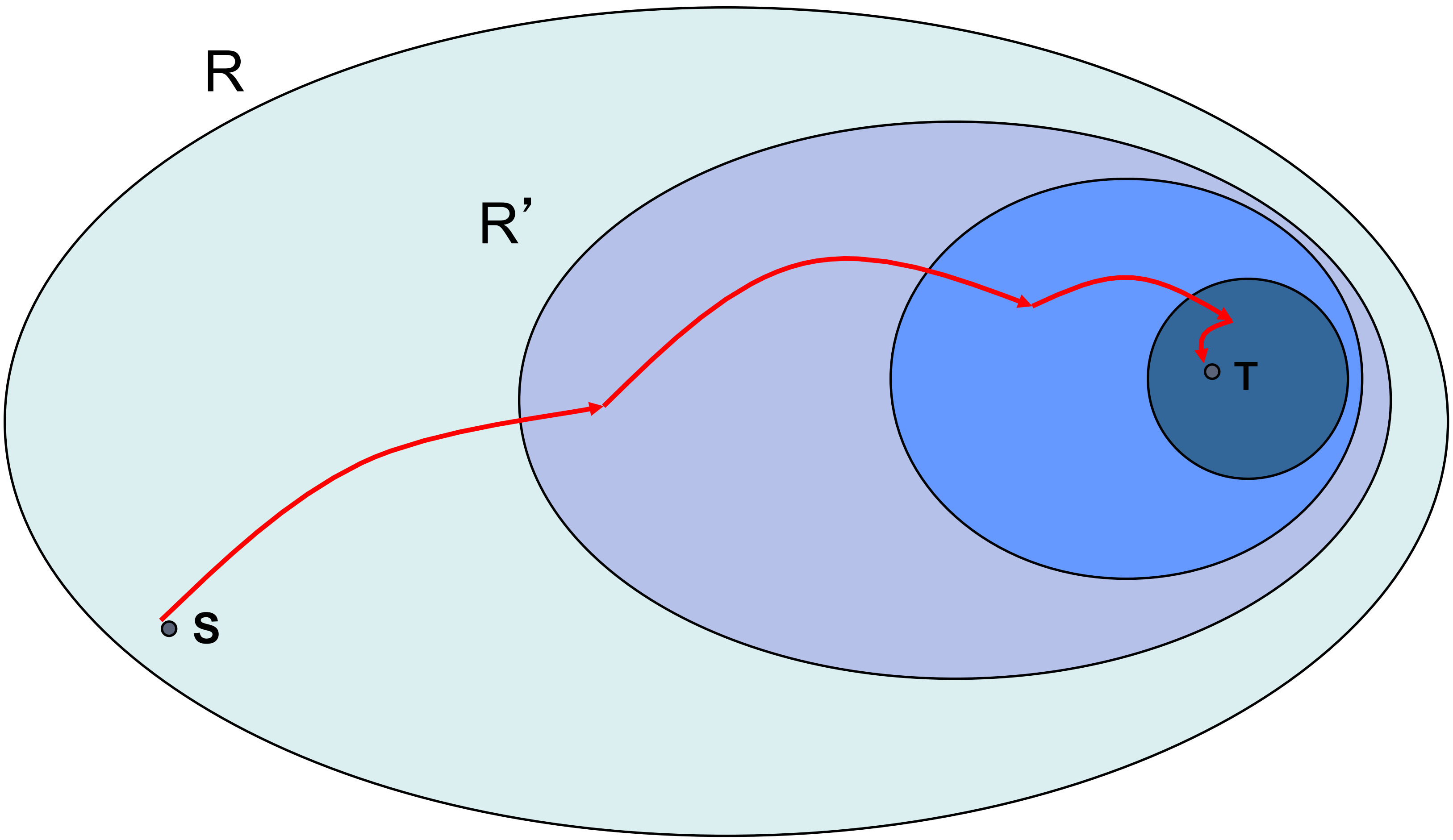
    - $d(u,v)$ ... grid distance between u and v
    - α ... parameter ≥ 0



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



Why Does It Work?

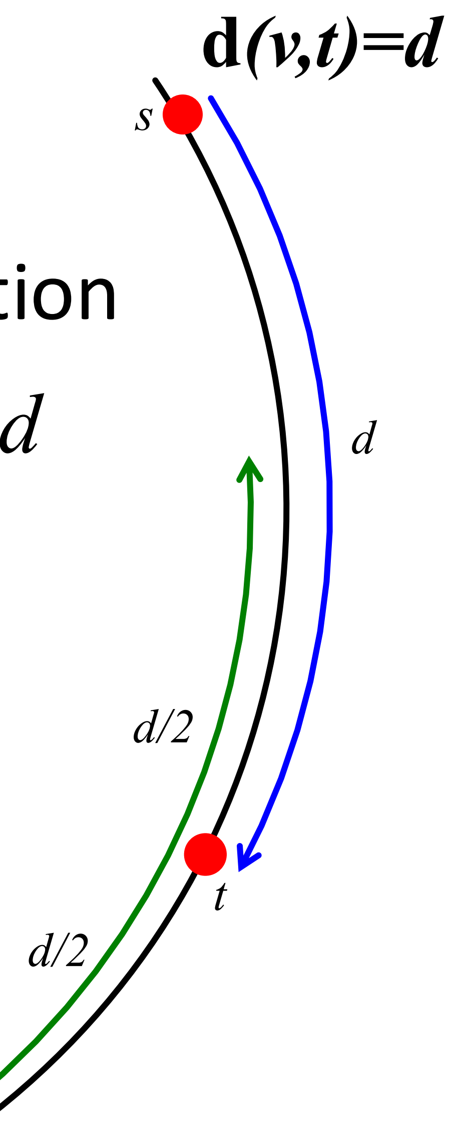


Kleinberg's Model in 1-Dimension

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)

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



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

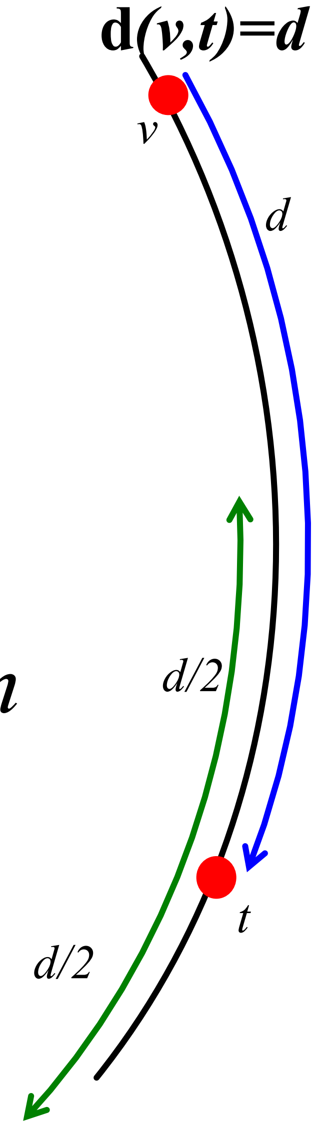
- **First we need: *P(v* points to *w)* =**

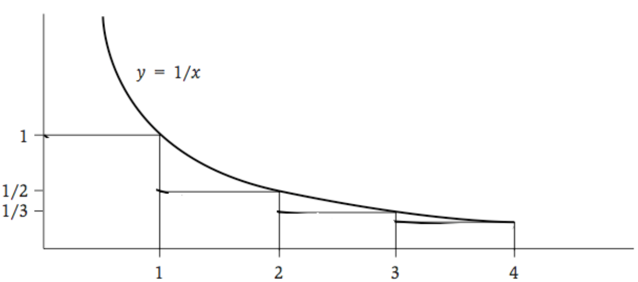
$$P(v \rightarrow w) = \frac{d(v,w)^{-1}}{\sum_{u \neq v} d(v,u)^{-1}}$$

- **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

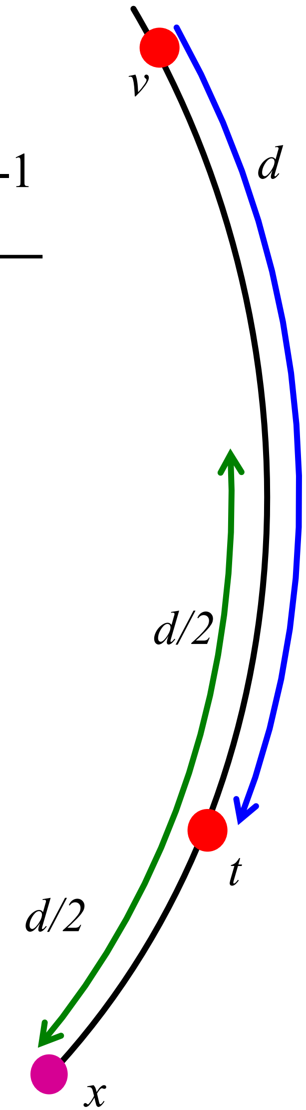
- **Next we need: *P(v points to I)* =**

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

$$= \frac{1}{2 \ln n} \sum_{w \in I} \frac{1}{\underbrace{d(v,w)}_{\substack{\text{What's the smallest of these terms?} \\ \textbf{All terms} \geq \textbf{2/(3d)}}}} \geq \frac{1}{2 \ln n} d \frac{2}{3d} = \frac{1}{3 \ln n}$$

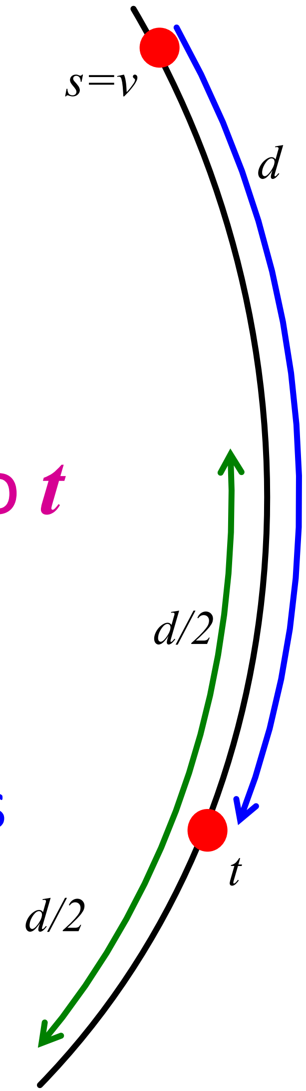
$$= O\left(\frac{1}{\ln n}\right)$$

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



Kleinberg's Model in 1-D

- **So, we have:**
  - I … interval of $d/2$ around t (where $d = \mathrm{d}(v,t)$)
  - $\mathbf{P}$(long link of v points to I) $= \mathbf{1/ln}(n)$
- **In expected # of steps ≤ $\ln(n)$ 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: $\mathbf{O(\log_2(n)^2)}$



Overview of the Results

Searchable

Search time T:

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

Kleinberg's model

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

Not searchable

Search time T:

$$O(n^{\alpha})$$

Watts-Strogatz

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

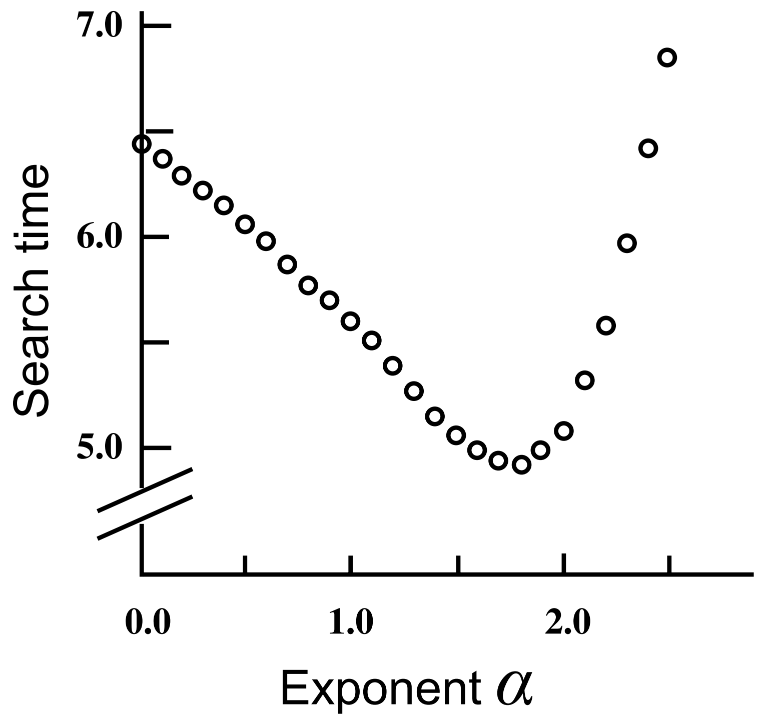
Erdős–Rényi

$$O(n)$$

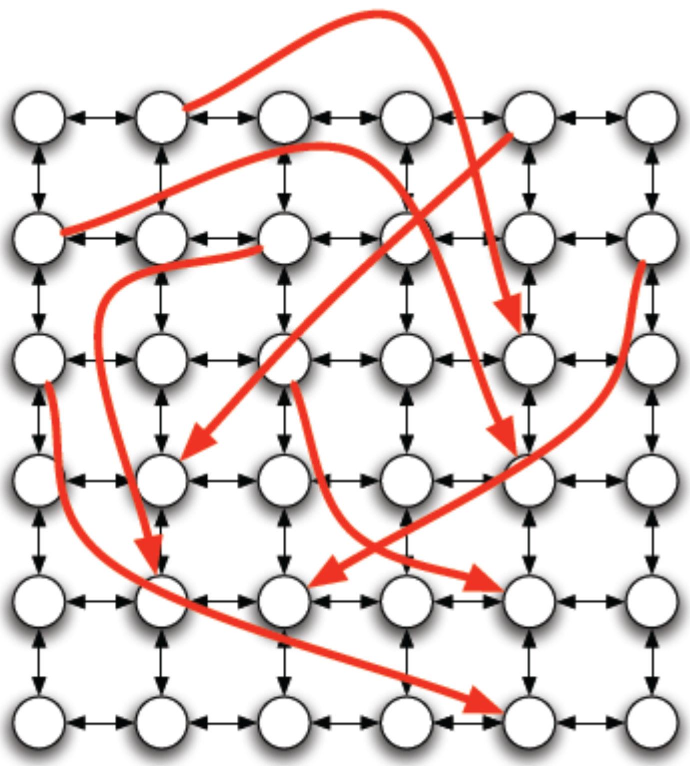
Note: We know these graphs have diameter *O(log n)*.
So in Kleinberg's model search time is polynomial in *log n*,
while in Watts-Strogatz it is exponential (in *log n*).

Kleinberg's Model: Search Time

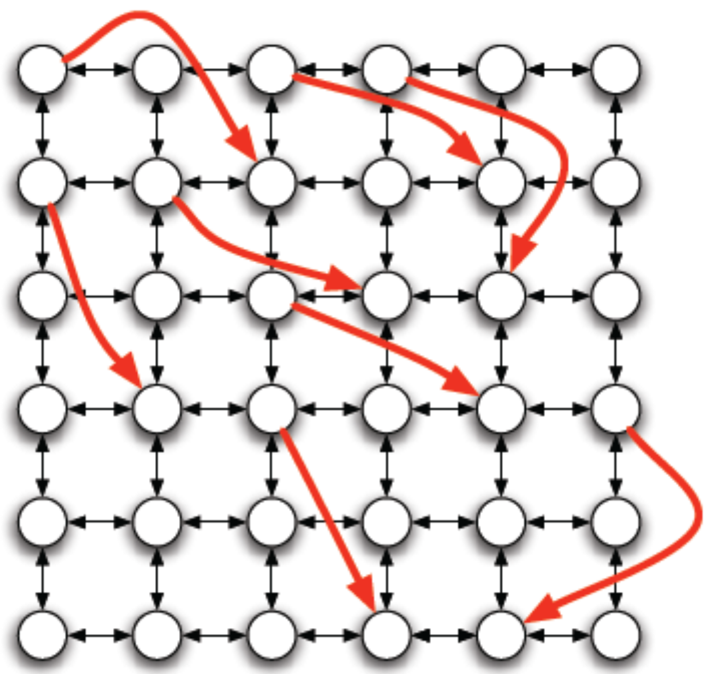
- **We know:**
  - $\alpha = 0$ (i.e., Watts-Strogatz): We need $O(\sqrt{n})$ steps
  - $\alpha = 1$: We need $O(log(n)^2)$ steps



Intuition: Why Search Takes Long



Small α: too many long links



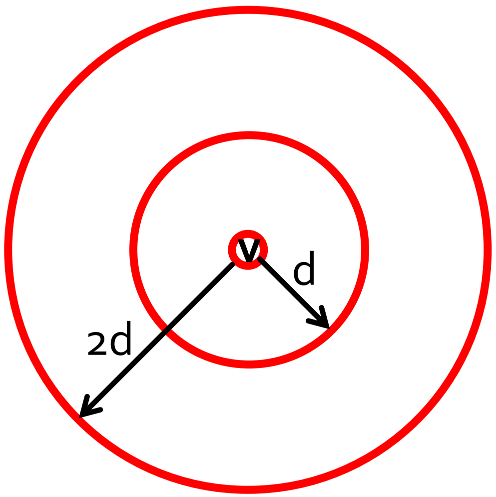
Big α: too many short links

Why Does It Work?

- **How does the argument change for 2-d grid:**
  - $P(u \text{ points to } I) > 1/Z \cdot size(I) \cdot P(u \rightarrow v)$

$$\ln n \qquad d^2 \qquad d^{-2} \Rightarrow \alpha = 2$$

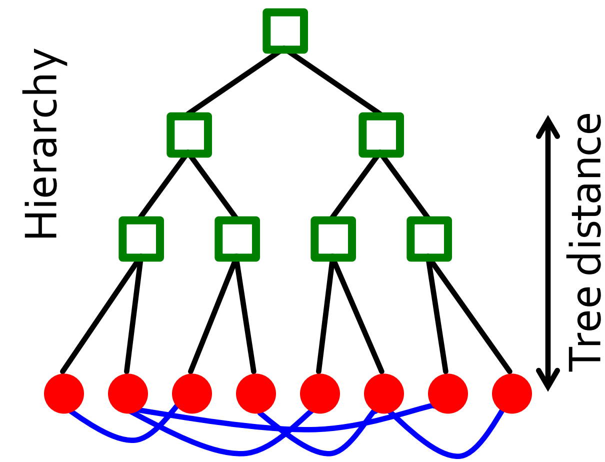
- **Why $P(u \rightarrow v) \sim d(u,v)^{-dim}$ works?**
  - Approx uniform over all "scales of resolution"
  - # points at distance d grows as d^{dim}, prob. d^{-dim} of each edge → const. prob. of a link, independent of d



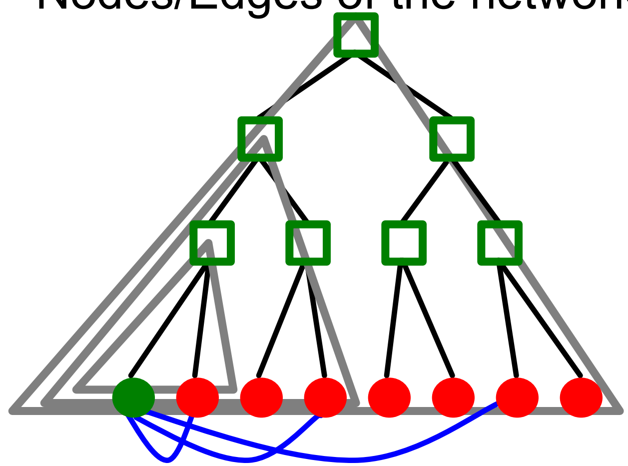
Number of nodes is $\propto d^2$
Prob. of linking each is $\propto d^{-2}$

Different Model: Hierarchies

- **Nodes are in the leaves of a tree:**
  - Departments, topics, …
- **Create k edges out of a node**
  - Create i-th 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
- **You will prove C1 & C2 in HW1!**



Nodes/Edges of the network

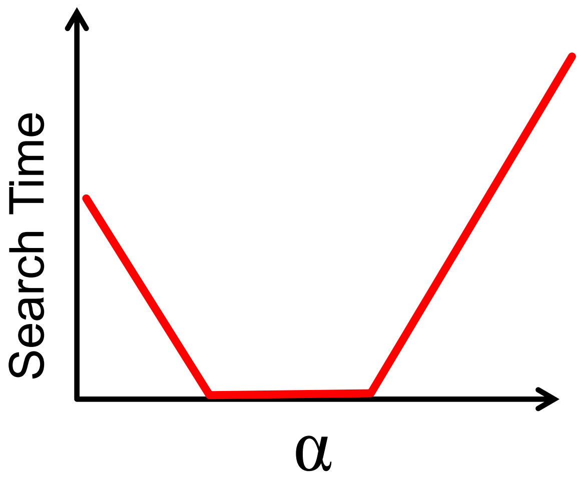


Node has 1 link to each direct subtree

[Watts-Dodds-Newman '02]

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

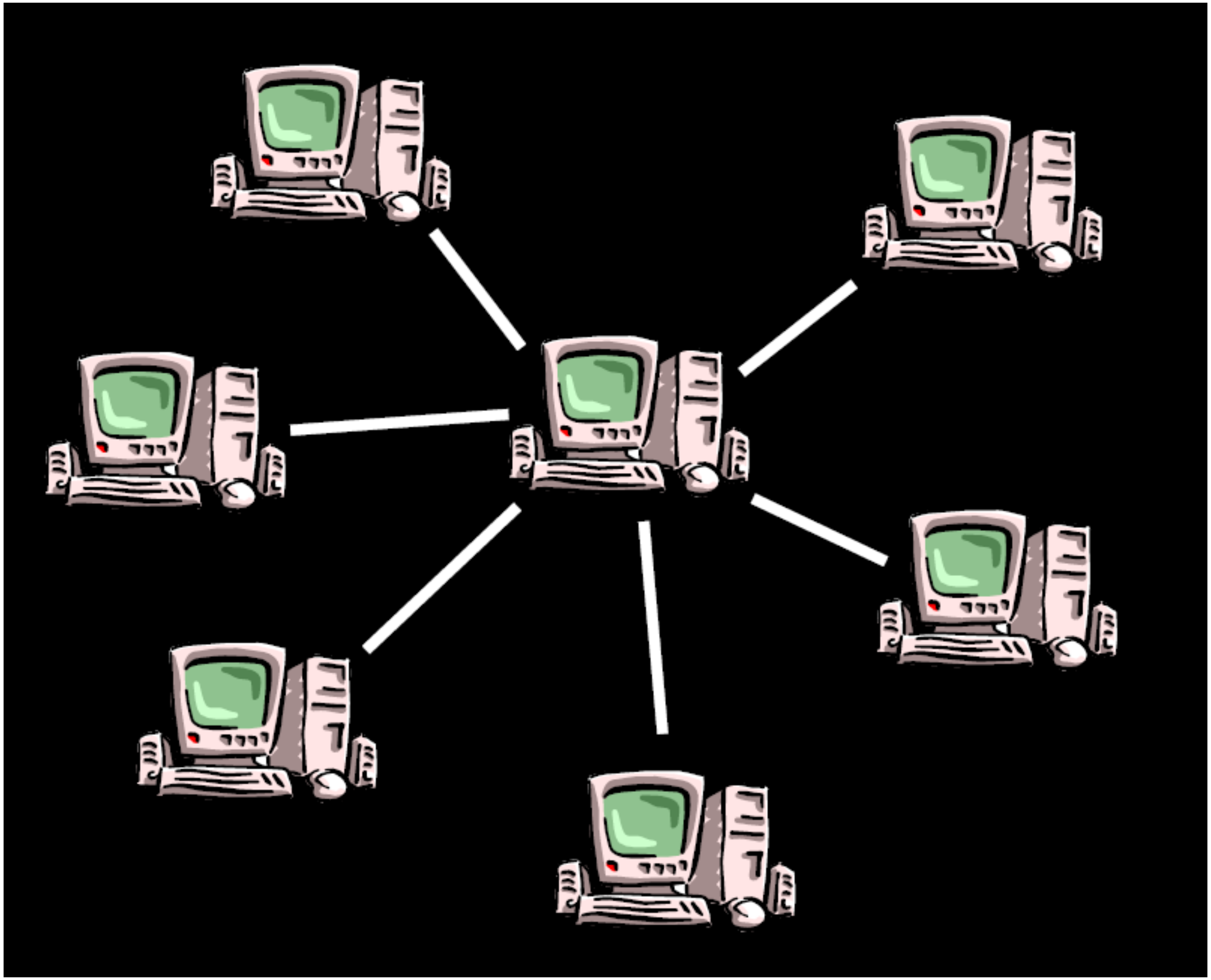


Search in P2P Networks

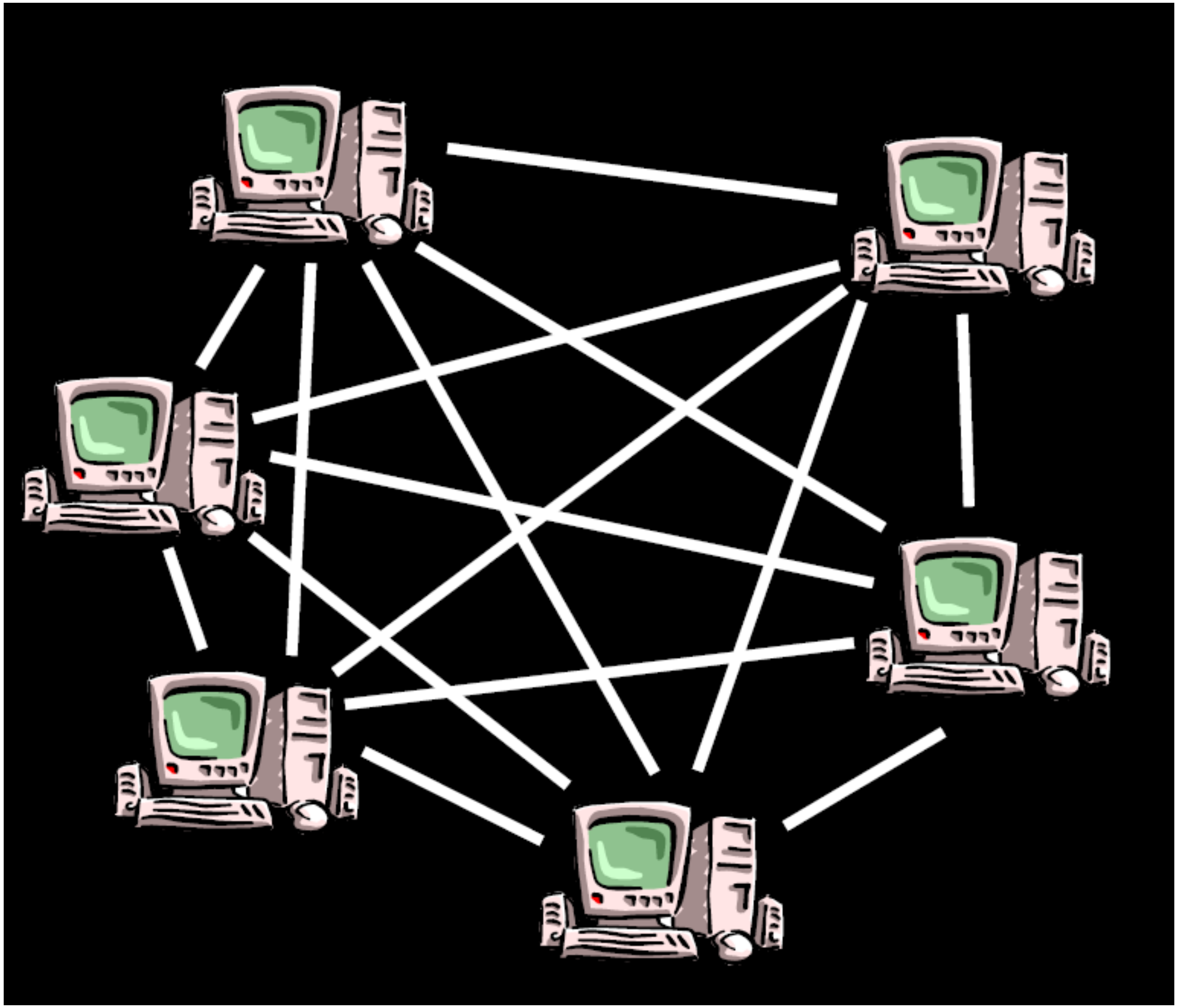
Algorithmic consequence of small-world:

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

Client – Server



P2P: Only Clients



Napster

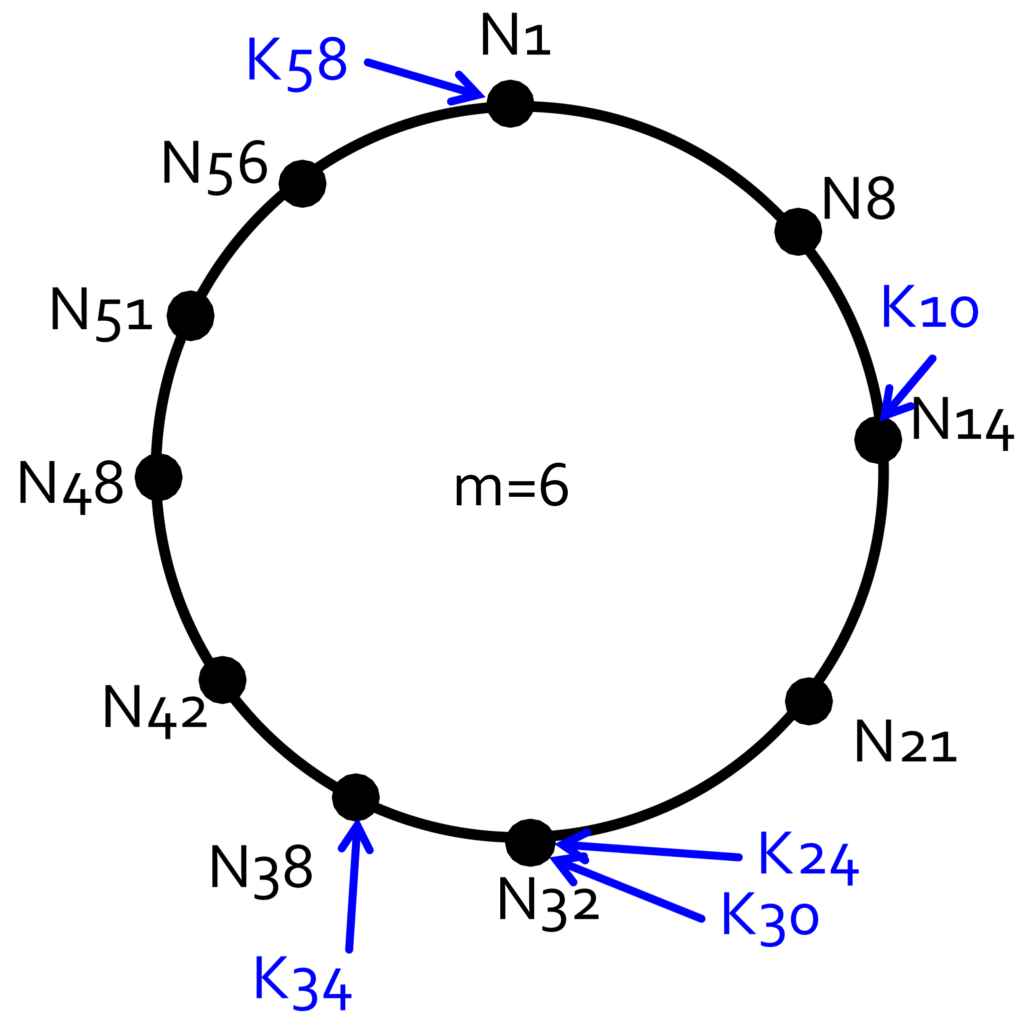
- Napster existed from June '99 and July '01
- Hybrid between P2P and a centralized network
- Once lawyers got the central server to shut down, the network fell apart

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 ≥ k

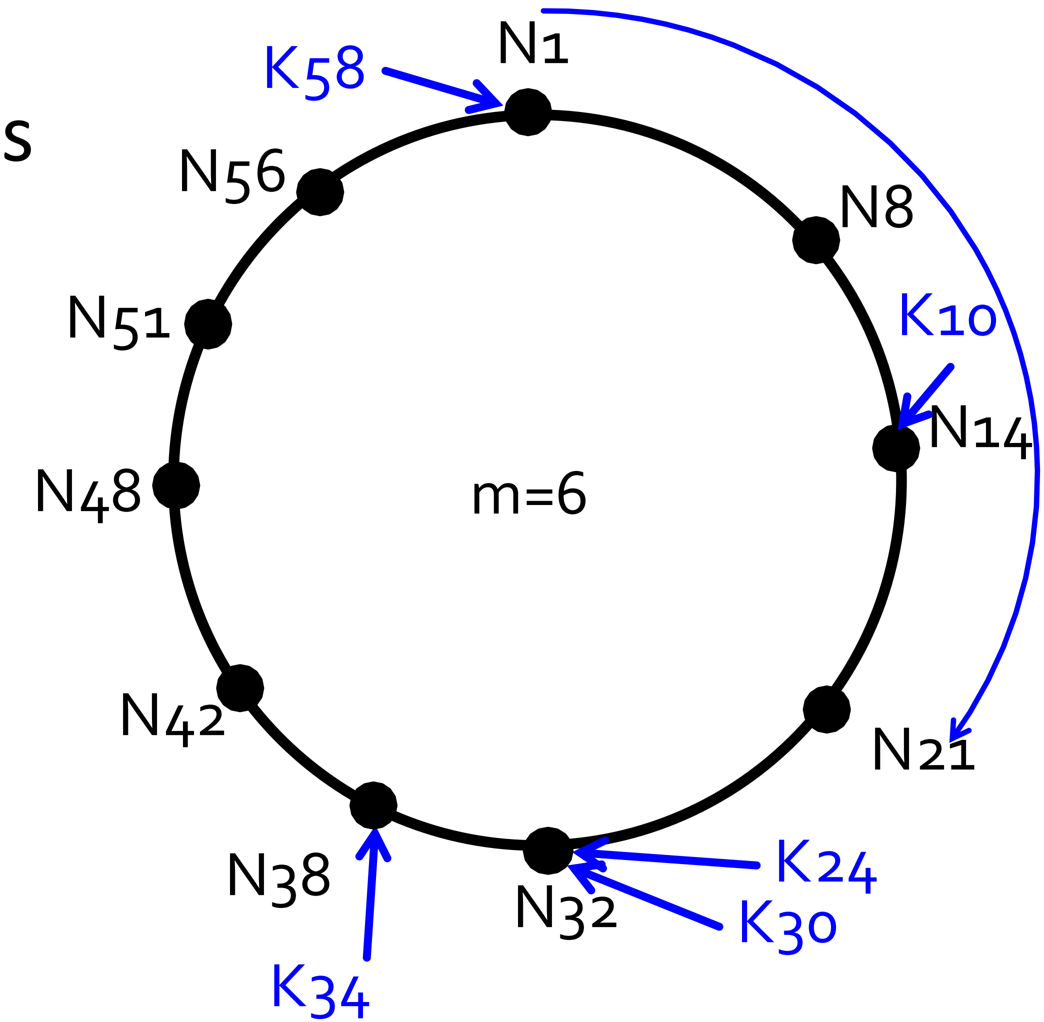


Chord: Basics

- 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

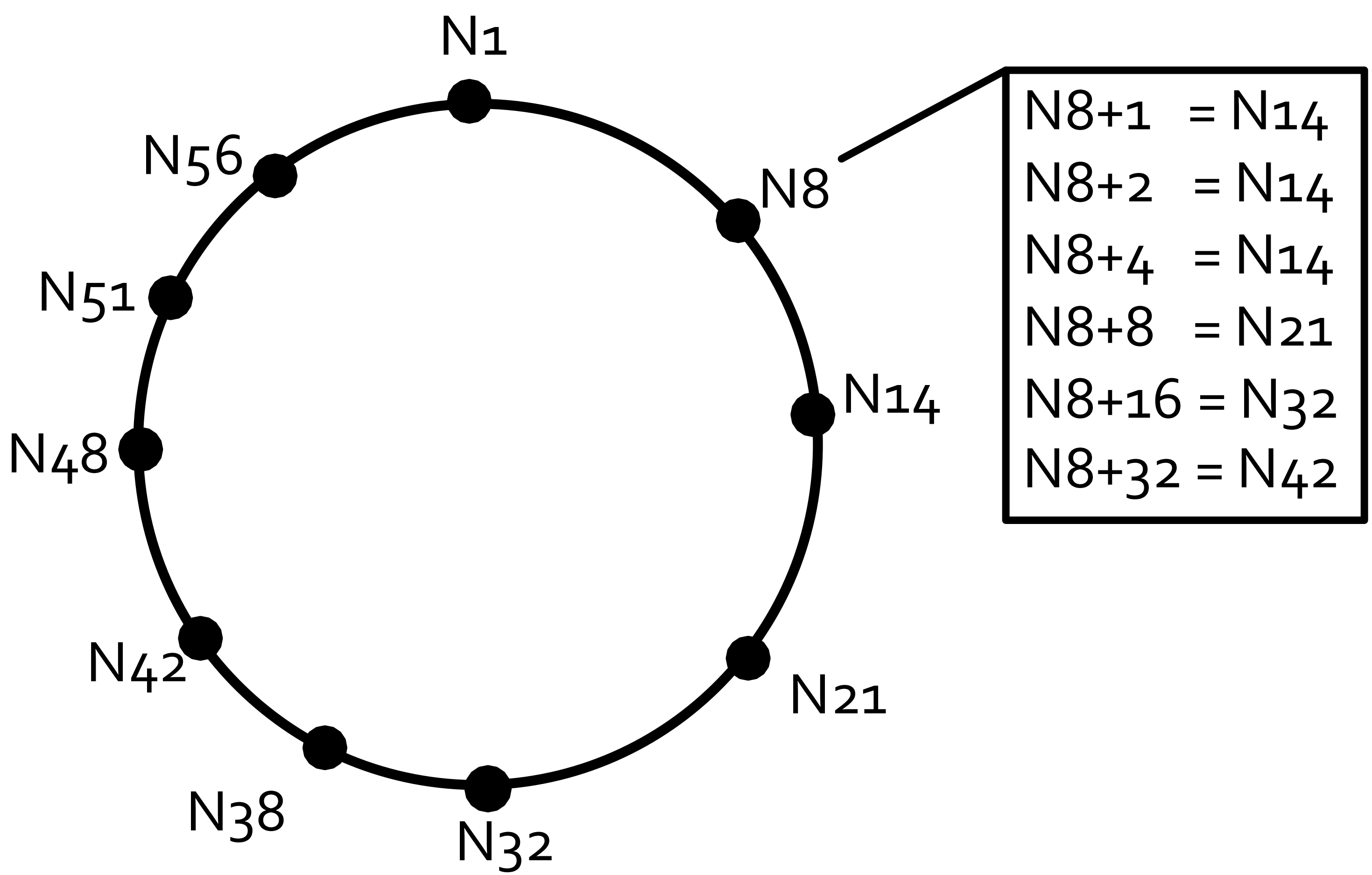
Faster Search:

- 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

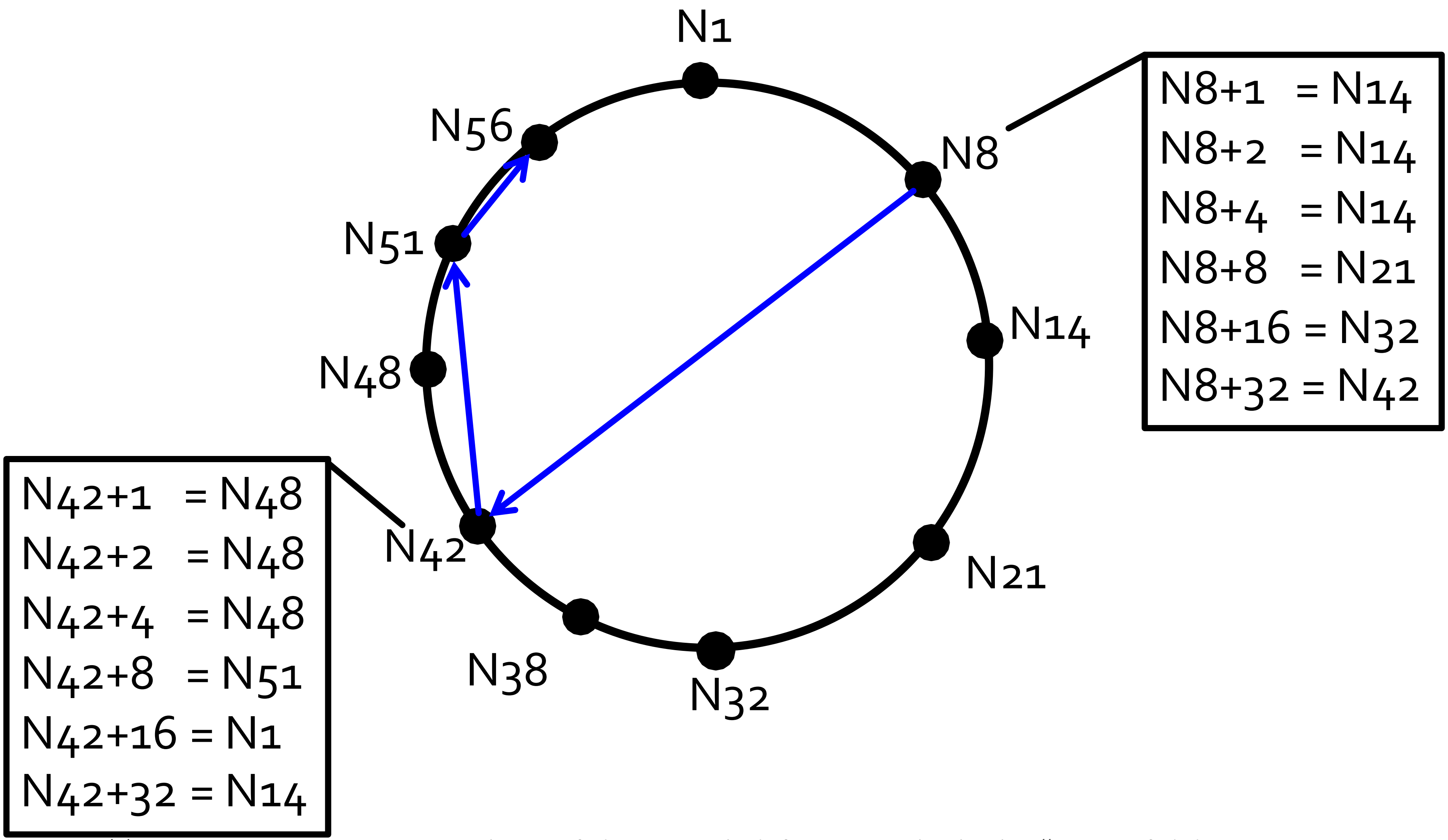
Search algorithm:

- 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



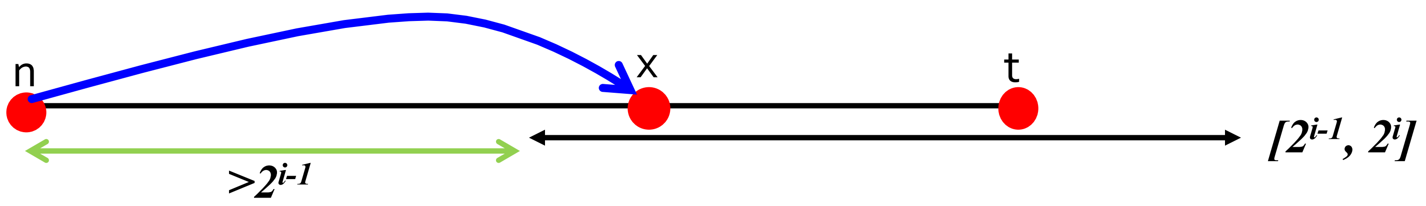
How Long Does It Take to Find a Key?

- **Claim:** Search for any key in the network of N nodes visits **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 $\boldsymbol{i}$ be a number such that t is contained in interval $[\boldsymbol{n}+2^{i-1}, \boldsymbol{n}+2^{i}]$ (for some i)
- Then the table at node $\boldsymbol{n}$ contains a pointer to node $\boldsymbol{x}$ that is the first node past node id $\boldsymbol{n}+2^{i-1}$
- **Claim:** Node $\boldsymbol{x}$ is closer to $\boldsymbol{t}$ than $\boldsymbol{n}$



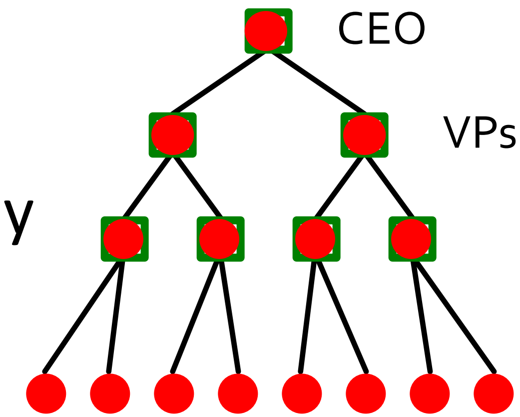
- **So, in one step we halved the distance to** $\boldsymbol{t}$
- We can do this at most $\boldsymbol{log_2 N}$ times
- Thus, we find $\boldsymbol{t}$ in $\mathbf{O}(\boldsymbol{log_2 N})$ steps

Empirical Studies of Navigation in Small-World Networks

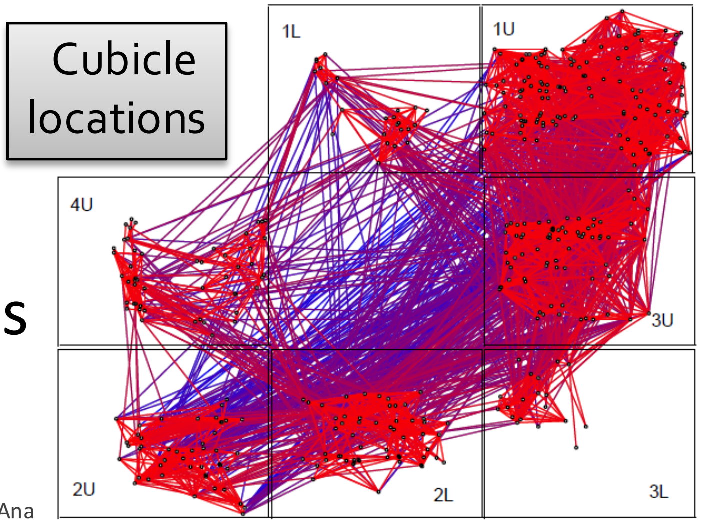
[Adamic-Adar 2005]

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:
      $P(u \rightarrow v) \sim 1 \,/\, (\text{org. hierarchy distance})^{3/4}$



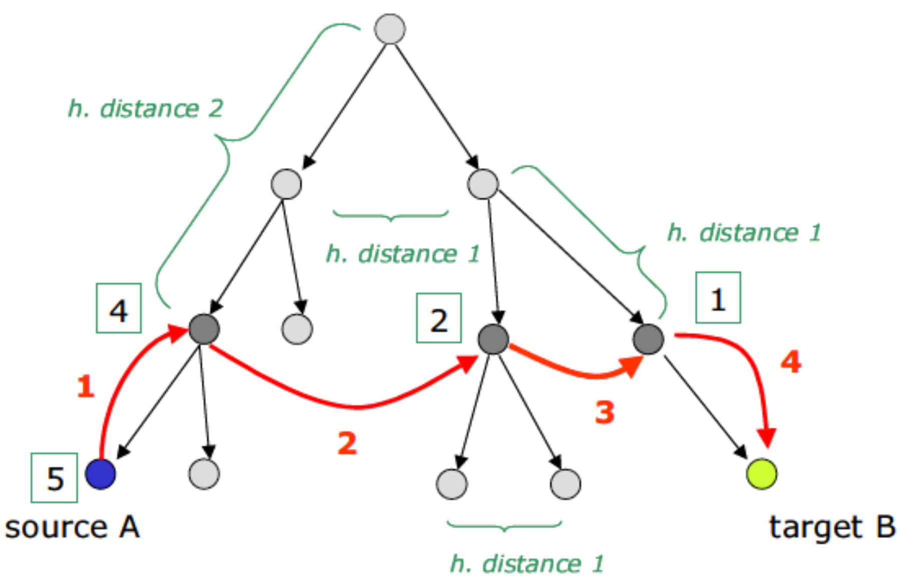
- **Differences from the hierarchical model:**
  - Data has weighted edges
  - Data has people on non-leaf nodes
  - Data not *b*-ary or uniform depth

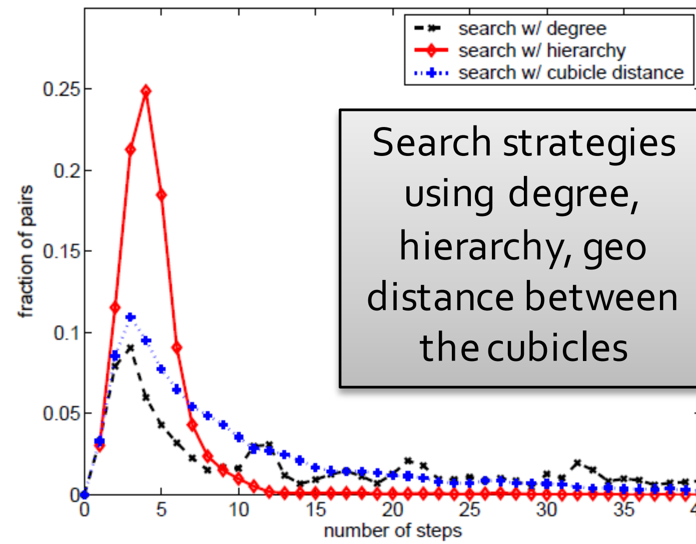


[Adamic-Adar 2005]

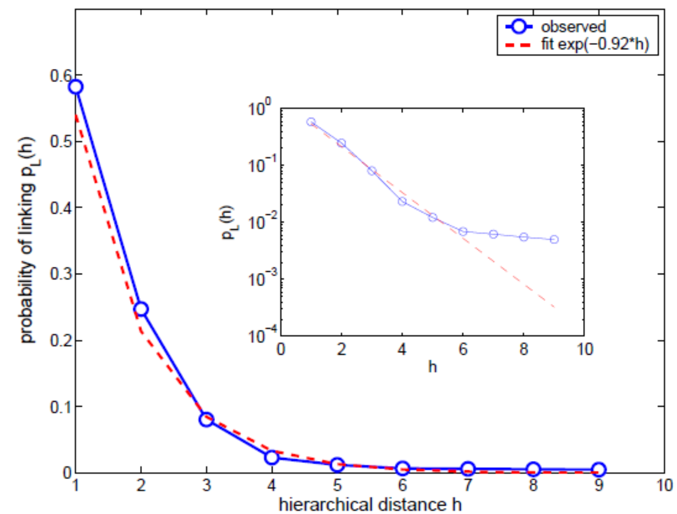
Small-World in HP Labs

- **Generalized hierar. model:**
  - Arbitrary tree defines "groups" = rooted subtrees
  - $P(u \rightarrow v) \sim 1 /$ (size of the smallest group containing u, v)





Search strategies using degree, hierarchy, geo distance between the cubicles



Prob. of link vs. distance in the hierarchy

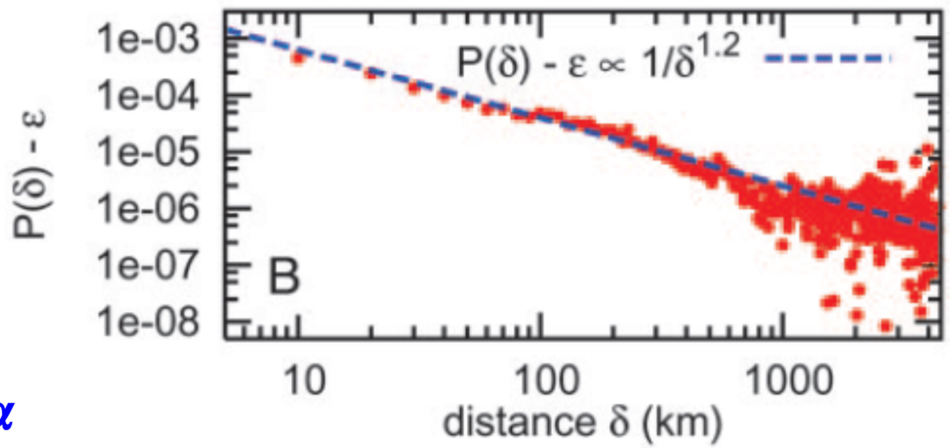
Small-World in LiveJournal

Liben-Nowell et al. '05:

- LiveJournal data
  - Bloggers + zip codes

- **Link prob.: $P(u,v)=\delta^{-\alpha}$**
- α = ?

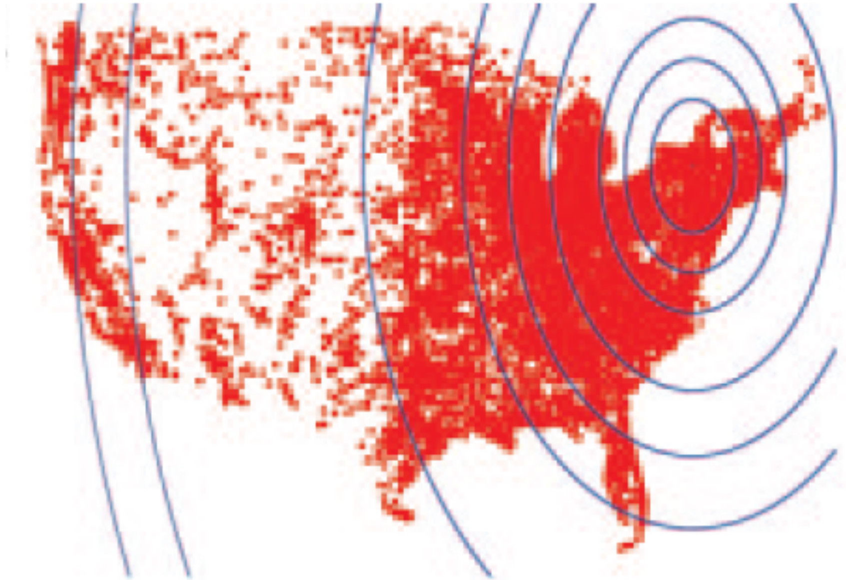
- **Problem:**
  - Non-uniform population density
- **Solution: Rank based friendship**

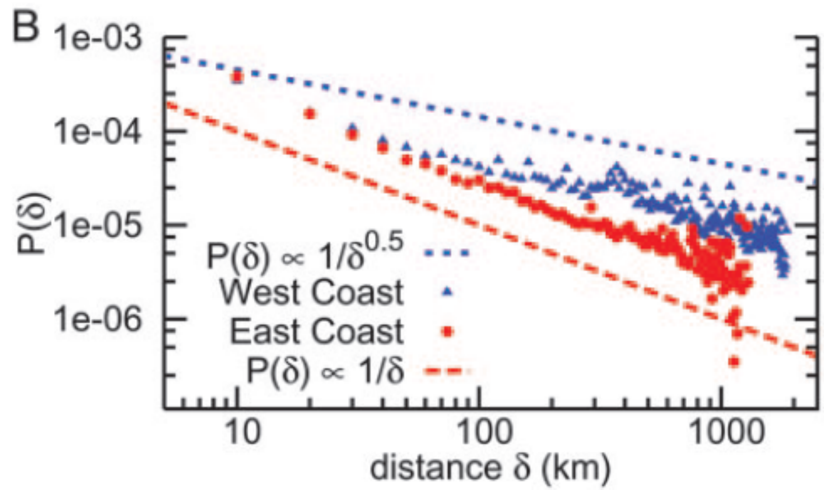


Link length in a network of bloggers (0.5 million bloggers, 4 million links)

Improved Model

[Liben-Nowell et al. '05]





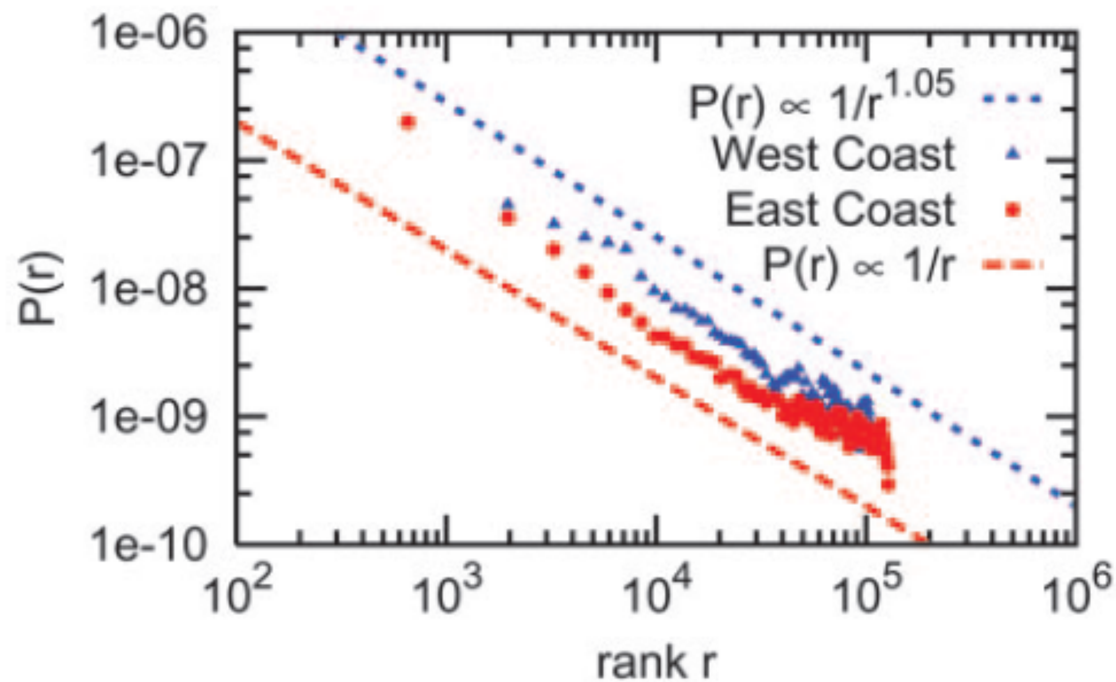
$$rank_u(v) := \left|\{w : d(u,w) < d(u,v)\}\right|$$

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

[Liben-Nowell et al. '05]

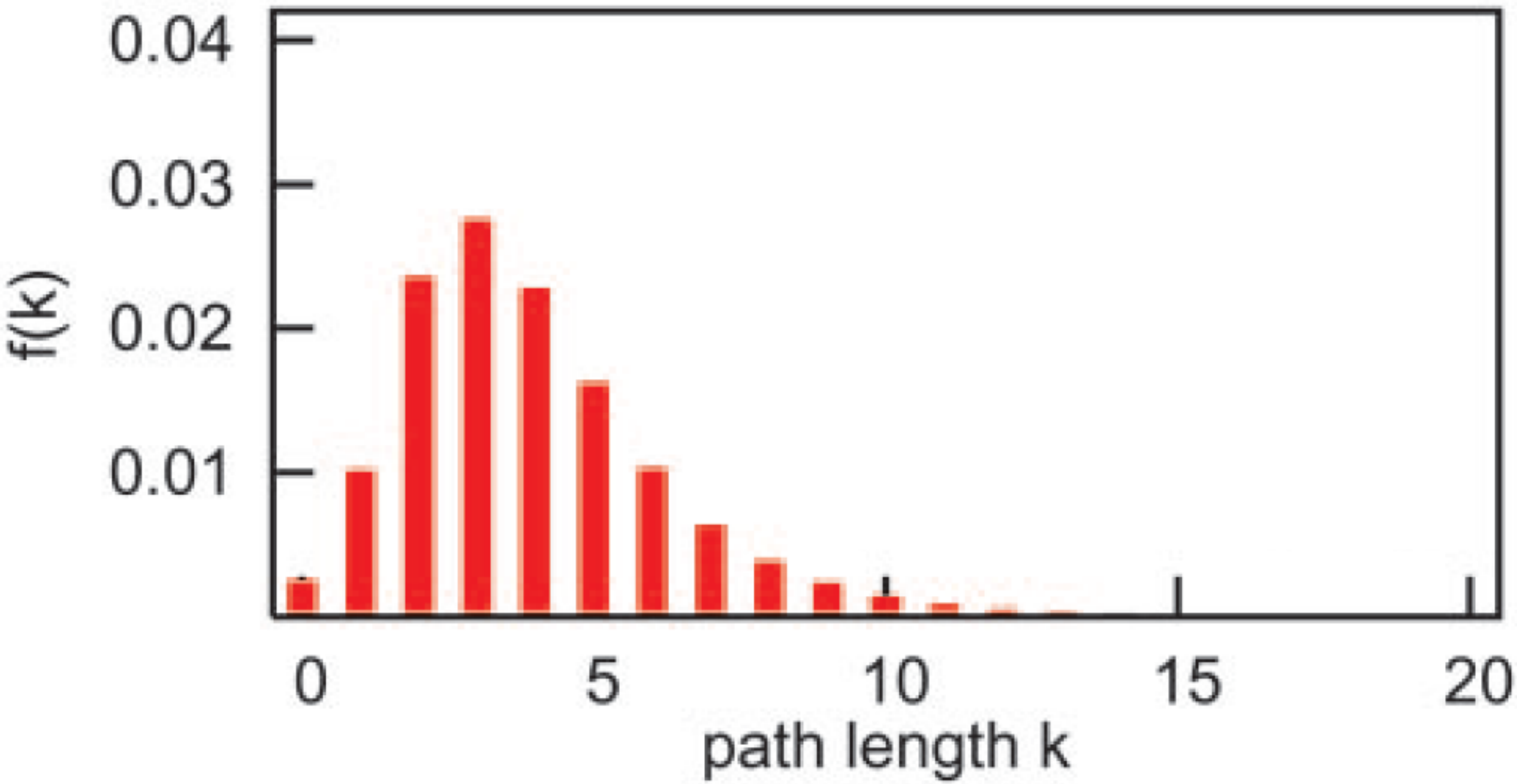
Rank Based Friendships

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



The difference between the East and West coast disappears!

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 α = 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	Models	Algorithms
Small diameter, Edge clustering	Erdös-Renyi model, Small-world model	Decentralized search
Patterns of signed edge creation	Structural balance, Theory of status	Models for predicting edge signs
Viral Marketing, Blogosphere, Memetracking	Independent cascade model, Game theoretic model	Influence maximization, Outbreak detection, LIM
Scale-Free	Preferential attachment, Copying model	PageRank, Hubs and authorities
Densification power law, Shrinking diameters	Microscopic model of evolving networks	Link prediction, Supervised random walks
Strength of weak ties, Core-periphery	Kronecker Graphs	Community detection: Girvan-Newman, Modularity