How to explore big networks?

- **Question**: Perform a random walk on *G*. What is the average node degree among visited nodes, if avg degree in G is 200?

Questions from last time…

- Avg. FB degree is 200 (suppose).
- Q1
  - Take a random node. What's its expected degree?
  - What's an avg degree of it's neighbors?
    - Or… why your friends are more popular than you are?
- **Q2**
  - **Select an edge *e* uniformly at random. What is the expected degree of *e*'s end-nodes?**

- **Q2: Select an edge *e* uniformly at random. What is the expected degree of *e*'s end-nodes?**

Consider the following process:

1) Select an edge *e* uniformly at random
2) Select uniformly at random one (of two) end-node of *e*. Call it *v**.

What is the expected node degree of *v**?

Probability that *e* neighbors with *v*

Probability that we select *v* out of the two end-nodes of *e*

$$\Pr(v^* = v) = \frac{k_v}{|E|} \cdot \frac{1}{2} = \frac{k_v}{\sum_{u \in V} k_u}$$

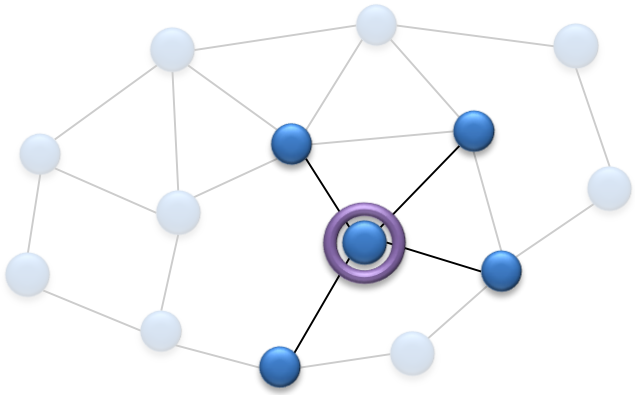
Average node degree in the graph

$$\mathrm{E}[k_{v^*}] = \sum_{v \in V} k_v \cdot \Pr(v^* = v) = \frac{\sum_{v \in V} k_v^2}{\sum_{v \in V} k_v} \geq \bar{k}$$

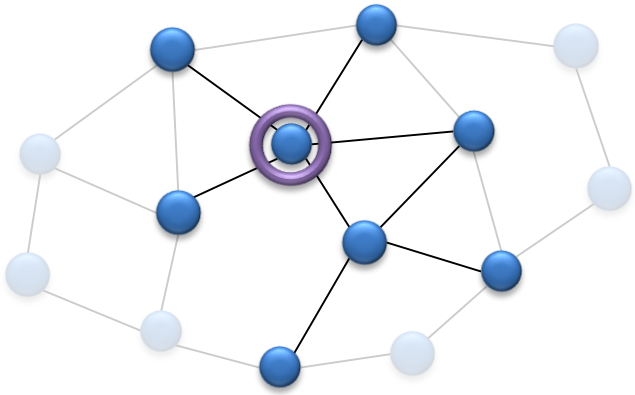
So discovering nodes through edges leads to bias towards nodes of high degree!

This and below slides are by dr. Maciej Kurant (Google Zurich)

Graph exploration

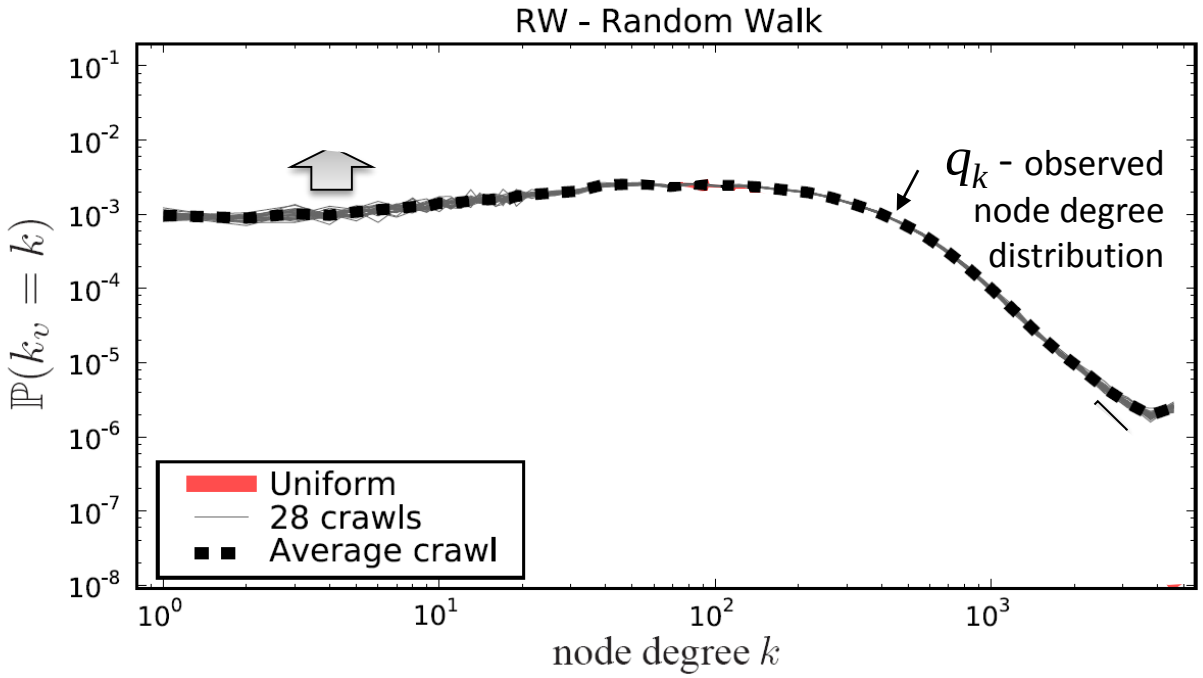


Graph exploration



- Often the only possibility
- Example: **Random Walk (RW):** At every iteration, the neighbor is selected uniformly at random.

A walk in Facebook



Real average node degree :

$\langle k \rangle = \sum_k k\, p_k \cong 94$

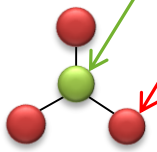
Observed average node degree :

$\sum_k k\, q_k \cong 338$

$Pr(\text{sampling } v) \sim k_v$

degree of node v

A neighbor of a typical node?



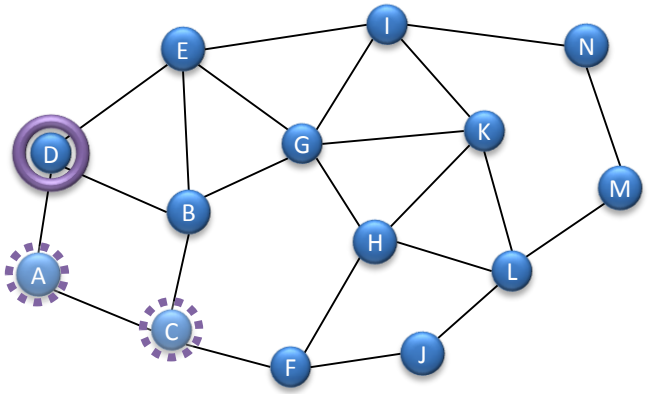
Not typical!

A end-node of a typical edge?

Not typical!

How to get an unbiased sample?

Metropolis-Hastings Random Walk (**MHRW**) on undirected graph:



S = D A A C …

$\deg(A) < \deg(D) \Rightarrow$ go with probability 1

$\deg(C) > \deg(A) \Rightarrow$ go with probability $\frac{\deg(A)}{\deg(C)}$

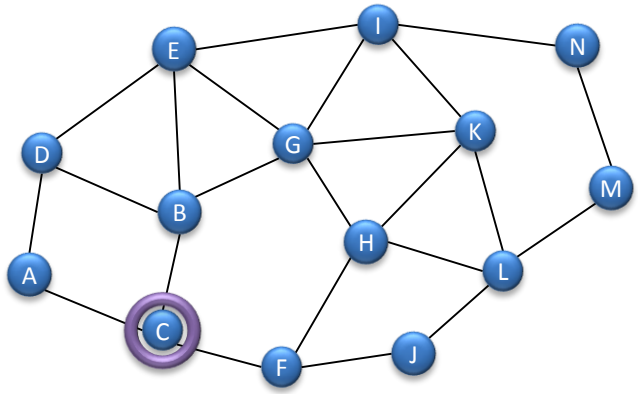
$\deg(C) > \deg(A) \Rightarrow$ go with probability $\frac{\deg(A)}{\deg(C)}$

…

$$\Pr(v \text{ is from Australia}) = \frac{\sum_{u \in S} 1_{\{v \text{ is from Australia}\}}}{\sum_{u \in S} 1}$$

How to get an unbiased sample?

Metropolis-Hastings Random Walk (**MHRW**) on undirected graph:



S = D A A C …


$\deg(A) < \deg(D) \Rightarrow$ go with probability 1

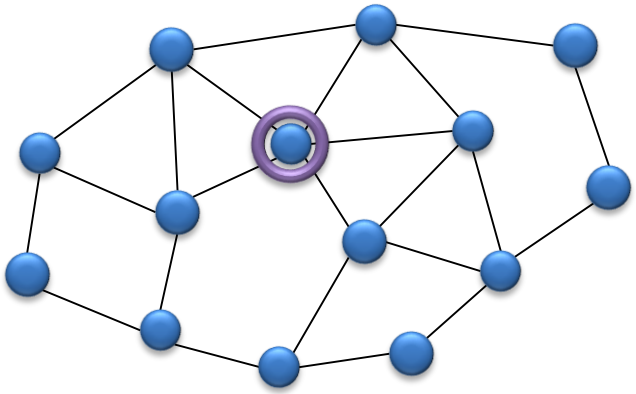

$\deg(C) > \deg(A) \Rightarrow$ go with probability $\frac{\deg(A)}{\deg(C)}$


$\deg(C) > \deg(A) \Rightarrow$ go with probability $\frac{\deg(A)}{\deg(C)}$

…

$$\Pr(v \text{ is from Australia}) = \frac{\sum_{u \in S} 1_{\{v \text{ is from Australia}\}}}{\sum_{u \in S} 1}$$

Re-Weighted Random Walk (**RWRW**) on undirected graph:



Collect a classic (biased) RW sample…

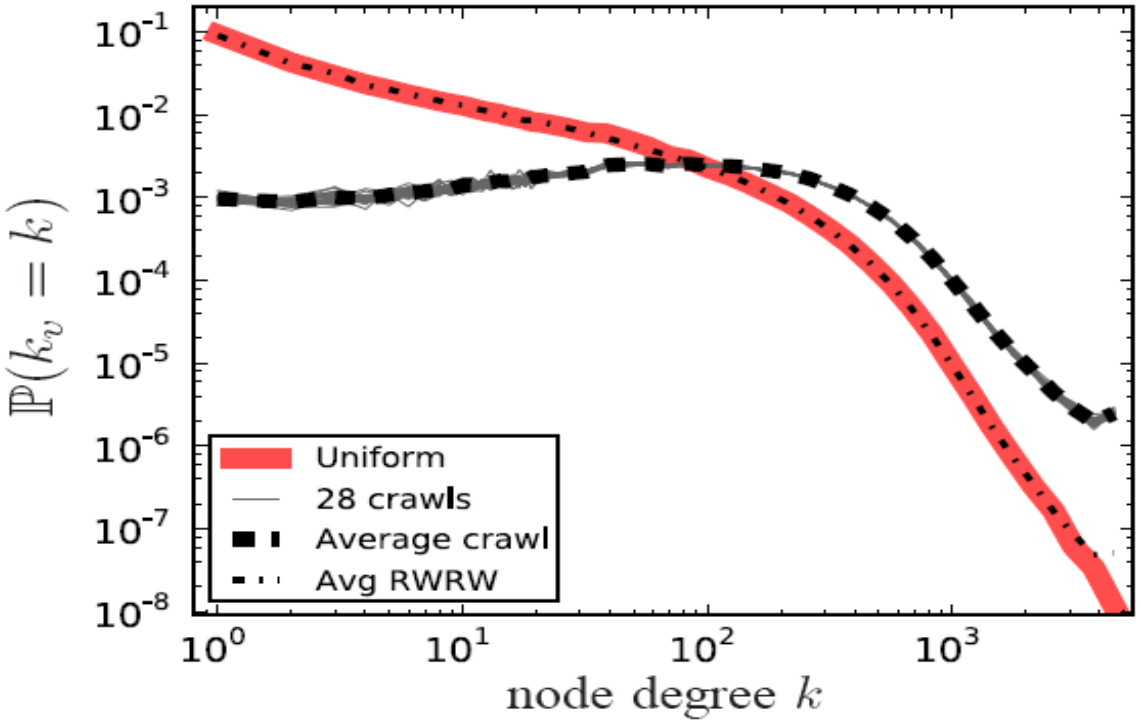
Now apply the Hansen-Hurwitz estimator:

$$\Pr(v \text{ is from Australia}) = \frac{\sum_{u \in S} 1_{\{u \text{ is from Australia}\}} / k_u}{\sum_{u \in S} 1 / k_u}$$

Example

$$p_k = \frac{\sum_{v \in S} \frac{1_{\{k_v = k\}}}{w_v}}{\sum_{v \in S} \frac{1}{w_v}}$$

E.g. node degree distribution
(p_k - fraction of nodes with degree k)



Influence of Node degree

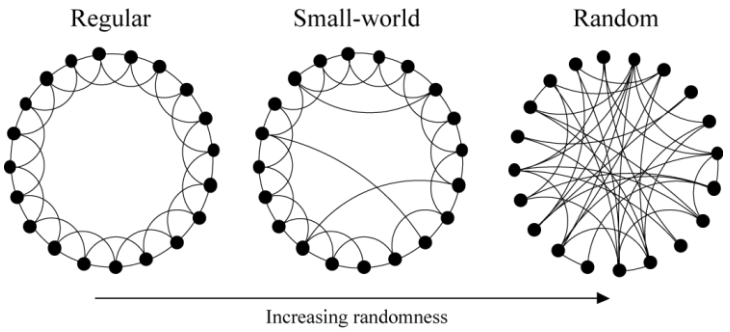
- Node **degree** is correlated with many other properties, such as age, nationality, activity, privacy awareness, … even node ID!
- Have to account for such biases!

Things to Remember

- Random walks in **connected undirected, non-bipartite** graphs lead to stationary distributions
  - the time spent on a node v is **proportional to the degree of v**
  - Directed graphs: requirements of **strong connectivity & aperiodicity**.
    - Sometimes we can fix them -> **PageRank**
- Influence of **node degree** on graph exploration using random walks
- Ways to get **unbiased samples**
  - **MHRW & RWRW**
- **Tricks of Spectral Graph Analysis**

Back to Small Worlds

- Watts-Strogatz model
  - High clusterisation;
  - **Short path length**.



- There is a range of p values where the network exhibits properties of both: random and regular graphs

Short paths in Real Networks

- Milgram's (Small-World) experiment
  - No Online Social Networks in 1960s
  - 296 random people in USA forwarding a letter to a "target" person in Boston.
  - Personal info on the "target" (including address and occupation)
  - Forward only to a person known by first-name basis.
  - 64 chains succeeded.
  - Result?

Milgram's (Small-World) experiment

- Average length of the chains that were completed lied between 5 and 6 steps;
- Coined as "**Six degrees of separation**" principle.
- This was far less than assumed under the 'grid-like' assumption !
  - Similar results have been found in many other social networks
  - In 2011 Facebook shrinked 'Six Degrees Of Separation' to just 4.74 (721m users, 69b friendships)
  - Twitter's 5,91 of 12,8M friendships
- ***So do we know WHY it works?***
- ***Because of Watts-Strogatz model?***
- ***Do you see anything strange even knowing the properties of Watts-Strogatz Small-Worlds?***

Recap: Distance

- The ***distance*** between two nodes in a graph to be the length of the shortest path between them.
  - Can be also interpreted as the ***sum of edge weights*** on a weighted graph

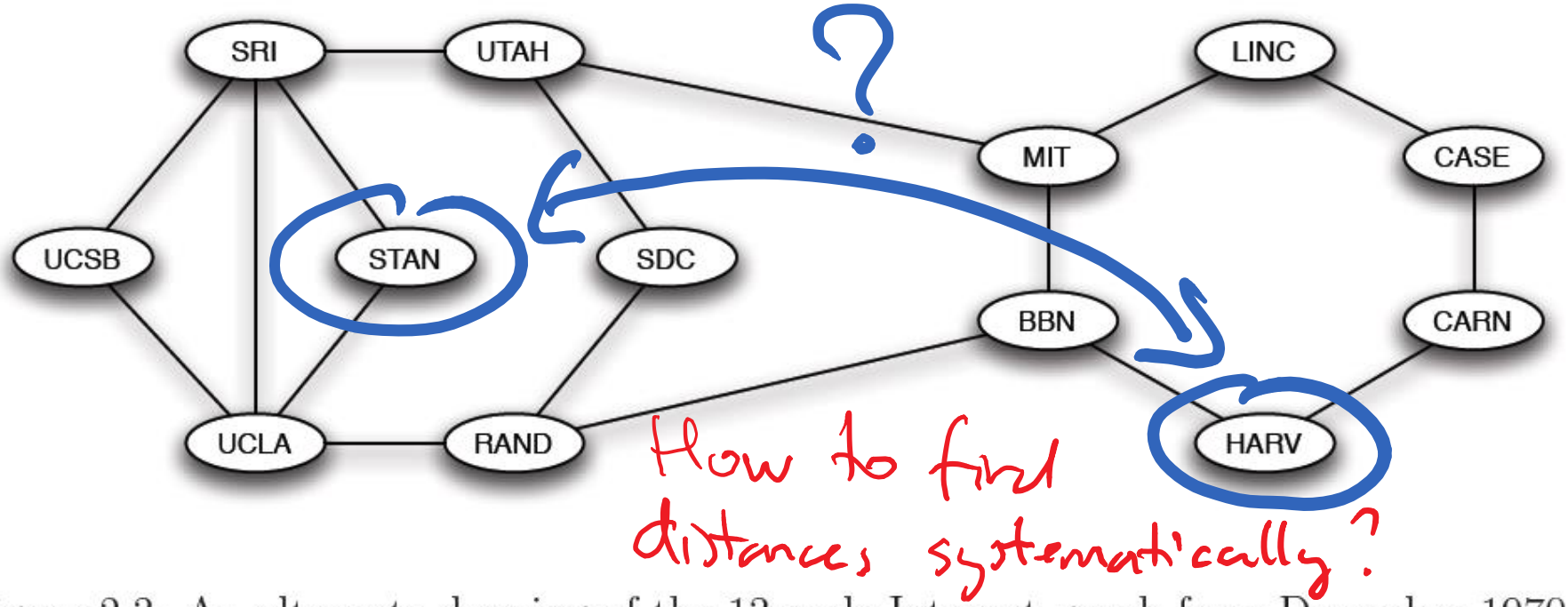


Figure 2.3: An alternate drawing of the 13-node Internet graph from December 1970.

BFS (Breath First Search)

- (1) You first declare all of your ***actual friends*** to be at ***distance 1***.
- (2) You then find all of ***their friends*** and declare these to be at ***distance 2***.
  - not counting people who are already friends of yours
- (3) Then you find all of ***their friends*** and declare these to be at ***distance 3***.
  - again, not counting people who you've already found at distances 1 and 2
- (...) Continuing in this way, you search in successive layers, each representing the next distance out.

- Each new layer is built from all those nodes that (i) have not already been discovered in earlier layers, and that (ii) have an edge to some node in the previous layer.

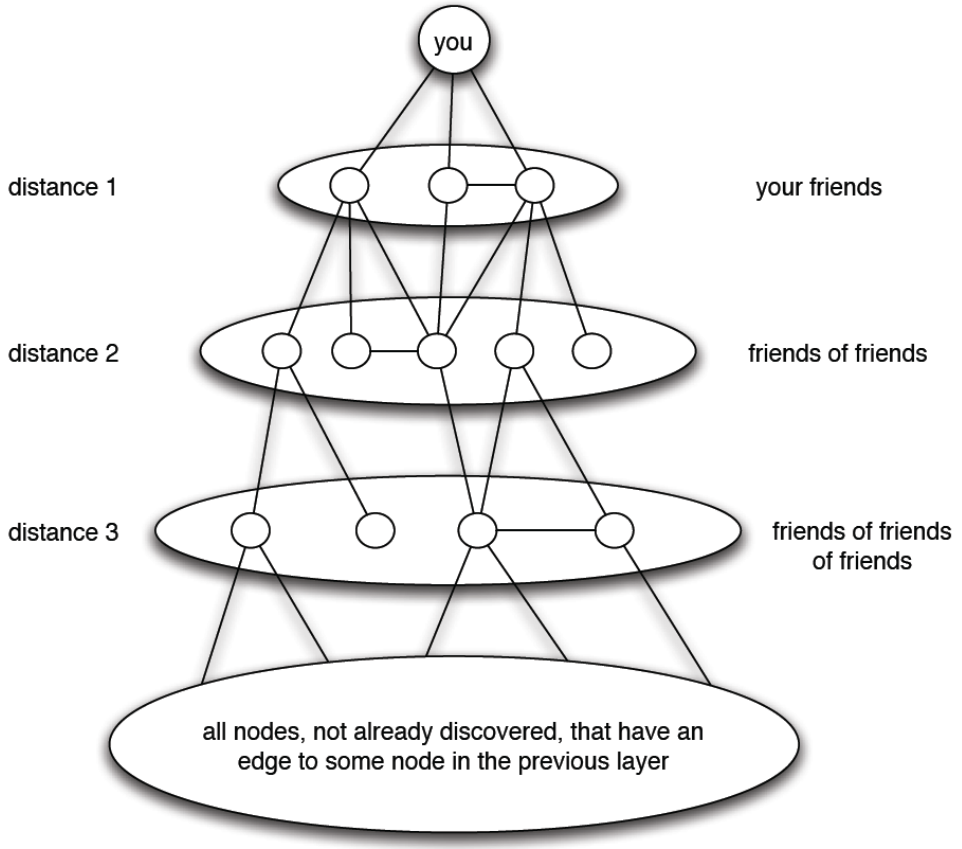


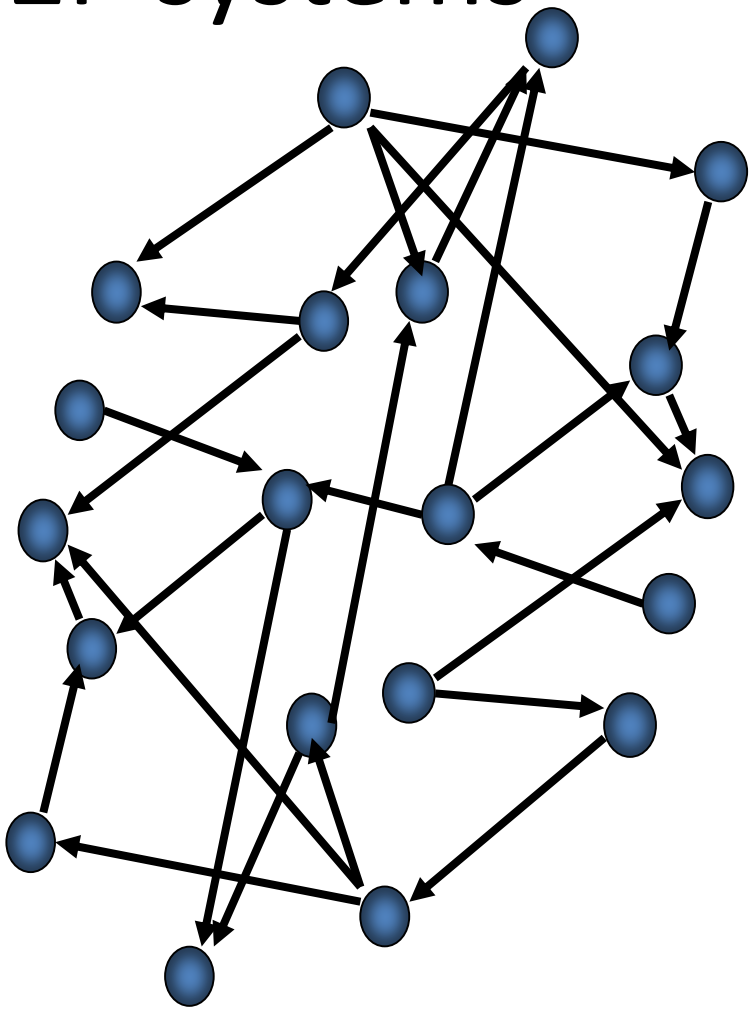
Figure 2.8: Breadth-first search discovers distances to nodes one "layer" at a time; each layer is built of nodes that have an edge to at least one node in the previous layer.

Small-World: remaining questions

- Is it enough to explain Milgram's experiment?
  - If there exist shortest path between any two nodes - **where is the global knowledge** that we can find this path?!
  - Why should **arbitrary pairs of strangers be able to find short chains** of acquaintances that link them together???
  - **Why decentralized "search algorithm" works?**
    - Very few nodes were involved in the discovery of the "shortest path"

Implications for P2P systems

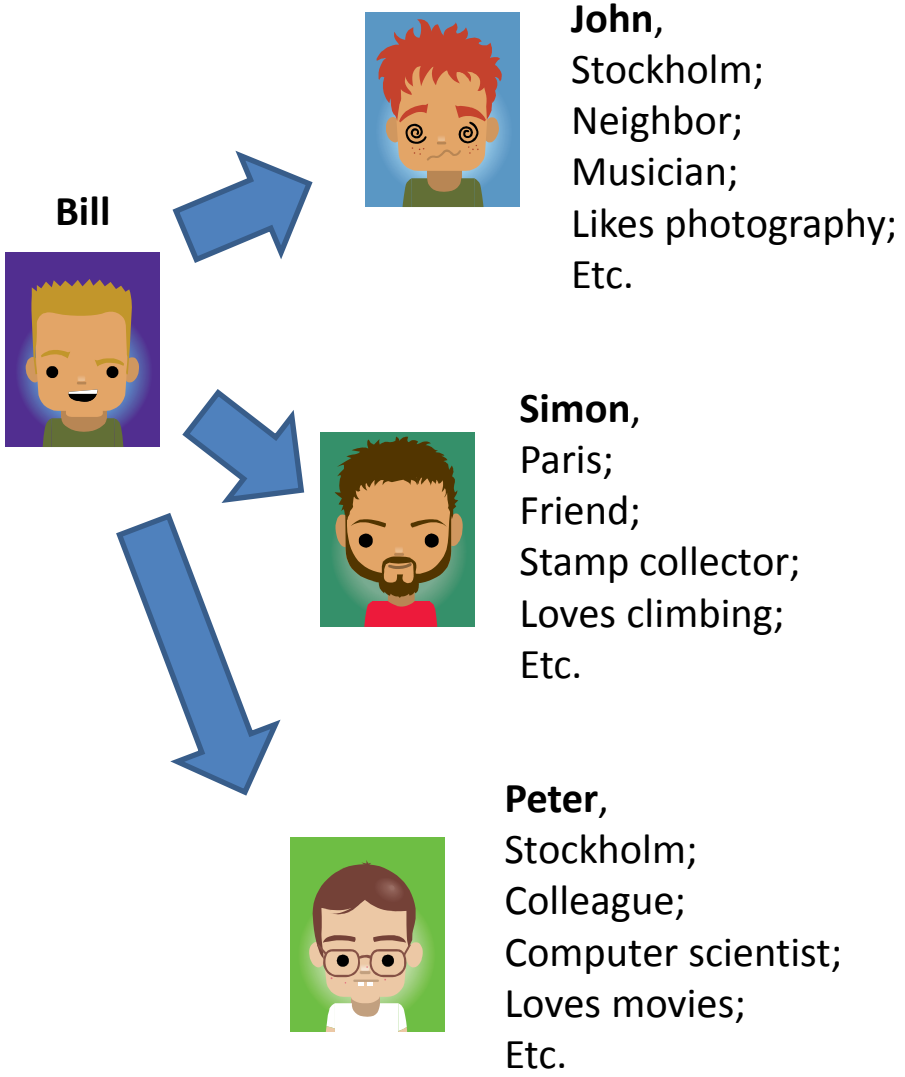
- Task for P2P:
  - Design a **completely decentralized algorithm** that would route message from any node A to any other node B with relatively few hops compared with the size of the graph



- **Is it possible?**
  - Milgram experiment suggests YES!

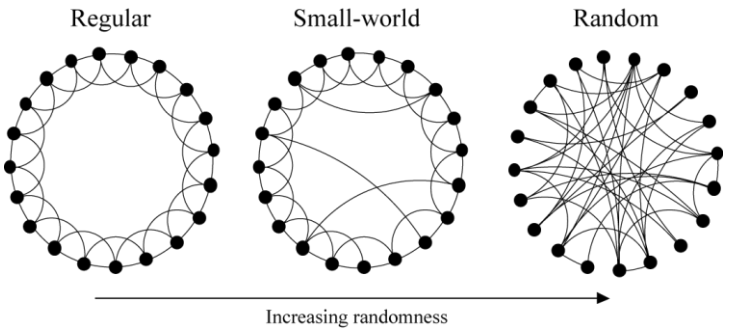
So why Milgram's experiment worked?

- Social network is not a bare graph of vertices and edges, but a graph with certain "labels"
- The "labels" representing various dimensions of our life
  - Hobbies, work, geographical distribution etc.
- There is (are multiple) "**labeling space(s)**" with a **distance metric**!!!
- We can greedily minimize the distance!!
  - Decentralized search: a greedy-routing algorithm
  - We need to build right graph where decentralized algorithm might perform the best



Navigation in Watts-Strogatz Small-Worlds

- Watts-Strogatz model
  - High clusterisation;
  - **Short path length**.



- Construction involves a notion of "**ID space**" and a "**distance**" function.
  - Think how to connect to k "closest neighbors" in the initial step…
- **Short Paths exist** in Watts-Strogatz model, but decentralized greedy routing **can not find** them!

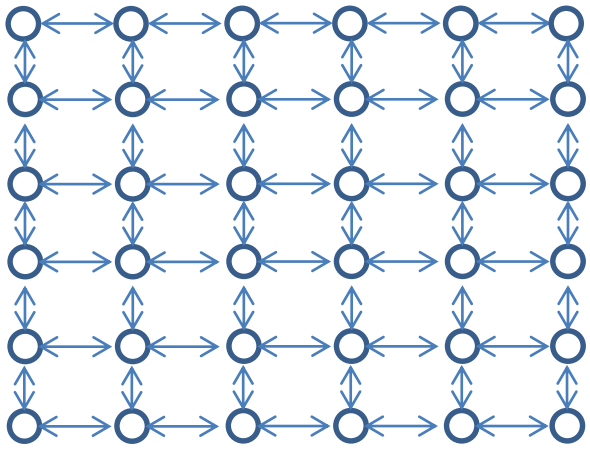
Kleinberg's model of Small-Worlds

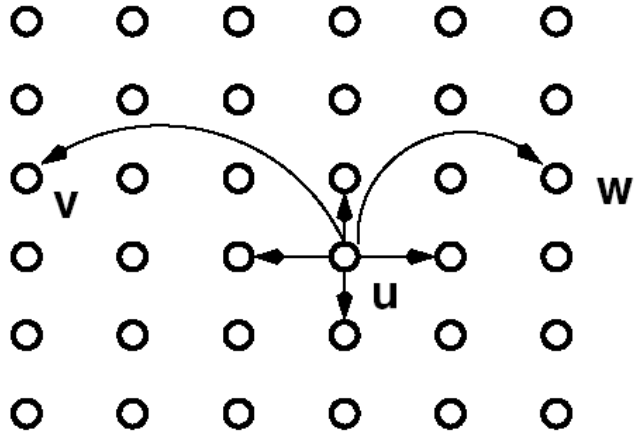
- **Research of Jon Kleinberg:**
  - Claims that Watts and Strogatz model **is not effective** for decentralized search;
  - presents the **infinite family of SW networks** that generalizes Watts and Strogatz model and shows that decentralized search algorithms can **find short paths** with high probability;
  - proves that there exist only **one unique model** within that family for which decentralized algorithms **are effective**.

Navigable Small-World networks

- Kleinberg's Small-World's model
  - 2-dimensional lattice
  - Lattice (Manhattan) **distance**
  - Two type of links:
    - Short range (neighborhood lattice)
    - Long range
      - Probability for a node *u* to have a node *v* as a long range contact is proportional to

$$P(u \to v) \sim \frac{1}{d(u,v)^r}$$





Influence of “r”

- Each peer *u* has link to the peer *v* with probability proportional to $\frac{1}{d(u,v)^r}$ where d(u,v) is the distance between *u* and *v.*

- **Tuning “r”**
  - When **r=0** – long range contacts are chosen uniformly. Random graph theory proves that there exist short paths between every pair of vertices, **BUT there is no decentralized algorithm capable finding these paths**
  - When **r<dim** we tend to choose more far away neighbors (decentralized algorithm can quickly approach the neighborhood of target, but then slows down till finally reaches target itself).
  - When **r>dim** we tend to choose more close neighbors (algorithm finds quickly target in it’s neighborhood, but reaches it slowly if it is far away).
  - When **r=dim** (dimension of the space), the algorithm exhibits optimal performance.

Performance with r=dim

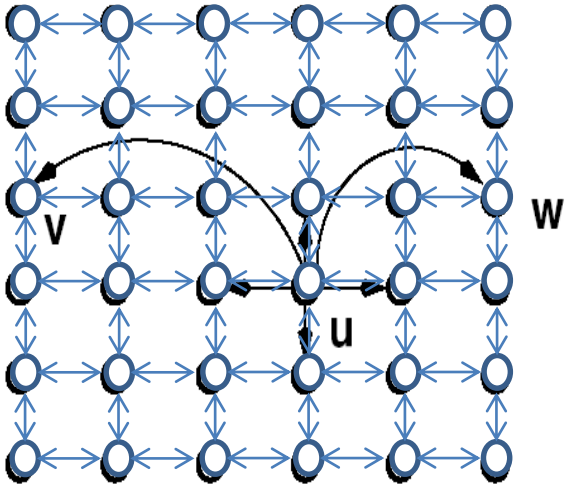
- When there is one long range link (q = 1):
  - The expected search cost is bounded **by $O(\log^2 N)$**
- When there are constant number of long range links (q = k):
  - The expected search cost is bounded by

    $O(\log^2 N)/k$
- When q = logN:
  - The expected search cost is bounded **by O(log N)**
  - Notice similarity with Chord's performance!

How does it work in practice?

$$P(u \to v) \sim \frac{1}{d(u,v)^r}$$

$$P(u \to v) = \frac{1}{d(u,v)^r} \cdot \frac{1}{Z}$$



- Normalization constant have to be calculated:

$$Z = \sum_{\forall i \neq u} \frac{1}{d(u,i)^r}$$

Example

- Choose among 3 friends (1-dimension)
  - A (1 mile away)
  - B (2 miles away)
  - C (3 miles away)
- Normalization constant

$$\sum_{\forall i \neq u} \frac{1}{d(u,i)} = \frac{1}{1} + \frac{1}{2} + \frac{1}{3} = \frac{11}{6}$$

$$P(selectingA) = \frac{\frac{1}{1}}{\frac{11}{6}} = \frac{6}{11}$$

$$P(selectingB) = \frac{\frac{1}{2}}{\frac{11}{6}} = \frac{3}{11}$$

$$P(selectingC) = \frac{\frac{1}{3}}{\frac{11}{6}} = \frac{2}{11}$$