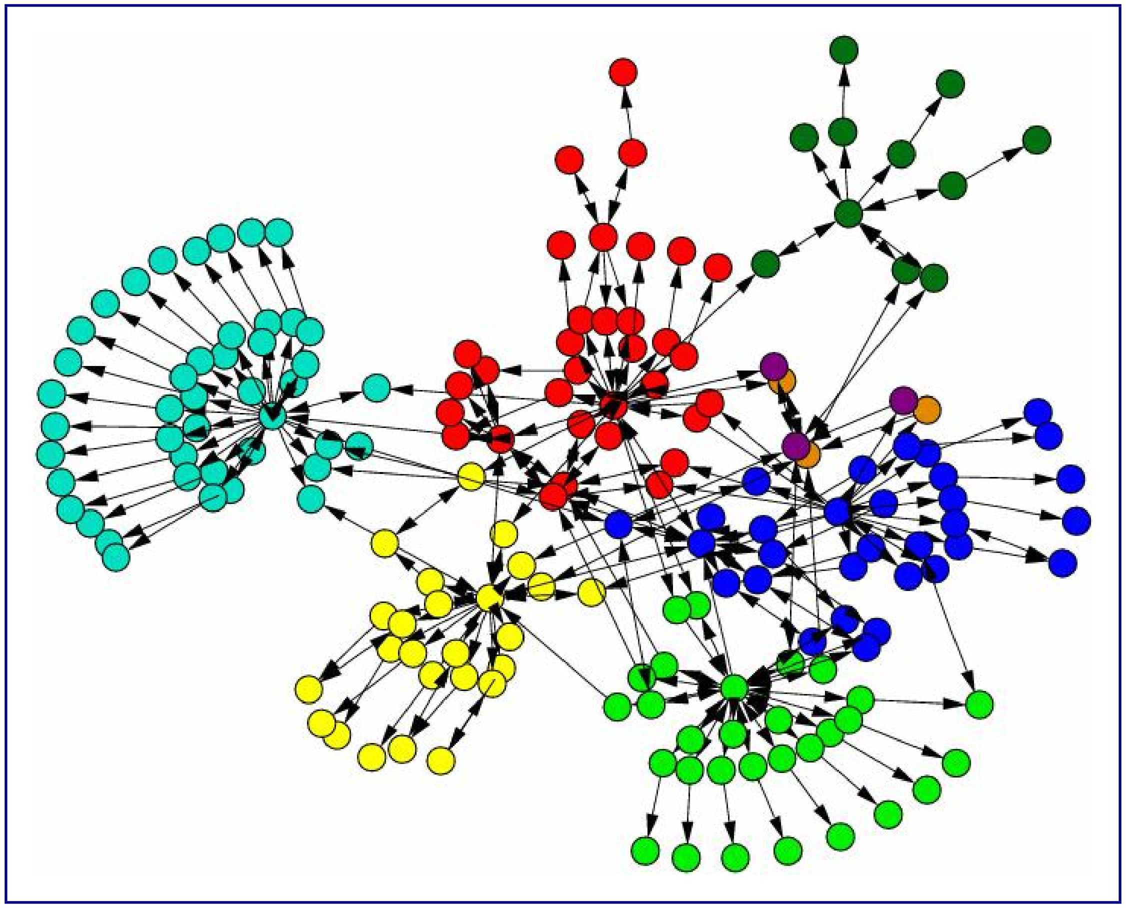
ECS 253 / MAE 253, Lecture 9

May 1, 2023



“Web search and decentralized search on small-world networks”

Search for information

Assume some resource of interest is stored at the vertices of a network:

- Web pages
- Files in a file-sharing network

Would like to determine rapidly where in the network a particular item of interest can be found.

To warehouse data or search on demand?

- Centralized : Catalogue data in one central place.

  – Makes most sense when high cost to search network in real time.

  – Requires resources for learning the data and storing it.

- Decentralized : Data is spread out in a distributed data base.

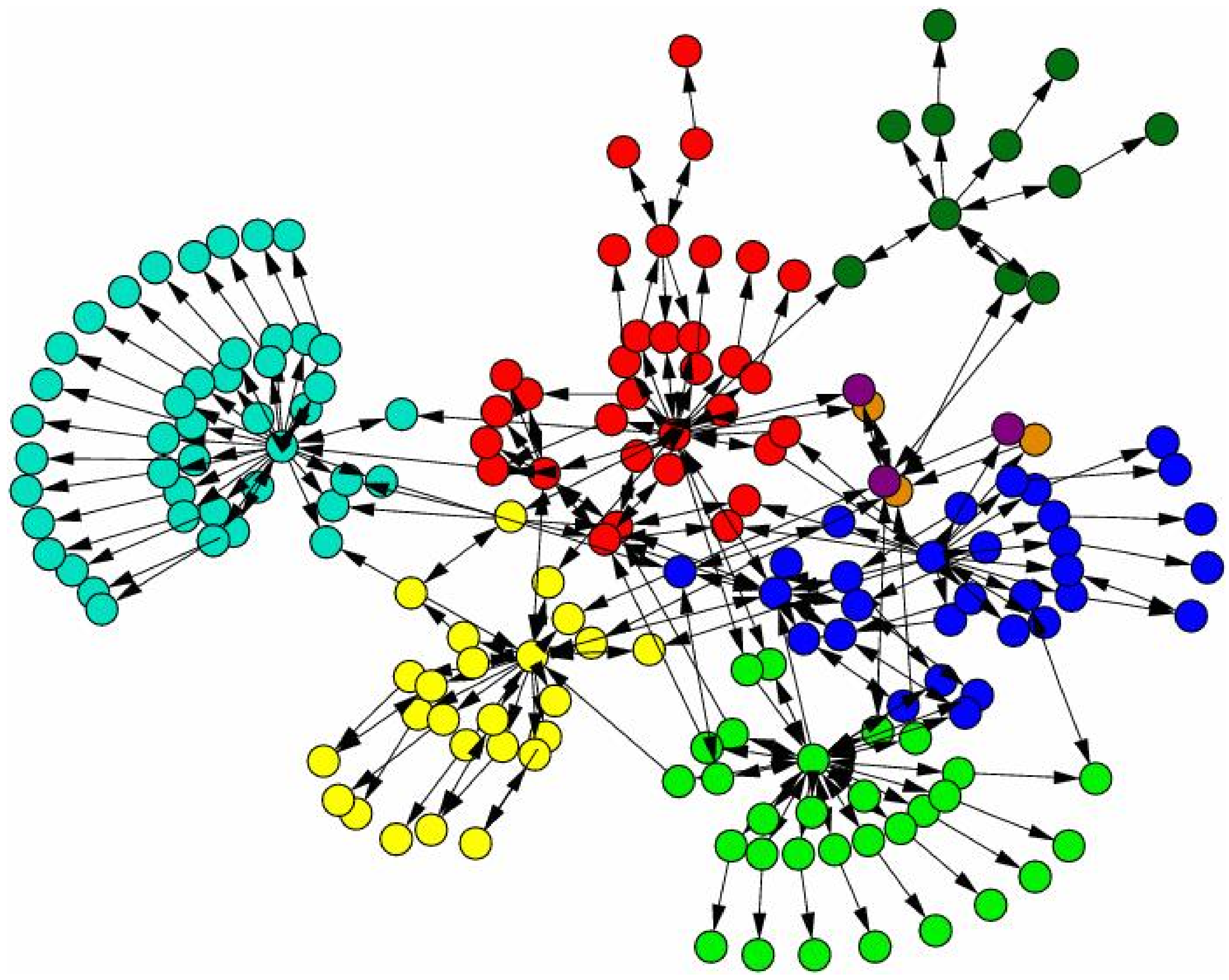
  – Can be a very slow process to search.

  – But dependent on network topology may be able to devise "quick" algorithms.

Web search

- Centralized warehousing of information (need results as quickly as possible)
- Key: Use information contained in the edges as well as the vertices! (Assumes edges contain information about relevance).
- Process: Query arrives, select subset of pages which match, order that subset by ranking based on link structure.

Typical web domain



M. E. J. Newman

Ranking pages in a connected component

- Each site starts with unit “rank” (i.e., weight).
- Each site transfers a fraction of this rank equally to each neighbor, in a discrete time process.
- So the rank of vertex i, r_i:

$$r_i \propto \sum_j A_{ij} \left(r_j / d_j \right) ,$$

where A is the adjacency matrix, and d_j degree of node j. This is r_j times the random walk dynamics on the network!

Ranking pages, cont.

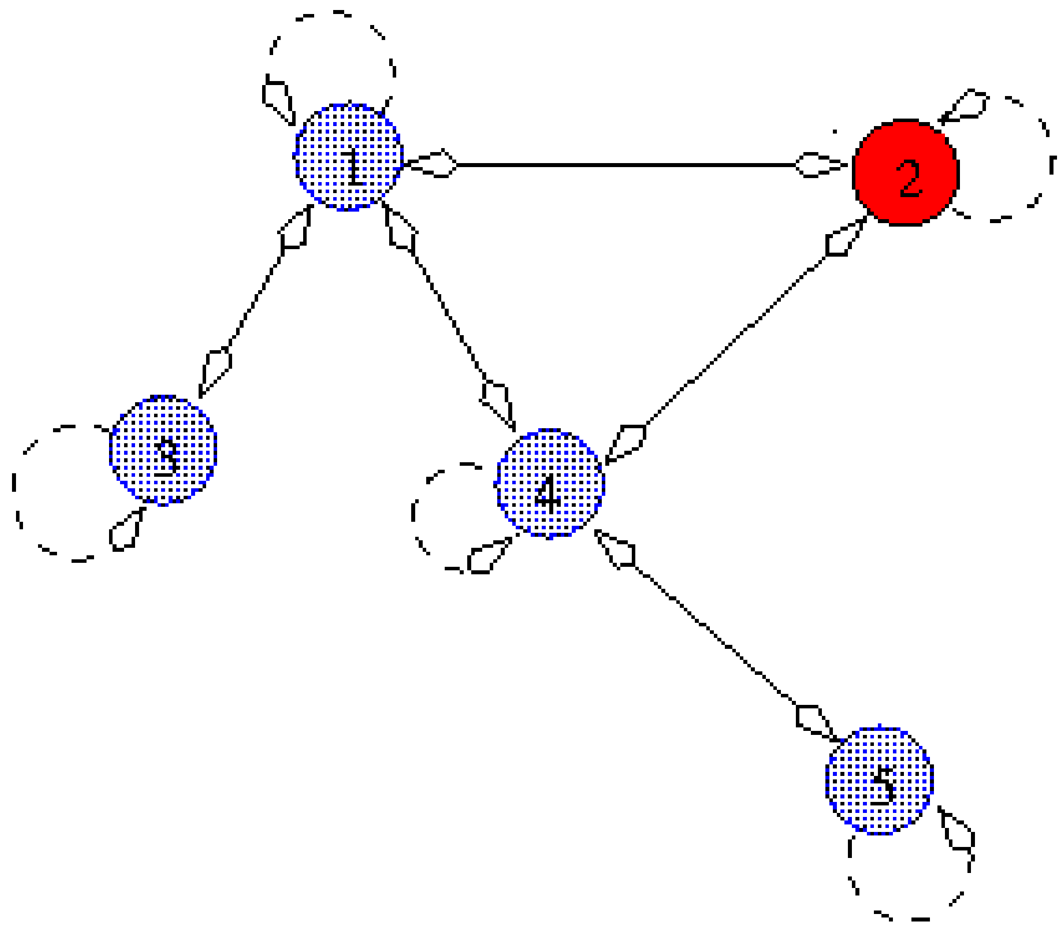
- The vector of ranks: $\vec{r} = [r_1, r_2, ...r_i, ...r_N]^{\mathrm{T}}$

- The dynamics: $\boxed{\vec{r} = M\vec{r}}$, where $M_{ij} \propto A_{ij}\,(r_j/d_j)$.

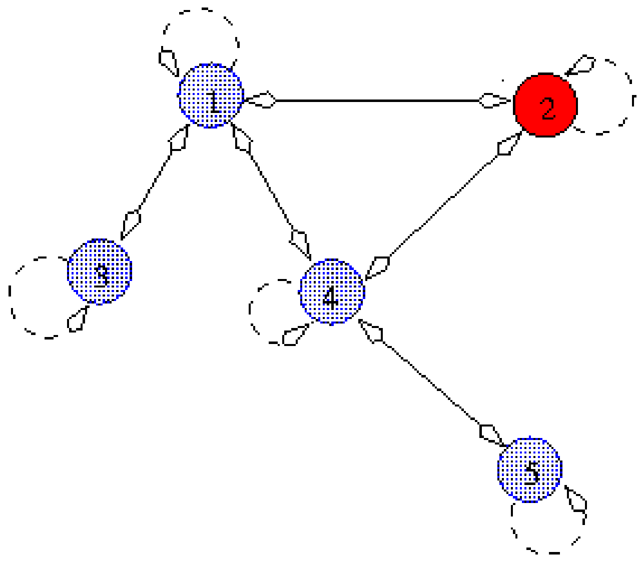
- Looking for a steady-state solution to the equation: $\boxed{\vec{r} = M\vec{r}}$, means finding eigenvector $\vec{r_1}$ with corresponding eigenvalue $\lambda_1 = 1$.

  Thus $\vec{r_1} = M\vec{r_1} = \lambda_1\vec{r_1} = \vec{r_1}$

Aside on Random Walks: Sample graph structure



Random walk: State Transition Matrix (Column-normalize the adjacency matrix)



$$M = \begin{pmatrix} 1/4 & 1/3 & 1/2 & 1/4 & 0 \\ 1/4 & 1/3 & 0 & 1/4 & 0 \\ 1/4 & 0 & 1/2 & 0 & 0 \\ 1/4 & 1/3 & 0 & 1/4 & 1/2 \\ 0 & 0 & 0 & 1/4 & 1/2 \end{pmatrix}$$

M will have a basis set of eigenvectors $\{\vec{u}_i\}$ and corresponding eigenvalues λ_i.

Perron-Frobenius Theorem

- Applies to **irreducible, positive, stochastic** matrices.
- "Irreducible" means cannot be block-diagonalized into disjoint pieces. (i.e., network is connected — only one component).
- "Positive" means each entry $M_{ij} > 0$.
- "Stochastic" means column normalized (or row normalized).

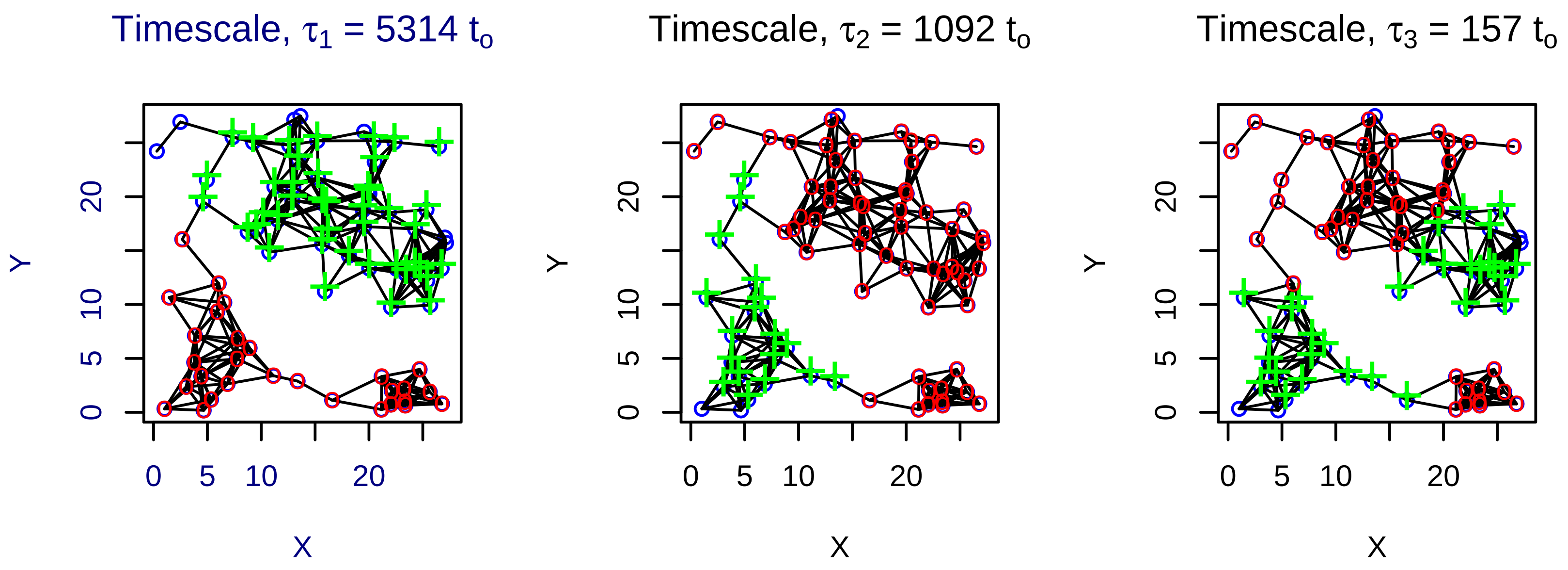
Perron-Frobenius Theorem
Leading eigenvalue

- One leading eigenvalue with $\lambda_1 = 1$.
- The corresponding eigenvector, v_1, has strictly positive entries and the sum over all the entries, $\sum_i v_1[i] = 1$.
- This is the stationary distribution of the random walk dynamics.

- For non-negative matrices ($M_{ij} \geq 0$), similar results, but:
  - Can't guarantee eigenvectors are positive (in practice, normally still works ... see R-code if interested : "transMatrix.R", "samp-matrices.R" and the "Session log").
  - The eigenvector entries are relatively normalized, but not absolutely normalized. (Need the extra renormalization step.)

Perron-Frobenius Theorem: Remaining eigenvalues ("Spectral partitioning" of graphs)

- All other eigenvalues are less than λ_1.
  (i.e., all $\lambda_i < 1$ for $i > 1$).
- These represent decaying modes that will eventually converge to the steady-state solution described by v_1.
- How long do they take to decay?



Decay / mixing times

- For each eigenmode, with eigenvector $\vec{v}_i$ and eigenvalue λ_i, $M\vec{v}_i = (\lambda_i)\,\vec{v}_i$ thus $M^t\vec{v}_i = (\lambda_i)^t\,\vec{v}_i$.

- The "relaxation time" for that mode τ_i: time for the original amplitude to decay to $1/e$. $\boxed{M^{\tau_i}\vec{v}_i = \frac{1}{e}\vec{v}_i \implies \tau_i = -1/\ln(\lambda_i)}$.

- Recall $\lambda_i < 1$ for all $i \geq 2$.

- So the biggest relaxation time is: $\tau_{\max} = \tau_2 = -1/\ln(\lambda_2)$.

- The difference $\lambda_1 - \lambda_2$ is called the *spectral gap*.

  ($\lambda_2 < 1$. Bigger λ_2 means smaller $|\ln(\lambda_2)|$, so larger τ_2.
  In other words, smaller spectral gap means *worse* mixing properties.)

Back to: Ranking pages in a single component

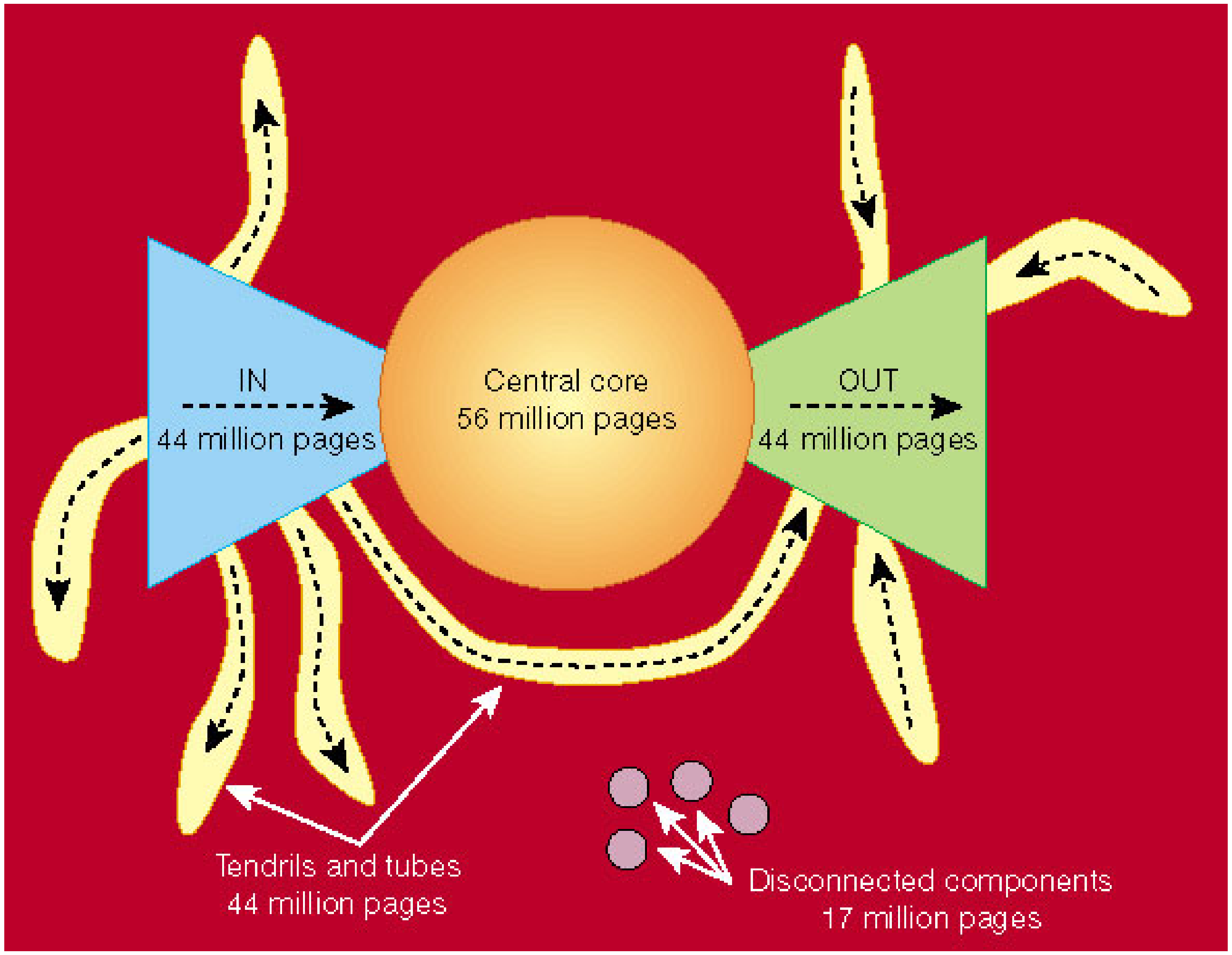
- Each site starts with unit “rank” (i.e., weight).
- Transfers fraction of this rank equally to each connected site.
- So at each iteration the rank of vertex i, r_i:

$$r_i = \sum_j A_{ij}(r_j/d_j) = \sum_j M_{ij} r_j$$

$$\texttt{More compactly:} \quad \vec{r} = M\vec{r}$$

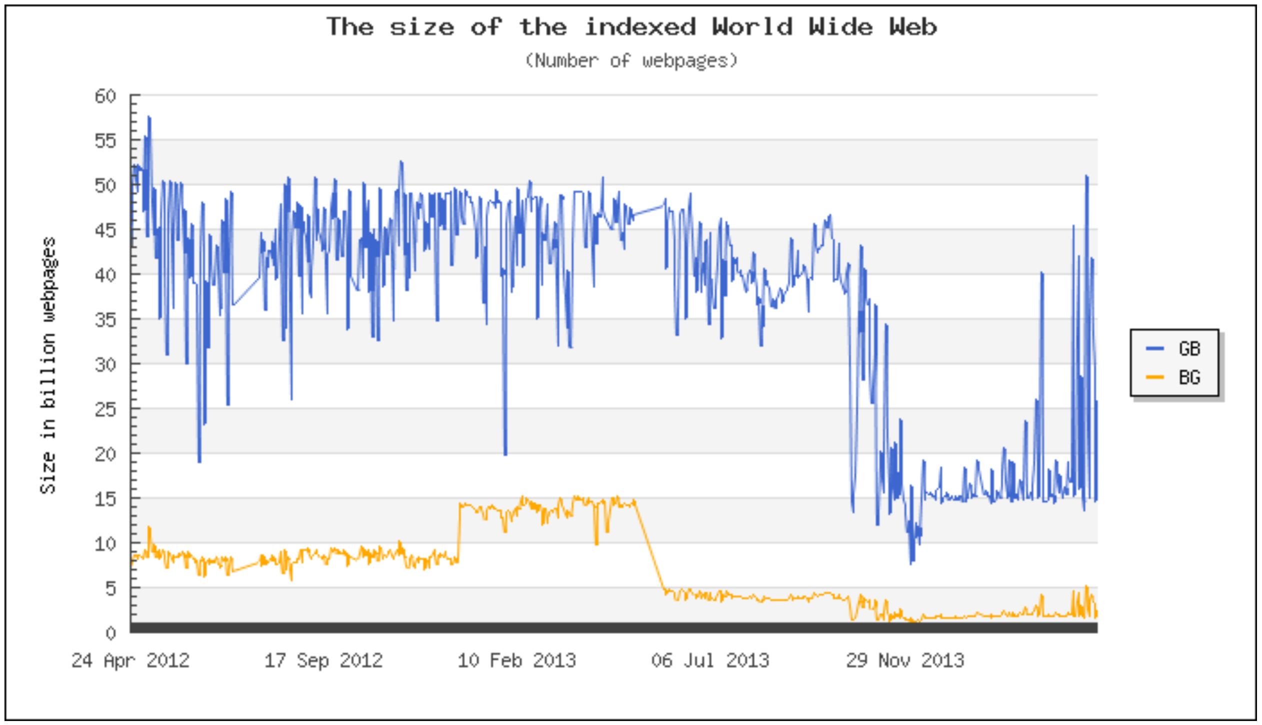
- Using a random walk formulation, the occupancy probabilities in steady-state (i.e., the vector corresponding the $\lambda = 1$): $\boxed{\vec{r} = M\vec{r}}$, where M is state transition matrix.

The Web as a whole



(This is an old picture, circa 2000)

Size of the *indexed* Web
2012-2013, see http://www.worldwidewebsize.com/

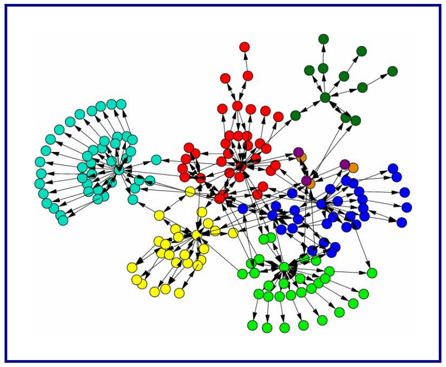


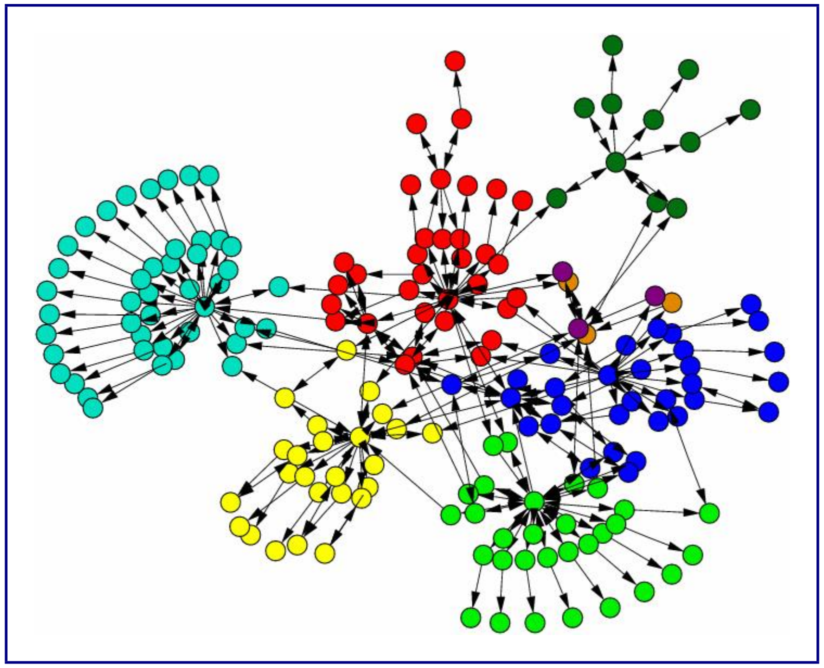
GB = Sorted on Google and Bing

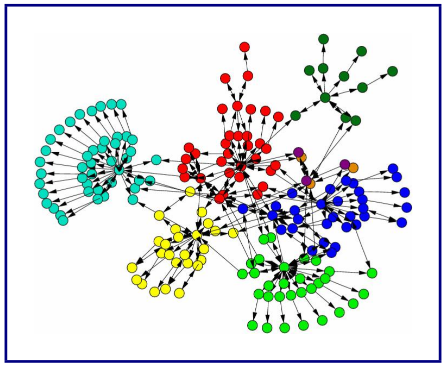
BG = Sorted on Bing and Google

- Neglects the “Dark Web” (e.g. subscriptions required)
- How much valid content is really out there?
- What is the size today?

What about disconnected components?







We understand how to deal with each components. How do we deal with getting a consistent rank across the whole web?

The “Random Surfer” model

[Brin and Page, “The anatomy of a large-scale hypertextual Web search engine”, *Computer Networks*, 30 (1998)]

[L. Page, S. Brin, R. Motwani, and T. Winograd, “The Pagerank Citation Ranking: Bringing Order to the web”, technical report, Stanford University, Stanford, CA, 1998.]

- With probability ϵ follow an out-link of current page.
- With probability $[1 - \epsilon]$ jump at random to some other web page. (Usually assume jump is random, so land at any site with prob $1/N$).

The “Random Surfer” model

The weight of a page j is the sum over all the in-links pointing to it, including those gained by the random jump:

$$r_j = \sum_{i \to j} \left\{ r_i \epsilon(i) + [1 - \epsilon(i)] J_{ij} \right\}.$$

(Note, this also makes the matrix positive, recall Perron-Frobenius).

<u>Rules of thumb (empirical):</u>

$$\epsilon(i) = \epsilon$$

$$J_{ij} = 1/N$$

$$\epsilon = 0.8.$$

Real-world complications: Page Rank

- Newer pages have less rank, even though they may be extremely relevant.
- Calculate eigenvalues of 10^9 by 10^9 matrix!!!
  (Many advances here due to Page Rank introduction)
- spam, spam, spam
- “Search engine optimizers” (reverse engineer search engines)
- Link farms (both invisible and visible)
- Selling highly ranked domain names
- delisting web sites

Real-world complications in general

- Stop words typically removed: and, of, the to, be, or.
  So how to handle query “to be or not to be”?

- Dealing with complications of multiple languages and cultures.

  – Chinese: no use of pronouns (he/she) instead repeat proper names (which is often a sign of low-quality page/spam in English).

  – Japanese: want breadth as well as depth (search 5 pages). US want a quick, definitive answer.

Alternate approaches – topology based

[Kleinberg and Lawrence, “The structure of the Web”, *Science*, 294 (2001)]

[Kleinberg, “Authoritative sources in a hyperlinked environment”, *J. ACM*, 46 (1999)]

- Slightly more sophisticated. Kleinberg proposes to use in-links and out-Links.
- Google assumes a page is important if other important pages point to it.
- Kleinberg identifies two kinds of importance: “hubs” and “authorities”

Hubs and authorities

- A page pointed to by highly ranked pages in an authority
- A page that points to highly ranked pages is a hub . (May not contain the information, but will tell you where to find it).

In use:

- Teoma search engine (http://search.ask.com/)
- Citeseer literature search engine.

Alternate approaches

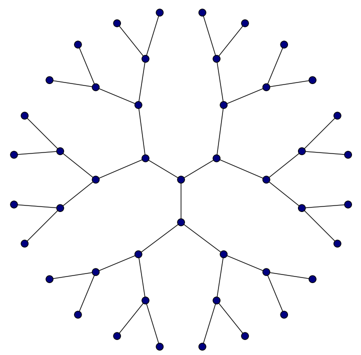
- Usage/click based.
- Negative edge weights: (Penalize spam linking to you.)
- Clustering results by subject, e.g., http://clusty.com, which became yippy.com, but is now defunct.
- . . .

Distributed search

Some resource of interest is stored at the vertices of a network:
i.e., Files in a file-sharing network

Search on arbitrary networks: $O(N)$ (search every node)

- Depth-first
- Breadth-first



Search on power-law random graphs

- Breadth-first passing always to highest-degree node possible. [Adamic, R. M. Lukose, A. R. Puniyani, and B. A. Huberman, "Search in power-law networks", Phys. Rev. E, 64 2001], find between $O(N^{2/3})$ and $O(N^{1/2})$.

Decentralized search on small-world networks

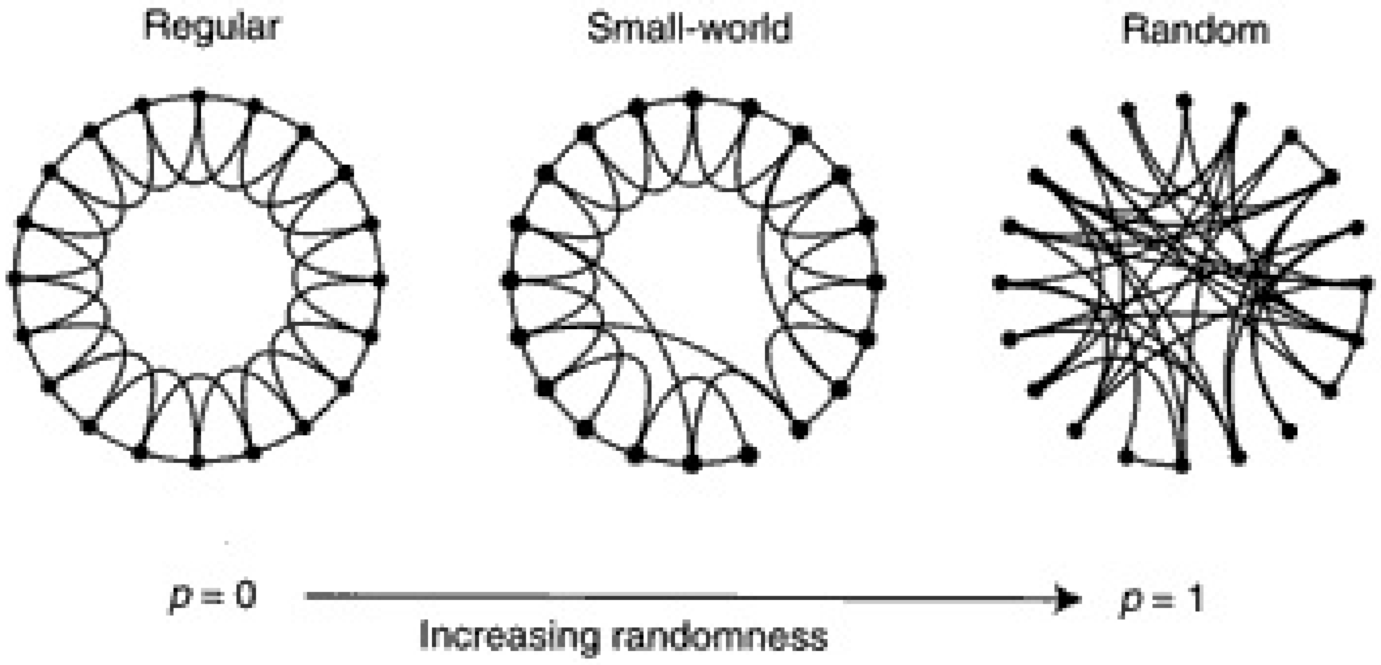
Consider a network with small diameter.

Will this help with decentralized search? (i.e. Find a local algorithm with performance less than $O(N)$?)

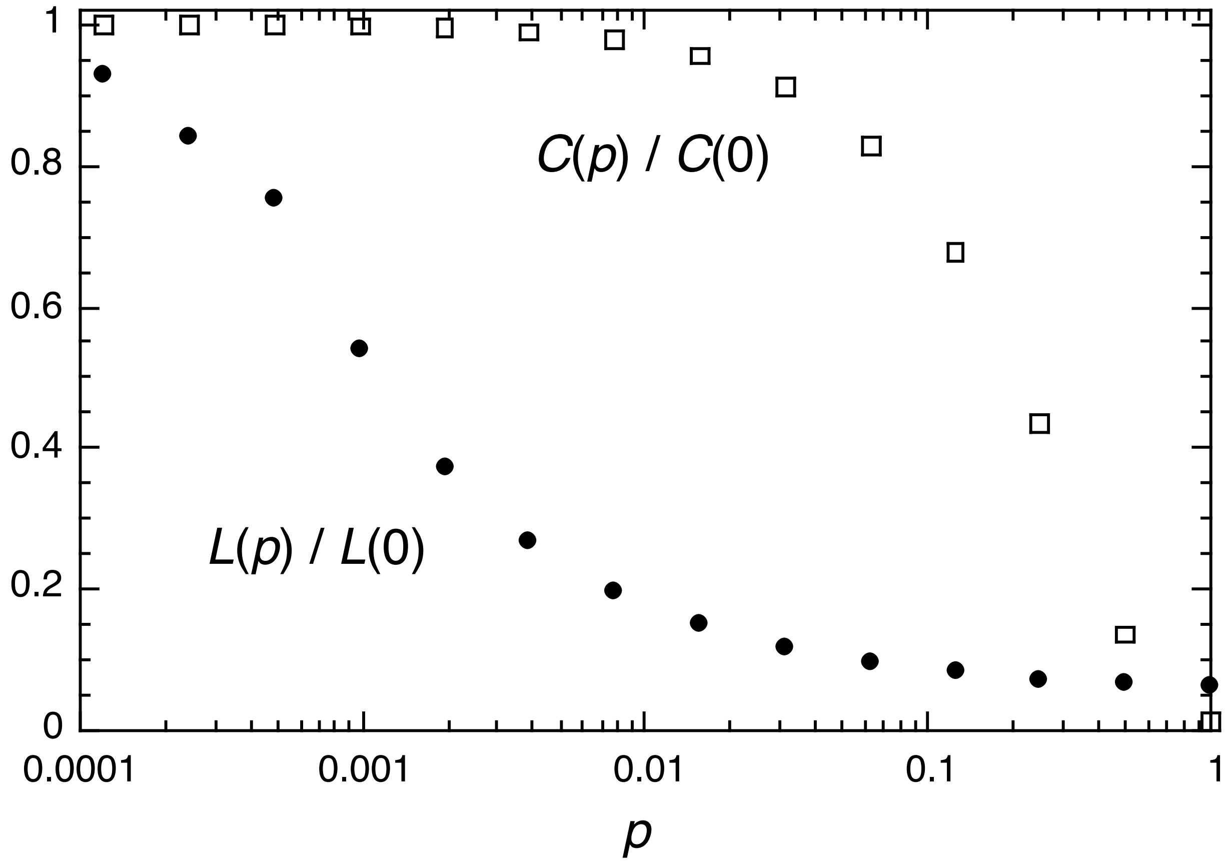
What is a small-world

[Watts and Strogatz, “Collective dynamics of ‘small-world’ networks”, *Nature*, 393 (1998)]

- Start with regular 1D lattice, with each node connected to it’s k nearest neighbors.
- Randomly re-wire each link independently with probability p.



Ave shortest path $L(p)$ and clustering coefficient $C(p)$

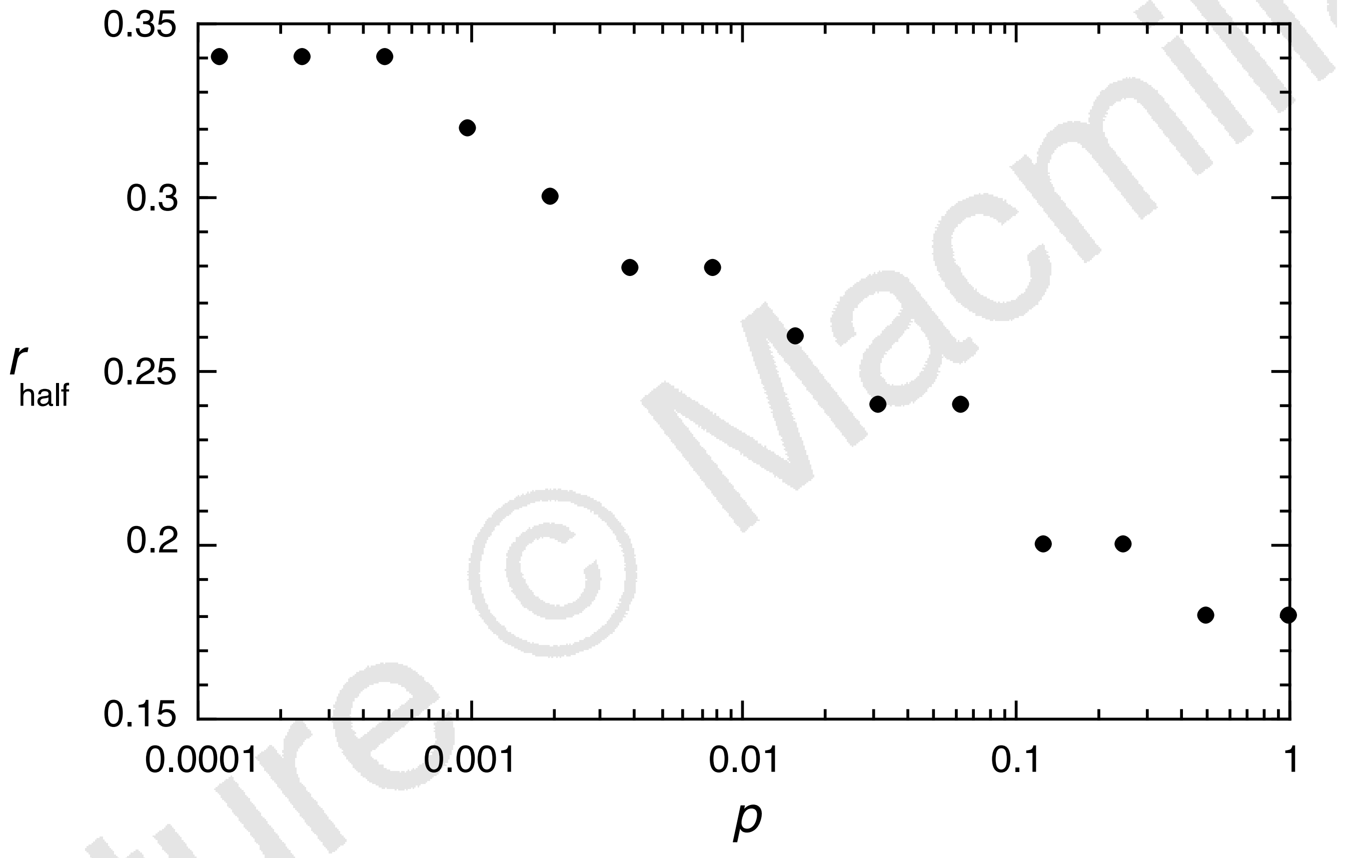


- Small-worlds have small diameter and large clustering coefficient.
- They are remarkably easy to generate (just a tiny p required).

Table 1 Empirical examples of small-world networks

	L_{actual}	L_{random}	C_{actual}	C_{random}
Film actors	3.65	2.99	0.79	0.00027
Power grid	18.7	12.4	0.080	0.005
C. elegans	2.65	2.25	0.28	0.05

Infection rates on Small-worlds



- Start with *one* infected individual.
- r is transmission rate of disease (infect neighbors at rate r).
- r_{half} is the the value of r required for half the population to get the disease.

Their concluding words

“We hope that our work will stimulate further studies of small-world networks. Their distinctive combination of high clustering with short characteristic path length cannot be captured by traditional approximations such as those based on regular lattices or random graphs. Although small-world architecture has not received much attention, we suggest that it will probably turn out to be widespread in biological, social and man-made systems,often with important dynamical consequences.”

Watts-Strogatz small-world model

- Together with Barabasi-Albert launched the flurry of activity on networks.
- Watts and Strogatz showed that networks from both the natural and manmade world, such as the neural network of C. elegans and power grids, exhibit the small-world property.
- Originally they wanted to understand the synchronization of cricket chirps.
- Introduced the mathematical formalism, which *interpolates between lattices and networks* .

A new paradigm

”I think I’ve been contacted by someone from just about every field outside of English literature. I’ve had letters from mathematicians, physicists, biochemists, neurophysiologists, epidemiologists, economists, sociologists; from people in marketing, information systems, civil engineering, and from a business enterprise that uses the concept of the small world for networking purposes on the Internet.” – Duncan Watts

Navigation

Clearly if central coordination, can use short paths to deliver info quickly.

But, can someone living in a small world actually make use of this info and do efficient decentralized routing?

- Instead of designing search algorithms, given a local greedy algorithm, are there any topologies that enable $O(\log N)$ delivery times?

Precise topologies required

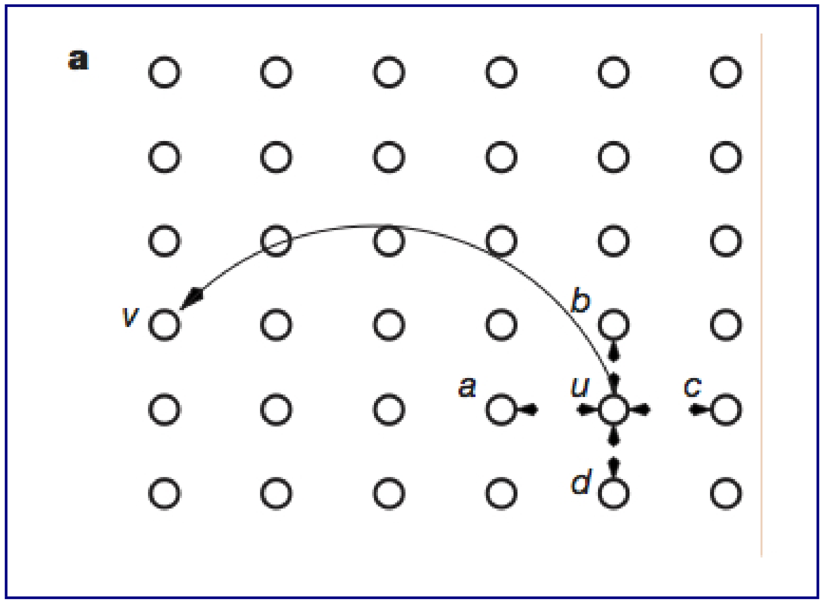
[J. M. Kleinberg, “Navigation in a small world”, *Nature*, 406 (2000)]

- Start with a regular 2D square lattice (consider vertices and edges).

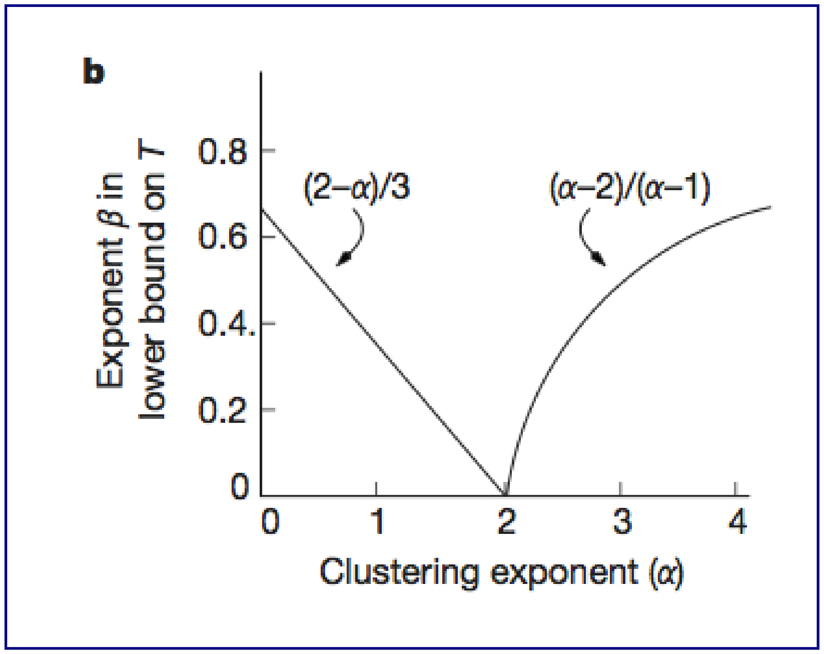
- Add random long links, with bias proportional to distance between two nodes:

$$\boxed{p(e_{ij}) \propto 1/d_{ij}^{\alpha}}$$

- Call α the “clustering exponent”


a
v
b
a
u
c
d

- Find mean delivery time $t \sim N^{\beta}$, unless $\alpha = 2$.
- Only for $\alpha = 2$ will decentralized routing work, and packet can go from source to destination in $O(\log N)$ steps.



- For d-dimensional lattice need $\alpha = d$.

But we know greedy decentralized routing works for human networks (c.f. Milgram's experiments “six-degrees of separation” [S. Milgram, “The small world problem”, *Psych. Today*, 2, 1967.]

So how do we get beyond a lattice model?

Navigating social networks

[Watts, P. S. Dodds, and M. E. J. Newman, “Identity and search in social networks”, Science, 296 (2002)]

[Kleinberg, “Small world phenomena and the dynamics of information”, in Proceedings of NIPS 2001].

- Premise: people navigate social networks by looking for common features between there acquaintances and the targets (occupation, city inhabited, age, ...)
- Brings in DATA!

Hierarchical “social distance” tree

- Individuals are grouped into categories along many attributes.
- One tree for each attribute.
- Trees are not the network, but complementary mental constructs believed to be at work.
- Assume likelyhood of acquaintance falls of exponentially with “social distance”.

Building P2P architectures

- “Chord A Scalable Peer-to- peer Lookup Service for Internet Applications

  I. Stoica, R. Morris, D. Karger, F. Kaashoek, H. Balakrishnan, ACM SIGCOMM, 2001.
  (Cited 15,215 times)

- Gnutella

  “Peer-to-Peer Architecture Case Study: Gnutella Network”, M Ripeanu, Proceedings of International Conference on Peer-to-peer Computing, 2001.

Summary

Web search

- Centralized
- Make use of link structure (topology)

Decentralized search

- Efficiency/Speed depends on underlying topology
- Gossip algorithms (D. Kempe and J. Kleinberg): spreading shared information quickly through local exchanges (e.g., sums, local averages/consensus).
- Applications to sensor networks .... communications ... satellites