Basic Network Properties and the Random Graph Model

CS224W: Social and Information Network Analysis

Jure Leskovec, Stanford University

http://cs224w.stanford.edu

Announcement: Recitations

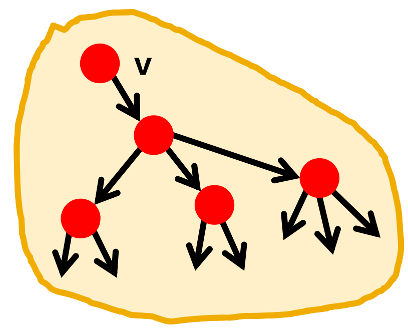
- **Intro sessions to SNAP C++ and SNAP.PY:**
  - **SNAP.PY:** Friday 9/27, 4:15-5:30pm in Gates B03
  - **SNAP C++:** Thursday 10/3, 4:15-5:30pm in Gates B03
  - **Sessions will be recorded and available via SCPD**
- **About the software libraries:**
  - TAs support SNAP C++ (Justin, Bell), SNAP.PY (Christie, Yoni)
  - You can use other libraries: NetworkX, JUNG, Boost, R
    - They will do the job but we don't offer support for them
  - Start early on HW0 since these packages are new to you, complex and non-trivial to use!
- **Review of:**
  - **Probability:** Friday, 10/4, 4:15-5:30pm in Gates B03
  - **Linear algebra**: Tuesday, 10/8, 2:15-3:30pm,Gates B03

How the Class Fits Together

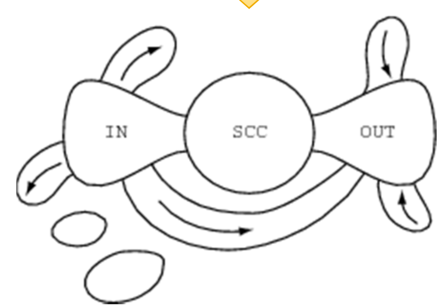
Observations	Models	Algorithms
Small diameter, Edge clustering	Erdös-Renyi model, Small-world model	Decentralized search
Patterns of signed edge creation	Structural balance, Theory of status	Models for predicting edge signs
Viral Marketing, Blogosphere, Memetracking	Independent cascade model, Game theoretic 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

Structure of Networks

- **For example, last time we talked about Observations and Models for the Web graph:**
  - **1)** We took a real system: **the Web**
  - **2)** We represented it as a **directed graph**
  - **3)** We used the language of graph theory
    - **Strongly Connected Components**
  - **4)** We designed a **computational experiment:**
    - Find In- and Out-components of a given node v
  - **5) We learned something about the structure of the Web: BOWTIE!**



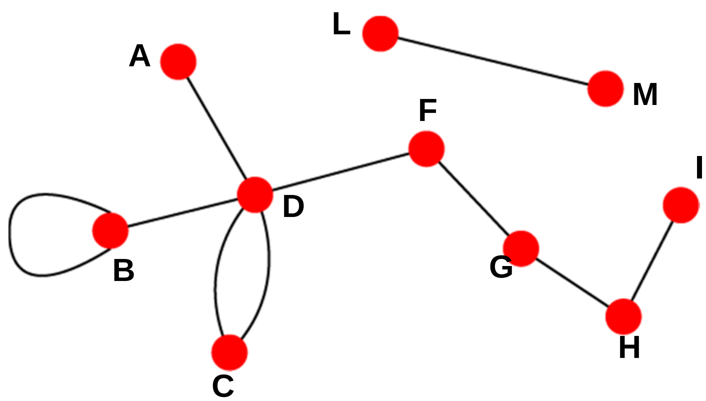
Out(v)



Undirected vs. Directed Networks

Undirected graphs

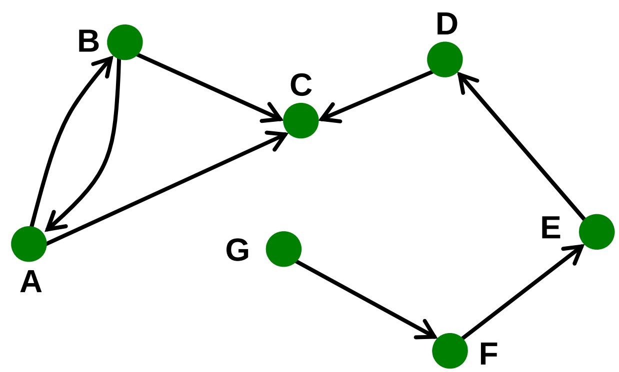
- **Links:** undirected (symmetrical, reciprocal relations)



- Undirected links:
  - Collaborations
  - Friendship on Facebook

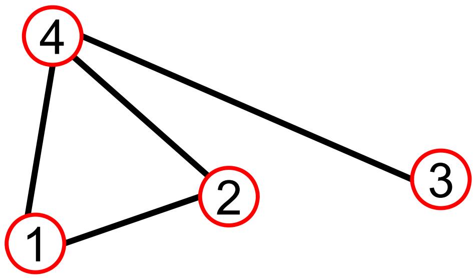
Directed graphs

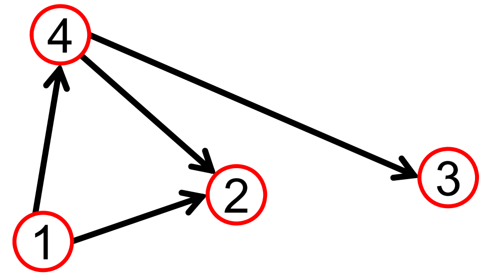
- **Links:** directed (asymmetrical relations)



- Directed links:
  - Phone calls
  - Following on Twitter

Adjacency Matrix





$A_{ij} = 1$ if there is a link from node ***i*** to node ***j***

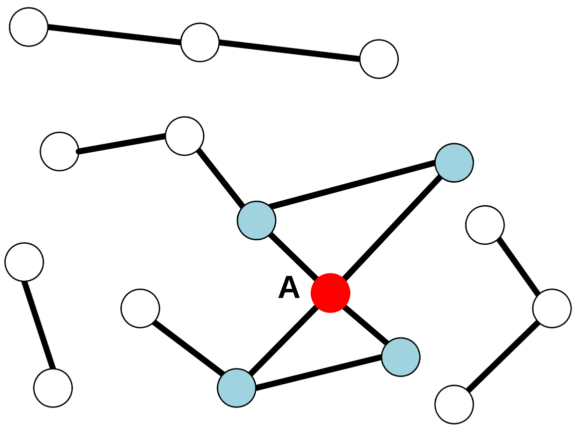
$A_{ij} = 0$ otherwise

$$A = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} \qquad A = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

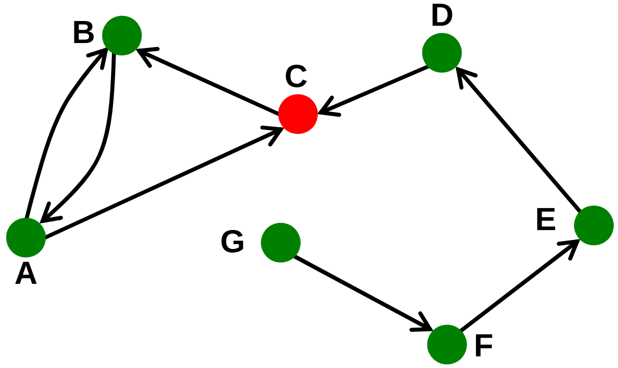
Note that for a directed graph (right) the matrix is not symmetric.

Node Degrees

Undirected



Directed



Source: node with $k^{in} = 0$

Sink: node with $k^{out} = 0$

Node degree, k_i: the number of edges adjacent to node $\boldsymbol{i}$

$$k_A = 4$$

Avg. degree: $\bar{k} = \langle k \rangle = \frac{1}{N} \sum_{i=1}^{N} k_i = \frac{2E}{N}$

In directed networks we define an **in-degree** and **out-degree.**

The (total) degree of a node is the sum of in- and out-degrees.

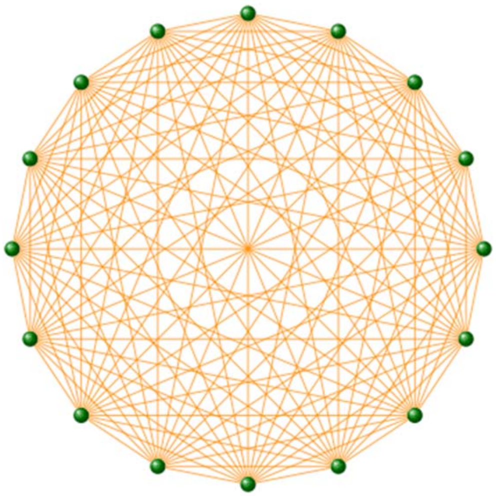
$$k_C^{in} = 2 \qquad k_C^{out} = 1 \qquad k_C = 3$$

$$\bar{k} = \frac{E}{N} \qquad \overline{k^{in}} = \overline{k^{out}}$$

Complete Graph

The **maximum number of edges** in an undirected graph on N nodes is

$$E_{\max} = \binom{N}{2} = \frac{N(N-1)}{2}$$



A graph with the number of edges $E = E_{max}$ is a **complete graph**, and its average degree is $N\text{-}1$

Networks are Sparse Graphs

Most real-world networks are sparse

$$E \ll E_{max} \quad (\text{or } \bar{k} \ll N-1)$$

WWW (Stanford-Berkeley):	N=319,717	$\langle k \rangle$=9.65
Social networks (LinkedIn):	N=6,946,668	$\langle k \rangle$=8.87
Communication (MSN IM):	N=242,720,596	$\langle k \rangle$=11.1
Coauthorships (DBLP):	N=317,080	$\langle k \rangle$=6.62
Internet (AS-Skitter):	N=1,719,037	$\langle k \rangle$=14.91
Roads (California):	N=1,957,027	$\langle k \rangle$=2.82
Proteins (S. Cerevisiae):	N=1,870	$\langle k \rangle$=2.39

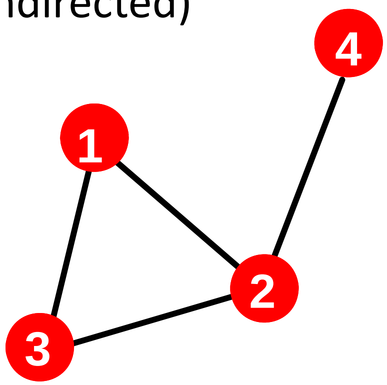
(Source: *Leskovec et al., Internet Mathematics, 2009*)

Consequence: Adjacency matrix is filled with zeros!

(**Density of the matrix (E/N^2):** WWW=1.51×10^{-5}, MSN IM = 2.27×10^{-8})

More Types of Graphs:

- **Unweighted**
  (undirected)



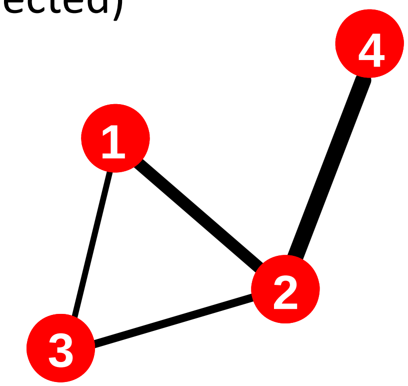
$$A_{ij} = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$$

$$A_{ii} = 0 \qquad A_{ij} = A_{ji}$$

$$E = \frac{1}{2}\sum_{i,j=1}^{N} A_{ij} \qquad \overline{k} = \frac{2E}{N}$$

Examples: Friendship, Hyperlink

- **Weighted**
  (undirected)



$$A_{ij} = \begin{pmatrix} 0 & 2 & 0.5 & 0 \\ 2 & 0 & 1 & 4 \\ 0.5 & 1 & 0 & 0 \\ 0 & 4 & 0 & 0 \end{pmatrix}$$

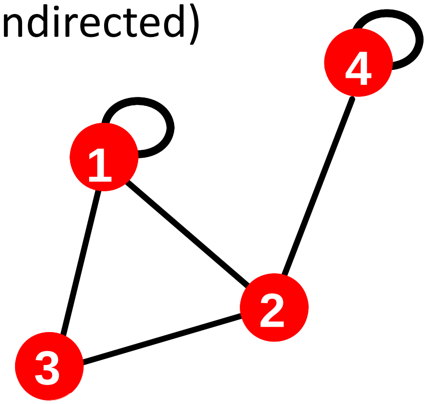
$$A_{ii} = 0 \qquad A_{ij} = A_{ji}$$

$$E = \frac{1}{2}\sum_{i,j=1}^{N} nonzero(A_{ij}) \qquad \overline{k} = \frac{2E}{N}$$

Examples: Collaboration, Internet, Roads

More Types of Graphs:

- **Self-edges (self-loops)**
  (undirected)



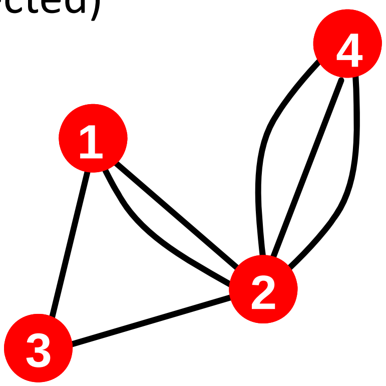
$$A_{ij} = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \end{pmatrix}$$

$$A_{ii} \neq 0 \qquad A_{ij} = A_{ji}$$

$$E = \frac{1}{2} \sum_{i,j=1, i \neq j}^{N} A_{ij} + \sum_{i=1}^{N} A_{ii} \qquad ?$$

Examples: Proteins, Hyperlink

- **Multigraph**
  (undirected)



$$A_{ij} = \begin{pmatrix} 0 & 2 & 1 & 0 \\ 2 & 0 & 1 & 3 \\ 1 & 1 & 0 & 0 \\ 0 & 3 & 0 & 0 \end{pmatrix}$$

$$A_{ii} = 0 \qquad A_{ij} = A_{ji}$$

$$E = \frac{1}{2} \sum_{i,j=1}^{N} nonzero(A_{ij}) \qquad \bar{k} = \frac{2E}{N}$$

Examples: Communication, Collaboration

Network Representations

WWW >> directed multigraph with self-edges

Facebook friendships >> undirected, unweighted

Citation networks >> unweighted, directed, acyclic

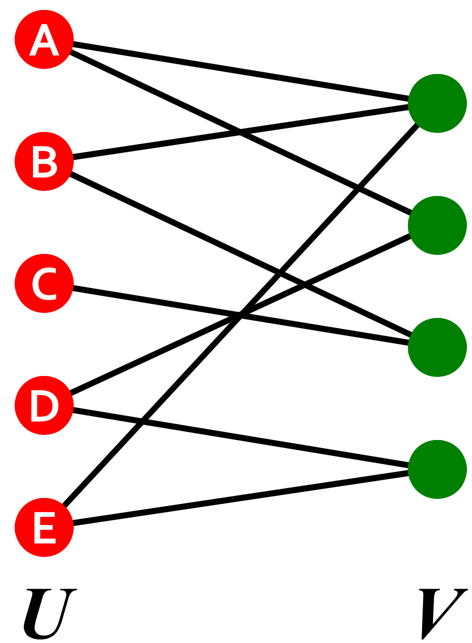
Collaboration networks >> undirected multigraph or weighted graph

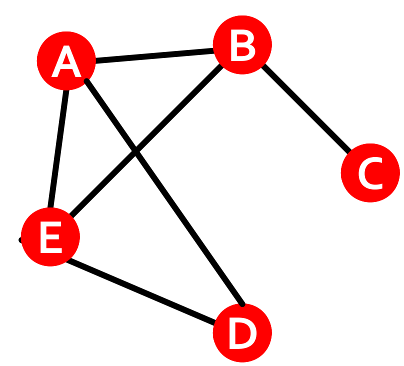
Mobile phone calls >> directed, (weighted?) multigraph

Protein Interactions >> undirected, unweighted with self-interactions

Bipartite Graph

- **Bipartite graph** is a graph whose nodes can be divided into two disjoint sets U and V such that every link connects a node in U to one in V; that is, U and V are **independent sets**
- **Examples:**
  - Authors-to-papers (they authored)
  - Actors-to-Movies (they appeared in)
  - Users-to-Movies (they rated)
- **"Folded" networks:**
  - Author collaboration networks
  - Movie co-rating networks





Folded version of the graph above

Network Properties: How to Characterize/Measure a Network?

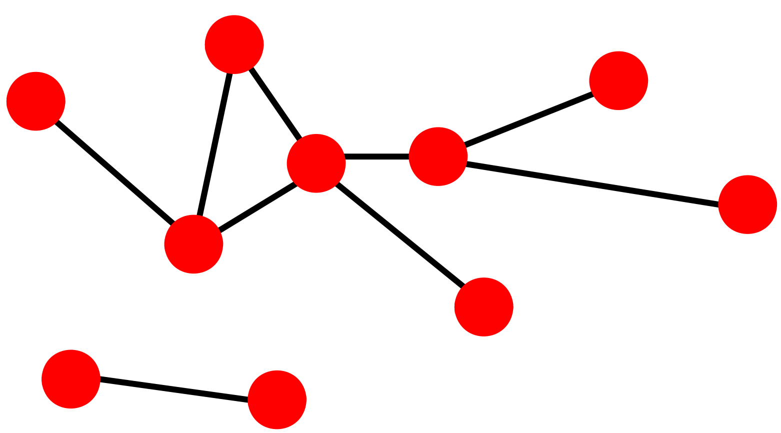
Degree Distribution

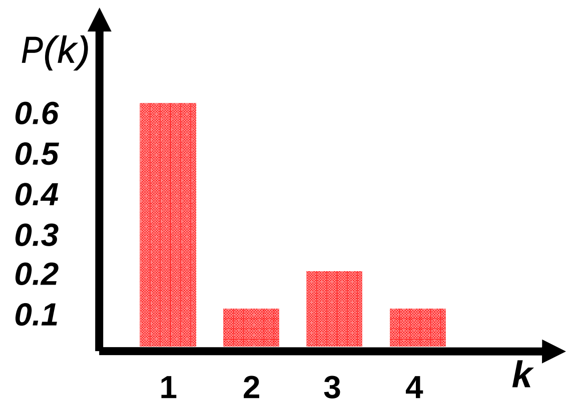
- **Degree distribution** $P(k)$: Probability that a randomly chosen node has degree k

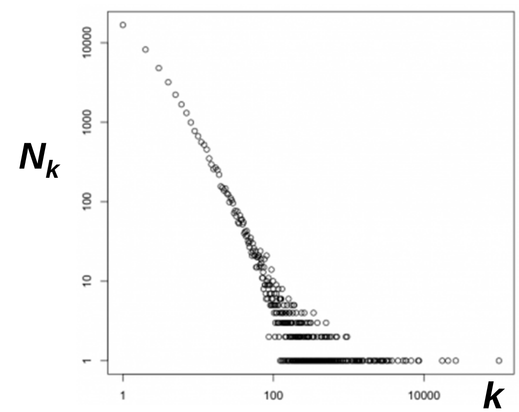
  N_k = # nodes with degree k

- Normalized histogram:

  $P(k) = N_k / N$ → **plot**





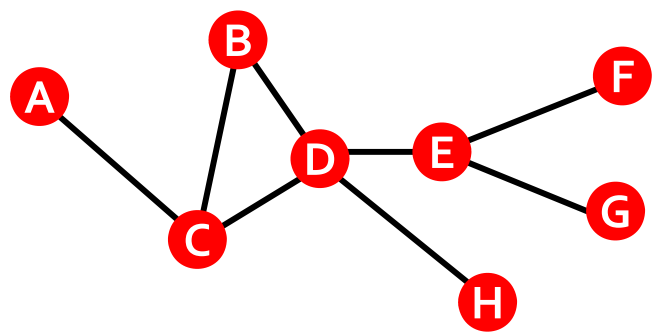


Paths in a Graph

- A ***path*** is a sequence of nodes in which each node is linked to the next one

$$P_n = \{i_0, i_1, i_2, \ldots, i_n\} \qquad P_n = \{(i_0, i_1), (i_1, i_2), (i_2, i_3), \ldots, (i_{n-1}, i_n)\}$$

- Path can intersect itself and pass through the same edge multiple times
  - E.g.: ACBDCDEG
  - In a directed graph a path can only follow the direction of the "arrow"



EXTRA: Number of Paths

- **Number of paths between nodes u and v :**
  - **Length $h=1$:** If there is a link between u and v, $A_{uv}=1$ else $A_{uv}=0$
  - **Length $h=2$:** If there is a path of length two between u and v then $A_{uk}A_{kv}=1$ else $A_{uk}A_{kv}=0$

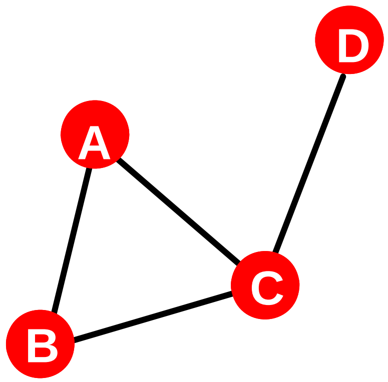
$$H_{uv}^{(2)} = \sum_{k=1}^{N} A_{uk} A_{kv} = [A^2]_{uv}$$

  - **Length h:** If there is a path of length h between u and v then $A_{uk} \dots A_{kv}=1$ else $A_{uk} \dots A_{kv}=0$
    So, the no. of paths of length h between u and v is

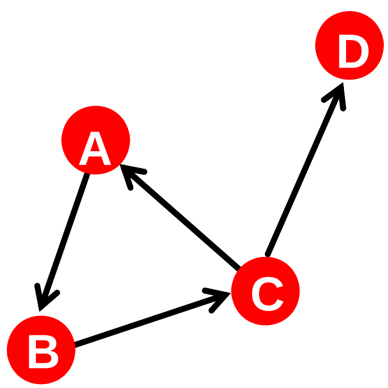
$$H_{uv}^{(h)} = [A^h]_{uv}$$

(holds for both directed and undirected graphs)

Distance in a Graph



$h_{B,D} = 2$



$h_{B,C} = 1$, $h_{C,B} = 2$

- **Distance (shortest path, geodesic)** between a pair of nodes is defined as the number of edges along the shortest path connecting the nodes
  - *If the two nodes are disconnected, the distance is usually defined as infinite
- In **directed graphs** paths need to follow the direction of the arrows
  - Consequence: Distance is **not symmetric**: $h_{A,C} \neq h_{C,A}$

Network Diameter

- **Diameter:** the maximum (shortest path) distance between any pair of nodes in a graph

- **Average path length** for a connected graph (component) or a strongly connected (component of a) directed graph

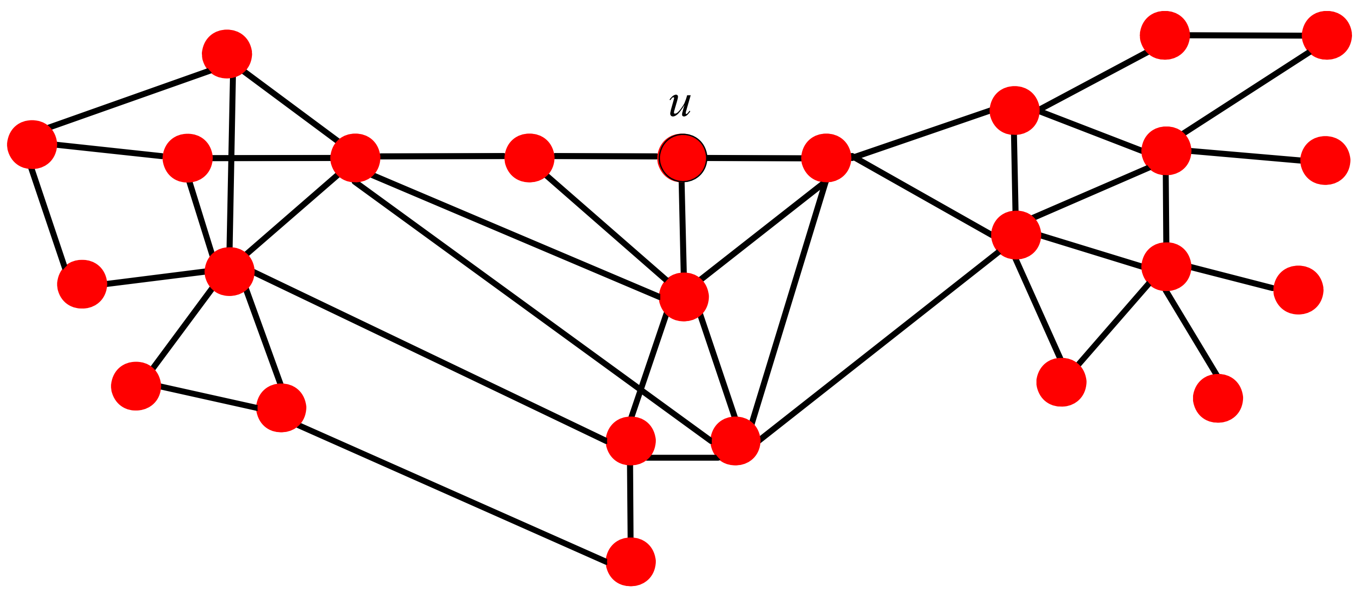
$$\bar{h} = \frac{1}{2E_{\max}} \sum_{i,j \neq i} h_{ij}$$

where h_{ij} is the distance from node i to node j

  - Many times we compute the average only over the connected pairs of nodes (we ignore "infinite" length paths)

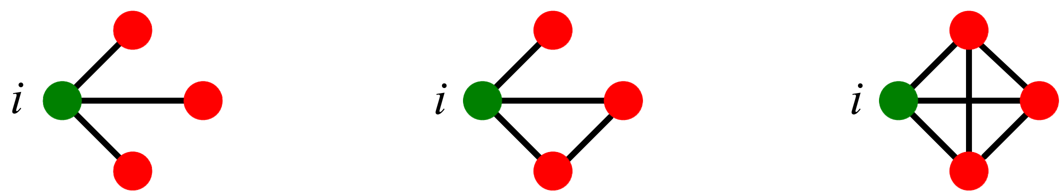
Finding Shortest Paths

- **Breath-First Search:**
  - Start with node u, mark it to be at distance $h_u(u)=0$, add u to the queue
  - While the queue not empty:
    - Take node v off the queue, put its unmarked neighbors w into the queue and mark $h_u(w)=h_u(v)+1$



Clustering Coefficient

- **Clustering coefficient:**
  - What portion of ***i***'s neighbors are connected?
  - Node ***i*** with degree $\boldsymbol{k_i}$
  - $C_i \in [0,1]$
  - $C_i = \dfrac{2e_i}{k_i(k_i - 1)}$ where e_i is the number of edges between the neighbors of node i

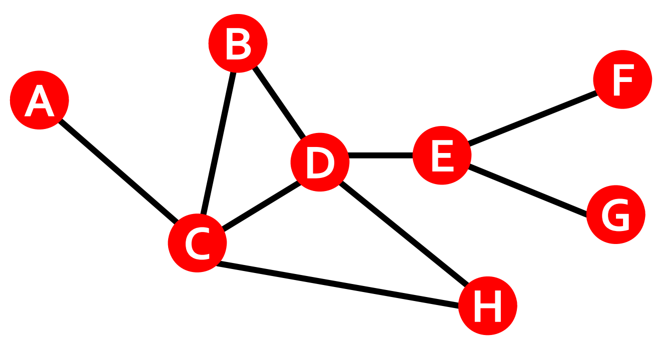


C_i=0 C_i=1/3 C_i=1

- **Average Clustering Coefficient:** $C = \dfrac{1}{N}\sum_i^N C_i$

Clustering Coefficient

- **Clustering coefficient:**
  - What portion of i's neighbors are connected?
  - Node i with degree k_i
  - $C_i = \dfrac{2e_i}{k_i(k_i - 1)}$ where e_i is the number of edges between the neighbors of node i



$k_B=2,\ e_B=1,\ C_B=2/2 = 1$

$k_D=4,\ e_D=2,\ C_D=4/12 = 1/3$

Key Network Properties

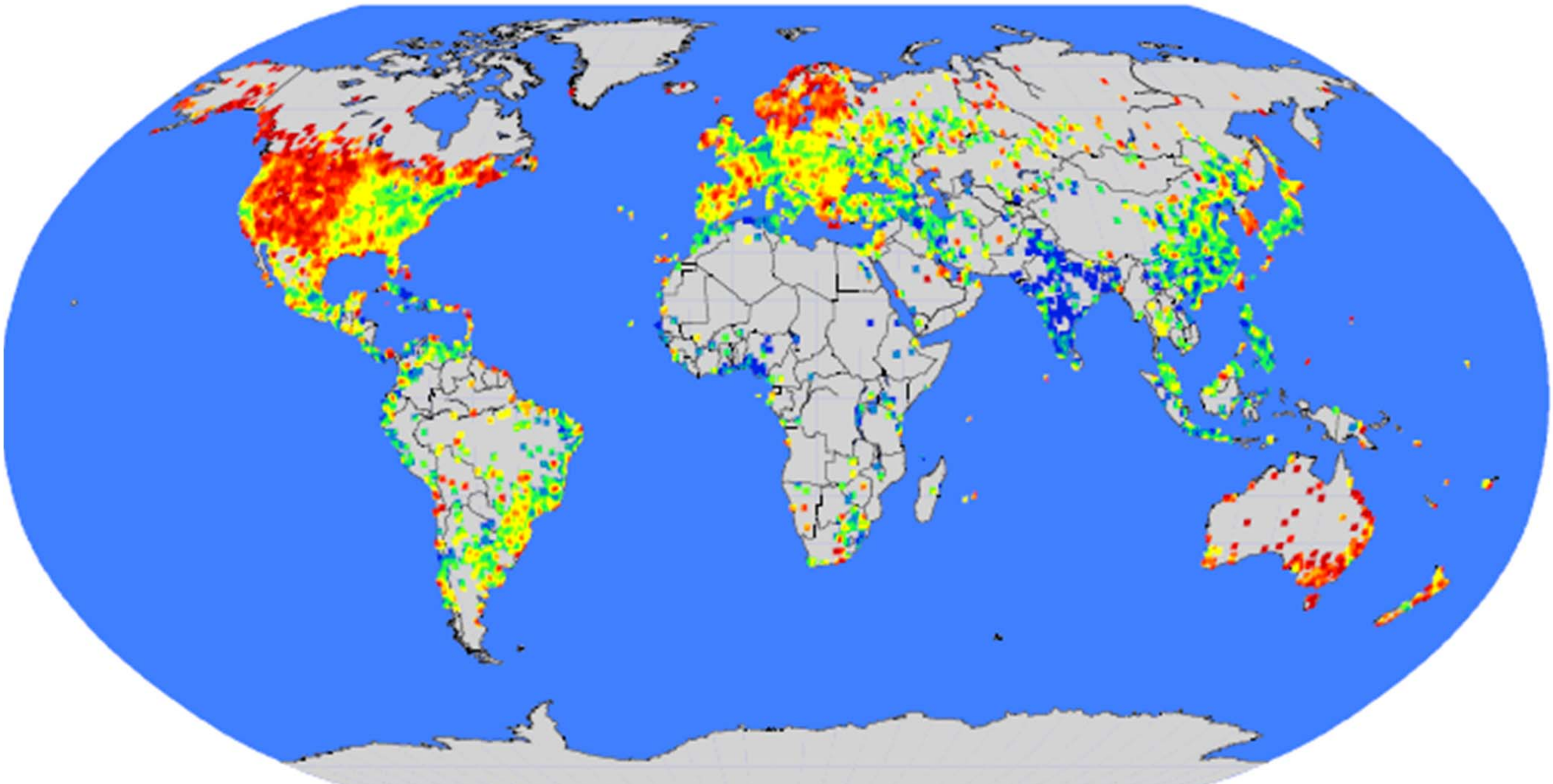
Degree distribution:	$P(k)$
Path length:	h
Clustering coefficient:	C

Let's measure P(k), h and C on a real-world network!

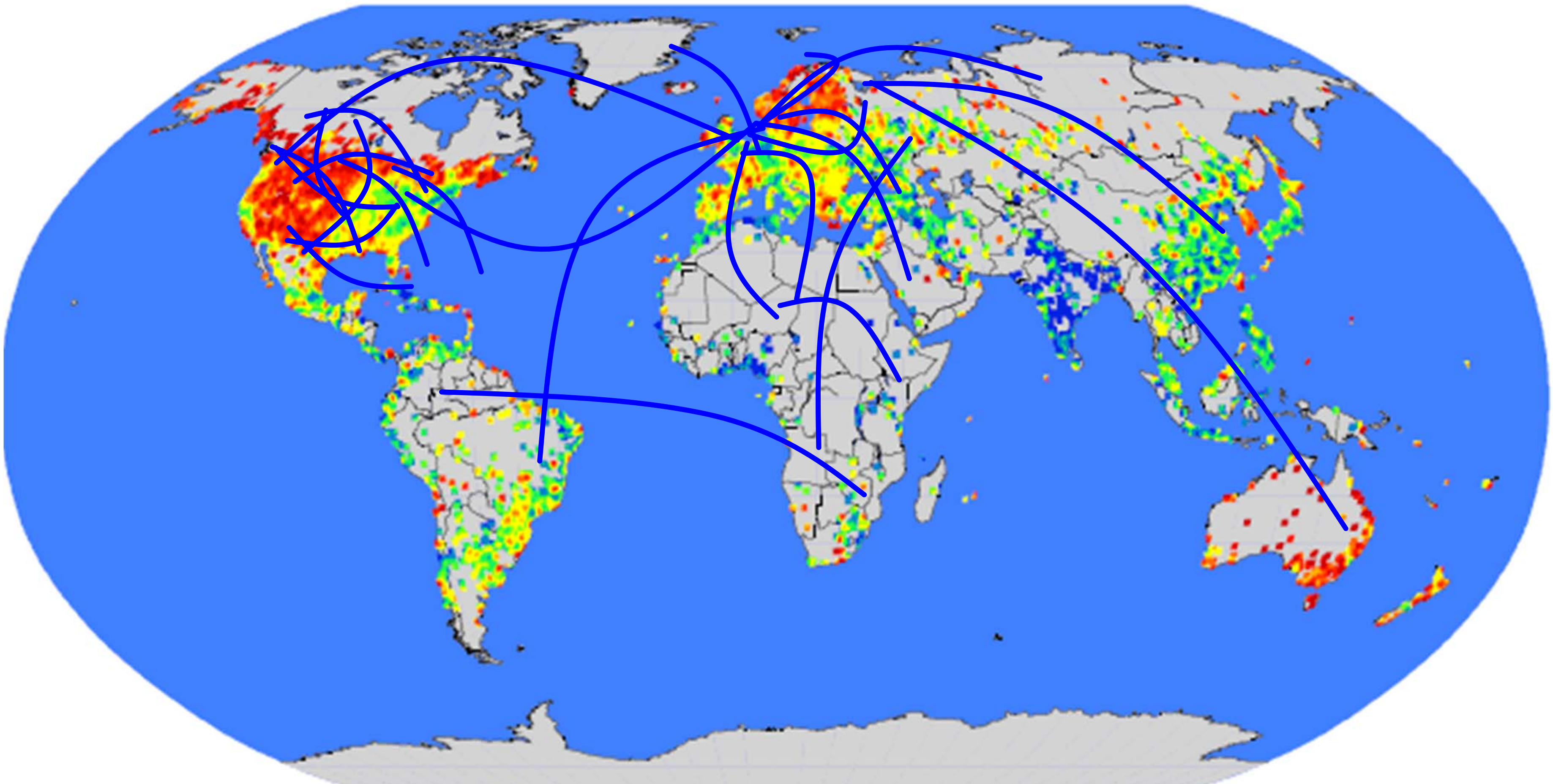
The MSN Messenger

- **MSN Messenger activity in June 2006:**
  - 150Gb/day (compressed)
  - 4.5Tb / month
  - 245 million users logged in
  - 180 million users engaged in conversations
  - More than 30 billion conversations
  - More than 255 billion exchanged messages

Communication: Geography

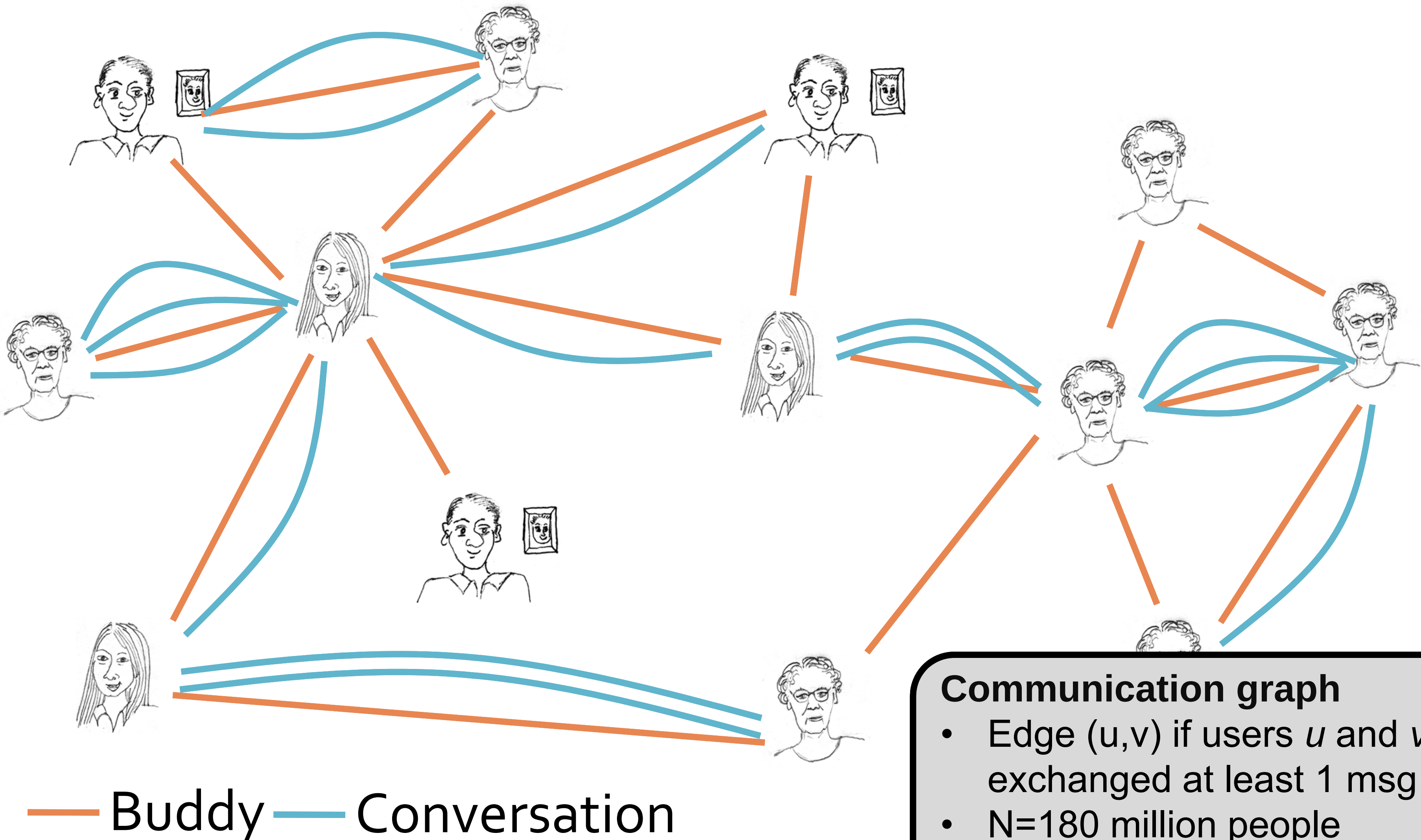


Communication network



Network: 180M people, 1.3B edges

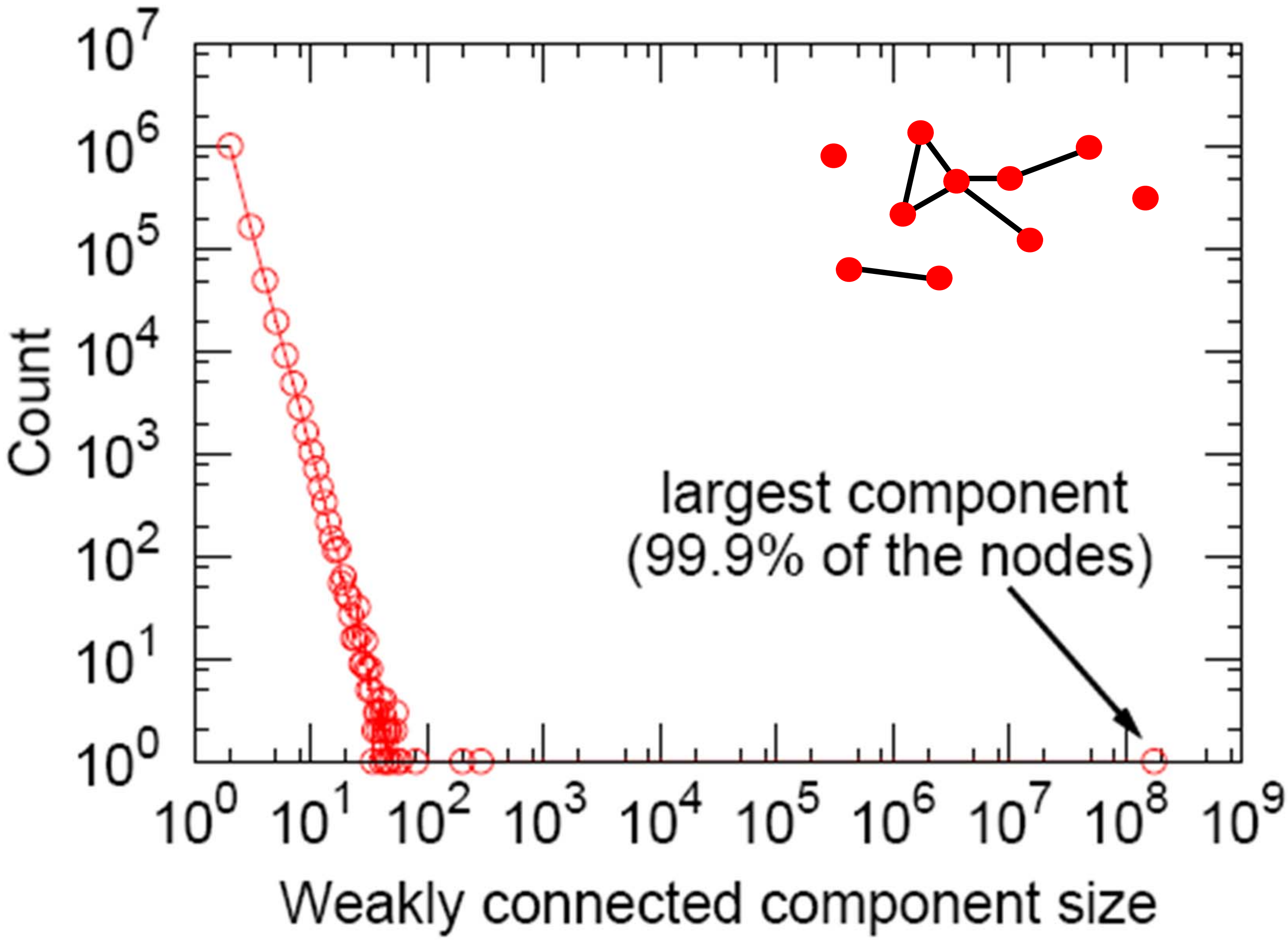
Messaging as a Network



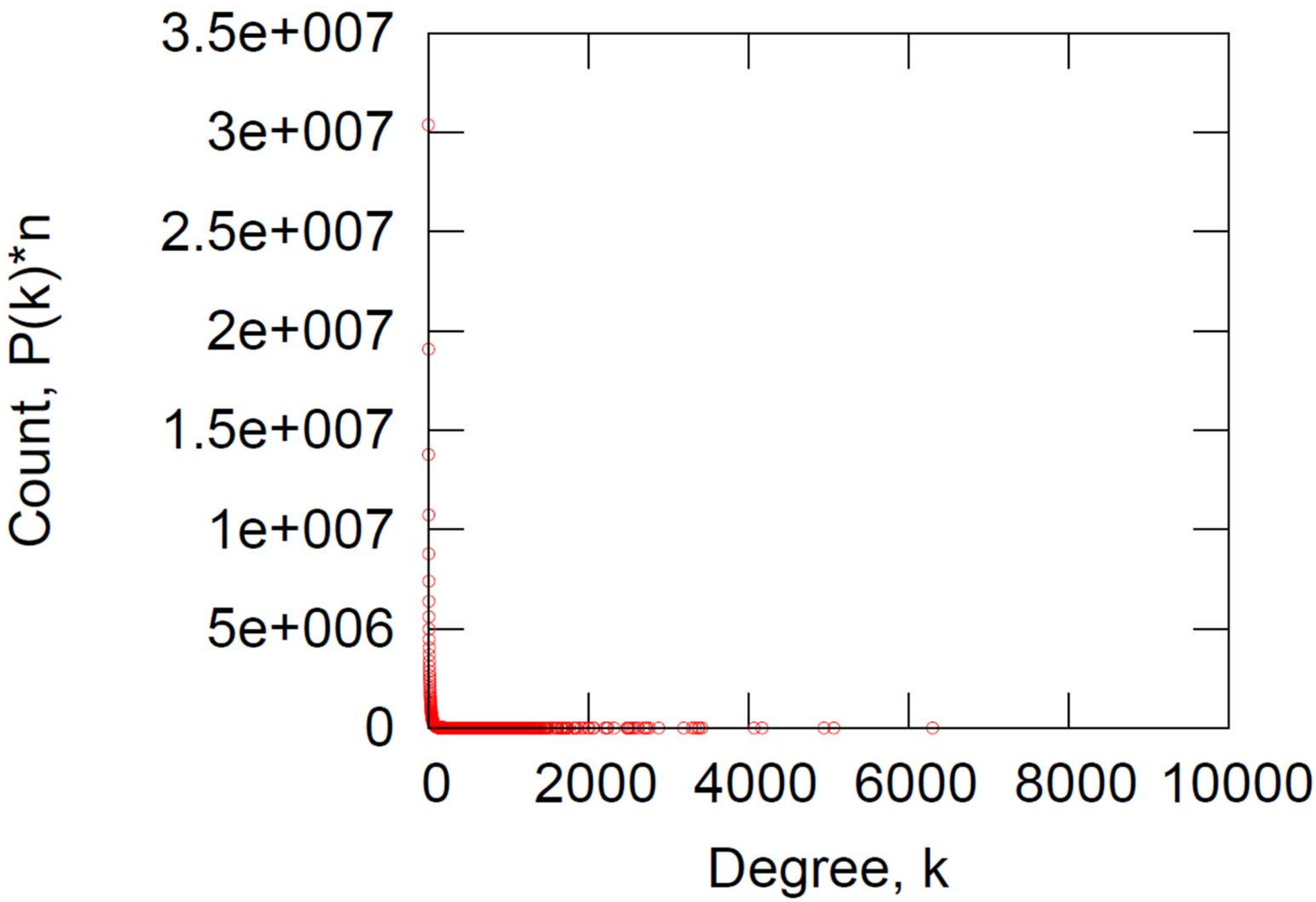
Communication graph

- Edge (u,v) if users *u* and *v* exchanged at least 1 msg
- N=180 million people
- E=1.3 billion edges

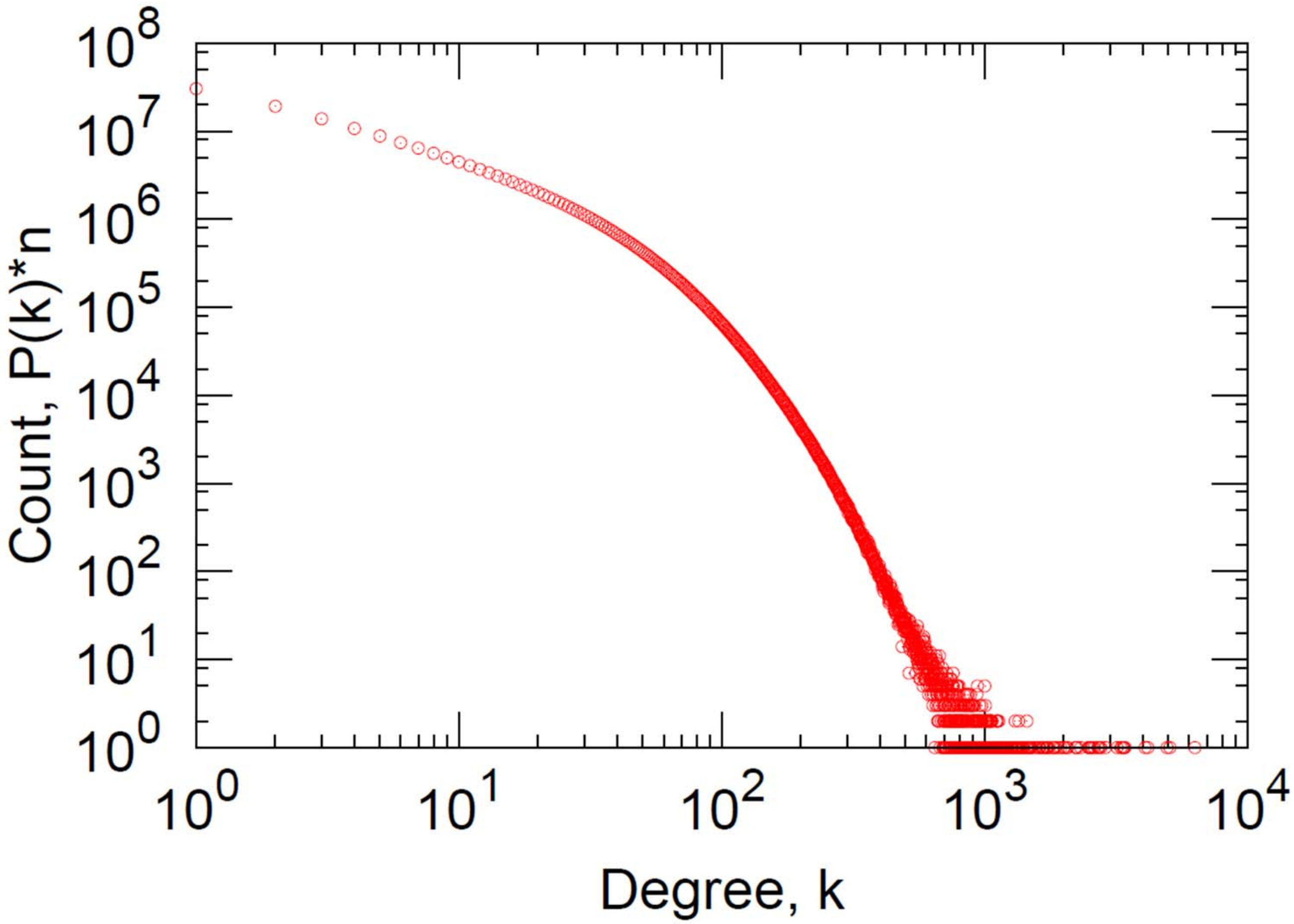
MSN Network: Connectivity



MSN: Degree Distribution

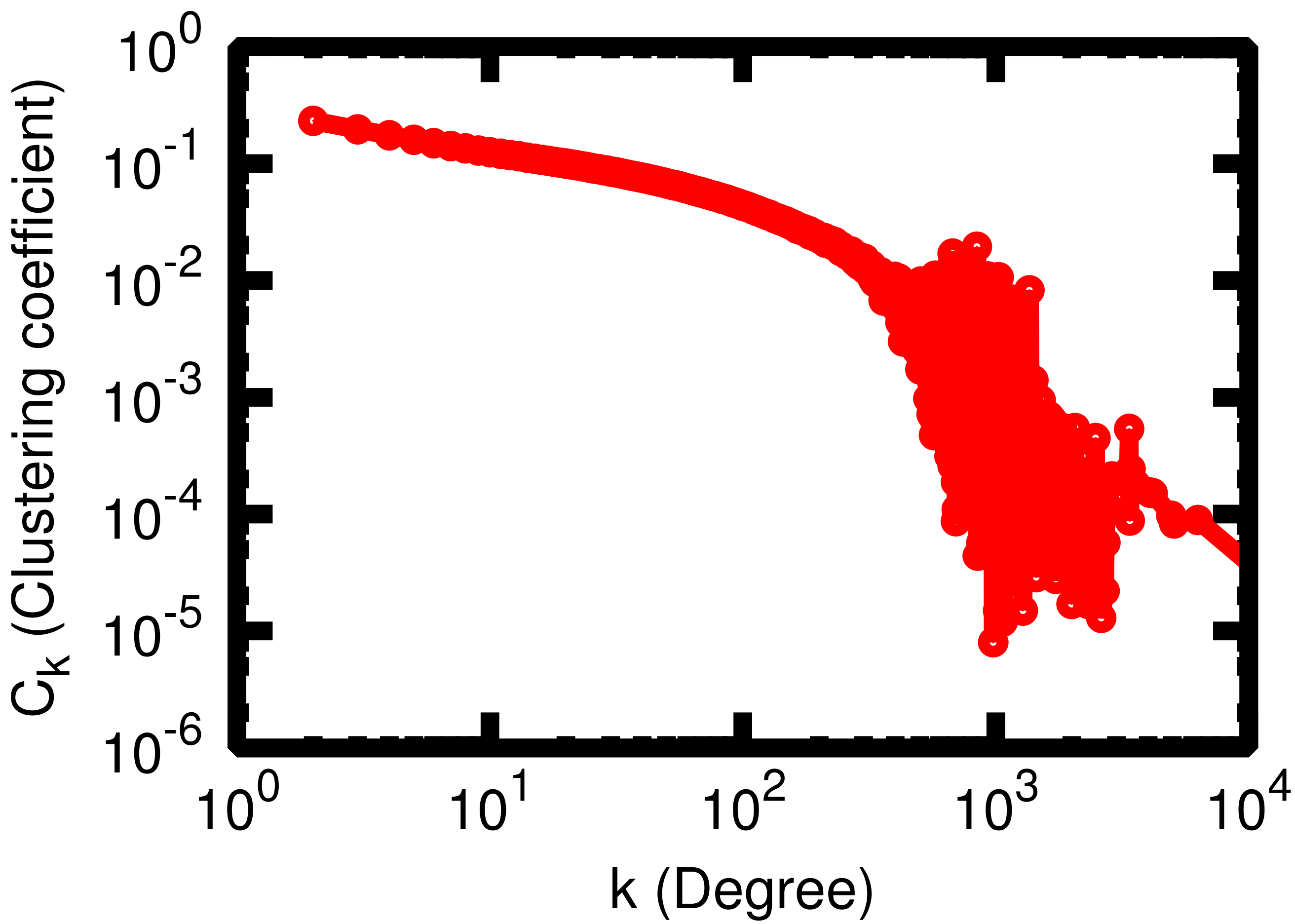


MSN: Log-Log Degree Distribution



We plot the same data as on the previous slide, just the axes are now logarithmic.

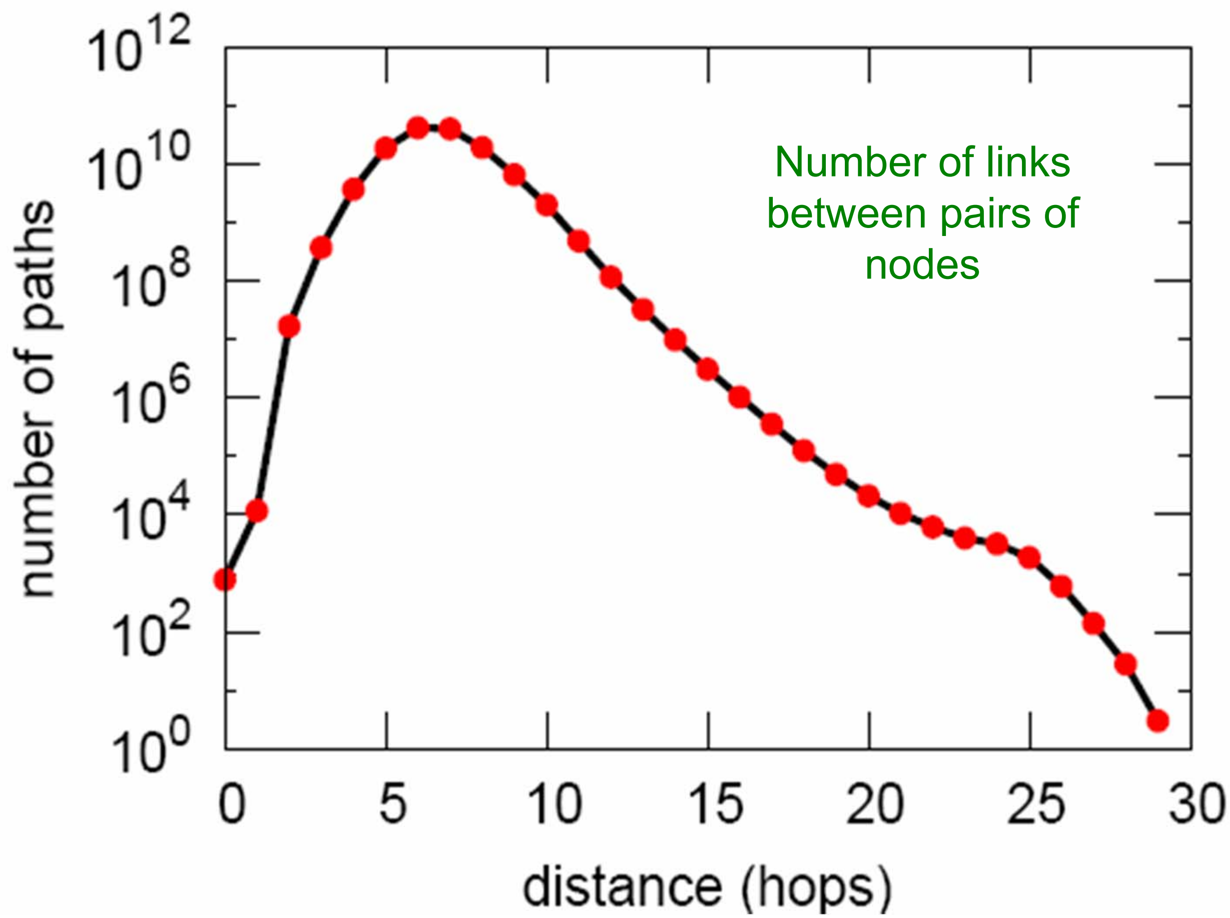
MSN: Clustering



Avg. clustering of the MSN:
C = 0.1140

C_k: average C_i of nodes *i* of degree *k*: $C_k = \frac{1}{N_k} \sum_{i:k_i=k} C_i$

MSN: Diameter



Avg. path length **6.6**
90% of the people can be reached in < 8 hops

nodes as we do BFS out of a random node

Steps	#Nodes
0	1
1	10
2	78
3	3,96
4	8,648
5	3,299,252
6	28,395,849
7	79,059,497
8	52,995,778
9	10,321,008
10	1,955,007
11	518,410
12	149,945
13	44,616
14	13,740
15	4,476
16	1,542
17	536
18	167
19	71
20	29
21	16
22	10
23	3
24	2
25	3

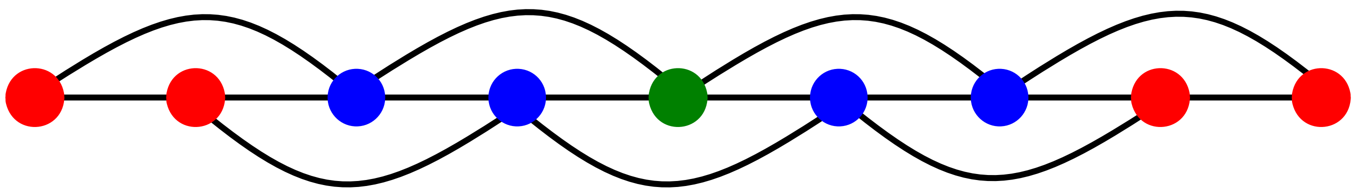
MSN: Key Network Properties

Degree distribution:	*heavily skewed* *avg. degree= 14.4*
Path length:	*6.6*
Clustering coefficient:	*0.11*

Are these metrics "expected"?
Are they "surprising"?

To answer this we need a null-model!

Is MSN Network like a "chain"?



- $P(k) = \delta(k\text{-}4)$ $k_i = 4$ for all nodes
- $C = ½$
- Path length: $h_{max} = \left\lceil \frac{N-1}{2} \right\rceil = O(N)$
  - The average shortest path-length: $\bar{h} = O(N)$

all as $N \rightarrow \infty$

- So, we have: **Constant degree, Constant avg. clustering coeff. Linear avg. path-length**

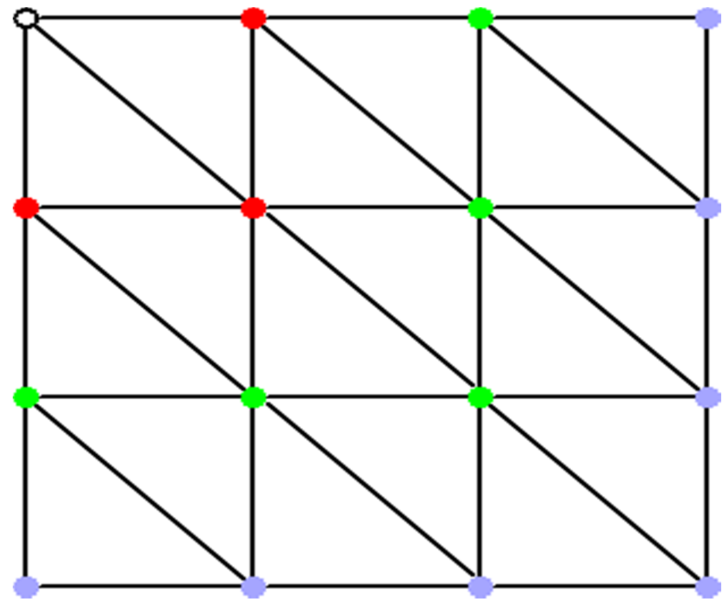
Note about calculations:
We are interested in quantities as graphs get large ($N\rightarrow\infty$)

We will use big-O:
$f(x) = O(g(x))$ as $x\rightarrow\infty$
if $f(x) < g(x)*c$ for all $x > x_0$
and some constant c.

Is MSN Network like a "grid"?

- $P(k) = \delta(k-6)$
  - $k = 6$ for each inside node
- $C = 6/15$ for inside nodes
- **Path length:**

$$h_{\max} = O(\sqrt{N})$$



- **In general, for lattices:**
  - Average path-length is $\bar{h} \approx N^{1/D}$ (D… lattice dimensionality)
  - Constant degree, constant clustering coefficient

Erdös-Renyi
Random Graph Model

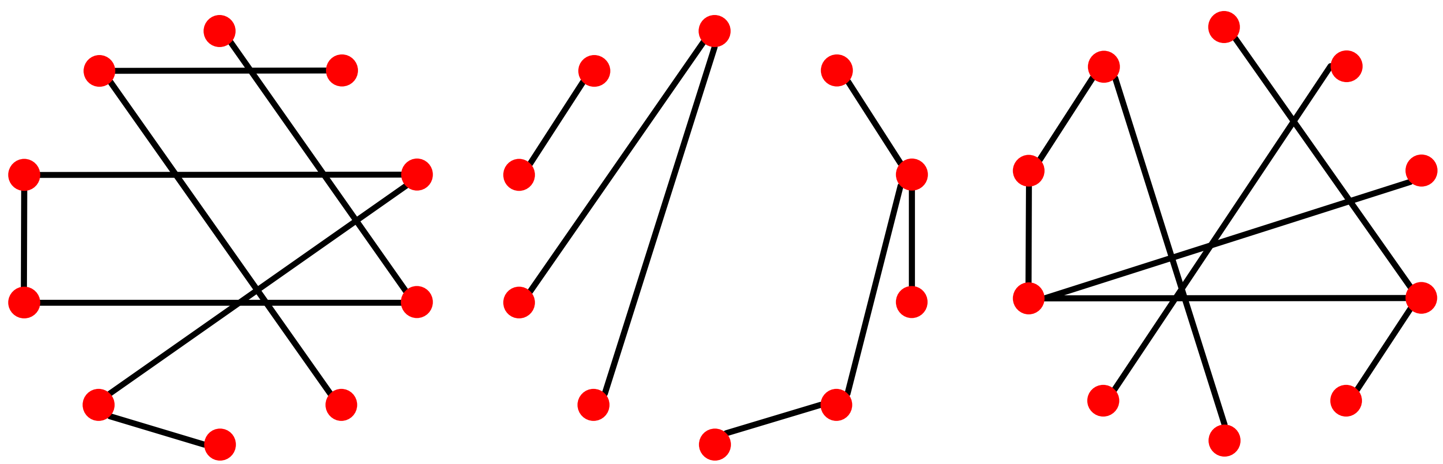
Simplest Model of Graphs

- **Erdös-Renyi Random Graphs** [Erdös-Renyi, ‘60]
- **Two variants:**
  - $G_{n,p}$: undirected graph on n nodes and each edge (u,v) appears i.i.d. with probability p
  - $G_{n,m}$: undirected graph with n nodes, and m uniformly at random picked edges

What kinds of networks does such model produce?

Random Graph Model

- ***n* and *p* do not uniquely determine the graph!**
  - The graph is a result of a random process
- We can have many different realizations given the same *n* and *p*



n = 10
p= 1/6

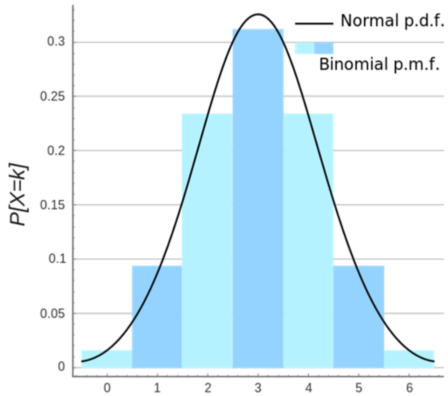
Random Graph Model: Edges

- **How likely is a graph on E edges?**
- $P(E)$: the probability that a given G_{np} generates a graph on exactly E edges:

$$P(E) = \binom{E_{\max}}{E} p^{E} (1-p)^{E_{\max}-E}$$

where $E_{max}=n(n-1)/2$ is the maximum possible number of edges in an undirected graph of n nodes

P(E) is exactly the
Binomial distribution >>>
Number of successes in a sequence of **n** independent yes/no experiments



Node Degrees in a Random Graph

- **What is expected degree of a node?**
  - Let X_v be a rnd. var. measuring the degree of node v
  - **We want to know:** $E[X_v] = \sum_{j=0}^{n-1} j\, P(X_v = j)$
    - **For the calculation we will need: Linearity of expectation**
      - For any random variables $Y_1, Y_2, \ldots, Y_k$
      - If $Y=Y_1+Y_2+\ldots Y_k$, then $E[Y]= \sum_i E[Y_i]$
- **An easier way:**
  - Decompose X_v to $X_v= X_{v,1}+X_{v,2}+\ldots+X_{v,n-1}$
    - where $X_{v,u}$ is a $\{0,1\}$-random variable which tells if edge (v,u) exists or not

$$E[X_v] = \sum_{u=1}^{n-1} E[X_{vu}] = (n-1)p$$

How to think about this?

- Prob. of node u linking to node v is p
- u can link (flips a coin) to all other $(n\text{-}1)$ nodes
- Thus, the expected degree of node u is: $p(n\text{-}1)$

Properties of G_{np}

Degree distribution: $P(k)$

Path length: h

Clustering coefficient: C

What are values of these properties for G_{np}?

Degree Distribution

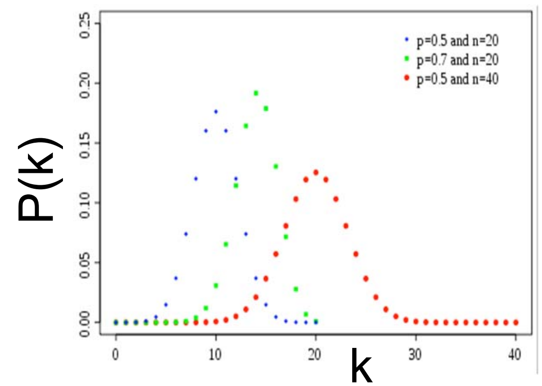
- **Fact:** **Degree distribution of G_{np} is Binomial.**
- Let $P(k)$ denote a fraction of nodes with degree k:

$$P(k) = \binom{n-1}{k} p^k (1-p)^{n-1-k}$$

Select k nodes out of n-1

Probability of having k edges

Probability of missing the rest of the n-1-k edges



Mean, variance of a binomial distribution

$$\bar{k} = p(n-1)$$

$$\sigma^2 = p(1-p)(n-1)$$

$$\frac{\sigma}{\bar{k}} = \left[\frac{1-p}{p}\frac{1}{(n-1)}\right]^{1/2} \approx \frac{1}{(n-1)^{1/2}}$$

As the network size increases, the distribution becomes increasingly narrow—we are increasingly confident that the degree of a node is in the vicinity of $\bar{k}$.

Clustering Coefficient of G_{np}

- **Remember:** $$C_i = \frac{2e_i}{k_i(k_i-1)}$$

Where e_i is the number of edges between i's neighbors

- Edges in G_{np} appear i.i.d with prob. p

- **So:** $$e_i = p\frac{k_i(k_i-1)}{2}$$

Each pair is connected with prob. *p*

Number of distinct pairs of neighbors of node *i* of degree k_i

- **Then:** $$C = \frac{p \cdot k_i(k_i-1)}{k_i(k_i-1)} = p = \frac{\bar{k}}{N}$$

Clustering coefficient of a random graph is small.
For a fixed avg. degree, *C* decreases with the graph size *N*.

Network Properties of G_{np}

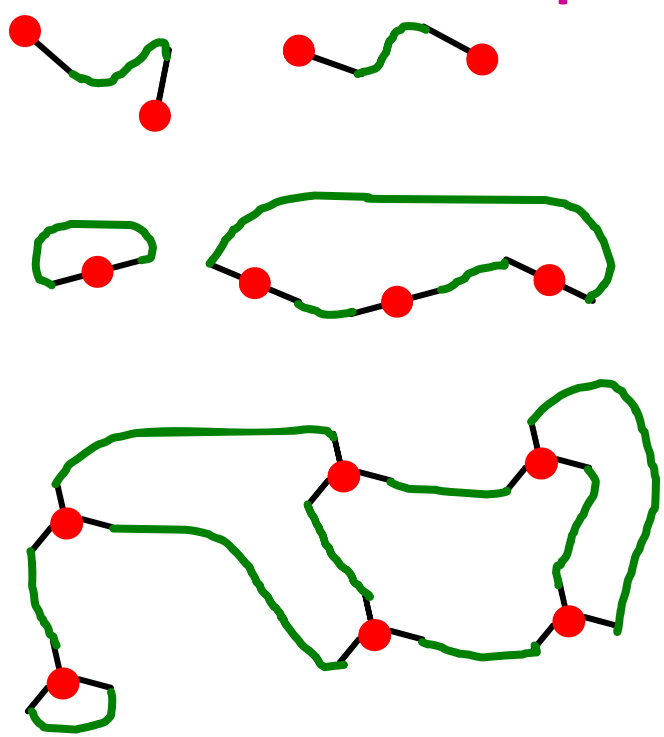
Degree distribution: $P(k) = \binom{n-1}{k} p^k (1-p)^{n-1-k}$

Clustering coefficient: $C = p = \bar{k}/n$

Path length: ***next!***

Def: Random k-Regular Graphs

- To prove the diameter of a G_{np} we define few concepts
- **Random k-Regular graph:**
  - **Assume each node has k spokes (half-edges)**
    - **k=1:**
    - **k=2:**
    - **k=3:**



Graph is a set of pairs

Graph is a set of cycles

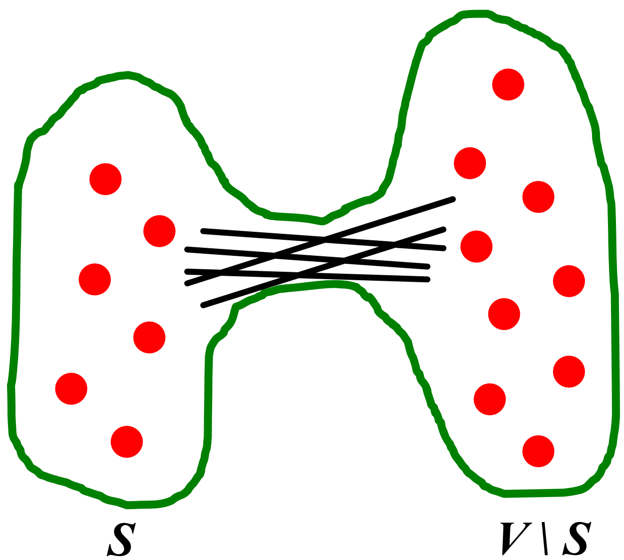
Arbitrarily complicated graphs

  - **Randomly pair them up!**

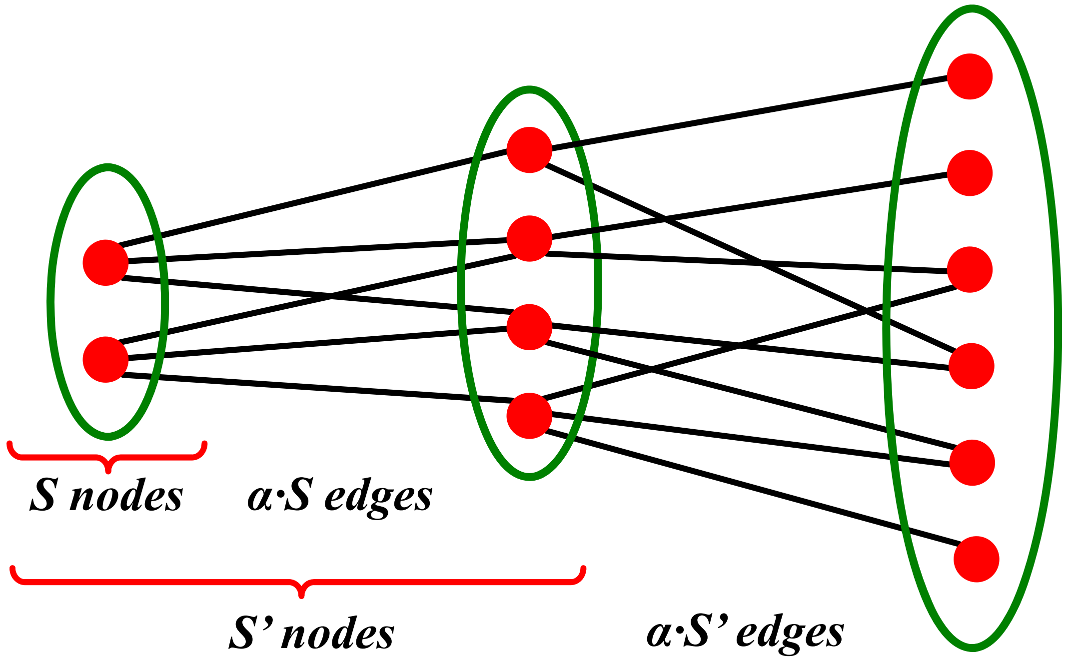
Def: Expansion

- Graph $G(V, E)$ has **expansion α**: if $\forall S \subseteq V$:
  $\#\text{ of edges leaving } S \geq \alpha \cdot \min(|S|, |V \setminus S|)$
- **Or equivalently:**

$$\alpha = \min_{S \subseteq V} \frac{\#\, edges\ leaving\ S}{\min(|S|, |V \setminus S|)}$$



Expansion: Intuition



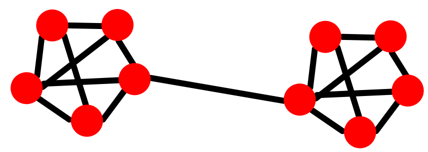
$$\alpha = \min_{S \subseteq V} \frac{\#\, edges\ leaving\ S}{\min(|S|, |V \setminus S|)}$$

(A big) graph with "good" expansion

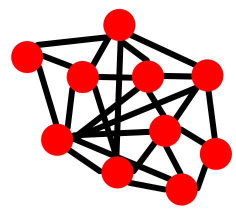
Expansion: Measures Robustness

$$\alpha = \min_{S \subseteq V} \frac{\#\, edges\ leaving\ S}{\min(|S|, |V \setminus S|)}$$

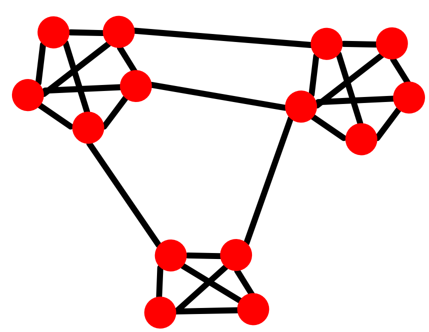
- Expansion is **measure of robustness:**
  - To disconnect l nodes, we need to cut $\geq \alpha \cdot l$ edges
- Low expansion:



- High expansion:



- Social networks:
  - "Communities"



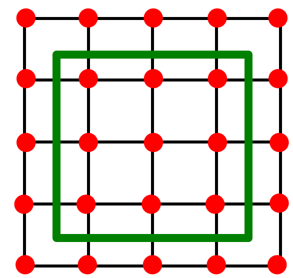
Expansion: k-Regular Graphs

$$\alpha = \min_{S \subseteq V} \frac{\#\, edges\ leaving\ S}{\min(|S|, |V \setminus S|)}$$

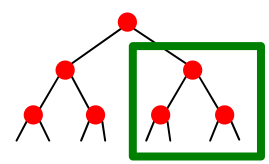
- ***k*-regular graph** (every node has degree k):
  - Expansion is at most k (when S is a single node)
- **Is there a graph on n nodes ($n \to \infty$), of fixed max deg. k, so that expansion α remains const?**

Examples:

- **n×n grid:** $k=4$: $\alpha = 2n/(n^2/4) \to 0$
  ($S = n/2 \times n/2$ square in the center)



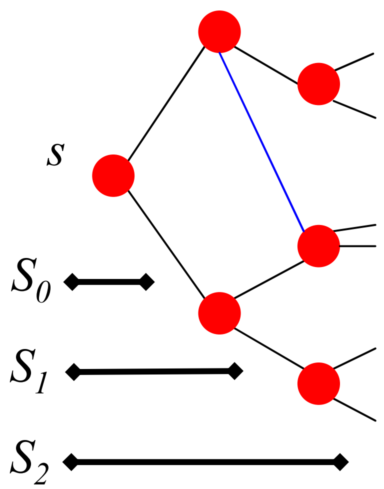
- **Complete binary tree:**
  $\alpha \to 0$ for $|S| = (n/2)-1$



- **Fact:** For a random **3-regular graph** on n nodes, there is some const α ($\alpha > 0$, independent. of n) such that w.h.p. the expansion of the graph is $\geq \alpha$

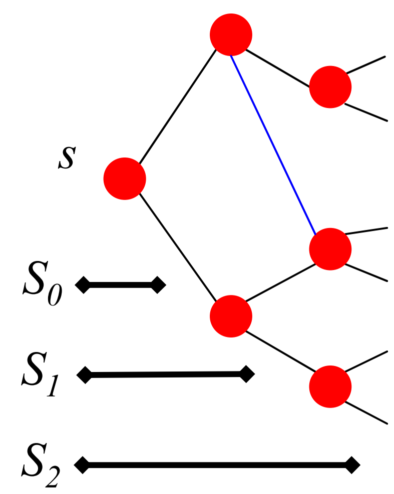
Diameter of 3-Regular Rnd. Graph

- **Fact:** In a graph on n nodes with expansion α for all pairs of nodes s and t there is a path of $O((\log n) / \alpha)$ edges connecting them.
- Proof:
  - Proof strategy:
    - We want to show that from any node s there is a path of length $O((\log n)/\alpha)$ to any other node t
  - Let S_j be a set of all nodes found within j steps of BFS from s.
  - **How does S_j increase as a function of j?**

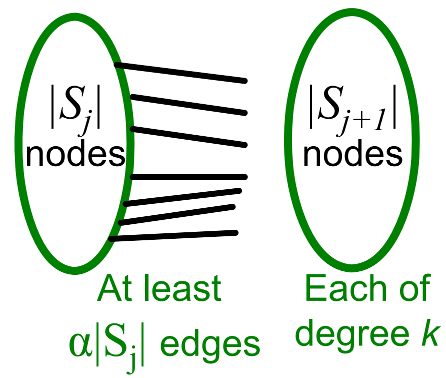


Diameter of 3-Regular Rnd. Graph

- <u>Proof (continued):</u>
  - Let S_j be a set of all nodes found within j steps of BFS from s.
  - **We want to relate S_j and S_{j+1}**



$$|S_{j+1}| \geq |S_j| + \frac{\overbrace{\alpha|S_j|}^{\text{Expansion}}}{\underbrace{k}_{\substack{\text{At most } k \text{ edges} \\ \text{"collide" at a node}}}} =$$



$$|S_{j+1}| \geq |S_j|\left(1+\frac{\alpha}{k}\right) = \left(1+\frac{\alpha}{k}\right)^{j+1}$$

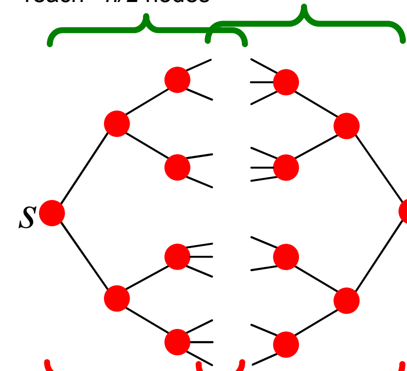
Diameter of 3-Regular Rnd. Graph

$$e = \lim_{x\to\infty}\left(1+\frac{1}{x}\right)^x$$

- <u>Proof (continued):</u>
  - **In how many steps of BFS we reach $>n/2$ nodes?**
  - Need j so that: $S_j = \left(1+\frac{\alpha}{k}\right)^j \geq \frac{n}{2}$
  - Let's set: $j = \frac{k\log_2 n}{\alpha}$
  - Then:

$$\left(1+\frac{\alpha}{k}\right)^{\frac{k\log_2 n}{\alpha}} \geq 2^{\log_2 n} = n > \frac{n}{2}$$

  - In $2k/\alpha\cdot\log n$ steps $|S_j|$ grows to $\Theta(n)$. So, **the diameter of G is $O(\log(n)/\alpha)$**



Claim: $\left(1+\frac{\alpha}{k}\right)^{\frac{k\log_2 n}{\alpha}} \geq 2^{\log_2 n}$

Remember $n>0$, $\alpha \leq k$ then:

if $\alpha = k: (1+1)^{\frac{1}{1}\log_2 n} = 2^{\log_2 n}$

if $\alpha \to 0$ then $\frac{k}{\alpha} = x \to \infty$:

and $\left(1+\frac{1}{x}\right)^{x\log_2 n} = e^{\log_2 n} > 2^{\log_2 n}$

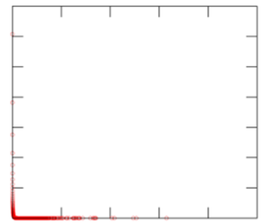
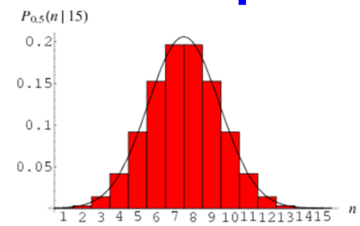
Network Properties of G_{np}

Degree distribution: $P(k) = \binom{n-1}{k} p^k (1-p)^{n-1-k}$

Path length: $O(\log n)$

Clustering coefficient: $C = p = \bar{k} / n$

MSN vs. G_{np}

	MSN	**G_{np}**
Degree distribution:		
Path length:	**6.6**	$O(\log n)$ $h \approx 8.2$
Clustering coefficient:	0.11	$\bar{k}/n$ $C \approx 8 \cdot 10^{-8}$

Real Networks vs. G_{np}

- **Are real networks like random graphs?**
  - Giant connected component: ☺
  - Average path length: ☺
  - Clustering Coefficient: ☹
  - Degree Distribution: ☹
- **Problems with the random network model:**
  - Degreed distribution differs from that of real networks
  - Giant component in most real network does NOT emerge through a phase transition
  - No local structure – clustering coefficient is too low
- **Most important: Are real networks random?**
  - The answer is simply: **NO!**

Real Networks vs. G_{np}

- **If G_{np} is wrong, why did we spend time on it?**
  - It is the reference model for the rest of the class.
  - It will help us calculate many quantities, that can then be compared to the real data
  - It will help us understand to what degree is a particular property the result of some random process

So, while G_{np} is WRONG, it will turn out to be extremly USEFUL!

EXTRA: "Evolution" of the G_{np}

What happens to G_{np} when we vary p?

Back to Node Degrees of G_{np}

- Remember, expected degree $E[X_v] = (n-1)p$
- We want $E[X_v]$ be independent of n
  So let: $p=c/(n-1)$
- Observation: If we build random graph G_{np} with $p=c/(n-1)$ we have many isolated nodes
- Why?

$$P[v \text{ has degree } 0] = (1-p)^{n-1} = \left(1 - \frac{c}{n-1}\right)^{n-1} \xrightarrow[n\to\infty]{} e^{-c}$$

$$\lim_{n\to\infty}\left(1-\frac{c}{n-1}\right)^{n-1} = \left(1-\frac{1}{x}\right)^{-x\cdot c} = \Bigg[\underbrace{\lim_{x\to\infty}\left(1-\frac{1}{x}\right)^{-x}}_{e}\Bigg]^{-c} = e^{-c}$$

Use substitution $\frac{1}{x} = \frac{c}{n-1}$

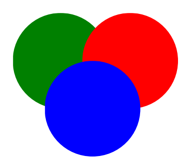
By definition: $e = \lim_{x\to\infty}\left(1+\frac{1}{x}\right)^{x}$

No Isolated Nodes

- **How big do we have to make p before we are likely to have no isolated nodes?**
- We know: $P[v$ has degree $0] = e^{-c}$
- Event we are asking about is:
  - I = some node is isolated
  - $I = \bigcup_{v \in N} I_v$ where I_v is the event that v is isolated
- **We have:**

$$P(I) = P\left(\bigcup_{v \in N} I_v\right) \le \sum_{v \in N} P(I_v) = ne^{-c}$$

Union bound



A_i

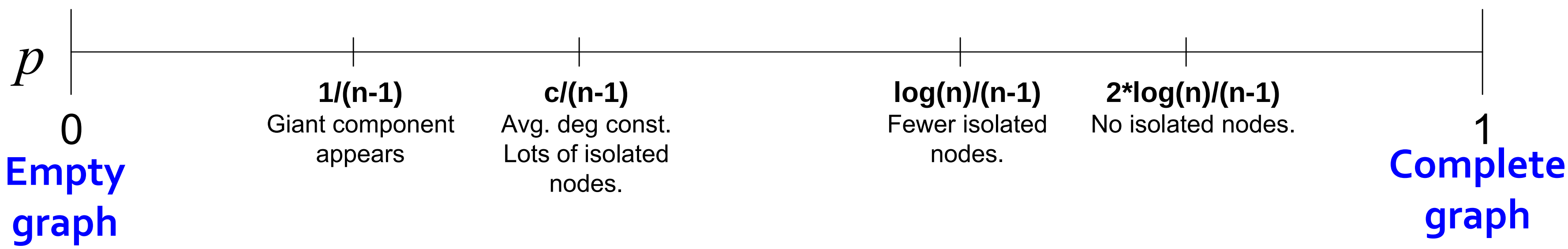
$$\left|\bigcup_i A_i\right| \le \sum_i |A_i|$$

No Isolated Nodes

- **We just learned: $P(I) = n\, e^{-c}$**
- Let's try:
  - $c = \ln n$ then: $n\, e^{-c} = n\, e^{-\ln n} = n \cdot 1/n = 1$
  - $c = 2 \ln n$ then: $n\, e^{-2 \ln n} = n \cdot 1/n^2 = 1/n$
- **So if:**
  - $p = \ln n$ then: $P(I) = 1$
  - $p = 2 \ln n$ then: $P(I) = 1/n \to 0$ as $n \to \infty$

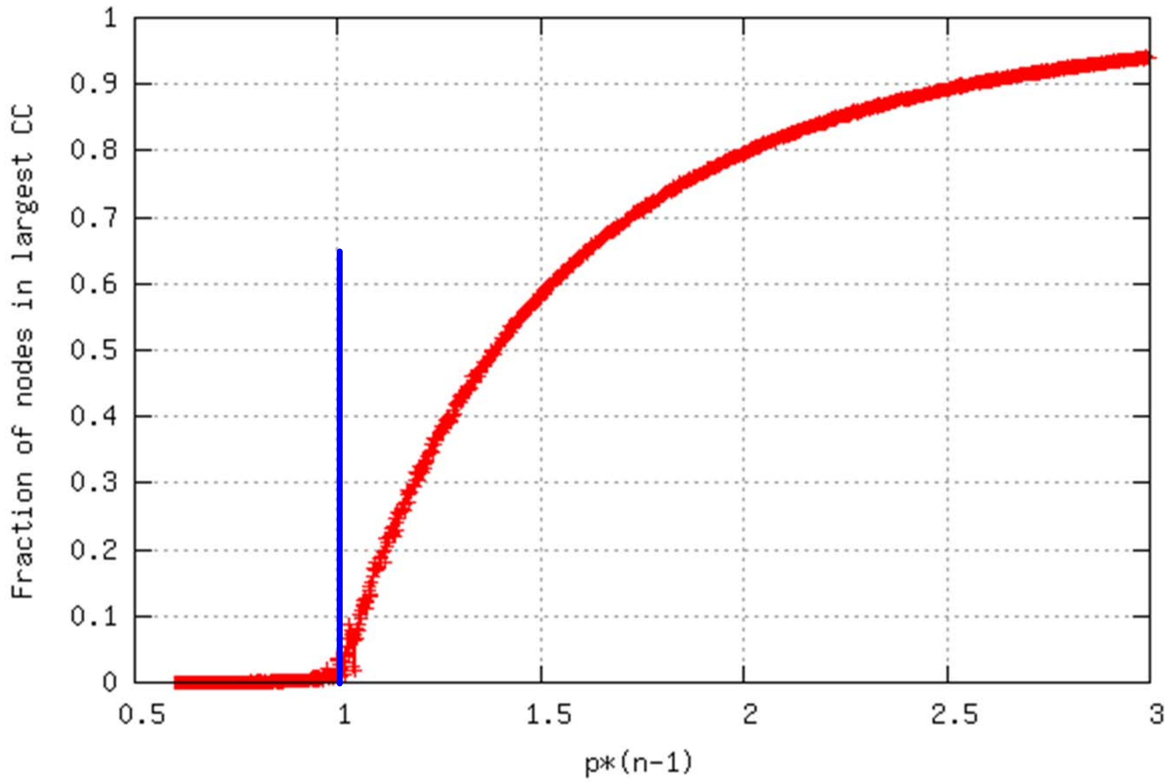
"Evolution" of a Random Graph

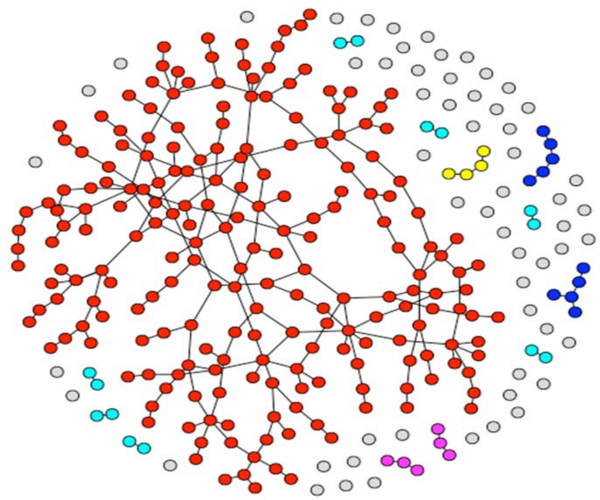
- **Graph structure of G_{np} as p changes:**



- **Emergence of a Giant Component:**
  avg. degree $k=2E/n$ or $p=k/(n-1)$
  - $k=1-\varepsilon$: all components are of size $\Omega(\log n)$
  - $k=1+\varepsilon$: 1 component of size $\Omega(n)$, others have size $\Omega(\log n)$

G_{np} Simulation Experiment





Fraction of nodes in the largest component

- G_{np}, n=100k, $p(n\text{-}1)$ = 0.5 … 3