Complex Dynamical Networks:

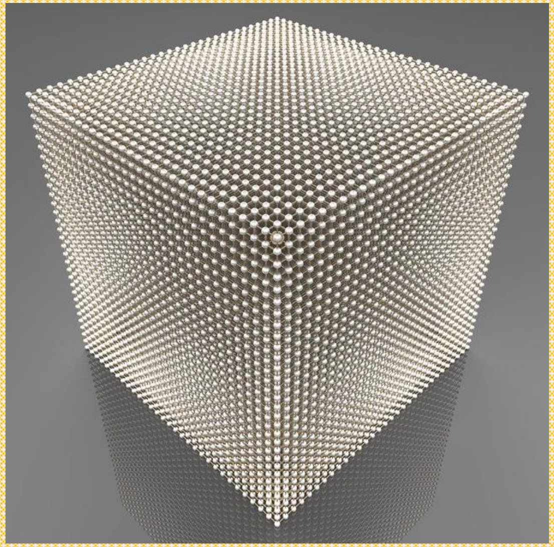
Lecture 6b: Network Robustness

EE 6605

Instructor: G Ron Chen

Percolation Theory

Percolation (渗流) theory studies clustering in complex networks



Example: Consider a cellular material (多孔材料).

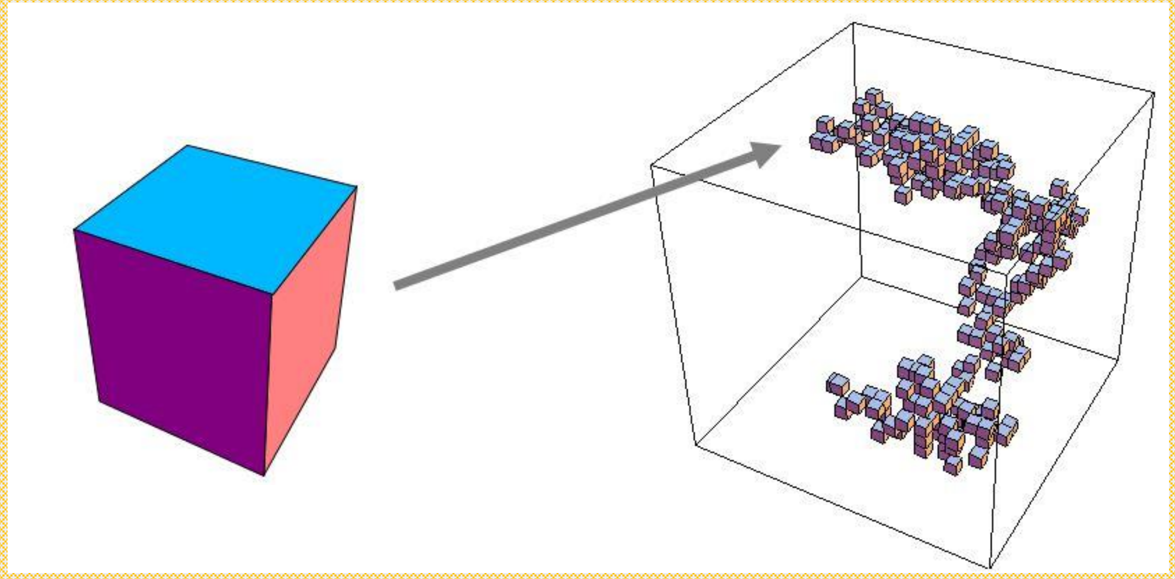
Can liquid flow from the top through to the bottom?

Describe the material by a 3D lattice

Percolation Theory

Mathematical framework: On an $n \times n \times n$ lattice, for each pair of adjacent vertices (nodes), with a probability p connect them together (namely, with probability $(1-p)$, do not connect them).

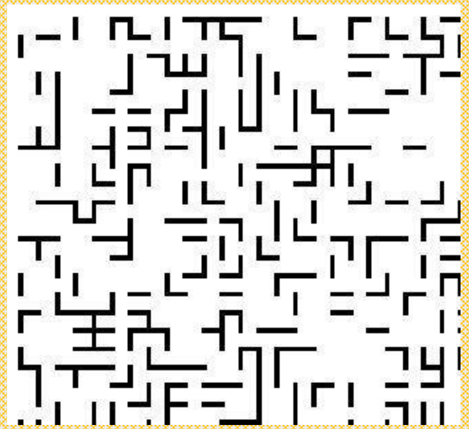
Assume that all connection events are mutually independent.

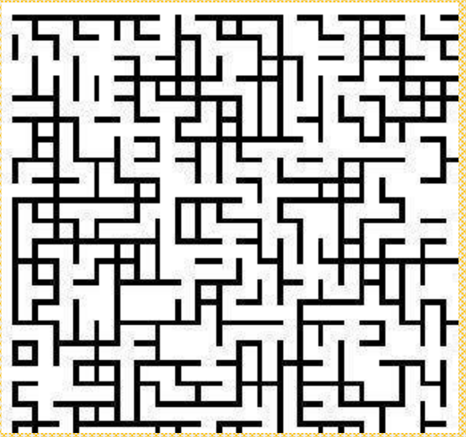


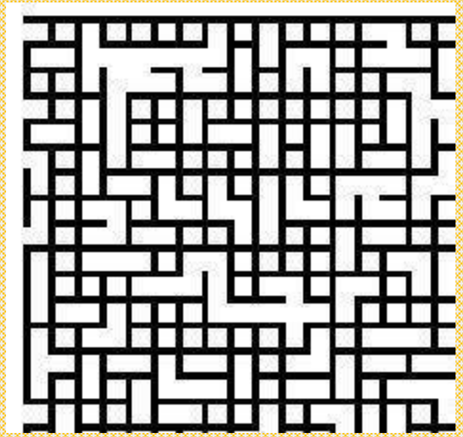
Percolation Theory

Bond percolation (边渗流): as $n \to \infty$, for what values of p , there exists at least one path that allows the liquid to flow through?

The lower bound of p, denoted p_c , is the percolation threshold







$p < p_c$ $p = p_c$ $p > p_c$

Random operation: Connect adjacent nodes with probability p (namely, not connecting with probability $1-p$)

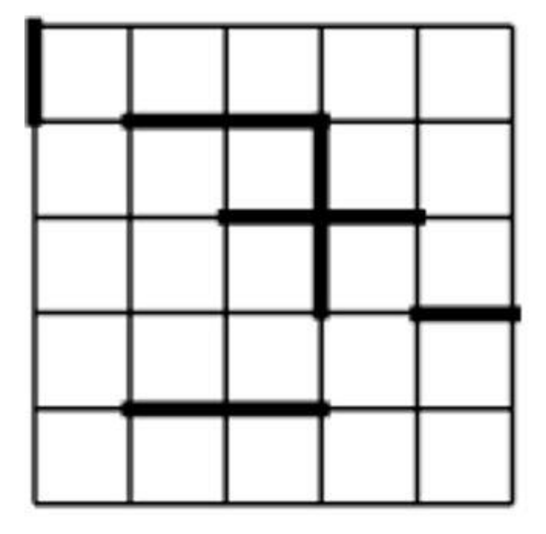
Bond percolation: for what values of p, there exists at least one path that allows the liquid to flow through?

1D: Chain: $p_c = 1$

2D: Lattice: $p_c = 0.5$

½ are connected horizontally

½ are connected vertically

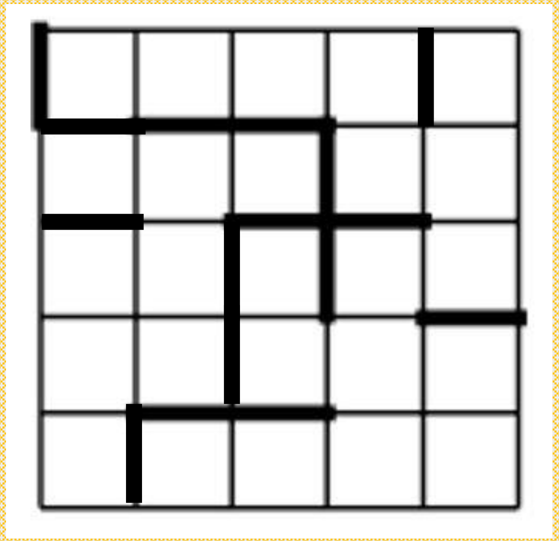


Inverse Problem: Given a lattice network

Edge-attack: with probability p, remove an edge (namely, with probability $(1-p)$, not remove it), such that the liquid cannot flow through the network

Robustness Problem 1:

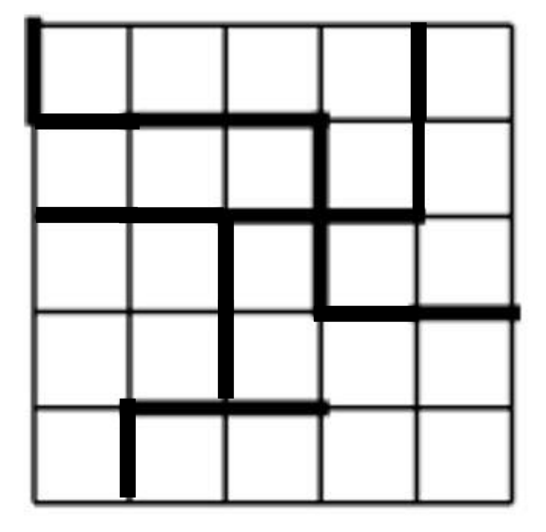
For what range of p values, there exists at least one path?



Inverse Problem: Given a lattice network

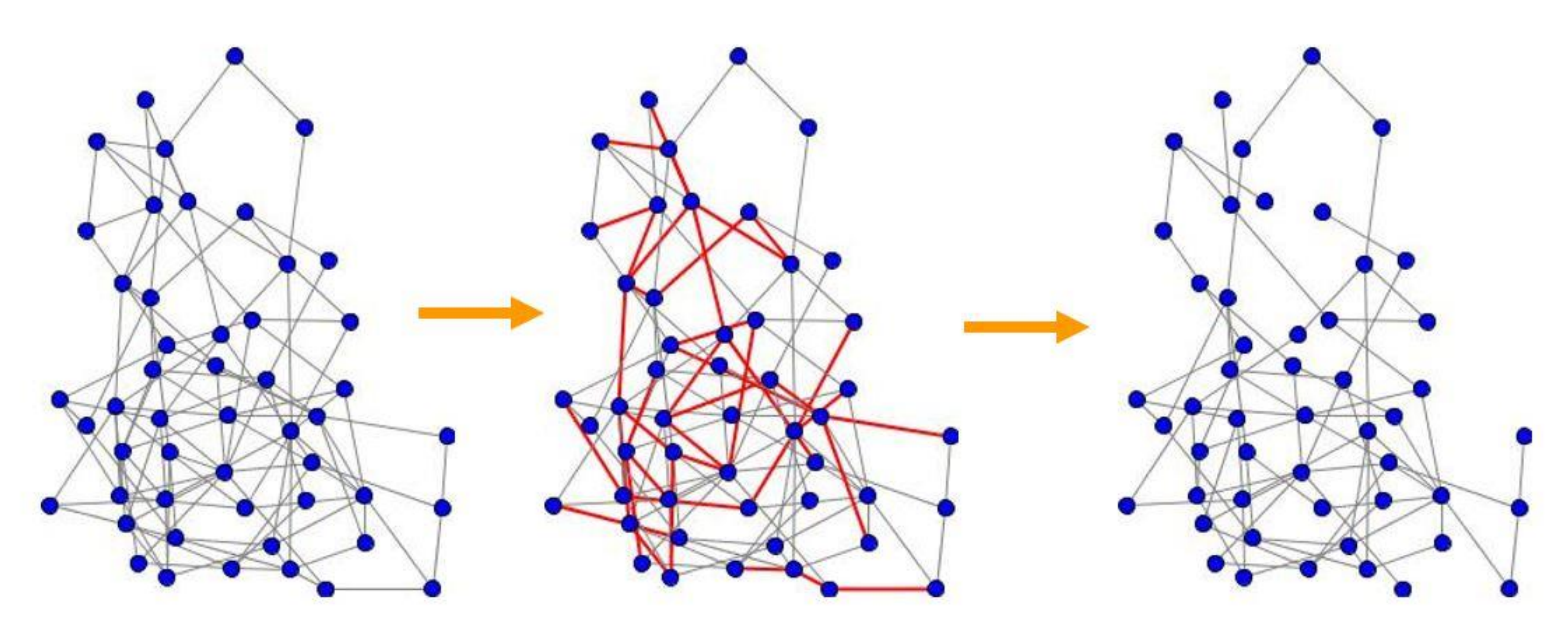
Edge-attack: with probability p remove an edge (namely, with probability $(1-p)$, not remove it), such that the liquid cannot flow through the network

Robustness Problem 2: For what range of p values, the lattice remains to be connected after being attacked?



Robustness Problem 3: After an attack, what is the size of the largest cluster (community)?

Example: (Erdos-Renyi random graph) 50 nodes, 116 edges, $\langle k \rangle = 4.64$



(Lada Adamic)

$p_c = 0.25$ → Removed 29 edges (on average)

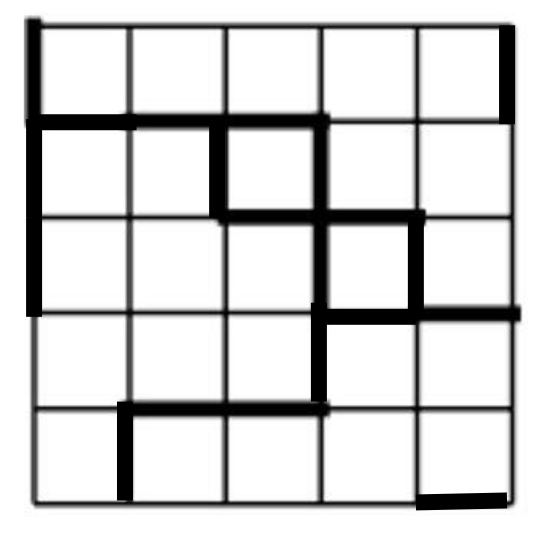
Above Graph: Even after removed 40 edges, the network remains to be connected, $\langle k \rangle = 3.04$

Site Percolation (点渗流):

Every node is "occupied" with probability *p* (namely, "not occupied" with probability $1 - p$).

["occupied" → allowing liquid to pass]

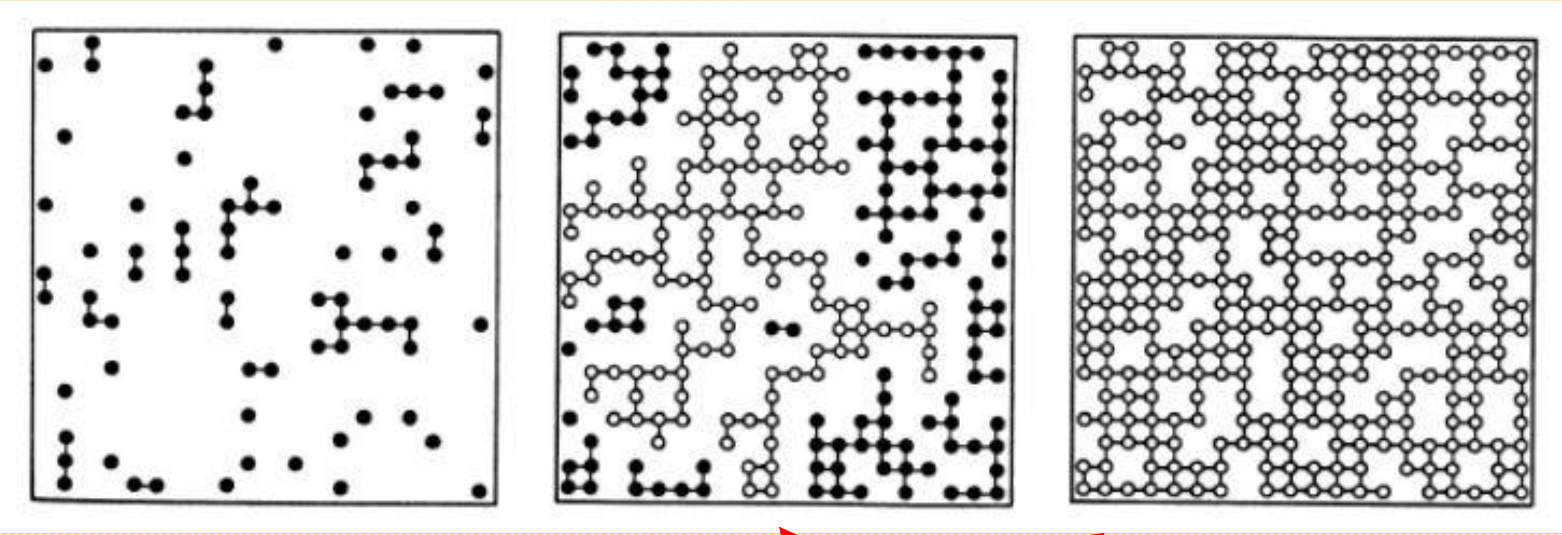
If two adjacent nodes are both occupied, then connect them together by an edge.



Site Percolation (点渗流):

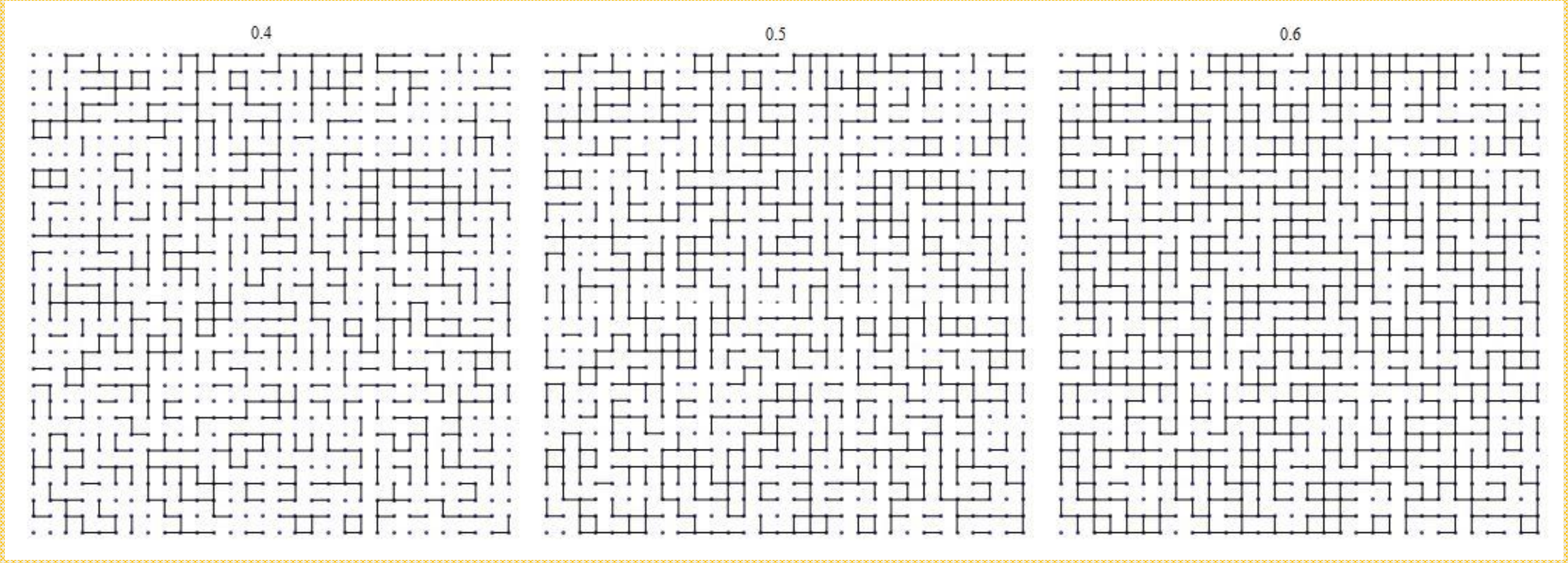
Problem 1:

For what range of p values, there exist at least one path?

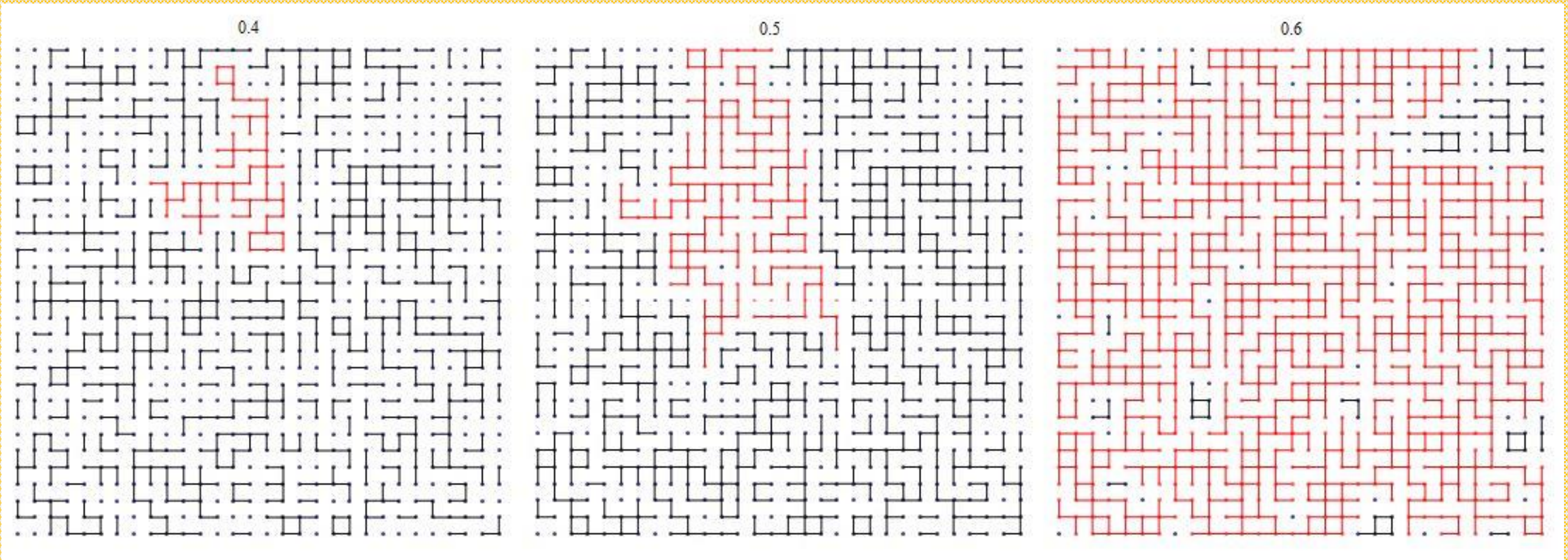


$p < p_c$ $p = p_c$ $p > p_c$

Giant Cluster appears



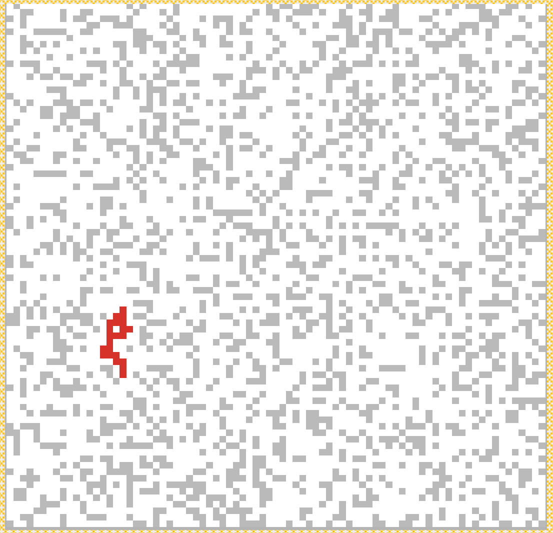
0.4
0.5
0.6



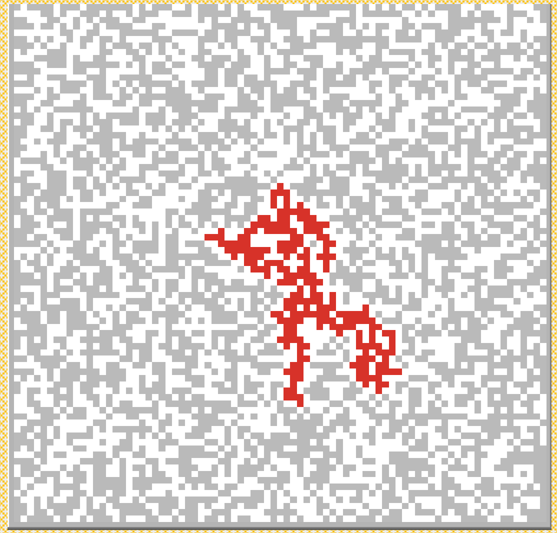
0.4
0.5
0.6

Problem 1:

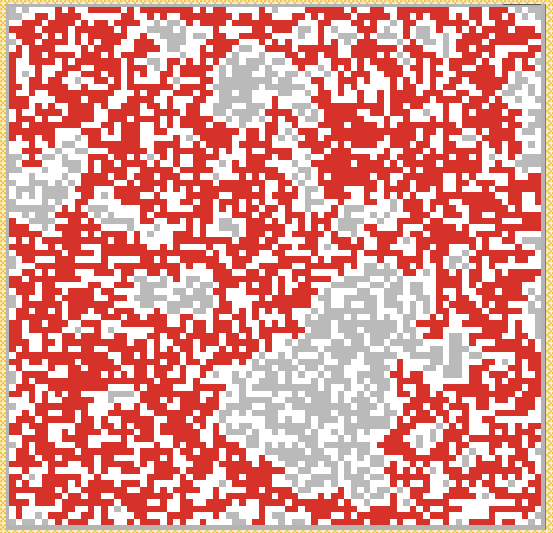
For what range of p values, there exist at least one path?



$p < p_c$



$p = p_c$



$p > p_c$

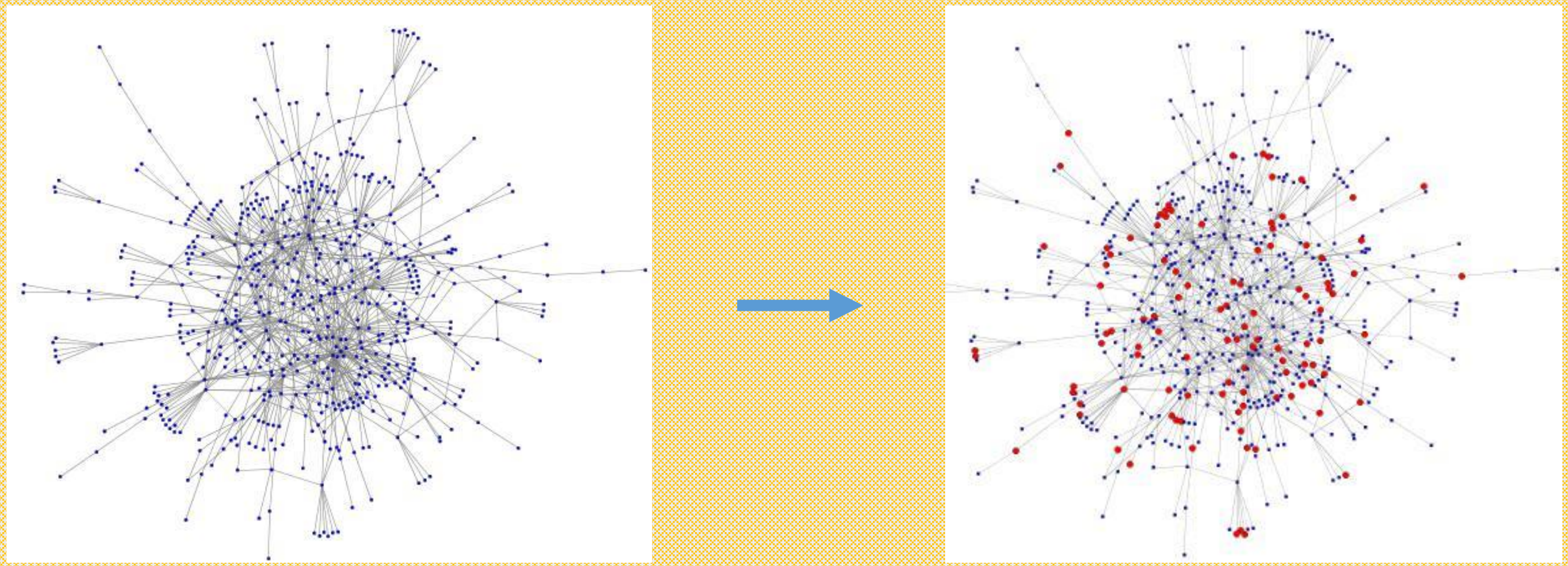
(Giant Cluster appears)

(Lada Adamic)

Percolation of General Complex Networks

Percolation concept can be extended to general complex networks

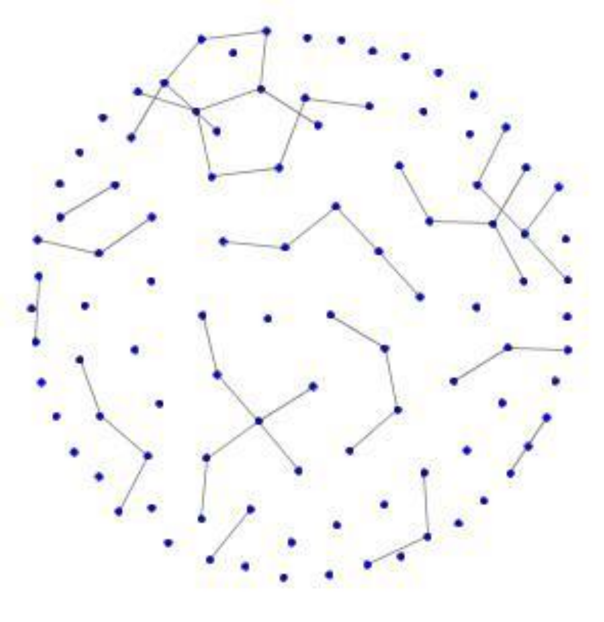
A network percolates if giant cluster(s) are emerged



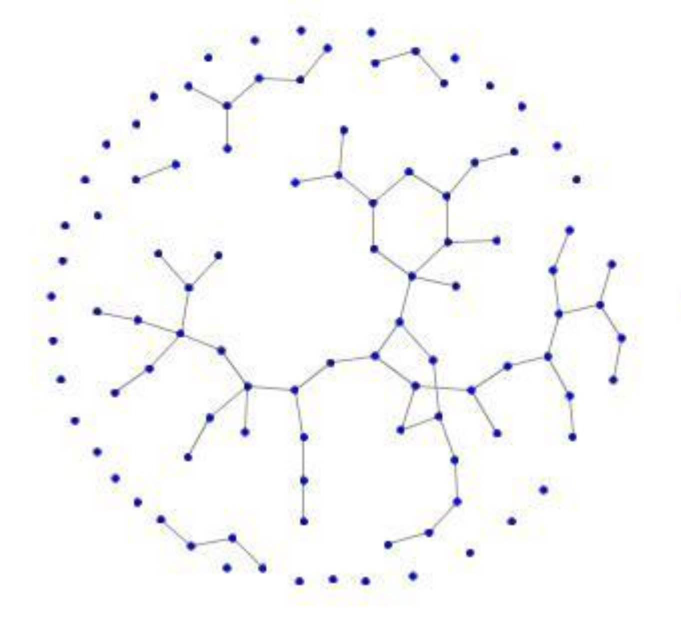
A scale-free network always has giant cluster(s), so it always percolates

Average Degree can be used as a measure

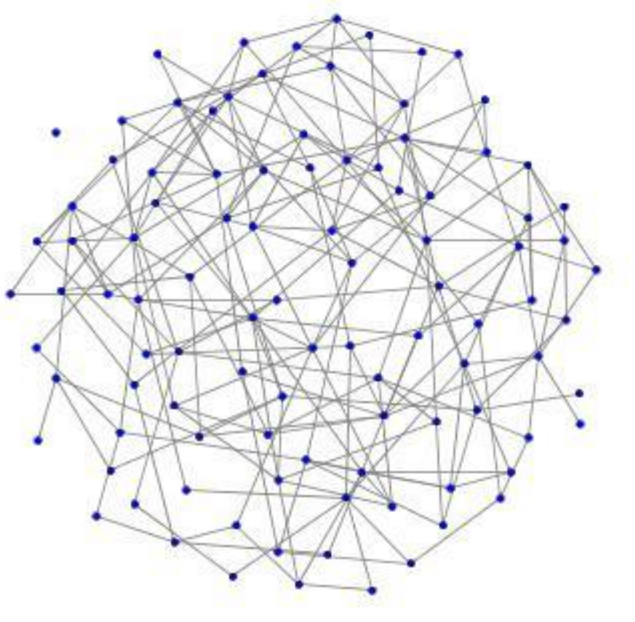
Example: Random Graph



ave deg = 0.99

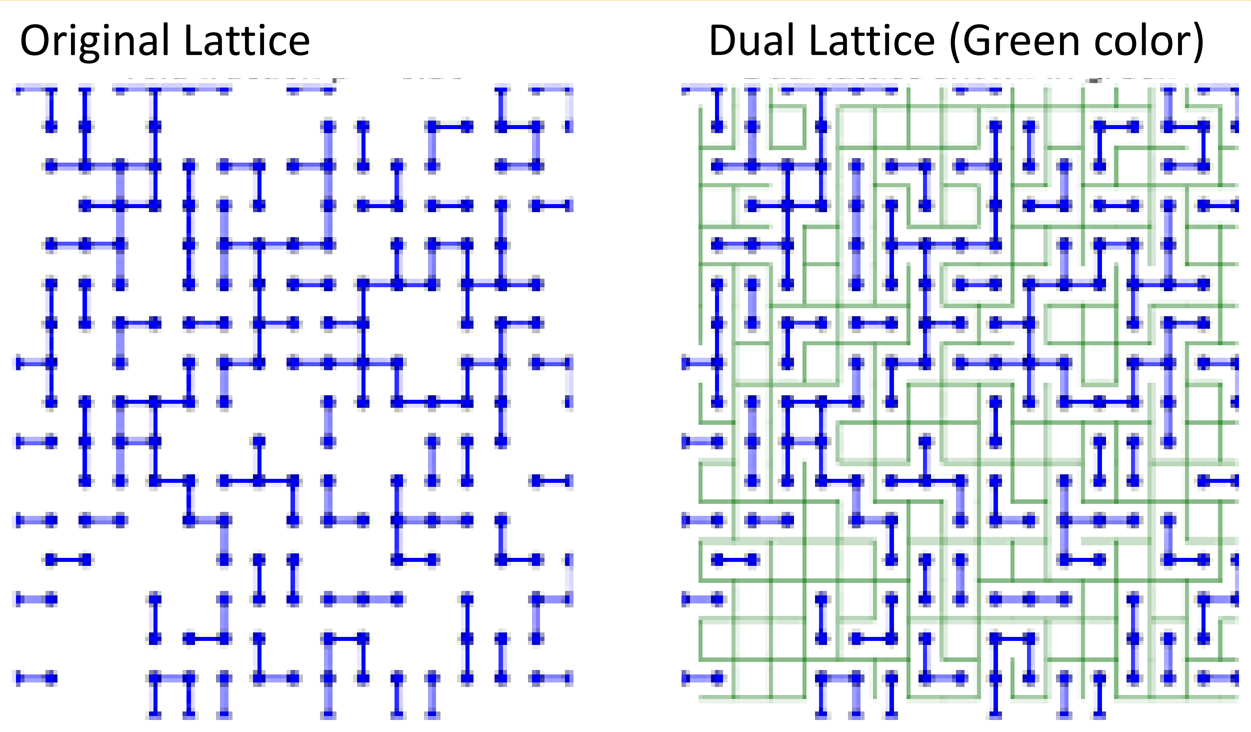


ave deg = 1.18

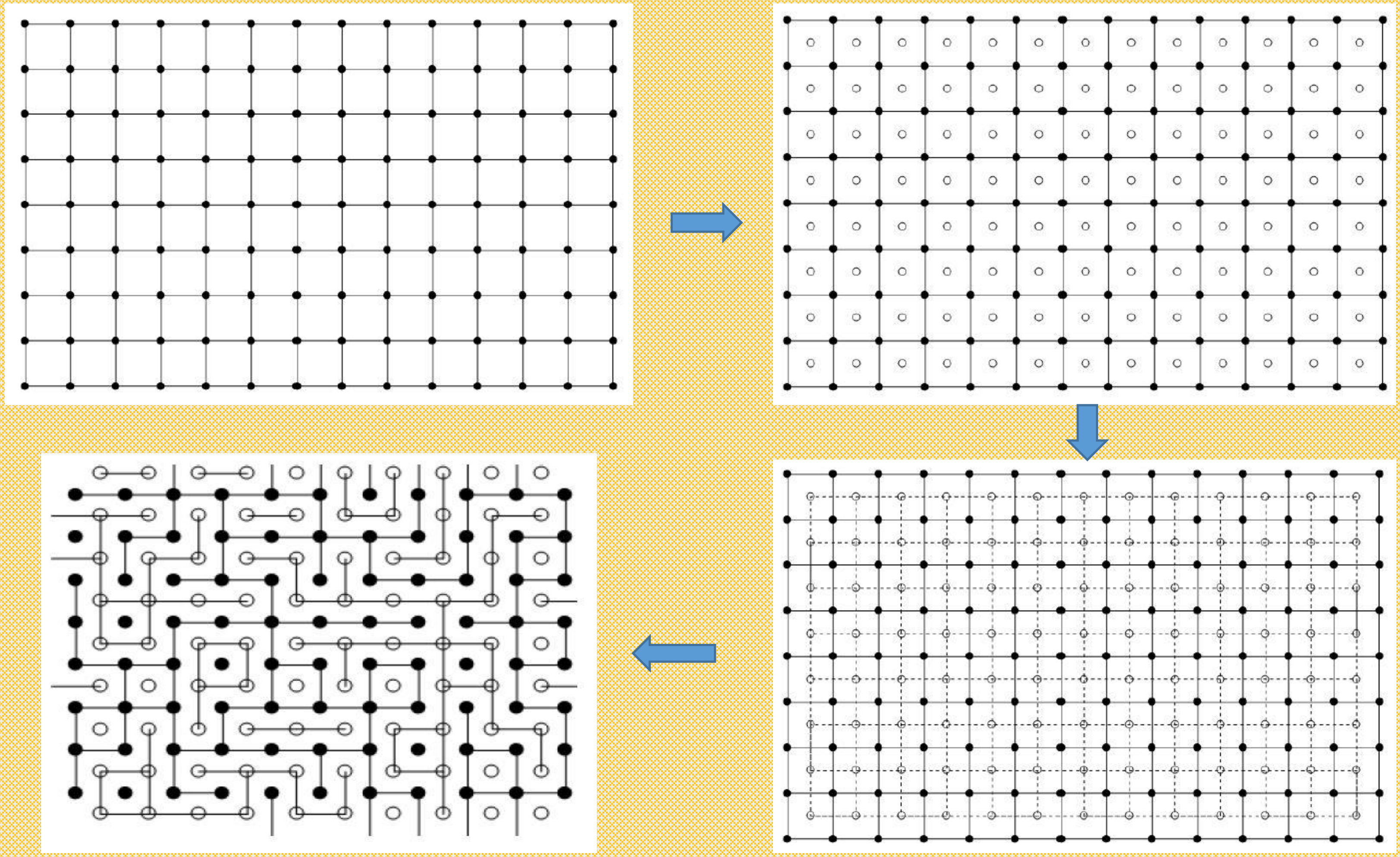


ave deg = 3.96

Dual Lattice can be used as a framework



Dual Lattice can be used as a framework



Inverse Problem: Given a lattice network

Node-attack: With probability p remove a node (namely, with probability $(1-p)$, not remove it), such that the liquid cannot flow through the network

Robustness Problem 1:

For what range of p values, there exist at least one path?

Robustness Problem 2:

For what range of p values, the lattice remains to be connected?

Robustness Problem 3:

After an attack, what is the size of the largest cluster?

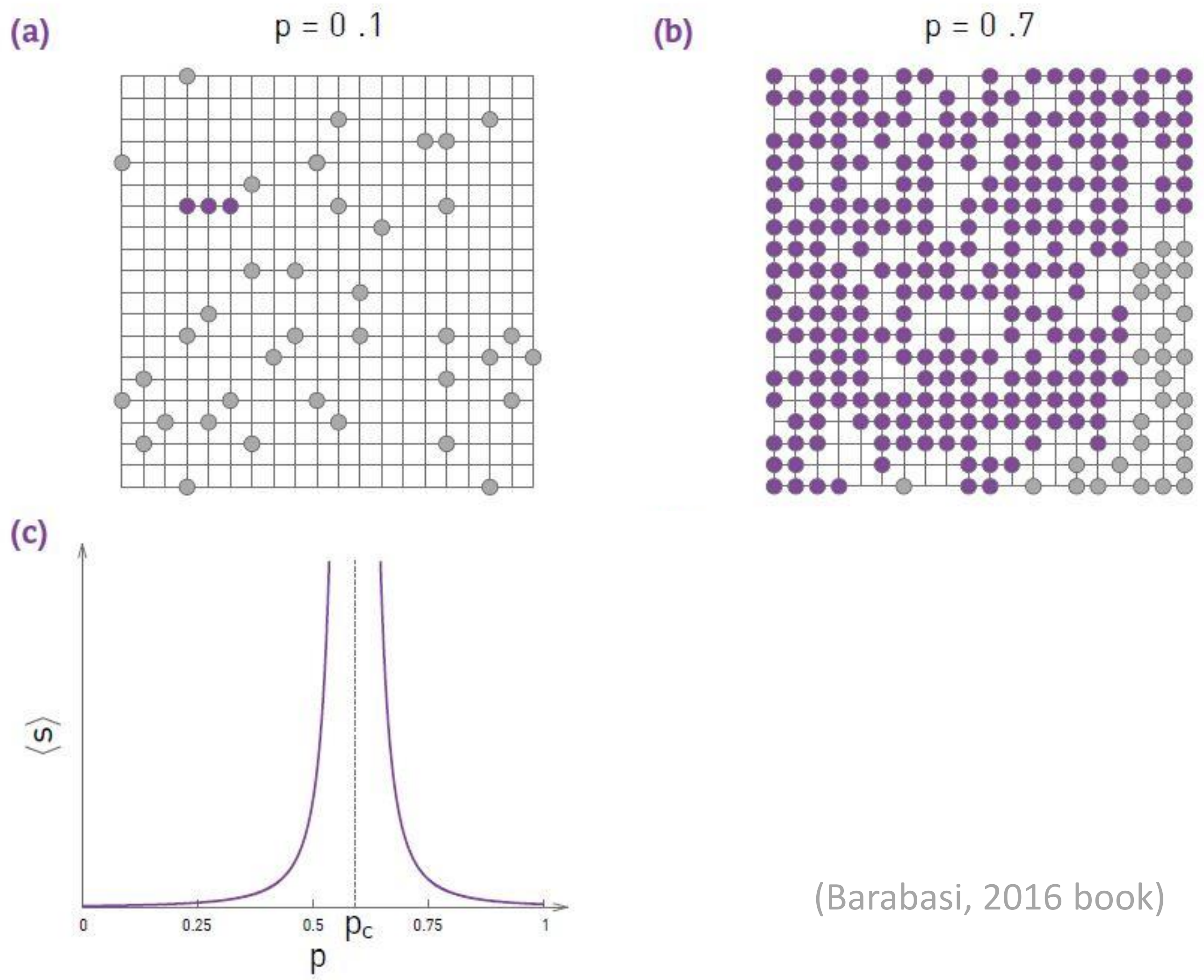
Average size of giant clusters:

$$\langle s \rangle \sim |p - p_c|^{-\gamma}$$

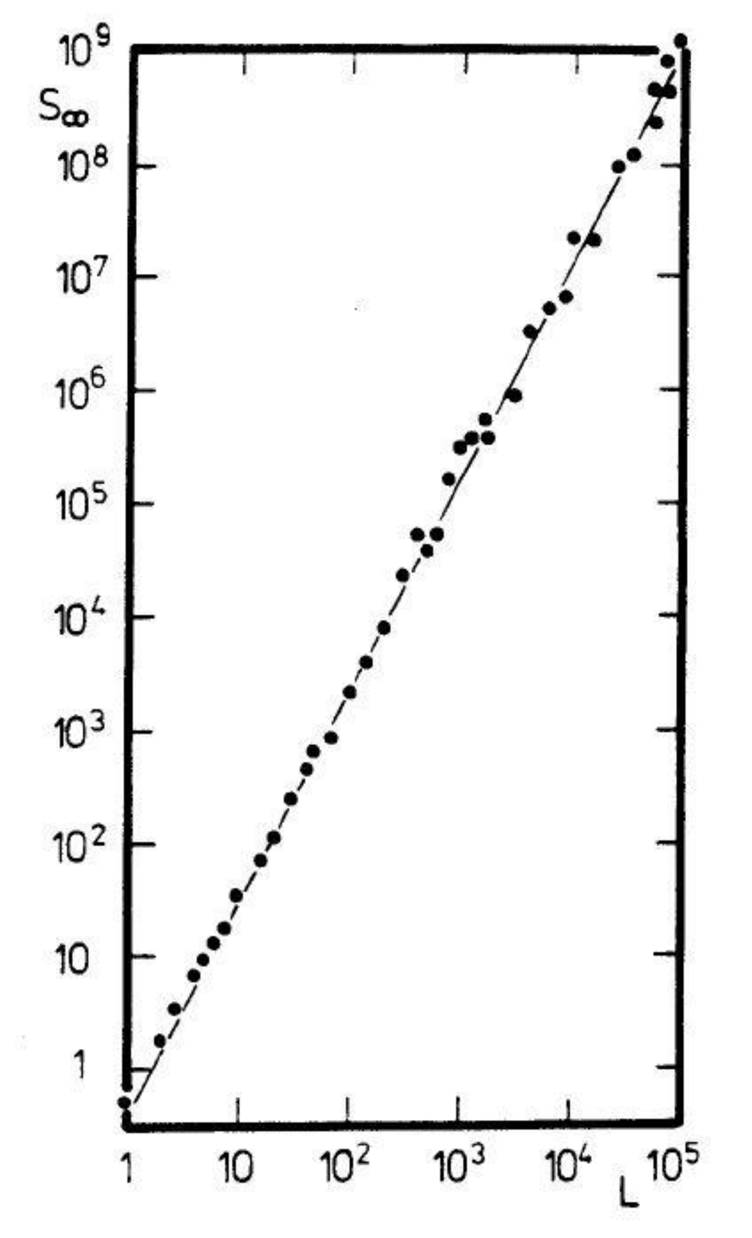
Constant γ is determined by the network

p far from p_c → $\langle s \rangle$ small

p near p_c → $\langle s \rangle$ large



(Barabasi, 2016 book)



Size of the largest cluster S_∞

versus lattice size L at $p = p_c$

$S_\infty \sim L^D$

D – dimension of lattice

Order Parameter:

Probability of a randomly picked cluster is a giant one:

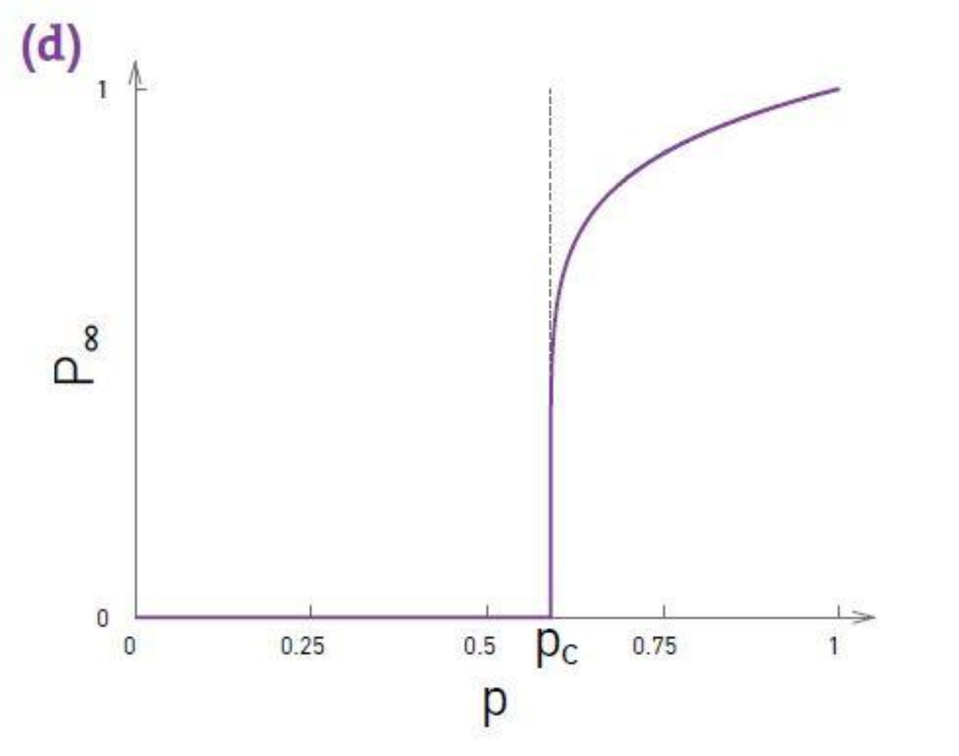
$$P_\infty \sim \begin{cases} (p - p_c)^\beta & p > p_c \\ \approx 0 & p \le p_c \end{cases}$$

$p =$ probability of occupying a node

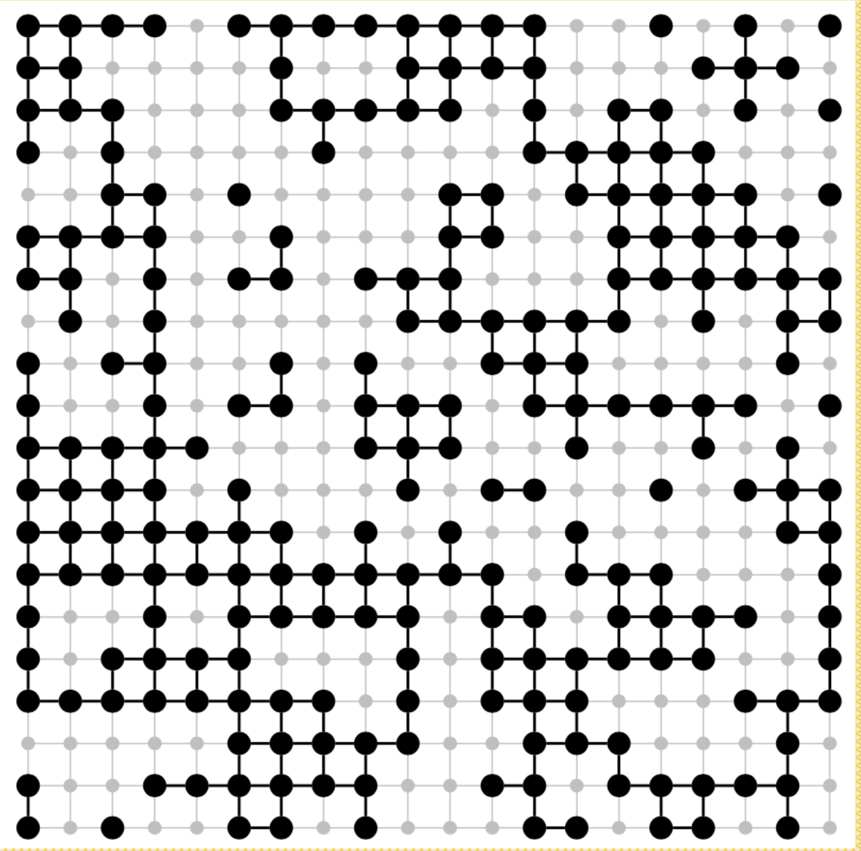
Constant $\beta > 0$ is determined by the network

$p \gg p_c \rightarrow P_\infty$ large $(\rightarrow 1)$

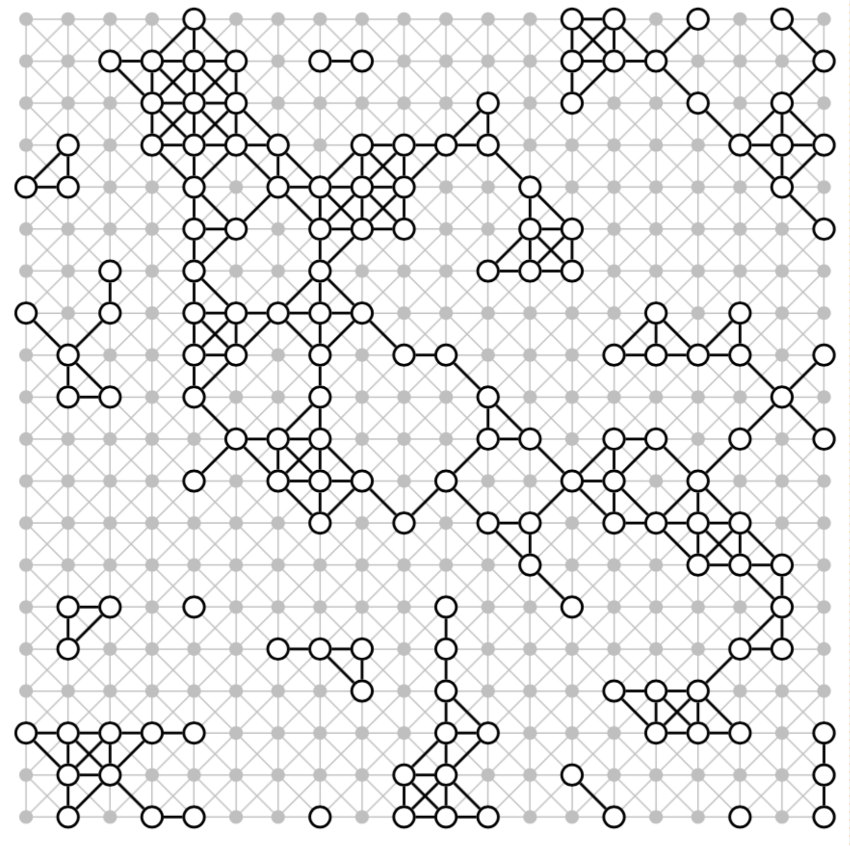
$p \ll p_c \rightarrow P_\infty$ small (≈ 0)



Complementary Lattices



p



$1 - p$

Order Parameter

(for Inverse Percolation) :

Probability of a randomly picked cluster is a giant one:

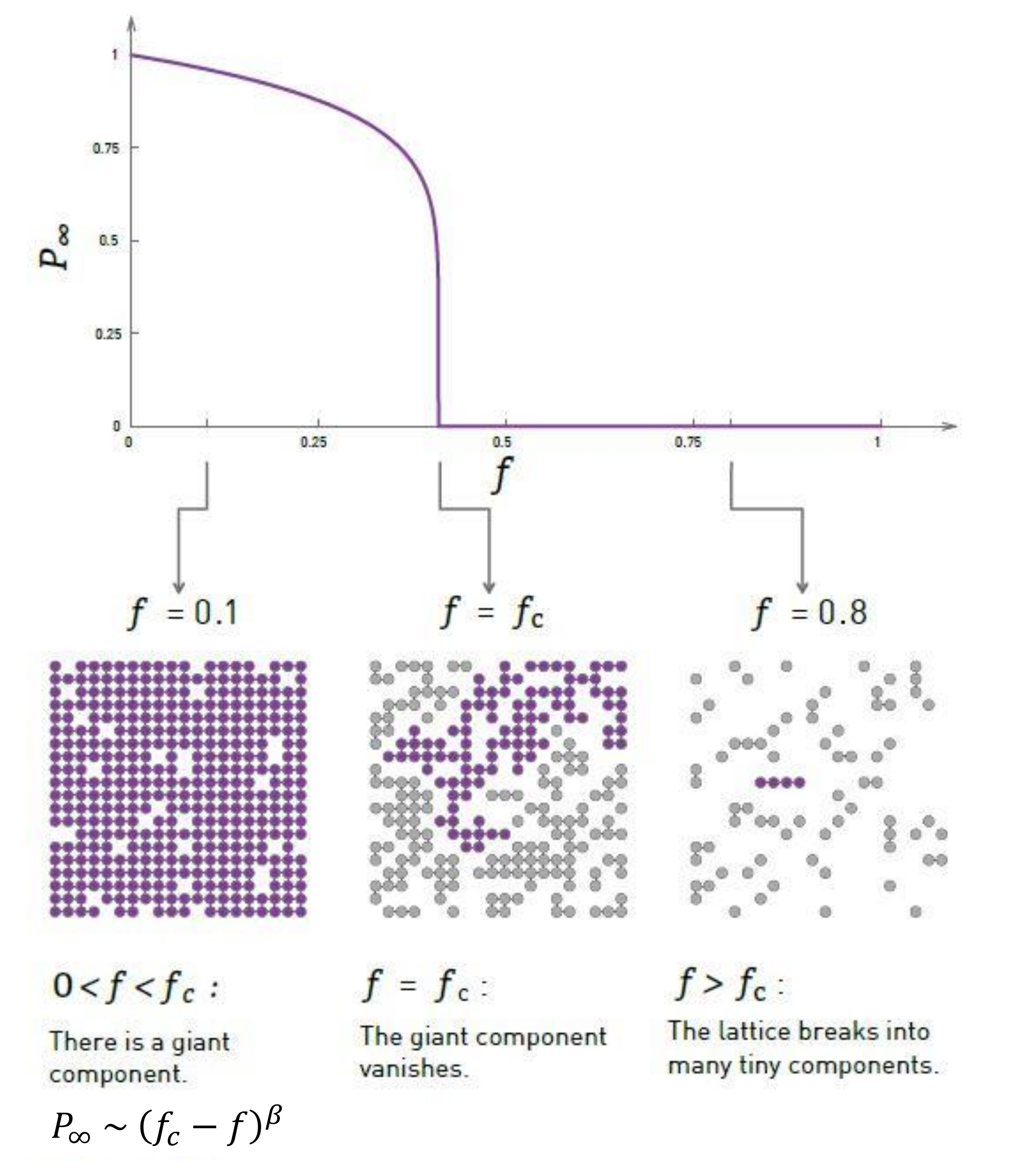
$$P_\infty \sim \begin{cases} (f - f_c)^\beta & f < f_c \\ \approx 0 & f \geq f_c \end{cases}$$

f = attacked nodes ($f = 1 - p$)

Constant $\beta > 0$ is determined by the network

$f \ll f_c \rightarrow P_\infty$ large ($\rightarrow 1$)

$f \gg f_c \rightarrow P_\infty$ small (≈ 0)

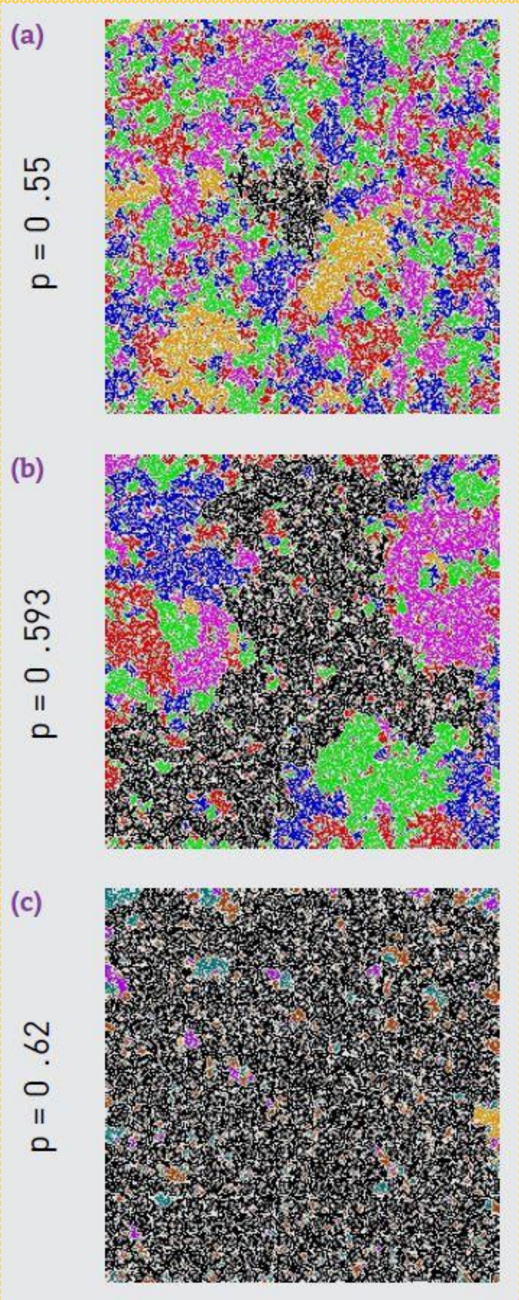


Example: Wild fire spreads on a forest

Suppose a forest is a lattice, where each square has at most one tree. With probability $p \in (0,1)$, place a tree into a square

One tree is on fired at random. This tree spreads fire to its neighboring trees (assuming the fire cannot spread over an empty square)

→ There exists a threshold value p_c



$p < p_c$ → not many trees → not many threes were burn (black squares)

$p = p_c$

$p > p_c$ → many trees → many threes were burn (black squares)

(Barabasi, 2016 book)

Molloy-Reed Index

Condition for existence of giant cluster: $\kappa := \frac{\langle k^2 \rangle}{\langle k \rangle} > 2$

Interpretation: Consider a cluster of k nodes, almost fully connected. Thus, every node has degree about $k-1$, so the cluster has $\frac{1}{2}k(k-1) \approx \frac{k^2}{2}$ edges

Consider the average of all such large clusters, which has $\left\langle \frac{k^2}{2} \right\rangle$ edges

Define Ratio:

(Average total edge of a large cluster) / (Average total edge of a node in the cluster)

$$\kappa' := \frac{\left\langle \frac{k^2}{2} \right\rangle}{\langle k \rangle}$$

(next page)

Molloy-Reed Index

Define a Ratio:

(Average total edge of a giant cluster) / (Average total edge of a node in the cluster)

$$\kappa' := \frac{\left\langle \frac{k^2}{2} \right\rangle}{\langle k \rangle}$$

If $\kappa' > 1$, then $\kappa := \frac{\langle k^2 \rangle}{\langle k \rangle} > 2$ → condition for giant cluster to exist

Because:

If $\kappa' = 1$, then $\langle k^2 \rangle = 2\langle k \rangle$ → $\langle k \rangle \approx 2$ → ring-shaped or chain-shaped

(far from being "almost fully connected" giant cluster)

If $\kappa' < 1$, then $\langle k \rangle < 2$ → star-shaped or broken pieces

(far from being "almost fully connected" giant cluster)

Random Graphs:

$$\langle k^2 \rangle = \langle k \rangle (\langle k \rangle + 1)$$

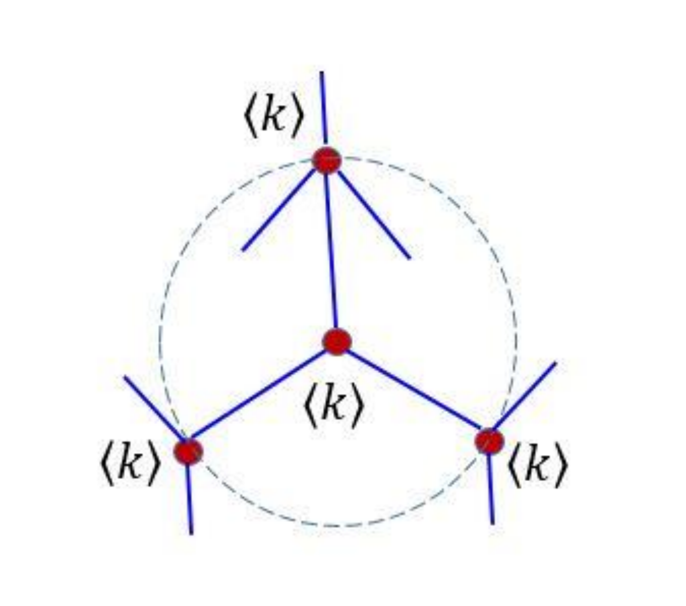
Interpretation:

Every node has degree $\langle k \rangle$, and every neighbor has degree $\langle k \rangle$ → total increasing degree is $\langle k^2 \rangle - \langle k \rangle$

Increasing rate of total degree: $\frac{\langle k^2 \rangle - \langle k \rangle}{\langle k \rangle}$

Since this is a linear growth: $\frac{\langle k^2 \rangle - \langle k \rangle}{\langle k \rangle} \sim \langle k \rangle$

→ $\langle k^2 \rangle = \langle k \rangle (\langle k \rangle + 1)$



Molloy-Reed Index

Condition for existence of giant cluster: $\kappa := \frac{\langle k^2 \rangle}{\langle k \rangle} > 2$

For **random networks**

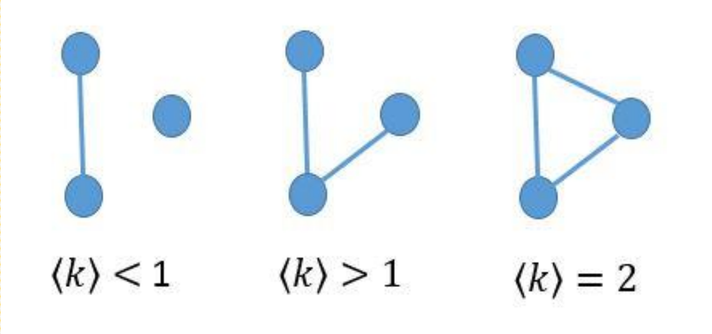
$\langle k^2 \rangle = \langle k \rangle (\langle k \rangle + 1)$

→

$$\kappa := \frac{\langle k^2 \rangle}{\langle k \rangle} = \frac{\langle k \rangle (\langle k \rangle + 1)}{\langle k \rangle} = \langle k \rangle + 1 > 2$$

→

$$\langle k \rangle > 1$$



This is a (necessary) condition for giant cluster to exist

General Complex Networks

Molloy-Reed Index →

$$f_c = 1 - \frac{1}{\frac{\langle k^2 \rangle}{\langle k \rangle} - 1}$$

(R. Cohen, K. Erez, D. ben-Avraham and S. Havlin. Phys. Rev. Lett., 2000)

where f = attacked nodes, f_c = threshold

Random Networks: $\langle k^2 \rangle = \langle k \rangle (\langle k \rangle + 1)$

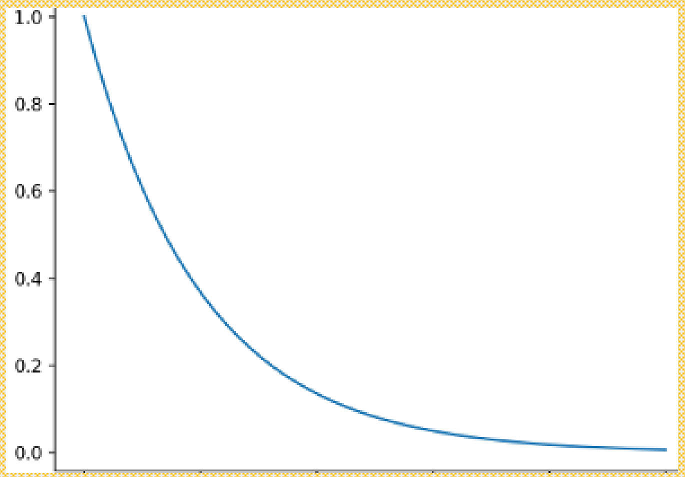
→

$$f_c = 1 - \frac{1}{\langle k \rangle}$$

$$f_c \to 1 \quad \text{as} \quad k \to \infty$$

→ Giant clusters disappear

Giant cluster



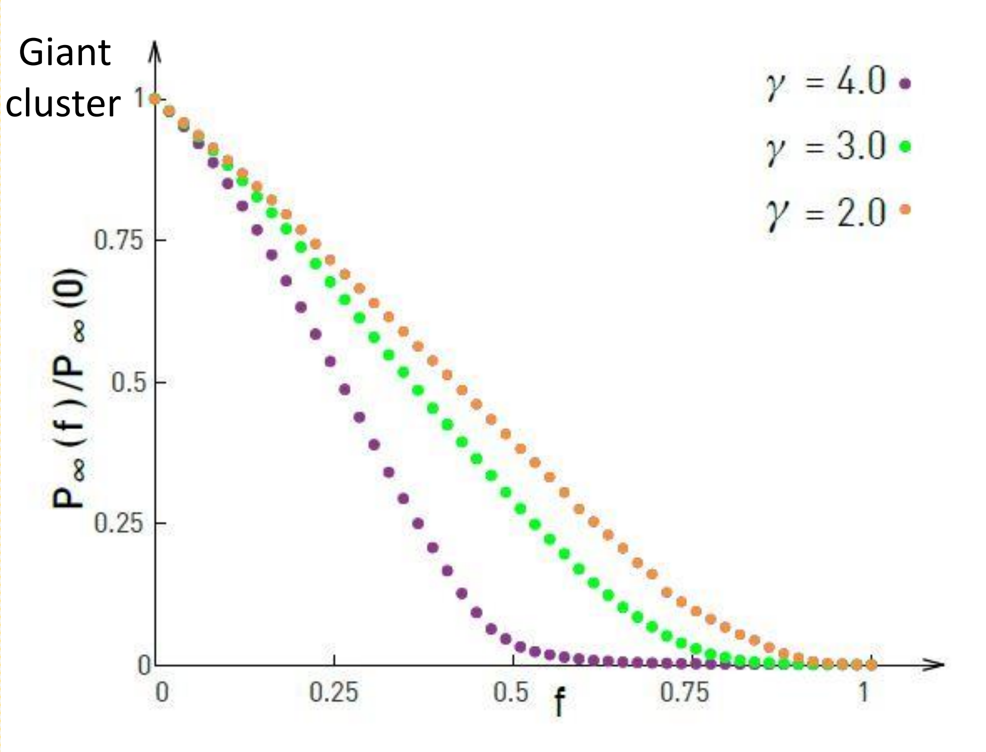
$f_c \to 1$ as $k \to \infty$

Scale-free Networks: $P_N(k) \sim k^{-\gamma}$

$$f_c = 1 - \frac{1}{\frac{\langle k^2 \rangle}{\langle k \rangle} - 1}$$

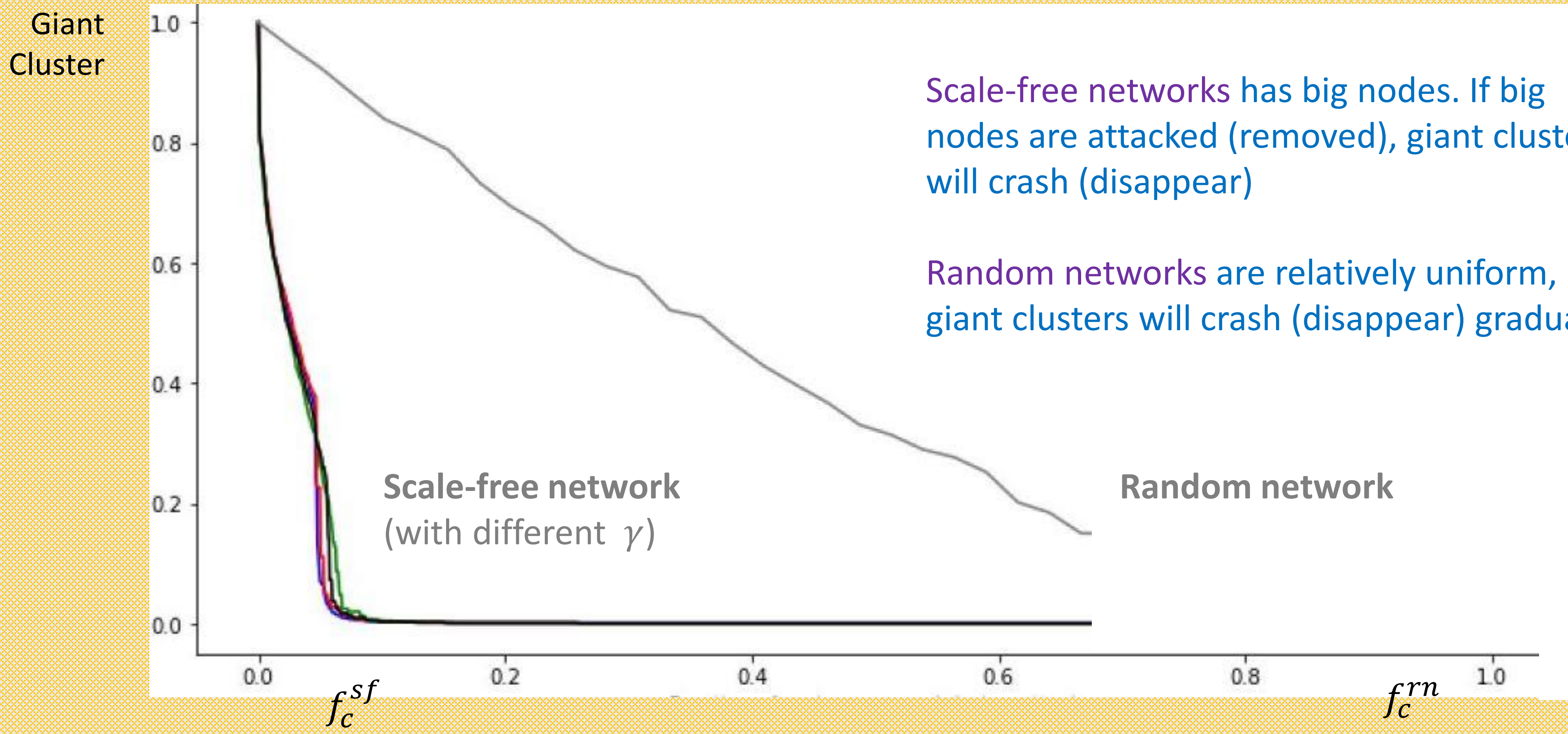
(Barabasi, 2016 book)

$$f_c = \begin{cases} 1 - \dfrac{1}{\frac{\gamma-2}{3-\gamma} k_{\min}^{\gamma-2} k_{\max}^{3-\gamma} - 1} & 2 < \gamma < 3 \\[2ex] 1 - \dfrac{1}{\frac{\gamma-2}{\gamma-3} k_{\min} - 1} & \gamma > 3 \end{cases}$$



$f_c \to 1$ as $k \to \infty$

Comparison



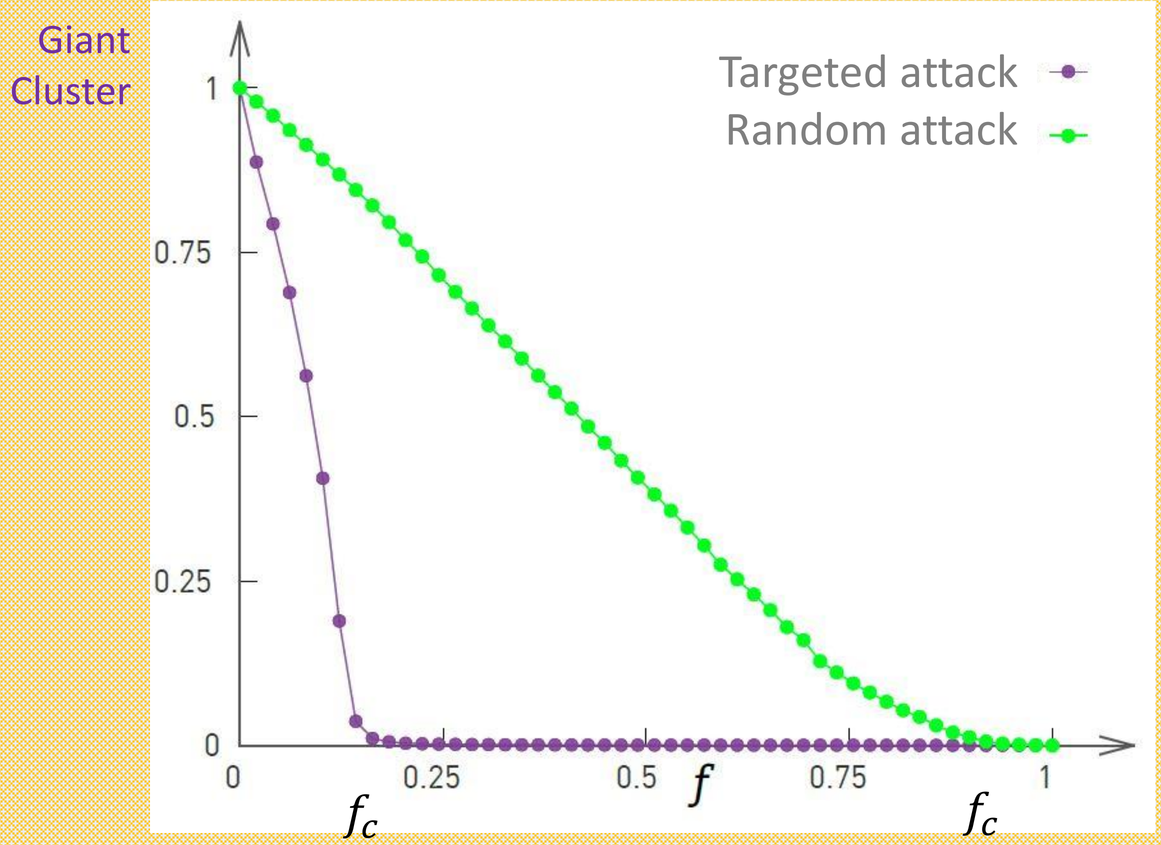
Scale-free networks has big nodes. If big nodes are attacked (removed), giant clusters will crash (disappear)

Random networks are relatively uniform, giant clusters will crash (disappear) gradually

Comparing Two Typical Attacks

Random attacks
(Random failures)

Targeted attacks
(Intentional removals)



Summary

Attack Strategies

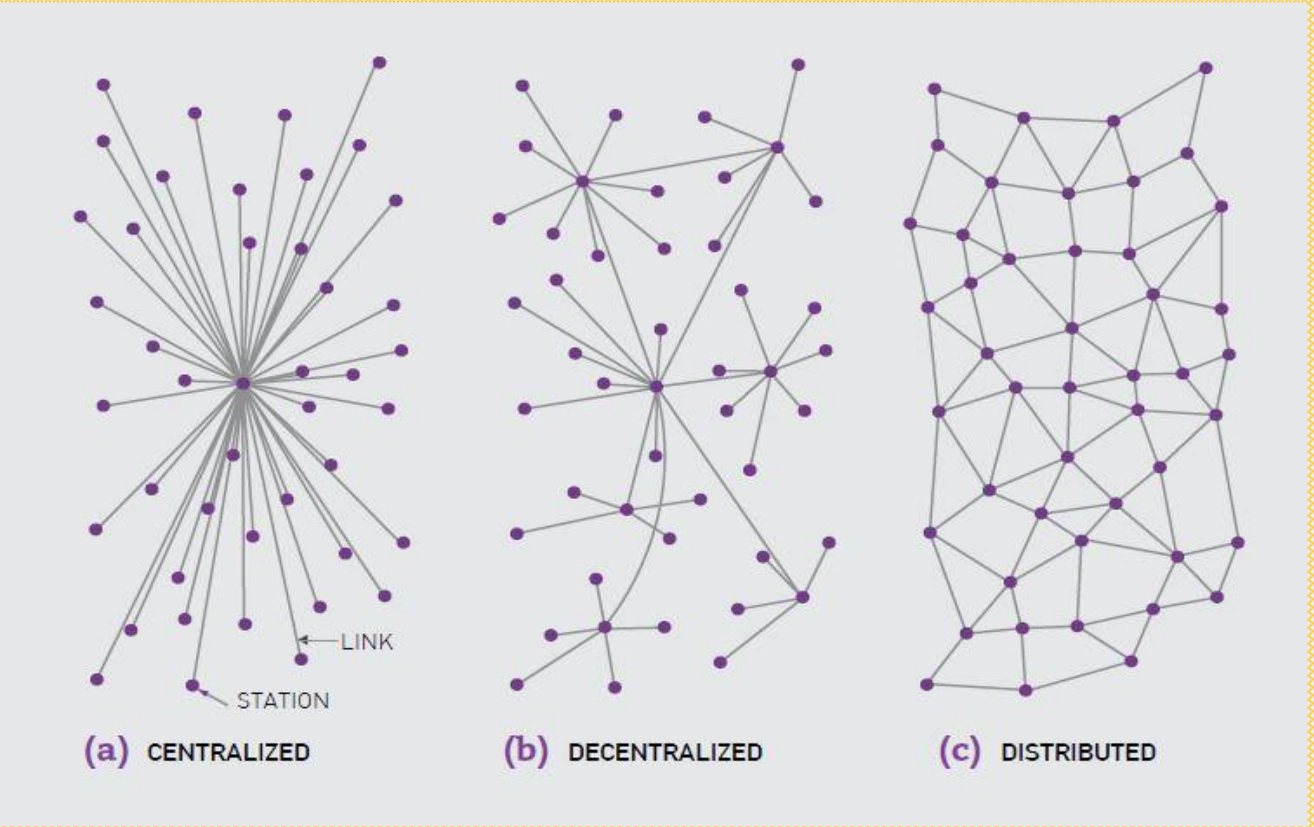
	Random	Targeted
Random Networks	Low cost	High cost Low effect
Scale-free Networks	Low cost Low effect	High cost High effect

Network Topologies

(a) Centralized: Low cost, Fragale

(b) Decentralized: Trade-off

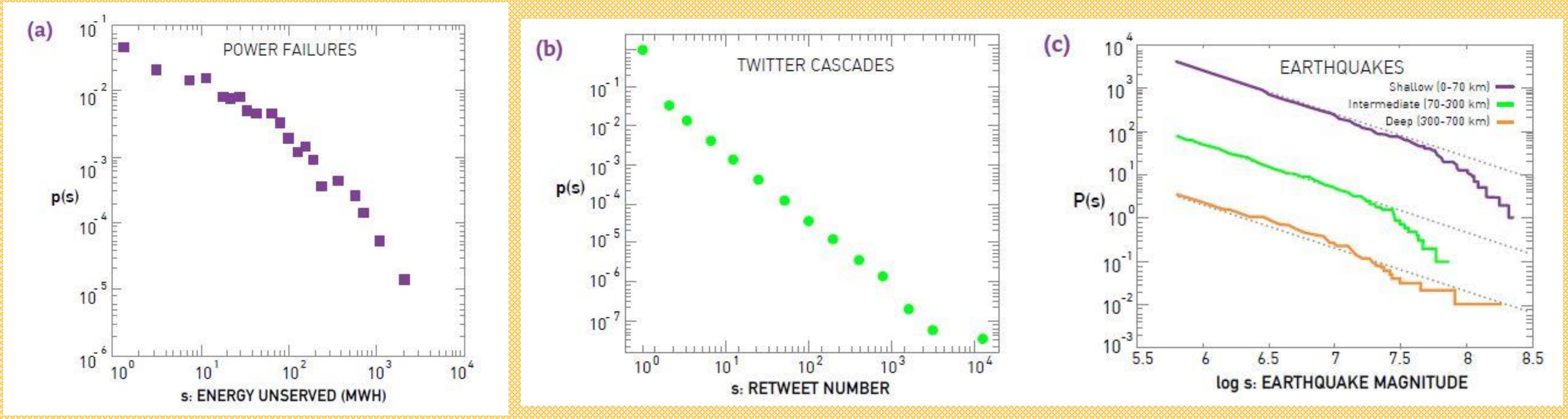
(c) Distributed: High cost, Robust



(Barabasi Book)

Building Robust Networks

* Power grid * Internet service denials * Financial crisis * Traffic jams * … …

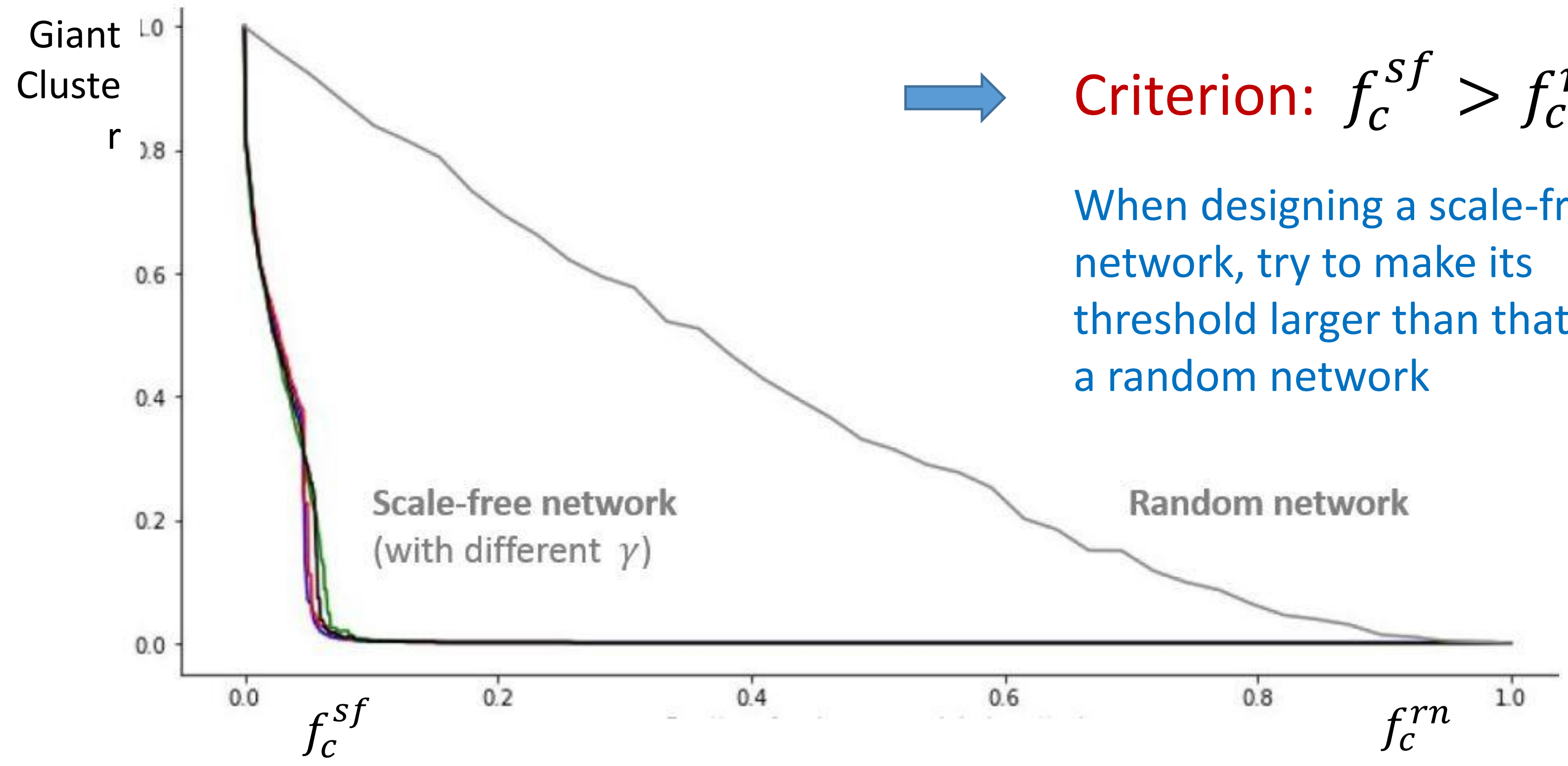


Power failures Data jams on Tritter Earthquakes

How to build robust power grid, Internet, transportation systems, … … ?

Criterion: Make its threshold as large as possible

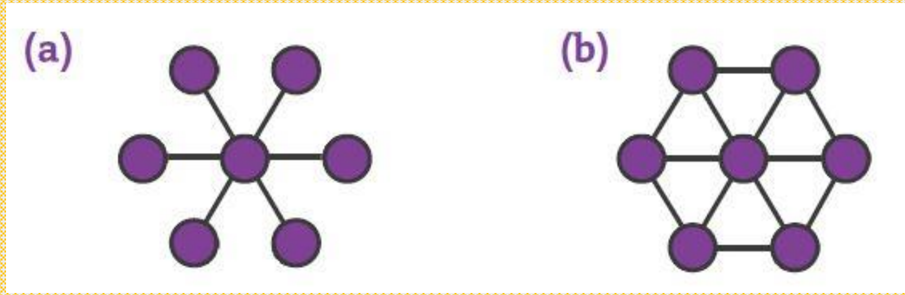
Recall:



Criterion: $f_c^{sf} > f_c^{rn}$

When designing a scale-free network, try to make its threshold larger than that of a random network

Which One is Better?



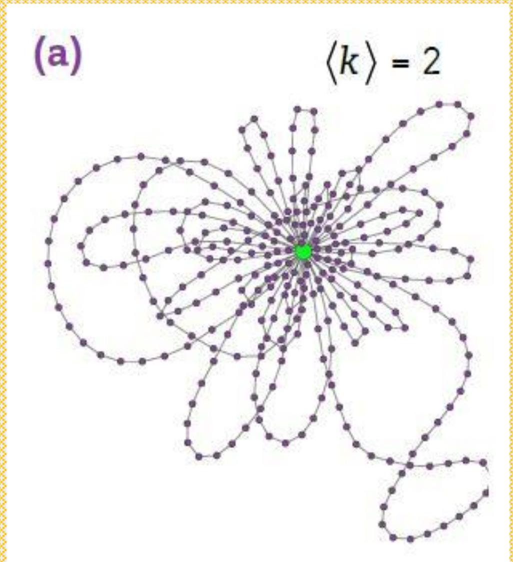
Constraint: Low cost

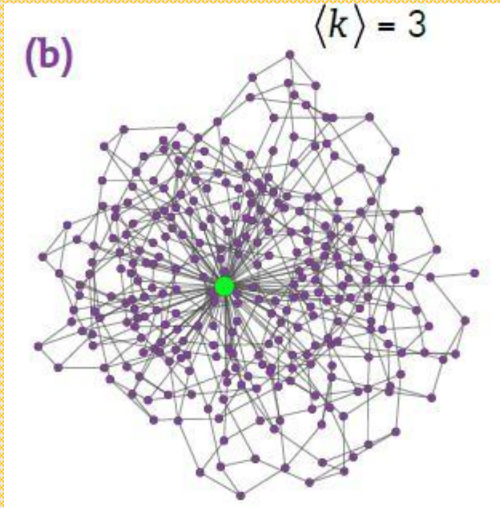
$$\min f_c^{tot} = f_c^{rand} + f_c^{tar}$$

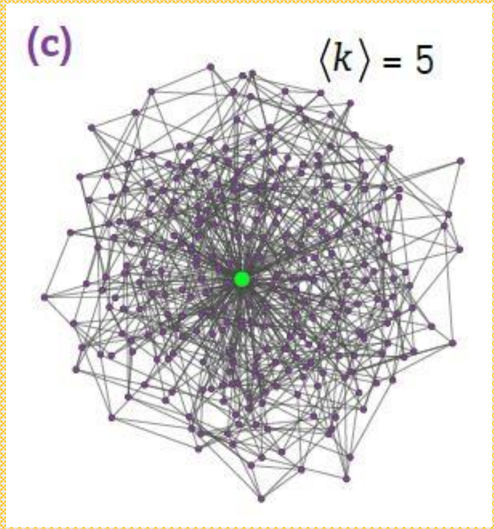
(tot = total, rand = random, tar = targeted)

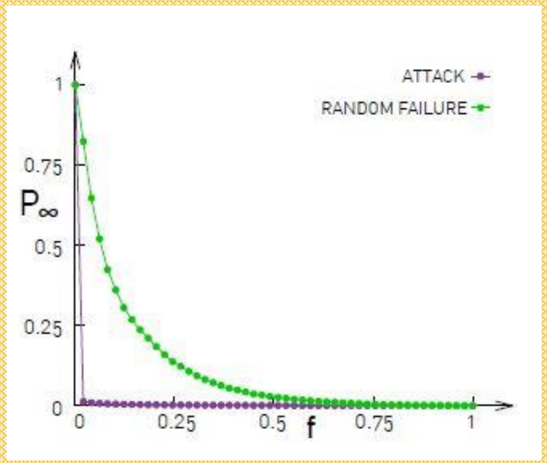
This is an Optimization problem

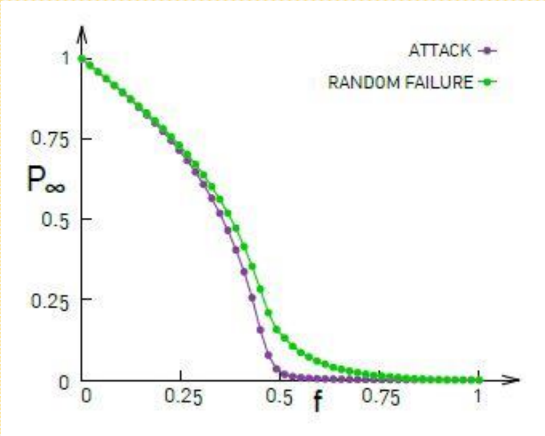
Example: Optimization (of Robustness vs Cost)

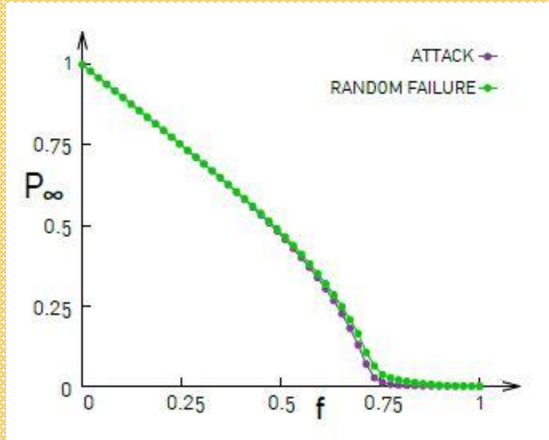












→ More robust but More costly →

END