Module 03: Network Robustness

Clustering Coefficient (3)

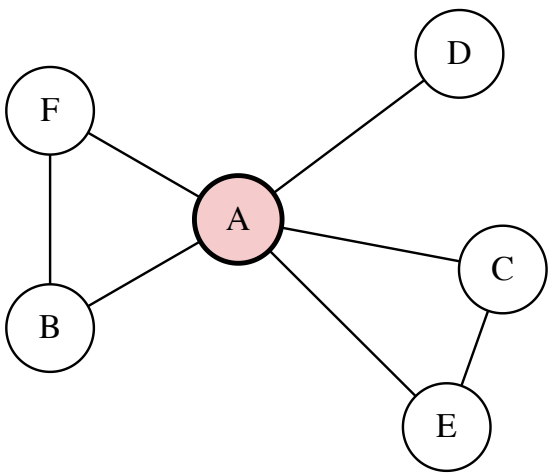
Global clustering coefficient focuses on the total number of triangles in the network.

$$C = \frac{3 \times \text{number of triangles}}{\text{number of connected triplets}} = \frac{3 \times \text{number of triangles}}{\sum_i k_i(k_i - 1)/2}$$

where k_i is the degree of node i.

Quiz:

- Compute the global clustering coefficient and the average path length of the following network.



- Let's define the ratio of the clustering coefficient to the average path length as the small-worldness index. What is the problem with this definition?

Learning Objectives

- **MST**: Minimum spanning trees and network design
- **Robustness**: Random failures vs. targeted attacks
- **Theory**: Percolation theory and connectivity
- **Design**: Robust networks balancing efficiency and resilience

Keywords: MST, Kruskal's algorithm, Prim's algorithm, percolation, R-index, robustness paradox

Pen & Paper Exercise

Post-WWI Czechoslovakia Challenge

- Connect all towns with electricity
- Limited resources for infrastructure
- Minimize cable length
- First systematic solution by Otakar Borůvka (1899-1995)

? **Question**: How would you connect all towns with the minimum total cable length?

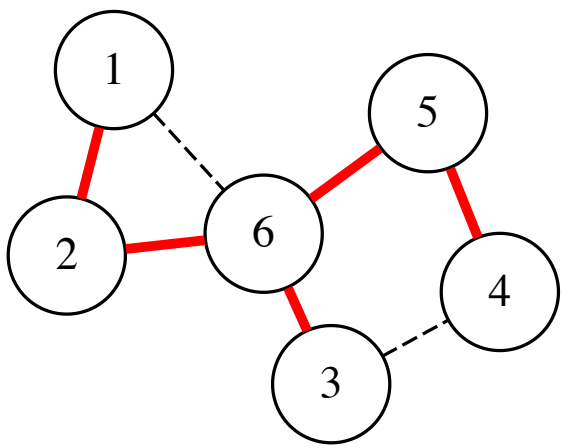
Take 30 seconds to think about your approach…

💡 Key Insight

Key Insight: We need a **tree** that connects all nodes with minimum total weight.

Why a tree?

- No redundant connections (cycles)
- Every node connected exactly once
- Minimum possible edges for full connectivity

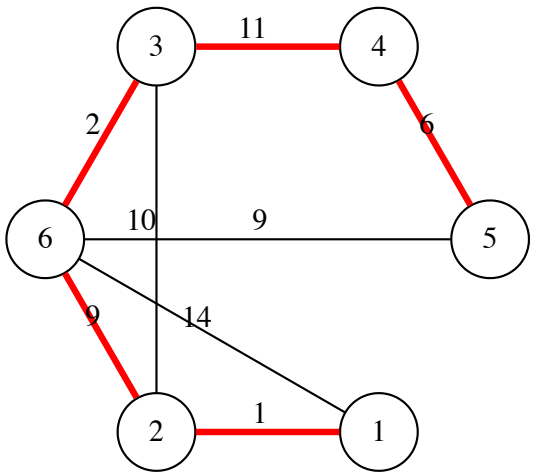


Red edges form a tree. Dashed edges are redundant.

Minimum Spanning Tree (MST)

A **minimum spanning tree (MST)** of a weighted network is a tree that:

- **Spans** all nodes (connects every location)
- Is a **tree** (no cycles, no redundant loops)
- Has **minimum total weight** among all spanning trees



🤔 Algorithm Design Question

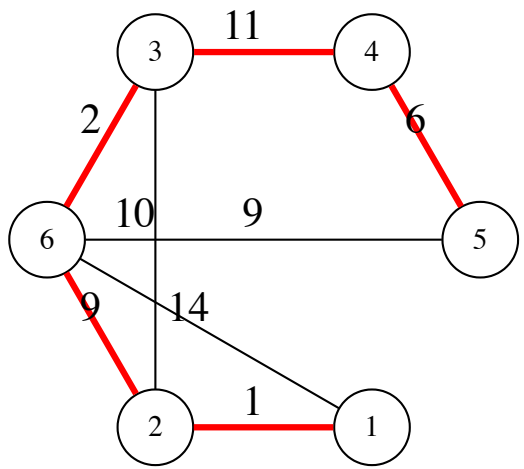
❓ **Question**: Given that we want the minimum spanning tree, what strategy would you use to build it?

Think about: Should we start with the cheapest connections? What if adding a cheap connection creates a loop?

💡 The Answer: Kruskal's Algorithm

Intuition: "Choose cheapest option, avoid wasteful loops"

1. Sort edges by weight (cheapest first)
2. Add edges in order
3. Skip if creates cycle
4. Continue until all connected

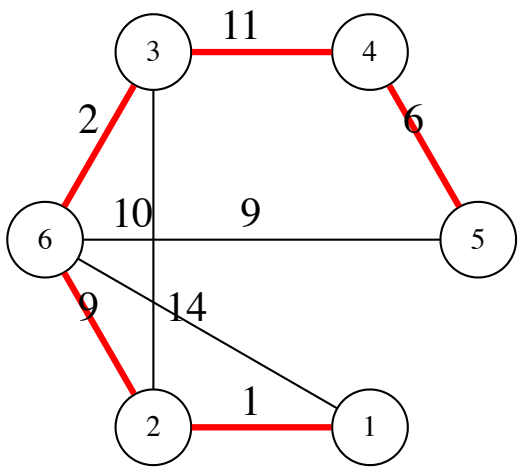


Global Perspective: Considers all connections simultaneously

💡 Prim's Algorithm

Intuition: "Organic growth from starting point"

1. Start from any node (power plant)
2. Find cheapest connection to an unconnected node
3. Add edge, mark node as connected
4. Repeat until all connected



Local Perspective: Builds incrementally from existing network

Kruskal vs. Prim

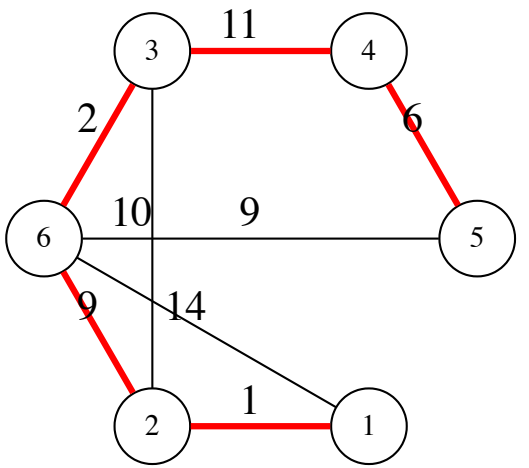
A demo in lecture notes

💬 **Discussion**: When do they find the same MST? When not?

🤔 The Vulnerability Problem

What happens if a single node in our MST fails?

Think about: How many towns would lose power? What does this mean for real infrastructure?

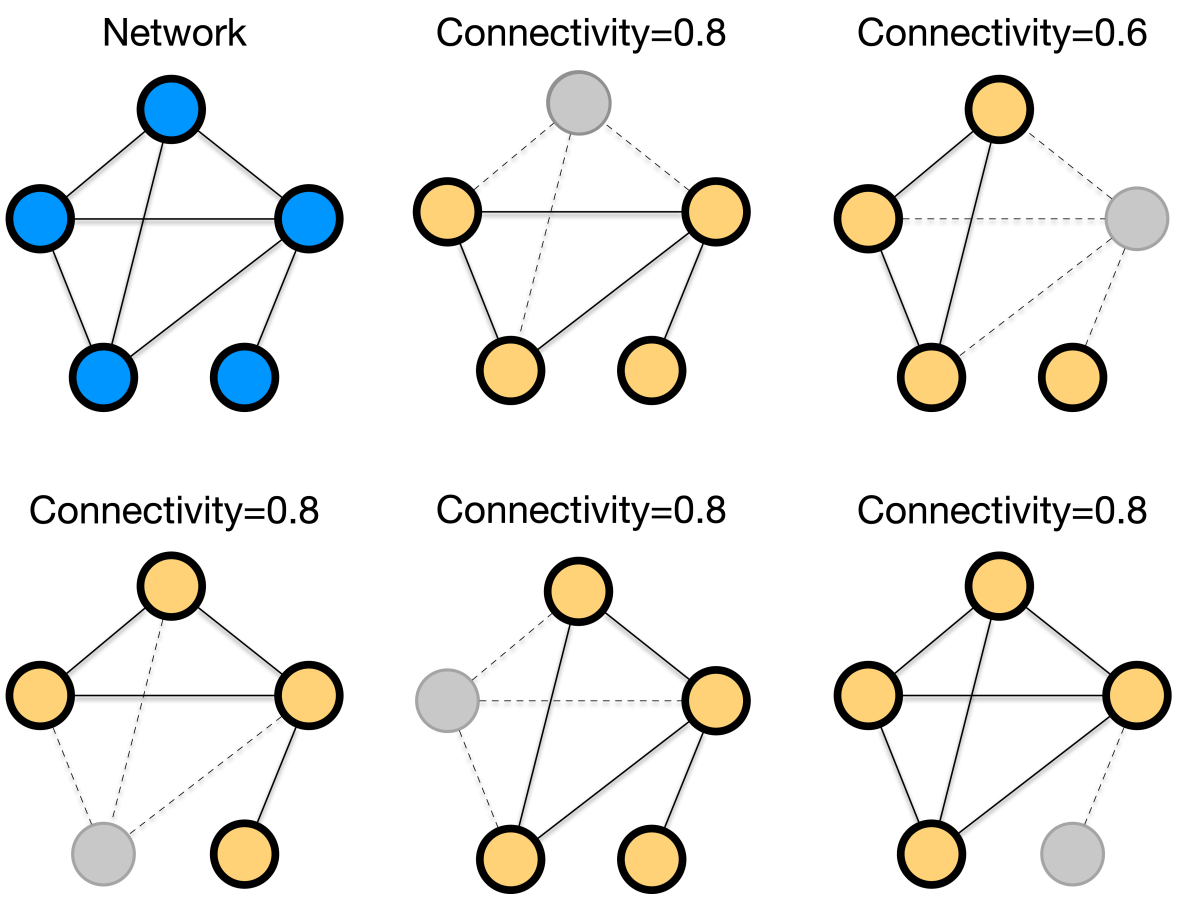


Cost efficiency ≠ Robustness

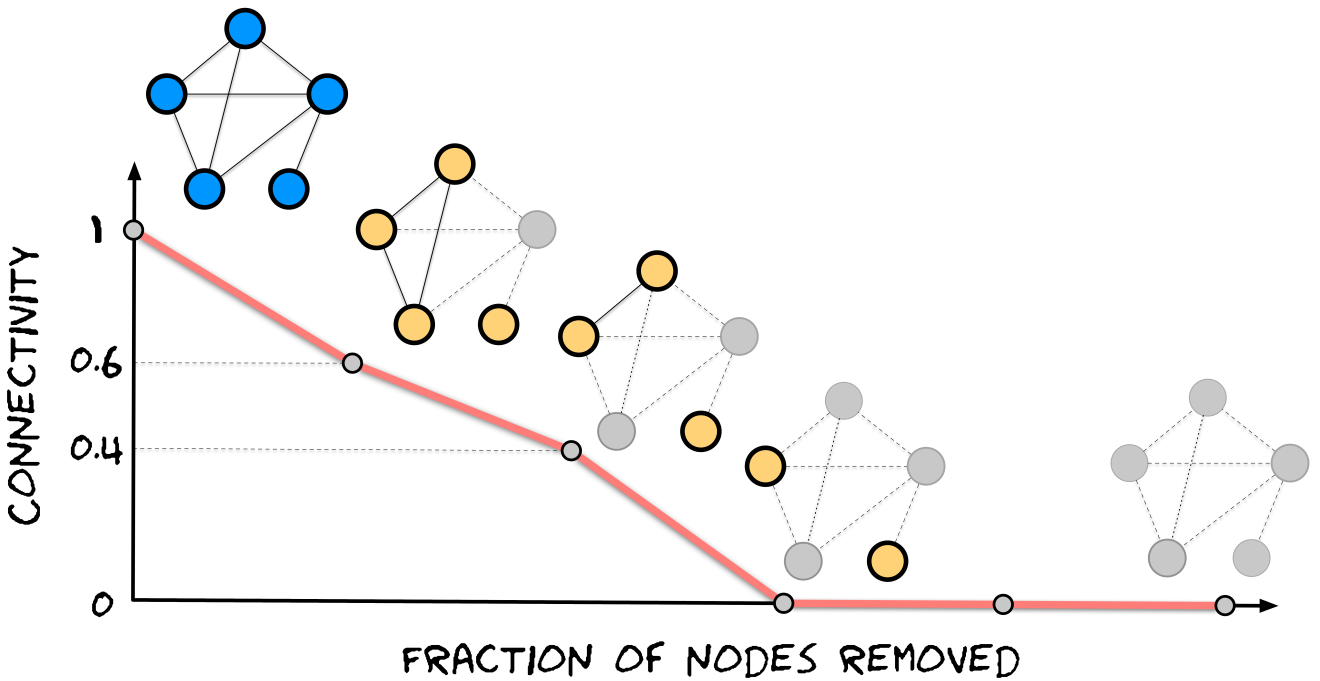
💬 **Discussion**: How would you modify the network to make it more robust? What are the trade-offs between cost and reliability?

Measuring Network Connectivity

$$\text{Connectivity} = \frac{\text{Size of largest component after removal}}{\text{Original network size}}$$



Robustness Profile: Connectivity vs. fraction of nodes removed



R-index: Area under robustness curve

$$R = \frac{1}{N} \sum_{k=1}^{N-1} y_k$$

🤔 Attack Strategies

Random Failures:

- Unpredictable events
- Earthquakes, equipment malfunctions
- Technical problems, random server crashes
- Characterized by equal probability of failure for all nodes

Targeted Attacks:

- Strategic node removal by adversaries
- Target busiest airports to disrupt air travel
- Characterized by removing highest-degree nodes, then next highest

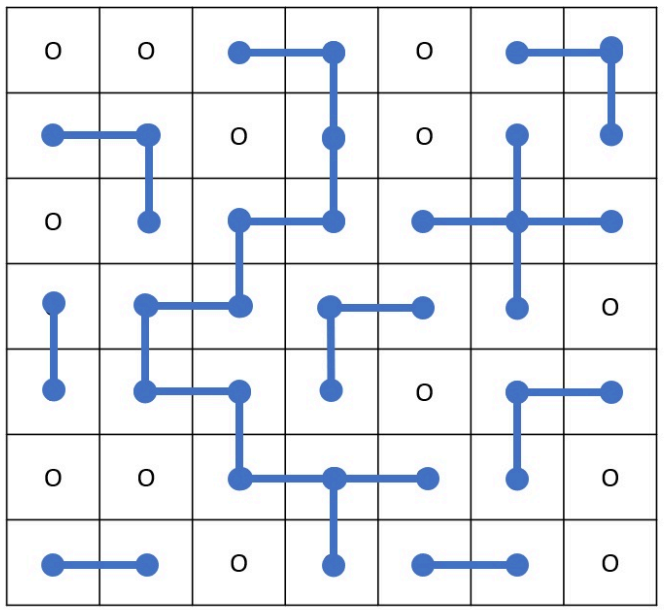
Game Time!

Interactive game

- Think about a structure that is robust to random failures/targeted attacks/both
- Why might the same network structure respond so differently to these two scenarios? What does this tell us about network design?

Percolation Theory

- Suppose puddles are scattered on the ground.
- As rain falls, puddles begin to merge with nearby puddles.
- At first, most puddles are isolated; as more rain falls, clusters form, and eventually a giant puddle spans a large part of the area.



$p = 0.4$
$p_{act} = 27 / 84 = 0.32$
$p_{max} = 12 / 27 = 0.44$

$p_{thresh} = 0.5$ (bond)

Percolation theory studies how local connections (like merging puddles) lead to sudden global connectivity.

Lecture notes for demonstration

Phase Transition

Critical Point (p_c): Threshold where giant component emerges/disappears

2D Lattice Example: $p_c \approx 0.593$

Lecture notes for demonstration

The sharp transition around p_c demonstrates a **phase transition**, i.e., a sudden change from a disconnected to connected state as we cross the critical threshold.

Network Robustness - Where is the critical point?

🤔 Let's Think Like Network Scientists

We've seen that networks can suddenly lose connectivity as nodes are removed.

Can you predict this critical point based on the network's structure alone, without running simulations?

Imagine you're a node in a network. For you to be part of the largest connected component, what do you need?

- You need **friends** (connections)
- But is that enough?
- What else do your friends need?

Take 3 mins to think about it…

Friends of Friends

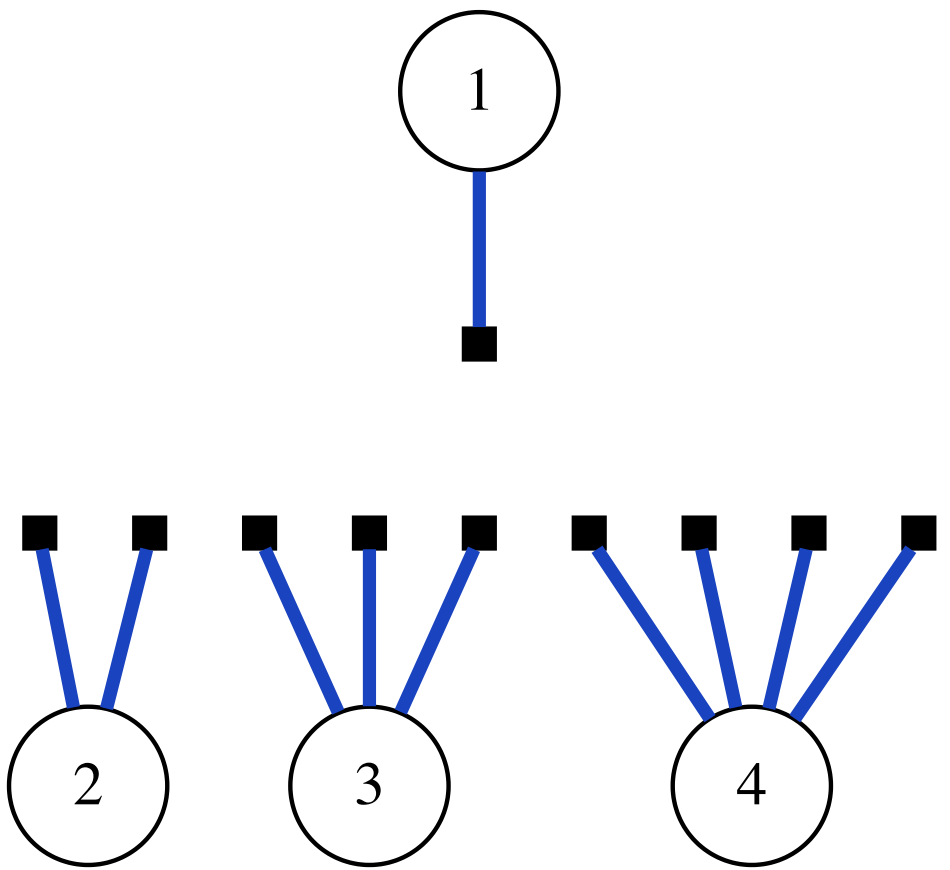
You have some friends. For you to be connected to the rest of the network through them, what must your friends have?

- Each friend needs at least **2 connections**
- One connection: **to you**
- Another connection: **to someone else** (to reach the rest of the network)
- *So what matters is not just how many friends you have, but how many friends your friends have!*

Average \# of friends that friends have ≥ 2

Let's formalize this insight. *Take 2 mins to think about it…*

1. Suppose you are node 1 with one hand
2. Other nodes have k hands with probability $p(k)$
3. When you handshake with a randomly picked **hand** (not node), how many hands do your friends would have on average 🤔?



There are $p(k)$ nodes with k hands. The total number of hands that nodes with k hands have is $kp(k)$.

If I randomly pick a hand, I would handshake with a node with k hands with probability proportional to $kp(k)$. That is

$$q(k) = \frac{kp(k)}{\sum_{k=1}^{\infty} kp(k)} = \frac{k}{\langle k \rangle} p(k),$$

where $\langle k \rangle = \sum_{k=1}^{\infty} kp(k)$ is the average degree. Now, the average number of hands that my friends would have on average is

$$\langle k \rangle_{q(k)} = \sum_{k=1}^{\infty} kq(k) = \sum_{k=1}^{\infty} k^2 / \langle k \rangle p(k) = \frac{\langle k^2 \rangle}{\langle k \rangle}.$$

🎯 The Molloy-Reed Criterion

You just discovered the **Molloy-Reed Criterion**!

$$\kappa = \frac{\langle k^2 \rangle}{\langle k \rangle} > 2$$

When is the κ smallest 🤔? What is the implication in terms of the robust network structure against random failures?

Take 2 mins to think about it…

- **High κ**: Hub-dominated networks (some nodes have many friends)
- **Low κ**: Degree homogeneous networks (everyone has similar friends)
- **Implication**: Hubs make networks more robust!

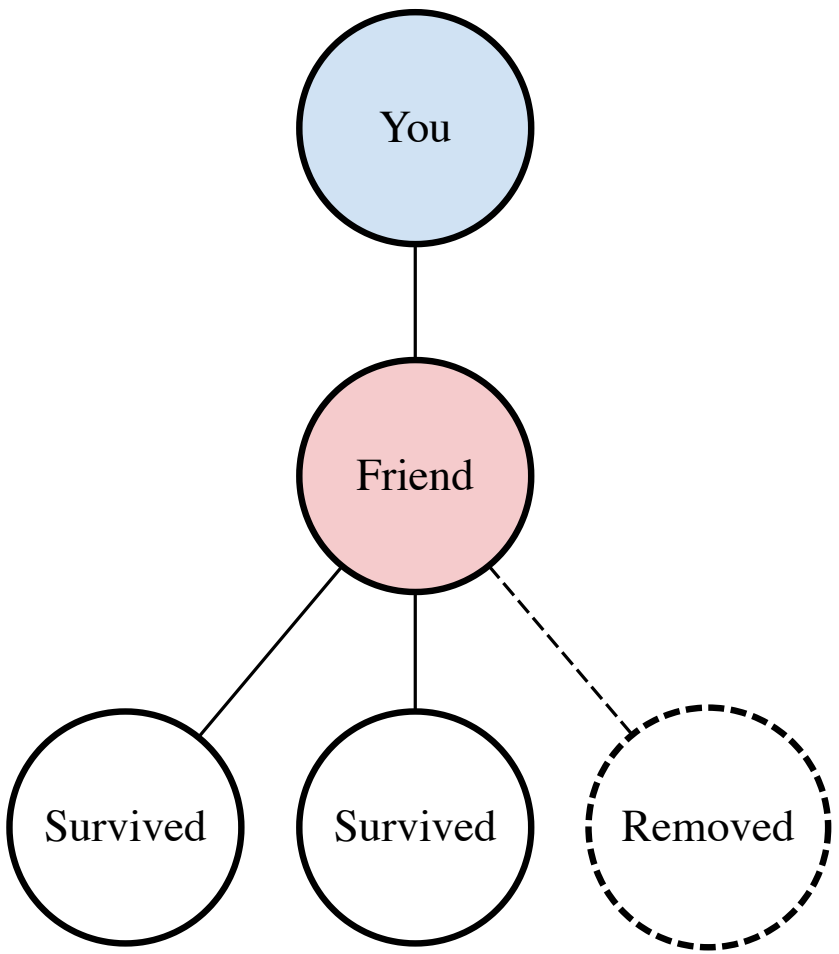
From κ to Critical Fraction

Now let's figure out: **What fraction of nodes can we remove before the network fragments?**

Hint: After removing a fraction f of nodes randomly, what happens to the number of friends a friend would have?

And when does the Molloy-Reed criterion break?

Take 3 mins to think about it.



- After removing fraction f, my friend who initially has—κ friends on average—have $(1 - f)(\kappa - 1)$ friends on average.
- It's $\kappa - 1$ not κ because one of the friends is me.
- Thus, the network breaks when $(1 - f)(\kappa - 1) = 1$
- Solving for f, we get

$$f_c = 1 - \frac{1}{\kappa - 1}$$

🤔 Case Study: Degree-Homogeneous Networks

Let's consider a degree-homogeneous network, where the degree distribution is Poisson with mean λ, i.e.,

$$k \sim \text{Poisson}(\lambda)$$

What is the critical point f_c?

Hint:

- $f_c = 1 - \frac{1}{\kappa - 1}, \kappa = \frac{\langle k^2 \rangle}{\langle k \rangle}$
- The mean and variance of the Poisson distribution are identical.

For a Poisson distribution with mean λ:

- The first moment (mean degree) is $\langle k \rangle = \lambda$
- The second moment is $\langle k^2 \rangle = \lambda^2 + \lambda$

Now, we can calculate κ:

$$\kappa = \frac{\langle k^2 \rangle}{\langle k \rangle} = \frac{\lambda^2 + \lambda}{\lambda} = \lambda + 1$$

Finally, we can find the critical fraction f_c:

$$f_c = 1 - \frac{1}{\kappa - 1} = 1 - \frac{1}{(\lambda + 1) - 1} = 1 - \frac{1}{\lambda}$$

Key Insight: Higher average degree makes the network more robust.

🤔 Case Study: Degree-Heterogenous networks

❓ **Question**: What happens to network robustness when we have a few very highly connected nodes (hubs) and many poorly connected nodes?

Think about: How would random failures affect this type of network? What about targeted attacks?

💡 The Answer: Scale-Free Networks (Power Law)

Degree Distribution:

$$P(k) \sim k^{-\gamma}$$

Two Regimes:

- **$2 < \gamma < 3$**: $f_c \to 1$ (extremely robust to random failures)
- **$\gamma > 3$**: Finite f_c (moderate robustness)

Critical Fraction:

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

Key Insight: Scale-free networks are remarkably robust to random failures

💬 **Discussion**: Why are scale-free networks so robust to random failures? What's the intuition behind this counterintuitive result?

🤔 The Achilles' Heel Question

❓ **Question**: If scale-free networks are so robust to random failures, what's their weakness?

Hint: Think about what happens when you deliberately target the most important nodes…

💡 The Answer: Vulnerability to Targeted Attacks

Vulnerability: Scale-free networks fragile under targeted attacks

Mathematical Analysis:

$$f_c^{\text{attack}} \ll f_c^{\text{random}}$$

Critical Attack Threshold:

$$f_c^{\frac{2-\gamma}{1-\gamma}} = \frac{2+2^{-\gamma}}{3-\gamma} k_{\min} \left(f_c^{\frac{3-\gamma}{1-\gamma}} - 1 \right)$$

Real-world Implications:

- Power grids protect major substations
- Internet infrastructure includes hub redundancy
- Transportation networks maintain backup routes

💬 **Discussion**: This creates a fundamental dilemma - how do you design a network that's both efficient and secure? What strategies might you use?

Part VII: Design Principles

🤔 Design Challenge

❓ **Question**: Given what we've learned about network vulnerabilities, how would you design a robust network?

Consider: What principles would you use? How would you balance cost, efficiency, and security?

💡 The Answer: Robust Network Design Principles

1. **Balanced Degree Distribution**: Avoid extreme homogeneity or hub concentration
2. **Multiple Redundant Pathways**: Ensure no single point of failure
3. **Strategic Hub Protection**: Invest heavily in protecting critical nodes
4. **Hierarchical Design**: Combine local clusters with hub connections
5. **Adaptive Responses**: Design systems that can reconfigure when attacks detected

The Robustness Paradox

Trade-off: Efficiency vs. Security

No Perfect Solution: No single structure optimal against all failure types

Design Challenge: Balance cost efficiency with resilience

Context-Dependent: Optimal design depends on threat model and cost constraints

Part VIII: Practical Applications

Coding Implementation

Tools:

- `igraph` - comprehensive network analysis
- `scipy.sparse.csgraph` - efficient connected component algorithms
- `networkx` - alternative approach with different robustness metrics

Key Functions:

- Connected components analysis
- MST algorithms
- Robustness simulation
- Percolation analysis

🤔 Real-World Application Question

❓ **Question**: How would you test our theoretical predictions about network robustness using real data?

Think about: What kind of network would be good to study? How would you measure robustness in practice?

💡 The Answer: Airport Network Analysis

Case Study: Airport connectivity network

Theoretical Prediction: Using Molloy-Reed criterion

- Calculate degree statistics: $\kappa = \langle k^2 \rangle / \langle k \rangle$
- Predict critical fraction: $f_c = 1 - 1/(\kappa - 1)$

Empirical Validation: Compare predicted vs. observed critical fractions

💬 **Discussion**: The high f_c value indicates the airport network is extremely robust to random failures. Why do you think this is? What does this tell us about how transportation networks are designed?

Part IX: Interactive Elements

Hands-on Exercises

Pen-and-Paper: MST to robust grid design

Interactive Demo: Network robustness simulation

Coding Exercises:

- Random vs. targeted attack comparison
- Percolation simulation
- Real-world network analysis

Assignment Overview

Repository Access:

- Enrolled students: Dedicated GitHub classroom link
- Others: Fork assignment repository

Grading: Automated testing framework

- `bash grading-toolkit/grade_notebook.sh tests/test_01.py assignment/assignment.ipynb`
- `bash grading-toolkit/grade_notebook.sh tests/test_02.py assignment/assignment.ipynb`

Learning Objectives: Apply concepts to real network data

Conclusion & Key Takeaways

Summary

- **MST**: Optimal cost efficiency but vulnerable to failures
- **Robustness**: Requires redundancy beyond minimum connectivity
- **Attack Asymmetry**: Random failures vs. targeted attacks create different vulnerabilities
- **Design Balance**: No single solution - context-dependent optimization

Next Steps

Assignment: Apply concepts to real network data

Further Reading:

- Borůvka (1926) - Original MST work
- Albert, Jeong & Barabási (2000) - Network robustness
- Cohen & Havlin (2010) - Comprehensive treatment

Applications: Infrastructure design, cybersecurity, system resilience

Questions?

Contact:
skojaku@binghamton.edu

Course Materials: Available on course website

Office Hours: Check course schedule

Interactive Resources:

- Network Robustness Demo
- Percolation Simulation
- Assignment Repository

References

1. **Borůvka, O.** (1926). O jistém problému minimálním. *Práce Moravské Přírodovědecké Společnosti*, 3, 37-58.
2. **Albert, R., Jeong, H., & Barabási, A. L.** (2000). Error and attack tolerance of complex networks. *Nature*, 406(6794), 378-382.
3. **Cohen, R., Erez, K., ben-Avraham, D., & Havlin, S.** (2000). Resilience of the Internet to random breakdowns. *Physical Review Letters*, 85(21), 4626-4629.
4. **Molloy, M., & Reed, B.** (1995). A critical point for random graphs with a given degree sequence. *Random Structures & Algorithms*, 6(2-3), 161-180.
5. **Cohen, R., & Havlin, S.** (2010). *Complex Networks: Structure, Robustness and Function*. Cambridge University Press.