Resilience in Complex Linked Systems

Fred Roberts, DIMACS

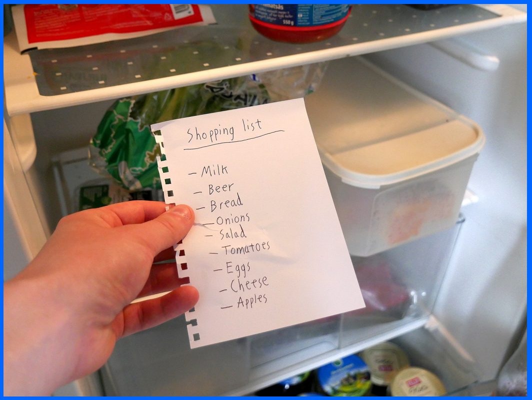
Image credits:
Hurricane damage: FEMA Photo by photographer Leif Skoogfors
Forest fire: USFS Region 5
Ebola treatment unit: CDC Global
No changes made in any image

DIMACS

Center for Discrete Mathematics & Theoretical Computer Science
Founded as a National Science Foundation Science and Technology Center

Resilience

- Today’s society has become dependent on complex systems, enabled by increased digitization of our world, that have had a great impact on virtually all facets of our lives:
  - Instant communication
  - Ability to move money anywhere and quickly
  - Ability to ask a machine to make our shopping list or turn on our favorite music.



- Yet these changes have made us vulnerable.
- To natural disasters, deliberate attacks, just plain errors.
- In recent years, ***“resilience”*** of complex natural and social systems has become a major area of emphasis.

Credit: Santeri Viinamäki via Wikimedia commons   no changes made

Resilience

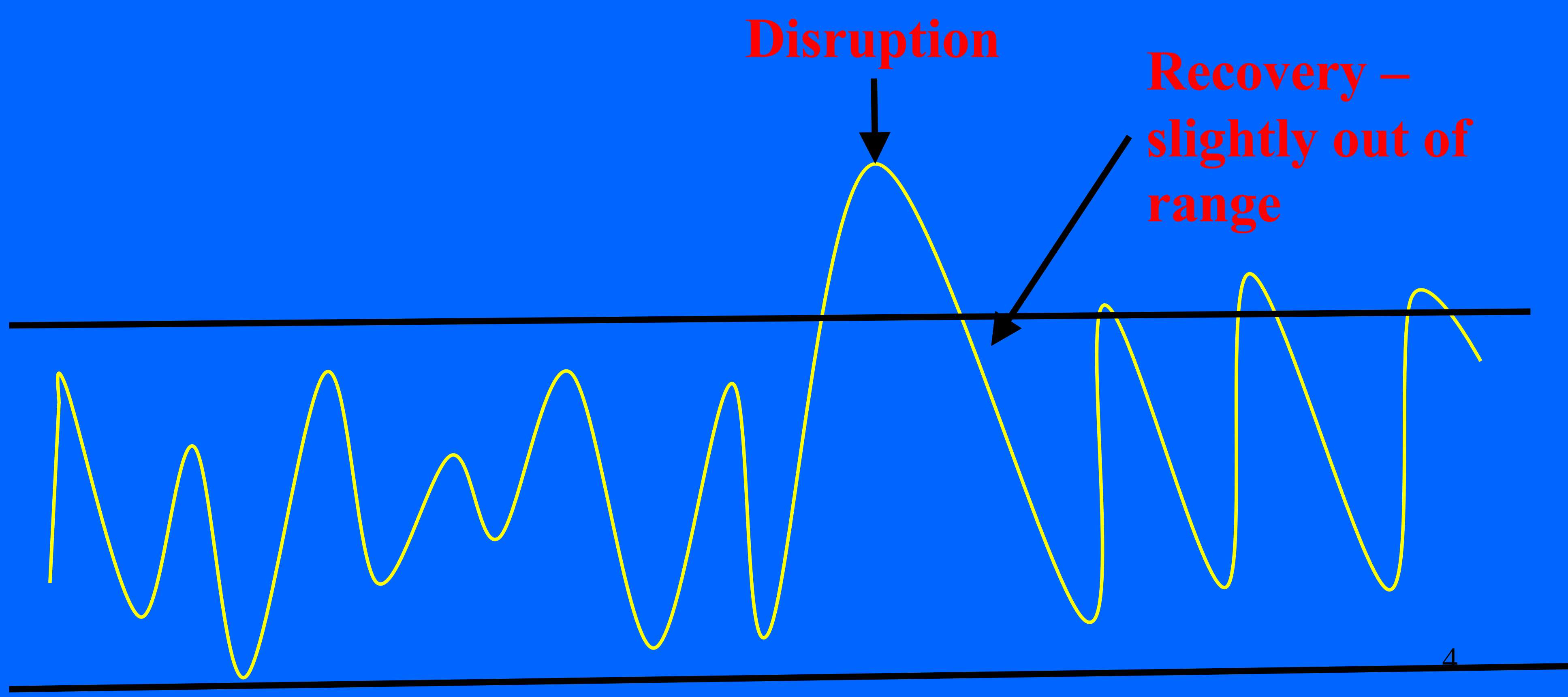
Hospital in Kansas during 1918 influenza pandemic

- Resilience in response to hurricanes, disease events, floods, earthquakes, cyber attacks, …

Image credits:
Earthquake: U.S. Air Force photo by Master Sgt. Jeremy Lock via Wikimedia commons
Flood: Voice of America Indonesian Service via Wikimedia commons
1918 influenza outbreak: Otis Historical Archives, National Museum of Health and Medicine via Wikimedia.com No changes made in any image

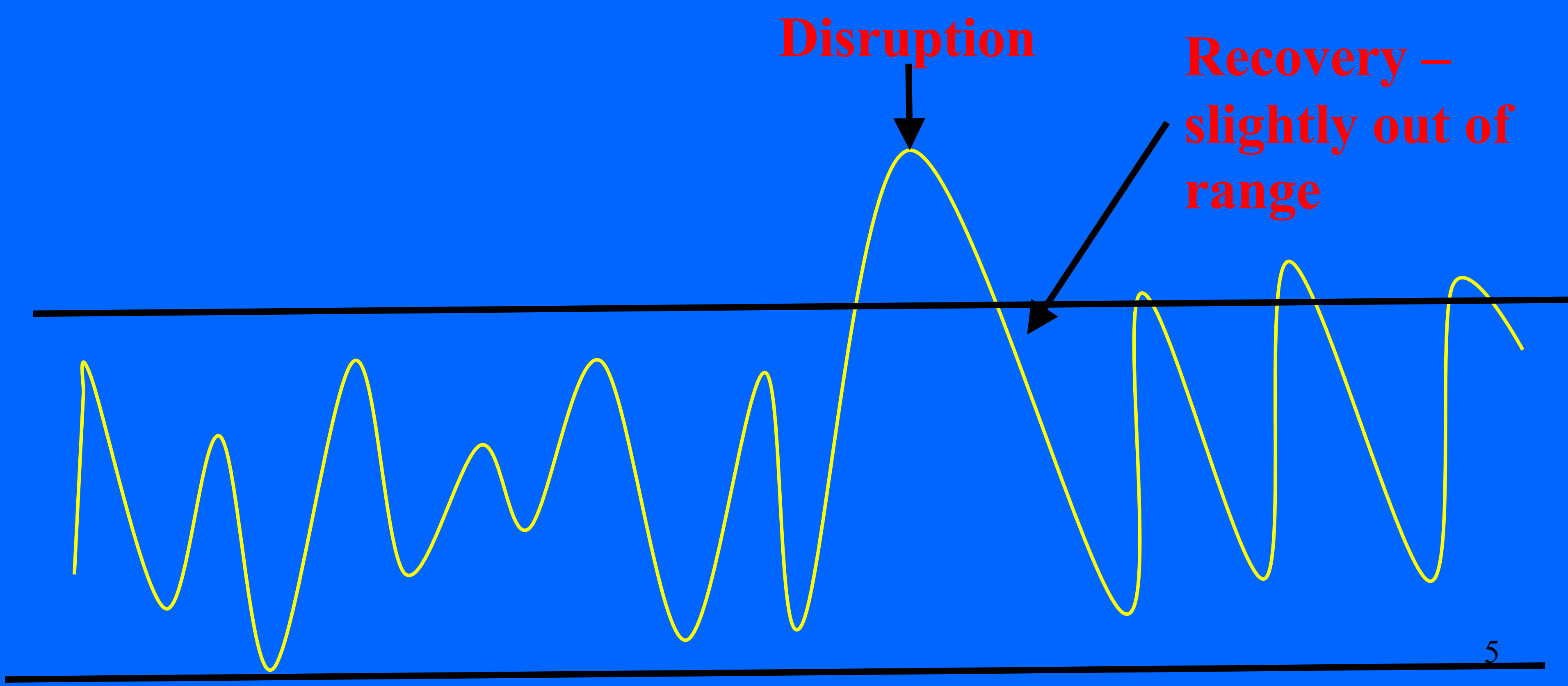
Resilience

•General concept of resilience: ability of a system to recover from disasters or attacks and avoid catastrophic collapse.



Resilience

- In a resilient system, values will return to the normal healthy range.
- Or they might establish a new healthy range – one that is not that far from the previous one



Resilience

- There are many parameters that measure a "healthy" system.
- Some will get back into their normal healthy range faster than others.
- Do we ask that the longest time to return to this range be small?
- Or that the average time to return to this range be small?

Approaches to Achieving Resilience

- One approach to resilience is to develop algorithms for responding to a disruption that will minimize the departure from the previous state when things settle down.
- Another is to design systems that can bounce back from disruptions quickly.
- I will emphasize the former.
- Will illustrate with four examples built around models using graphs and networks.

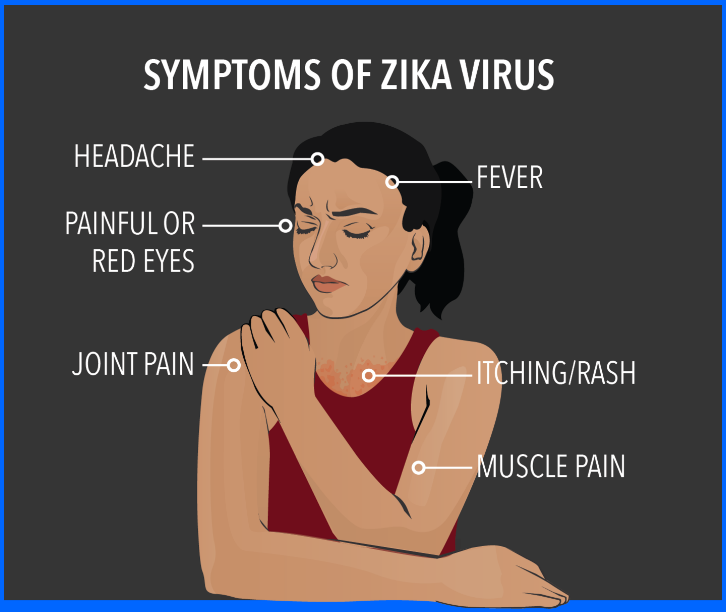
Example I: Spread and Control of Disease

•The spread of the new Coronavirus COVID-19 is just the latest worrisome example of a newly emerging disease that threatens not only lives but our economy and our social systems.

Image credit: Wikimedia commons
https://www.youtube.com/watch?v=SBboFVjLQak , 1:10
Chinanews.com/China News Service
Unchanged

Example I: Spread and Control of Disease

Ebola



Zika

- Ebola, Zika are other recent examples

Image credits: Wikimedia commons
Ebola: Army Medicine; Zika: Beth.herlin no changes made

Example I: Spread and Control of Disease

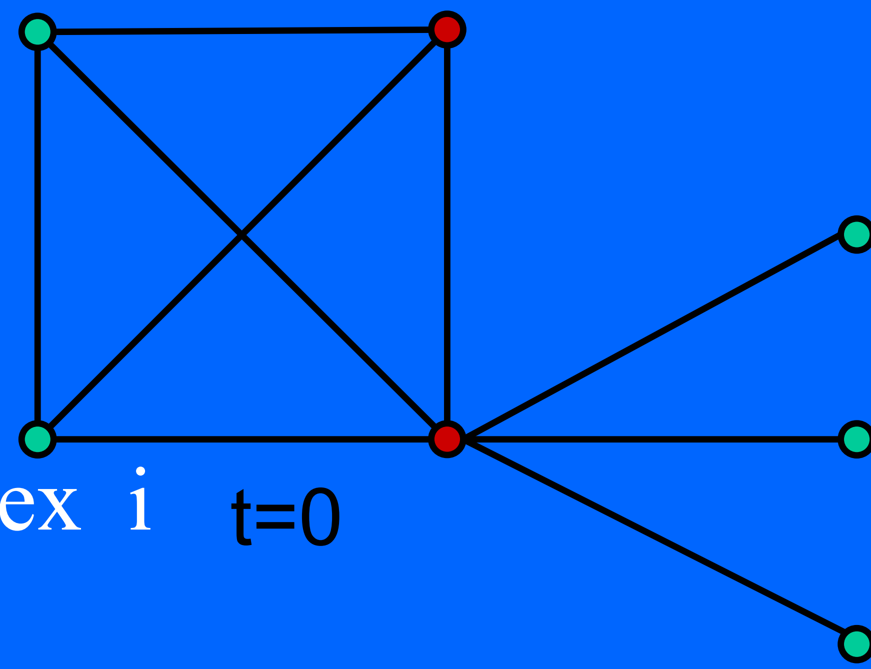
- Modern transportation systems allow for rapid spread of diseases.
- Diseases are spread through social networks.
- "*Contact tracing*" is an important part of any strategy to combat outbreaks of infectious diseases, whether naturally occurring or resulting from bioterrorist attacks.
- I will illustrate the ideas with some fairly simple ”toy” models that will illustrate concepts of resilience.

Simple Model: Moving From State to State

Social Network = Graph

Vertices = People

Edges = contact



Let $s_i(t)$ give the state of vertex i at time t.

Very simplified "toy" model: Two states:

= susceptible, = infected (SI Model)

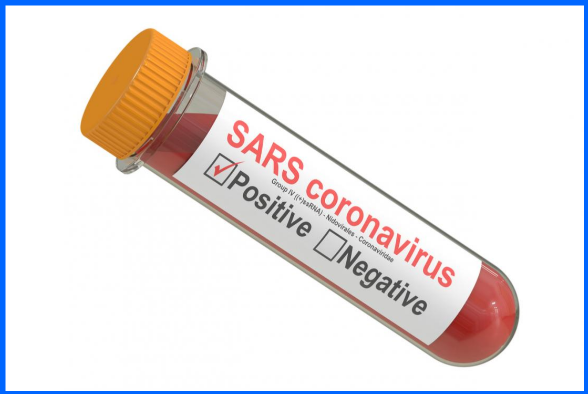
Times are discrete: t = 0, 1, 2, …

The Model: Moving From State to State

measles

More complex models: SI, SEI, SEIR, etc.

S = susceptible, E = exposed, I = infected, R = recovered (or removed)



SARS

Credit: measles: Wikimedia.org
SARS: Medical News Today

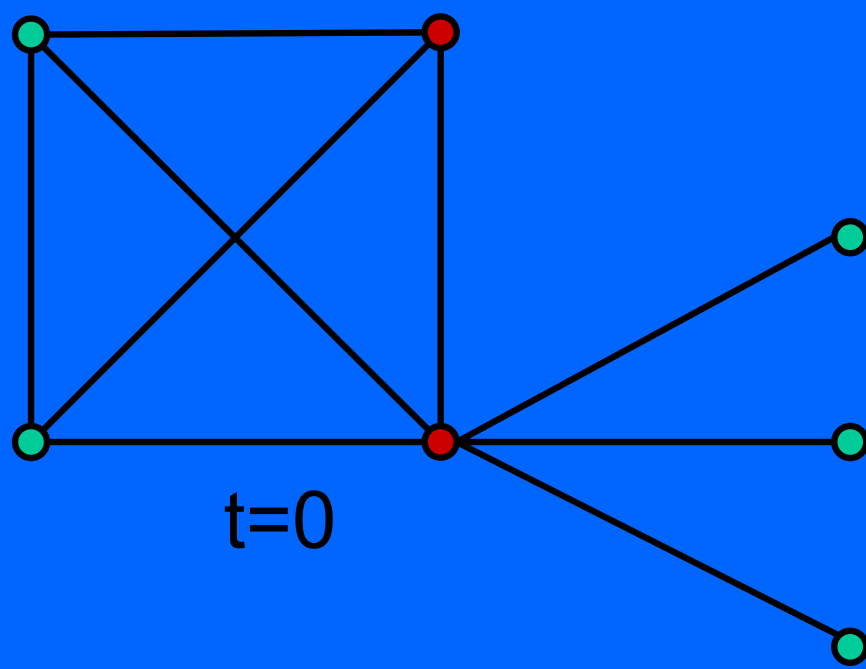
Threshold Processes

Irreversible k-Threshold Process: You change your state from to at time t+1 if at least k of your neighbors have state at time t. You never leave state .

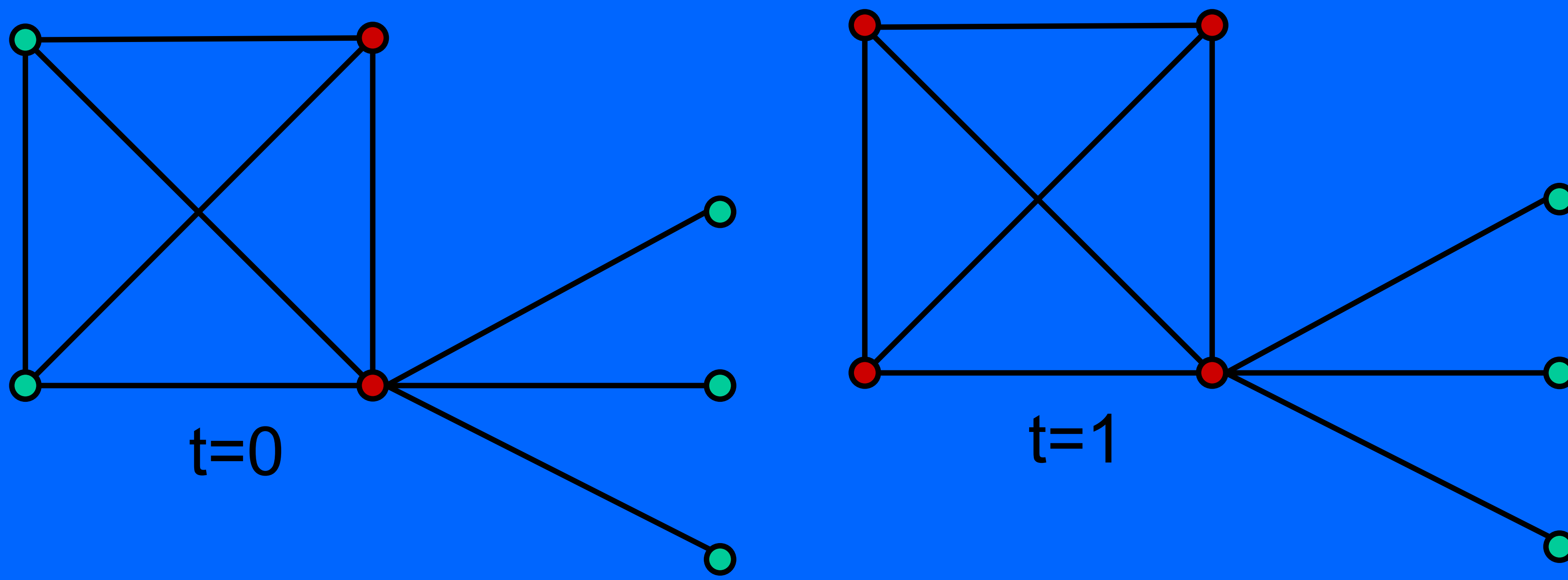
Disease interpretation? Infected if sufficiently many of your neighbors are infected.

Special Case k = 1: Infected if any of your neighbors is infected.

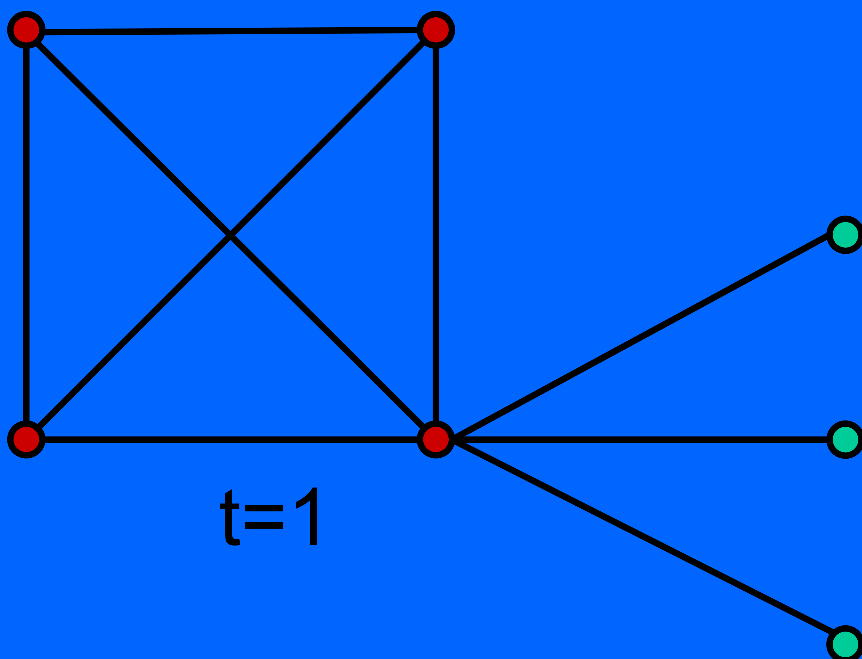
Irreversible 2-Threshold Process

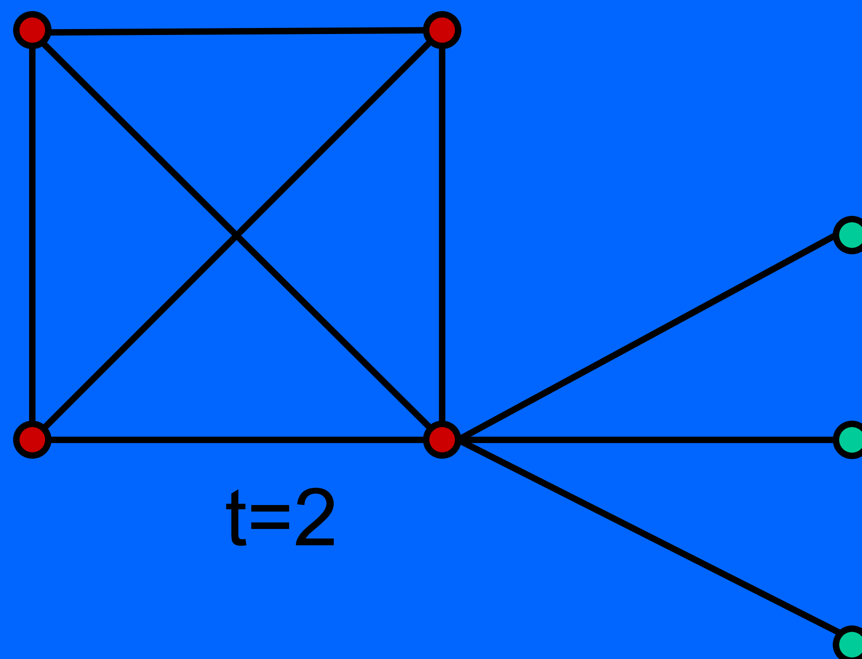


Irreversible 2-Threshold Process

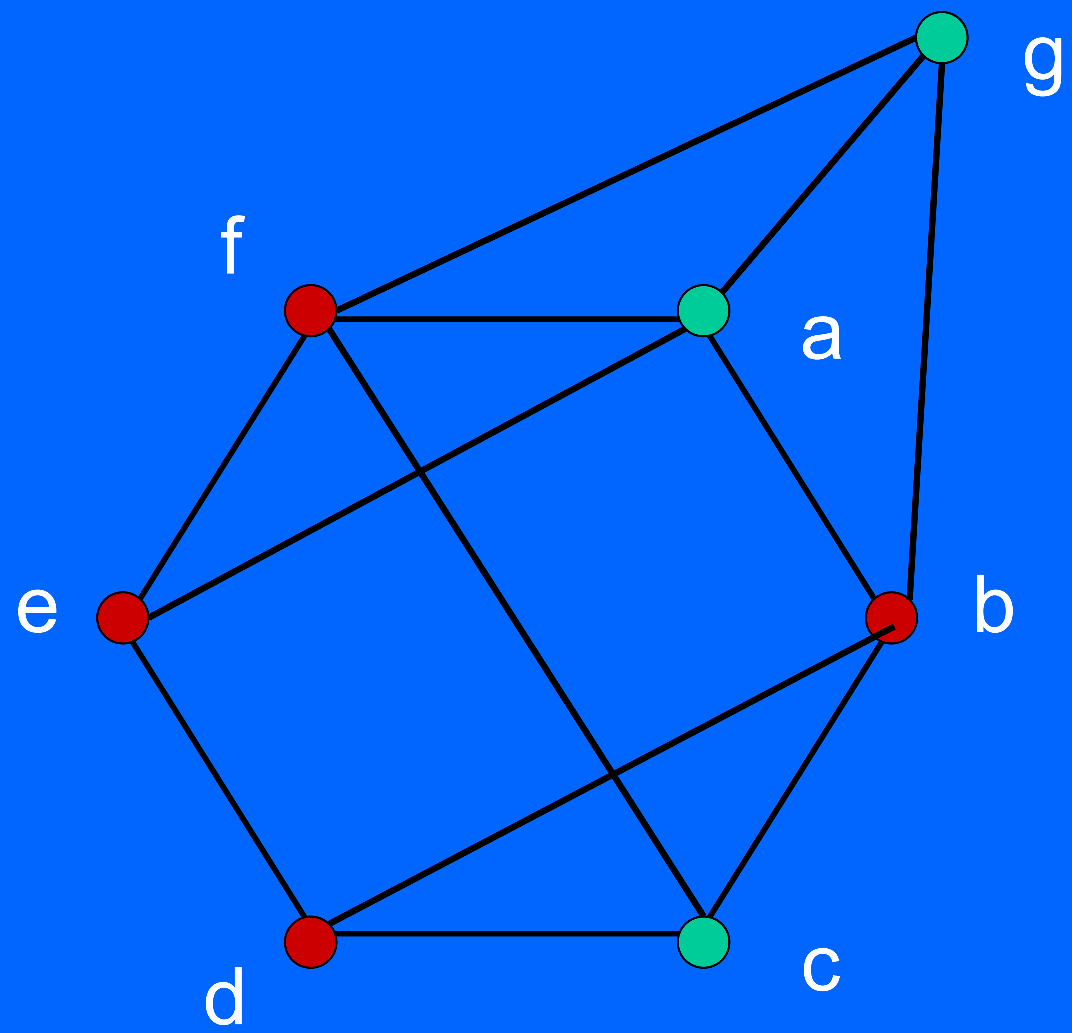


Irreversible 2-Threshold Process



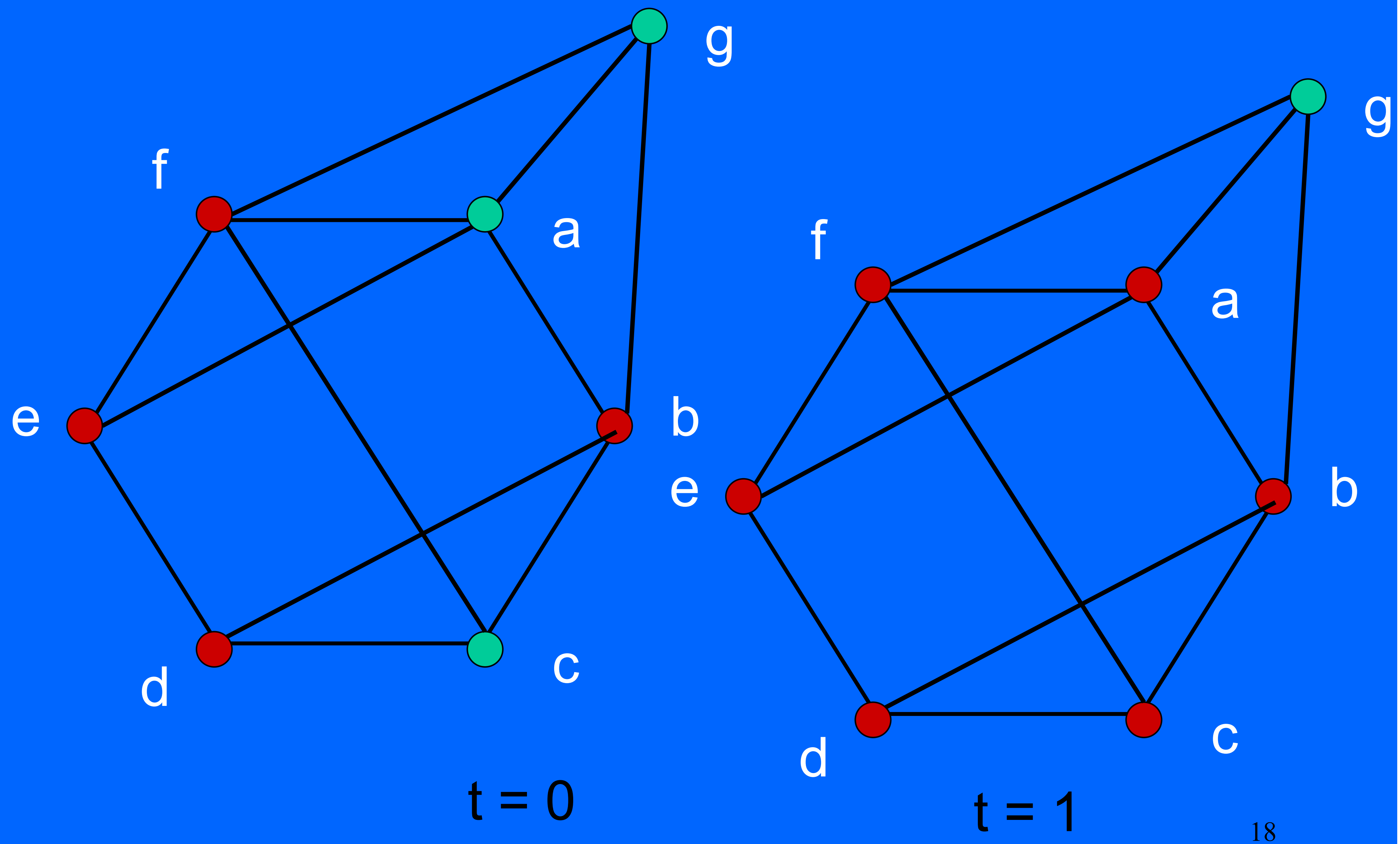


Irreversible 3-Threshold Process

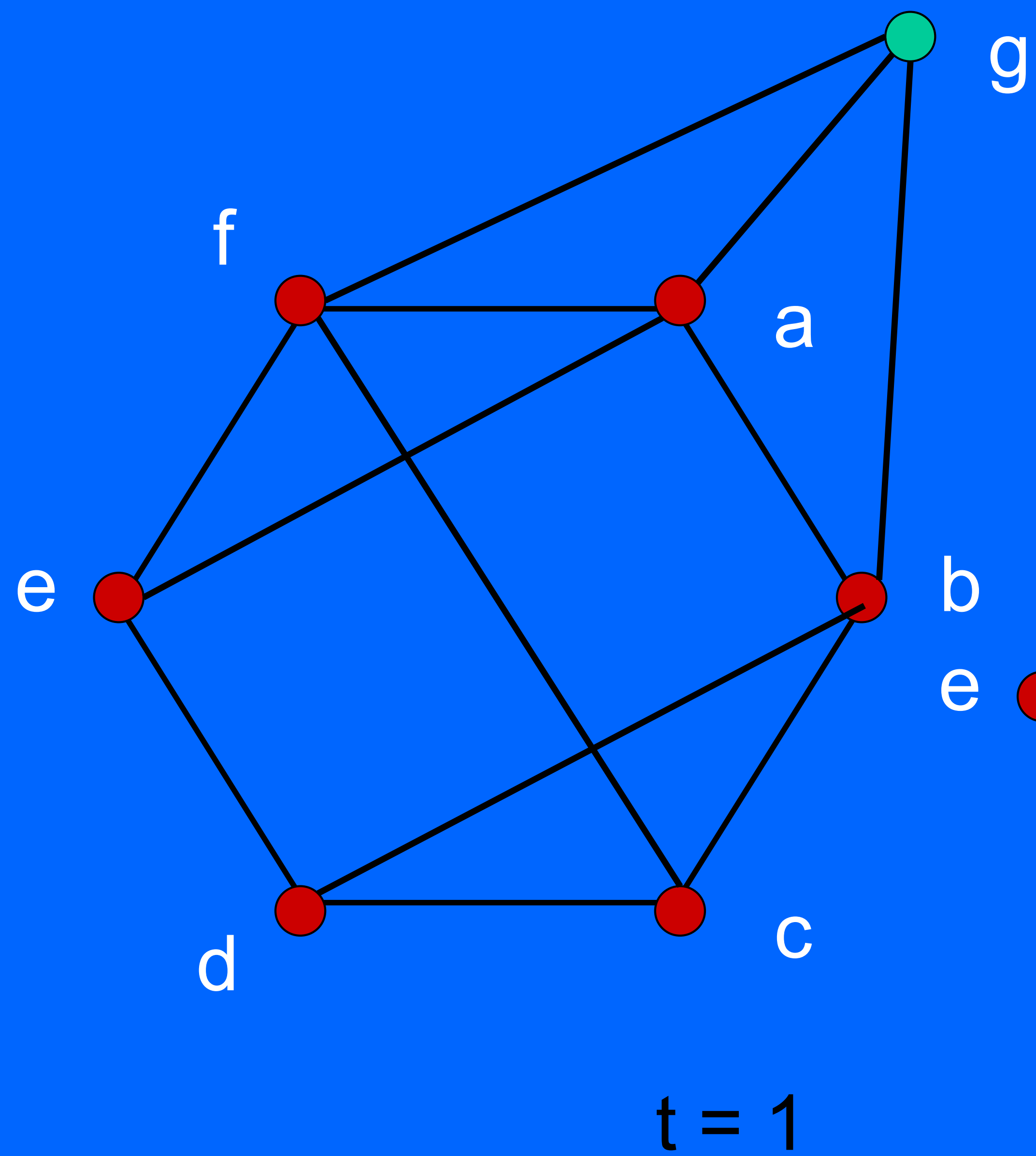


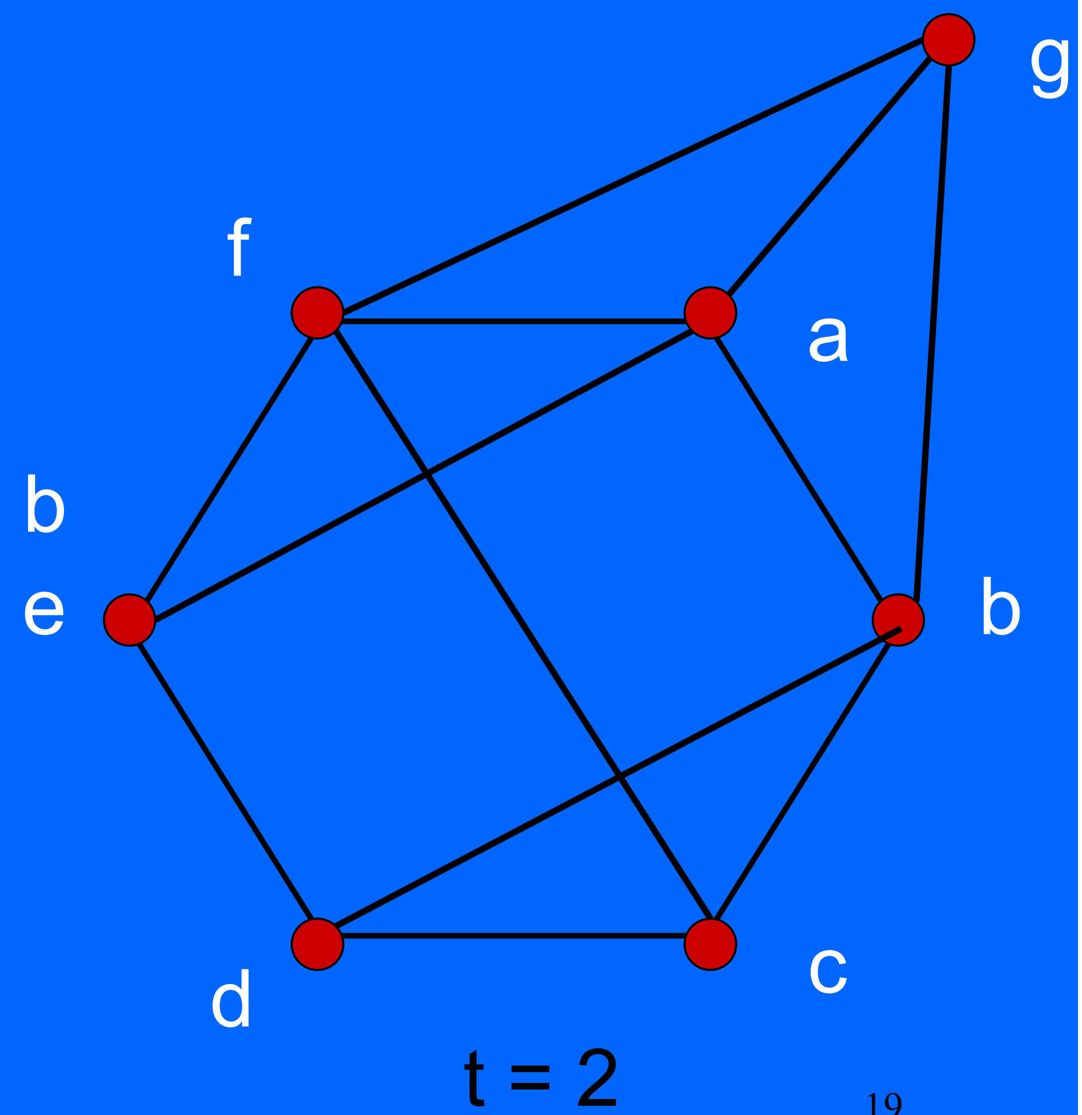
t = 0

Irreversible 3-Threshold Process



Irreversible 3-Threshold Process





The Saturation Problem

A great deal of attention has been paid to:
Attacker's Problem: Given a graph, what subsets S of the vertices should we plant a disease with so that ultimately the maximum number of people will get it?

Economic interpretation: What set of people do we place a new product with to guarantee "saturation" of the product in the population?

These Problems are "Hard"

Problem **IRREVERSIBLE k-CONVERSION SET**: Given a positive integer p and a graph G, does G have a set S of size at most p so that if all vertices of S are infected at the beginning, then all vertices will ultimately be infected?

Theorem (Dreyer and Roberts): IRREVERSIBLE k-CONVERSION SET is NP-complete for fixed k > 2.

Complications to Add to Model

- k = 1, but you only get infected with a certain probability.
- You are automatically cured after you are in the infected state for d time periods.
- ***A public health authority has the ability to "vaccinate" a certain number of vertices, making them immune from infection.***
- It's the vaccination strategy that relates to the resilience question.

Smallpox

Image credit: https://wellcomeimages.org/indexplus/obf_images/b2/a8/_9ca500938fc44f77d4c4e49a4d90.jpg

Vaccination Strategies

Credit: wikimedia commons.org

Mathematical models are very helpful in comparing alternative vaccination strategies. The problem is especially interesting if we think of protecting against deliberate infection by a bioterrorist arttacker but applies if we think of ”nature” as the attacker.

Example II: Vaccinations and Fighting Fires

Stephen Hartke and others worked on a vaccination problem:

Defender: can vaccinate v people ***per time period***.

Attacker: can only infect people at the beginning.
Irreversible k-threshold model.

What vaccination strategy minimizes number of people infected?

Variation: The vaccinator and infector alternate turns, having v vaccinations per period and i doses of pathogen per period. What is a good strategy for the vaccinator?

Example II: Vaccinations an Fighting Fires

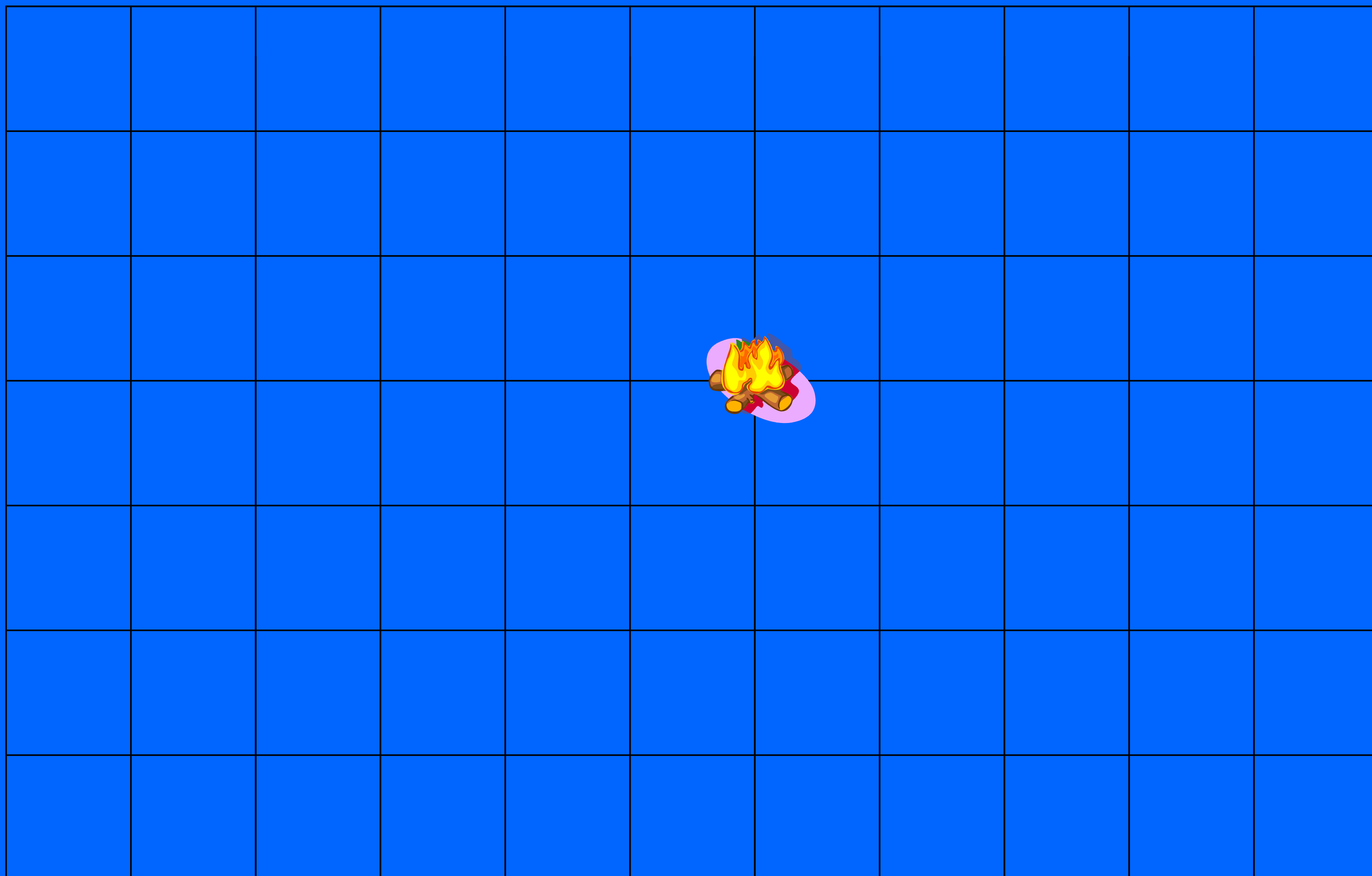
Sometimes called the ***firefighter problem***: alternate fire spread and firefighter placement.

Usual assumption: k = 1. (We will assume this.)

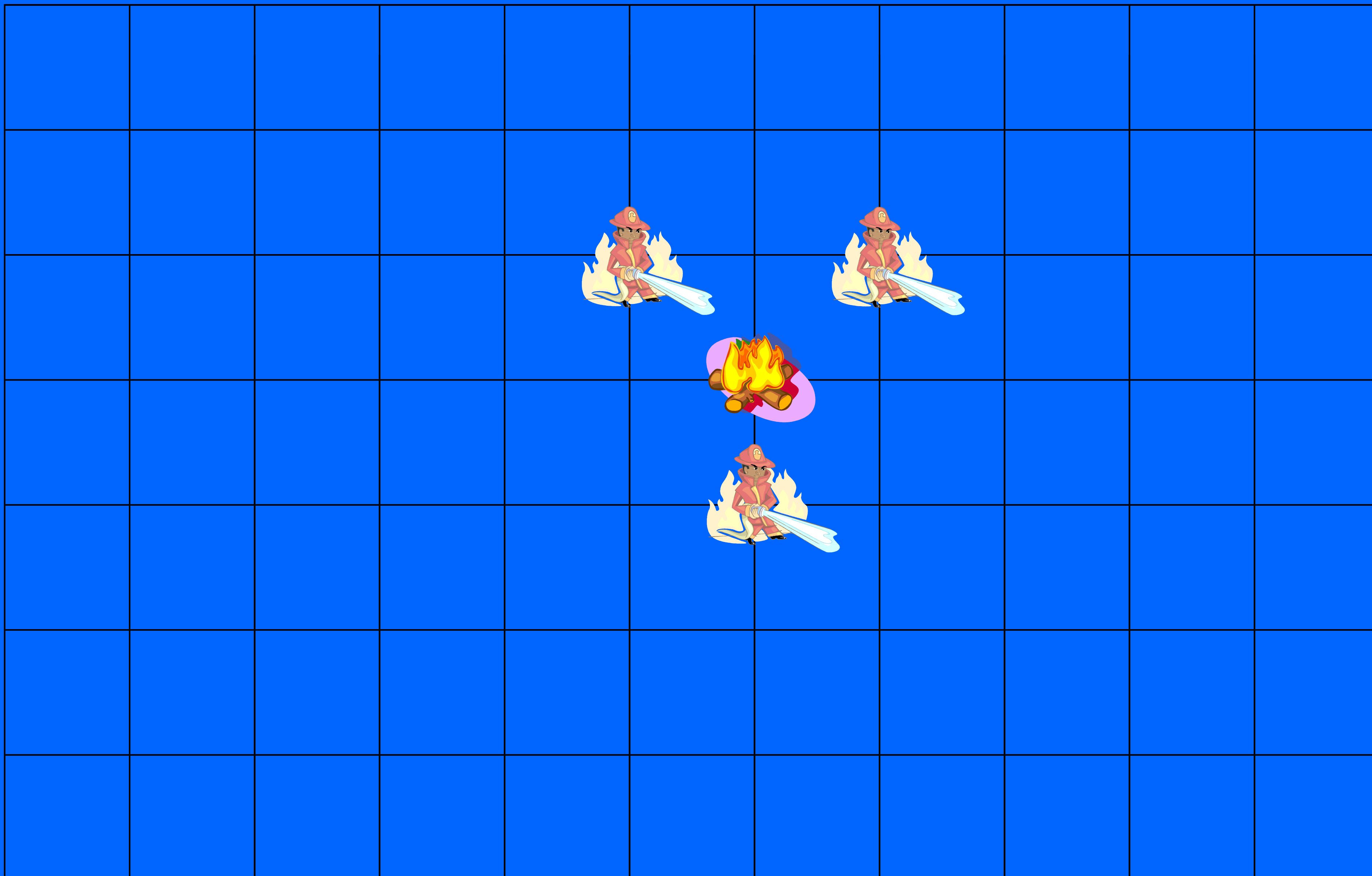
Problem goes back to Bert Hartnell 1995

Image credit:
Flickr/Bundesheer Fotos

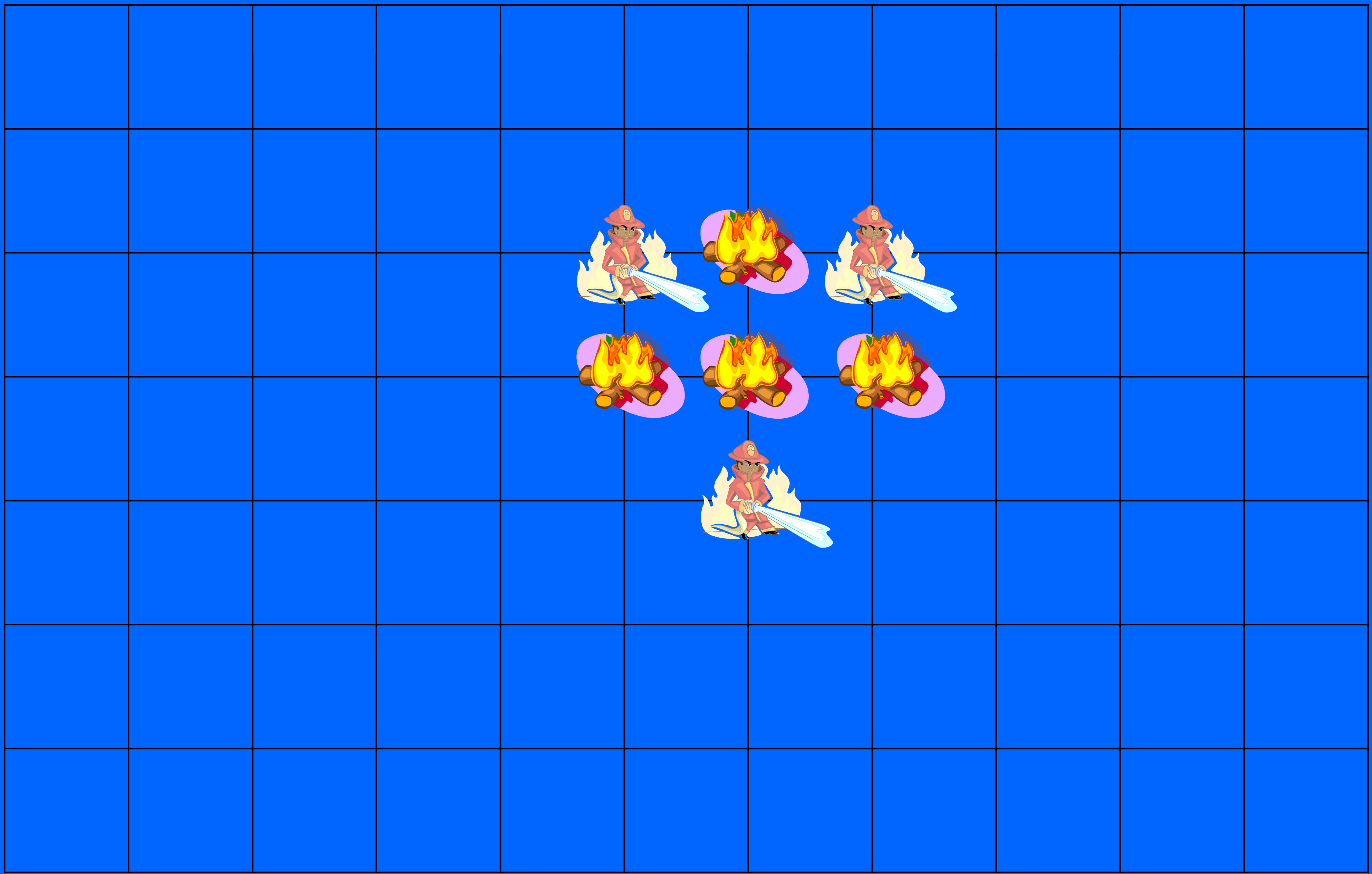
A Simple Model (k = 1) (v = 3)



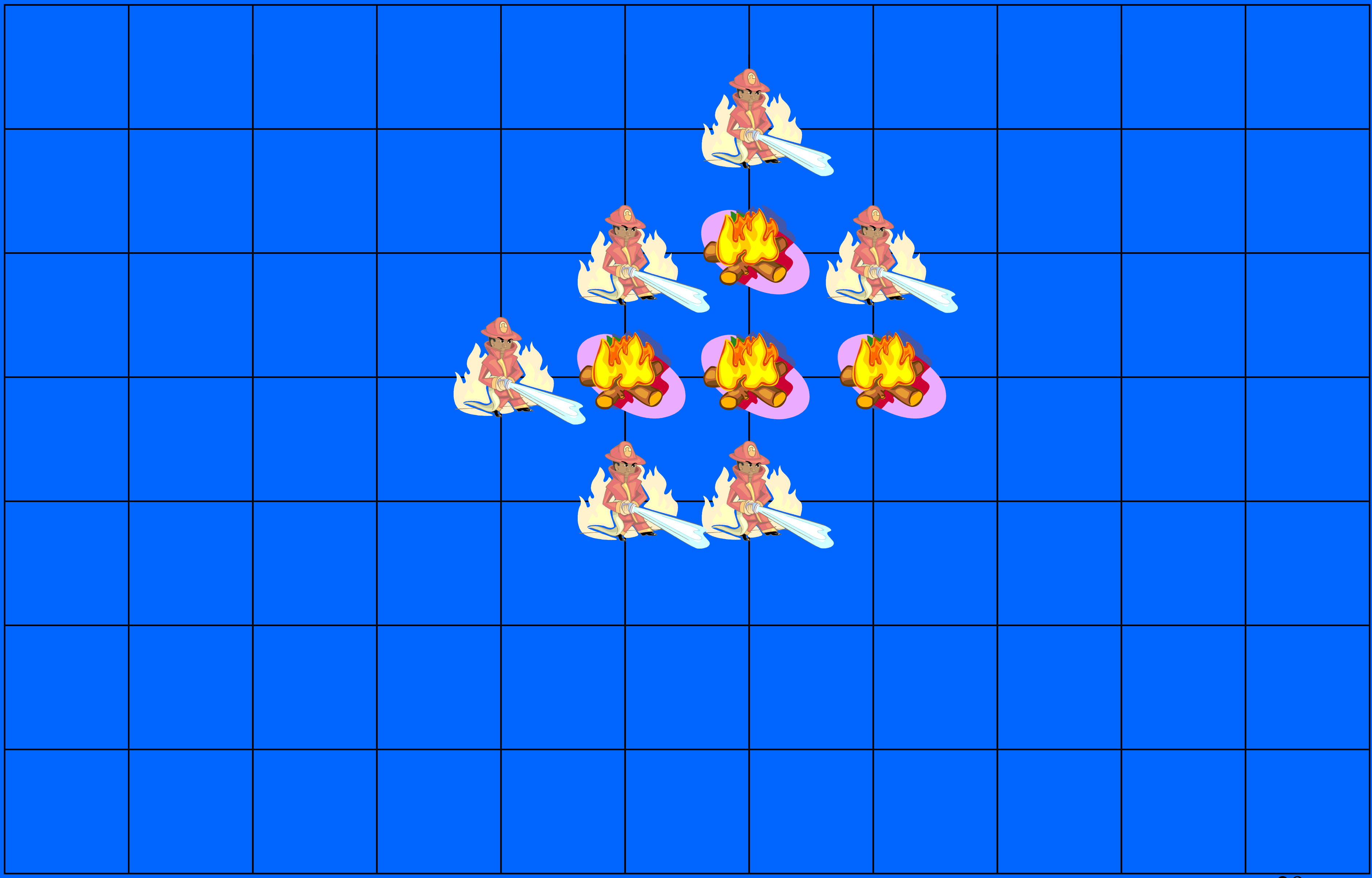
A Simple Model



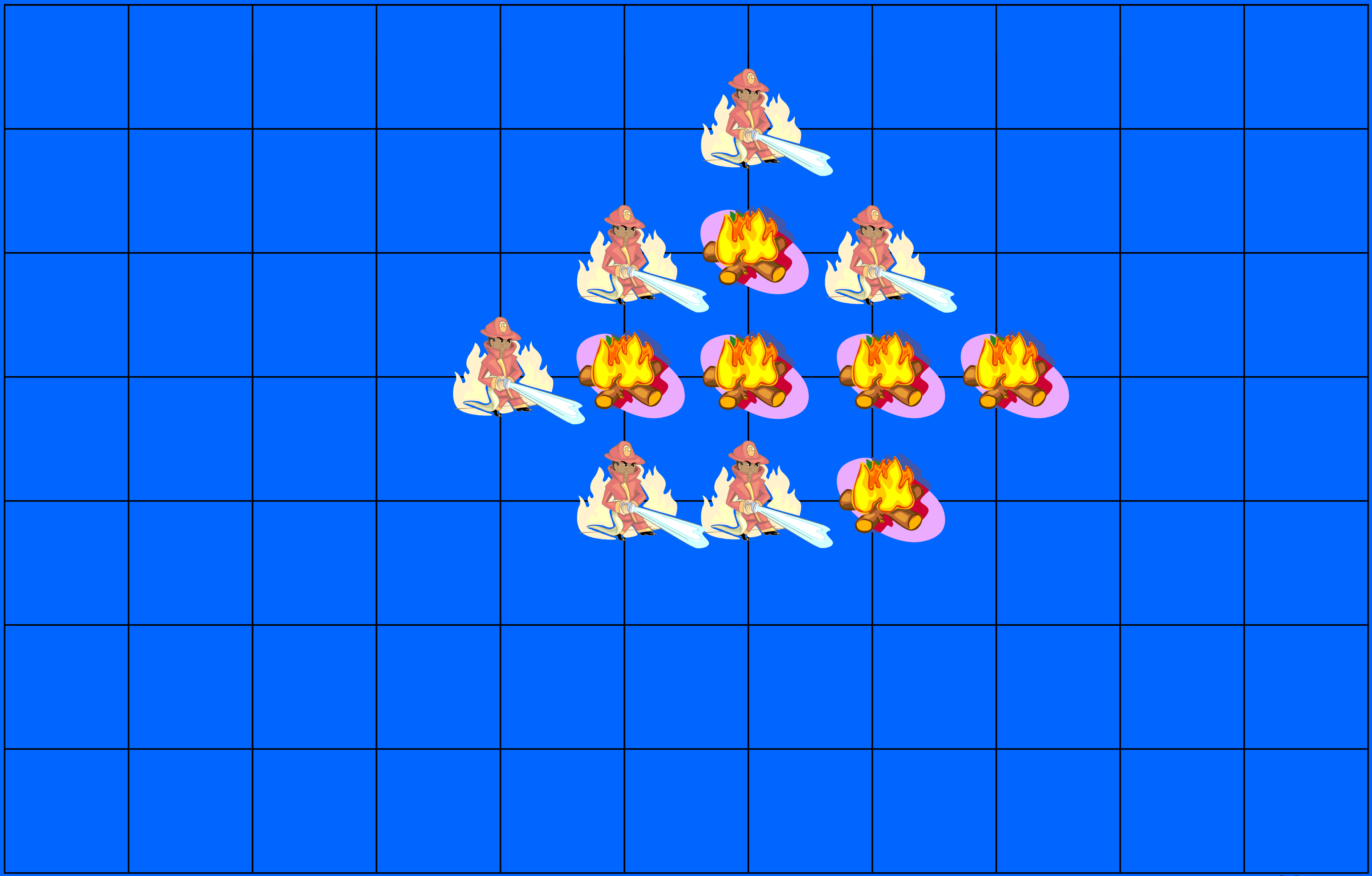
A Simple Model



A Simple Model



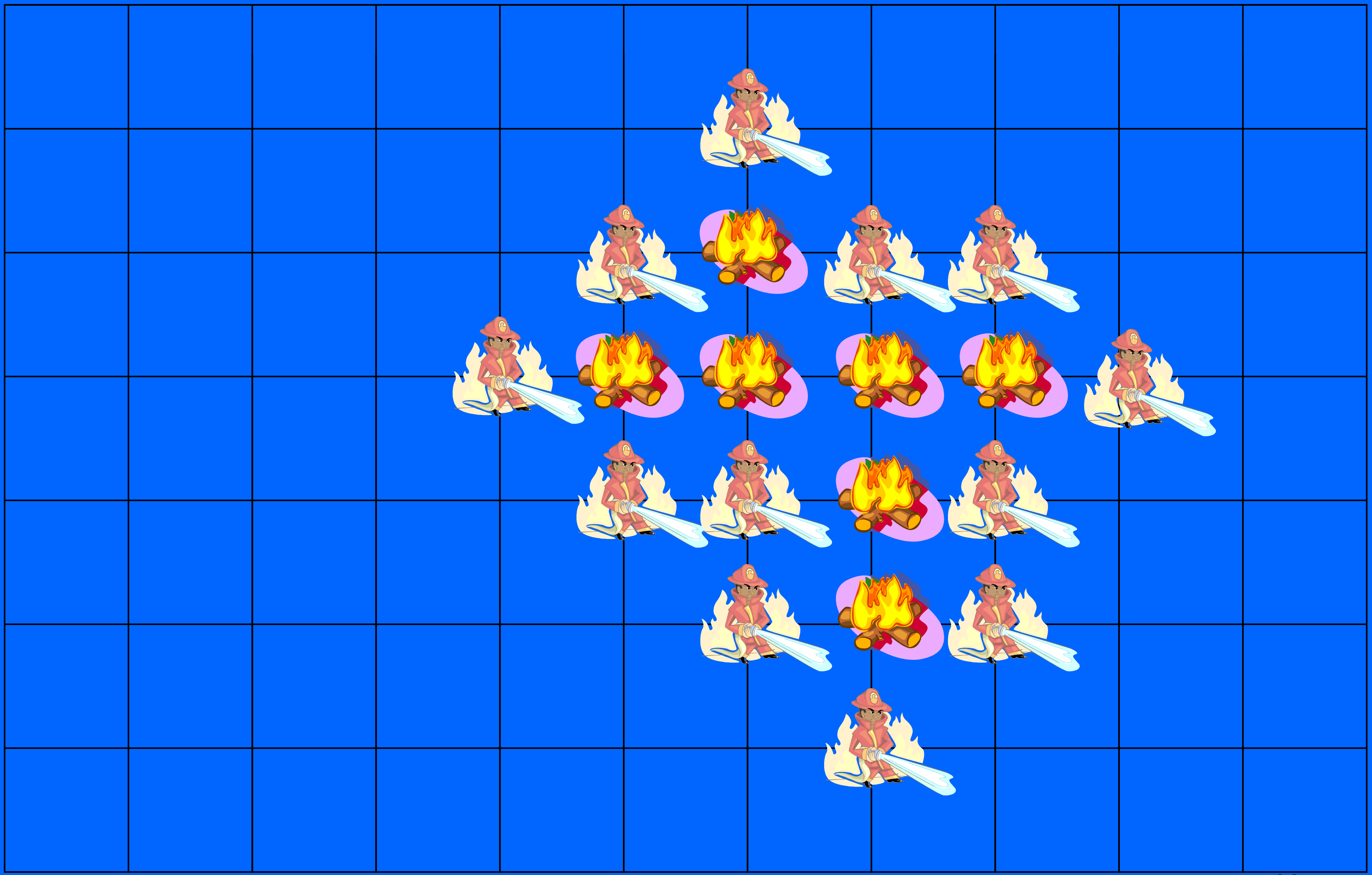
A Simple Model



A Simple Model

A Simple Model

A Simple Model



Some resilience questions that can be asked

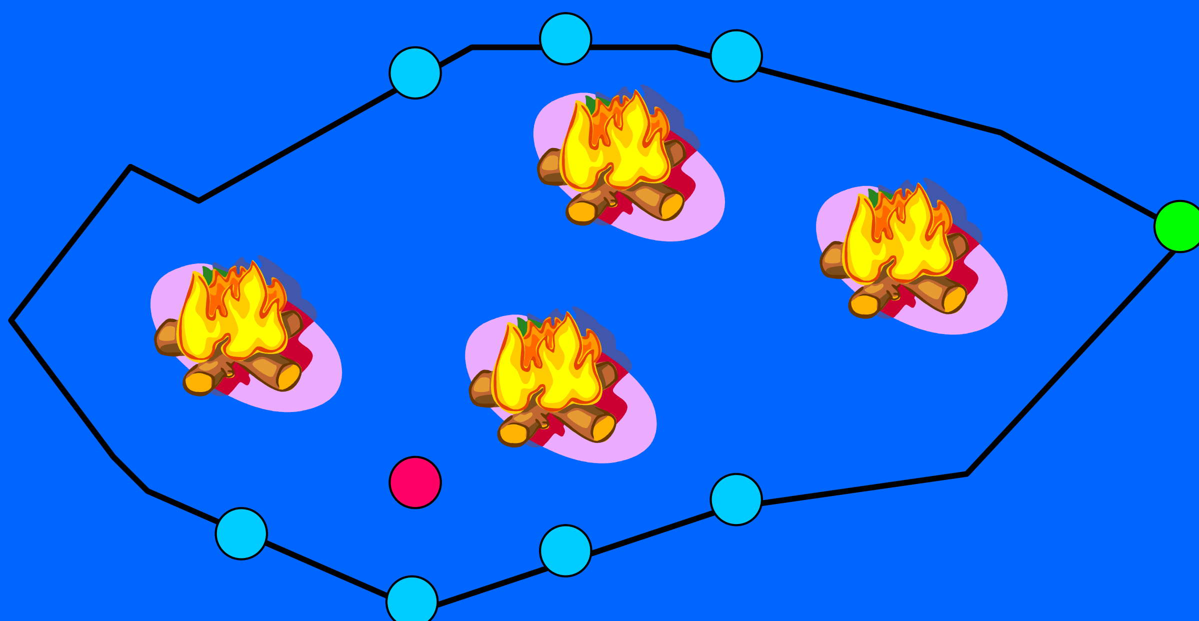
- Can the fire (epidemic) be contained?
- How many time steps are required before fire is contained?
- How many firefighters per time step are necessary?
- What fraction of all vertices will be saved (burnt)?
- Does where the fire breaks out matter?
- Fire starting at more than 1 vertex?

Containing Fires in Infinite Grids L_d

Fire starts at only one vertex:

d =1: Trivial.

d = 2: Impossible to contain the fire with 1 firefighter per time step

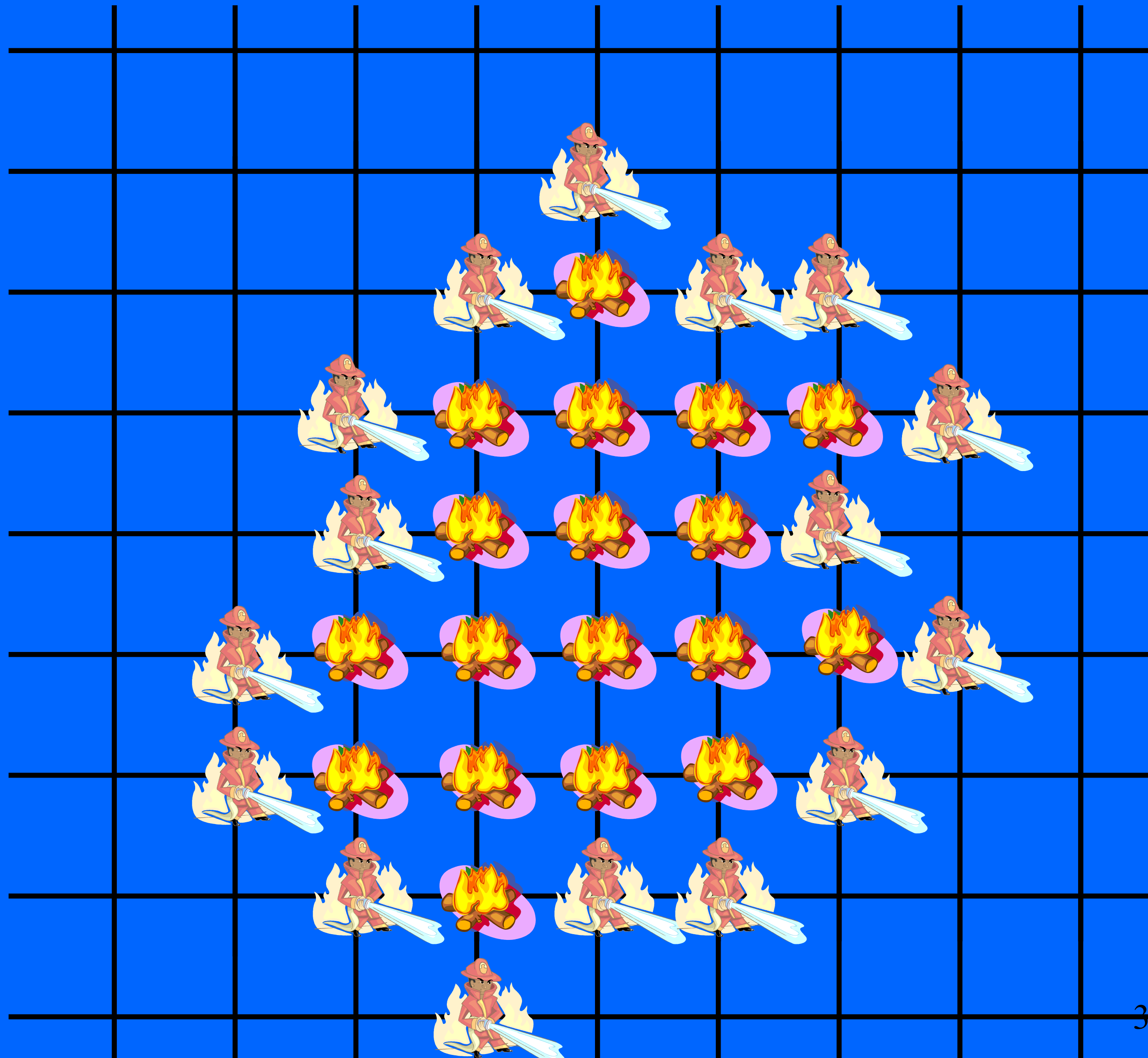


Containing Fires in Infinite Grids L_d

$d = 2$: Two firefighters per time step needed to contain the fire.

8 time steps

18 burnt vertices



Containing Fires in Infinite Grids L_d

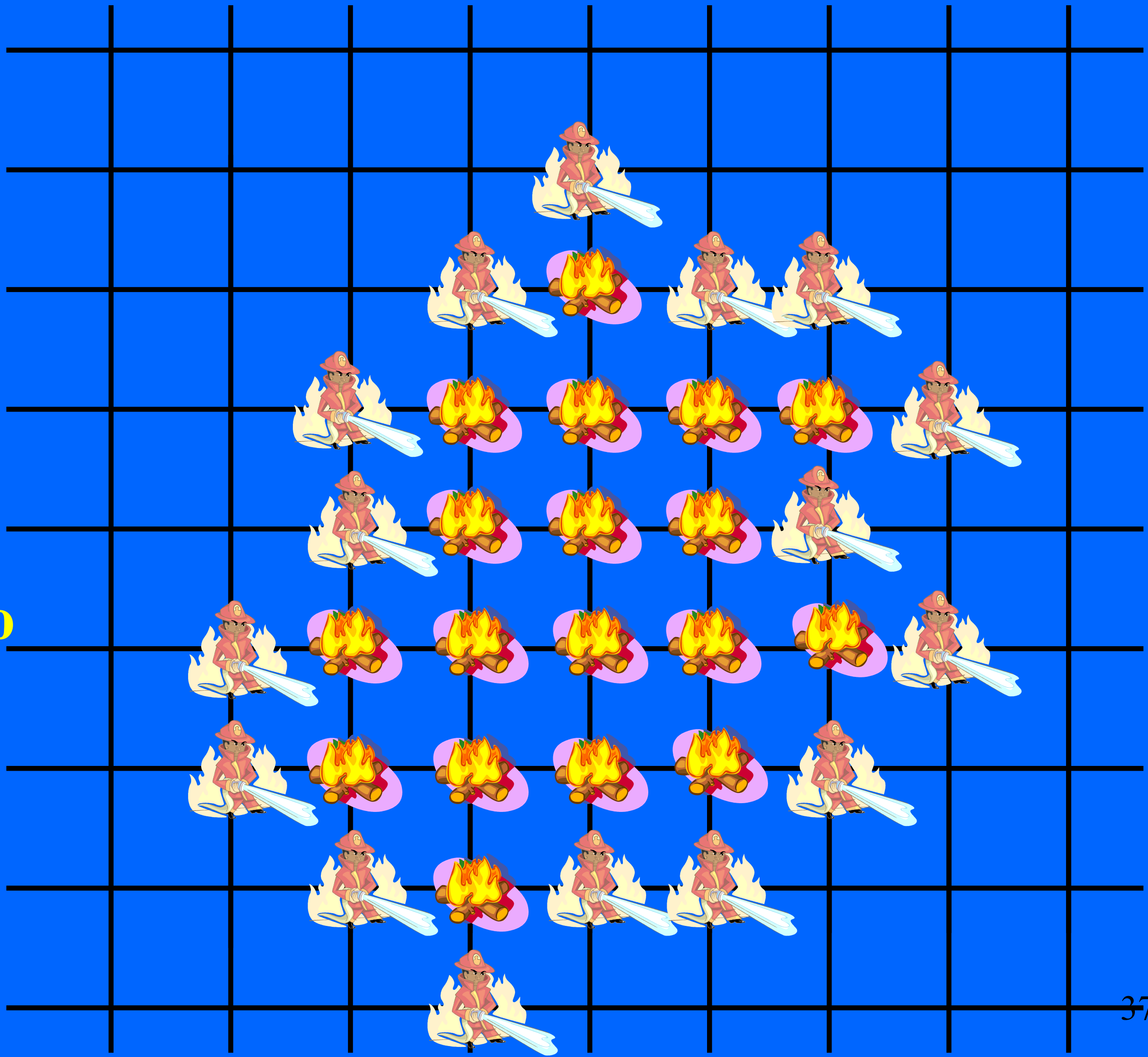
$d = 2$: Two firefighters per time step needed to contain the fire.

8 time steps

18 burnt vertices

Develin & Hartke: cannot do better than 18

Wang & Moeller: Cannot contain fire in < 8 steps



Containing Fires in Infinite Grids L_d

Sample Result:

$d \geq 3$: In L_d, every vertex has 2d neighbors.

Thus: $2d$-1 firefighters per time step are sufficient to contain any outbreak starting at a single vertex.

Theorem (Develin and Hartke): If $d \geq 3$, $2d - 2$ firefighters per time step are not enough to contain an outbreak in $L_{d.}$

Thus, $2d - 1$ firefighters per time step is the minimum number required to contain an outbreak in L_d and containment can be attained in 2 time steps.

More Realistic Models

- You stay in the infected state (state ●) for d time periods after entering it and then go back to the uninfected state (state ●).
- We vaccinate a person in state ● once k-1 neighbors are infected (in state ●).
- What if you only get infected with a certain probability if you meet an infected person?
- What if vaccines only work with a certain probability?
- What if the amount of time you remain infective exhibits a probability distribution?

Example III: Cascading Outages in the Power Grid

•Today's electric power systems operate under considerable uncertainty.

•Cascading failures can have dramatic consequences.

Blackout

Image credit:
Wikimedia commons; David Shankbone   no changes

Cascading Outages in the Power Grid

Grid Resilience:

•How can we design "control" procedures so that the power grid can quickly and efficiently respond to disturbances and quickly be restored to its healthy state?

•Grid disruptions can cascade so fast that a human being may not be able to react fast enough to prevent the cascading disaster leading to a major blackout.

•We are dependent on rapid response through algorithms.

Cascading Outages in the Power Grid

Grid Resilience:

- We are dependent on rapid response through algorithms.
- Need fast, reliable algorithm to respond to a detected problem.
  - Should not necessarily require human input
  - Has to be able to handle multiple possible "solutions"
  - Has to be able to understand what to do if all possible solutions are "bad"

Cascading Outages in Power Grid

Grid Resilience:

- Tool of interest: cascade model of Dobson, et al.
  - An initial "event" takes place
  - Reconfigure demands and generator output levels
  - New power flows are instantiated
  - The next set of faults takes place according to some stochastic model

Cascading Outages in Power Grid

Grid Resilience:

- The power grid model is not the same as a disease-spread model.
- Energy flows from generators through power lines (edges in the power grid graph).
- Each edge has a maximum capacity.
- When a vertex (substation) or edge (transmission line) outage occurs, power reroutes according to physical laws (Kirchhoff’s Law, Ohm’s Law).

Cascading Outages in Power Grid

Grid Resilience:

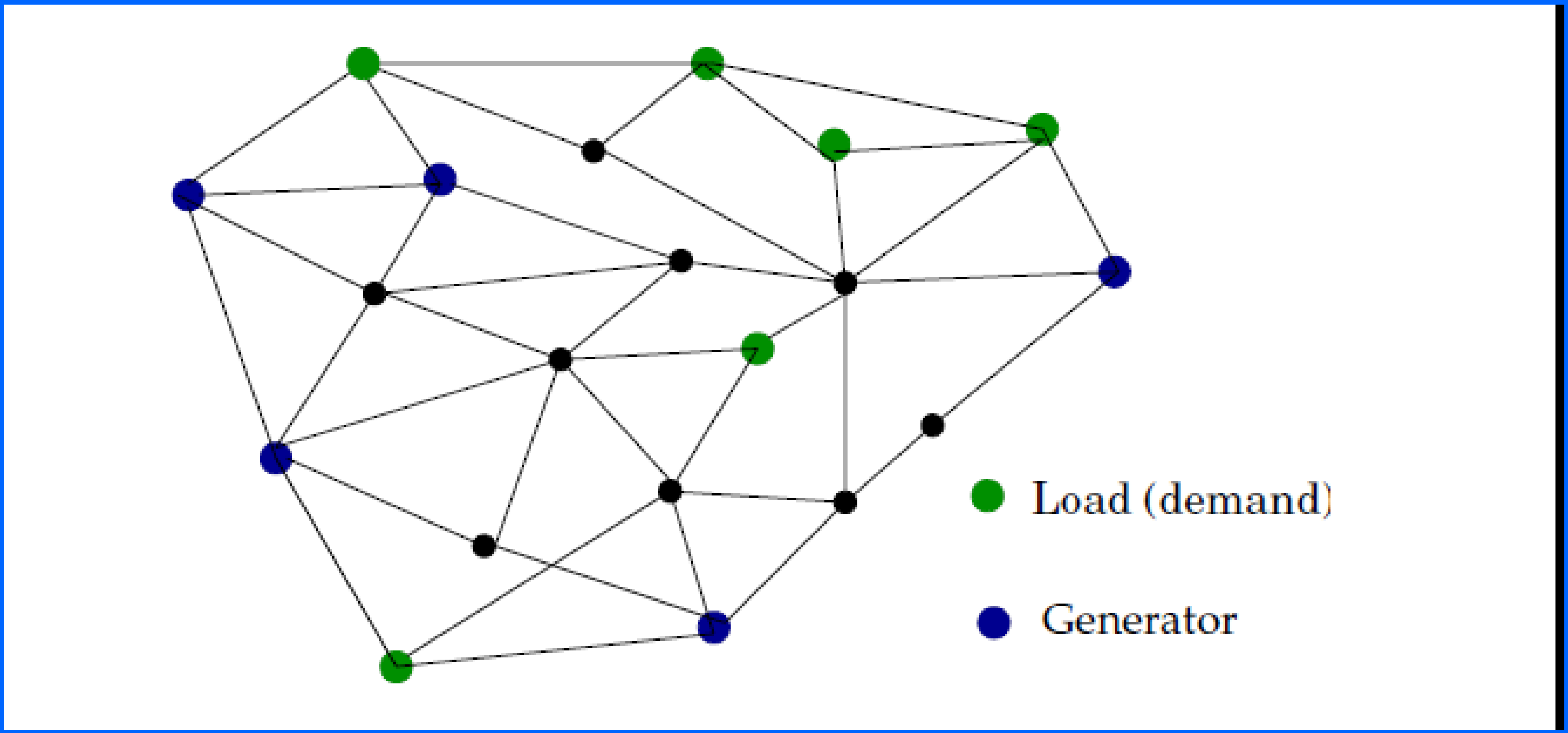
- Because of the rerouting, flows on parallel paths are increased.
- This could cause an overload in a distant transmission line.
- So failures can take place non-locally.

Korkali, et al. (2017)

Cascading Outages in the Power Grid

Grid Resilience:

Cascade Model (Dobson, et al.)

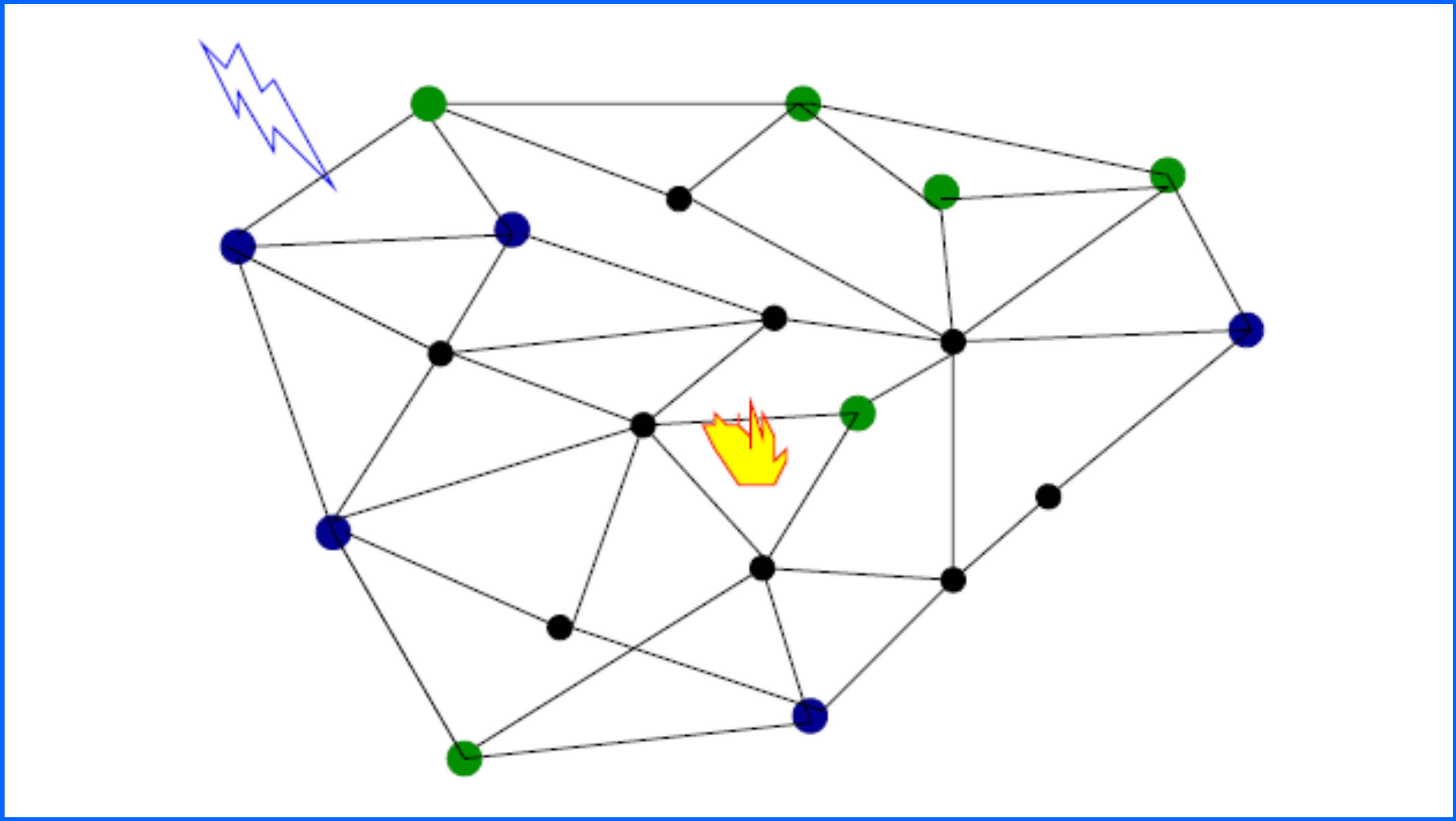


Credit: Daniel Bienstock

Cascading Outages in the Power Grid

Grid Resilience:

Cascade Model (Dobson, et al.)

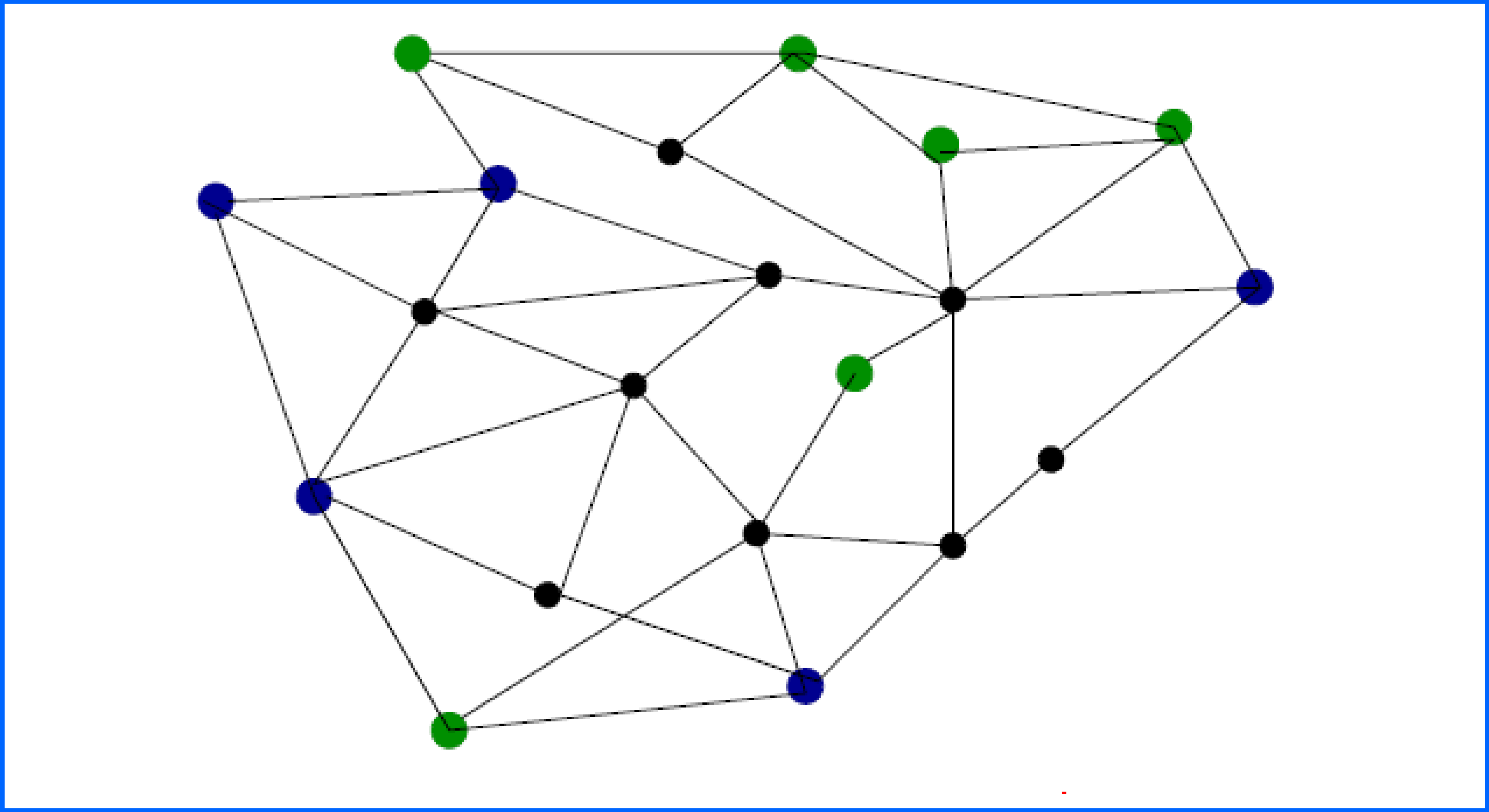


Credit: Daniel Bienstock

Cascading Outages in the Power Grid

Grid Resilience:

Cascade Model (Dobson, et al.)

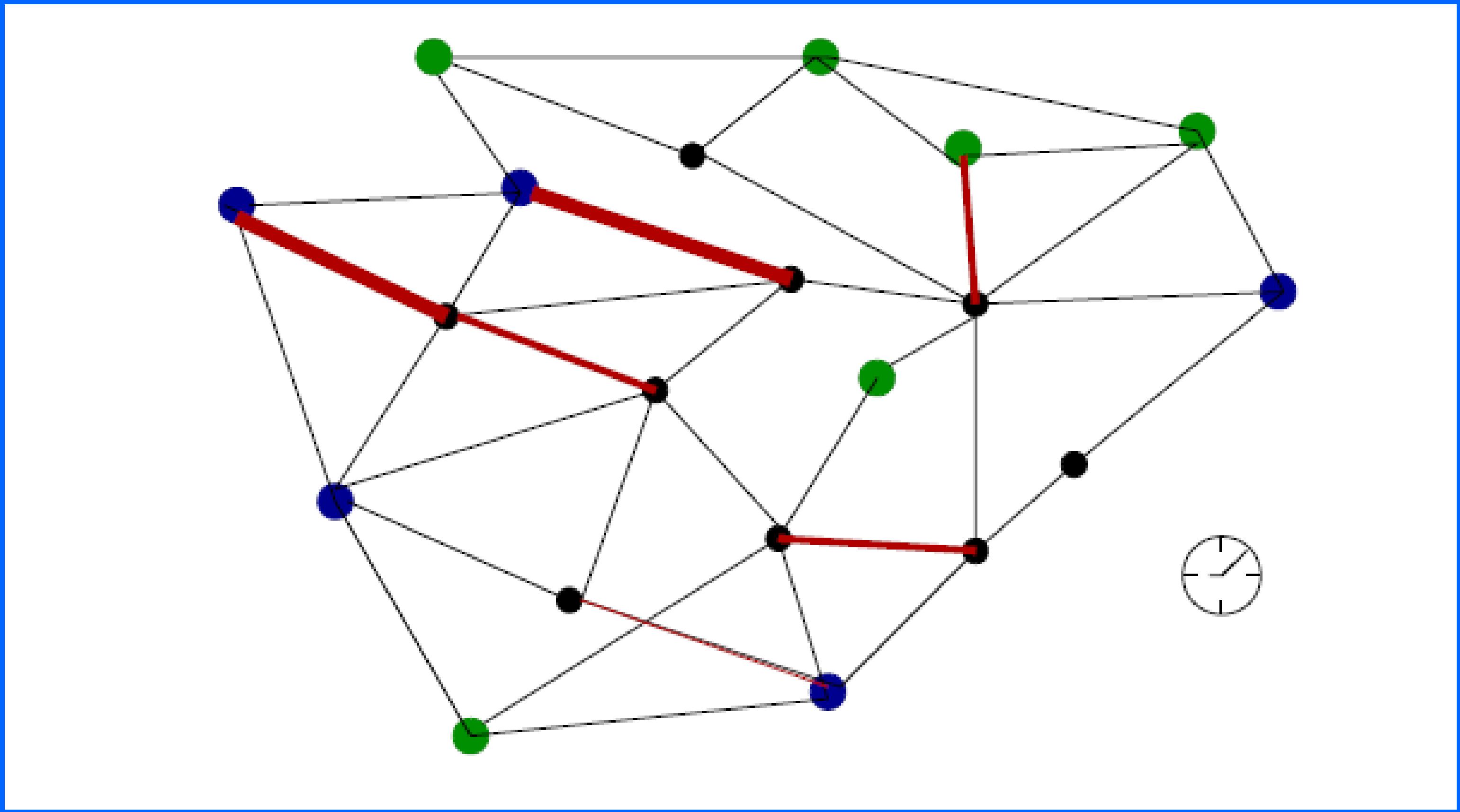


Credit: Daniel Bienstock

Cascading Outages in the Power Grid

Grid Resilience:

Cascade Model (Dobson, et al.)



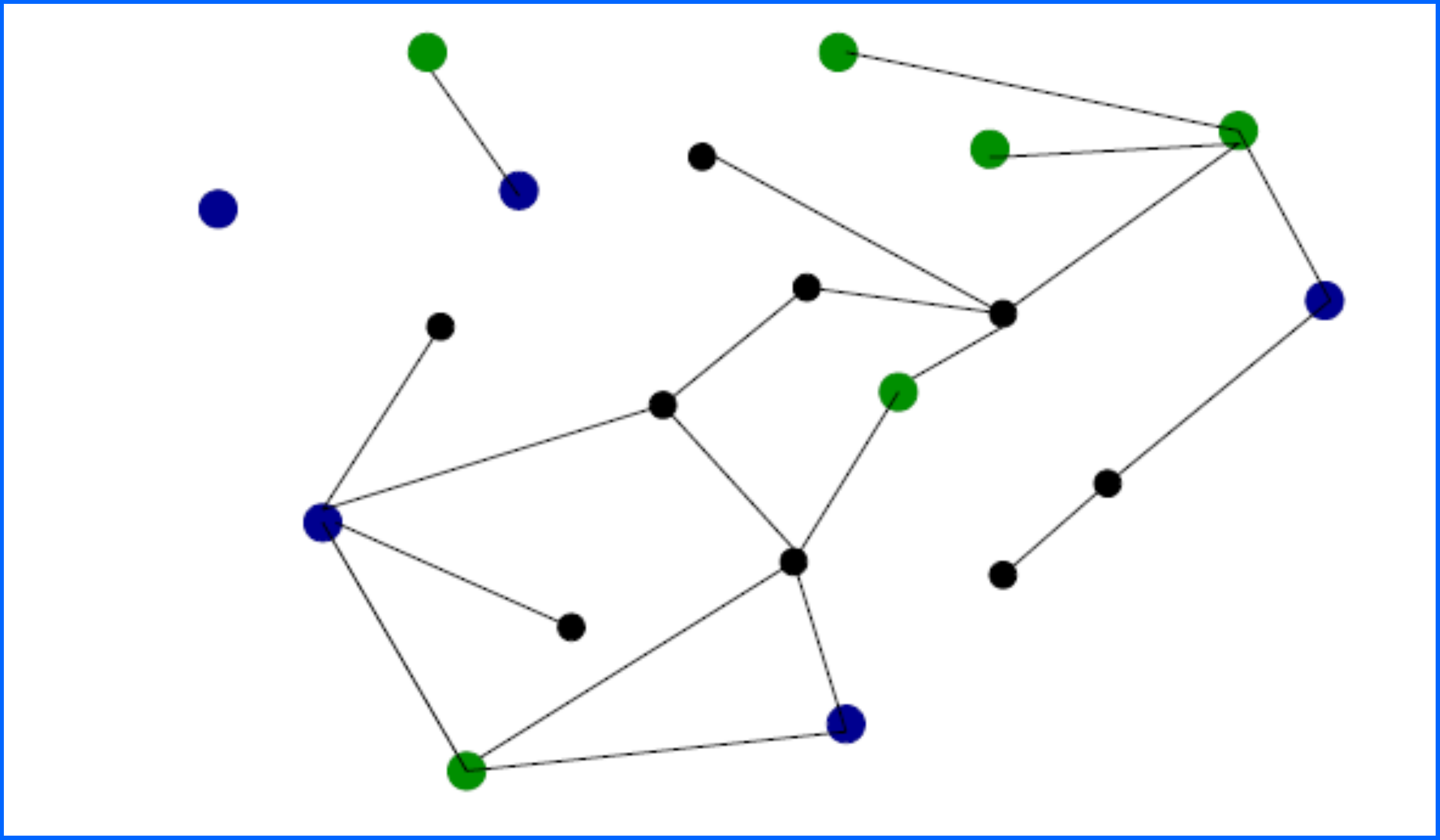
Credit: Daniel Bienstock

Increased flows on some lines

Cascading Outages in the Power Grid

Grid Resilience:

Cascade Model (Dobson, et al.)

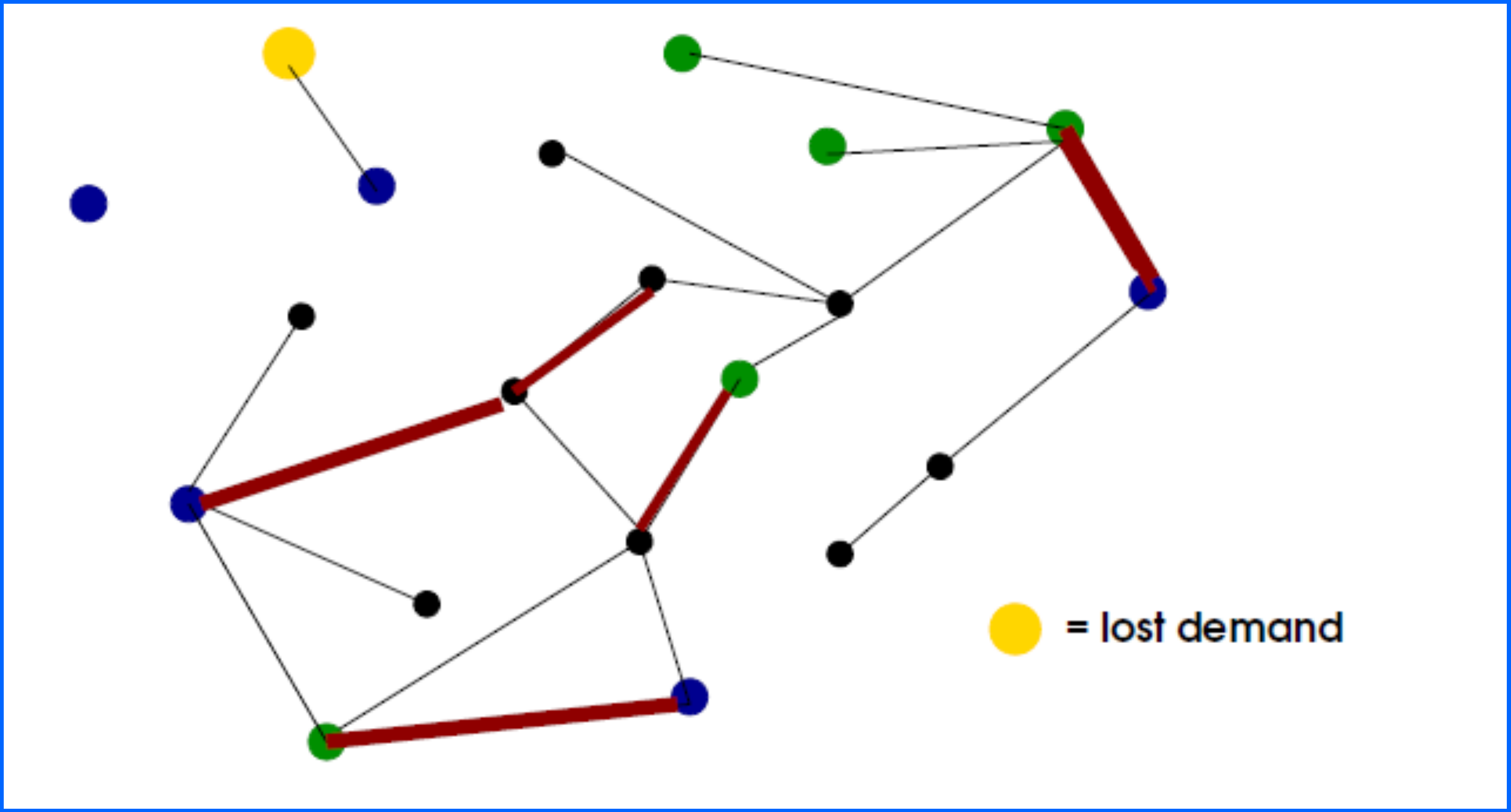


Credit: Daniel Bienstock

Cascading Outages in the Power Grid

Grid Resilience:

Cascade Model (Dobson, et al.)

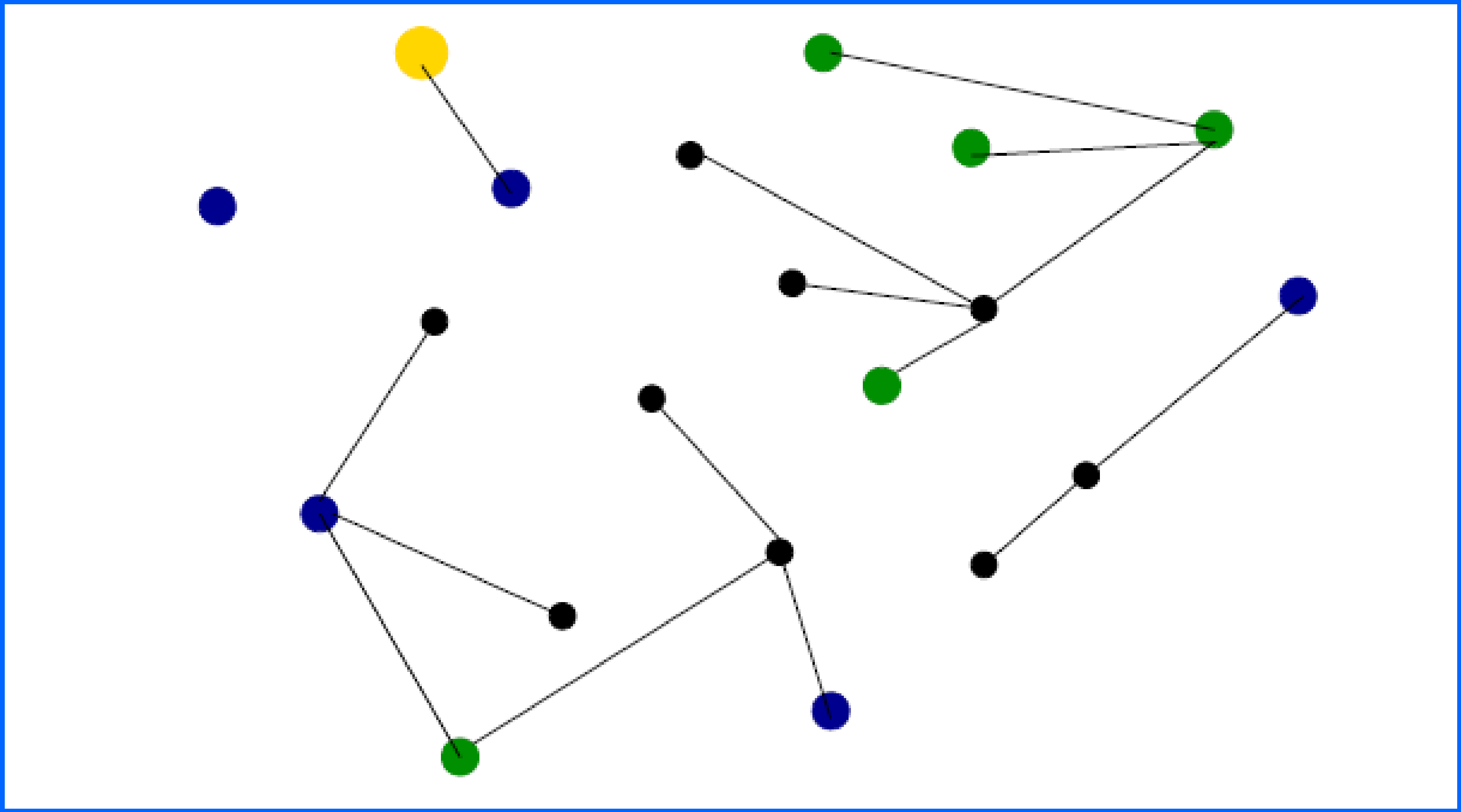


Credit: Daniel Bienstock

Cascading Outages in the Power Grid

Grid Resilience:

Cascade Model (Dobson, et al.)

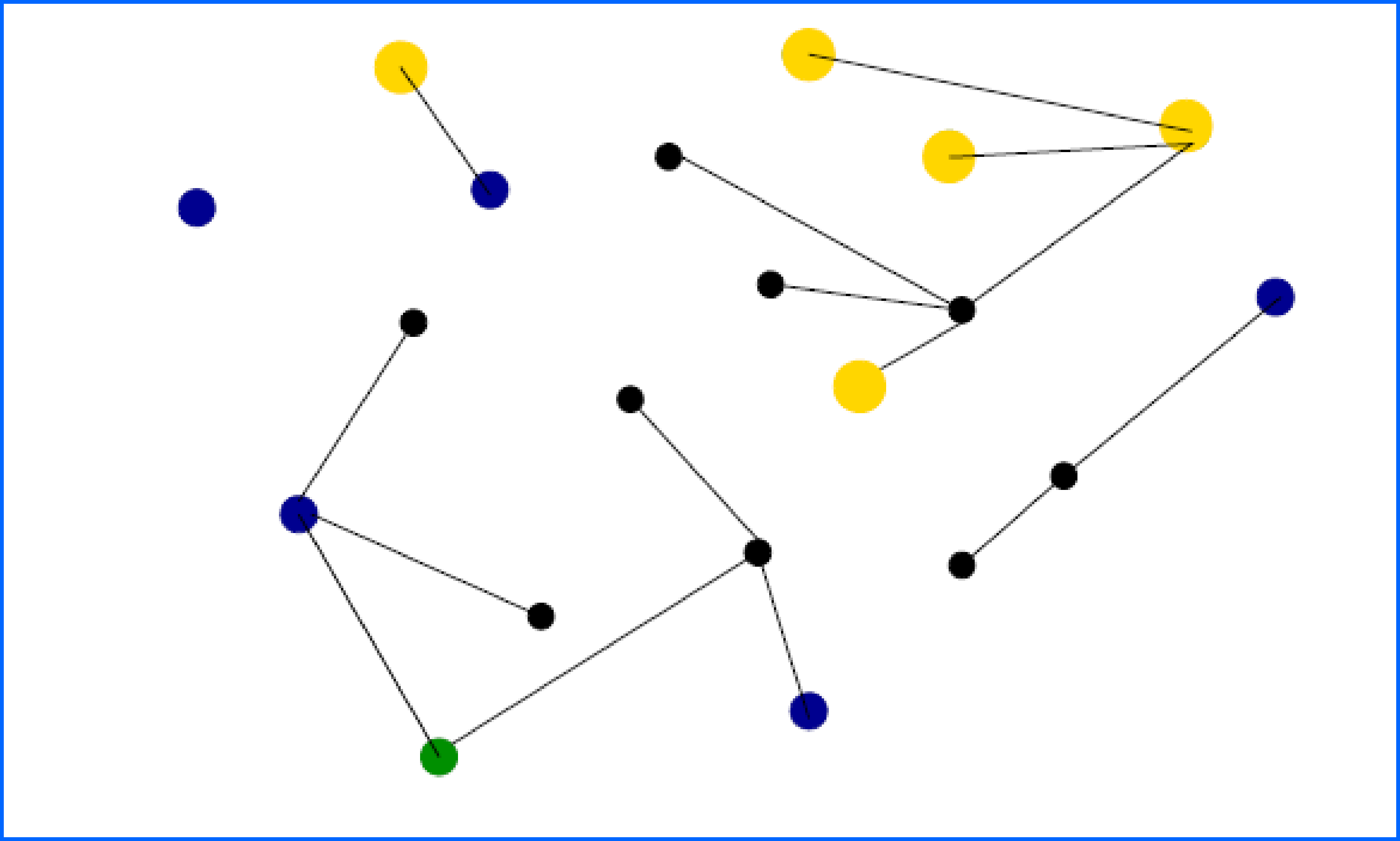


Credit: Daniel Bienstock

Cascading Outages in the Power Grid

Grid Resilience:

Cascade Model (Dobson, et al.)



Credit: Daniel Bienstock

Cascading Outages in the Power Grid

Grid Resilience:

- Cascade model of Dobson, et al.: Exercising "Control"
  - An initial "event" takes place
  - Reconfigure demands and generator output levels
  - New power flows are instantiated
  - Instead of waiting for the next set of faults to take place according to some stochastic process, use the cascade model to learn how to:
    - Take measurements and apply control to shed demand.
    - Reconfigure generator outputs; get new power flows

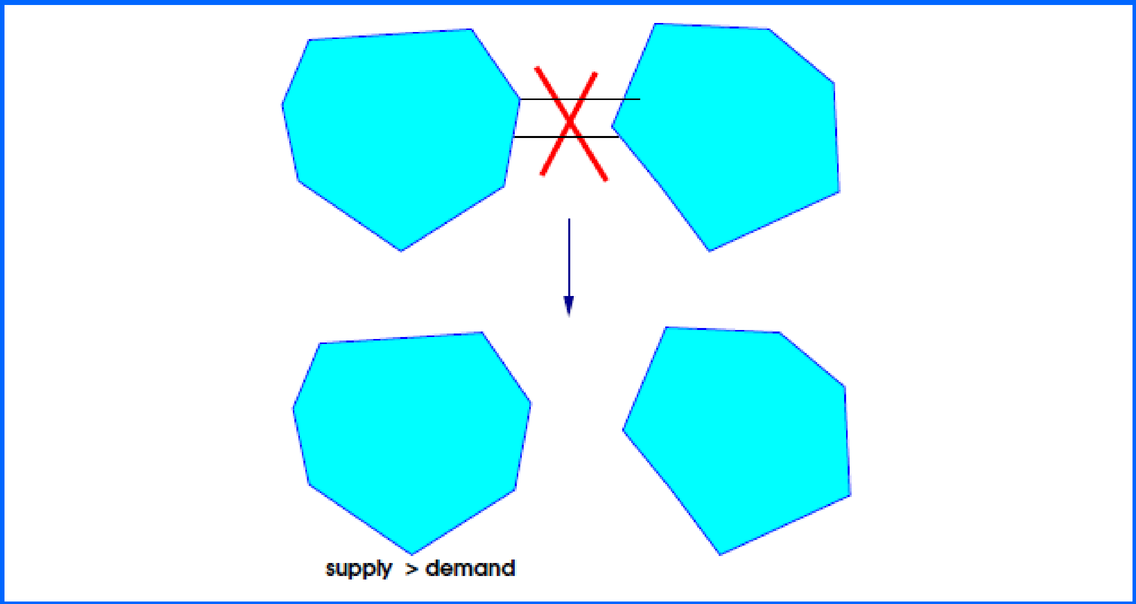
Cascading Outages in the Power Grid

Grid Resilience:

Cascade Model (Dobson, et al.)

•Use Model to Learn how Best to Create Islands to Protect Part of the Grid

•Hopefully the islands are small and in the rest of the grid, supply > demand.



Credit: Daniel Bienstock

Example IV: Infrastructure Resilience

- Critical infrastructure systems include:
  - Transportation systems
  - Telecom
  - Water supply systems
  - Wastewater systems
  - Electric power systems

- After a disruption, system begins to restore service until returning to performance level *at or below* the level before the disruption.

Image credit: Metropolitan Transportation Authority of the State of New York via Wikimedia commons, no changes made.

Example IV: Infrastructure Resilience

Hurricane Sandy, NJ

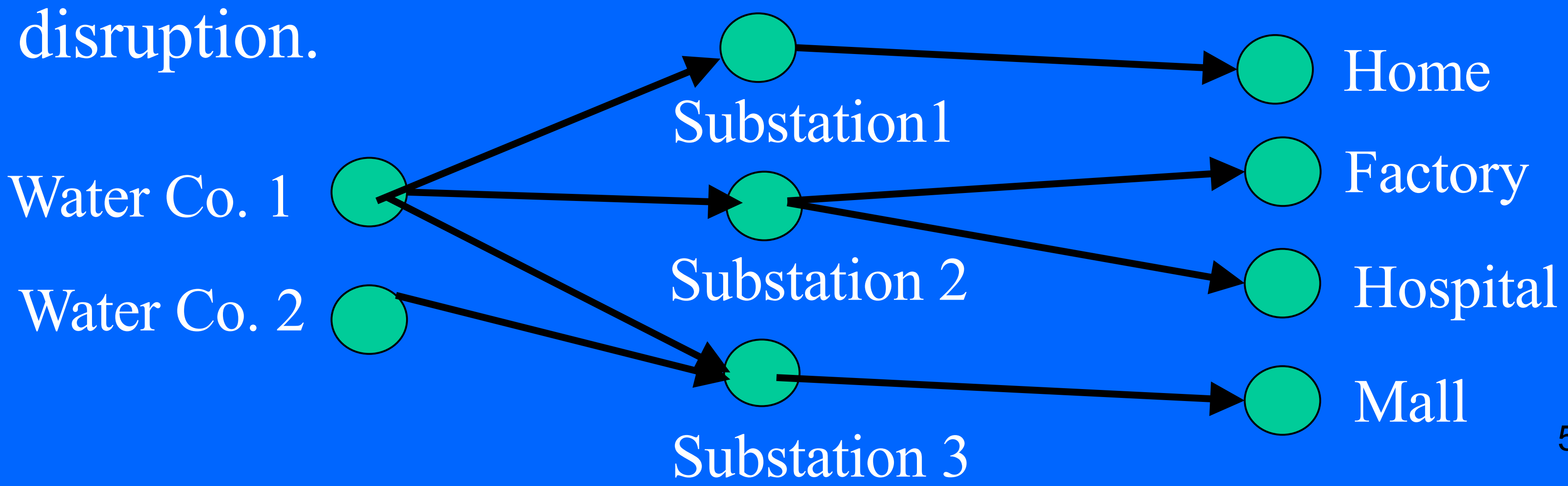
- The following models were developed by Sharkey and Pinkley (2019).
- Service is modeled by flows in networks.

Infrastructure Resilience

- Network now has vertices and directed edges (called ***arcs***).
- Flow can only go from vertex i to vertex j along an arc directed from i to j.
- Vertices represent:
  - Components that generate services (supply vertices)
  - Alter the routes of the services (transshipment vertices)
  - Consume services (demand vertices)
- Arcs move the services from one vertex to another.

Infrastructure Resilience

- Example: Water supply system
  - Supply vertices = water companies
  - Transshipment vertices = substations
  - Demand vertices are at households, factories, hospitals, malls, etc.
- Pipes are the arcs, and water is the flow.
- Meeting as much demand as possible is modeled as the classical maximum flow problem – both before and after a disruption.



Infrastructure Resilience

Maximum Flow Problem

- Consider a network $G = (V,A)$
- V = set of vertices, A = set of arcs.
- The arc i to j has a ***capacity*** u_{ij}.
- Fix one supply vertex s and one demand vertex t.
- There is a ***supply*** A(s) at s and a ***demand*** B(t) at t.
- We seek to assign a ***flow*** x_{ij} to the arc from i to j.
- The flow along that arc must at most the capacity:

$$x_{ij} \leq u_{ij}.$$

Infrastructure Resilience

Maximum Flow Problem

- ***Flow conservation***: the sum of flows on arcs into a vertex =- the sum of flows out of the vertex.
- If $A(i) = \{j: (i,j) \in A\}$, then this says:

$$\sum_{j \in A(i)} x_{ij} = \sum_{j:i \in A(j)} x_{ji}$$

- The total flow out of s cannot exceed the supply A(s) and the total flow into t cannot exceed the demand B(t).
- We seek to maximize the total flow that reaches t.

Infrastructure Resilience

Maximum Flow Problem

The ***Maximum Flow Problem*** seeks to determine the largest amount of flow that can reach t while:

- Keeping the flow on each arc at most the capacity
- Not exceeding total supply and demand
- Satisfying the flow conservation requirement at each vertex.

Infrastructure Resilience

Maximum Flow Problem

- The famous augmenting path algorithm (Ford-Fulkerson Algorithm) finds the maximum flow.
- Note: the maximum flow problem is a simplification.
- It assumes that there are no other constraints on flow.
- This might apply to supply chain networks:
  - E.g., physical goods move through intermediate warehouses and distribution centers.

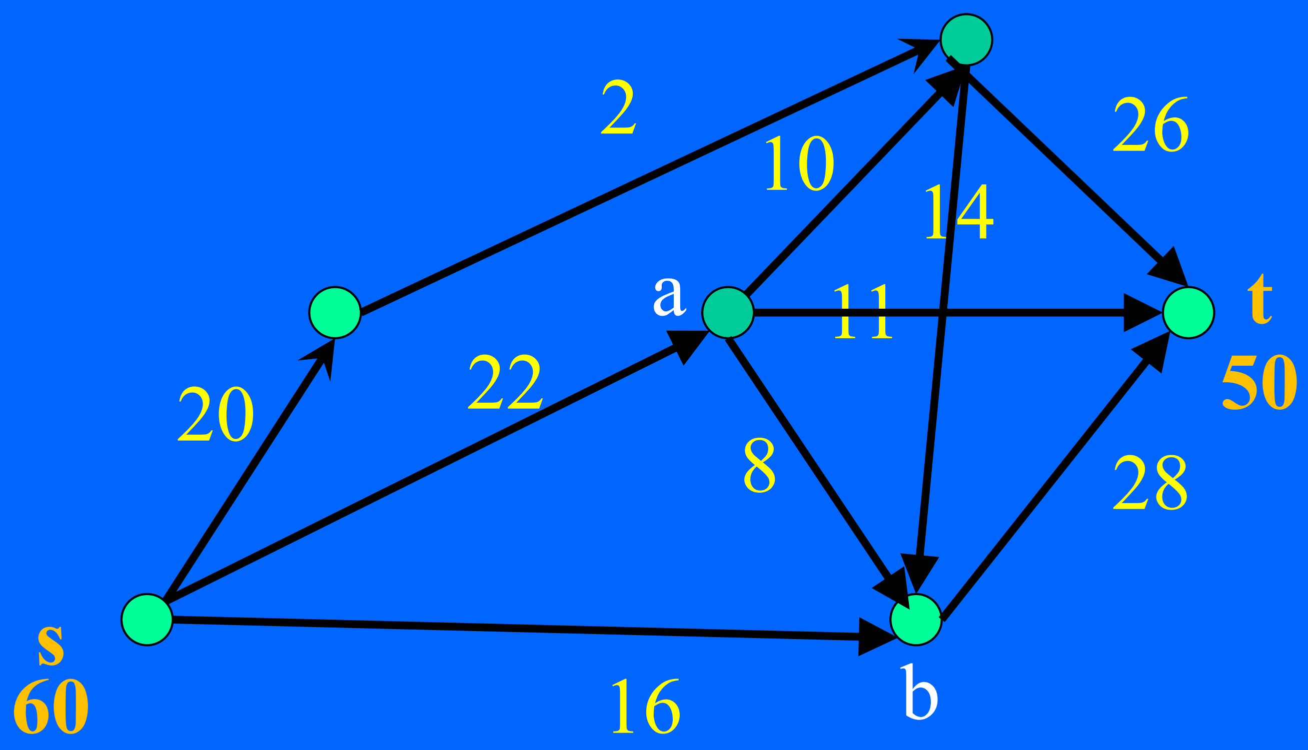
Infrastructure Resilience

Maximum Flow Problem

- For more complicated infrastructure, there are things like physical laws offering additional constraints.
- Example: Kirchhoff's and Ohm's Laws for power grid networks.
- Example: water distribution networks involve constraints involving the relation between flow of water and pressure.

Infrastructure Resilience

Maximum Flow Problem



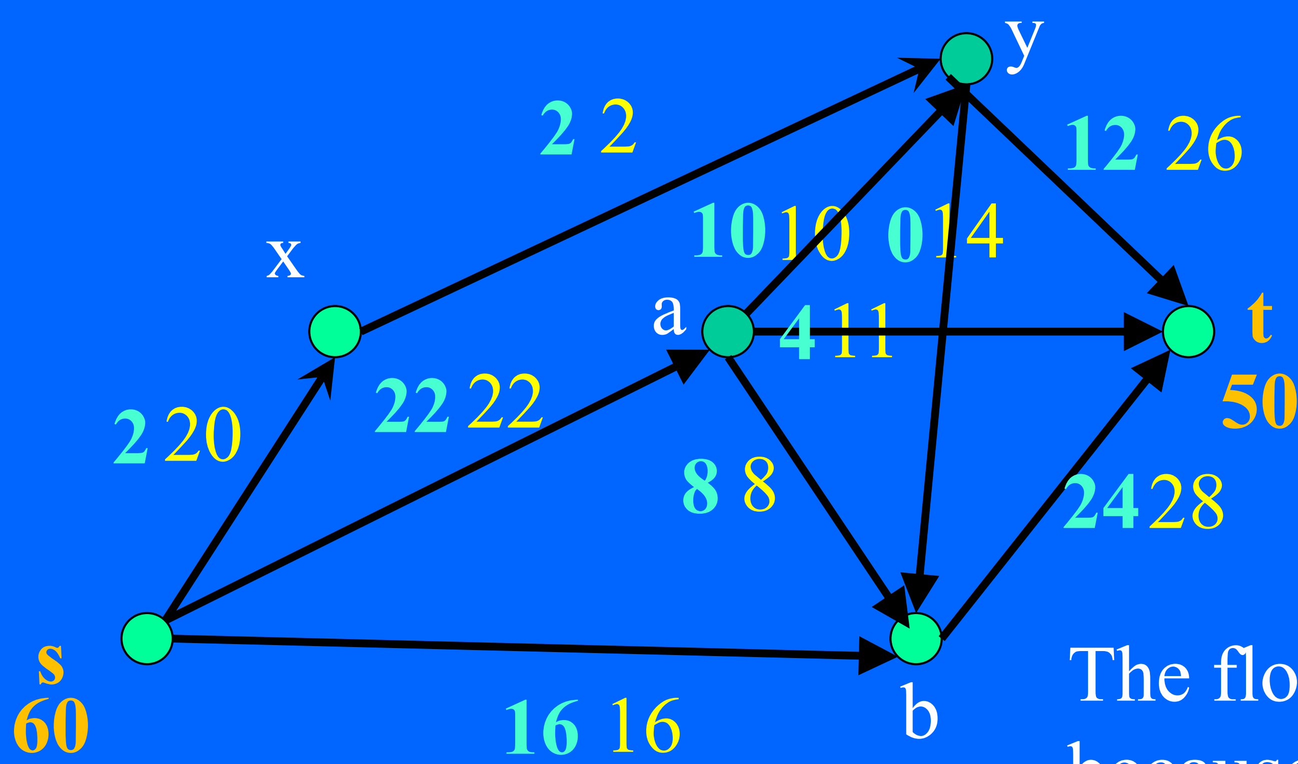
Key:

Orange: supply and demand

Yellow: Capacity

Infrastructure Resilience

Maximum Flow Problem



Total Flow:
40

The flow is max because flow from x to y is a most 2, so s to x is at most 2, so out of s is at most 40.

Key:
Orange: supply and demand
Yellow: Capacity
Green: Flow

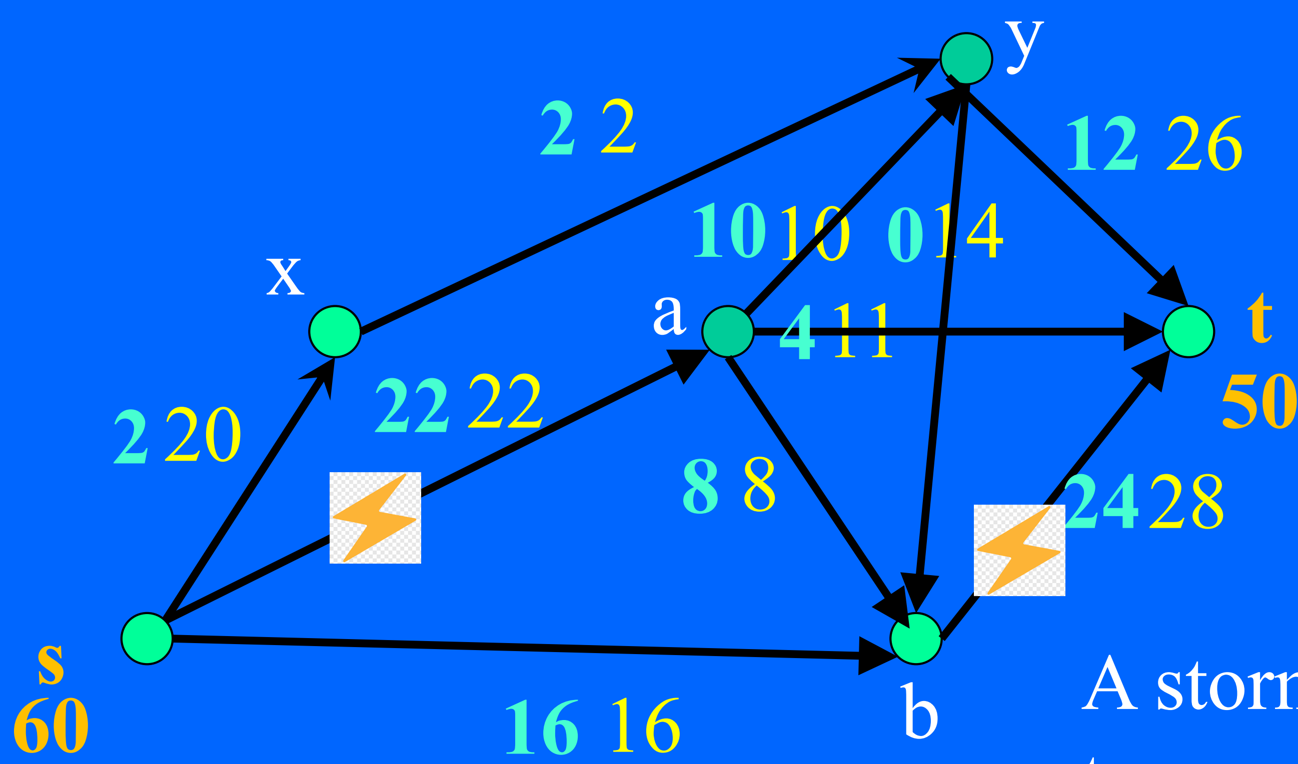
Infrastructure Resilience

Maximum Flow Problem

- If some of the arcs are destroyed, in what order should we reopen them?
- One goal: get closest to original maximum flow as early as possible.

Infrastructure Resilience

Maximum Flow Problem



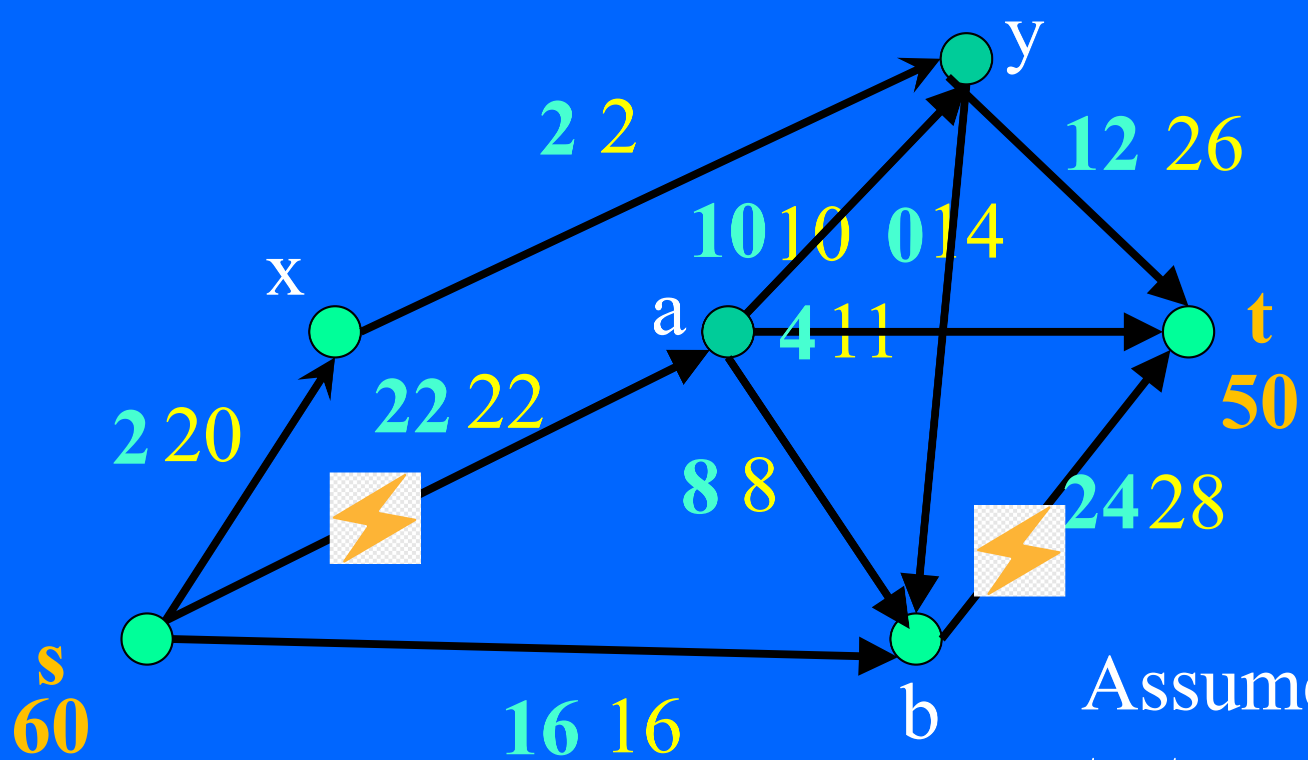
Total Flow:
40

A storm takes out two components. In what order should we bring them back online?

Key:
Orange: supply and demand
Yellow: Capacity
Green: Flow

Infrastructure Resilience

Maximum Flow Problem



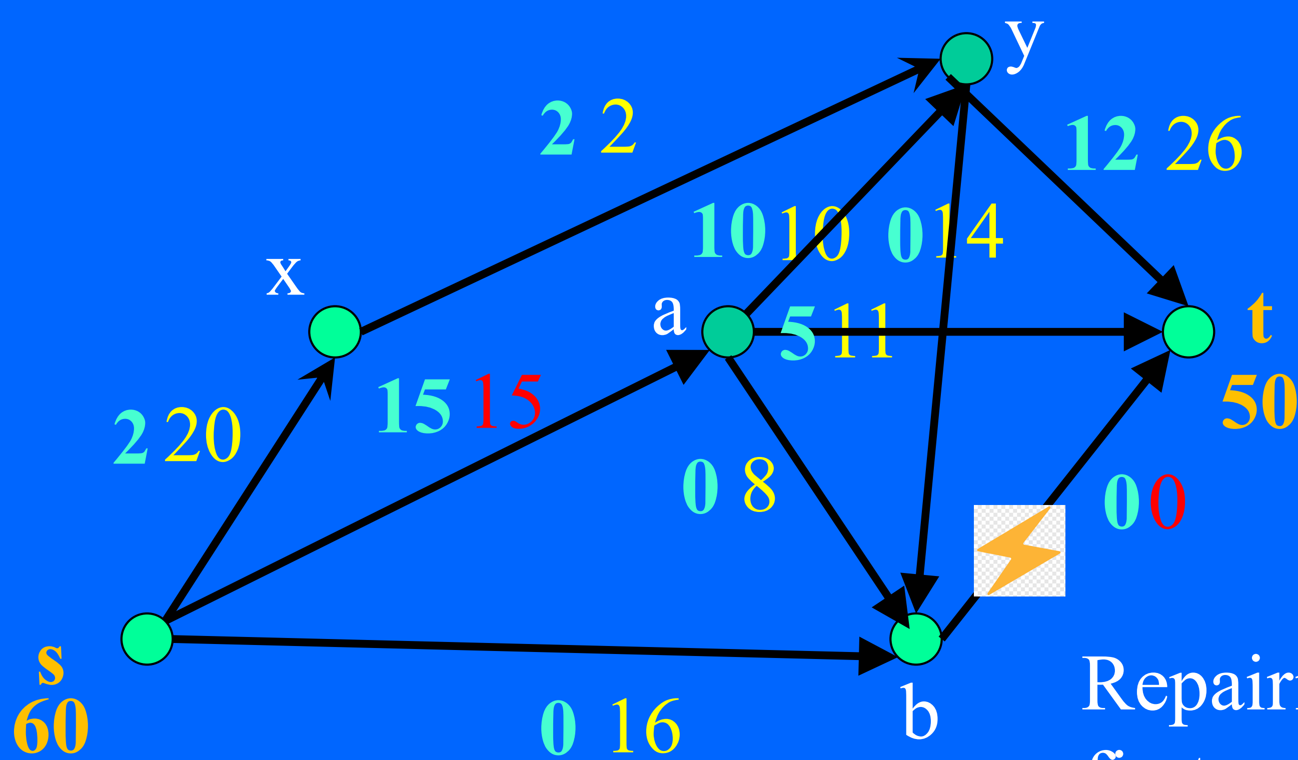
Total Flow:
40

Key:
Orange: supply and demand
Yellow: Capacity
Green: Flow

Assume that arc b to t can be repaired fully but arc s to a can only be repaired to a lower capacity of 15.

Infrastructure Resilience

Maximum Flow Problem



Total Flow: 17

Key:

Orange: supply and demand

Yellow: Capacity

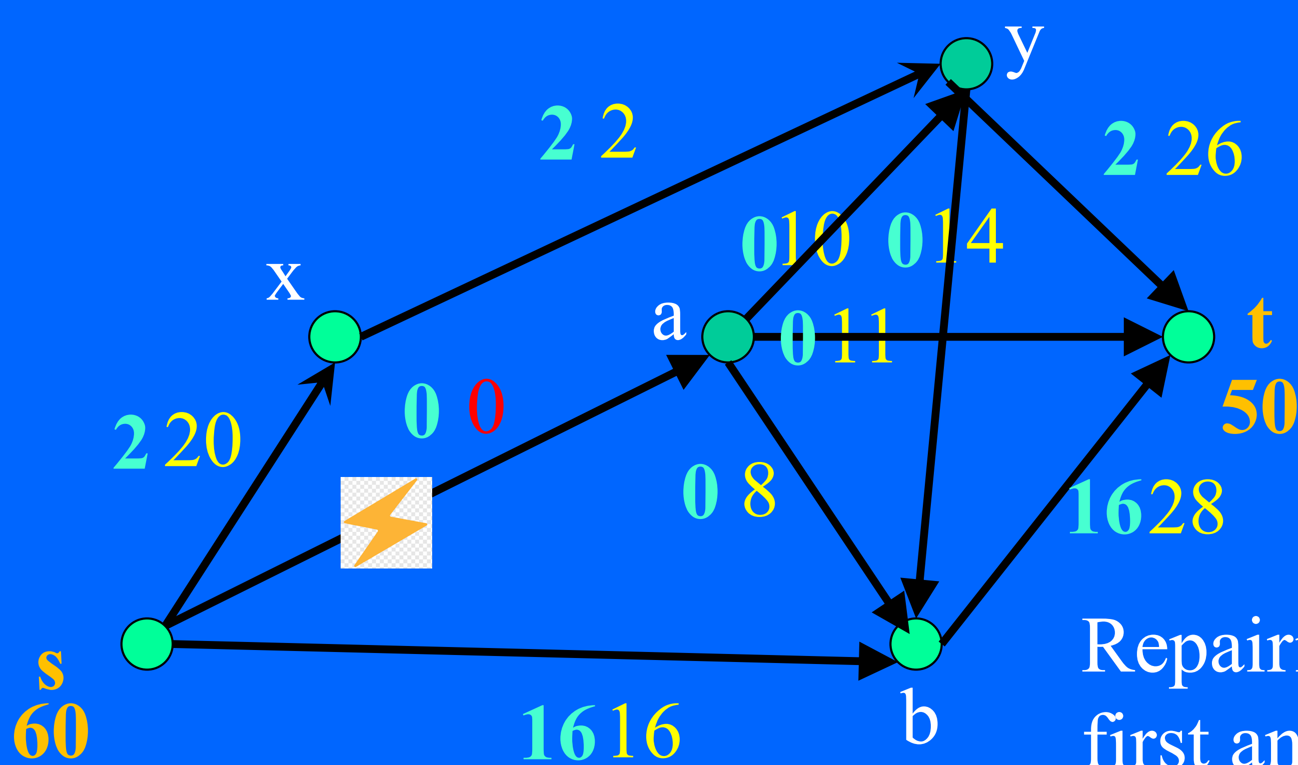
Green: Flow

Reduced Capacity

Repairing arc s to a first and leaving arc b to t unrepaired gives a max flow of 17.

Infrastructure Resilience

Maximum Flow Problem



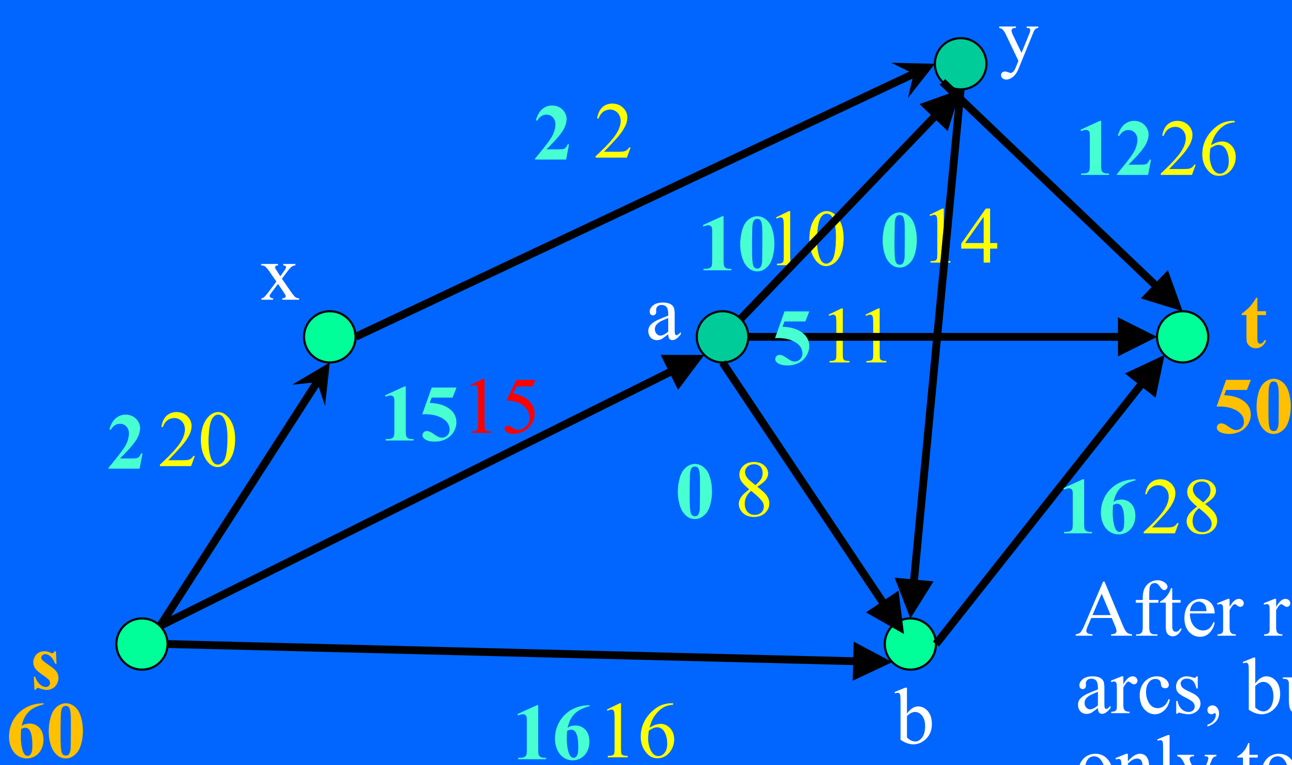
Total Flow:
18

Key:
Orange: supply and demand
Yellow: Capacity
Green: Flow
Reduced Capacity

Repairing arc b to t first and leaving arc s to a unrepaired gives a max flow of 18. So repair this arc first.

Infrastructure Resilience

Maximum Flow Problem



Total Flow:
33

Key:
Orange: supply and demand
Yellow: Capacity
Green: Flow
Reduced Capacity

After repairing both arcs, but arc s to a only to reduced capacity, means we only regained a reduced max flow of 33 – didn't fully restore flow of 40

Infrastructure Resilience

Maximum Flow Problem

- We made the simplifying assumption that there was one supply vertex and one demand vertex.
- In practice, there are many supply vertices s_1, s_2, …, and demand vertices t_1, t_2, …, with supply $A(s_i)$ at s_i and $B(t_i)$ at t_i.
- But we can reduce this to a single supply and demand vertex by adding a supply vertex S with supply $A(S) = \sum A(s_i)$ and an arc from S to each s_i with capacity $A(s_i)$ and a demand vertex T with demand $B(T) = \sum B(t_i)$ and an arc from each t_i to T with capacity $B(t_i)$.

Infrastructure Resilience

Maximum Flow Problem

- As different components of a network are repaired (to the extent possible), the maximum flow increases.
- How far off it is from the original max flow when repairs are done is one metric for resilience.
- How long it takes to complete the repairs is another metric for resilience.
- We turn next to the repair process.

Infrastructure Resilience
The Repair Process

- A different approach to reopening damaged components uses the theory of machine scheduling.
- After a disruption, repairs are made so services can be restored.
- Repairs use scarce ***resources***: work crews, equipment.

Credit: Patrick Cashin / MTA. Edited and cropped slightly by Daniel Case; via Wikimedia commons.

Infrastructure Resilience

The Repair Process

•Simplifying assumption: can only repair one component at a time (one vertex or arc).

•Need a schedule for when a resource is repairing a component.

•In the scheduling literature, we talk about jobs on a set of machines, and processing them.

•***Jobs*** here correspond to damaged components.

•***Machines*** correspond to work crews.

Infrastructure Resilience

The Repair Process

- Each job (damaged component) k has a different level of ***importance*** w_k.
- Each job also has a ***duration*** p_k.
- In the scheduling literature, each job k is assigned to a machine (work crew) m_k.
- The jobs assigned to a machine (work crew) m are given an order.
- So the ***completion time*** C_k of job k is the sum of the durations of all jobs assigned to the machine (crew) m_k that precede job k plus the duration of job k.

Infrastructure Resilience

The Repair Process

- There are various objectives for a good repair schedule.
- One is to minimize the weighted average completion time over all jobs, with the weight measuring the importance of the job.

$$\min \sum_k w_k C_k$$

- This is sometimes called the ***restoration performance***.
- If there is just one work crew, a greedy algorithm minimizes this: Repair component k in non-increasing order of the ratio w_k/p_k.

Infrastructure Resilience
The Repair Process

- A similar algorithm works if there many machines but each has the same processing time for repairing a given component.
- However, in general, most such scheduling problems are hard: NP-hard.

Infrastructure Resilience

Repairing Multiple Interdependent Infrastructures

- In a complex city, there are many infrastructures.
- They have interdependencies.
- Examples:
  - A subway (transportation infrastructure) needs power (electrical infrastructure) before it can be reopened.
  - A hospital needs both power and water before it can be reopened.
- This is modeled by studying a collection of networks, one for each infrastructure.
- A given infrastructure cannot operate until there is sufficient level of service (flow) on certain specific vertices in other infrastructures.

Infrastructure Resilience

Repairing Multiple Interdependent Infrastructures

- Scheduling repair of different infrastructures will therefore depend on these interdependencies.
- There is a considerable literature on this topic.
- Another complication: interdependencies among repair jobs – sometimes in different infrastructures.

Infrastructure Resilience
Repairing Multiple Interdependent Infrastructures

- Example:
  - To reopen subway lines, you need to repair a line.
  - Once you repair the lines, you need to run a test train on the line to check for safety and quality of the repair.
  - But power to the line must be restored before you can run a test train.

Subway tunnel pump train

Image credits:
Flood: ---=XEON=---
Pump train: Metropolitan Transportation Authority of the State of New York

Infrastructure Resilience
Repairing Multiple Interdependent Infrastructures

- Example:
  - Trees bring power lines down on a road.
  - First need to do a safety inspection to make sure it's safe to enter the road.
  - Then clear debris from the road.
  - Then repair downed power lines.

Closing Comment

- I have presented several simple examples of how to generate responses to disruptive events.
- Even these simple examples lead to problems that are "hard" in a precise sense.
- Another approach is to study ways to design graphs or networks so as to make them more resilient in case of disruption.
- That is a topic for another day.