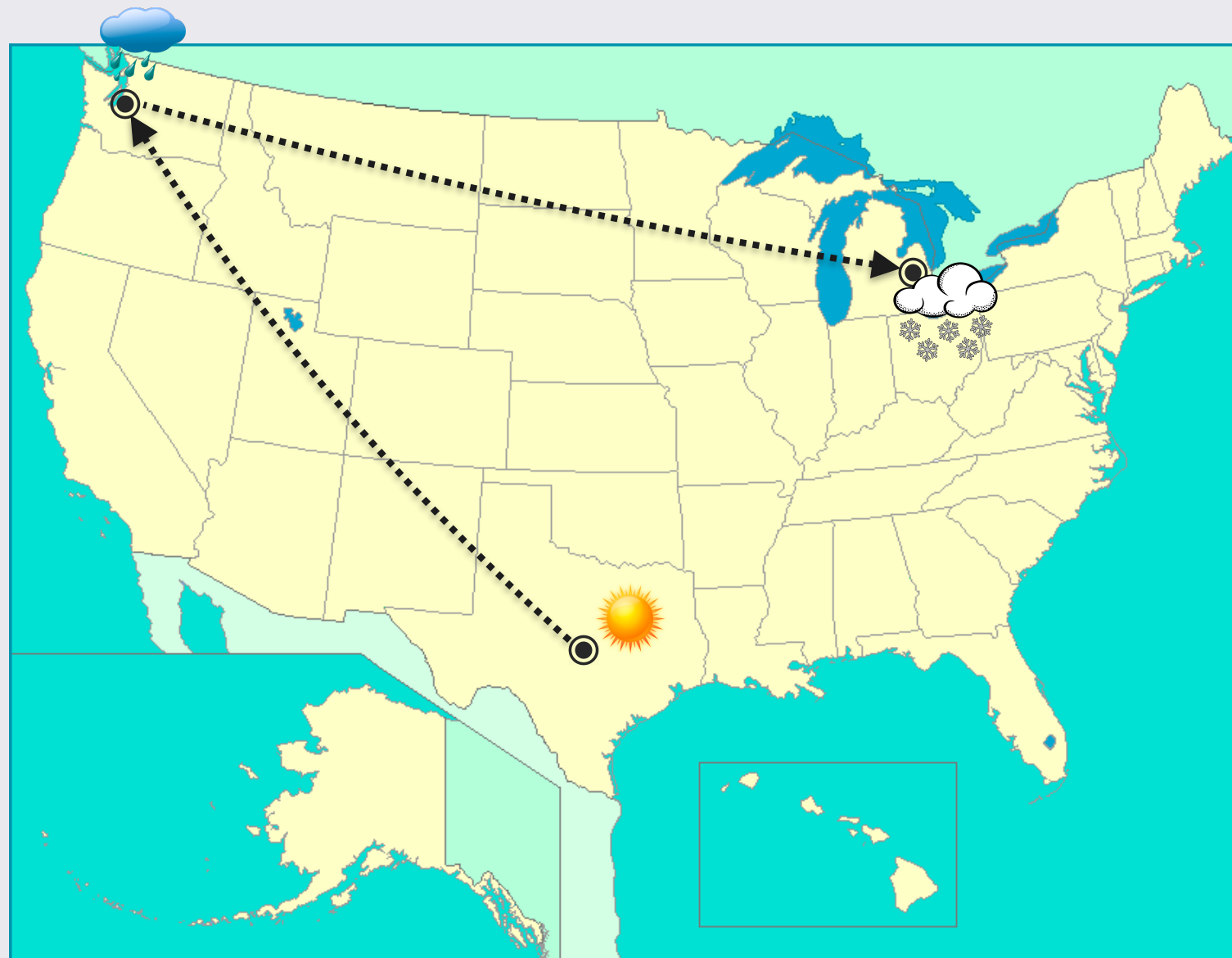# EECS 591
Distributed Systems

Manos Kapritsos
Fall 2021

ABOUT ME

Manos Kapritsos (manosk@umich.edu)

Area of research: Distributed Systems

ABOUT YOU AND ME

- I **love** teaching and interacting with my students

- I want to get to know you all by name
  - **Send me a selfie of you, if you are enrolled**
    - My email is: **manosk@umich.edu**
    - Subject: **[EECS591] Picture of <first name (preferred)> <lastname>**
  - State your first name when I call on you
- I'm here to help. Come to me with any question!
  - course-related: office hours
  - Life, The Universe, and Everything: any time

IF YOU NEED MORE HELP...

Our TA, **Tony Zhang**, is here to help

A Jedi Master in Distributed Systems

Office hours: Wednesday 4-5pm

Zoom link on website and Piazza

ADMINISTRIVIA

- Class: M-W 12-1:20pm

  - Lectures will be recorded

  - Link on website and Piazza

- Office hours

  - Tuesday, 2-3pm

  - Queue: https://oh.eecs.umich.edu/courses/eecs591

    - Wait to be dequeued before joining Zoom

  - Zoom links on website and Piazza

  Contact me by email: **manosk@umich.edu**

CLASS RESOURCES

- Class webpage

  - https://web.eecs.umich.edu/~manosk/eecs591-f21.html

- Piazza forum, make sure to subscribe

  - **Ask** questions, **monitor** questions, **answer** questions

ENROLLMENT

- Enrolled: 27

  - Outstanding overrides: 5

  - Capacity: 30

  - Waitlisted: 20

- # of overrides = # of drops - 2

ABOUT EECS 591

- First part: fundamentals

  - Undergrad style: I present the material

- Second part: systems

  - Graduate style

    - Everyone reads the paper

    - One or two people present and lead the discussion

Adminis-not-so-Trivia

First part: fundamentals

- 2 problem sets

- 1 implementation project

- 1 midterm exam

  - Held in class

Second part: systems

- 1 paper review per class

- 1 in-class presentation

  - groups of 1-2

- 1 research project

  - groups of 3

- No final exam

GRADING

- Problem sets: 10% each

- Implementation project: 15%

- Midterm exam: 20%

- Reviews: 10%

- Class participation: 10%

- In-class presentation: 10%

- Research project: 15%

RECIPE FOR SUCCESS

1. Attend classes and be active
2. Work on problem sets and implementation project
3. Read papers and participate in the discussion
4. Make good progress on research project

THANK YOU, LORENZO ALVISI!

For the slides and everything else!

A FEW QUESTIONS

- How do we order events in a distributed system?

- How do we reach agreement in a distributed system?

- What if part of the system is down? Can we still do useful work?

- What if, instead, part of the system becomes "possessed" and starts behaving arbitrarily? All bets are off?

- Are there problems that simply cannot be solved?

"To be or not to be?"

"Why do we drive on parkways and park on driveways?"

"What do you say when someone says you're in denial, but you're not?"
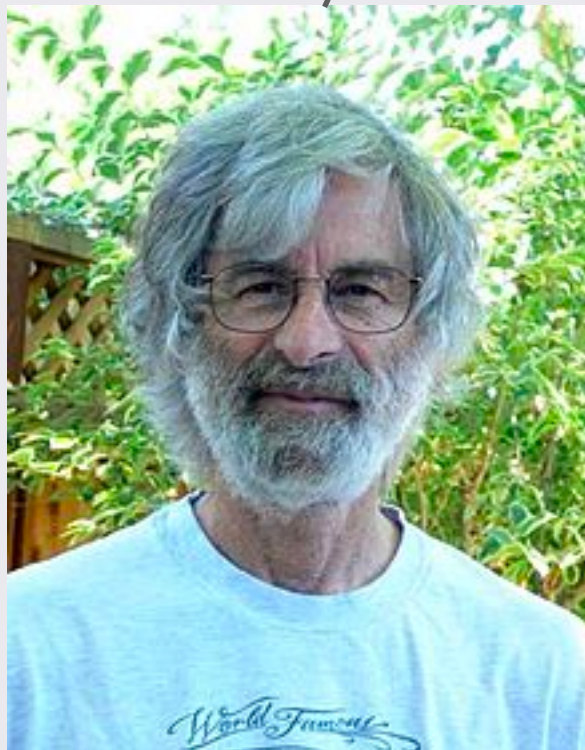
What is a distributed system?

What is a distributed system?

"A distributed system is one in which the failure of a computer you didn't even know existed can render your own computer unusable."

Leslie Lamport

What is a distributed system?

A collection of distinct processes that:

- Are spatially separated

- communicate with one another by exchanging messages

- have non-negligible communication delay

- do not share fate

Save Western civilization!
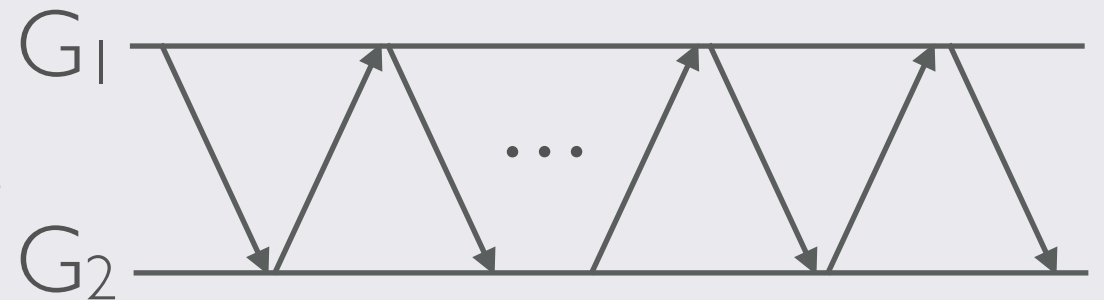
TWO GENERALS' PROBLEM

Claim: there is no protocol that guarantees that the Generals will always attack simultaneously.

Proof: by contradiction

Let n be the smallest number of messages needed by a solution

Consider the nth message m_{last}

G₁

- The decision of the sender of m_{last} cannot depend on it being received G₂

- The decision of the receiver of m_{last} cannot depend on it being received, because in some executions m_{last} could be lost

- So both the sender and the receiver would come to the same decision, even if m_{last} was never sent

- We now have a solution with only $n - 1$ messages: Contradiction!

Solving the two generals' problem requires **common knowledge**: (everyone knows that everyone knows that...)

Alas! Common knowledge **cannot be achieved** by communicating through unreliable channels

THE CASE OF THE MUDDY CHILDREN

(truthful, perceptive and intelligent)

A group of children go playing. Some of them (let's say k) get mud on their forehead.

Their father arrives, looks around and says: "Some of you have mud on your forehead."

Mud?
What mud?

The father then says repeatedly: "If you know that you have mud on your forehead, raise your hand."

What will happen?

Elementary...

Claim

The first $k - 1$ times the father asks, **no** child will raise their hand. On the kth time **all** dirty children will raise their hand.

Proof

By (informal) induction on k

- $k = 1$ The child with the muddy forehead sees that no other child has mud. Father said someone has, so it must be her.

- $k = 2$ Two muddy children: Alice and Bob. The first time, they each see the other, so they don't raise their hand. When Alice sees that Bob doesn't raise their hand, she realizes she has mud, too.

- $k = 3$ And so on...

ELEMENTARY?

Consider the case where $k = 2$

If the father does not say "*some of you have mud on your forehead*", no child will ever raise their hand!

- $k = 1$ The child with the muddy forehead sees that no other child has mud. **Father said someone has,** so it must be her.

But every child **already knows** that someone has a muddy forehead

Paradox?

COMMON KNOWLEDGE

Let p = "Someone's forehead is dirty"

Everyone knows p

But, unless father speaks, if $k = 2$, not everyone knows that everyone knows p

- A and B are dirty. Before the father speaks, A does not know whether B knows p

If $k = 3$, not everyone knows that everyone knows that everyone knows p

WHAT WOULD HAPPEN IF...

…the father told each child in private that someone's forehead is muddy?

…and every child had (in secret) placed a miniature microphone on all the other children so they can hear what the father told them in private?