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 picture of you**
    - My email is: manosk@umich.edu
    - Subject: [EECS591] Picture of <first name (preferred)> <lastname> <UMID>
    - 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, Remzi Aksoy, is here to help

A Jedi Master in Distributed Systems

Office hours: Thursday 1-2pm, BBB 1637
Class: M-W 3-4:30pm, DOW 1005

Office hours

- Monday, Wednesday, 4:30-5pm, DOW 1005
- Tuesday 2-3pm, BBB 4824

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

- Class webpage
  - [https://web.eecs.umich.edu/~manosk/eecs591-f18.html](https://web.eecs.umich.edu/~manosk/eecs591-f18.html)
- Piazza forum, make sure to subscribe
  - **Ask** questions, **monitor** questions, **answer** questions
ENROLLMENT

- # of overrides depends on # of drops
 About EECS 591

- First part: fundamentals
  - Undergrad style: I present the material
- Second part: systems
  - Graduate style
    - Everyone reads the paper
    - 1-2 students present and lead the discussion
ADMINIS-Not-So-Trivia

First part: fundamentals
- 2 problem sets
- 1 implementation project
- 1 midterm exam

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: 25%
- Reviews and class participation: 10%
- In-class presentation: 10%
- Research project: 20%
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
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?
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

Both generals must attack together or face defeat.

Communication is only by messengers sneaking through the valley.

Messengers may not make it through…
**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 $n^{th}$ message $m_{last}$

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

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

What will happen?
Claim
The first \( k - 1 \) times the father asks, no child will raise their hand. On the \( k^{th} \) 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...
What if the father does not say “someone has mud on their forehead”?

Consider the case where $k = 2$

Claim: in this case, 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?
Let \( p = \text{“Someone’s forehead is dirty”} \)

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?
Parallel Worlds

Each node is labeled with a tuple that represents a possible world: 
(1, 0, 1) represents a world where only child 2 does not have a muddy forehead.

Each edge is marked after the child for which the two endpoints are both possible worlds.

- Child 1
- Child 2
- Child 3
AFTER THE FATHER SPEAKS

The state \((0, 0, 0)\) becomes impossible

All edges that depart from it are eliminated

- Child 1
- Child 2
- Child 3
After the father speaks

The state $(0, 0, 0)$ becomes impossible

All edges that depart from it are eliminated

(1, 0, 0) — (0, 0, 1) — (0, 1, 0) — (1, 1, 0) — (1, 1, 1)

--- Child 1

----- Child 2

· · · · · · Child 3
IF NO ONE RAISES THEIR HAND AFTER THE FIRST ROUND

All states with a single 1 are eliminated

All edges that depart from them are eliminated
IF NO ONE RAISES THEIR HAND AFTER THE FIRST ROUND

All states with a single 1 are eliminated

All edges that depart from them are eliminated

(1, 0, 1) → (1, 1, 1) → (1, 1, 0) → (0, 1, 1)

Child 1

Child 2

Child 3
Both muddied children see that the second child is clean.

All worlds \((*, 1, \ast)\) are eliminated for the muddy children.
Both muddied children see that the second child is clean

\[(1, 0, 1)\]

All worlds \((*, 1, *)\) are eliminated for the muddy children

---------- Child 1
---------- Child 2
---------- Child 3
The clean child is still uncertain

Child #2 sees that the other children are muddy

It eliminates worlds (1, 1, 0) and (0, 1, 1)
The clean child is still uncertain

Child #2 sees that the other children are muddy

It eliminates worlds (1, 1, 0) and (0, 1, 1)