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

A few intriguing questions

- How do we talk about a distributed execution?
- Can we draw global conclusions from local information?
- Can we coordinate operations without relying on synchrony?
- For the problems we know how to solve, how do we characterize the “goodness” of our solution?
- Are there problems that simply cannot be solved?
- What are useful notions of consistency, and how do we maintain them?
- 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?

Saving the world before bedtime
Claim: There is no non-trivial protocol that guarantees that the Romans will always attack simultaneously.

Proof: By contradiction, consider a protocol that solves the Two Generals problem using the least number of messages. Let that number be $n$. Consider the $n$-th message $m_{\text{last}}$.

- The state of sender of $m_{\text{last}}$ cannot depend on $m_{\text{last}}$ receipt.
- The state of receiver of $m_{\text{last}}$ cannot depend on $m_{\text{last}}$ receipt because in some executions $m_{\text{last}}$ could be lost!
- So both sender and receiver would come to the same conclusion even without sending $m_{\text{last}}$.
- We now have a new solution requiring only $n-1$ messages!
Two Generals’ Problem

Claim: There is no non-trivial protocol that guarantees that the Romans will always attack simultaneously.

Proof: By contradiction, consider a protocol that solves the Two Generals problem using the least number of messages. Let that number be \( n \). Consider the \( n \)-th message \( m_{\text{last}} \):
- The state of sender of \( m_{\text{last}} \) cannot depend on \( m_{\text{last}} \) receipt.
- The state of receiver of \( m_{\text{last}} \) cannot depend on \( m_{\text{last}} \) receipt because in some executions \( m_{\text{last}} \) could be lost.
- So both sender and receiver would come to the same conclusion even without sending \( m_{\text{last}} \).

Conclusion: A solution requires reliable message delivery.

If only I had known...

- Solving the two generals problem requires “common knowledge”
  - “I know that you know that I know that you know...”
- Also interesting: distributed knowledge
  - By pooling knowledge together, the members of a group know \( \varphi \) even if no single member knows \( \varphi \).

Muddy Children: the Puzzle

- \( n \) children playing
- Mom says: “Don’t get muddy!”
- Children are truthful, perceptive, intelligent
- \( k \) children get mud on their forehead
- Daddy comes and says: “At least one of you has mud on your forehead!”

Muddy Children: the Puzzle

- \( n \) children playing
- Mom says: “Don’t get muddy!”
- Children are truthful, perceptive, intelligent
- \( k \) children get mud on their forehead
- Daddy comes and says: “At least one of you has mud on your forehead!”
- Dad then asks over and over:
  - “Does any of you know whether you have mud on your own forehead?”
- What happens?