THE GRAINGER COLLEGE OF ENGINEERING
SIEBEL SCHOOL OF COMPUTING AND DATA SCIENCE

CS 521

Technological Foundations of Blockchain and Cryptocurrency

Grigore Rosu

Topic 4 – Consensus

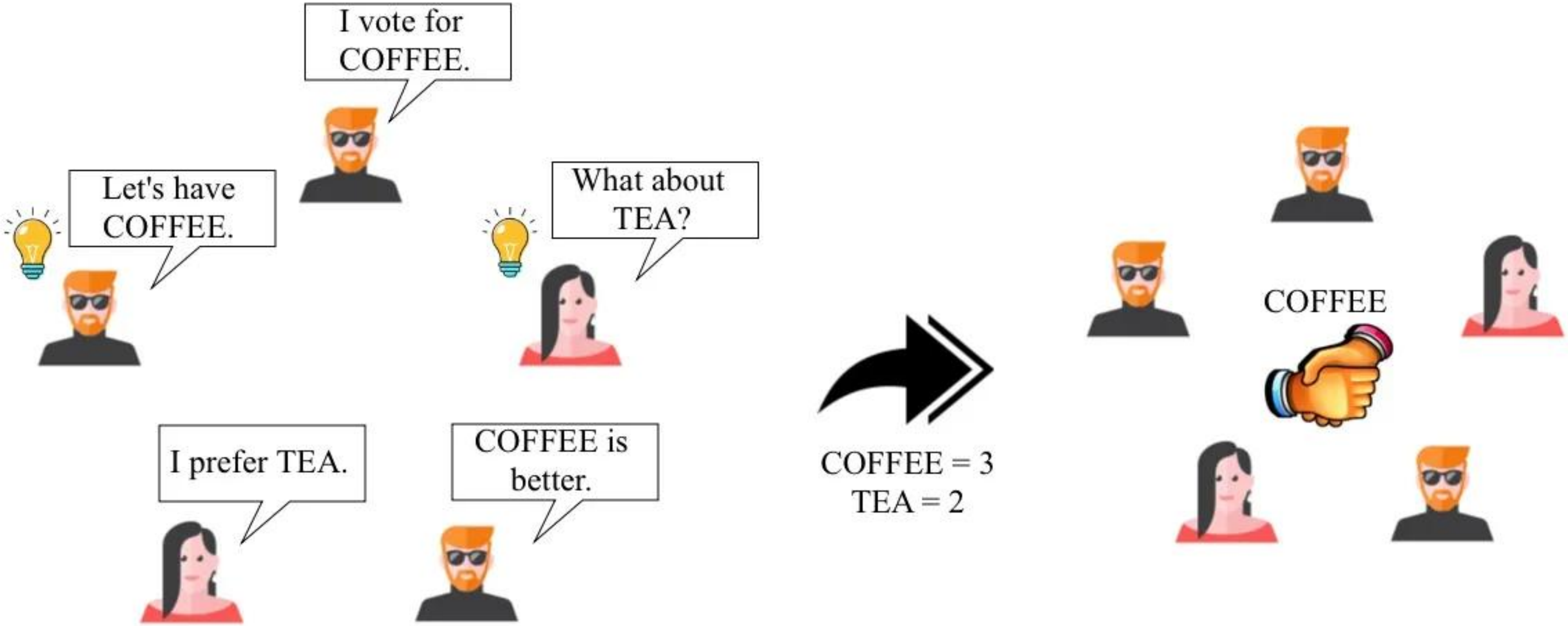
ILLINOIS

Thanks

- Musab Al Turki

The Consensus Problem

- Distributed processes agreeing on a single, correct value, despite failures
- Much harder when decentralized



The Consensus Problem

- Distributed processes agreeing on a single, correct value, despite failures.
- Much harder when decentralized

Key Properties

- **Agreement**: All non-faulty nodes must agree on the same value
- **Validity**: The agreed value must be a correct one (proposed by a non-faulty node)
- **Termination**: All non-faulty nodes must eventually decide on a value

Main Challenges

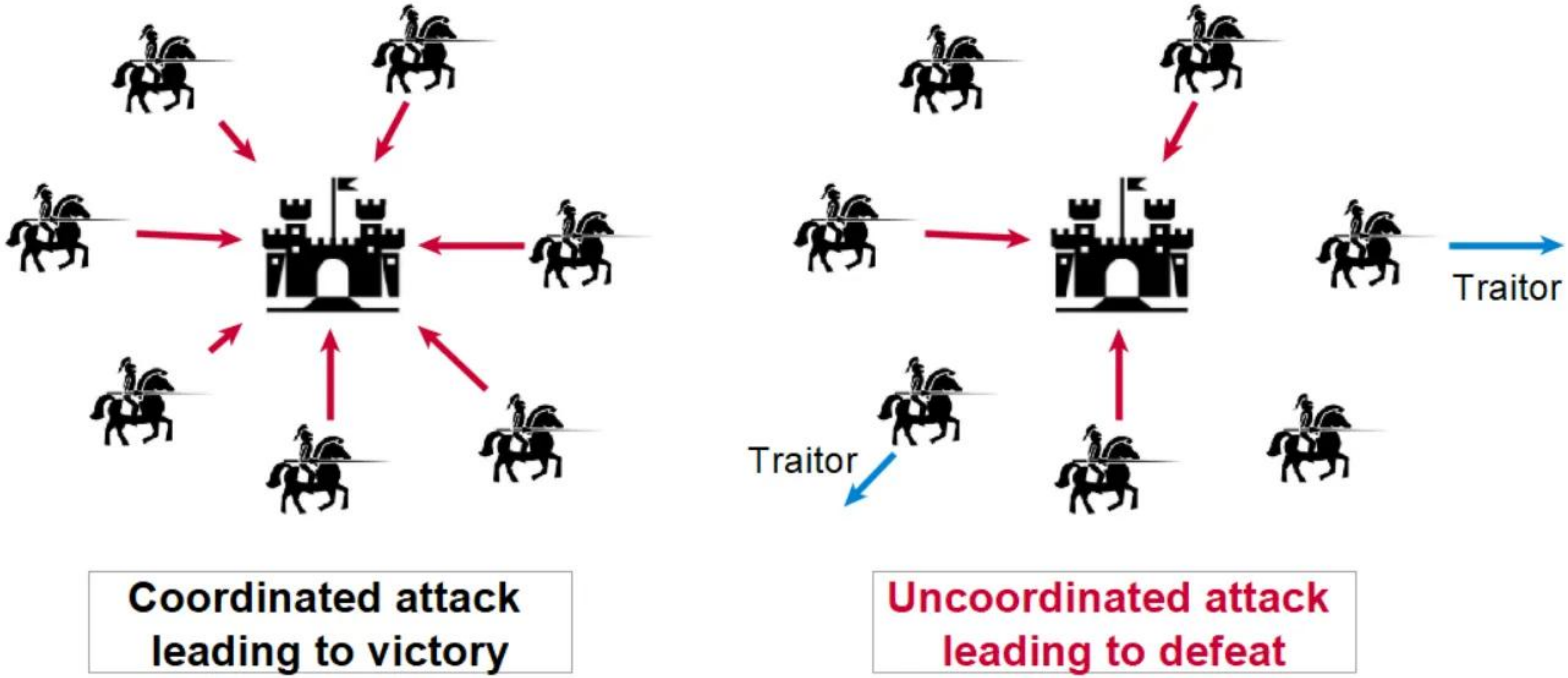
- **Byzantine Faults** : Some nodes may act arbitrarily or maliciously or may crash
- **Asynchrony**: Nodes operate at different speeds with unpredictable message delivery delays

The Byzantine Generals Problem

1982: Lamport, Shostak, and Pease

Formalization of BFT Consensus

source



Coordinated attack leading to victory

Uncoordinated attack leading to defeat

Can only tolerate up to f faulty generals if there are 3f + 1 generals

The FLP Impossibility Result

1985: Fischer, Lynch, and Paterson

No **deterministic** consensus algorithm can simultaneously guarantee **safety** and **liveness** (correctness) in an **asynchronous** distributed system where even one process may **crash**

Real-world protocols must introduce additional assumptions, such as:

- Crash failure recovery mechanisms (e.g., Paxos, Raft)
- Synchrony or partial synchrony (e.g., PBFT, Tendermint, HotStuff).
- Randomization (e.g., Bitcoin, Algorand, Bullshark)

Or they may sacrifice safety for liveness or vice versa (e.g. Paxos, PBFT, Bitcoin, Ethereum)

Blockchain Consensus

A state-machine-replication problem: Agreement on a sequence of states

- Clients submit transactions to nodes
- Each node locally maintains an ordered sequence of txs (in blocks)
- Nodes need to agree on a canonical, totally ordered sequence of txs (defining the canonical sequence of states)
  - Assume an initial state (the genesis state)
  - In each round, the proposed state is valid if the ordered list of transactions in the proposed block applied to the last consensus state is valid and yields proposed state
- A blockchain protocol guarantees a *total order* on transactions
  - Original motivation in Bitcoin: Simulate centralized ledgers
  - Long believed to be a practical necessity for real-world decentralized applications
  - Enforced globally on all applications built on top