System models

Permissioned vs unpermissioned and failure models

Leander Jehl

System models

Permissioned vs unpermissioned

Unpermissioned: A system that anyone can join, usually anonymously.
Example: Bitcoin, Ethereum, PoW or PoStake blockchain

- In unpermissioned systems, typically sybils pose a problem

Permissioned: Need permission to join a network. System comprised of nodes with known identity.

Example:

- Several organisations running a system together, each running one servers.

Unpermissioned systems

Registration

How can we register in an unpermissioned system, if there is no one in charge of registration?

Can we get access to a wallet that already belongs to someone?

Unpermissioned systems

Anonymity

Are Unpermissioned systems, anonymous?

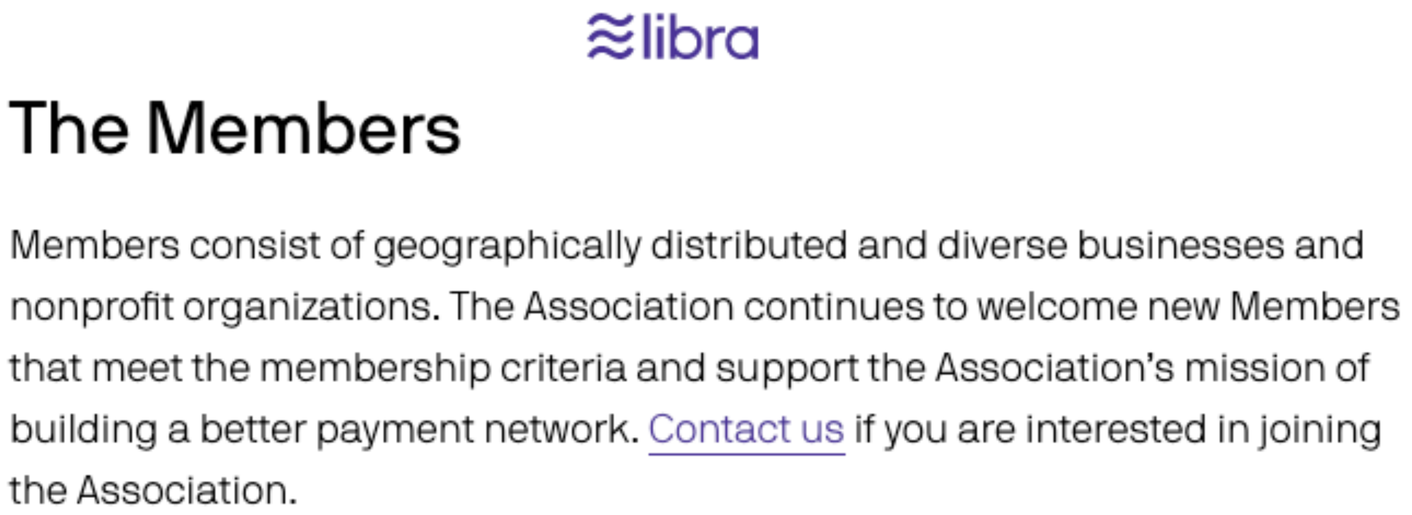
To what extent your identity is safe?

Permissioned systems

Libra

Example:

- Several organisations running a system together, each running one server



Permissioned systems

Paxos - BFT

Example:

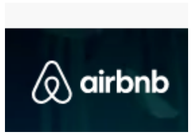
- Multiple servers, replicating one system for fault tolerance.

Permissioned systems

Identity based system

Example:

- Each peer is uniquely identified as person by passport, Bank-ID, Social-media profile



Permissioned systems

Properties

Permissioned: Need permission to join a network. System comprised of nodes with known identity.

- List of participants available
- Participants addressable, e.g. by public key and IP

Failure Models

Failure Models

Crash

Crash failure: Assumes nodes may stop responding at arbitrary points. But nodes do not behave against the protocol.

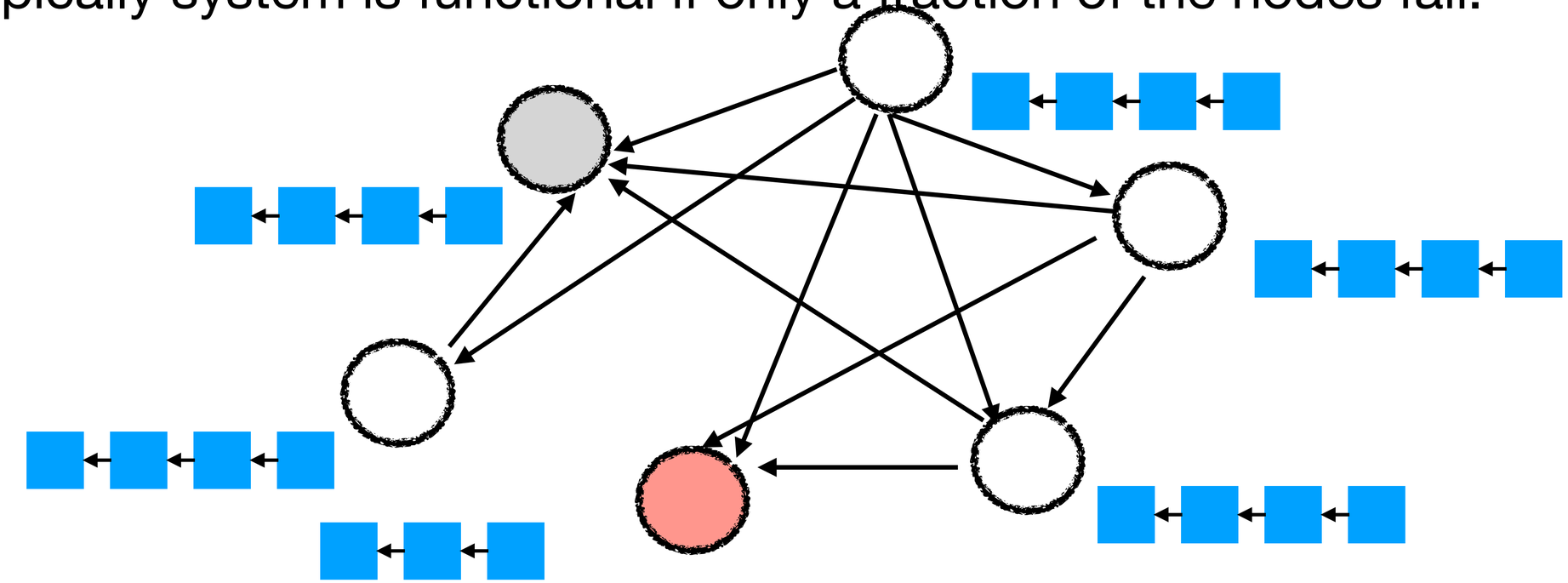
- Cannot use a single coordinator since coordinator may fail.
- Typically system is functional if only a fraction of the nodes fail.
- Example: Paxos (DAT520), less than half of the nodes may fail.
- Possible to have services that are correct (safe), but possibly not functional (live) even if all but one node fail.

Failure Models

Crash

Crash failure: Nodes stop responding

- Typically system is functional if only a fraction of the nodes fail.

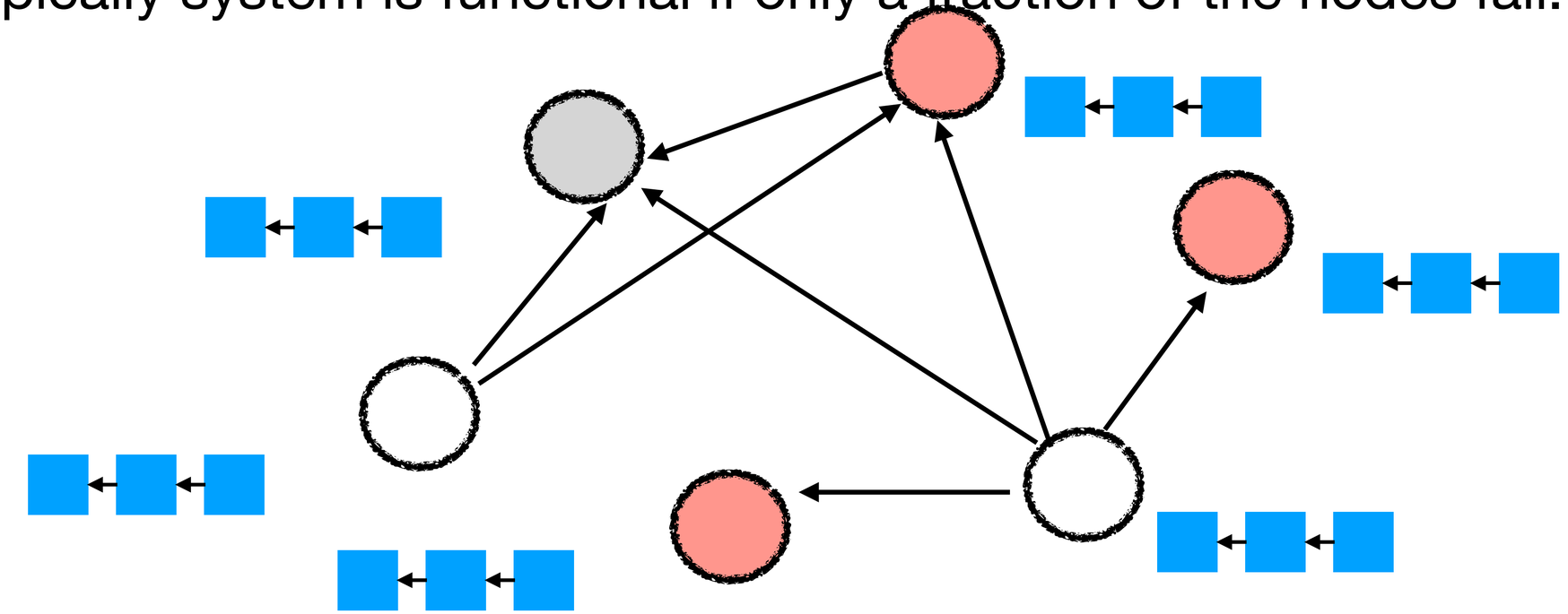


Failure Models

Crash

Crash failure: Nodes stop responding

- Typically system is functional if only a fraction of the nodes fail.



Failure Models

Byzantine fault tolerance (BFT)

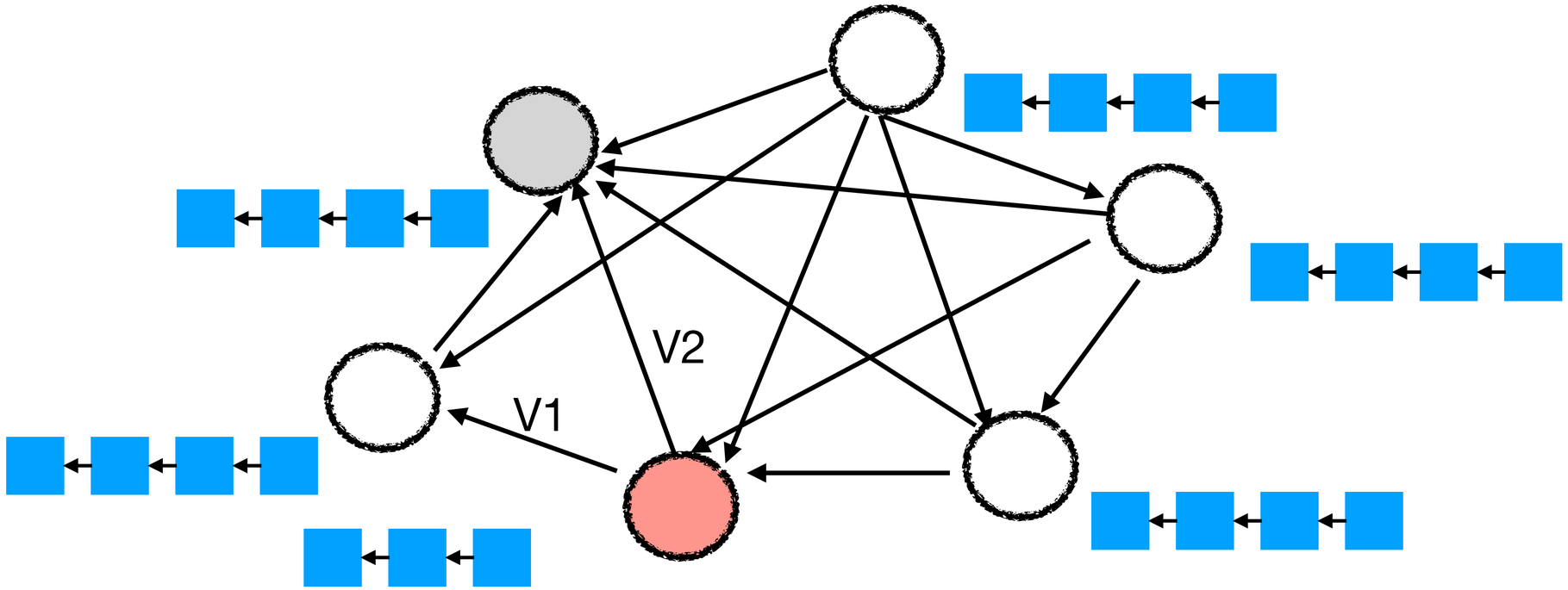
Byzantine failures: Assume a fraction of the nodes may fail arbitrarily, i.e. they may stop or even disobey/attack the protocol.

- Typically up to 1/3 or 1/2 of the nodes may fail.
- If failure threshhold is violated, bad things may happen.
- Example Bitcoin
- If attacker has 1/3 of mining power he can do selfish mining, but
  - cannot steal bitcoin.

Failure Models

Byzantine

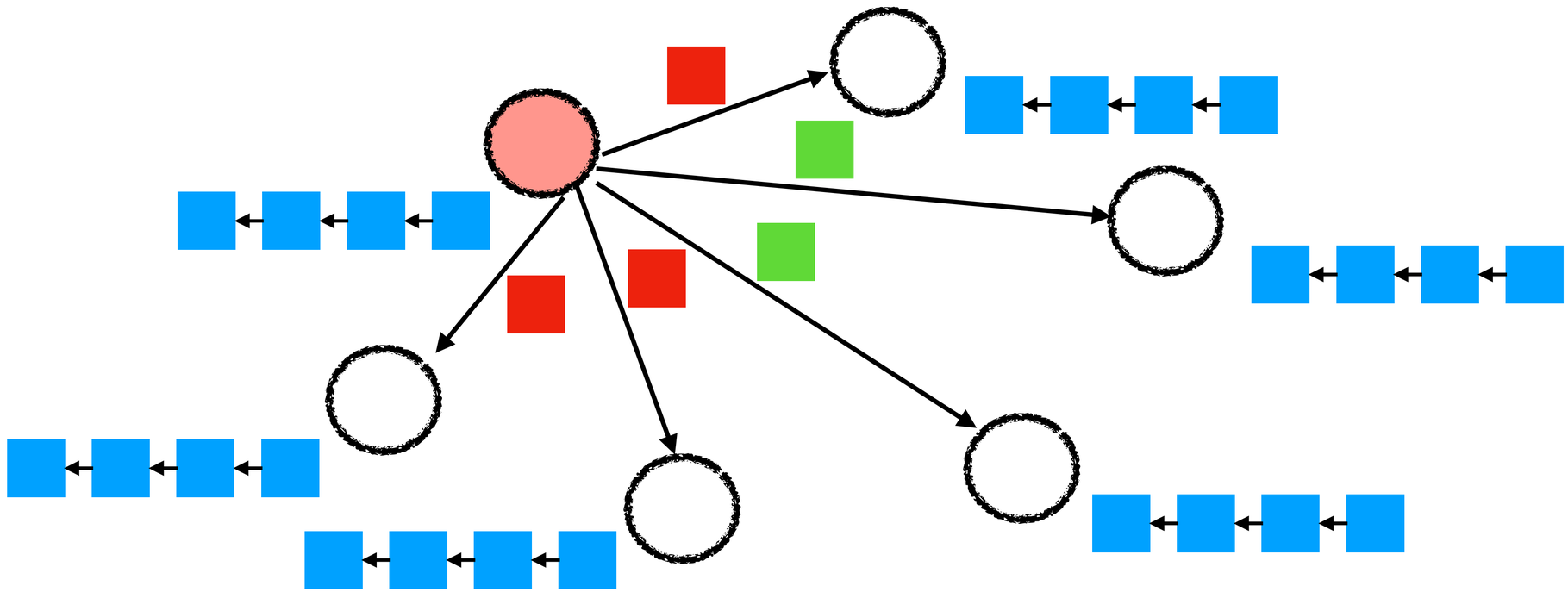
Byzantine failures: Assume a fraction of the nodes may fail arbitrarily, i.e. they may stop or even disobey/attack the protocol.



Failure Models

Byzantine

Byzantine failures: Assume a fraction of the nodes may fail arbitrarily, i.e. they may stop or even disobey/attack the protocol.



Failure Models

Byzantine

Byzantine failures: Assume a fraction of the nodes may fail arbitrarily, i.e. they may stop or even disobey/attack the protocol.

- Typically up to 1/3 or 1/2 of the nodes may fail.
- If failure threshhold is violated, bad things may happen.
- In Unpermissioned systems, instead of 1/3 of nodes, may reference, 1/3 of mining power or 1/3 of stake
- If detectable, node will be punished

Failure Models

Rational

Rational failures: Nodes has a well defined utility function and will deviate from the protocol, if it increases their utility.

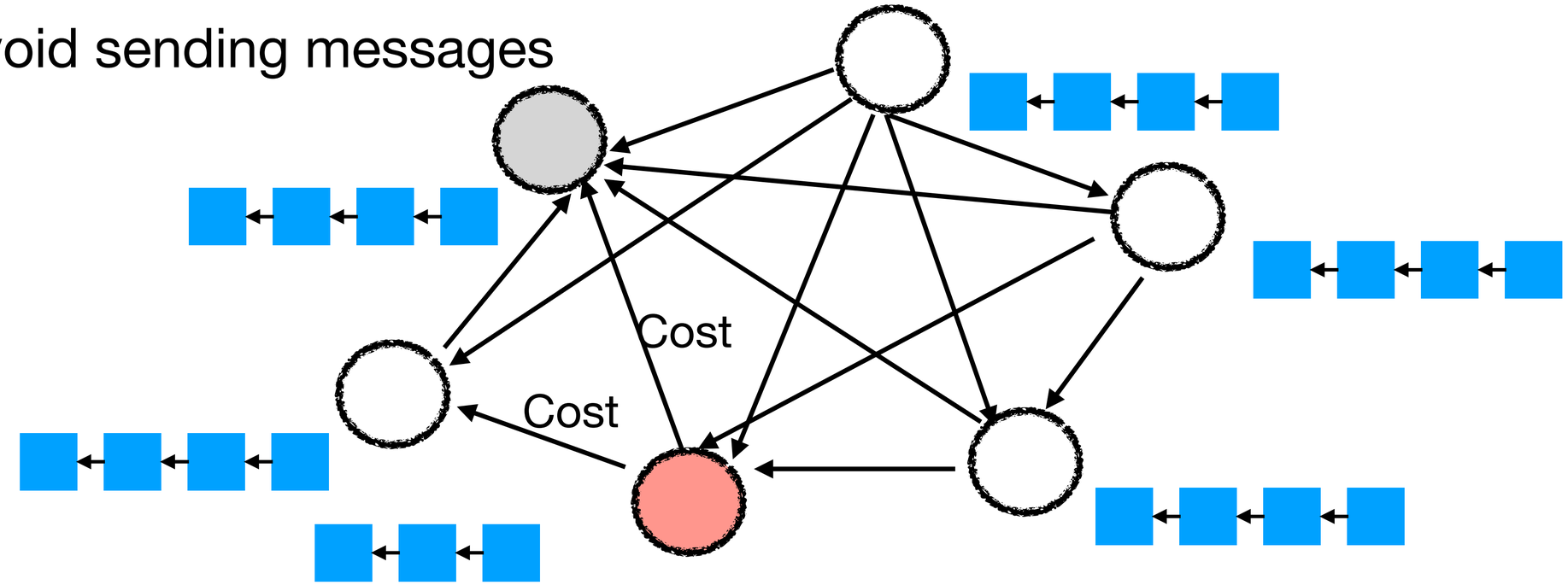
- Different from BFT here all nodes may fail at once.
- Utility functions based on
  - Reward
  - Access to functioning service
  - Cost (computation, networking, …)

Failure Models

Rational

Rational failures: Nodes has a well defined utility function and will deviate from the protocol, if it increases their utility.

- Avoid sending messages



Failure Models

Rational

Rational failures: Nodes has a well defined utility function and will deviate from the protocol, if it increases their utility.

- Different from BFT here all nodes may fail at once.
- Utility functions based on
  - Reward
  - Access to functioning service
  - Cost (computation, networking, …)
- Protocol may be game theoretic equilibrium

Nash-Equilibrium

Rational failures

Example: Prisoners dilemma

Two prisoners are interrogated separately. Both can either choose to confess or deny charges.

- *If both deny, both get one year in jail.*
- *If both confess, both get 2 years in jail.*
- *If only one confesses, he goes free and the other one gets 3 years in jail.*

Without knowing what the other one will do, a prisoner can improve his uility, by confessing.

Nash-Equilibrium

Rational failures

Example: Prisoners dilemma

	Prisoner 1 confesses	**Prisoner 1 denys**
Prisoner 2 confesses	-2\-2 Nash-Equilibrium	-3\0
Prisoner 2 denys	0\-3	-1/-1

Nash-Equilibrium

Rational failures

A **Nash-Equilibrium** is a set of strategies strategy s.t. if all other players follow this strategy, a single player cannot improve his utility by deviation.

In the prisoners dilemma, the strategy where both prisoners confess is a nash-equilibrium.

The strategy where both players deny is not an Nash-equilibrium.

Nash-Equilibrium

Example

Assume a PoW system with 3 miners, each holding equal mining power.

Each miner can choose one strategy:

1. Mine honestly or
2. Perform selfish mining

We are interested in 2 differnt utilities:

A. Utility of a miner is the total amount of blocks it gets.

B. Utility of a miner is the share of all blocks in the longest chain it gets.

Nash-Equilibrium

Example

Assume a PoW system with 3 miners, each holding equal mining power.

Each miner can choose one strategy:

1. Mine honestly or
2. Perform selfish mining

We are interested in 2 differnt utilities:

A. Utility of a miner is the total amount of blocks it gets.

B. Utility of a miner is the share of all blocks in the longest chain it gets.

With utility A, honest mining (1) is a Nash-Equilibrium.

With utility B, honest mining (1) is no Nash-Equilibrium.

Failure Models

BAR Fault tolerance

In **BAR fault tolerance**, we assume that a large fraction of nodes is rational, few nodes are honest (do not fail) and some nodes are byzantine.

In a game theoretic view, a byzantine player is one that has a unknown or different utility function.

Is it permissioned or unpermissioned?

If it assumes BFT, what is the failure threshold?

Failure Models

BAR Fault tolerance

Example: Committee-based blockchains

Assumptions

- *If a block is confirmed, reward is 10.*
- *Voting has a cost of 1.*

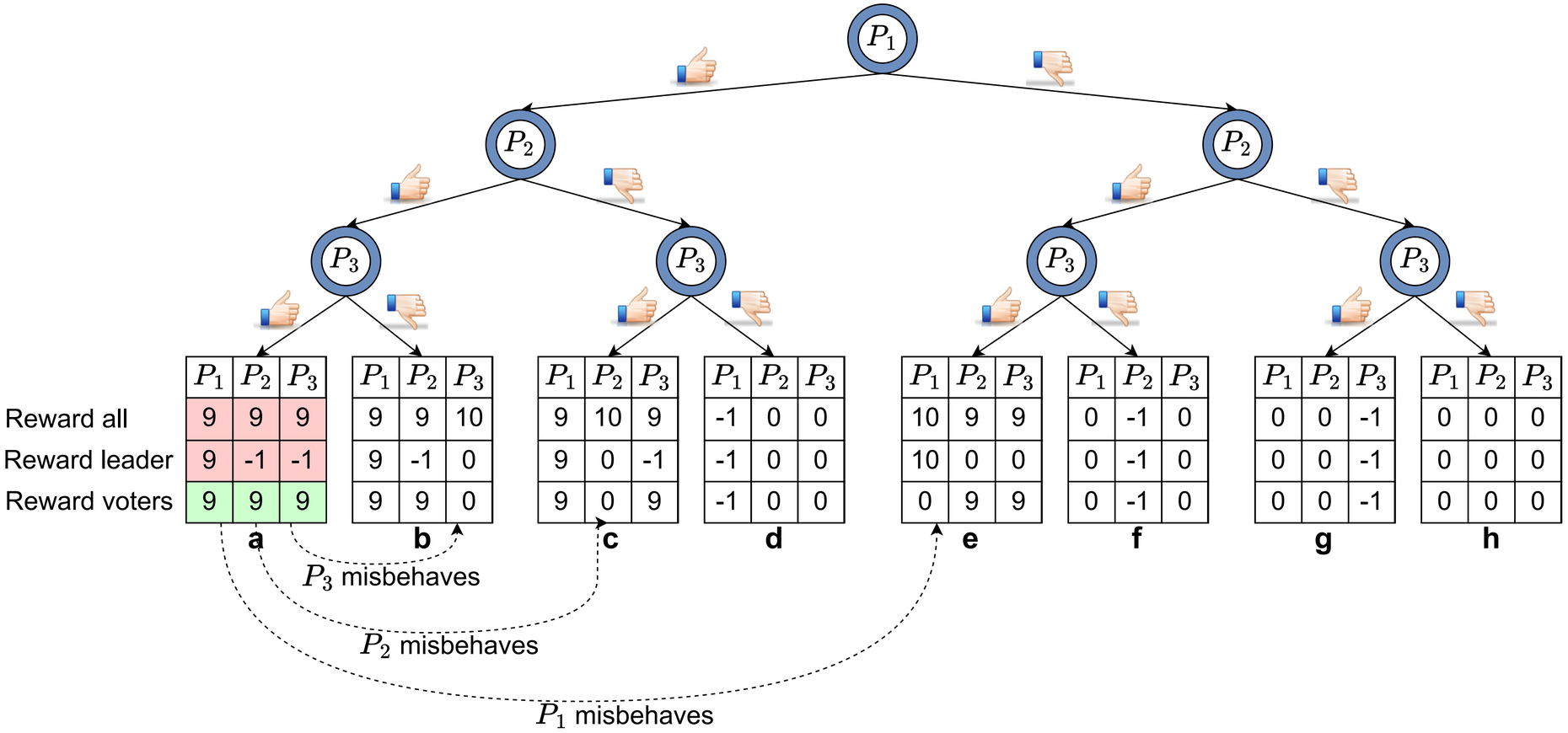
How to reward committee members?

- *Reward everyone (despite whether or not they vote).*
- *Reward only the leader (proposer).*
- *Reward Only voters.*

Failure Models

BAR Fault tolerance

Example: Committee-based blockchains



System example

Stellar

- In Stellar, not all nodes are equal. Each node defines his requirements.
- Slides: https://sosp19.rcs.uwaterloo.ca/slides/mazieres.pdf
- Video: https://sosp19.rcs.uwaterloo.ca/videos/D1-S2-P2.mp4

Repetition questions

Permissioned vs. Unpermissioned

- How can unpermissioned systems handle sybils?
- How can permissioned systems handle sybils?

Repetition questions

BFT vs Rational

- What is a typical failure threshhold in BFT systems?
- What is a typical failure threshhold in rational systems?
- What can a byzantine node do, that a rational node cannot?

Repetition questions

Nash-Equilibrium

- Developers want to include a new feature into bitcoin.
  - Miners in group A want that feature, group B does not want that feature.
  - If all miners go to the new version, group A gains 5 utility, group B gains 2.
  - If all miners stay in the old version (without feature), group B gains 4 utility, group A gains 2.
  - If group A goes to the new version and group B stays with the old, A gains 3 utility and B gains 1.
  - If group B goes to the new version and group A stays, A both get -1 utility.
- Both groups need to choose their strategy. Is there a Nash-Equilibrium in this game.

Repetition questions

Nash-Equilibrium

- Developers want to include a new feature into bitcoin.

	Group A goes to new feature	**Group A stays on old version**
Group B goes to new feature	2\5 Nash-Equilibrium	-1\-1
Group B stays on old version	1\3	4\2