Sessions 08
Blockchain Protocol (2)

Blockchain Protocols and Decentralized Applications

Goals

- **Consistency**
  - History agreement
  - What about the “lazy nodes”?
  - Is Consistency enough?
- **Liveness**
  - Every transaction submitted to at least one node is eventually added to every node’s history
- Issues
  - Communications
  - Fault tolerance

Assumptions

- Permissioned setting
  - Number of nodes are known
  - Each node is identifiable (IP addr, number, name, etc.)
  - Before Bitcoin, only permissioned
- PKI (*public key infrastructure*)
  - pki/ski pair; pki known to all nodes upfront
- Synchronous Model
  - Global clock
  - Round 1,2,3,4…
  - Msg sent at round **t**, will be received at **t+1**
- All honest nodes
  - Way too strong

Solution

- Round-Robin Leaders
  - Nodes take turns as “leaders”
  - Leader sends ordered list of txs (“block”)
  - Node receives ordered list, append to local history
  - **Consistency + Liveness**

Byzantine Generals' Problem

- Lamport et al.: *Imagine that several divisions of the Byzantine army are camped outside an enemy city, each division commanded by its own general. The generals can communicate with one another only by messenger. After observing the enemy, they must decide upon a common plan of action. However, some of the generals may be traitors, trying to prevent the loyal generals from reaching agreement.*
- **n** generals
- **1** commanding general
- Goal:
  - All loyal generals reach same decision (ATTACK / RETREAT)
  - If loyal commanding general, all loyal generals will obey

Faulty/Byzantine Nodes

- Faulty/Byzantine - not “honest”
  - **Crash fault** - run honestly until some failure
  - **Omission fault** - selectively withhold msgs it’s supposed to send
    - Consistency?
  - **Byzantine fault** - arbitrarily behaviour
    - Can’t break cryptography
    - Send conflicting msgs to different nodes
- New assumption
  - Most ***f*** faulty nodes, ***n-f*** honest nodes
  - Saboteur

Byzantine Broadcast Problem

- Round Robin Leaders
- Sender - known at the start of the round
- Goal:
  - **Agreement** - all honest nodes choose the same value
  - **Validity** - if honest sender, its private msg gets to honest nodes
  - **Termination**

SMR Reduces to BB

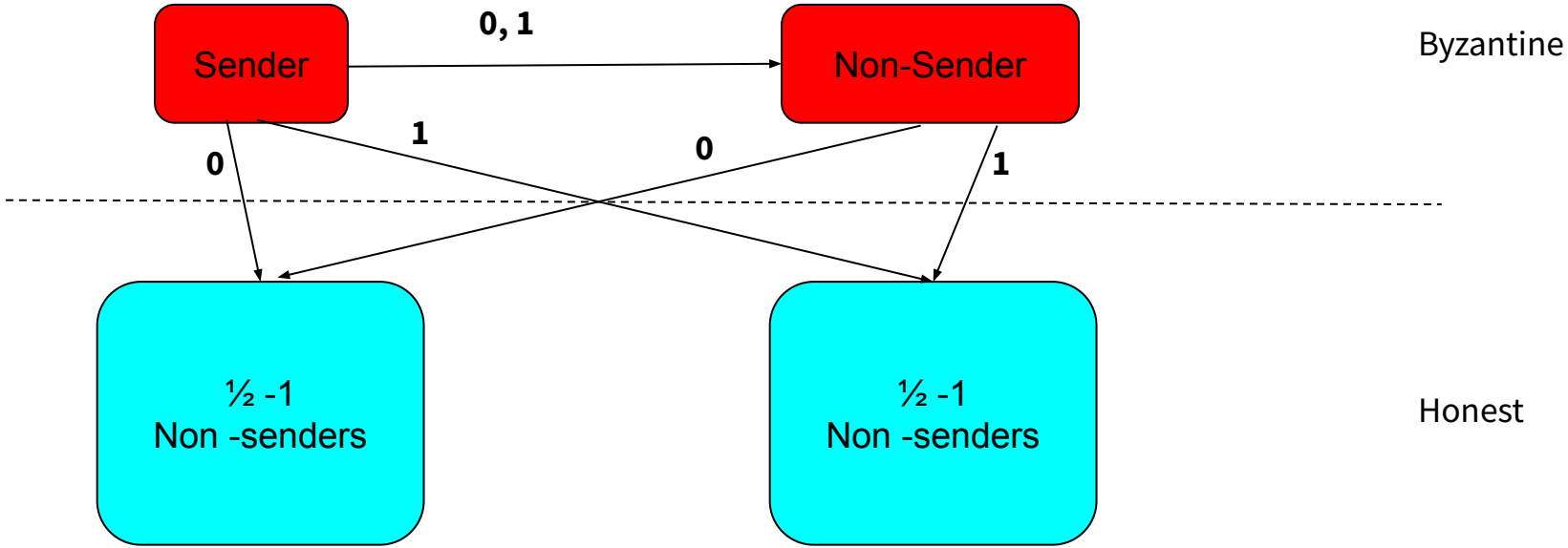
- Round Robin Leaders
- Run BB protocol (sender = leader)
  - Agree on a tx list **L**
- Each node appends **L** to its local history
- BB **agreement** => SMR **consistency**
- BB **validity** => SMR **liveness**
- Solution to BB => Solution to SMR

Simple Protocol for BB

- $f = 1$
- Proposed Protocol:
  - $t = 0$: senders send its private input (signed)
  - $t = 1$: nodes echo msg from sender to all other nodes (signed)
  - $t = 2$: each node outputs the majority vote
- $n = 4, f = 1$
  - Validity
  - Agreement
- $n = 4, f = 2$
  - Validity
  - Agreement?
- One round of cross-checking not enough

Simple Protocol for BB

- *n = 4, f = 2*
  - Validity
  - Agreement?



Dolev-Strong Protocol

- 1983
- Depends on synchronous model
- Parameter ***f*** is known
- Nodes should be "*convinced*" of a value
- Cross-checking messages for ***f+1*** rounds
- If nodes convinced of more than one value, than output canonical value

Dolev-Strong Protocol

- How Big Can ***f*** Be?
- Protocol scales linearly with **f**
- In SMR context ***f < n/2***

Assumptions - Practical Protocols

- Permissioned setting
  - Number of nodes are known
  - Each node is identifiable (IP addr, number, name, etc.)
  - Before Bitcoin, only permissioned
- PKI (*public key infrastructure*)
  - pki/ski pair; pki known to all nodes upfront
- Synchronous Model
  - Global clock
  - Round 1,2,3,4…
  - Msg sent at round **t**, will be received at **t+1**
- All honest nodes
  - Way too strong

Relaxing other assumptions

http://elaineshi.com/docs/blockchain-book.pdf

Asynchronous model

- http://elaineshi.com/docs/blockchain-book.pdf
- Chapter 12 & 13
- Good news
  - weak assumptions: any positive results are impressive
- Bad news: no positive results possible

Partially synchronous model

Partially synchronous model

- Normal condition: synchronous model
- “Attack” condition:
  - Safety/Consistency **or** Liveness
- Assumption:
  - Shared global clock
  - Known bound (delta) on max message delay in normal condition
- Goals:
  - Liveness
  - Safety/Consistency

Partially synchronous model

- Goals:
  - sanity check - achieve everything you want (e.g., safety and liveness) during “normal operation conditions”;
  - stress test - don’t break too badly when under attack (e.g., give up only safety, or only liveness);
  - recovery -recover quickly after an attack once the network returns to normal operating conditions.
- **33% Impossibility Result**
  - http://elaineshi.com/docs/blockchain-book.pdf
  - Chapter 4, 5