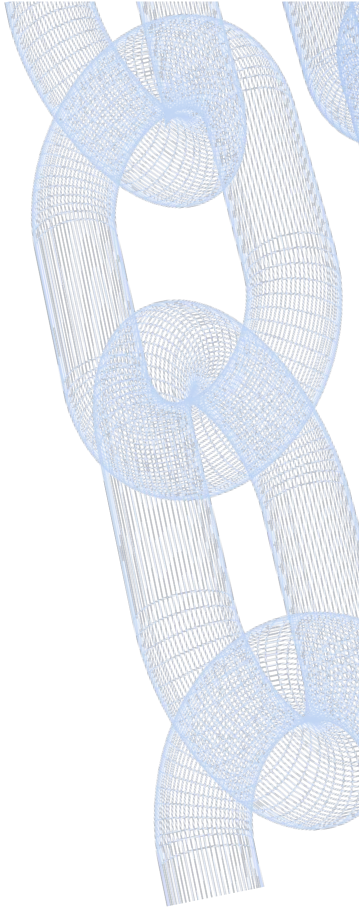
Distributed consensus and blockchains

Elaine Shi

Textbook: www.distributedconsensus.net



What is distributed consensus?

A class of methods/algorithms for a system of distributed nodes to reach agreement

- Consistency (a.k.a. safety)
- Liveness

Why is distributed consensus challenging?

Players (also called nodes) can be faulty:

- **Byzantine fault**: faulty nodes can behave arbitrarily
- **Crash fault**: faulty nodes stop responding

Why is distributed consensus challenging?

Players (also called nodes) can be faulty

Wanted: **honest players satisfy safety and liveness properties**

Terminology

distributed system

Crypto

player = node

faulty = malicious = corrupt

(by default, we consider Byzantine faults)

Applications of distributed consensus

Bitcoin has **10,000** full nodes today, and Ethereum has **8,000** full nodes

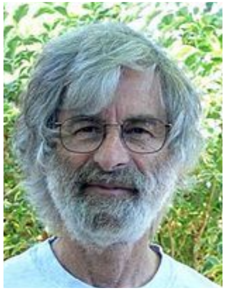
Distributed consensus is a 30-year old problem

1970s:

NASA, robust aircraft control system

Software Implemented Fault Tolerance (SIFT) project

3 computers, assume **1** might be faulty



Recipient of 2013 Turing Award
One of the founders of distributed consensus

Applications of distributed consensus

Apache Zookeeper

YAHOO!

facebook

reddit

NetApp

Solr

UBER

ebay

rackspace

This Lecture

Byzantine broadcast

Single-shot consensus

Theoretical underpinning

This Lecture

Byzantine broadcast

- Single-shot consensus
- Theoretical underpinning

Blockchains (a.k.a. State machine replication)

- Repeated consensus over time
- Linearly ordered log
- Often needed in practical applications

Bitcoin, incentives

Fall 2022: 15435 Foundations of Blockchains

Byzantine Broadcast

(i.e., single-shot consensus)

15451/15651 final exam

Virtual or Physical?





Communication model:

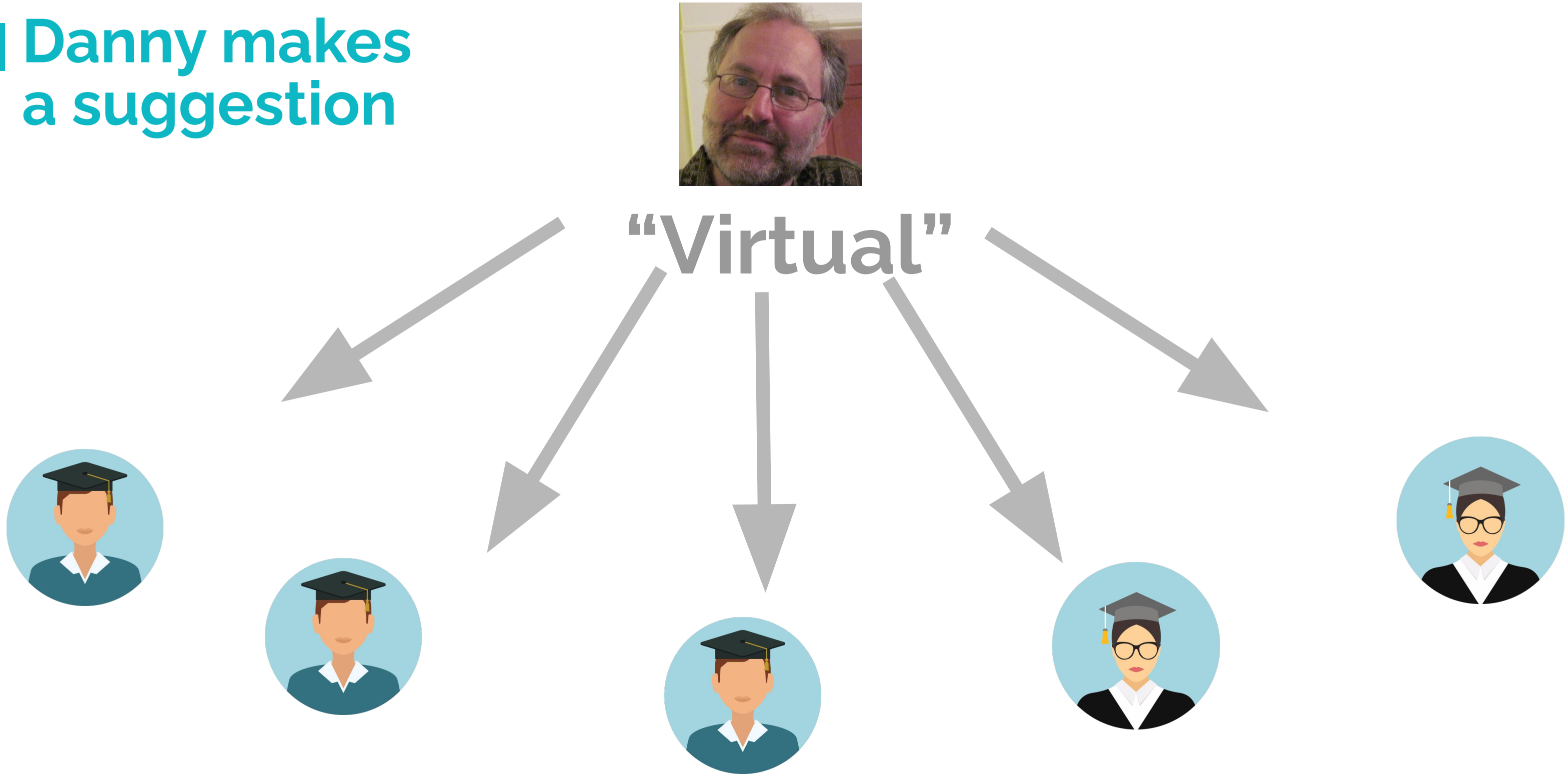
- exchange pairwise emails
- emails sent today delivered next morning
- emails authenticated with signatures

Synchronous model
"round" = day

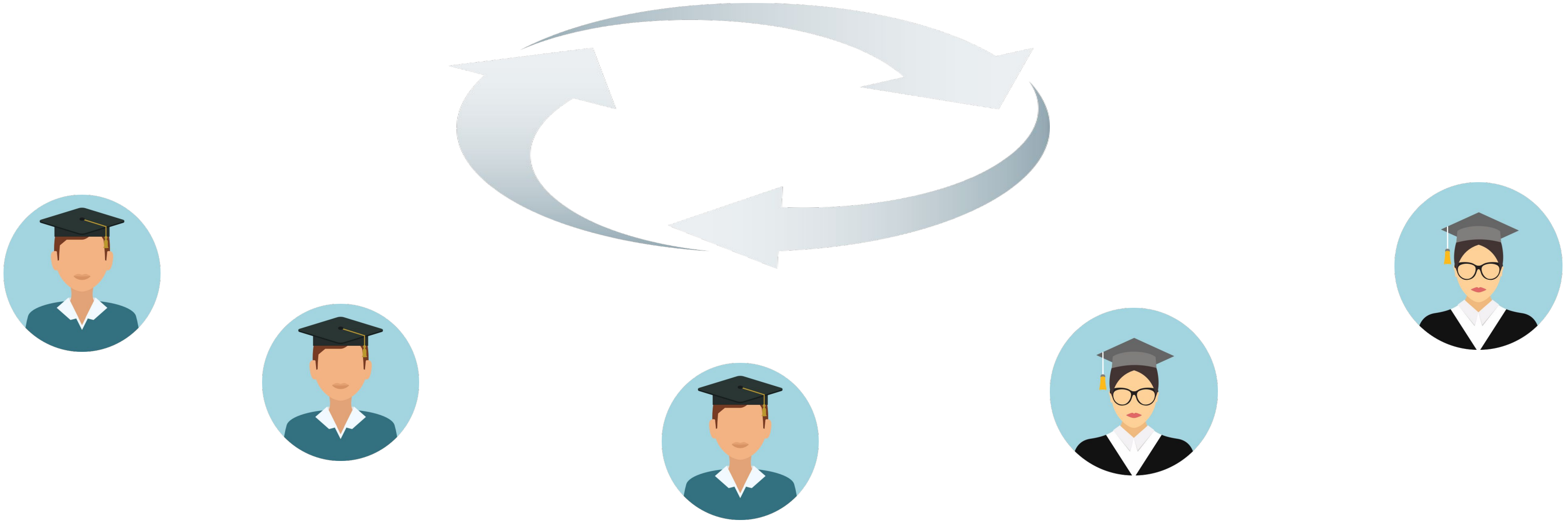




Danny makes a suggestion



Everyone discusses



Everyone decides

Virtual

Virtual

Virtual

Virtual

Virtual

Virtual

Everyone decides

Virtual

Some are unhappy

(e.g., don't want a final exam)

Virtual

Virtual

Virtual

Virtual

Virtual

Everyone decides

Virtual

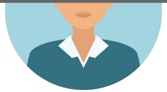
Consistency

happy players agree on decision

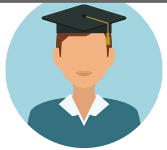
Validity

if Danny happy, agree on D's proposal

Virtual



Virtual



Virtual

Virtual

Virtual

Byzantine Broadcast

Consistency

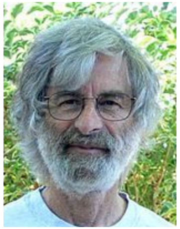
happy players agree on decision

Validity

if Danny happy, agree on D's proposal

Both properties are needed for the problem to be non-trivial

Byzantine Broadcast: Lamport's Formulation



Byzantine empire

n generals, among whom 1 is **commander**

Want to agree on: **Attack** or **retreat**?

Some generals (including commander) may be traitors

also called the "**Byzantine Generals**" problem

Byzantine Broadcast in more general terms

n players, among whom 1 is the **designated sender**

Want to agree on 1 bit: either 0 or 1

Some (including sender) may be corrupt

Want to achieve: consistency + validity

Consistency

If two honest players output b and b' respectively, then b = b'

Validity

if sender honest, every honest player outputs sender's input bit

How do we design a Byzantine Broadcast protocol?

c.f. traditional signatures

More about digital signatures

- Signer uses a **private key** to sign, verifier uses a **public key** to verify
- Computationally infeasible to forge without the private signing key
- A signed message can be forwarded

RSA assumption:

Given: N with unknown factoring, y, e hard to compute x such that $x^e = y \mod N$

$N = p \cdot q$ p, q prime

RSA signatures:

Public key: N, e

Private key: d s.t. $(x^d)^e = x \mod N$ for any x

Sign: $\sigma = \text{Hash}(m)^d \mod N$

Vf: $\sigma^e =?= \text{Hash}(m) \mod N$

Strawman idea 1: Listen to the Sender

R0: Sender signs and sends a bit to everyone

R1: Everyone outputs what it hears from the sender

Assume: messages with invalid sigs discarded

Strawman idea 1: Listen to the Sender

R0: Sender signs and sends a bit to everyone

R1: Everyone outputs what it hears from the sender

This is called a 1-round protocol

Assume: messages with invalid sigs discarded

Strawman idea 1: Listen to the Sender

R0: Sender signs and sends a bit to everyone

R1: Everyone outputs what it hears from the sender

Assume: messages with invalid sigs discarded

Strawman idea 2: Wait for All Votes

R0: Sender signs and sends a bit to everyone

R1: Everyone votes for what it hears from the sender, vote is sent to everyone. If the sender sent 0 or 2 bits, then vote for 0.

R2: Everyone outputs the bit that has collected all players' votes. If no bit has collected all players' votes, output 0.

Assume: messages with invalid sigs discarded

Strawman idea 3: Majority Vote

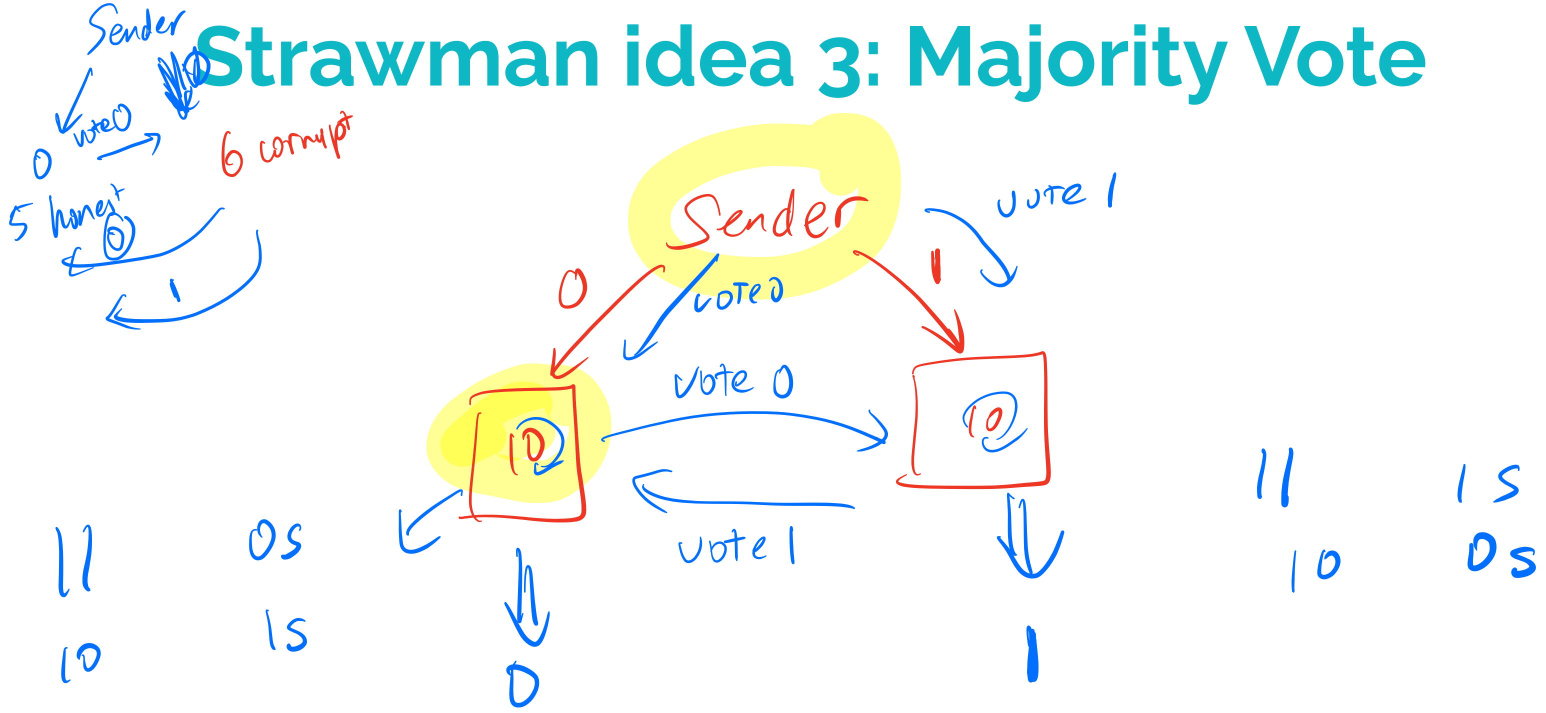
R0: Sender signs and sends a bit to everyone

R1: Everyone votes for what it hears from the sender, vote is sent to everyone. If the sender sent 0 or 2 bits, then vote for 0.

R2: Everyone outputs the bit that has collected more votes

Assume: messages with invalid sigs discarded

Strawman idea 3: Majority Vote



Assume: sender is player 1

[Dolev-Strong 83]

The Dolev-Strong protocol

- **Round** 0: Sender sends $\langle b \rangle_1$ to every node.
- **For each round** $r = 1$ **to** $f + 1$:

  For every message $\langle \tilde{b} \rangle_{1,j_1,j_2,\ldots,j_{r-1}}$ node i receives with r signatures from distinct nodes including the sender:

  – If $\tilde{b} \notin \mathsf{extr}_i$: add $\tilde{b}$ to extr_i and send $\langle \tilde{b} \rangle_{1,j_1,\ldots,j_{r-1},i}$ to everyone — note that here node i added its own signature to the set of r signatures it received.

- **At the end of round** $f + 1$: If $|\mathsf{extr}_i| = 1$: node i outputs the bit in extr_i; else node i outputs 0.

local set maintained by player i initially empty

f: number of faulty players

$\langle b \rangle_{i,j}$: a bit b and sigs from i and j

Dolev-Strong: if all are honest, what happens during the execution?

Dolev-Strong: if all are honest, what happens during the execution?

- Sender signs and sends a bit b in R0, and everyone adds b to their extracted sets and votes on b in R1.

- At the end, everyone outputs b.

Dolev-Strong: validity is easy to show

Claim 1: for r <= f, if b in some honest node's extracted set by the end of round r, then b in every honest node's extracted set by the end of round r+1

If I know it now you'll know it in next round.

proof: suppose $\tilde{b}$ is in honest node i's extracted set by the end of round r. it must be that i added $\tilde{b}$ to $extr_i$ in some round $r' \le r \le f$. in the round r', i sends $\tilde{b}$ along with r'+1 sigs on it. in round r'+1, every honest player with receive ↓, will add $\tilde{b}$ to their extracted set if they haven't done so.

Claim 2: if some honest node has bit b in its extracted set by the end of round f+1, then every honest node has b in its extracted set by the end of round f+1

In round f+1 : If I know it now you know it now

magic

proof: some honest player i has b in its extracted set by end of round f+1

① i added b to its extracted set in r ≤ f the claim holds due to Claim 1

② i added b to its extracted set in round f+1 it must be that i receives ~~b~~ b along with f+1 sigs from distinct senders in round f+1

one of these sigs comes from an honest player

an honest player signed b in $r \leq f$

$\Rightarrow$ claim 1

(1) Find an attack if protocol runs for only f round

Dolev-Strong: why is f rounds not enough?

Lower bound: f+1 rounds necessary for any deterministic consensus protocol

(initially proved also by Dolev and Strong)

This lower bound can be circumvented through the use of randomness

Muddy Children Problem

n children playing in the playground, and k ≤ n of them have mud on their forehead.

Teacher gathers children, declares, “one or more of you have mud on your forehead”.

Everyone can see if others have mud on their forehead, but cannot tell for themselves

The teacher says, “at this moment, if you know you have mud on your forehead, pls step forward”. The teacher waits for a min, no one steps forward. The teacher says again, “2nd call: at this moment, if you know that you have mud on your forehead, please step forward.”. This goes on until some children step forward.

Q: in which round will some children step forward? Note that the children do not communicate with each other. They know that at least one of them has mud on their forehead, and they know the current round number

Round 1: if k = 1, then the muddy kid see no one else with mud, and will know she's muddy and step forward

Round 2: if k = 2, then the two muddy kids each see one other muddy kid. They know that k > 1 because no one stepped forward in round 1. So they now step forward

This goes on.

What we learned

- Consensus is possible in a **synchronous** network
- Assume public-key infrastructure (PKI) and digital signatures, we can secure against any number of Byzantine corruptions!
- The Dolev-Strong protocol isn't quite so efficient, and typically it's not used in practical implementation.