BYZANTINE

FAULT

TOLERANCE

ANNOUNCEMENTS

* No office hours on 11/15, 11/22, 11/24

  → I am not in New York

  Thanksgiving

* Will send out notes on project proposals by Sunday. Assume that they are fine.

What are Byzantine Failures?

- Malicious messages sent by a process
- Behave arbitrarily

What causes Byzantine Failures

* Designer: Malicious users in large applications

* Bugs When Implementing Safety Critical Systems

Brief Excursion Into Cryptography

- Digital Signatures

$y = Sig_{key}(message)$

- key : Secret known to only one node
- Given y anyone can check
  - ↳ Generated from message
  - → Generated by someon who has key.
- Usually implemented using public key cryptography

$\exists\ a, b$ s.t.

$$dec_b(enc_a(M)) = M\ \forall M.$$

↳ Known to all nodes

key from above

Assume:
- Given b finding a is computationally infeasible
- Given M & b computing

$enc_a(M)$ is computationally infeasible

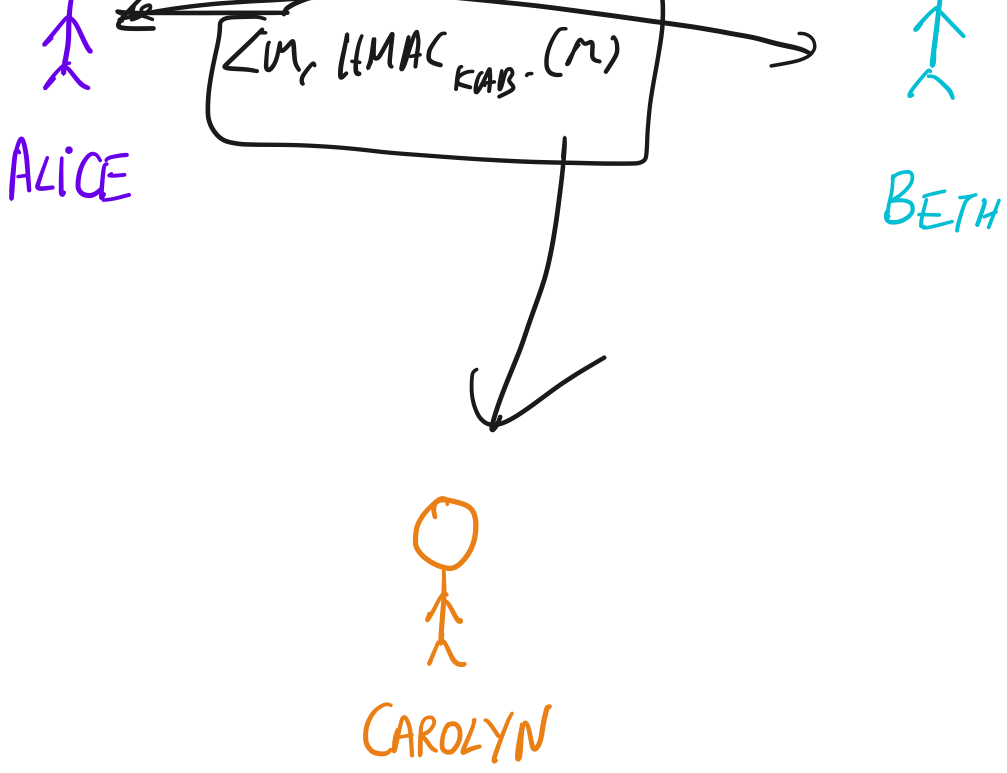
$\langle hash(M), enc_a(M) \rangle$ acts as a signature.

- Hash-based Message Authentication Key

  $HMAC_{KEY}(M)$

  - KEY known to a set of nodes
  - Cannot compute $HMAC_{KEY}(M)$ without KEY.
  - GENERALLY FASTER TO COMPUTE THAN DIGITAL SIGNATURES

How USED?



BYZANTINE FAILURES, CRYPTOGRAPHY, BUGS...

What does this have to do with last class?

- Want to protect against Byzantine failures
- Thus far: State machine replication for surviving fail-stop failures
- Today: RSM for Byzantine?

Remember: RSMs Depend on Consensus.

Start with Consensus in Byzantine Environments

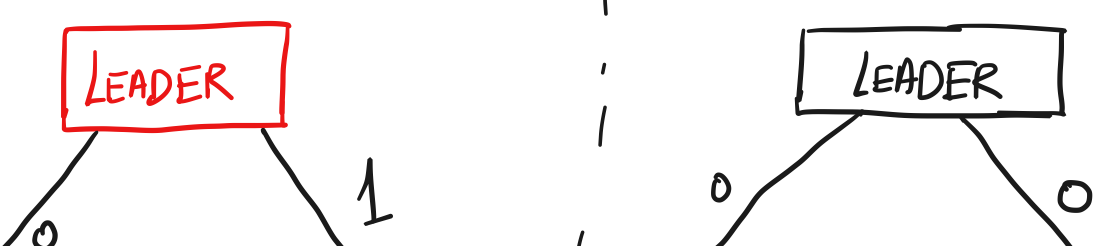
- Agreement: All correct nodes agree on the selected value (IC1)
- Validity: If the leader (??) is correct then correct nodes accept values proposed by the leader
- Liveness: Byzantine nodes cannot prevent progress

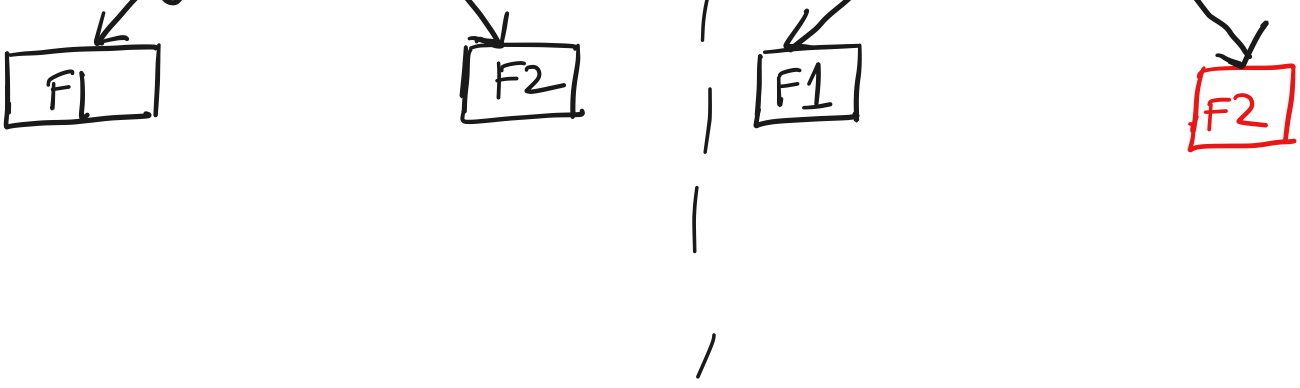
Lamport's impossibility

- Without authenticated channels need $3f+1$ replicas to survive f failures.
- Authenticated Channel:

- Q: Asynchronous / Partially synchronous / Synchronous

Lamport's impossibility





OM: Example of a BFT Consensus protocol

Q1: Assumed processing model?

* Unauthenticated

1 0

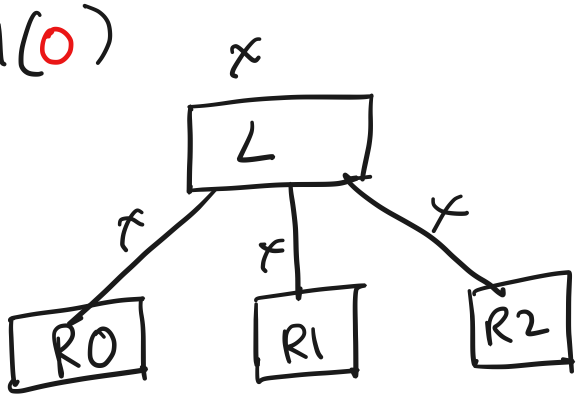
Requirement:

OM(f) — Up to f faulty nodes, 3f+1 total nodes.

- At each correct process OM(f) produces vector $(v_0, \ldots v_n)$ where $n = ?$ such that any two correct processes i, j agree on a majority of values.

How?

OM(0)

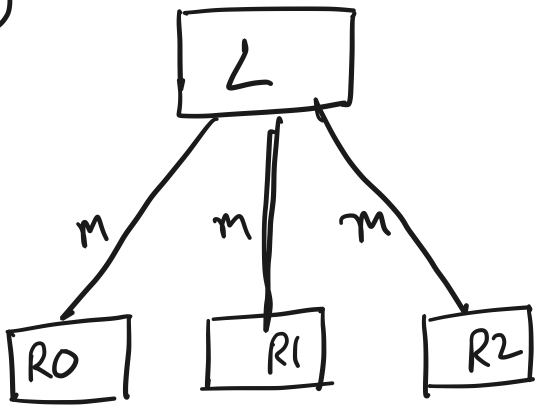


$f = 0 \Rightarrow$ No faulty node

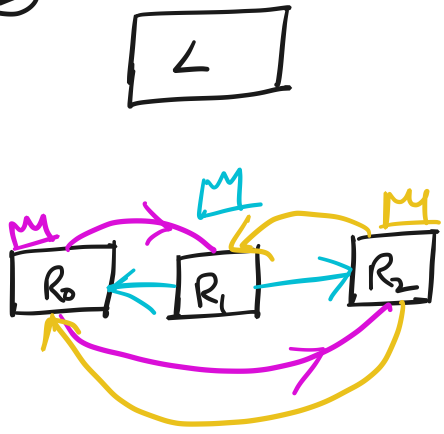
↳ Everyone can trust all messages.

OM(1)

①



②



R_0: V_L, V_{R_1}, V_{R_2}

R_1: V_L, V_{R_0}, V_{R_2}

R_2: V_L, V_{R_0}, V_{R_1}

} At most one of L, R_0, R_1, R_2 faulty

Case split:

* L faulty

R_0: x y z

R_1: y x z

R_2: z x y

* R_0 (or R_1 or R_2) faulty

R_0 send y to R_1; z to R_2

R_1 x y x

R_2 x z x

* So we know:

OM(0) works [Trivially]

OM(1) works by demo above

Should show OM(f) works.

How? Induction: Assume OM(f-1) works

[satisfies agreement/IC1 ∧ validity/IC2]

Interactive agreement, IC1 & Validity IC2

First OM(f)

1. Leader sends value to all followers.

$OM_i(f-1)$ [2. Follower i invokes OM(f-1) where i Is LEADER & there are n-2 followers

3. Each follower i collects values from n-2 OM_i : $(v_1, v_1, \ldots v_{n-2})$ and takes majority.

Validity : Requires correct Leader.

Lemma 1 If at most f failures, leader is correct & $n > 2f + m$ then OM(m) meets validity.

Proof . OM(0) Trivially true.

Channels are reliable.

. Assume true for OM(m-1)& leader sends value v.

.

$\Rightarrow$ At least

$n-1-f > f+(m-1)$

OM(m-1) are run

copies of OM(m-1) ...

with (n-1) processes to send v.

- But $(n-1) > 2f + (m-1)$

  ⇒ By inductive hypothesis each process gets at least $f + (m-1)$ copies of v

- But $f + (m-1)$ is a majority

  ⇒ choose v

Lemma 1 with $m = f$ ⇒ Validity.

Agreement:

- If leader is correct, this follows from Lemma 1
- Only need to consider faulty leader

  Leader is faulty ⇒ at most $f-1$ faulty followers

- Correct followers run $OM(f-1)$ with $n-1$ participants of which $f-1$ are faulty.

  We know $n > 3f$

  $\Rightarrow n-1 > 3(f-1)$

  $\Rightarrow$ At least $(n-1)-(f-1)$ correct versions of $OM(f-1)$ are run & reach agreement.

  $(n-1)-(f-1)$ is a majority of values $\Rightarrow OM(f)$ reaches agreement

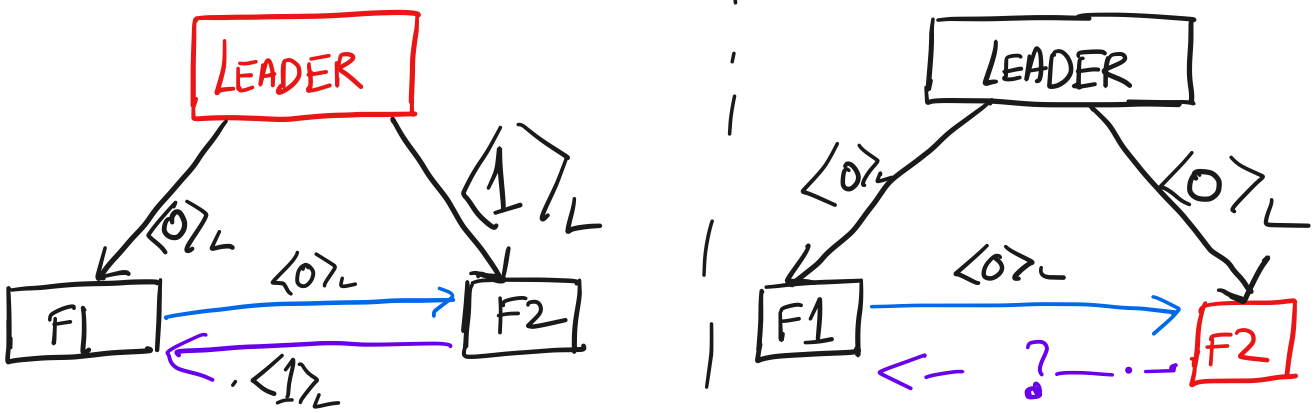
---

Bottom Line

Agreement & Validity without authentication requires $3f+1$ participants

Liveness! !

Why Authentication Helps?



PBFT

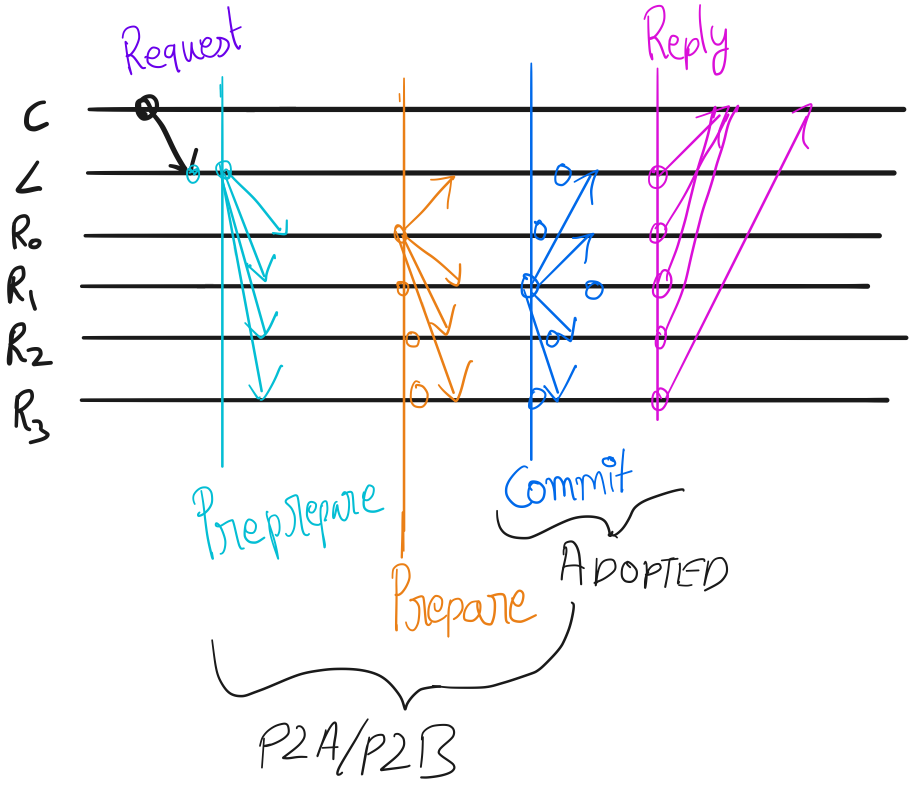
- Uses authentication (Digital Signatures OR HMACs)
- Still requires $3f+1$ processes.
- Why?

- Why we need at least $2f+1$?

  - $2f+1$ — $f+1$

- Why we need $3f+1$?

Protocol: "Paxos" Like



Differences

- Every process counts decisions
- Signatures/HMAC to validate messages
- Every one responds to the client

- Message Authentication

  * Client requests are signed
    ↳ Malicious leader cannot impersonate clients
  * Pre-Prepare carries command but command & pre-prepare message signed separately

$$[\langle \text{PREPREPARE}, \text{VIEW}, \text{SLOT}, \text{HASH}(c) \rangle_L , c]$$

Why?

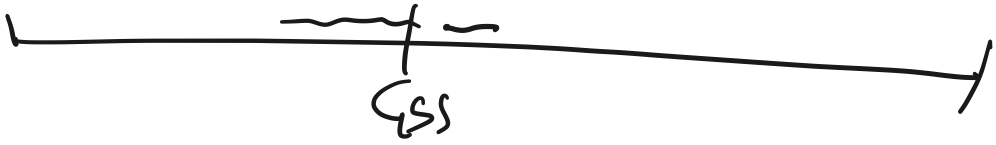
* CLIENT MUST WAIT FOR $f+1$ RESPONSES

Liveness?

Remember: Faulty nodes should not impede the algorithm's progress.

But, FLP => No termination.

What is going on here?



Achieving Liveness

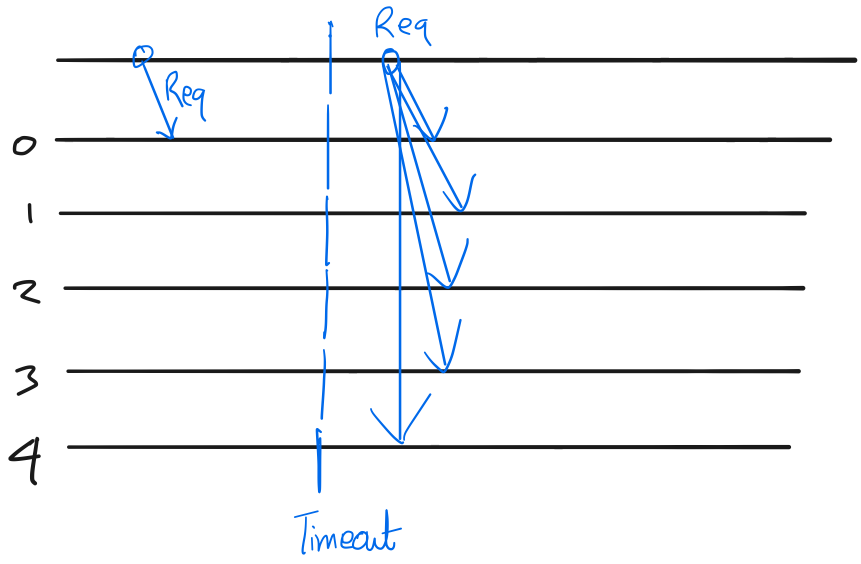
Two problems

* Faulty leader might not send pre-prepare

* Faulty follower launches view change frequently.

Faulty Leaders

* Clients can timeout & broadcast



Signals that leader might be faulty

Q: Failure assumption about clients?

* Need to make sure non-faulty leaders can be elected.

Raft & MultiPaxos:

PBFT: View number dictate leader

No more than f view changes before correct leader

Faulty Followers

* Require $2f+1$ view change messages before changing
  ↳ Faulty followers cannot unilaterally trigger changes

PBFT in practice