Lecture 6:
The Partially Synchronous Model and the CAP Theorem

(Lecture Series on Foundations of Blockchains)

The Story So Far

Synchronous model: (lectures 2+3)

- shared global clock, a priori known bound Δ on maximum message delay
  - good news: strong positive results (Dolev-Strong => BB + SMR, no matter what f is)
  - bad news: assumptions too strong (ignores outages + attacks)

Asynchronous model: (lectures 4+5)

- no global clock, no assumptions on message delivery (other than eventual delivery)
  - good news: weak assumptions => any positive results automatically impressive + useful
  - bad news: FLP => no positive results possible! (even if $f = 1$)

Idea: outages/attaks end eventually, right?

Dwork / Lynch / Stockmeyer 1988

The Partially Synchronous Model

Idea: "normal conditions" = synchronous, "attack" = asynchronous
(once attack stops, want protocol to quickly resume normal operation)

Assumptions:
- shared global clock (ok to relax to bounded drift)
- known bound Δ on max message delay in normal conditions
- unknown transition time GST ("global stabilization time") from asynchronous to synchronous

Promises on Message Delivery

① sent at time $t \leq GST \Rightarrow$ arrives by time $GST + \Delta$

② sent at time $t \geq GST \Rightarrow$ arrives by time $t + \Delta$

[protocol = specify msgs to send as function of node's private input, received msgs & current time step]

Roughly equivalent:

like synchronous model, but Δ unknown a priori.

Goals for a Consensus Protocol

Note: FLP impossibility does not immediately apply (except in asynchronous phase).

Traditional goals:

1. not long after GST, safety + liveness both hold.
2. safety holds always (even in asynchronous phase).

← by FLP, must give up something

[longest chain protocols instead favor liveness over safety]

Big result: 1 + 2 achievable if and only if $f < n/3$. [i.e., $n \geq 3f+1$]

- "only if" - see next video (impossibility result)
- "if" - see Lecture 7 (Tendermint protocol)

Intuition for Impossibility ($f \geq n/3$)

Fact: impossibility result will hold even under PKI assumption (see next video).
=> must be driven by threat of unbounded delays, not by simulation of honest nodes by Byzantine nodes (as in Lecture 3's hexagon proof).

Intuition:

(1) can only wait to hear from $n-f$ nodes before taking action
[by termination, + fact that Byzantine nodes may never respond, even after GST]
[issue: the f outstanding nodes might be delayed (if pre-GST) rather than Byzantine, for the $n-f$ nodes you heard from could still be Byzantine]

(2) to avoid getting tricked, need >50% of these to be honest (else, whom to believe?)
- i.e., $f < \frac{1}{2}(n-f)$, or equivalently, $f < n/3$

Proof of Impossibility ($f \geq n/3$)

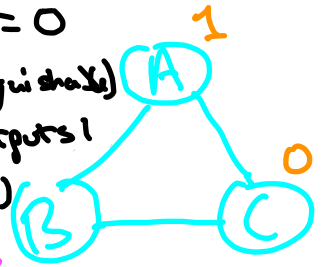
set GST to this

Theorem: in the partially synchronous model with $f \geq n/3$, there is no protocol for Byzantine agreement satisfying agreement, validity, and (eventual, post-GST) termination.

$\max\{T_1, T_2\} + 1$

Proof: [for the $n=3$, $f=1$ case] [exercise: extend to all f, n with $f \geq \frac{n}{3}$]

- adversary delays messages between A, C for a long time (TBD) until
- B interacts with A as if an honest node with private input = 1
- B interacts with C as if an honest node with private input = 0
- for all A knows, B is honest (with input = 1) and C has crashed forever (indistinguishable)
  - $\Rightarrow$ by eventual termination + validity, A eventually (at some time T_1) outputs 1
- for all C knows, B is honest (with input = 0) and A has crashed forever (indistinguishable)
  - $\Rightarrow$ by eventual termination + validity, C eventually (at some time T_2) outputs 0
- $\Rightarrow$ but A, C both honest, so this contradicts agreement.



Byzantine

QED!

The CAP Theorem

stated by Brewer, proved by Gilbert + Lynch (early 2000s)

- C for "consistency" [distributed system's behavior indistinguishable from a centralized system]
- A for "availability" [every client command eventually carried out]
- P for "partition tolerance" [properties should hold even when there's a network partition]

CAP: must pick 2 of 3.

⇒ with a network partition, must choose between consistency + availability

Proof idea: Initially x is 0.

Client issues to node i the command x:=1.

Next client issues "return x" to node i.

⇒ if i ever answers "1", violates consistency

⇒ if i always answers "0," violates availability

↑ since network is partitioned, nodes of B unaware of update

Network partition

(e.g., due to a DoS attack)

FLP Theorem vs. CAP Theorem

Takeaway: when under attack (asynchrony/network partition), need to choose between safety/consistency and liveness/availability.

CAP)

- network partition can last forever
- adversary restricted to network partitions
- applies even with all honest nodes [only adversary is message delivery]

FLP)

- every msg eventually delivered
  - adversary can do whatever (subject to)
- needs at least one faulty node (though one crash fault suffices)

captures enough of the power of infinite message delays