Tendermint Protocol

Saravanan Vijayakumaran

Associate Professor
Department of Electrical Engineering
Indian Institute of Technology Bombay

March 21, 2024

Tendermint

- Widely used consensus protocol that achieves consistency and eventual (post-GST) liveness when $f < \frac{n}{3}$
  - See `www.mintscan.io` for real-world deployments
- Assumptions
  - Partially synchronous network model
  - Permissioned, PKI
- Main ideas
  - Iterated single-shot consensus
  - Rotating leaders
  - Restarts after a timeout if messages are delayed
  - Two stages of voting
- We will describe the case when each node has a single vote
  - In practice, nodes cast weighted votes proportional to their stake

Rounds

- In the partially synchronous model, there is a known upper bound Δ on the message delays after GST
- In Tendermint, a **round** corresponds to 4Δ time steps
  - First round begins at $t = 0$ and ends at $t = 4\Delta$
  - Second round begins at $t = 4\Delta$ and ends at $t = 8\Delta$, and so on
- All nodes know the current round number r
- Each round has **four phases** each lasting Δ time steps
- Each round also has **two stages** of voting
  - Stage-1 voting happens in the second phase that begins at $t = 4\Delta r + \Delta$
  - Stage-2 voting happens in the third phase that begins at $t = 4\Delta r + 2\Delta$
- Each round has a unique **leader** whose ID is known to all nodes
- The leader proposes a block of transactions in a round
- A round may occur before GST has passed
- If a round does not conclude with consensus on the block, the nodes move on to the next round

Quorum Certificates

- Nodes vote on blocks. Each vote has five attributes
  - Identity i of the voter
  - The block B the vote is for
  - The block height h
  - The round number r
  - The voting stage s (first or second)
- Let us call the triple (h, r, s) a **referendum** (think of it as an election)
- **Definition**: A **quorum certificate (QC)** is a set of votes from at least $\frac{2}{3}n$ distinct voters that are all for the same block in the same referendum
- **Lemma**: Every pair of QCs overlaps in at least $\frac{n}{3}$ nodes
- **Corollary**: If $f < \frac{n}{3}$, then every pair of QCs overlaps in at least one honest node
- **Corollary**: Suppose that every honest node votes at most once per referendum and that $f < \frac{n}{3}$. Then if Q_1 and Q_2 are QCs for the same referendum, then Q_1 and Q_2 support the same block.

Ordering Quorum Certificates

- Given two QCs for a block height h, we want to say that one is newer than the other
- Every honest node i maintains two local variables for height h
  - A block B_i
  - A QC Q_i that supports B_i
  - For new blocks, Q_i is set to null
- B_i is node i's current belief about what the next block (at some height h) should be
- Node will change their beliefs as new information becomes available
- QCs are ordered by age as follows
  - Any non-null QC is **more recent** than a null QC
  - A non-null QC Q_1 with referendum (h, r_1, s_1) is **more recent** than another non-null QC Q_2 with referendum (h, r_2, s_2) if
    1. Q_1 is from a later round, i.e. $r_1 > r_2$, or
    2. Q_1, Q_2 are from the same round but Q_1 is from a later stage, i.e. $r_1 = r_2$ and $s_1 > s_2$
- If $f < \frac{n}{3}$, QCs with $r_1 = r_2$ and $s_1 = s_2$ support the same block; no ordering required

Protocol Pseudocode: Phases 1, 2

- Assumptions
  - Node i is working on block height h_i with local variables B_i and Q_i
  - Messages for older block heights are ignored
  - QCs for block heights $h_i + 1, h_i + 2$ are stored for future use
  - Current round is r with leader l
- **Phase 1** executed at time $t = 4\Delta r$

```
if i = l then                                                   // node is current leader
    if l has received a height-h_i QC newer than (B_l, Q_l) then
        B_l := B_j, Q_l := Q_j                                  // (B_j, Q_j) is the newest QC
    end
    broadcast(B_l, Q_l) to all nodes                            // annotated with h_i, r, signature
end
```

- **Phase 2** executed at time $t = 4\Delta r + \Delta$

```
if i has received (B_l, Q_l) from l then                        // must be signed by leader
    if Q_l is at least as recent as Q_i then
        B_i := B_l, Q_i := Q_l
        broadcast(B_i, Q_i)                                     // keep all nodes up-to-date
        broadcast first-stage vote for B_i                      // annotated with h_i, r, signature
    end
end
```

Protocol Pseudocode: Phases 3, 4

- **Phase 3** executed at time $t = 4\Delta r + 2\Delta$

if *i has received at least $\frac{2}{3}n$ round-r first-stage votes for B* **then**

    $B_i := B$ // may or may not change the value of B_i
    $Q_i :=$ the votes received // constitute a round-r stage-1 QC
    broadcast(B_i, Q_i) // keep all nodes up-to-date
    broadcast second-stage vote for B_i // annotated with h_i, r, signature

end

- **Phase 4** executed at time $t = 4\Delta r + 3\Delta$

if *i has received at least $\frac{2}{3}n$ round-r second-stage votes for B* **then**

    $B_i := B$ // may or may not change the value of B_i
    $Q_i :=$ the votes received // constitute a round-r stage-2 QC
    broadcast(B_i, Q_i) // keep all nodes up-to-date
    commit B_i to local history as block at height h_i
    increment h_i // start working on next block height
    reset B_i to list of not-yet-executed transactions
    reset Q_i to null

end

- **Addendum**: If height-h_i stage-2 QCs are available, execute phase 4 as needed before the first phase of round $r + 1$

Proof of Consistency

- **Theorem:** In the Tendermint protocol, if $f < \frac{n}{3}$ and two honest nodes commit blocks B and B' to their local histories at the same block height h, then $B = B'$.
- As soon as a single honest node commits a block B to its local history at height h, B is considered **finalized**
- What can go wrong?
  - Nodes i and j may commit different blocks B, B'
  - Node i commits block B but node j does not commit any block
- Proof
  - Let r denote the first round in which $> \frac{n}{3}$ honest nodes contribute height-h stage-2 votes in support of a common block B^*
  - Such an event is a prerequisite for a stage-2 QC as $f < \frac{n}{3}$
  - Denote this set of honest nodes by S
  - To support a different block $B \neq B^*$ in the referendum $(h, r, 2)$, at least one node from S must contribute a vote
    - Not possible as honest nodes do not vote twice in the same referendum
  - But what if a stage-2 QC supports a block $B \neq B^*$ for block height h in round $r + 1$?
    - $(h, r + 1, s)$ is a new referendum which can receive votes from S

Proof of Consistency (contd)

- State at the end of round r and before round $r+1$ begins
  - Some nodes in S may have already committed B^* at height h
  - If $i \in S$ has not already committed B^* at height h, then $B_i = B^*$ and Q_i is a stage-1 QC for referendum $(h, r, 1)$ supporting B^*
  - Every QC for referendums $(h, r, 1)$ and $(h, r, 2)$ supports B^*
- For a different block $B \neq B^*$ to be committed in round $r+1$, some node i in S has to vote for B at height h
  - This node did not commit B^* in round r
  - If node i is the leader of round $r+1$, it will broadcast (B^*, Q_i) to all nodes
  - If node i is not the leader of round $r+1$, it will cast a first stage vote only if it receives a QC which is at least as recent as Q_i.
    - But Q_i is a round-r QC and all QCs in round r support B^*
    - So node i can cast a first-stage vote only for B^*
    - As $|S| > \frac{n}{3}$, the referendum $(h, r+1, 1)$ cannot produce a QC for any block $B \neq B^*$
  - Node i can only vote for B^* in the second stage
  - As $|S| > \frac{n}{3}$, the referendum $(h, r+1, 2)$ cannot produce a QC for any block $B \neq B^*$
- The end of round $r+1$ satisfies the same three properties as the end of round r

References

- Foundations of Blockchains: Video lectures by Tim Roughgarden
- Lecture 7 from 2021 FoB course
  `https://timroughgarden.github.io/fob21/l/l6.pdf`
- E. Buchman, J. Kwon, Z. Milosevic, *The latest gossip on BFT consensus*,
  `https://arxiv.org/abs/1807.04938`