BFT Protocols

Introduction to Blockchain Science and Engineering

Aggelos Kiayias

Nikos Leonardos, Dionysis Zindros, Christos Nasikas

Byzantine Consensus (Binary inputs)

n parties (1,2,...,n), t adversarial.

Let $x_i \in \{0,1\}$ be the input of party i.

Honest parties should *decide* on values $y_i \in \{0,1\}$ satisfying the following two properties.

- **Agreement**: if parties i and j are honest, then $y_i = y_j$.
- **Validity**: if there exists $v \in \{0,1\}$ such that $x_i = v$ for each honest party i, then $y_i = v$ for each honest party i.

Furthermore, they reach a decision in finite time.

Exponential Information Gathering Algorithm (EIG)

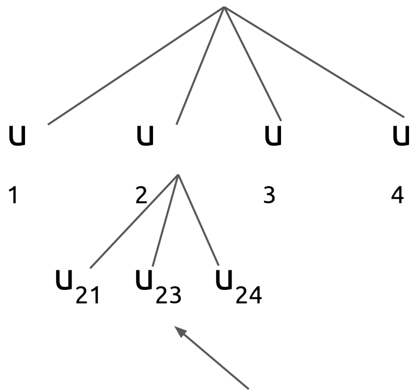
Algorithm Sketch.

- At round 1, send to everyone your input.
- At round r+1, send to everyone all messages you received at round r (avoiding redundant messages).

Each party arranges the messages in each own EIG tree.

- Let $u_1,\dots,u_n$ be the messages received in the first round.
- Subsequently, u_{xj} is the value received from j as the value u_x in j's tree.

Note: there need be no repetitions in the label of a node (e.g., x in u_x should contain distinct identifiers).



The value party 2 told me that party 3 send him in the previous round.

EIG Termination

The EIG algorithm terminates after t+1 rounds. The output value of each party is defined as follows.

- For each leaf v in the EIG tree, set $z_v=u_v$.
- For an internal node v, set z_v equal to the majority of the z-values of its children. If the majority is not defined, set $z_v=z_0$, for some default value z_0.
- Define the output as z_{root}.

Impossibility results I

Theorem[LSP1982] Impossible for $n<3t+1$.

Theorem[FL1982] Impossible in t rounds.

Example The EIG algorithm with $t=1$ needs at least 2 rounds.

1. If a party received a single 1, its output should be 1. (Because the 1 could be coming from the adversary.)
2. If a party received two 1s, its output should be 1. (Because one of them could have been send from the adversary, while another party received a single 1 and will decide on 1 according to the previous statement.)
3. And so on...

Theorem[GM1998] Doable for $n>3t$ in $t+1$ rounds.

Impossibility results II

Theorem[BT1985] Asynchronous Byzantine Consensus is impossible with $n<3t+1$, even if the parties have agreed on a PKI.

Proof Partition parties into sets A, B, C of size at most t. Consider 3 scenarios.

A. A malicious, B and C honest with inputs 0. The adversary sends no messages. The honest parties should decide on 0 until some time T_A.
B. B malicious, A and C honest with inputs 1. The adversary sends no messages. The honest parties should decide on 1 until some time T_B.
C. C malicious, B and A honest with inputs 0 and 1 respectively. The adversary communicates with B as the honest C in scenario A and with A as the honest C in scenario B. At the same time every communication between A and B is delayed for time at least $\max\{T_A,T_B\}$.

The crux is that A has the same view in scenarios B and C. Similarly for B, in scenarios A and C. Agreement in scenario C is impossible, if validity is achieved in scenarios A and B.

A blockchain related to proof-of-stake

Servers $S_1,\ldots,S_n$ with shared verification keys $pk_1,\ldots,pk_n$ and private signing keys $sk_1,\ldots,sk_n$.

B_0: Genesis block containing the public info.

$B_i = (k,d,sl,\sigma_{sl},\sigma_{block})$, where

k: hash of previous block

d: data

sl: slot number

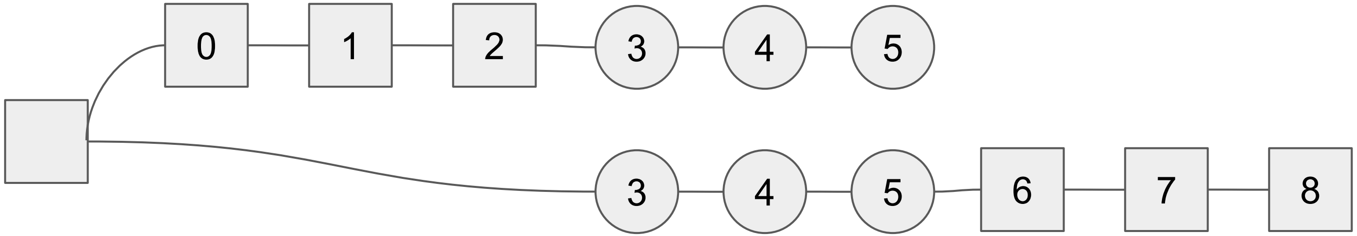
σ_{sl}: signature on sl by $S_{sl \bmod n}$

σ_{sl}: signature on whole block by $S_{sl \bmod n}$

Characteristic sequences and executions

The characteristic sequence of an execution with L slots 0,1,...,L-1 is a binary string w in $\{0,1\}^L$ such that w_i=1 iff $S_{i \bmod n}$ is adversarial.

Example w = 000111000. Squares denote honest slots (including the genesis) and circles the adversarial ones. The adversary keeps the chain ending with slot 5 on top hidden so that S_6 extends the bottom chain. This results in a **fork,** two disjoint chains (except for the genesis block) of maximum length. The corresponding characteristic sequence w is called **forkable.**



Forkable sequences

Fact If weight(w)<length(w)/3, then w is not forkable.

Proof Let n=length(w) and t=weight(w) and assume w is forkable. The two chains must have length at least n-t (prove formally by induction). Thus, 2(n-t) is a lower bound on the sum of their lengths. On the other hand, the n-t honest parties have contributed at most n-t, while the adversarial parties have contributed at most 2t (because a given chain contains a slot at most once). Thus, (n-t)+2t is an upper bound on the sum of their lengths. We have

$$2(n-t) \leq (n-t) + 2t \Longrightarrow n \leq 3t.$$

Consensus inspired from proof-of-stake [KR2018]

- Fix an arbitrary ordering of the servers: $S_1,\ldots,S_n$.
- Construct a blockchain for 5t+2 rounds, recording your own input bit as data in any block you create.
- Upon termination output the majority of the first 2t+1 blocks of your chain.

Theorem If $n > 3t$, the protocol satisfies agreement and validity.

Proof [KR2018] It can be shown that the first 2t+1 blocks are common to all honest parties; this implies agreement. Validity follows from the fact that among the first 2t+1 at most t are adversarial and so the majority of them belong to honest parties.

Bitcoin Consensus

- Miners run the Bitcoin protocol recording their own input bit as data in any block they compute.
- When their chain has at least 2k blocks (for some security parameter k), the broadcast it and stop.
- Output is the majority of the bits recorded in the first k blocks.

Theorem [GKL15] If $t<n/3$, the above protocol satisfies Agreement and Validity with probability $1-e^{-\Omega(k)}$.

Remark For Agreement, $t<n/2$ suffices.

Common-Prefix Property and Agreement

Common-Prefix Property For any pair of honest parties adopting the chains C_1 and C_2 at rounds r_1 and r_2 respectively. If $r_1 \leq r_2$, then $C_1[-k]$ is a prefix of C_2, where $C_1[-k]$ is C_1 without its last k blocks.

Common-Prefix Property implies Agreement. This is because the parties are pruning at least k blocks from their chains when keeping only their initial k blocks. Thus, the initial k blocks are common to all honest parties.

In [GKL2015] it is shown that in Bitcoin the Common-Prefix fails with probability exponentially small in k, if the adversary's hashing power is sufficiently bounded below ½. It follows that Agreement is satisfied with probability $1-e^{-\Omega(k)}$, when t is sufficiently less than n/2.

Chain-Quality Property and Validity

Chain-Quality Property (informal) Among any sufficiently large number of consecutive blocks in an honest party's chain, a fraction of at least (n-2t)/(n-t) have been computed by honest parties.

Chain-Quality implies Validity, when t is sufficiently less than n/3. This is because, in that case, the majority of the first k blocks have been computed by honest parties. Thus, if all honest parties have input v, the majority of values recorded in the first k blocks will be v.

In [GKL2015] it is shown that in Bitcoin Chain-Quality fails with probability exponentially small in k, if the adversary's hashing power is sufficiently bounded below ½. It follows that Validity is satisfied with probability $1-e^{-\Omega(k)}$, when t is sufficiently less than n/3.

References

[BT1985] Bracha, Toueg. Asynchronous consensus and broadcast protocols.

[FL1982] Fischer, Lynch. A lower bound on the time to assure interactive consistency.

[GKL2015] Garay, Kiayias, Leonardos. The Bitcoin Backbone Protocol: Analysis and Applications.

[GM1998] Garay, Moses. Fully polynomial Byzantine agreement for n>3t processors in t+1 rounds.

[KR2018] Kiayias, Russell. Ouroboros-BFT. A simple Byzantine Fault Tolerant Consensus Protocol.

[LSP1982] Lamport, Shostak, Pease. The Byzantine generals problem.