# Consensus

## Fast Byzantine Fault Tolerance (FBFT)

The consensus algorithm is a critical component of any blockchain. It determines the security and performance and is often referred to as the "engine" of a blockchain. Harmony's consensus algorithm is called **Fast Byzantine Fault Tolerance (FBFT)**, an innovative upgrade to the famous PBFT algorithm. FBFT is one order of magnitude faster and more scalable than PBFT because BLS (Boneh–Lynn–Shacham) aggregate signature reduces the communication cost significantly. Specifically, FBFT allows at least 250 validators to reach a consensus within 2 seconds.

For every round of consensus in FBFT, one validator serves as the "leader," There are three phases: the announce phase, the prepare phase, and the commit phase. In the announce phase, the leader proposes a new block and broadcasts the block hash to all validators. In the prepare phase, validators verify the message, sign on the block hash, and send the signature back to the leader. The prepare phase finishes when signatures with more than 2/3 of the voting power are collected. After that, the leader aggregated the collected signatures into an O(1)-sized BLS aggregate signature and then broadcast it with the whole block to start the commit phase. The commit phase involves validators verifying the block and doing a similar signing process as the prepare phase (i.e., 2/3 voting power collection). The consensus is reached after the commit phase is complete. This whole process can be within 2 seconds on the mainnet.

Network communication during a single round of consensus.

Specifically, Harmony's FBFT consensus involves the following steps:

1. The leader constructs the new block and broadcasts the block hash to all validators. This action is called the "announce" phase.
2. The validators check the message's validity, sign the block hash with a BLS signature, and send the signature back to the leader.
3. The leader waits for valid signatures with more than 2/3 voting power from validators (including the leader itself) and aggregates them into a BLS aggregate signature. Then the leader broadcasts the new block and the aggregated signature along with a bitmap indicating which validators have signed. Together with Step 2, this concludes the "prepare" phase.
4. The validators check that the aggregate signature has at least 2/3 of total voting power, verify the new block, sign the received block from Step 3, and send it back to the leader.
5. The leader waits for valid signatures with more than 2/3 voting power (can be different signers from Step 3), aggregates them together into a BLS aggregate signature, and creates a bitmap for all the signers. Finally, the leader commits the new block with the aggregate signatures and bitmaps into local DB and broadcasts the aggregate signature and bitmap for all validators to commit the block confidently. Together with Step 4, this concludes the "commit" phase.

## Synchronous View Change

One drawback of BFT-based consensus is the potential installment of the consensus if the leader is malicious. The solution to this in the PBFT algorithm is the additional view change protocol on top of the consensus, which can switch leaders when the consensus becomes stalled. The view change in PBFT works well in the traditional distributed system setting, but it fails in the more complicated blockchain space. Notably, the view change in PBFT relies on the timeout mechanism where the timer is triggered based on the live progress of the consensus. This works fine if the nodes are always online and are in sync with the consensus progress. However, this won't be true in the actual world situation where nodes can be down for extended time or restarted due to machine crashes, which will make the nodes having inconsistent view IDs and not able to make progress in the view change.

We solve this problem by using an improved version of the view change protocol that's fully synchronous based on the local clock rather than assuming the liveness of the nodes. Specifically, instead of setting the validator's view ID based on the live progress of the consensus, the view ID is calculated based on the elapsed time of the last successfully committed block's timestamp. Of course, there is no global clock to rely on in the calculation of elapsed time. Each validator will use their local clock for that. However, this is fine as long as more than 2/3 of the validators maintain a relatively accurate local watch that's not drifting by a few seconds. This accuracy is achievable in the protocol to make validators synchronize their local clock with NTP time periodically.

This improvement makes our view change protocol fully robust and functional as long as more than 2/3 of the honest validators are online, guaranteeing the liveness of the FBFT consensus. Besides, BLS aggregate signatures are also used in the view change protocol to reduce network communication costs, making it a very efficient process that takes a few seconds to finish.