Why we need BFT consensus？

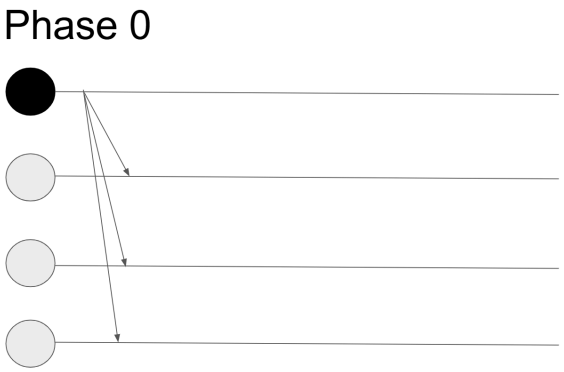
Consensus is essential in distributed systems, particularly in blockchain, to ensure that all nodes agree on a single, consistent version of data despite being spread across different locations and potentially facing failures or malicious actors. Without consensus, nodes might have conflicting data, leading to inconsistencies and unreliable operations. Key reasons why consensus is needed:

- Data Consistency: Ensures all nodes maintain the same state, even when nodes join or leave the network.
- Fault Tolerance: Allows the system to function correctly even if some nodes fail or act maliciously (Byzantine Fault Tolerance).
- Trustless Environment: In blockchain, nodes may not trust each other, so consensus ensures trust through protocol rather than authority.
- Prevents Double Spending (all transactions are in sequence): In cryptocurrencies, consensus ensures that digital assets are not duplicated or fraudulently used more than once.
- Decentralization: Removes the need for a central authority, distributing decision-making across the network.
- Reliable Operations: Guarantees that transactions or operations are processed correctly (among honest nodes) and consistently, regardless of network conditions or failures.

Why need 3 phases?

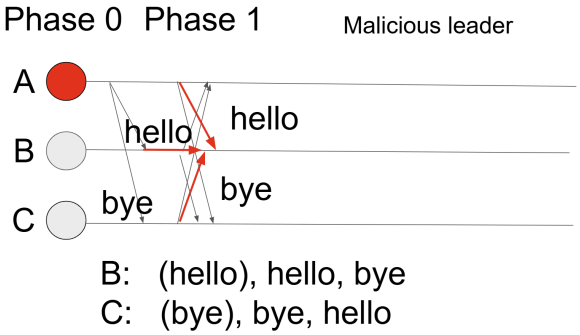
First phase: Proposal (we name it *proposal* phase)

- The leader proposes a value and sends a *message* request to all nodes to vote.

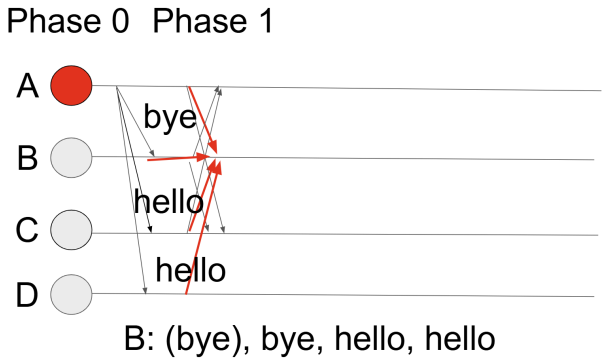


Second phase: Accept phase (we name it accept phase)

- All the nodes need to broadcast messages to all the users, and all the users will receive **n** messages from different nodes. E.g., you received a message from an untrusted user, you don't know whether this message is secure or not, you have to send a message to all the students to ask for evidence. And other students also don't know whether it is secure. So, that's the reason everyone in the phase will broadcast and receive.
- In these **n** messages, it may have malicious messages, so each node needs to make a decision to accept one message (majority vs minority).
- However, if there are only 3 nodes, it can **not** tolerate the malicious leader.



So, we need 4 nodes to tolerate 1 malicious node. E.g., in phase 1 below, B will receive 2 byes and 2 hello, B will get confused and will report this to all the correct nodes, say: our leader sends me bye, but send both of you hello, he is a malicious leader, let's change a new leader.

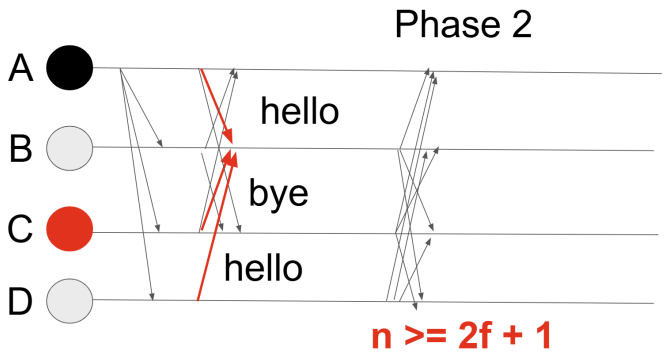


And the final equation will be : **n >= 3f +1**

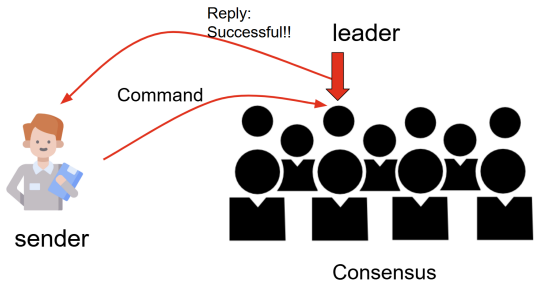
faulty	total number
1	4
2	7
3	10
4	13
5	16
6	19
7	22
8	25
9	28
10	31
11	34
……..	
992	2977
993	2980
994	2983
995	2986
996	2989
997	2992
998	2995
999	2998
1000	3001

Tolerating rate = 1000 / 3001 = 0.333 = 33.3%

Third phase: (Commit Phase) Reply the final message to the leader.
When you make the final decision, you need to tell everyone, to make sure everybody is on the same page.



When the leader gets notified that all the nodes have already made the decision, he/she will report to the sender, and a new consensus will start.



Drawback of BFT consensus:
Consensus helps computers agree on the same thing, but it has some problems:

1. **Slow (High latency)**: It takes time for all computers to talk to each other and agree.
2. **Hard to Scale**: More computers mean more talking, which slows everything down.
3. **Uses Lots of Power**: Some methods need a lot of computer power, like Bitcoin's system.
4. **One Leader Can Fail (Malicious leader)**: If one computer is the leader, the system can slow down or fail if that leader has a problem.
5. **Can Be Attacked**: Bad people can try to break the system by pretending to be many computers.
6. **Complicated**: Setting up and running these systems is not easy.
7. **Not Fast Enough**: Sometimes, they can't handle too many requests at once.