Proposal for Github Externship 2021

Consensus Algorithm

Project Objective :

Development of a universal yet flexible consensus algorithm supporting the participation of IOT and Mobile devices with most efficient utilization of resources.

Introduction :

Blockchain is a decentralized, distributed network which provides all its users privacy, security, transparency in various walks of life. The three main features of any blockchain based system is – **decentralization**, **security** and **scalability**.

The absence of a central authority gives rise to the problem of validation of the blocks and transactions, hence, a major issue of security and reliability. This issue is dealt with consensus. A consensus algorithm is a procedure through which all peers of the blockchain reach a common agreement about the state of any distributed ledger.

The main issues addressed by consensus algorithms include agreement, termination and integrity. **Termination**, a liveness property, means all correct processes eventually produce a value. **Agreement**, a safety property, means that all correct processes select the same value and these values are valid within the dictates of the protocol. In other words, the state of being "correct" is determined by how many other processes share one value. **Integrity** addresses the ability to recover from the failure of a participating node.

Safety and Liveness :

A property is an attribute of a program that is true for every possible execution of that program. Before proceeding to the CAP theorem, one must know the impossibility of guaranteeing both safety and liveness in an unreliable distributed system. A **safety** property asserts that nothing bad happens during execution. A **liveness** property asserts that something good eventually happens. Another way of putting this is that safety is concerned with a program not reaching a bad state and that liveness is concerned with a program eventually reaching a good state.

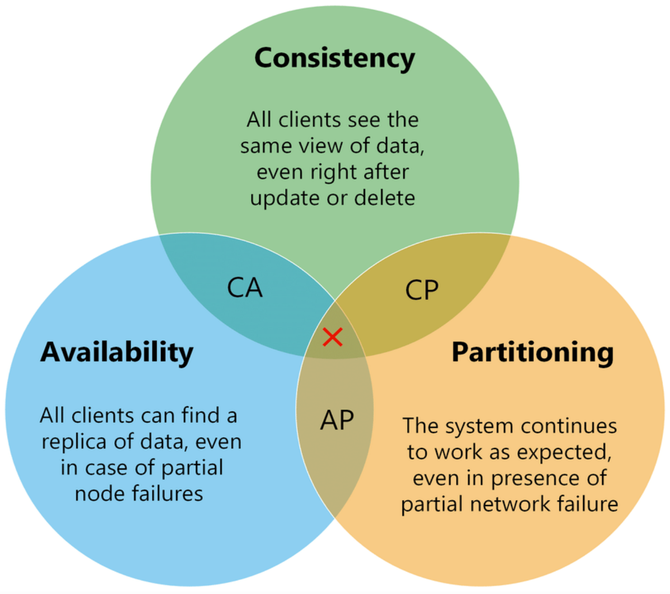
The CAP Theorem :

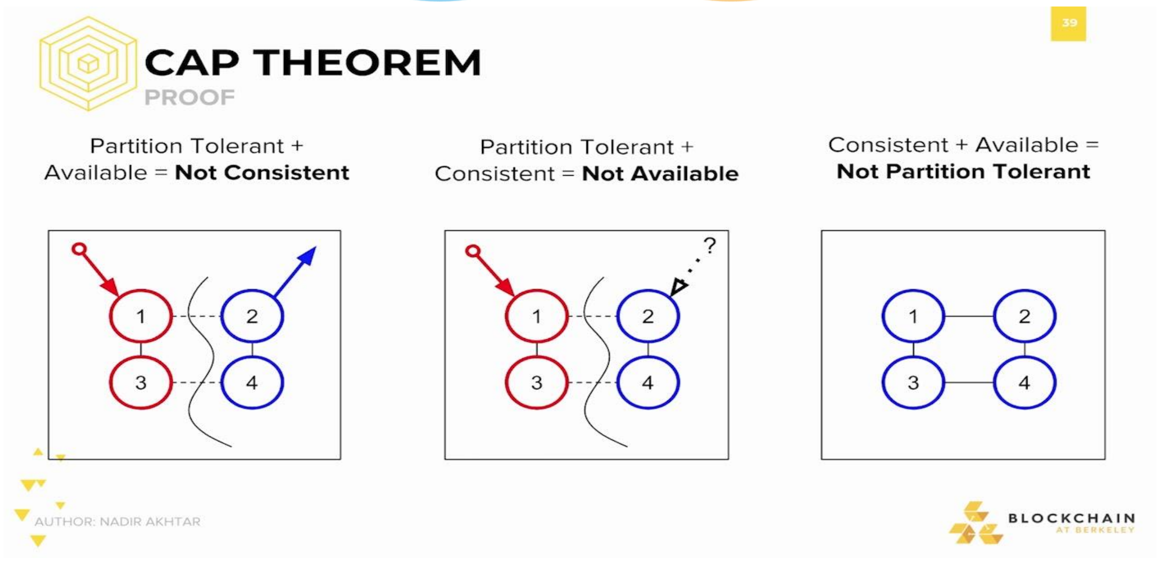
The CAP theorem states, it is impossible to for a distributed data store to simultaneously provide more than two out of the three guarantees : Consistency, Availability and Partition tolerance.

Consistency - Every node provides the most recent state, or does not provide a state at all. This builds up the **idea of safety** as it refers to something that will never happen.

Availability - An available system allows a user to make updates and retrieve the state of a system without any delay. This builds up the **idea of liveness**.

Partition Tolerance – Partition is the inability of two or more nodes in a network to communicate with each other. Partition tolerance is the ability of a network to work in spite of partition. This also builds up the **idea of safety**.


Consistency
All clients see the same view of data, even right after update or delete
CA
CP
Availability
All clients can find a replica of data, even in case of partial node failures
Partitioning
AP
The system continues to work as expected, even in presence of partial network failure


39
CAP THEOREM
PROOF
Partition Tolerant + Available = Not Consistent
Partition Tolerant + Consistent = Not Available
Consistent + Available = Not Partition Tolerant
1
2
3
4
?
AUTHOR: NADIR AKHTAR
BLOCKCHAIN AT BERKELEY

Problems :

A few basic consensus algorithms and their drawbacks have been mentioned in table 1A. Besides these disadvantages, there is also that none of them are resistant to both – **crash failure** and **Byzantine failure**. Fortunately, the modern day blockchain organisations use a combination of different consensus theorems to make their blockchain more secure and scalable. Still, a perfect solution is yet to be discovered.

In recent years there have been many proposals that attempt to overcome the performance hurdle of these consensus. The Delegated Proof-of-Stake approach achieves high throughput, but at the **expense of decentralization**. **Algorand** and **DFINITY** achieve a better level of decentralization by introducing a rotating set of committees selected at random to produce new blocks. However, both techniques lack a mechanism to allow other nodes to approve the block produced by the committee. Although unlikely, the random selection process could happen to **elect a committee with more than two-third malicious nodes**. In such cases, the resulting blockchain might contain invalid blocks.

On a different track, there have been other proposals like **ByzCoin** and **OmniLedger**, which allow many nodes to jointly finalize new blocks. They employ a single leader to produce blocks in order to achieve high throughput, with a number of nodes validating the blocks afterwards. This significantly **defeats the purpose of blockchain which is supposed to be 'decentralized'**.

Besides these, **Ouroboros**, the protocol used by **Cardano** is "*the first provably secure proof-of-stake*". It combines unique technology and mathematically – verified mechanisms with behavioural and economic psychology. **Theta Network** has introduced a **multi – level BFT consensus mechanism** that allows thousands of consensus participants, and yet can achieve high transaction throughput.

In the field of **CFT consensus protocols**, there are a few worth mentioning including **Paxos** and Raft, which is an implementation of Paxos. An improved version of **Raft**, **Tangaroa** (Raft BFT), has also gained popularity among permissioned blockchains but is still to be implemented on a blockchain platform.

Sr. No.	Consensus Theorems	Drawbacks
1.	Practical Byzantine Fault Tolerance	> Suitable only for small number of nodes > Reduced decentralisation
2.	Proof of Work	> Energy inefficient > Not scalable
3.	Proof of Stake	> Not scalable
4.	Delegated Proof of Stake	> Reduced decentralisation > Byzantine intolerant
5.	Paxos	> Reduced decentralisation > Byzantine intolerant

Table 1A : A few basic consensus theorems and their drawbacks

Consensus Protocol	Crash Fault Tolerant	Byzantine Fault Tolerant	Disadvantages
Theta Network / multi BFT Consensus	✓	X	> voting mechanism is always a subject to two-thirds malicious nodes risk
Quorum / Raft	✓	X	> Does not focus on scalability
Bitcoin / PoW	X	✓	> Energy inefficient
EOS/DPoS	X	✓	> Generates decentralization > Requires large number of users
IOTA / Tangle (DAG)	-	-	> No support for DApps > 'Tip' or a new transaction in Tangle is not verified

Table 1B : Different consensus algorithms used in the blockchain industry and their disadvantages

The Proposal :

Before the formation of a block takes place, every transaction needs to be validated. Transaction validation is much slower than block construction, so my first pass would be to drastically increase the number of transaction validator nodes. So first, let's deal with the issue of **validation of transaction along with scalability**. My idea includes the use of **sharding** just like Quarkchain. Sharding is a way of partitioning to **spread out the computational and storage workload** across a peer-to-peer network so that each node isn't responsible for processing the entire network's transactional load. Instead, each node only maintains information related to its partition, or shard.

Sharding is to be combined with the **Proof of Stake** consensus which is used in Ethereum, Peercoin. This makes the blockchain **energy efficient** and makes it **Byzantine Fault tolerant** as well as **Crash Fault tolerant**. Using sharding and PoS together **increases security** as well. Similar to Ethereum 2.0, there will be **collector nodes** and **validator nodes**. The collector nodes will **gather mini-descriptions of transactions and the current state of the shard** – and send it up to the validator nodes. The validator nodes will **collect the data from the collators, process the transactions and validate the blocks**. The collectors and validators are **assigned randomly** and **reshuffled regularly** as well, so it becomes almost impossible for an attacker to "target" one specific shard for an attack.

Further, to prevent any **malicious node** to creep in or any **attacker** to corrupt the blockchain every node uses the **BLS based signature aggregation** scheme. Attackers will not be able to manipulate or partition the network because fake signatures cannot be forged and cryptographic hashes cannot be inverted.

Therefore, I believe that this approach will be able to solve many problems and tends to achieve, till some extent, consistency, availability as well as partial tolerance.

Name: Utkarsh Singh
Institute: Thapar University
Email: usingh3_be19@thapar.edu