Blockchains Architecture, Design and Use Cases

Prof. Sandip Chakraborty
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture – 48
Research Aspects - I (Consensus Scalability)

Welcome back to the course on Blockchain. By this time I guess that you got a good grasp of this blockchain technology and the many of its use cases. So, we have broadly covered into the fundamental building blocks of a blockchain platform. Many of its use cases that myself along with Praveen, we have covered from the perspective of industry use cases, the use cases at the government level, the use cases at the financial sectors and many other things.

Praveen has also given you many examples of these hyperledger fabric platform through which you can write a smart contract and you have learned that how to write those smart contracts and how to utilise the concept of smart contract for many of the use cases. So, far is good. I guess that by this time you have written you have started writing your own blockchain application and you have got familiar about the centre smart contract platform and how to build upper smart contracts from scratch.

But well, although the entire picture looks good, but it is not as good as we are thinking of. So, there are certain glitches, certain shortcoming of the existing platform on blockchain and lots of researches are going on in the academics or as well as in the industry to make the platform more suitable for the application use cases.

So, in the next few lectures we will look into different such use cases which are they are on the development of the blockchain platform and how you can perform or how you can participate in this research procedural, what are the different open challenges that people are still facing and people are trying to search collectively, what are the different scopes of doing research in this blockchain platform apart from the general application development. You can always think of a novel application or you can always have a out of the box thinking to device or generate or design certain blockchain application.

But apart from that what are the different research challenges which are there at the blockchain system level along with the blockchain implementation and its performance level that we will look into in details. So, let us start looking into the various research aspects in the blockchain domain.

(Refer Slide Time: 02:55)

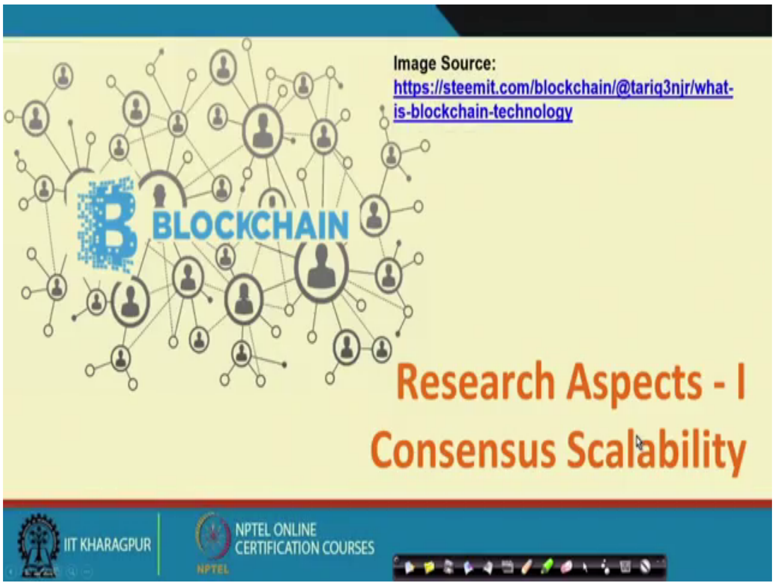


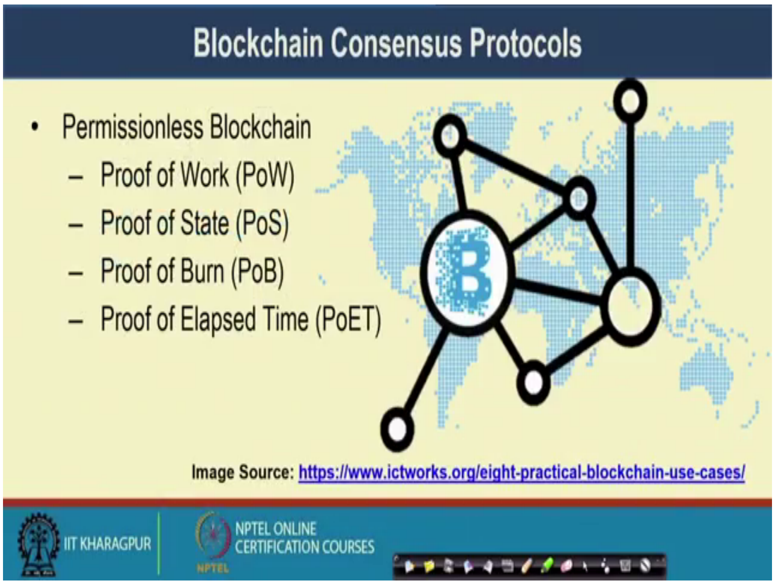
Image Source: https://steemit.com/blockchain/@tariq3njr/what-is-blockchain-technology

Research Aspects - I

Consensus Scalability

First, we will start with the blockchain consensus algorithm that is there. So, already we have seen that there are two classes of blockchain environment and like the permission less blockchain and the private blockchain or the permission blockchain. And these two classes of blockchain they used to separate classes of consensus algorithm, whereas the permission less blockchain it uses consensus algorithm based on challenge response strategy like the proof of work proof of state this kind of algorithms. Whereas, the permission model of blockchain it uses byzantine fault tolerance base algorithm, the BFT classes of consensus algorithm.

(Refer Slide Time: 03:40)



Blockchain Consensus Protocols

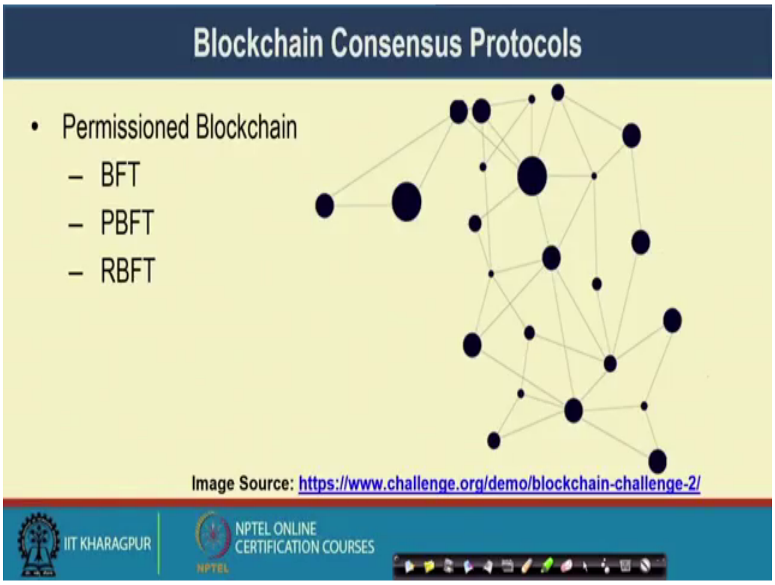
- Permissionless Blockchain
  - Proof of Work (PoW)
  - Proof of State (PoS)
  - Proof of Burn (PoB)
  - Proof of Elapsed Time (PoET)

Image Source: https://www.ictworks.org/eight-practical-blockchain-use-cases/

So, let us look into the details of that. So, what we have learnt in the basic discussion of blockchain platform the consensus algorithm, that the permission less model of blockchain which is open environment where anyone can join in the blockchain platform. There we have look into these different groups of consensus protocols like this proof of work which was the first consensus protocol proposed by Satoshi Nakamoto sometimes we call it as a Nakamoto consensus.

Then we have looked into the problems of proof of work because of this mining share and we have moved into different other consensus protocols to solve the problem of proof of work in terms of mining shared and in terms of the resources which are being utilised the power. So, we have looked we have seen that proof of work is very power hungry and that is the reason we moved to the other different classes of protocols for open blockchain or the permission less blockchain environment. Like this proof of state proof of burn and proof of elapsed time.

(Refer Slide Time: 04:45)



Blockchain Consensus Protocols

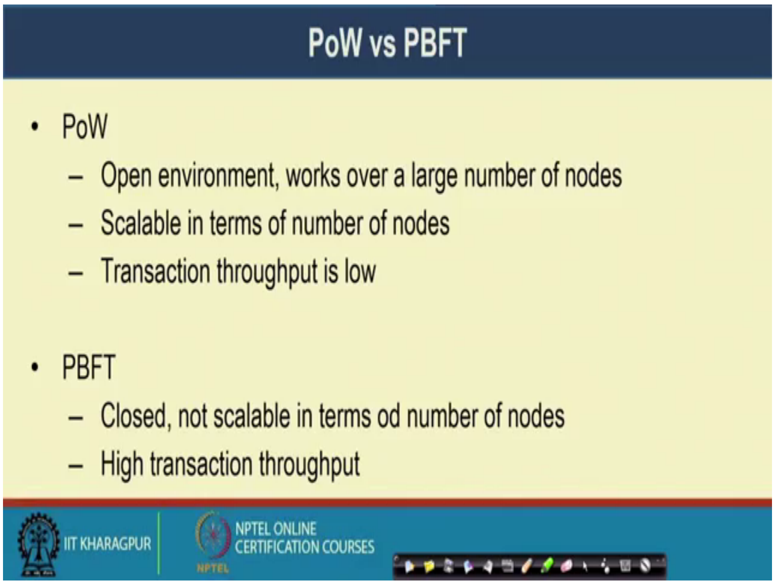
- Permissioned Blockchain
  - BFT
  - PBFT
  - RBFT

Image Source: https://www.challenge.org/demo/blockchain-challenge-2/

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

Similarly in the permission blockchain environment, we have seen that the entire consensus algorithm it is primarily governed by different variants of byzantine fault tolerance protocols, ranging from standard byzantine fault and algorithms the PBFT the practical byzantine algorithm. And in case of hyper ledger in the platform we have seen another class of BFT algorithm which is called as the redundant byzantine fault tolerant or RBFT class of algorithms.

(Refer Slide Time: 05:14)



PoW vs PBFT

- PoW
  - Open environment, works over a large number of nodes
  - Scalable in terms of number of nodes
  - Transaction throughput is low
- PBFT
  - Closed, not scalable in terms od number of nodes
  - High transaction throughput

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

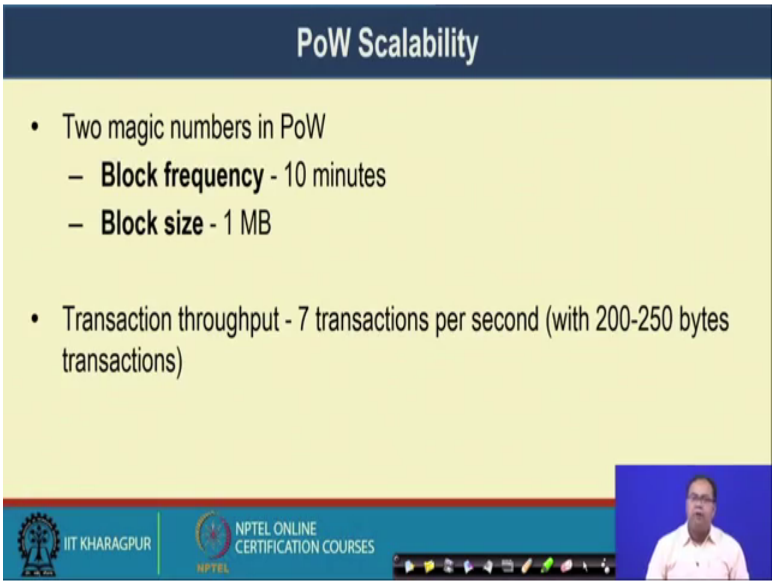
Now, if we compare these two classes of algorithms the proof of work verses the BFT classes of algorithms. So, this proof of war they are particularly designed for a open environment based on a challenge response. So, at the network closer challenge and the minus they collectively solves that challenge tries to solve that channel challenge and whoever is first able to give the response he is able to civilise the block and put the block in the blockchain. So, it works over a large number of nodes in a open environment.

So, in that particular aspect proof of work is scalable in terms of number of nodes but after a few minutes we will see that the transaction throughput. That means, the number of transaction that can be supported per unit time say per second that is very low for proof of work on the other hand the PBFT or the byzantine fault tolerance classes of algorithm they are closed algorithm.

So, you have seen that the pre payer and the comic phase in the PBFT algorithm you need to share the messages among the peers and in that case our assumption is the network is closed everyone knows what the peer is. So, the primary knows who are the backups as well as the backup knows that who are the other backups and the primary is and that way they multi caste the messages among that closed group. But the problem is there that because of this message passing, if the number of nodes in this particular network grows up then you have to transfer a lot of message.

Ideally the message complexity of a general BFT algorithm is o of n cube, where n is the number of nodes or participating nodes in the network. So, in that particular case PBFT is not scalable in terms of number of nodes. But it can support high transaction throughput because you can include any number of transactions which is possible there and those transactions can get validated and the consensus can get reached.

(Refer Slide Time: 07:16)



PoW Scalability

- Two magic numbers in PoW
  - **Block frequency** - 10 minutes
  - **Block size** - 1 MB
- Transaction throughput - 7 transactions per second (with 200-250 bytes transactions)

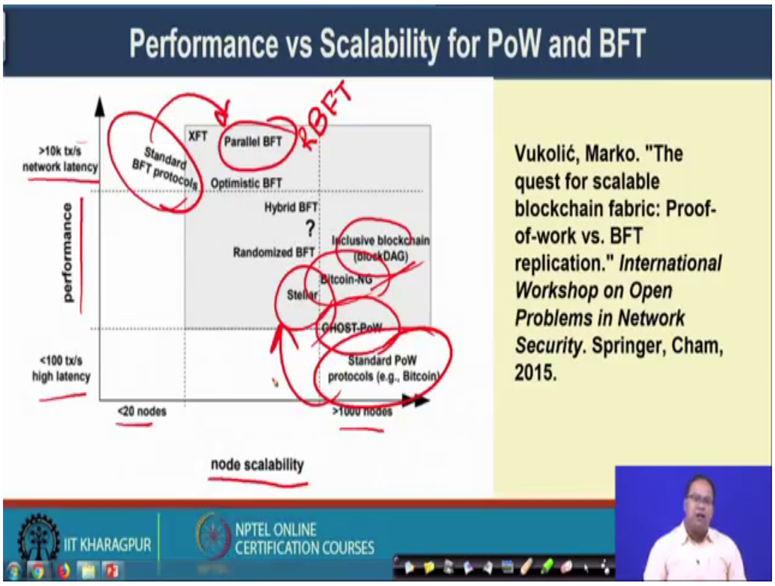
So, in terms of proof of our scalability as I have mentioned that proof of work is not very scalable in terms of transaction throughput, to see why that the proof of work has two magic numbers.

The first one is the block frequency the block frequency indicates that at what frequent you generate the block which is controlled by the mining difficulty. And as we have seen earlier that we try to ensure that every 10 minutes we will generate 1 block. So, on average at every 10 minutes 1 block will get generated. And also it makes a restriction on the block size, so a block size is restricted into 1 MB as per the original bitcoin proposal by Satoshi Nakamoto later on it got increased up to 8 MB. But with these particular standard parameters like the block frequency of 10 minutes and a block size of 1 MB you can see that the transaction throughput can be something like 7 transactions per second, with a transaction size close to 200 to 250 bytes. So, you can get around 7 transaction per second at maximum.

So, that way this transaction scalability or the transaction throughput is very low for the proof of work type of algorithm. So, if you just compared with a normal financial system, where transactions are being done like visa or master card environment there are around forty million transactions per second.

So, you can think of that well there is a huge difference between the transaction throughput that proof of work promises with its default setup of this magic numbers and actual transactions which are being done on a real number in a financial sector on top of visa and master card. So, in one hand for visa and master card you need to support around 40 million transactions per second but proof of what can give you only 7 transaction per second. So, that way proof of work is very limited in terms of transaction scalability. So, it was always the hot topic of research to look into that how you can improve the transaction scalability for a proof of work base system.

(Refer Slide Time: 09:29)



Performance vs Scalability for PoW and BFT

Vukolić, Marko. "The quest for scalable blockchain fabric: Proof-of-work vs. BFT replication." *International Workshop on Open Problems in Network Security*. Springer, Cham, 2015.

Well, so there was a interesting paper or interesting system that was proposed or that was this is a interesting paper which is most likely a report that was came from Marko Vukolic. So, Vukolic is a researcher at IBM Zurich. So, in this paper the quest for scalable blockchain fabric proof of work verses BFT replication.

So, Vukolic has given this nice diagram that actually puts all the transaction, all the consensus algorithm into a two scale in one dimension in x dimension we have this node scalability. It says that how many number of nodes this particular transaction can support or how many number of nodes this particular consensus protocol can support, starting from less than 20 nodes to more than 1000 nodes. And in an another axis the y axis you have the performance in terms of transaction throughput. So, from some less than 100 transaction per second that means, your block commitment latency is very high to some a 10k transaction per second to where your network latency is the prime factor, the block latency is not a prime factor.

So, if you look into the PoW kind the protocol the proof of work kind of protocol, proof of work kind of protocol comes into this end where it has good scalability in terms of number of nodes that can be supported. But the scalability is very less in terms of the transaction per second that can be supported. On the other hand, the standard BFT protocols comes into this coordinates, like where it supports good transaction scalability like more than 10k transaction per second. But the number of nodes that can be supported with the standard BFT protocol it is typically more less than 20 number of nodes.

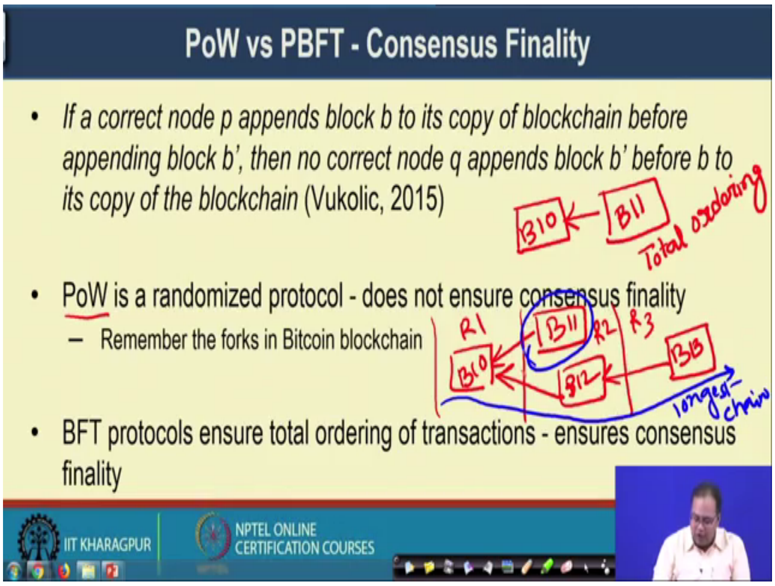
So, there were multiple variants that came from both this direction and this direction where people have tried to find out a scalable way of having consensus in blockchain environment. So, there are multiple variant of BFT that was being proposed one is this parallel BFT. So, the RBFT that, the RBFT algorithm that we have seen in the context of hyper ledger indeed, that is a parallel BFT algorithm then you have the optimistic BFT, hybrid BFT, randomized BFT or these different BFT classes of protocols on the that.

Those are kind of protocols which actually which actually make modification or make amendments on top of the standard BFT protocol to make it more scalable in terms of number of nodes that can be supported. On the other hand for the proof of work based protocol people are also looking into how you can increase the performance of a proof of work based protocol.

So, in that direction there are multiple other attempt like this ghost PoW, this block DAG, the Bitcoin-NG. So, we will look into Bitcoin-NG details, the stellar consensus protocol. So, multiple search consensus protocol came into existence from the researchers at various level. They tried to find out of proof of work based protocol which will help you to achieve more transaction scalability compared to the standard Nakamoto consensus. So, the standard proof of work protocol we will term it as a Nakamoto consensus that was proposed by Satoshi Nakamoto and there are these multiple variants of proof of work.

So, with this as scalability problem in proof of work and PBFT. So, both are scalable in one dimension, but it is not scalable in another direction or on under dimension. It is either scalable in terms of number of nodes that can be supported, but not scalable in terms of a transaction throughput at the performance or the vice versa. So, this kind of approaches or various kind of approaches are coming from the research domains both from academy as well as from industry that how you can make this blockchain consensus protocol more scalable.

(Refer Slide Time: 13:53)



PoW vs PBFT - Consensus Finality

- *If a correct node p appends block b to its copy of blockchain before appending block b', then no correct node q appends block b' before b to its copy of the blockchain* (Vukolic, 2015)
- PoW is a randomized protocol - does not ensure consensus finality
  - Remember the forks in Bitcoin blockchain
- BFT protocols ensure total ordering of transactions - ensures consensus finality

IIT KHARAGPUR

NPTEL ONLINE CERTIFICATION COURSES

So, other issues that was there where we can compare the performance of proof of work and PBFT in terms of consensus finality. So, let us see what is consensus finality is. The definition which was given by Vukolic in that paper in 2015 paper that, if a correct node p appends block b to its copy of blockchain before appending block b prime then no correct node q appends block b price before b to its copy of blockchain.

So, it is just like that if you have this particular block in a blockchain followed by another block say this is a block 10 and this is blocked 11 then every correct nodes every correct node in the network will append block 11 after block 10. So, that is the concept of consensus finality, which ensures that there is always a total ordering among the blocks, there is always a total ordering among the blocks.

So, this total ordering ensures that well whatever transactions which are there on top of this blocks they are in this serializable and that is realizability has been accepted by all the nodes in the network. But interestingly if you look into the proof of work based protocol so this proof of work inherently it is a randomised protocol. Why we call it a randomised? That challenge is thrown by the system that you have to generate certain has value based on a set of given constant like you have to include the previous block has the Markey root and you have to find out a norms with a constant that the generated has will be less than certain target value.

With that particular constant every minors they are independently trying to generate the blocks. Now, whenever they are independently trying to generate the blocks and vita cryptographic has function although the desirable properties that it should be collision free but at theoretically it can never be collision free. Collision can always happen. Because collision can always happen vita cryptographic has function it may happen that two nodes two minors, at two ends, they come to two different blocks come with different blocks with the correct hash value.

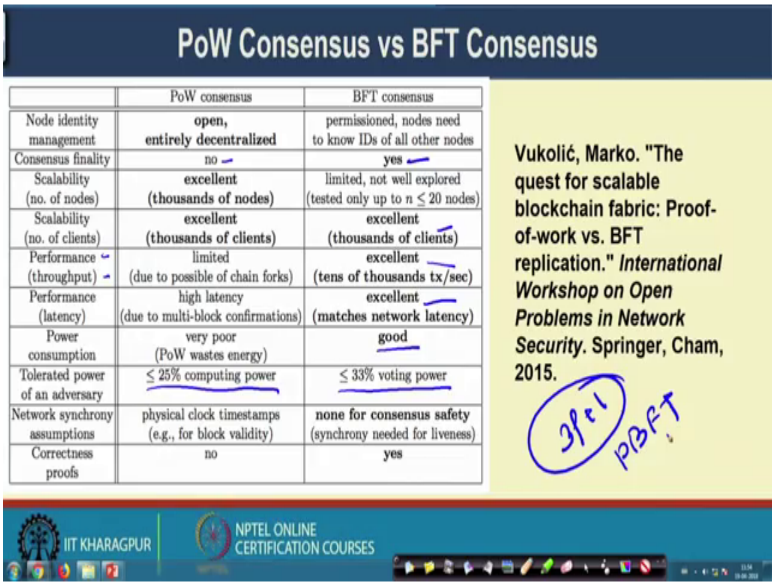
Now, if it happens that means, you are having the forks in the bitcoin blockchain. So, if you remember the concept of this forking in bitcoin blockchain that means, in one block after B 10, 1 node or 1 minor it has appended B 11 another minor it has appended say B 12. Now, after that if in the next round, so this was one round say R 1, round 1 this was round 2, now at say around 3 if it happens that well a minor adds up the next block say B 13 after B 12 that means, this B 11 it becomes a for; so this B 11 becomes a for and this is never been going to use. So, we always use the longest chain. So, this is the longest chain. So, you always use the longest chain as a part of my blockchain.

So, that way what happens that this particular concept that well you can have for in bitcoin blockchain based on proof of work or based on Nakamoto consensus it violates this properties of consensus finality. So, the consensus finality says that this ordering of the blocks will always be same.

So, all the correct nodes will come to a single chain, there would be no sub faults. But because there is fault because of this randomise nature of the proof of work algorithm that the probability it always depends on the probability based on the mining difficulty a. Based on this probability and the availability of the computation power certain nodes can generate the blocks and it is always possible that more than two minors generate two different blocks simultaneously this proof of work does not support this consensus finality.

On the other hand, the BFT protocols they ensure the total ordering of the transaction. So, they ensure consensus finality. So, the BFT protocol if you remember that you had a primary. The primary proposed certain sequence number for every transaction or for every request and all the backup if they agree on that particular sequence number then that particular request gets committed at the client. So, it says that every transaction that comes with a sequence number which is proposed by the primary and because of that you can always ensure the total ordering of the transaction. So, at the total ordering of the request that way the BFT class of protocols they support consensus finality. But proof of work based protocols they do not support the consensus finality in principal.

(Refer Slide Time: 19:07)



PoW Consensus vs BFT Consensus

	PoW consensus	BFT consensus
Node identity management	**open, entirely decentralized**	permissioned, nodes need to know IDs of all other nodes
Consensus finality	no	**yes**
Scalability (no. of nodes)	**excellent (thousands of nodes)**	limited, not well explored (tested only up to $n \leq 20$ nodes)
Scalability (no. of clients)	**excellent (thousands of clients)**	**excellent (thousands of clients)**
Performance (throughput)	limited (due to possible of chain forks)	**excellent (tens of thousands tx/sec)**
Performance (latency)	high latency (due to multi-block confirmations)	**excellent (matches network latency)**
Power consumption	very poor (PoW wastes energy)	**good**
Tolerated power of an adversary	$\leq 25\%$ computing power	$\leq 33\%$ voting power
Network synchrony assumptions	physical clock timestamps (e.g., for block validity)	**none for consensus safety** (synchrony needed for liveness)
Correctness proofs	no	**yes**

Vukolić, Marko. "The quest for scalable blockchain fabric: Proof-of-work vs. BFT replication." *International Workshop on Open Problems in Network Security*. Springer, Cham, 2015.

3f+1

pBFT

Well, so based on this, so Vukolic also gives a nice comparison between this proof of work consensus mechanism and BFT based consensus mechanism under multiple parameters. So, let us look into that in briefly. So, in terms of node identity management the proof of work consensus protocol is open and entirely this centralized that means, anyone can join in the network and the nodes do not need to reveal their identity to other. So, it is open environment.

On the other hand the BFT consensus protocol they are applied on a permission to environment because it relies on the message passing architecture you need to know that who are the other backups in the system the primary needs to know that, who are the other backups in the system and every backup needs to know who are the other backups in the system as well as the primary. So, the identity of every node needs to be available to others. So here identity means in terms of message passing it may be like the IP address of that node, even if you do not know whether it is Bourbonnais. At least you know the IP address through which you need to communicate with that particular machine.

So, that is why it is for a closed environment the BFT consensus protocols are primary designed for a closed environment and we can apply it in the permission block chain settings. In terms of consensus finality as we have discussed just now that proof of work consensus does not support consensus finality, whereas BFT based consensus support consensus finality.

In terms of scalability, in terms of number of nodes proof of work power from very good it can support 1000 number of nodes whereas, BFT is limited. Now, although the research says do not have not explored the scalability of BFT based protocols practically in a practical aspect. So, most of the practical implementation of the BFT consensus protocols, they have been tested only up to some 20 number of nodes. So, the existing research people papers they have not went beyond that.

In terms of number of client support from the scalability aspect both of them are excellent. So, proof of work consensus can support a 1000s of clients and at the same time BFT can support 1000s of clients. In terms of performance like throughput like the transaction of second transactions per second that you are going to support proof of work on consensus is limited due to the possible of chain forks. So, you need to wait for certain amount of time whereas, the BFT consensus they perform very good. So, you can support tens of 1000s of transactions per second.

In terms of latency proof of work consensus has such a high latency because you need to solve that particular challenge based on the mining difficulty. So, the block commitment time depends on the mining difficulty, if your mining difficulty is high you have to wait for a large amount of time, but on the other hand BFT consensus performs very good in this particular aspect. So, it matches on the network latency.

So, it only depends on the network latency the consensus time. So, if you have n number of nodes and you need to do o of n cube message passing so the amount of time that you have to do for a o of n cube message passing just based on the network latency, within that time you will be able to commit a block and in general it is in the order of a few seconds.

In terms of power consumption the power consumption aspect of proof of work consensus is very poor because you have seen that proof of work waste huge amount of energy you are you are randomly generating the norms or just iterating over the norms to find out the require has value. And there is no guarantee that every minor will get a has value indeed only a handful of minors only 1 or 2 minors will be able to get the norms get the norms value and in that blockchain that info website the statistics that we have seen a few days back that nowadays most of the time only one minor or a mining pool that owns or that gets the correct block. So, all other mining pools who are doing the mining parallelly they are not able to successfully generate the particular block. So, a huge amount of power gets wasted. But in terms of power consumption again BFT consensus is very good because you have to just do a certain level of message passing.

Now, for an adversary the tolerating power is in case of poor consensus it is less than equal to 20 percent of the computing power for generating a hash. If you have more than 25 percent of the computing power with high probability at every different round that particular minor will win the case and that way that minor may have the ability to control the entire blockchain network. So, until an adversary have less than 25 percent of the total mining power that is required the system will work perfectly.

On the other hand for BFT consensus you require less than 33 percent of the voting power. So, if you remember that if there are f number of faulty nodes then you require 3 f plus 1 number of total number of nodes in case of your PBFT protocol which ensures that if you can have less than 33 percent of the voting power, you will be happy to get a consensus.

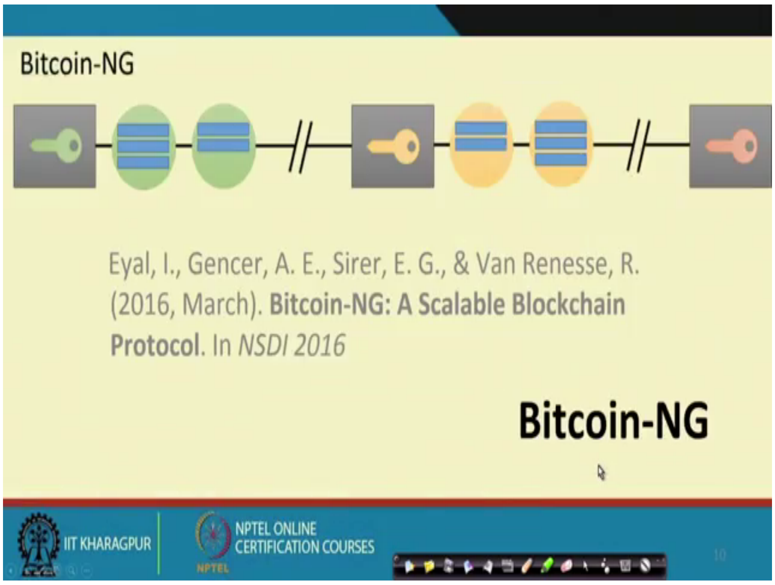
Now, in terms of network synchronic assumptions proof of work depends on the physical clock time stands for block validity. So, you need to have synchronisation among the physical clocks at different machines, otherwise if your machine clock is too old it may put a old time stamp to your block which may not get accepted by other nodes in the network. But in case of BFT consensus we do not have any constraint on terms of these network synchronic assumptions, but for safety you do not require synchrony in case of PBFT algorithm as we have seen but for lightness you require certain amount of synchrony.

So, if you remember the PBFT algorithm, PBFT algorithm uses synchrony for only for ensuring lightness, but for safety there is no such synchronic assumptions. On the same line if you remember the; if you remember that principle like in a pure asynchronous network it is impossible to reach in a consensus and if one node is faulty. So, with this impossibility principal we cannot have a system where the system is complete asynchronous or pure asynchronous still you are able to reach to the consensus.

So, that is why the PBFT algorithm as we discussed in the last class that it do not have any such assumption in terms of synchronisation assumption with respect to consensus safety but for liveness, it requires the synchronisation assumption.

For the correctness proof we have nice theoretical proofs for the PBFT algorithms and I encourage all of you to go to the PBFT paper to look into the proof. We have discussed during the lecture very briefly to make it simple. But whoever is more interested in knowing the distributed system concept and going in depth of that particular concept or the theoretical proof behind PBFT algorithm, you are encouraged to look into the PBFT paper. But for proof of work consensus we do not have any such theoretical proof. It says that proof of work will always be able to provide a good consensus in the system. And in it a proof of work is basically a probabilistic protocol.

(Refer Slide Time: 27:19)



Bitcoin-NG

Eyal, I., Gencer, A. E., Sirer, E. G., & Van Renesse, R. (2016, March). **Bitcoin-NG: A Scalable Blockchain Protocol**. In *NSDI 2016*

Bitcoin-NG

Well, so with this I will stop today’s lecture. In tomorrow’s lecture we will look into on enhancement over the bitcoin protocol which is called as the Bitcoin-NG. So, this

Bitcoin-NG is developed on the top of this proof of work mechanism to ensure a little bit more scalability in comparison with the standard base of work proof protocol. So, in the next class we will look onto the BFT protocol in detail.

Thank you all for attending to this lecture.