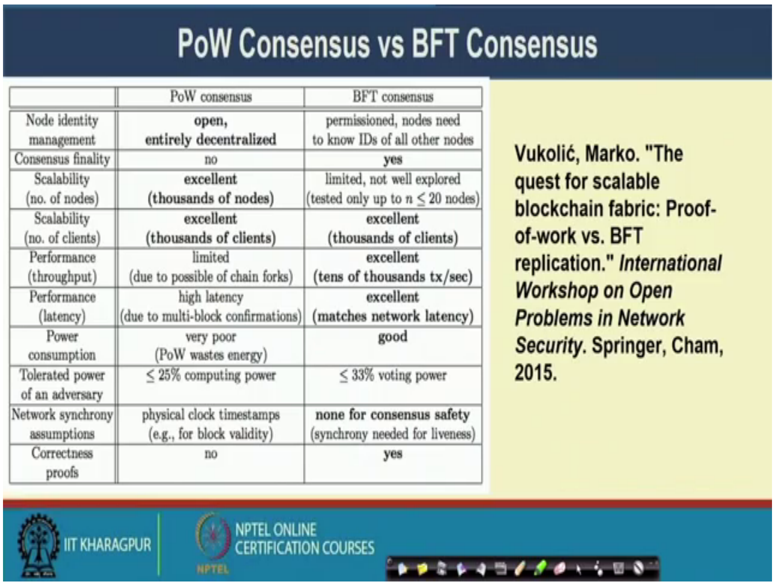
Blockchains Architecture, Design and Use Cases
Prof. Sandip Chakraborty
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 49
Research Aspects – II (Bitcoin - NG)

Welcome back on the course on Blockchain. So, in the last class, we are looking into the aspects of scalability, and we have looked into the comparison between proof of work and PBFT based protocols. So, we will start from there today and we will continue with different research aspects on the sense of consensus scalability, which will help you in derive for scalable consensus protocol for blockchains.

So, let us start with this Bitcoin-NG protocol, which is a successor of the standard bitcoin protocol by replacing the proof of what scalability with a more sophisticated work, which in it use as proof of work, but a modified version of it.

(Refer Slide Time: 01:08)



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.

So, in the last class, you are looking into the comparison between this proof of work and the BFT consensus protocol. And there we have seen that the other two major aspects, where the consensus scalability protocol differs; one is the scalability in terms of number of nodes, the second thing is the scalability in terms of number of clients. So, we have looked into that in terms of number of clients both proof of work and PBFT based protocols perform good. But, whenever you will look into the scalability in terms of number of nodes, this proof of consensus performs excellent, but this PBFT based consensus performs very bad or poorly.

On the other hand, if you look into the performance or throughput for this proof of work and the PBFT based protocol, so the proof of work base protocol their throughput is very limited. So, if you use the standard value of a block duration of 10 minutes between two subsequent blocks and a 1 MB a block size as suggested by the paper in Satoshi Nakamoto, so you will get a maximum of 7 to 8 transactions per second, and not more than that. Whereas, if you go to the PBFT based protocol, your scalability is much higher compared to proof of work.

Apart from these three aspects, there is another important aspect that we have looked into it is the consensus finality. So, in terms of consensus finality, we have seen that in case of proof of work based protocol, because it is a probabilistic protocol, where multiple miners can start mining the block together, and it may happen always that. More than one miners can mind the block subsequently. So, under that case, there is a possibility of having a duplicate path in the blockchain, so we call that duplicate path as a fork.

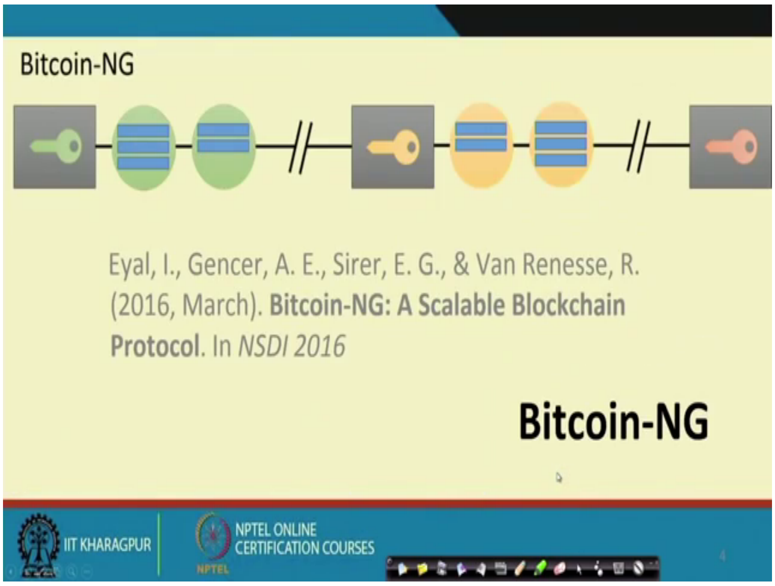
So, in generally, we avoid the fork, but the problem with this fork is that ultimately the system need to take care of those forks by identifying the forks say in a major change of a blockchain. And while removing the forks that is in data over it for the entire system, so that is why in terms of consensus finality you will never be able to say that well. The proof of work based protocol will provide the same order of transaction serializability always, so the transaction serializability depends on in what order the blocks are getting mine.

And then if two blocks are getting mine simultaneously, then you have to wait for the subsequent blocks to find out that which is the longest chain among all the forks which are there. So, because of that this consciences finality was a problem of your proof of work based protocol, where until you are mining subsequent blocks you will not be able to see your that the latest mine block is the final one, because there is a possibility that it may be a fork, and not a part of the main blockchain.

Now, to avoid that, in case of PBFT, whenever you are making a consensus by utilising these byzantine fault tolerance consensus protocol. So, this PBFT protocol always ensures that well, there is a single copy of the block, which is generated being generated by all the nodes to this agreement protocol the PBFT agreement protocol, at there is no fork in the system. So, you will always get a unique blockchain for the end to end system, so that will ensure you that well your consensus a protocol has provided total order of the transactions, which are there in this entire chain.

Now, from this particular point, we have seen that there are two major shortcomings of proof of work based consensus protocol. So, the first shortcoming is in the terms of consensus finality, where it does not supports this consensus finality. And the second shortcoming is in terms of performance, where the transaction throughput is very limited. So, we want to have a improved version of this proof of work based protocol, which will provide similar type of a system similar type of blockchain system, but it will be able to support more transaction throughput, and we will also try to provide transaction finality to the end system that we are going to produce.

(Refer Slide Time: 05:41)



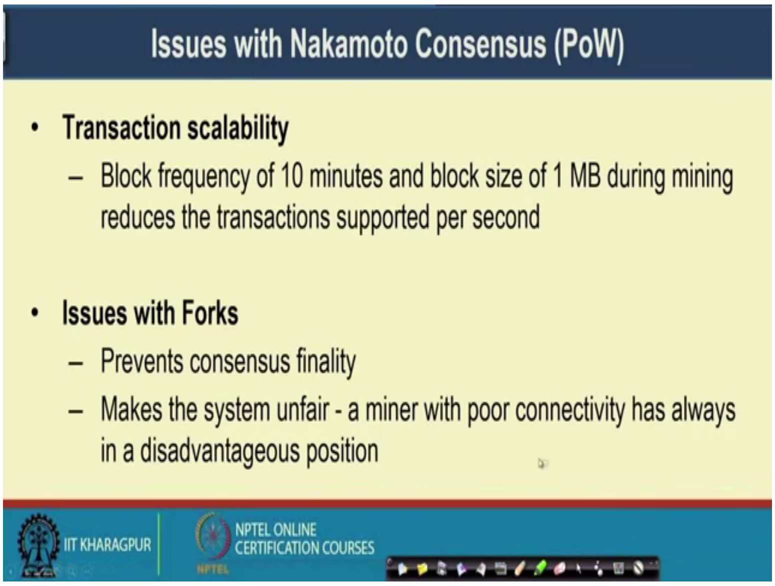
Bitcoin-NG

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

Bitcoin-NG

So, to this end let us look into one protocol or one mechanism for supporting the shortcomings of bitcoin based protocol, where the primary objective is to provide the transaction throughput the better transaction throughput. And along with that, the objective would be to avoid the forks as much as possible, because forks are problematic for the system perspective. So, we will look into this bitcoin protocol in details Bitcoin-NG protocol in details. So, this Bitcoin-NG protocol it was the published in NSDI 2016 by Eyal, Gencer and Van Renesse in 2016, march in as a part of the NSDI 2016 consensus proceeding.

(Refer Slide Time: 06:31)



Issues with Nakamoto Consensus (PoW)

- Transaction scalability
  - Block frequency of 10 minutes and block size of 1 MB during mining reduces the transactions supported per second
- Issues with Forks
  - Prevents consensus finality
  - Makes the system unfair - a miner with poor connectivity has always in a disadvantageous position

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES

So, to start with this particular modification in the bitcoin consensus protocol, so we look into the details or the problems, which were there in Nakamoto consensus. So, this proof of work based mechanism in literature people also refer it to the Nakamoto consensus so, when you use the standard values, which have been used in the proof of work based protocol.

So, the first problem was transaction scalability. So, as I have mentioned that there are two magic parameters in Nakamoto consensus. First is the block frequency, so you can generate a block at every 10 minutes, so that is add of estimation of how frequent you can generate a block based on the mining difficulty, so the mining difficulty actually controls the average frequency of block generation, so that is one magic parameter. The second magic parameter is the block size. So, the block size has been limited to 1 megabyte in case of Nakamoto consensus.

So, there are multiple debates, which are going on to increase the block size, but increasing the block size also has its side effects in terms of forks. So, if you just increase the block size from 1 MB to 4 MB to 8 MB and so on, so a fork will collect or it provide the incorrect block to the system of larger size, so that is a kind of more costlier from the system perspective. So, increasing the block size is not at all are very good solution rather will try to look into the problem from some different aspects.

The second issue as I have mentioned is the issues with forks. So, the fork first of all it prevents the consensus finality. The second thing is that it makes the system unfair, so a miner with poor connectivity has always disadvantageous position because of this fork. So, if a miner has a poor connectivity, it may happened that it has generated a new block, and assume a scenario when the miner has sufficient power, but the problem is at the network site, so the connectivity is poor.

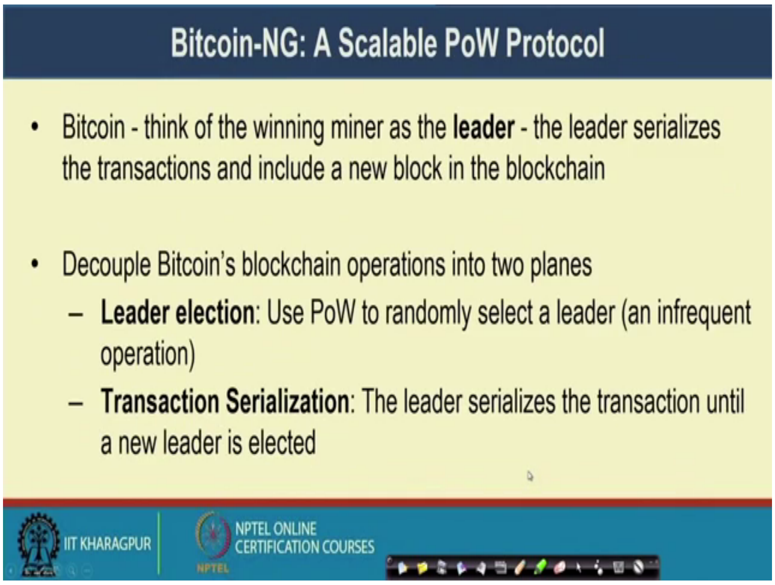
So, the miner the newly generated block from that miner, it is taking time sometime to propagate to the network. If it happens, then it may happen that well in a given blockchain, so a new miner has say generated this new block. So, this is my new block N. Now, by the time, this block gets propagated to the system, it may happen that other miners are generated, another set of blocks which provide the longest chain in the system.

Now, if that is the case, then obviously this longest chain will be accepted by the system. And this particular block which has been generated by miner because of this poor network connectivity, the block will not get propagated to the system. So, the miner is always is (Refer Time: 09:25) position, if you are using this proof of work based systems, where there is a possibility of fork can fork can always make the system problematic.

So, there are these two major issues, which we are trying to solve with the help of Bitcoin-NG. So, the first problem is in terms of the consensus scalability, in terms of consensus throughput that means the number of transactions that you will be able to support per second or per unit time. The second is with the fork that fork if you have multiple forks in the system, then that particular forks hampers the fairness of the system.

So, a miner if it is has, if it is in a part of the globe, where the network connectivity is poor. So, those miners are always in a disadvantages position. And this basic philosophy of bitcoin that or proof of work that you will be able to evenly distribute the hash power across the globe that gets valuated here. So, let us see that how Bitcoin-NG tackles this two problems.

(Refer Slide Time: 10:34)



Bitcoin-NG: A Scalable PoW Protocol

- Bitcoin - think of the winning miner as the **leader** - the leader serializes the transactions and include a new block in the blockchain
- Decouple Bitcoin's blockchain operations into two planes
  - **Leader election**: Use PoW to randomly select a leader (an infrequent operation)
  - **Transaction Serialization**: The leader serializes the transaction until a new leader is elected

So, this Bitcoin-NG, we call it as a scalable proof of work based protocol. So, this Bitcoin-NG is if you just look into the standard bitcoin protocol, in the standard bitcoin protocol you have a miner, now that miner the task of that miner is to generate a new block. So, to solve a puzzle which has been provided by the network, and then to mine a new block.

So, you can just think of that miner as the leader of the system. So, here in bitcoin, we also have this inherit concept of leader similar to the PBFT type of consensus protocols that we have seen, where we definitely try to choose a leader who will do the transaction serializability. But, in case of bitcoin as well, we have this concept of leader implicitly they are inside the protocol, whose task is to mine a new block. So, one of the miner, who is able to solve the puzzle that miner works like a leader. So, you can think of the winning miner as the leader, where the task of the leader is to serialize the transactions, and include a new block in the blockchain.

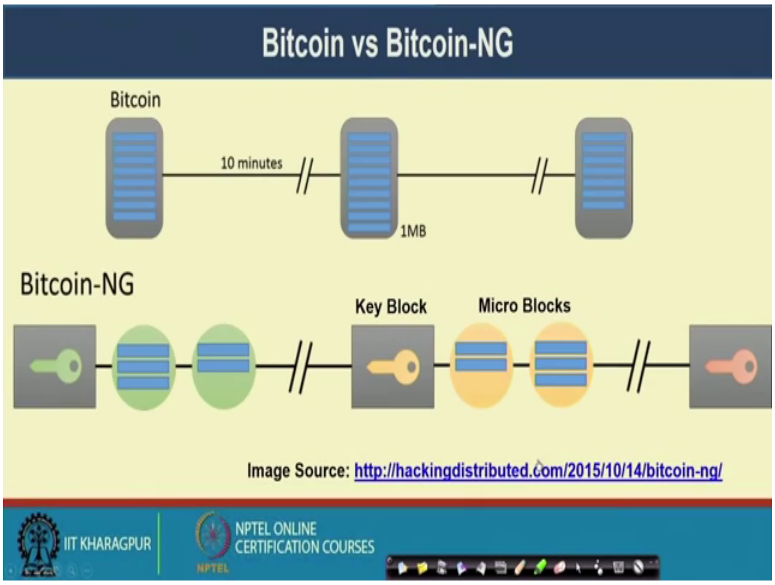
Now, the first step of bitcoin-NG is to decouple the two functionalities from bitcoin proof of work for the miners. So, the first functionality is the leader election to find out that who is going to be a leader in a particular round. And the second is the serializability of the transaction. So, the major problem if you just try to think of from the bitcoin proof of work perspective that why bitcoin proof of work does not provide good scalability, because at every round, you are electing a different leader. So, at every round, you want the new leader to come or a new miner to get the problem or the challenge solved based on the mining difficulty, who will be able to serialize the transactions, so that means, at every individual round, you are generating a new miner, and the task of that miner would be to serialize the transactions.

So, because of this delay in the leader election, which is getting introduced in the system, what is happening that the transactions they are remaining in the queue. So, until the new leader is getting selected or a new miner is winning the mining puzzle up to that time, you are not able to serialize the transactions, and at the transactions to the existing blockchain. So that is actually limiting the performance of proof of work to the mention number that means, the 7 to 8 transactions per second, because your mining difficulty is saying that well. On average at every 10 minutes, you will get a new leader, who will do the transaction serialisation, so that is why the first principle that was introduced in bitcoin-NG was to decouple this two functionalities; the decouple the functionality of leader election and the transactions realizability.

So, the idea here is that you use the proof of work to randomly select a leader, but whenever you are using proof of work to select a leader, this particular operation is an in frequent operation in Bitcoin-NG. So, you do not select leader at every individual round rather you select the leader, then give some scope to that leader to do the transaction serializability for some subsequent round. And then after few rounds, you elect the new leader. So, this leader election is an in frequent operation, which will help you to improve the performance of the system.

The second thing is this transaction serialization. So, the leader will serialize the transaction until a new leader is elected, which is a more frequent operation. So, whenever the leader will get subsequent or sufficient number of transactions from the clients, whenever it will be able to hear the transaction sufficient number of transaction, it can serializest the transaction, and it will be able to add the transaction to the existing blockchain.

(Refer Slide Time: 14:52)



Bitcoin vs Bitcoin-NG

Image Source: http://hackingdistributed.com/2015/10/14/bitcoin-ng/

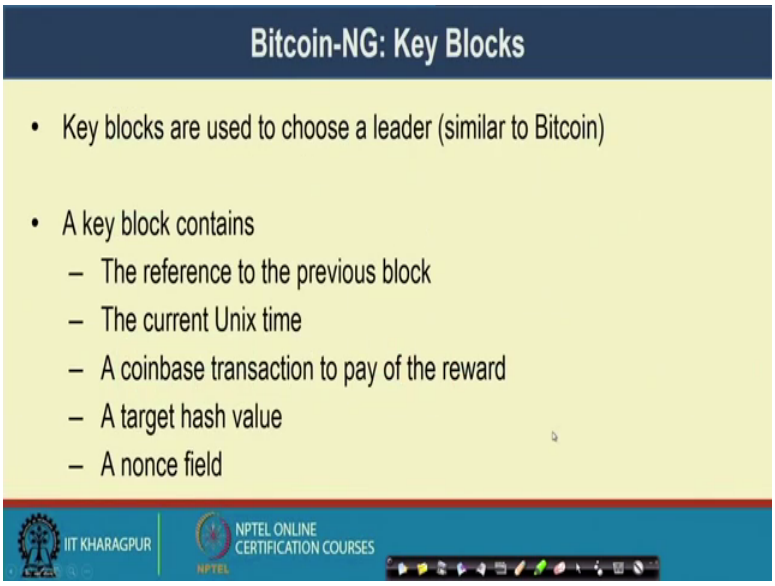
Now, with this idea, the entire architecture of Bitcoin-NG is something like this. So, in if you look into the standard bitcoin, so at every 10 minutes, you are electing a new leader, and the task of that new leader will be adds up this new blocks. And at every almost at every 10 minutes, you are electing a new leader that means, the miner who is able to solve the puzzle, and the miner will add up this new blocks in the system.

Now, in case of Bitcoin-NG, we have two different type of blocks. One type of block, we call them as the key blocks. So, why the name is key block, I will discuss that within a couple of minutes. And there are subsequent blocks between two key blocks, which we call as the microblocks. Now, these key blocks are generated by the proof of work mechanism.

And the ideas is here is the that in every proof of work round, you generate one key block, and that key block will contain the public key of a the miner, who has been able to solve the challenge. That means, that miner has wound round for solving the proof of work challenge, and it will work like a leader in for some duration or for some amount of time. Now, that particular leader will be able to generate multiple such blocks, which we call as the microblocks. So, these microblocks will contain the set of transactions inside.

So, the now the microblocks contains the transactions. So, a leader can generate multiple such microblocks which will be encrypted to the public to the private key of this leader. So, the leader will encrypt all these microblocks with his or power private key, and others will have the corresponding public key, so that public key can be used to verify this microblocks that this microblocks has been in the generated by the leader that has been elected by the proof of work mechanism. Now, let us look into the details of this key block and the microblock in case of Bitcoin-NG that what are they are inside the blocks.

(Refer Slide Time: 17:03)



Bitcoin-NG: Key Blocks

- Key blocks are used to choose a leader (similar to Bitcoin)
- A key block contains
  - The reference to the previous block
  - The current Unix time
  - A coinbase transaction to pay of the reward
  - A target hash value
  - A nonce field

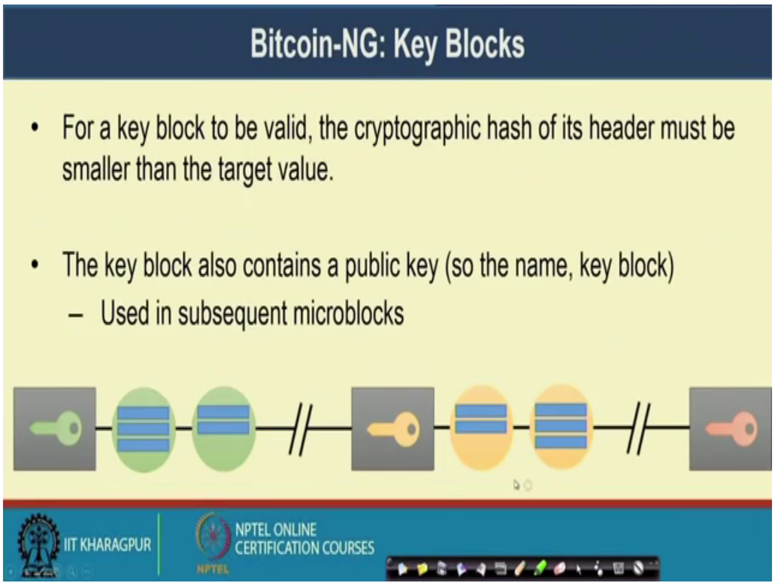
IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
NPTEL

So, the first block is the key blocks. So, these key blocks are used to choose a leader, which is similar to the bitcoin protocol. Now, a key block it contains this parameter similar to a standard block in case of a bitcoin like the difference to the previous block that has pointed, then the current Unix time just to have a timestamp, a coinbase transaction to pay the reward. So, if you remember that inside a block, we also put a special transaction that actually tells about that how much reward is provided to that particular miner. So, this particular transaction we call it as a coin base transaction in terms of bitcoin. So, it contains this coinbase transaction to pay the reward to the miner.

The target hash value, which is there and the nonce field. So, similar to that standard bitcoin protocol, the idea of that that there is a challenge. So, you have a target hash value that you have to get. So, your target is to get the hash value either equal to this or something less than this target hash value just like the standard bitcoin protocol. And every miner, periodically they will try to find out the nonce.

Now, the task is similar to bitcoin proof of work that you find out the nonce, which will actually help you to get a hash value equal to a list and the target hash value. And if you are able to find out that nonce value then you have been elected as a leader. So, you can generate the corresponding key block, and then you can generate multiple microblocks or subsequent microblocks, which will contain the serialize transactions that you have over heard, so that is the broad idea of a key block the contents of a block.

(Refer Slide Time: 18:56)



Bitcoin-NG: Key Blocks

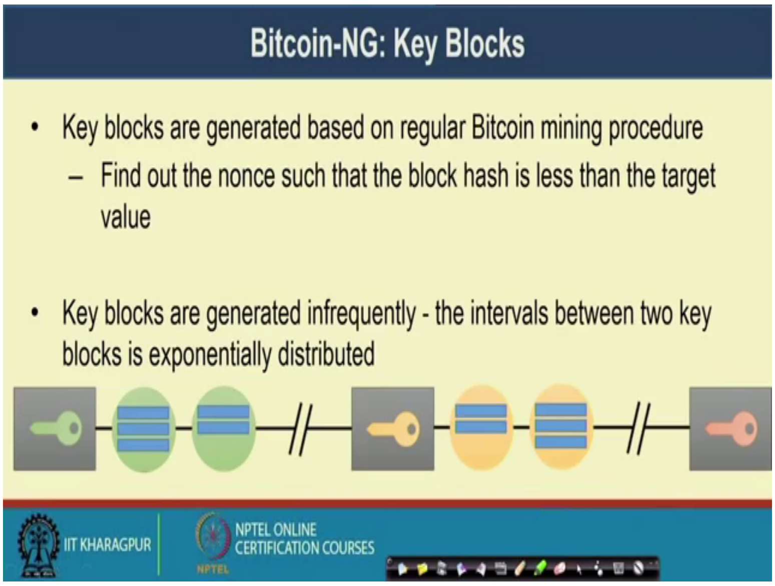
- For a key block to be valid, the cryptographic hash of its header must be smaller than the target value.
- The key block also contains a public key (so the name, key block)
  - Used in subsequent microblocks

And then you have a various microblocks, which are there. So, now in case of Bitcoin-NG for a key block to be valid, the cryptographic has of its header must be smaller than the target value, which is similar to the bitcoin proof of work protocol. So, you need to find out the nonce, which will provide you this one. And the key block it also contains a public key that is why you have the name key block, which is used in the subsequent microblocks.

So, the key that you are generating in this key block, so this key which is basically the public key of the miner, which was able to solve the challenge based on the proof of work mining procedure. And this public keys used in the subsequent mining a microblock to validate that this microblocks has been generated by this particular miner, so that way you will be able to validate that well. All these microblocks are actually generated by this particular miner, which has the public key in this block. And so, these are the kind of valid blocks and this microblocks contains all the transactions.

And at the end of one key block round, another round of key block generation happens in Bitcoin-NG, where again challenge is given and the miners they are trying to generate the next key block. And the miner who will be able to generate this next key block, it will be able to use the subsequent microblocks to serialize the transactions, so that is broad ideas of this Bitcoin-NG protocol.

(Refer Slide Time: 20:45)

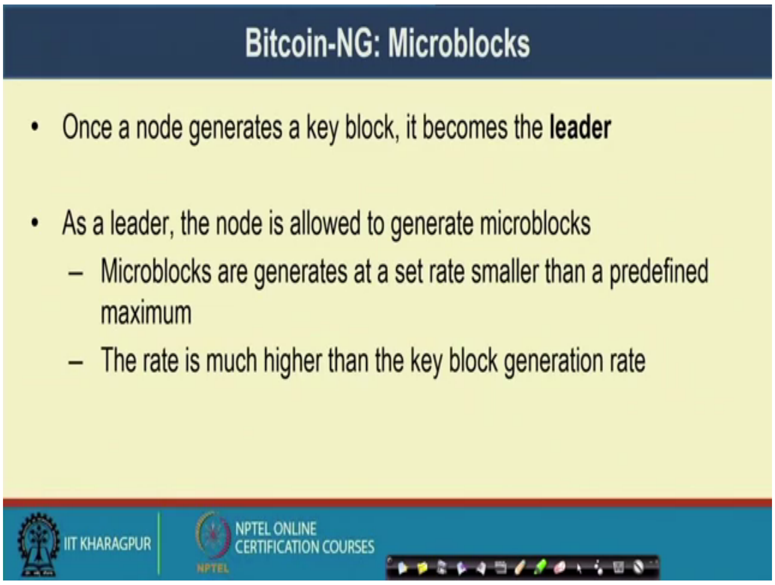


Bitcoin-NG: Key Blocks

- Key blocks are generated based on regular Bitcoin mining procedure
  - Find out the nonce such that the block hash is less than the target value
- Key blocks are generated infrequently - the intervals between two key blocks is exponentially distributed

So, this key blocks they are generated based on regular bitcoin mining procedure that means, that every 10 minutes you can generate the key block. And the idea is that to find out the nonce, such that the block has is less than the target value. And this key blocks we generate this key blocks infrequently, so the intervals between two key block is in general exponentially distributed. And average duration is just like 10 minutes similar to your standard bitcoin protocol.

(Refer Slide Time: 21:17)



Bitcoin-NG: Microblocks

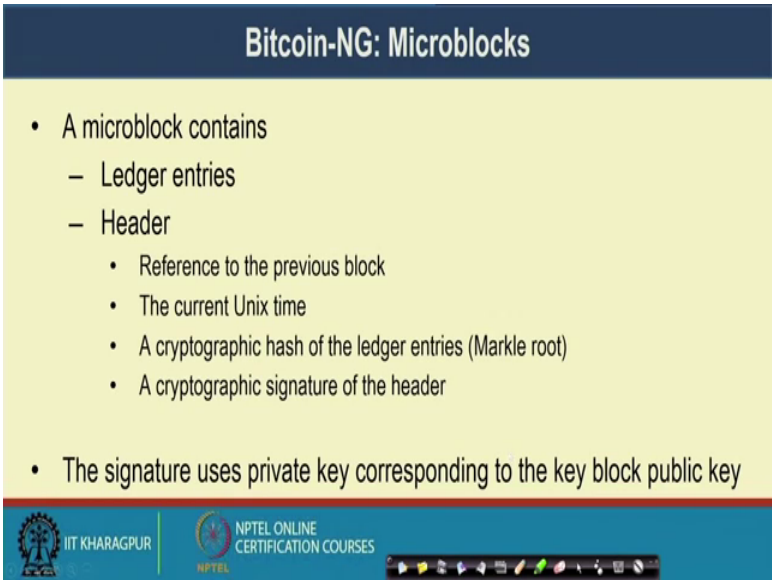
- Once a node generates a key block, it becomes the **leader**
- As a leader, the node is allowed to generate microblocks
  - Microblocks are generates at a set rate smaller than a predefined maximum
  - The rate is much higher than the key block generation rate

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
NPTEL

So, coming to the microblock so, once a node generates a key block, it becomes the leader. And as a leader, the node is allowed to generate the microblocks. So, this microblocks are generated at a set of at a rate smaller than a predefined maximum. So, you have a certain predefined rate, so you generate the microblocks at a rate smaller than that predefined rate. And this rate of generating microblocks, it is much higher compared to the key block generation rate.

So, if you generate the key block at a say every 10 minutes, then it is like that at every minute you can generate the microblock. That means, within 1 minute whatever transaction has been receipt to that particular miner which is working as a leader of this round, so the miner can generate a new microblock with the serialise transactions and add it with the existing blockchain, so that way you will be able to generate the microblocks more frequently. And that is why you will be able to add up more number of transactions in the system, which will help you to increase the scalability of this entire system.

(Refer Slide Time: 22:32)



Bitcoin-NG: Microblocks

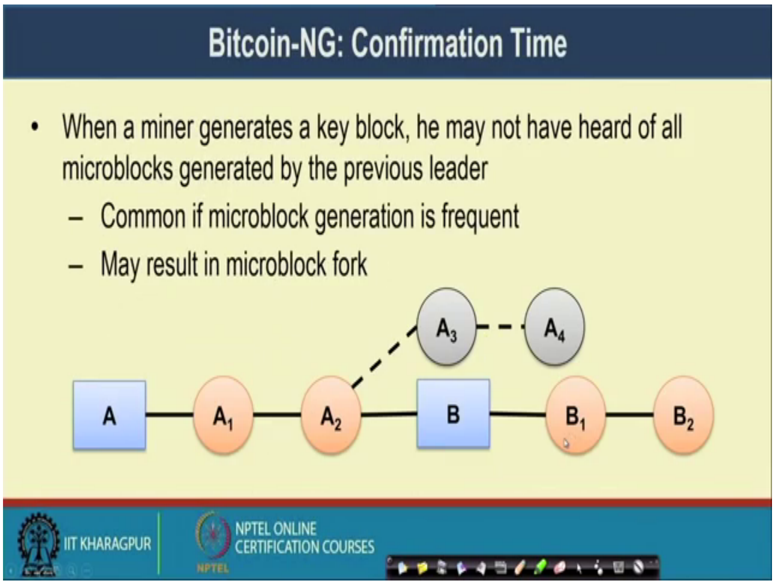
- A microblock contains
  - Ledger entries
  - Header
    - Reference to the previous block
    - The current Unix time
    - A cryptographic hash of the ledger entries (Markle root)
    - A cryptographic signature of the header
- The signature uses private key corresponding to the key block public key

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES

Well, so a microblock contains the ledger entries, microblock has its own header. So, this header is the contains the reference to the previous block, previous microblock or key block whichever is there, the current Unix time, cryptographic hash of the ledger entries that means, the Markle root, and a cryptographic signature of the header. So, this cryptographic signature is something which is based on the private key private key corresponds to the key block after public key.

So, this key block contains the public key and all the microblocks, they say this is the microblock header. So, this microblock header contains the signature. So, this signature is corresponds to the public key, which was there. So, this signature is generated based on the standard digital signature mechanism using the private key corresponds to this public key, so that means, as a verifier, you will be able to verify this signature that well. This signature is coming from the miner, who has generated this particular key block, so that way this microblock which have been generated that has been generated by the miner corresponds to this public key. So, this is a valid microblock.

(Refer Slide Time: 24:10)



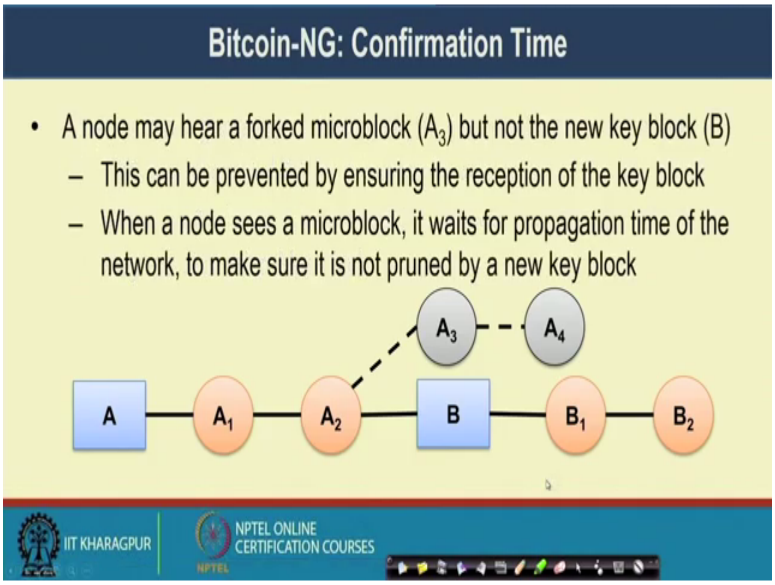
Bitcoin-NG: Confirmation Time

- When a miner generates a key block, he may not have heard of all microblocks generated by the previous leader
  - Common if microblock generation is frequent
  - May result in microblock fork

Well now, there is a one issue in Bitcoin-NG, where there is a possibility of having a fork. So, the problem is something like this that when a miner generates the key block, he may not have heard of all the microblocks, which has been generated by the previously leader. So, this example say A was a miner, who has generated this key block. And at this time instance another miner B, it has generated this key block. Now, it may happen that well, when B has generated this key block during that time, it has not heard about all the microblocks, which has been generated by node A, so that way what may happen that well, certain microblocks may become a fork.

So, B can generate the key block, B has heard up to say A, so whenever B has heard up to A. So, B update this key block, and B connect this key block with A 2, and if B connects this A block key block to A 2. And after that if we talks about the microblocks A 3 and A 4, so this microblocks A 3 and A 4 becomes a fork a microblock fork whatever you tell it, because B can then subsequent adds up to its own microblocks B 1 and B 2. So, this kind of things are common, if microblock generations are frequent. So, if you are frequently generating a microblock, then it may happen that well. You have generated a microblock that is that is not heard by B, and this may result in a microblock fork.

(Refer Slide Time: 26:04)

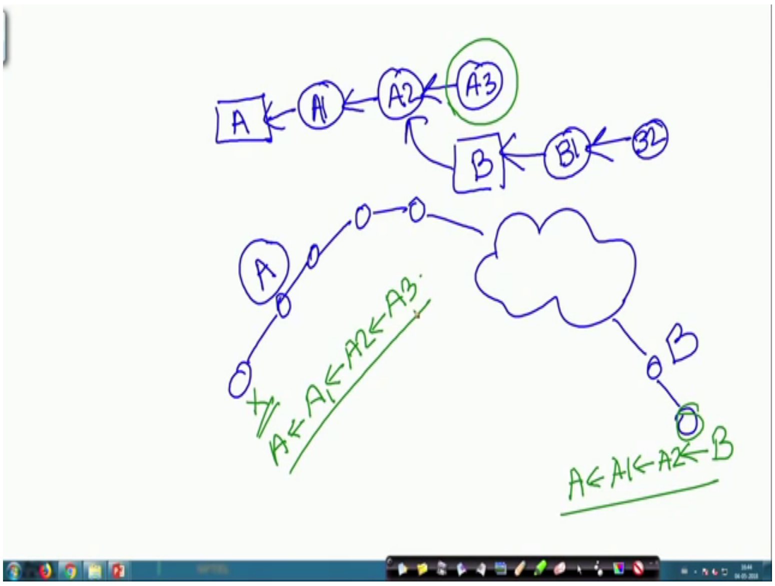


Bitcoin-NG: Confirmation Time

- A node may hear a forked microblock (A_3) but not the new key block (B)
  - This can be prevented by ensuring the reception of the key block
  - When a node sees a microblock, it waits for propagation time of the network, to make sure it is not pruned by a new key block

Now, to prevent this, we apply this mechanism in Bitcoin-NG. So, node may here a fork microblock, but not the new key block, so that is our problem. So, it may happen that well a node it has heard about A 3, because maybe it is closer to the miner A, so it has heard this new microblock, but it has not heard the new key block. So, the key block is generated.

(Refer Slide Time: 26:34)



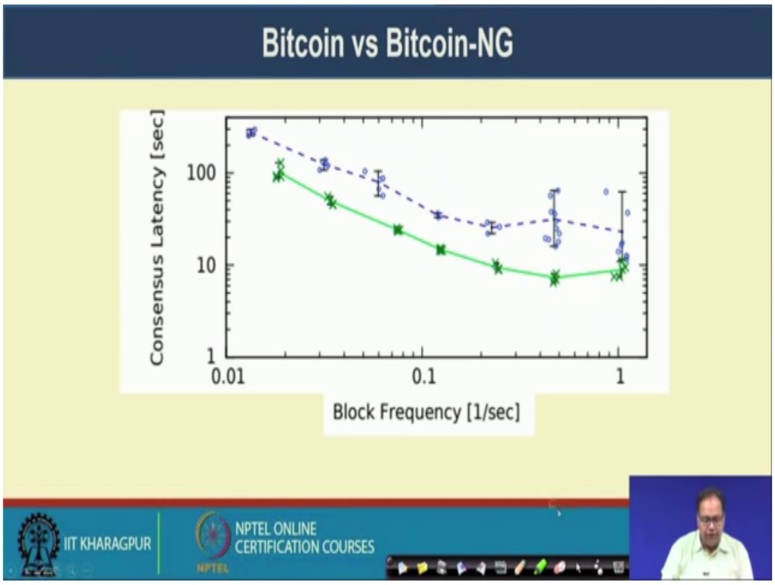
So, you can just think of a random topology of something like this, so you have multiple nodes in between in the network, and then again you have this nodes. Now, say this is node A, and this is node B. Now, A has generated a key block as I have shown in the example. So, A has generated a key block and then A is generating multiple microblock A 1, A 2, A 3 and so on. Now, say B has generated one key block, say it is B, and it is pointing to A. Now, if this is the case, and then we start generating the corresponding microblocks B 1, B 2 and so on. So, in this case this block A 3 becomes a fork.

Now, if you just think about this node say X, which is much closer to A, so it may happen that this node has heard this A 3, so it connects A 3 here. And by that time, it has not heard of the new key block B. So, in this case, there may be a inconsistency, which is there in the network. So, to this particular node, the structure can be A followed by A is the key block followed by A 1, A 2, A 3 like this. Whereas, the node here for it, the blockchain can be A, A 1, A 2, and then B. So, to avoid this inconsistency in the network, so you can have this kind of inconsistency in the network.

Now, to avoid this kind of inconsistency, the idea is something like this that when a node sees a microblock, it waits for the propagation time of the network, to make sure that it is not pruned by the new key block, so that means, whenever you are hearing a new microblock, then you wait for the propagation the maximum propagation time in the network.

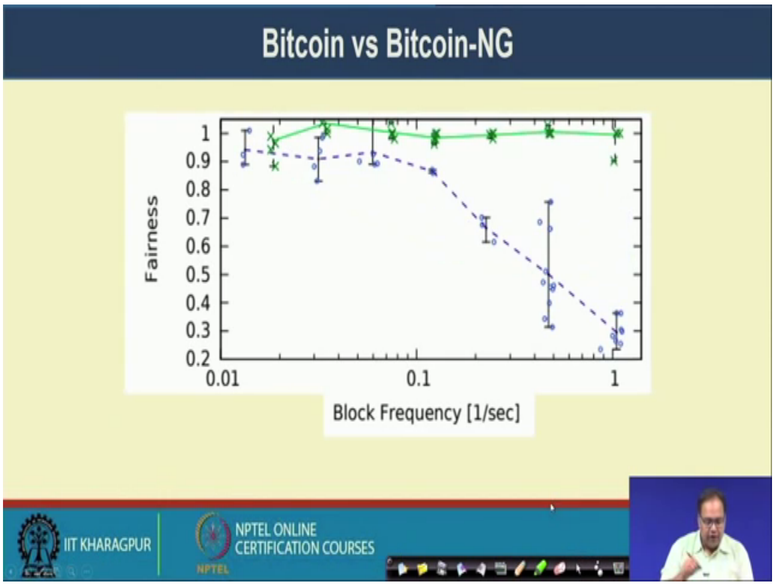
If you wait for that time, and if you are not having any key block, if you are not hearing any key block that means, no key block has been generated. If any key block has been generated in the network, by that time obviously, we will be able to hear the key block, so that means, whenever you are hearing a new microblock before adding it to your local blockchain, you wait for the time equal to the propagation delay the maximum propagation delay of the network, so that way you will basically tackle this particular problem.

(Refer Slide Time: 29:27)


Bitcoin vs Bitcoin-NG

So, let us look in to the performance of a Bitcoin-NG in comparison to bitcoin. So, this blue line is the performance for bitcoin whereas, so this is the performance for bitcoin whereas, this is the performance for Bitcoin-NG. So, here you see that if you look into this consensus latency that means how much time it is taking to commit a particular block in the system. So, you observe that the commitment latency, so this is with respect to block frequency. So, the commitment latency for the Bitcoin-NG is significantly less compared to bitcoin.

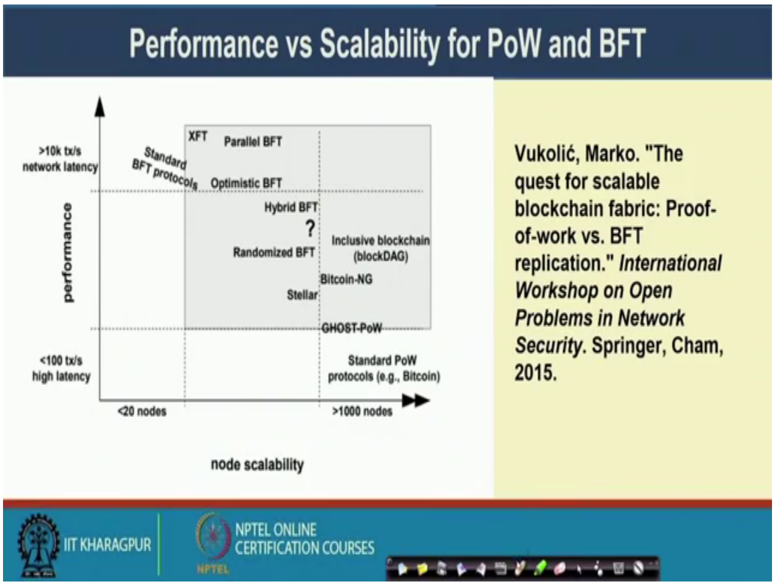
(Refer Slide Time: 30:25)


Bitcoin vs Bitcoin-NG

Similarly, if we observe the fairness, which was another parameter, so what we have looked into earlier that the fairness get sample because of the forks in the system. Now, here in case of Bitcoin-NG, you can think of that the forks are very infrequent. Because, whenever you have a new leader, which is generating multiple microblocks, during that time you do not have a possibility of forks. So, you only have a possibility of fork in the level of microblock. So, this key block forking is rare in case of Bitcoin-NG.

So, by doing that we can see that the fairness measure for a Bitcoin-NG is significantly higher compared to the standard bitcoin. So, the standard bitcoin as you increase the block frequency, your fairness gets drop significantly. But, your Bitcoin-NG protocol, it always maintains a fairness value, which is close to one that means, it always try to maintain a fairness in the system by avoiding forks in the in the consensus protocol, so that is all about Bitcoin-NG.

(Refer Slide Time: 31:40)



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.

Now this particular diagram, we looked into the last class. From these Vukolic paper, where he has tried to place different kind of consensus protocol in terms of node scalability and in terms of performance. And you have seen that the standard proof of work protocol falls here in this coordinate, whereas the standard BFT protocols falls here. So, our target was to build something, which will work in between these two, where this standard proof of work protocol, they have high scalability in terms of number of nodes, but they have low scalability in terms of performance that means, the transaction per second it can support.

So, if you look into the Bitcoin-NG protocol, Bitcoin-NG protocol falls somewhere here. So, it was able to improve the performance compared to the standard proof of work mechanism, but it is able to provide scalability, node scalability similar to the standard proof of work, so that is all about the Bitcoin-NG protocol. In the next class, we will come up with the limitations of this Bitcoin-NG, and another improvement on top of Bitcoin-NG consensus protocol, which will be able to further scale of the system in terms of performance and node scalability.

So, thank you all for attending this class.