Blockchains Architecture, Design and Use Cases

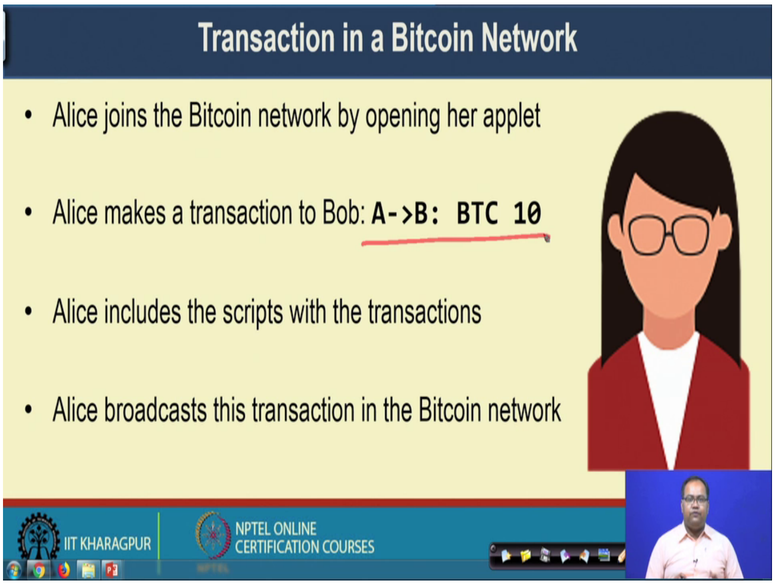
Prof. Sandip Chakraborty
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture – 09
Bitcoin Basics – III

So, welcome back to the course on Blockchain Technologies. So, in the last lecture we have discussed about that a whenever a new node comes to join for a bitcoin network how that a new node can join in a existing bitcoin network and then get updated the most recent copy of the blockchain.

Now, in this particular lecture we will look into that how the node can start initiating the transaction and how the transaction gets committed in the current blockchain that a network has.

(Refer Slide Time: 00:54)



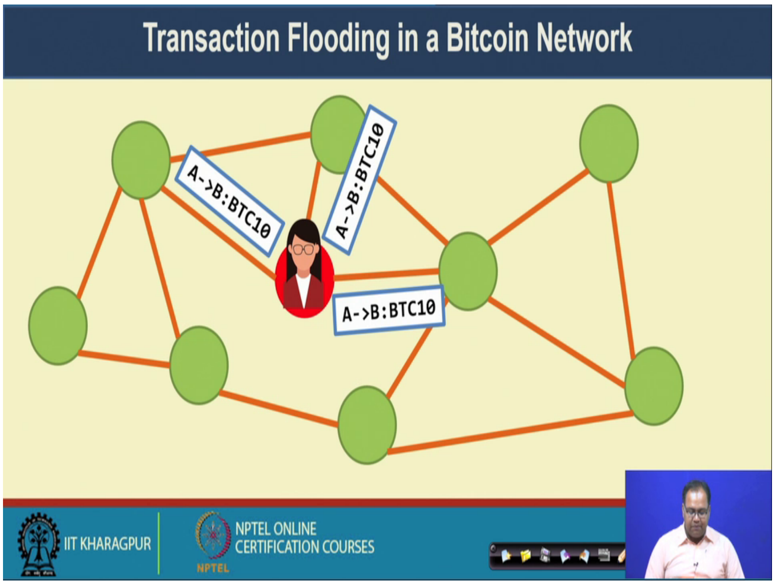
Transaction in a Bitcoin Network

- Alice joins the Bitcoin network by opening her applet
- Alice makes a transaction to Bob: `A->B: BTC 10`
- Alice includes the scripts with the transactions
- Alice broadcasts this transaction in the Bitcoin network

So, the transaction in a bitcoin network it happens in this particular way. So, after Alice joins the bitcoin network after that joining by opening her applet Alice can initiate a transaction to Bob. So, this is a sample transaction that a Alice initiates. So, this a sample transaction that Alice initiates.

So, once a Alice constructed this transaction to Bob with certain bitcoin say 10 bitcoin. Alice includes the scripts the inputs scripts and outputs scripts with this transaction to validate the authority of this particular transaction. And once this transaction messages get a computed constructed then Alice broadcast this transaction information in the bitcoin network. So, let us a look into the way this things are getting broadcasted.

(Refer Slide Time: 01:48)

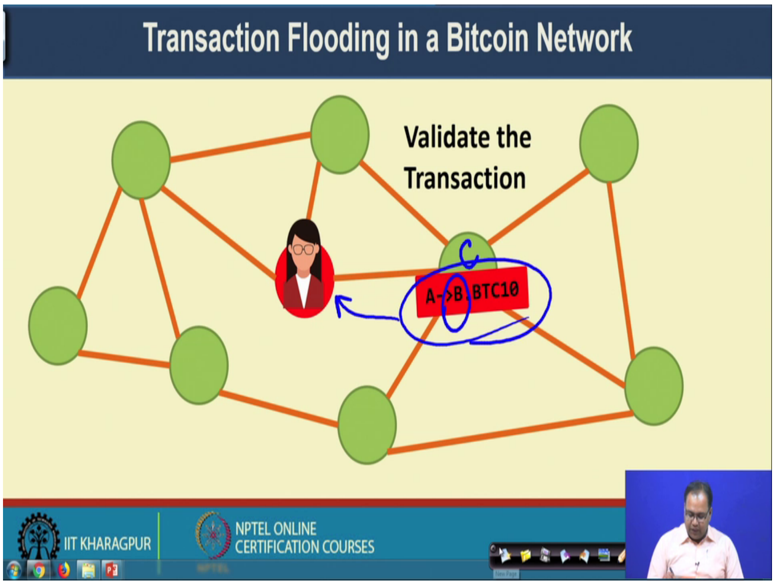


Transaction Flooding in a Bitcoin Network

So, when Alice joins the network after joining the network the initial step is to get the most recent copy of the blockchain from that we are neighbours, after getting the most recent copy of the blockchain from that we are neighbours Alice can thought initiating the transactions. So, once Alice initiate a transaction. So, see basically broadcast this transaction to her peers. So, the transaction get broadcasted in the peer network.

So, once we are making a broadcast of the transaction in the peer network every of node in the peer network they can receive this transaction, and they can validate whether the transaction is a valid transactions or not by looking into the scripts by executing the scripts.

(Refer Slide Time: 02:41)



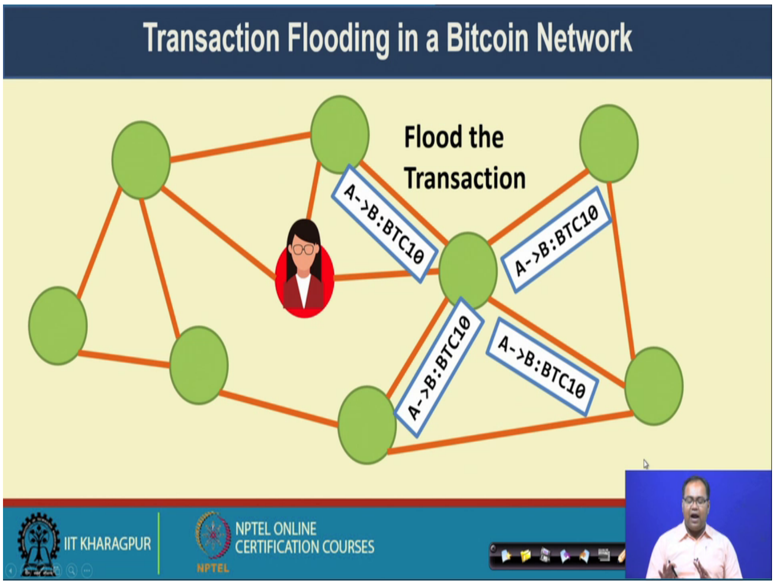
Transaction Flooding in a Bitcoin Network

Validate the Transaction

So, the nodes individual nodes they basically validate the transaction that is their first tasks, through validate the transaction what you can do. So, there are multiple algorithm to validate a transaction. So, to validate a transactions you can basically first run the input script and output script which is associated to it with your bitcoin transactions, by executing that script you can find out whether Alice is actually initiated the transactions or not. Well, in that particular transactions a transactions looks like from Alice to Bob in this transactions say if this node is not Bob say this node is something like C, Charlie.

So, Charlie will be able to validate this particular transaction, but because the destination address is Bob's address. So, Charlie will not be able to accepts this transactions or include this bitcoin in his wallet. Only Bob will be able to include this bitcoin in his wallet and will be able to use those bitcoin for further transactions. But any intermediate node they can run the scripts, they can look into the scripts, and by looking into the scripts by executing the scripts they can validate weather whether this particular transaction is actually originated from Alice or not.

(Refer Slide Time: 04:05)

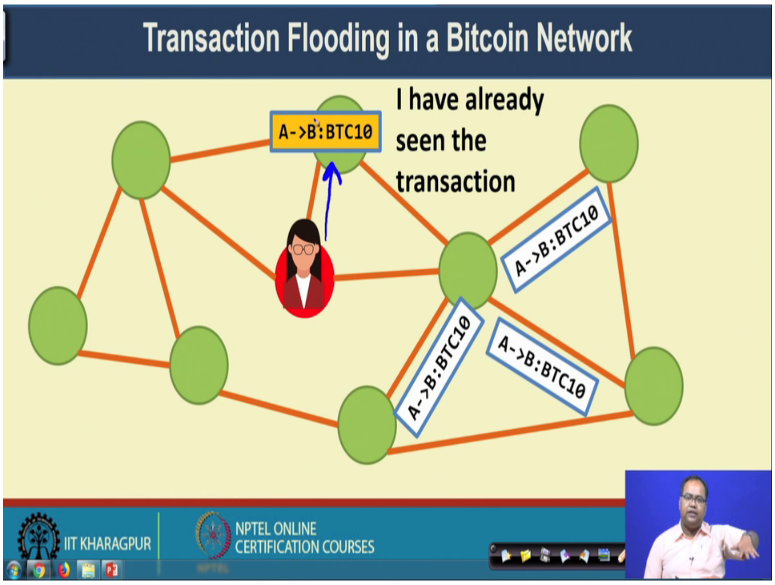


Transaction Flooding in a Bitcoin Network

Now, after doing this validation every node they again flood the transactions in the network. So, that way every transaction is actually getting propagated in the entire bitcoin network and a everyone in the entire bitcoin network they will be able to see all the transactions if they are online during that time and they will be able to validate the transactions. So, that is the interesting concept behind the bitcoin mining that the miners will also be able to receive all the transactions which are getting propagated in the bitcoin network and they will be able to construct a block with the help of those transactions.

So, let us look into that particular mining procedure in further details.

(Refer Slide Time: 04:48)



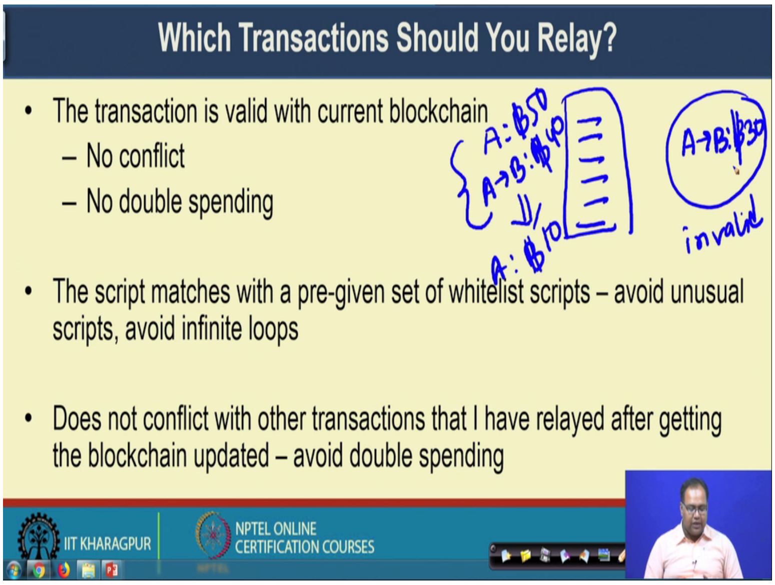
Transaction Flooding in a Bitcoin Network

I have already seen the transaction

So, whenever you are flooding the transactions during that time it may happen that this particular node it has already seen the transactions. If the node has already seen the transaction they need does not consider this transactions any further it simply drops that transactions or it do not do a rebroadcast of that transactions. So, for this particular node initially it had already received the transactions directly from Alice. So, during that time it had already observed these transactions, so every node may also maintains a list of a fast transactions that the node has observed and flooded.

So, if it receives any such kind of duplicate transactions it do not flood the transactions for a second time. So, it is this avoids having kind of the entire bitcoin network getting clogged with this kind of flooding message. We basically limits the amount of flooding in the bitcoin network, that you are not unnecessary doing any flooding of the transactions.

(Refer Slide Time: 05:58)



Which Transactions Should You Relay?

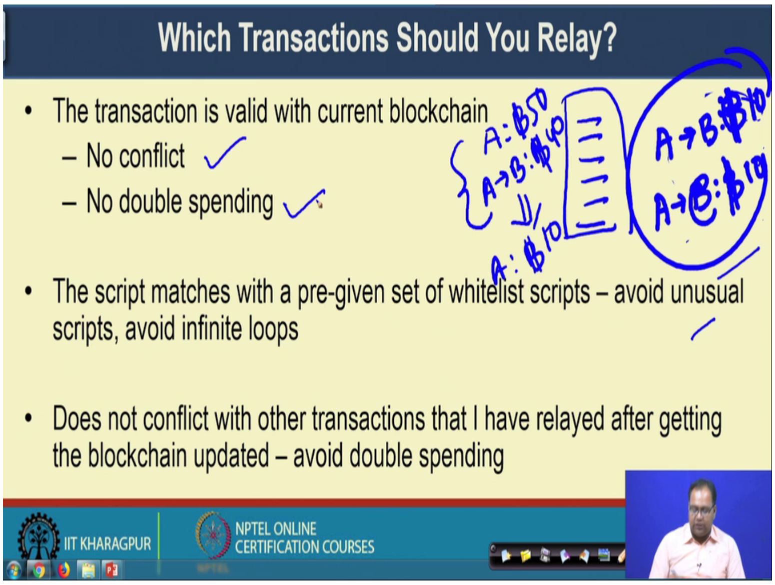
- The transaction is valid with current blockchain
  - No conflict
  - No double spending
- The script matches with a pre-given set of whitelist scripts – avoid unusual scripts, avoid infinite loops
- Does not conflict with other transactions that I have relayed after getting the blockchain updated – avoid double spending

Now, the question comes that which transaction should I relay. So, first of all you need to check that the transaction is valid with the current blockchain. So, in the blockchain you already have a list of transactions. So, you already have a blockchain which has list of transactions and whenever you are getting a new transactions of some bitcoin thirty from Alice you check with the existing transactions in the blockchain that whether Alice is allowed to spend that much of bitcoin or not.

So, it may happen in the transactions you can find out that initially Alice had some say 50 bitcoin with her and Alice already made a transactions of say 40 bitcoin. So, if you can see from the existing blockchain that this particular two transaction has been happened already that means, from here you can compute that currently Alice has only some 10 bitcoin with her. So, this particular transaction which is she is going to do right. Now, this is not a valid transaction. So, that way by looking into the existing history of the transactions you can determine whether a particular transaction is valid or not.

So, that is the first task that every node need to do that after doing the script validation they need to validate that there is no such conflict with the existing transactions, and there is no such double spending. Double spending in the sense that you are looking into there are two transactions from Alice that a say initially Alice had some 10 10 bitcoin with him. Now, now say, now say Alice is making one transaction of bitcoin 10 to Bob, and Alice is making another transactions of say bitcoin 10 to say not Bob to Charlie.

(Refer Slide Time: 07:57)



Which Transactions Should You Relay?

- The transaction is valid with current blockchain
  - No conflict ✓
  - No double spending ✓
- The script matches with a pre-given set of whitelist scripts – avoid unusual scripts, avoid infinite loops
- Does not conflict with other transactions that I have relayed after getting the blockchain updated – avoid double spending

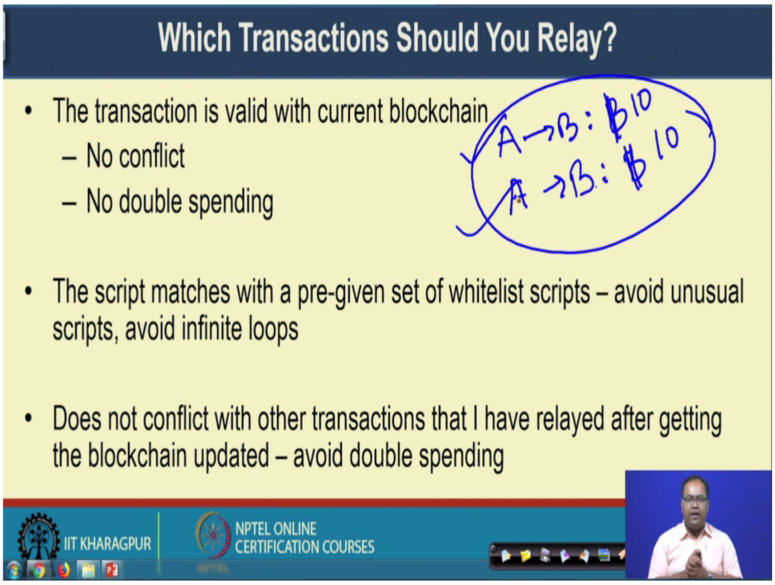
If Alice is making two such transactions that mean this is a example of double spending and Alice is not allowed to make this transaction. So, while doing this validation you have to check that with the existing blockchain there is no such conflict and at the same time there is no such double spending.

So, after doing this check then you check the script that the script matches with a pre given set of whitelist, whitelist script. So, in a bitcoin network there are already certain set of scripts which are normal or which are usual scripts. So, you check whether the script belongs to those set of whitelist script or not, and in general if you are normal bitcoin user you avoid the unusual scripts. Say for example, if a script somehow implemented a kind of indefinite loop inside the code, if that is the case then that is not a general feature of a bitcoin script.

So, you discard that particular transaction. And then the point that I have mentioned that although the double spending can be a part of the existing blockchain like the Alice has already made a transaction which has been recorded in the existing blockchain. And then Alice is making the transaction of the same bitcoin again which is an example of double spending another example of double spending is that Alice is sending two transactions one after another and in both the transactions she is try to spend spend that the same bitcoins. That means, Alice has made a Alice has made a transaction to Bob with some

10 bitcoins and immediately Alice has initiated another transaction to say Charlie with the similar 10 bitcoin.

(Refer Slide Time: 09:53)



Which Transactions Should You Relay?

- The transaction is valid with current blockchain
  - No conflict
  - No double spending
- The script matches with a pre-given set of whitelist scripts – avoid unusual scripts, avoid infinite loops
- Does not conflict with other transactions that I have relayed after getting the blockchain updated – avoid double spending

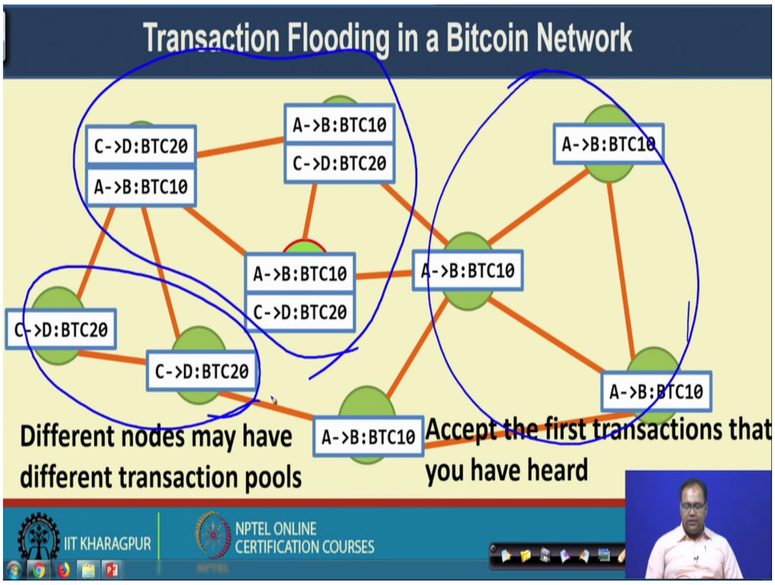
A→B: ฿10
A→B: ฿10

So, if that is the case that can be one example. The second example would be like Alice has a transferred Bob of 10 bitcoins and after few seconds Alice has initiated another transaction that maybe intention on transaction or that sometime becomes unintentional transaction; that means, you are you have kept the same button twice in your wallet if you tap the same button twice in your wallet that will generate two different transactions. So, that will generate two different transactions from Alice to Bob.

If you if you if it generous two different such transactions. So, you do not include both the transactions in the same blockchain. So, if you have already seen this transaction when you do not relay this transactions any further. So, this also avoid this kind of double spending where unintentionally Alice has tapped the same button twice.

So, that way you can validate that whether the transactions that you have received from one of the peer node whether you should relay the transactions further or not. Now, if you agree to relay the transactions further then you broadcast the transactions again among your peer nodes.

(Refer Slide Time: 11:27)


Transaction Flooding in a Bitcoin Network

Different nodes may have different transaction pools

Accept the first transactions that you have heard

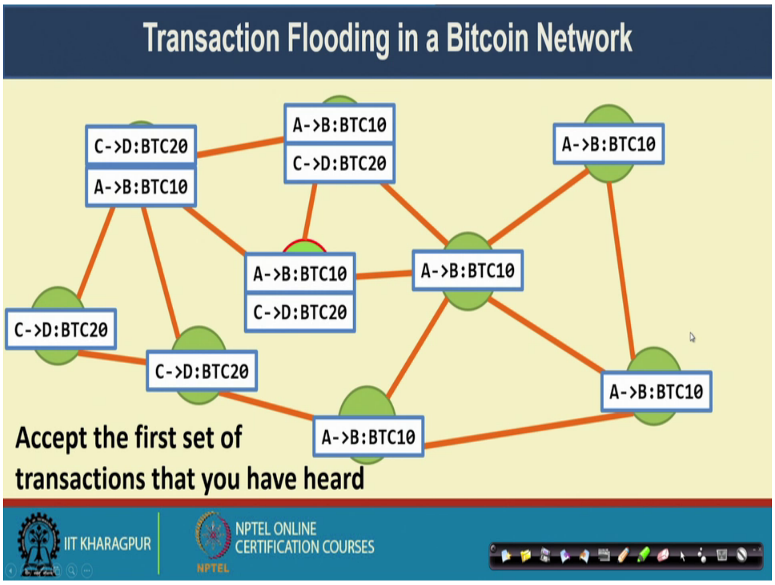
Now, whenever you are flooding the transactions in the network there are certain interesting scenarios it may happen that different nodes have different views of the transactions or they have different transactions pool.

So, for example, in this in this particular scenario this A to B transaction that gets propagated in this direction, there is another transaction C to D which is getting propagated. Now, this nodes this nodes they have only seen the C to D transactions this set of nodes they have only seen the A to B transaction. This set of nodes they have only seen the C to D transactions and this set of nodes they have observed both the transactions. So, ideally what happens that this can always happen in a network, and if this happens then you accept the first set of transactions that you have heard and use those set of transactions to construct a new block.

But whenever you are constructing a new block it is always like that different miners in the network, they can start working with different set of transactions and they try to push the transaction in a existing block. And in general what we do in case of a bitcoin network you wait for around 10 minutes of time for taking all the transactions which you are observing and then construct a block of a maximum block size.

Now, if you are waiting for a 10 minutes of time to collect all the transactions later on will see that it is guaranteed that by that time you will also receive the most updated block. So, once you received the most updated block, you can see that which are the transaction that you have adopt whether they have already included in the existing block or not, if they are not included in the existing block and you are a miner then you can construct a new block with the remaining transactions and you can try to mine the corresponding has function to find out the nonce and connect this block try to connect this block with the existing blockchain.

(Refer Slide Time: 13:40)

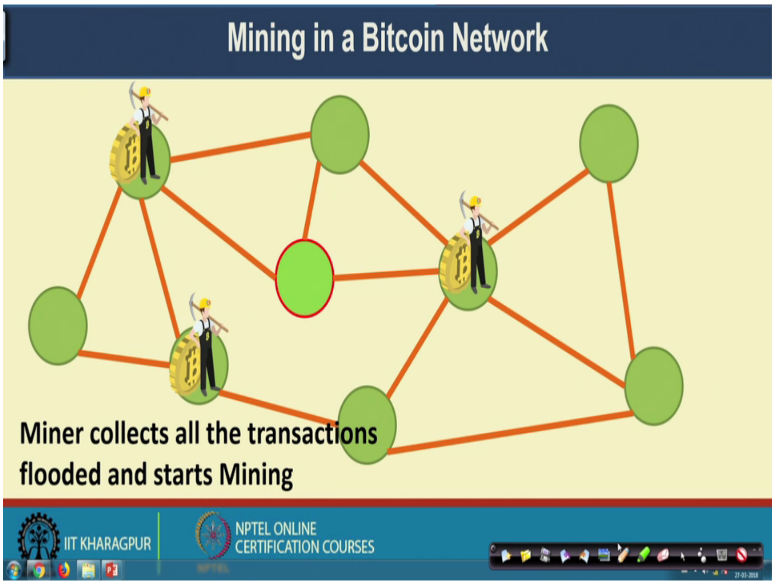


Transaction Flooding in a Bitcoin Network

Accept the first set of transactions that you have heard

Well, well so, that was the idea that you always accept the first setup transactions that you have heard.

(Refer Slide Time: 13:47)



Mining in a Bitcoin Network

Miner collects all the transactions flooded and starts Mining

Now, once you have constructed the blocks with the set of transactions that you have heard there are certain nodes in the bitcoin network which work as a miner.

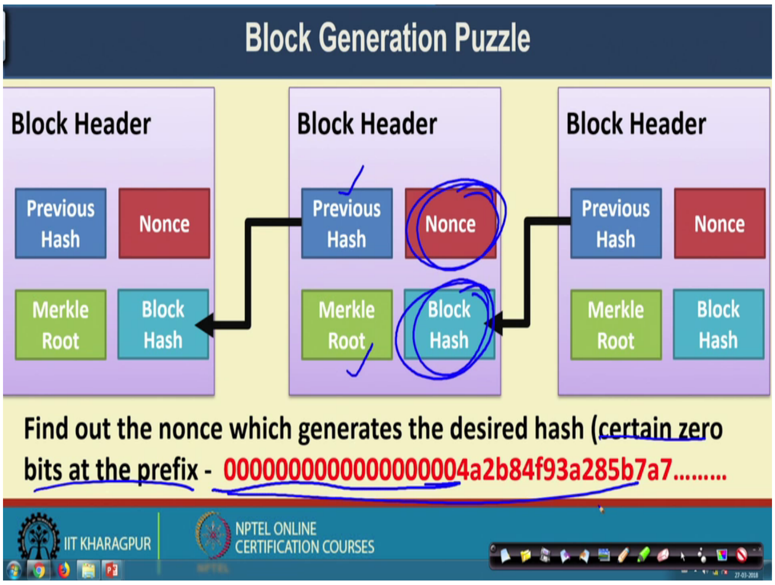
Now, remember that it is not necessary that every participant in the bitcoin network need to be a miner. There can be certain special node who has some good computation power and they can invest time to participate in the mining process new, and a mining procedure that we have discuss briefly the broad idea is to solve that hash puzzle where your task is to find out nonce.

So, that the resultant has value becomes a has the condition like you have certain number of zeros at the prefix. So, if that particular condition it is imposed and based on that condition you are trying to mine the new blocks you are trying to find out the new blocks that is a computationally difficult task and you need to you need to support with the power of your machine and the computation time.

So, that is why, so it is not necessary that all the nodes in the network work as a miner you can have certain number of nodes in the network which are ready to dedicate their resource for the mining purpose and they can participate in the mining procedure. And remember that as we discussed earlier like in this scenario the incentive for the miner is that if they are successfully able to mine a new block they will be able to get started reward from the network, which is the kind of incentive that they get.

Now, the miners they collect all the transactions from the network after collecting all the transaction from the network which have been flooded within the last 10 minute duration that I have mentioned, they construct the new block if the transactions are not already included in the most updated blockchain and they can start mining.

(Refer Slide Time: 15:49)



Block Generation Puzzle

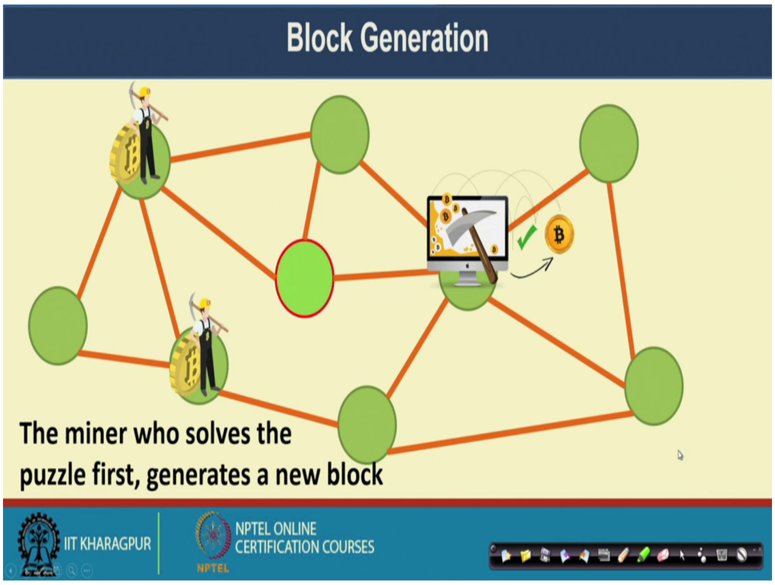
Find out the nonce which generates the desired hash (certain zero bits at the prefix - 000000000000000004a2b84f93a285b7a7.........

Now, in the mining part as we have mention that in a blockchain architecture you have the, this hash value of this block is included in the hash value of the next block. So, every block header contains the previous hash value and along with that the task of the miner is to find out this nonce. So, the task of the miners is to find out this nonce such that the hash value the block hash value that it is generating that particular block hash value has certain fixed number of zeros at the prefix.

So, a block has looks like this where you have certain number of zeros at the prefix. So, that is the difficulty of the problem that is imposed from the bitcoin network, and it works like a challenge to the miner and the task of the minor is to accept this challenge and try to solve this particular puzzle. Their task is to find out this nonce value such that they can find out the block has by including this merkle root and the previous hash so that this hash has this property it has certain number of fix zeros at the prefix.

Now, if that is the case based on that the blockchain architecture you can understand that more than one miners can actually find out the non simultaneously. So, we will discuss that how we actually handle this kind of problems in a blockchain or in a bitcoin network.

(Refer Slide Time: 17:32)

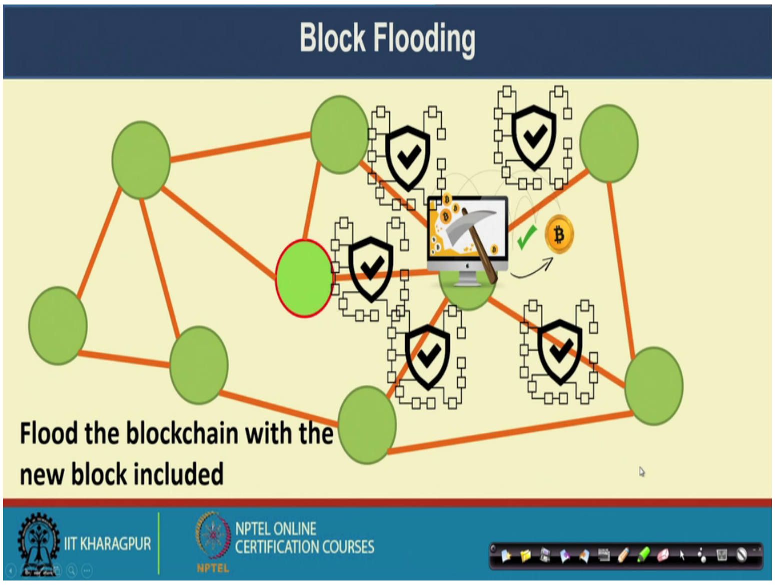


Block Generation

The miner who solves the puzzle first, generates a new block

So, in general in a in a ideal case one miner the miner will be able to correctly find out the nonce value by solving that puzzle. So, once the miner is able to find out the nonce value by solving that puzzle he or she is able to generate the new hash which connect the blockchain with the which connect the block with the previous blockchain and that way by connecting the new block with the previous blockchain the entire blockchain get updated. And after updating the blockchain you actually update you actually broadcast this updated blockchain to your to peer neighbors.

(Refer Slide Time: 18:06)

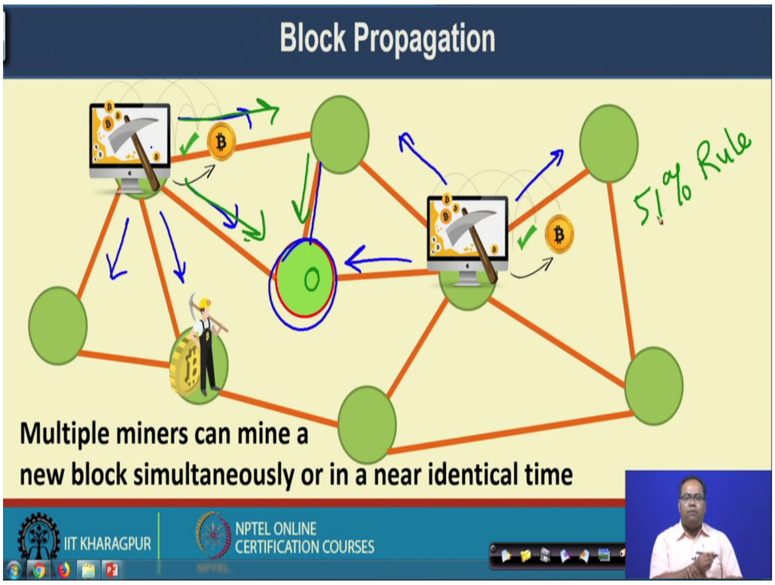


Block Flooding

Flood the blockchain with the new block included

So, that way the blocks, the updated blocks they are basically getting flooded again in the network. So, we have to floodings here one is the flooding of the transactions and the second is the flooding of the blocks from the miners.

(Refer Slide Time: 18:25)

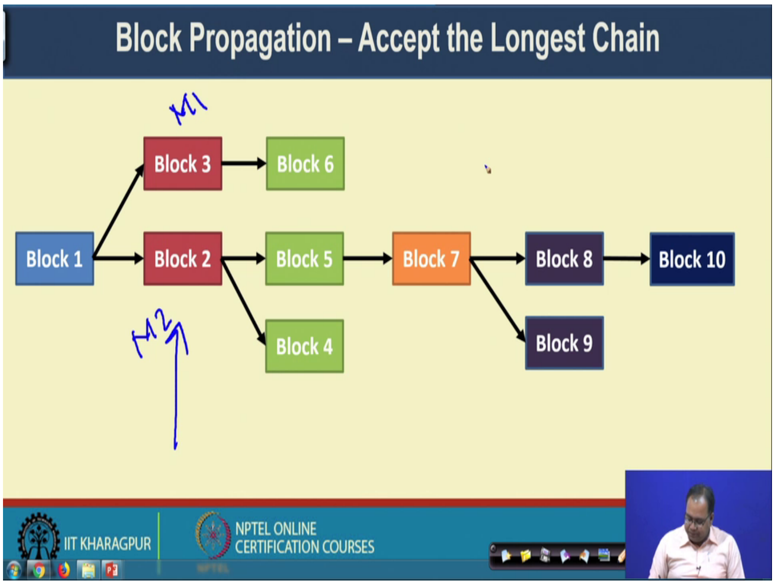


Block Propagation

Multiple miners can mine a new block simultaneously or in a near identical time

Now, as I have mentioned earlier that it is certainly possible that more than one miners has generated the blocks. Now, if more than one miners has generated the block there are multiple cases that may arise, it may happen that both the miners has mind the same block or it may happen that both the miners has mind different blocks because they have a different view of the transactions. So, what may happen that if they if they generate new blocks simultaneously or at least in a near identical type that the blockchain can have more than one blocks connected to a particular block.

(Refer Slide Time: 19:01)



Block Propagation – Accept the Longest Chain

So, in this example that we had seen earlier during or initial discussion that a particular block can have more than one trades through which the blocks are getting added. So, it is like that in this particular example that at that time duration one miner say M 1, that miner was able to find out the cryptographic hash for this block which connect block 3 with block 1, at the same time miner 2 this also able to connect this block to block 1.

Now, in a typical bitcoin network what will happen that let me go to the previous example that whenever two miners are able to construct the block simultaneously. So, these miner will start sending the blocks and this miner will also start sending the blocks ok, the updated blocks. So, these node consider this node in between. Now, this node is getting the blocks from 3 different from 3 different links, these node will get the blocks to this link to this link as well as to this link.

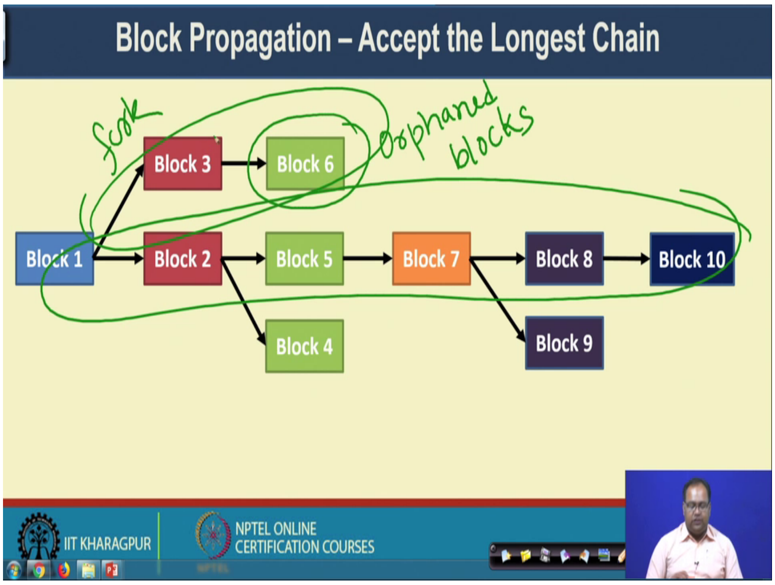
Now, whenever that particular particular node is getting more than one copy of the blockchain. So, this node accept the copy which has been transferred by maximum number of peers. So, if this copy of the blockchain from this node if it goes to say let us use a different colour. So, if it goes from here to here and then if it comes in this particular link; that means, this block has been received by this node from two peer. So, add I this blue block has been received from one peer. So, this node then will accept this brown block and it will broadcast this brown block further.

So, this particular rule we call as 50 percent rule. So, this 50 percent rule sorry it calls we call it as a 51 percent rule not 50 percent rule, we call it as a 51 percent rule. It says that the copy of the block that you are getting from more than 50 percent of the neighbors, you accept that a particular block and broadcasted further in the network and the other block you can discarding.

So, that way a different copies of the blockchain can propagated in the network and whenever different copies of the blockchains have propagating in the network then you can you can a select the blocks based on this principle. Otherwise if you are getting two copies of the blockchain where the maximum chain plaint is different if the maximum chain lengths differ between the two receive copies of the blockchain then you accept the copy which has the longest chain length that we discussed earlier.

So, that way, that way whenever you have such a multiple leaks multiple parts you always accept either the longest chain which is there and all the new newly mine blocks will be added to the longest chain.

(Refer Slide Time: 22:28)

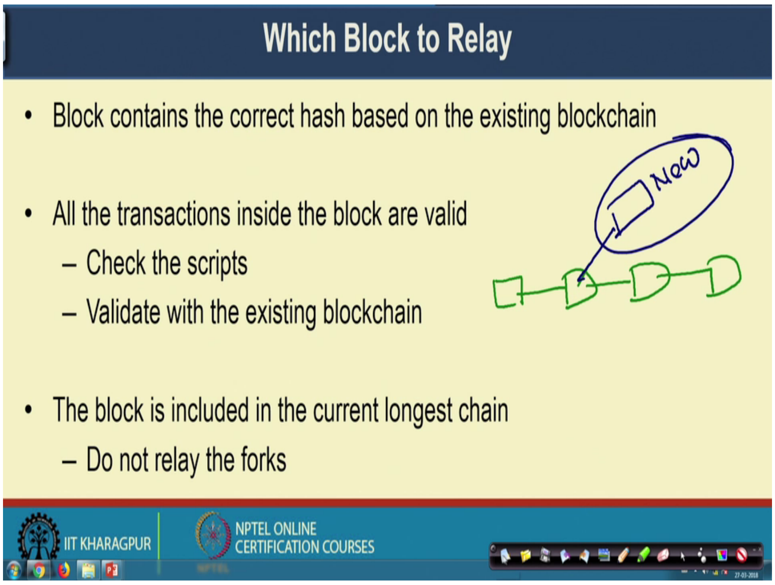


Block Propagation – Accept the Longest Chain

And this other parts which are there which can eventually construct and after sometime they will not be used by any of the nodes in the blockchain and they will get discarded this particular blocks we call as the orphaned blocks. These are termed as the orphaned blocks, and this procedure when where you can have a different part other than the longest chain this particular part is known as a fork. So, this is the longest chain and these are the fork. So, you always accept the longest chain and you ignore the forks.

Now, as I have mentioned that sometime it may happen that you have two different chains of the same length if there are two different chains of the same length then you accept the chain which has been which has been broadcasted by a more number of peers, ok.

(Refer Slide Time: 23:38)

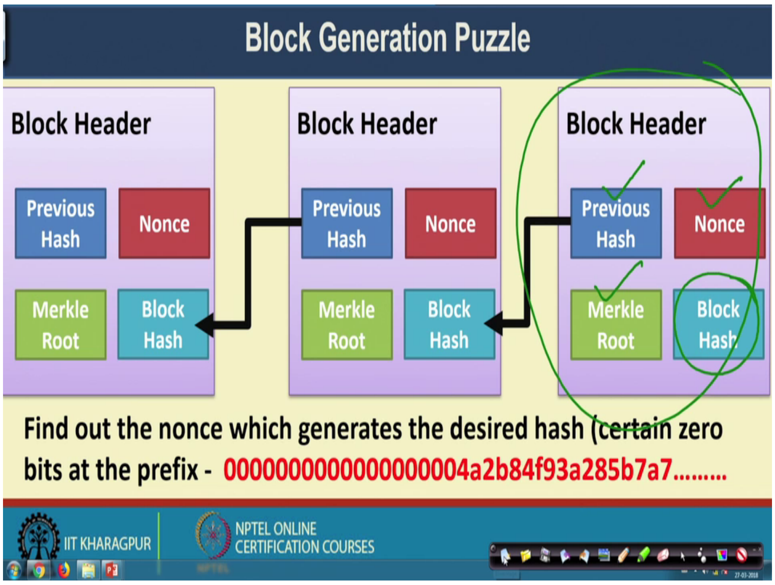


Which Block to Relay

- Block contains the correct hash based on the existing blockchain
- All the transactions inside the block are valid
  - Check the scripts
  - Validate with the existing blockchain
- The block is included in the current longest chain
  - Do not relay the forks

So, the question comes that whenever you are getting a block with block to relay. So, first you check the validity of the block like the block contain the correct hash based on the existing blockchain. So, the hash value is already included there. So, as we have mentioned during the discussion of the hash function that finding out the revers of a hash that means, if the message digest is given finding out the original original message it is difficult, but given a message you can easily construct the digest.

(Refer Slide Time: 24:19)



Block Generation Puzzle

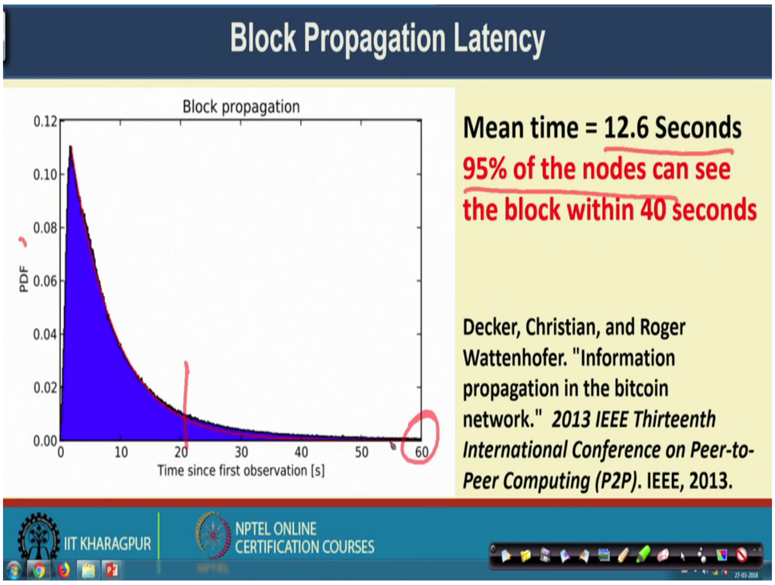
Find out the nonce which generates the desired hash (certain zero bits at the prefix - 0000000000000000004a2b84f93a285b7a7……..

Now, whenever a new blockchain is getting propagated in the network say for example, in this case whenever this blockchain is getting propagated in the network and you are getting this information say you this is the latest block that you have a received, this is the latest block that you have received. So, whenever you have received the blocks. So, in the block header you already have the block hash and you know that this block hash is generated by doing a hash over the previous hash, the nonce and the merkle root.

So, you take these 3 parameters apply play the hash function and you check whether you are able to get the block hash or not. If you are able to get the block hash that means, you are able to you are able to successful validate the authenticity of this particular block at this block can get added in the network, ok.

So, that was the first point. The second point was that you need to change check that all transactions inside the block are valid. So, you can do that again by checking the scripts or validating it with the existing blockchain that you already had and the third point that a block is included in the longest chain. So, you should not relay the floods. So that means, in a in a blockchain if it is a case like that say this is the longest chain, and a new block is a included here say a new block is included here, if this is a new block then you should not accept this new block because this new block is included as a fork and not a not as a part of the longest chain that you have.

(Refer Slide Time: 26:10)



Block Propagation Latency

Mean time = 12.6 Seconds

95% of the nodes can see the block within 40 seconds

Decker, Christian, and Roger Wattenhofer. "Information propagation in the bitcoin network." *2013 IEEE Thirteenth International Conference on Peer-to-Peer Computing (P2P)*. IEEE, 2013.

So, a the interesting fact about this block propagation is the block propagation latency. So, this is a one particular example which was taken from this paper by Christian Decker and Wattenhofer Roger. So, they have done a lot of statistical study over this bitcoin network and they have found out a distribution of the block propagation duration.

So, in this curve you can see that a you can see that the x axis it is the time for this block propagation that once the block has been created and after that you have a received update what is the time difference between the 2, that is the time since first observation which is in second and in the y axis you have the probability density function of that. And you can see that most of the blocks most of the time you can you can actually see the blocks with some twenty second duration.

So, the meantime for this block observation is comes to be 12.6 seconds; that means, on average within 12.6 seconds every node in the bitcoin network and get the most updated blockchain. And 95 percent of the nodes can see the block within 40 seconds. So, if you can wait up to 60 seconds or 1 minute within 1 minute if a new block is created it is sure that we need 1 minute you will be able to see the new block.

Now, you are waiting for 10 minutes to construct the next block. So, if that is the case then the interesting factories that by the time you will construct a new block you have already seen the most updated block. So, because you have seen the most updated block you can see only find out which are the transactions that you should include in the new block and try to add that new block with the existing blockchain if you are a miner and then you can you can start the mining procedure with a expectation that you will be able to solve the hash puzzle and you will be able to include that block in the blockchain.

So, that is all about a this particular lectures. So, we have broadly looked into that how the transactions are generated in a bitcoin network and how these a concept of concept of script checking is used to validate a particular transactions in a bitcoin network and finally, how the blocks are constructed and gradually included in a in a bitcoin network.

So, in the next set of lectures we will look into the details of the consensus distributed consensus algorithm which is a part of a blockchain. So, will start with the consensus as algorithm which is used in a general bitcoin network like the proof of our consensus such that is widely accepted as a consensus algorithm for bitcoin network, which was originally proposed for bitcoin network. And then gradually we will look into that details about other variants of consensus algorithm which people have proposed for bitcoin kind of network or in general any kind of permission list blockchain that that there a in our current system.

So, with this we have covered the basic part the basic understanding of a the cryptographic primitives a for blockchain and with bitcoin we have seen an example about how you can apply these concept of blockchain in a practical example of bitcoin to create a digital currency. So, with this as a starting point we will move further with the details of a different blockchain consensus algorithm and the blockchain security algorithm in the subsequent lectures. So, see you later.

Thank you.