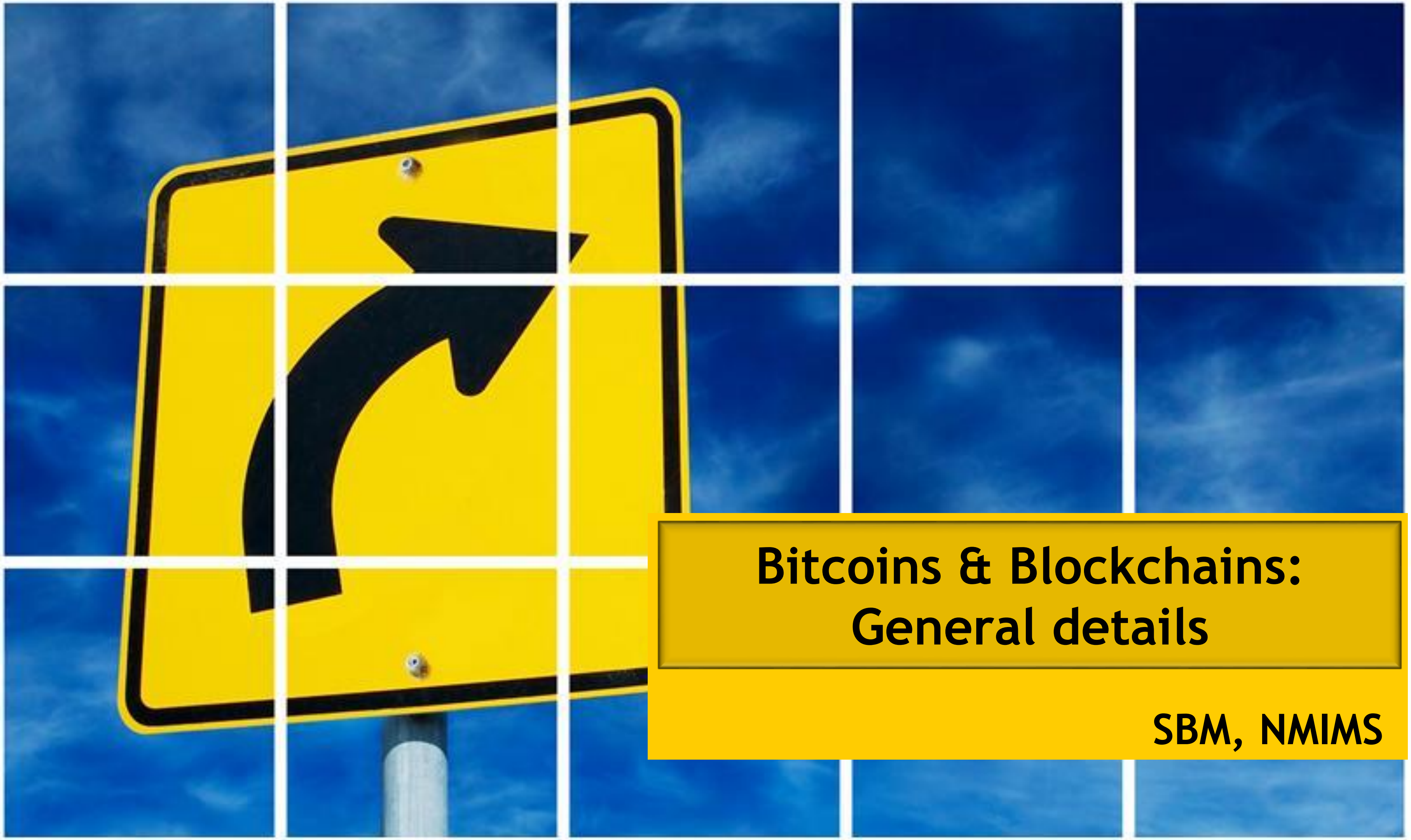


Bitcoins & Blockchains: General details

SBM, NMIMS

Aug-25

Bitcoin – at a glance

- Two different types of data structures are involved
  - Transactions, blocks
- Transactions are grouped together in blocks.
- Blocks are connected together via hashes of their predecessors
    - This forms an authenticated data structure: the blockchain
- Transactions and blocks are disseminated among all participating nodes
  - A gossiping protocol is used over a peer-to-peer (P2P) network

Bitcoin – Contd.

- A new block is added to the blockchain
  - If a node can provide a valid proof-of-work (PoW) for it
- PoW acts as a defence mechanism against Sybil attacks
  - Also provides a form of keyless signature to authenticate new blocks and the blockchain as a whole
- Honest nodes agree that at any point in time only the longest blockchain is considered valid
  - Commonly referred to as longest chain rule
  - Actually it is the heaviest chain
    - blockchain that is the hardest to compute in terms of PoW
- If a node does not consider a block to be valid, then the block is not added to its blockchain
  - Implicit consensus process
  - Can be described as a "random leader election" on each solved PoW
  - Leader is allowed to propose a new block
  - Implicit agreement on all previous blocks
    - By appending its new block to the end of the respective blockchain

Bitcoin – Contd.

- Bitcoin
  - Distributed system
  - Uses PoW and a blockchain as a probabilistic consensus mechanism
    - To agree on the contained set of transactions as well as their order
  - System ensures that all peers agree on the current ownership status of bitcoins
    - Necessary to correctly handle state transitions in the ownership from one block to the next block
  - Underlying consensus approach to achieve this
    - Nakamoto consensus
    - The leader is allowed to decide one block
    - Then another leader is elected based on solving a PoW puzzle
    - Leaders signal their approval of previous blocks by appending to the rightful chain of blocks (in their view)

Bitcoin – Contd.

- Miners: people who provide their computational resources and run Bitcoin nodes
  - Miners are rewarded with currency units (i.e., bitcoins) for every valid PoW provided for a block and its associated transactions: motivation
  - Security and decentralization of Bitcoin: comes from
    - Technical aspects
    - Incentive engineering

Bitcoin operations

- Two main components
  - Consensus management
    - Everything that is relevant to consensus
      - Consensus algorithms, communication aspects
  - Digital asset management
    - All applications that build upon the agreed state and act upon it,
      - Key and transaction management
  - Can be replaced independently of each other

Bitcoin operations

- Consensus management component
  - Network subsystem
  - Storage subsystem
  - Consensus algorithm subsystem
- Digital asset management component
  - Key management subsystem
  - Transaction management subsystem
- Changing components: no impact as long as they can communicate with each other
  - A wallet can run on any current instance of Bitcoin

Bitcoin operations

- Subsystems within one component
  - Cannot be directly replaced without impact
    - Replacing the P2P networking implementation with a different gossiping protocol
      - No impact on the basic rules of Nakamoto consensus
      - May alter message propagation times
        - Directly influences the security properties of the consensus algorithm

TOC

Core Concepts

- Basic data structures used in Bitcoin:
  - Addresses, transactions, and blocks
- Block: most fundamental data structure in Bitcoin
  - Components:
    - One block header
    - Other transactions associated with that block
- Blocks are chained together
  - Each block contains cryptographic hashes of their predecessors
  - Forms a list where each member is linked to its previous member
    - Commonly referred to as a blockchain
  - The current state of currency is represented by the order of the blocks in the chain
    - Represent a ledger of all performed transactions
      - Transactions are processed sequentially
      - Depending on their position in the block in which they occur

Core Concepts

- Transactions in a block are processed in sequential order
  - All transactions in a block
    - Tied to the respective block via a Merkle tree root hash
      - Included in the block header
      - Like a hash value over all transactions

Core Concepts

- Bitcoin Address: Elliptic Curve Digital Signature Algorithm (ECDSA) public/private key pair
  - Public: address
    - Can be compared to an account number in banking
  - Private: corresponding secret key
    - Can be compared to the password or signature
    - Required to withdraw money from an ordinary savings account

Core Concepts

- Transaction:
  - Used to transfer currency units from one address to another
  - Can be created by any entity that is in possession of currency units (bitcoins)
    - Possession: control over the private key of the respective address (i.e., public key)
      - That currently holds the currency units that are to be transferred

      ⇒ Address that has received transactions in the past
  - Comprises one or multiple inputs and one or multiple outputs
    - Input:
      - Unlocks a previous output by providing a valid cryptographic signature
      - Proof - holder of the address that previously received bitcoins is also in possession of the required private key
        - » Private key is needed to generate the signature that unlocks the funds so that they can be used

          ⇒ Transferred to another Bitcoin address

Core Concepts

- Alice wants to transfer 5 bitcoins to Bob
  - She first requires Bob’s Bitcoin address
    - Assume that this is transferred through a communication channel
  - Bob’s address goes to the output of the transaction that is being constructed
    - Together with the number of coins that is to be transferred to this account, i.e., 5
  - Alice needs to prove that she is in possession of the required number of bitcoins
    - And that she really wants to transfer them to Bob
  - Alice searches the blockchain for previous transactions where bitcoins were sent to addresses that are under her control
    - Where she is in possession of the corresponding private keys
  - She unlocks as many of these previous transactions as needed to cover the desired output of 5 bitcoins
    - Assume that she uses two previous transactions (outputs) for this, comprising of 4 and 3 bitcoins

Core Concepts

- WRT previous transactions
  - Alice creates an input in the current transaction
  - For every previous output she wants to unlock
    - Each input uniquely identify previous outputs by their transaction ID and number
    - To unlock those outputs, she has to prove that she is the rightful owner
      - She does this by providing cryptographic signatures along with every input
- Alice now adds an output to the transaction which transfers 5 bitcoins to Bobs Bitcoin address
  - The two unlocked inputs sum up to more than the desired value of 5 bitcoins
- Alice adds another output for transferring the change of 2 bitcoins back to a Bitcoin address that is under her control

Core Concepts

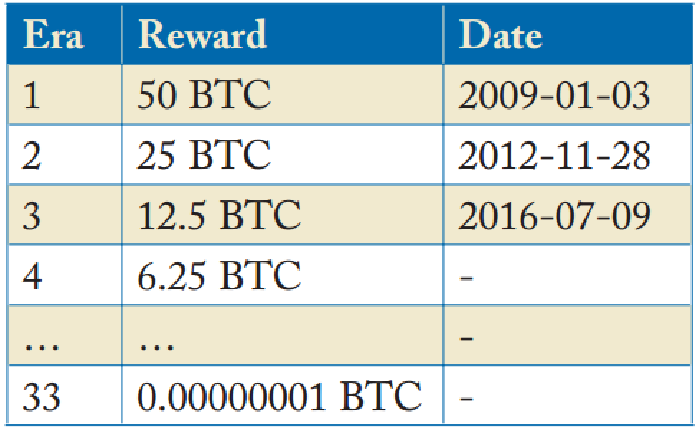
- Post-construction of the transaction
  - Broadcast to the Bitcoin peer-to-peer network
    - Waits until it is included in a block (to be newly generated)
- New Block => at the head of the blockchain
  - After inclusion of transaction => called confirmed
- The number of confirmations
  - Defined by the number of blocks
    - Building on top of the block that contains the transaction
  - Represent the number of blocks in the blockchain
    - That has been accepted by the network
      - After the block that contains the transaction
  - Each node checks that a transaction is valid when it learns about it and sends a confirmation

Core Concepts

- Transaction Validation
  - Validity is construed based on the following criteria:
    - All unlocked inputs have not been spent (i.e., unlocked and used) in a previous transaction
    - All cryptographic signatures in the inputs are valid
    - The sum of all values unlocked in the inputs
      - Is greater than or equal to
        - The sum of all values specified in the outputs of the transaction
- Coinbase
  - Actual creation of bitcoins happens in the coinbase transaction
    - The first transaction in every block
    - Has a special status among all other transactions
    - The block creator is allowed to create a predetermined number of bitcoins out of thin air
      - As a reward for finding a valid proof-of-work
      - Dependent on the number of blocks from the genesis block

Core Concepts

- Finite Supply
  - – The smallest currency unit in the Bitcoin ecosystem is one Satoshi
    - One bitcoin is defined as 1 * 10^8 Satoshis
  - – The official currency symbol for bitcoin is XBT (ISO 4217)
    - Community still widely uses BTC as symbol
  - – Limited supply of 21 million bitcoin
    - Artificial limit, established programmatically
      - – Reference implementation Bitcoin Core
      - – Algorithm that issues the mining rewards
      - – Every 210,000 blocks, a new era is reached
        - » Received mining reward is halved
        - » The algorithm defines 33 eras



Era	Reward	Date
1	50 BTC	2009-01-03
2	25 BTC	2012-11-28
3	12.5 BTC	2016-07-09
4	6.25 BTC	-
…	…	-
33	0.00000001 BTC	-

Core Concepts

- Fees and Change
  - The amount associated with a specific Bitcoin address cannot be split up
    - Has to be unlocked as a whole in a transaction input
    - It is necessary to include additional outputs in a transaction
      - if the surplus of currency units is to be transferred back as change to the sender
        - ⇒ A Bitcoin address which is under the control of the sender
  - If the sum of the values in the inputs of the transaction is still greater than the sum of the values in the outputs of the transaction
    - The difference is collected as a transaction fee by the miner of the block
      - In which the transaction is included
      - All transaction fees of a block are added to the reward of the coinbase transaction

Core Concepts

- Transaction Scripts
  - Bitcoin clients validate transactions by executing a script
  - Bitcoin uses a Forth-like, stack-based, reverse-polish, Turing Incomplete language
  - split up into two parts
    - *scriptPubKey* - first part resides in the transaction output that is to be spent (transferred) – locking script
    - *scriptSig* - second part is given in the respective transaction input that is to unlock the funds
    - For evaluation, both parts are concatenated and executed
    - If the execution returns true the script is considered valid
      - Respective fund is allowed to be spent in the associated transaction

Core Concepts

- Turing Incomplete
  - Limited functionality
  - Not capable of making jumps and/or loops
    - They can’t enter an endless loop
    - Not possible to create complicated transactions, slow down the system
- Reverse Polish (aka Postfix notation)
  - Operators follow the operands
  - Examples
    - 3+4 will appear as 34+
    - So, for longer more complicated sums:
      - 5*3+4 will appear as 534+*
- Stack: type of data structure
  - Last In First Out
- Forth Like
  - Similar to another stack based language

Consensus Management

- All entities in the network agree on a single universal "truth"
  - Who owns what, without having to trust anyone
- Traditional payment systems depend on a trust model
  - Central authority providing a clearinghouse service
  - Verifying and clearing all transactions
- No central authority in Bitcoin
  - Every full node has a complete copy of a public ledger that it can trust as the authoritative record
  - This ledger is not created by a central authority
    - Assembled independently by every node in the network
    - Nodes act on information transmitted across insecure network connections
      - Arrive at the same conclusion
      - Assembles a copy of the same public ledger as everyone else
  - Bitcoin network achieves global consensus without central authority
    - Authoritative, trusted, public, global ledger

Consensus Management

- Bitcoin - decentralized mechanism for emergent consensus
  - Consensus is not achieved explicitly
    - There is no election or fixed moment when consensus occurs
    - Consensus is an emergent artefact
      - Resulting from asynchronous interaction of independent nodes
      - All following simple rules
- Decentralized consensus in Bitcoin emerges from the interplay of four processes
  - Independent verification of each transaction, by every full node
  - Independent aggregation of those transactions into new blocks by mining nodes
    - Coupled with demonstrated computation through a proof-of-work algorithm
  - Independent verification of the new blocks by every node and assembly into a chain
  - Independent selection, by every node
    - Of the chain with the most cumulative computation
      - Demonstrated through proof of work

Consensus Management

- Transaction creation using the scripts
  - Resulting transaction is sent to the neighbouring nodes in the network
    - So that it can be propagated across the entire bitcoin network
  - Before forwarding transactions to its neighbours
    - Bitcoin nodes will verify the transaction after receipt
    - Only valid transactions are propagated across the network
      - Invalid transactions are discarded at the first node that encounters them
  - Every node builds a pool of valid new transactions (the transaction pool)
    - After validating transactions
      - A bitcoin node will add them to the memory pool, or transaction pool
      - Transactions await there until they can be included into a block

Mining

- Mining Nodes aggregates new transactions into blocks
  - Keeps listening for new transactions
    - New blocks discovered by other nodes
    - As well as trying to mine a new block
  - On receiving an new block (and validating it)
    - Abandon efforts for existing work
    - Check all the transactions in the memory pool
      - Remove any that were included in block
      - Only unconfirmed transactions in the memory pool
    - New empty block is constructed: called candidate block
      - Not yet a valid block: no proof of work
      - Will become valid only if the miner succeeds in finding a solution to the proof-of-work algorithm

Mining

- Competition among miners effectively ends with the propagation of a new block
  - Acts as an announcement of a winner
  - To miners, receiving a new block means someone else won the competition and they lost
    - The end of one round of a competition is also the beginning of the next round

Mining

- Construction of the candidate block
  - Mining nodes selects transactions from the memory pool
    - Applying a priority metric to each transaction
    - Adding the highest priority transactions first
      - Old and high-value inputs to be prioritized over newer, smaller inputs
    - Prioritized transactions can be sent without any fees, if there is enough space in the block
  - Some space reserved for priority transactions
    - Some space may be filled with no-fee transactions
    - Rest filled with transactions with highest fee
  - Transactions left in the memory pool
    - After the candidate block is filled
    - Remain in the pool for inclusion in the next block
    - No expiration time-out for bitcoin transactions
      - Valid transaction will be valid in perpetuity

Mining

- If a transaction is only propagated across the network once
  - It will persist only as long as it is held in a mining node memory pool
    - When a mining node is restarted, its memory pool is wiped clear,
      - It is a transient non-persistent form of storage
  - If a valid transaction is not executed it may eventually not reside in the memory pool of any miner
  - Transaction initiator is expected to retransmit such transactions
    - Or reconstruct them with higher fees
      - If they are not successfully executed within a reasonable amount of time

Mining

- Mining of the candidate block
  - To find a solution to the proof-of-work algorithm that makes the block valid
  - Mining:
    - Process of hashing the block header repeatedly
    - Changing one parameter
    - Until the resulting hash matches a specific target
  - The hash function's result cannot be determined in advance
    - Nor can a pattern be created that will produce a specific hash value
    - Only way to produce a hash result matching a specific target is to try again and again
      - Randomly modifying the input until the desired hash result appears by chance

Mining

- Difficulty
  - Bitcoin mining block contains the difficulty target
  - Difficulty is a measure of how difficult it is to find a hash below a given target

  - Bitcoin's blocks are generated every 10 minutes, on average
  - The Bitcoin network has a global block difficulty
    - Valid blocks must have a hash below this target
    - Network difficulty change: Every 2016 blocks
      - Average hashrate during that period is measured
      - Difficulty is adjusted based on that
    - Difficulty target is set to whatever mining power will result in a 10-minute block interval

Mining

- On successful mining of the candidate block
  - The concerned node transmits the block to all its peers
  - The peer nodes receive, validate, and propagate the new block
    - Each node adds it to its own copy of the blockchain
    - They abandon their efforts to find a block
      - Immediately start computing the next block in the chain
  - Independent validation: only valid blocks are propagated on the network
    - Miners who act honestly get their blocks incorporated in the blockchain
    - Ensures that the miners can’t cheat
      - Miners have to construct a perfect block
        - Based on the shared rules that all nodes follow
        - Mine it with a correct solution to the proof of work

Consensus Management

- Rules on which the majority of the nodes have to agree
  - To eventually reach consensus on the state of the blockchain
    - If there is an agreement on the validity and order of the blocks in the chain
      - There is also an agreement on the order of transactions
    - Required to determine whether a certain transaction is valid
      - Uses only transaction outputs that have not been spent so far
- Randomized consensus based on proof-of-work
  - Randomly select one node in the network to be the leader for the next "round" (i.e., the next block)
  - Leader is allowed to propose the next block
    - Another leader is chosen according to the same principle
    - Current leader can implicitly agree with the chain collected previously
      - By appending his newly created block at the head of the chain
      - Or disagree by choosing a different (previously created), block to append to
      - The chance of a node being selected as leader is dependent on its relative hashing power in comparison to all other nodes.
      - Therefore, any node can increase its chances to be selected by increasing its computational share

Consensus Management

- Protection against Sybil attacks
  - Name comes from a book, "Sybil"
    - About a woman with dissociative identity disorder
  - Attempt to control a peer network by creating multiple fake identities
    - To outside observers, these fake identities appear to be unique users
    - Behind the scenes, a single entity controls many identities at once
    - As a result, that entity can influence the network through additional voting power in a democratic network
      - Or echo chamber messaging in a social network
    - Grants undue influence to a single entity
      - Simply because that entity controls many pseudonyms
        - Fake Reddit accounts that upvote posts on behalf of a given company or cause
        - Amazon sellers can buy fake reviews from accounts around the world. These pseudonyms are hard to detect and remove
  - Nodes can join (and leave) the Bitcoin P2P network and start (or stop) participating in the protocol at will
- Prevention:
  - Cost to Create an Identity; Chain of Trust; Unequal Reputation

Block Assembly

- Final step in bitcoin's decentralized consensus mechanism
  - Assembly of blocks into chains
  - Selection of the chain with the most proof of work
- Post Validation, node will attempt to assemble a chain
  - By connecting the validated block to the existing blockchain
- Nodes maintain three sets of blocks:
  - Blocks connected to the main blockchain
  - Blocks that form branches off the main blockchain (secondary chains)
  - Blocks that do not have a known parent in the known chains (orphans)
    - Invalid blocks are rejected: not included in any chain

Block Assembly

- Main chain: whichever chain of blocks has the most cumulative difficulty associated with it
  - Usually this is also the chain with the most blocks in it
  - Can also be two equal-length chains: one has more proof of work
  - Also have branches with blocks that are "siblings" to the blocks on the main chain
    - Valid blocks but not part of the main chain
    - Kept for future reference, in case one of those chains is extended to exceed the main chain in difficulty
    - Secondary chains occur as a result of an almost simultaneous mining of blocks at the same level

Block Assembly

- New block received ⇒ node attempt to put into the existing blockchain
  - Lookup the parent block in the existing main chain
    - If present, node puts the new block in the existing chain
  - Sometimes new block extends a secondary chain
    - Node will attach the new block to the secondary chain it extends
    - Node will compare the difficulty of the secondary chain to the main chain
      - If the secondary chain has more cumulative difficulty than the main chain
        - Node will re-converge on the secondary chain
          - ⇒ It will select the secondary chain as its new main chain
          - Making the earlier main chain a secondary chain
      - (If the node is a miner) Node will now construct a block extending this new, longer, chain

Block Assembly

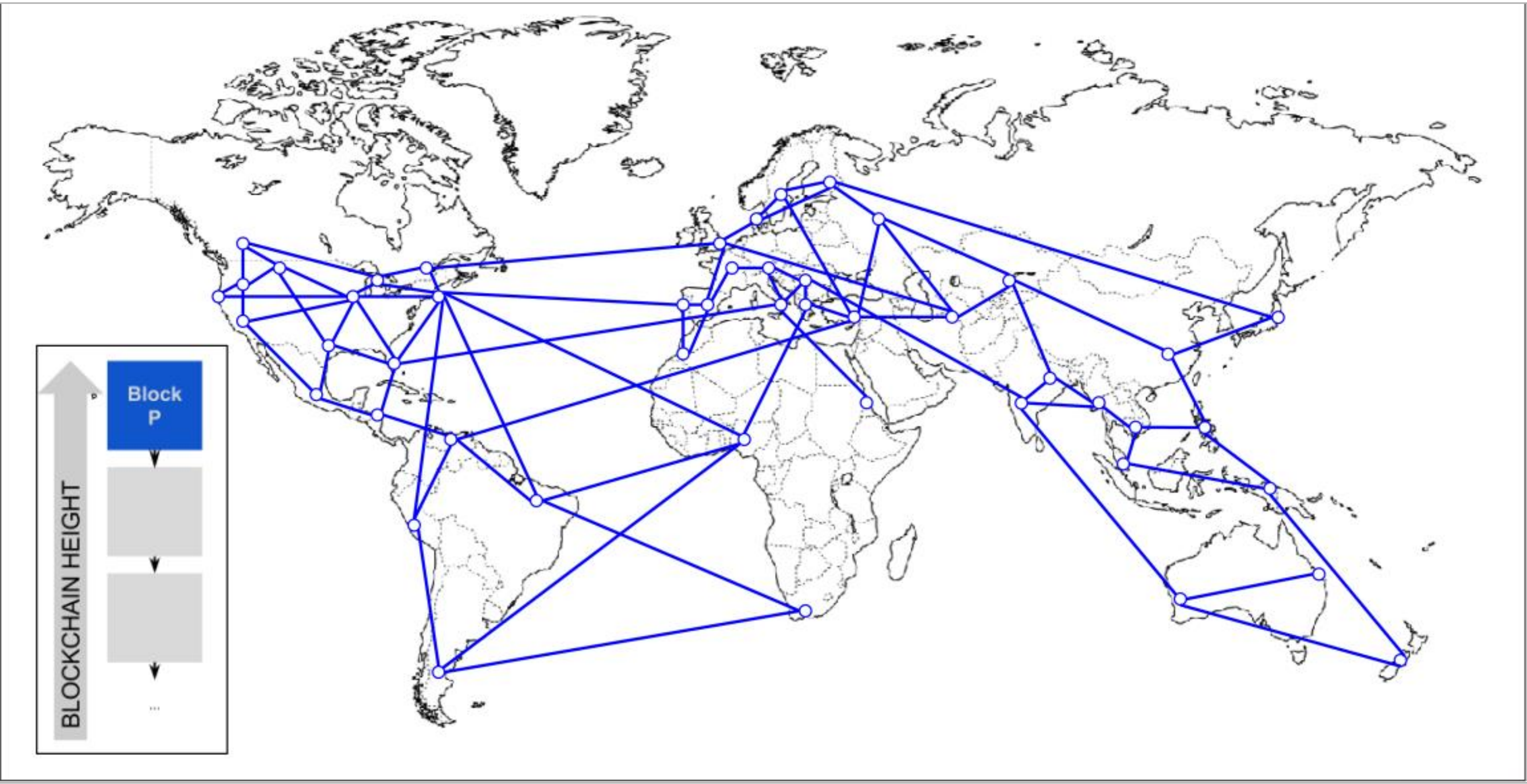
- Valid block with no parent found in the existing chains
  - Block is considered an "orphan"
  - Orphan blocks are saved in the orphan block pool
  - They will stay there until their parent is received
  - Once the parent is received and linked into the existing chains
    - Orphan blocks pulled out of the orphan pool and linked to the parent
      - Making it part of a chain
  - Usually occurs when two blocks that were mined within a short time of each other
    - Received in reverse order (child before parent)
- By selecting the greatest-difficulty chain, all nodes eventually achieve network-wide consensus
  - Temporary discrepancies between chains are resolved eventually as more proof of work is added
    - Extending one of the possible chains
  - Mining nodes "vote" with their mining power
    - By choosing which chain to extend by mining the next block
      - When they mine a new block and extend the chain, the placement of the new block represents their vote

TOC

Block Assembly

- Blockchain Forks
  - Blockchain is a decentralized data structure
    - Different copies of it are not always consistent
      - Blocks arrive at different nodes at different times
      - Leading to differences in their chaining
    - Resolution:
      - Each node always selects and attempts to extend the chain of blocks that represents the most proof of work
        - Also known as the longest chain or greatest cumulative difficulty chain
        - Cumulative difficulty: sum of the difficulty recorded in each block in a chain
          - Total amount of proof of work that has been expended to create that chain
      - With this approach the global bitcoin network converges to a consistent state
        - Forks occur as temporary inconsistencies between versions of the blockchain
          - Resolved by eventual re-convergence as more blocks are added to one of the forks

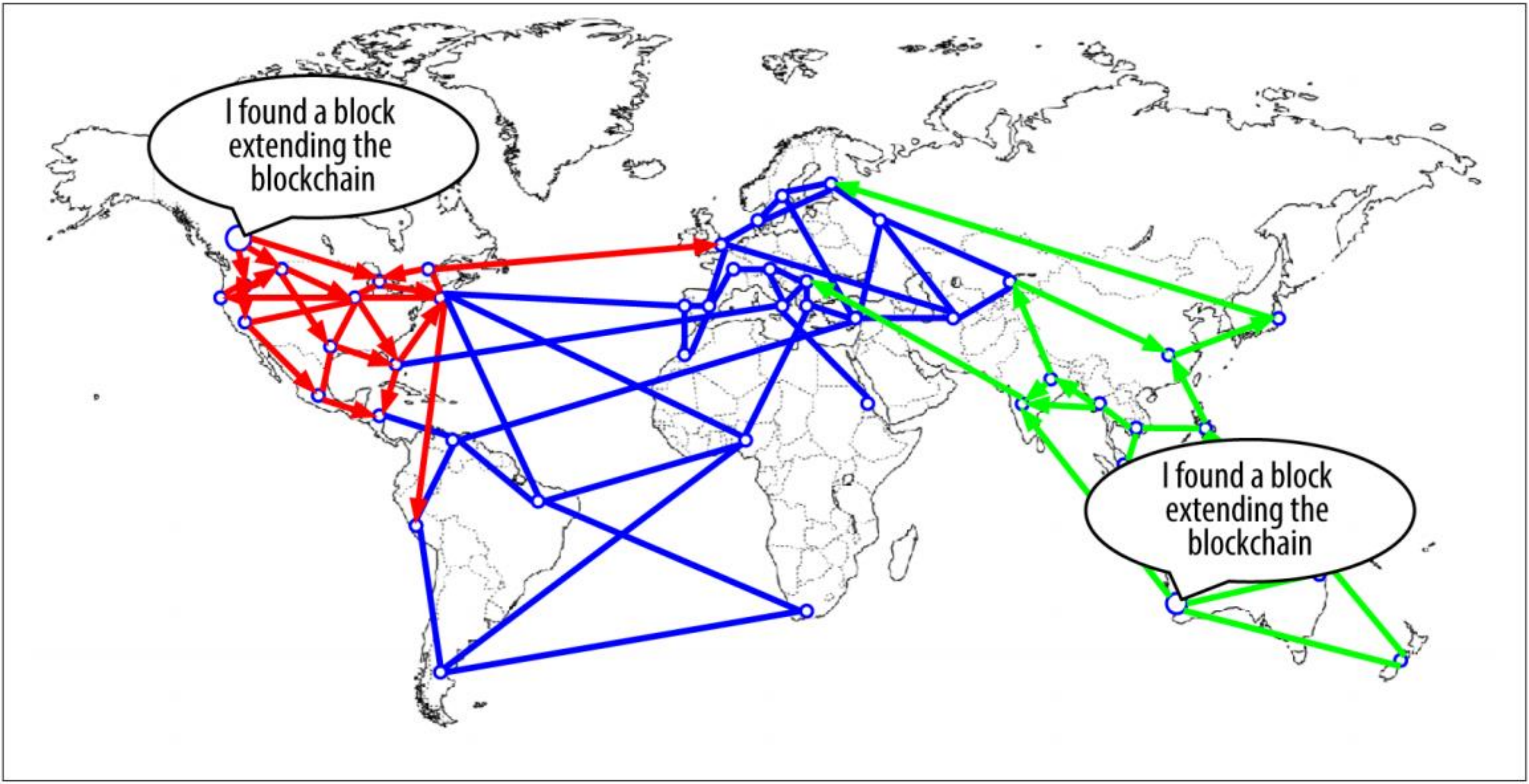
Block Assembly



Blockchain fork event - before the fork

TOC

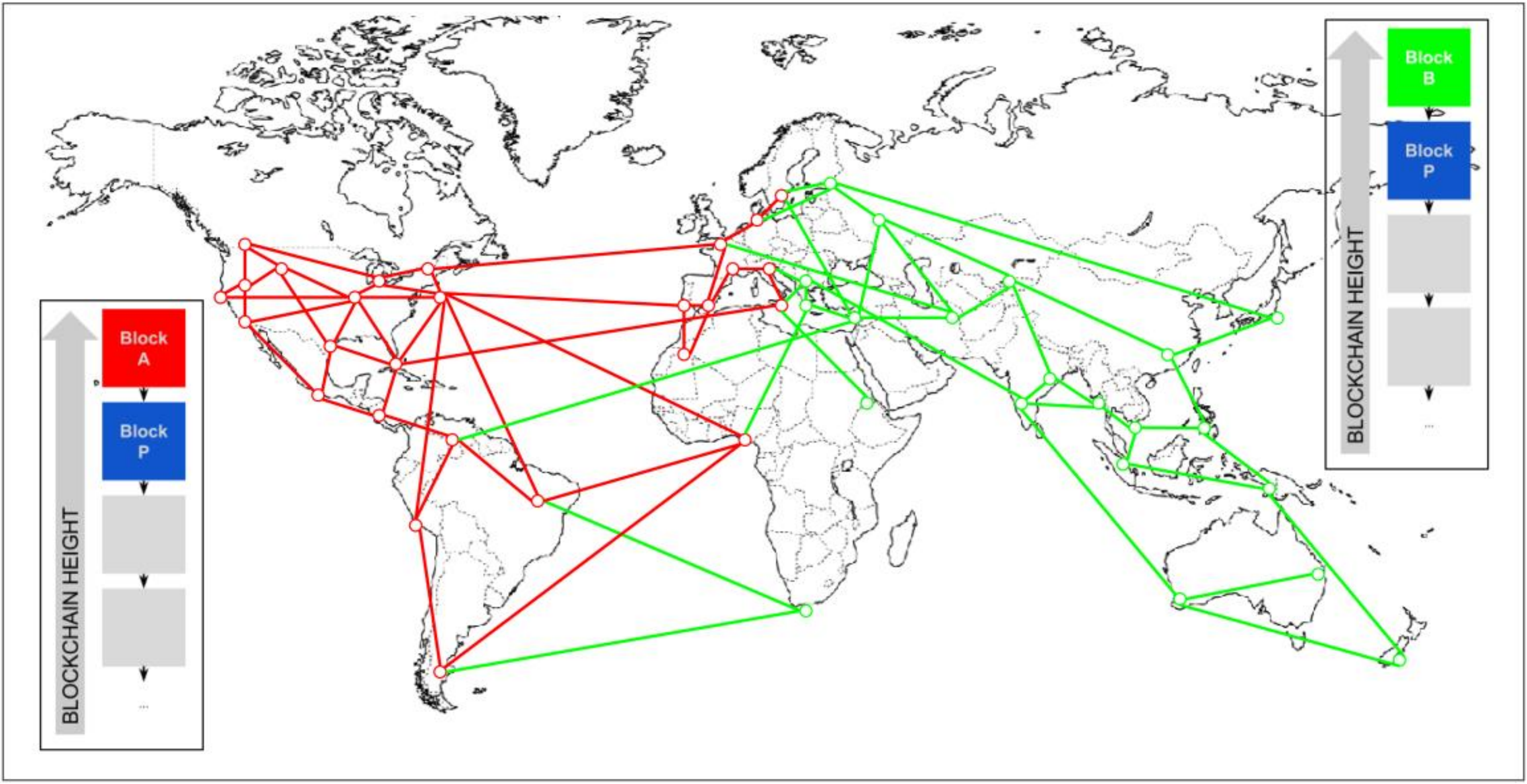
Block Assembly


I found a block extending the blockchain
I found a block extending the blockchain

Blockchain fork event: two blocks found simultaneously

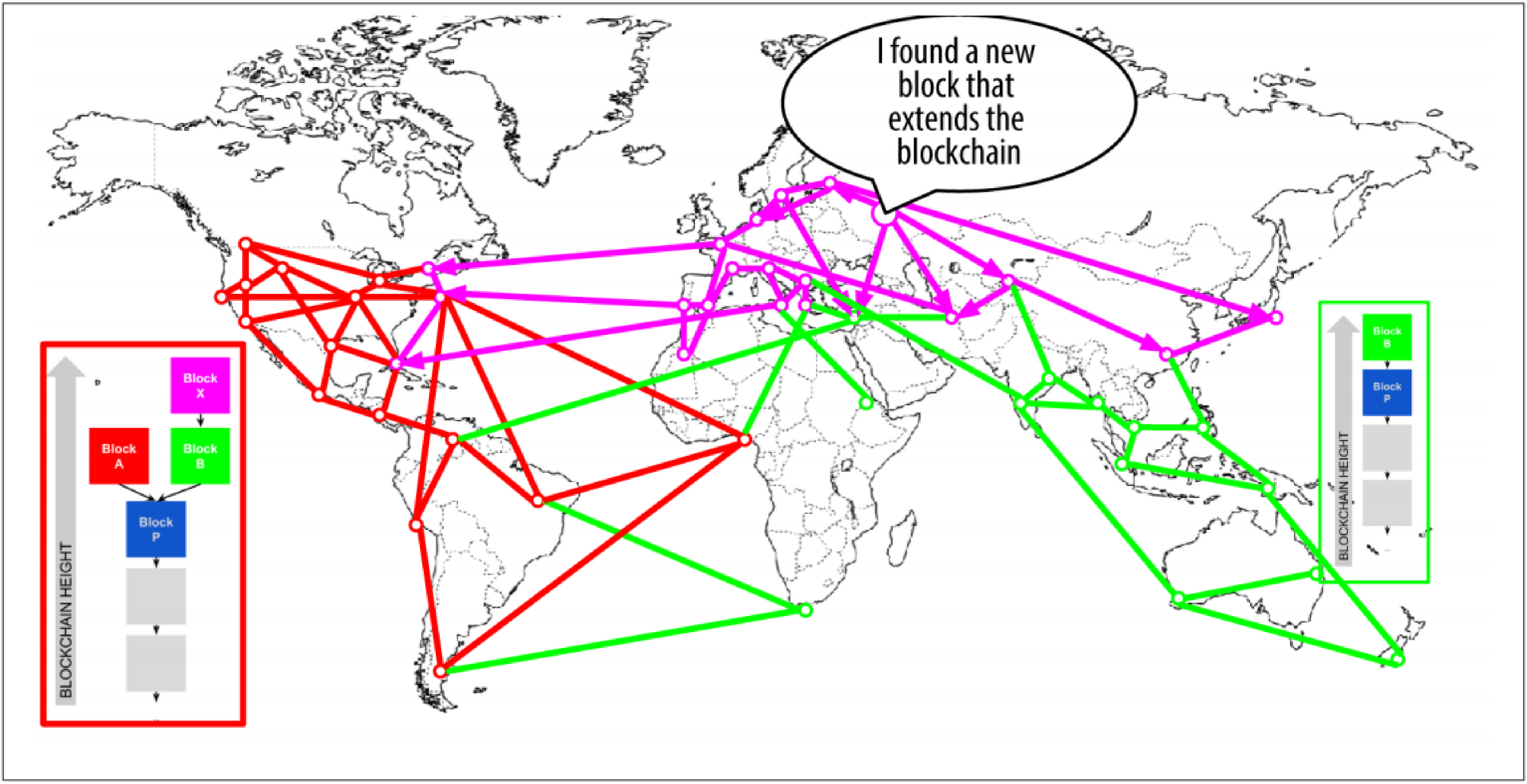
TOC

Block Assembly



Blockchain fork event: two blocks propagate, splitting the network

Block Assembly



Blockchain fork event: a new block extends one fork

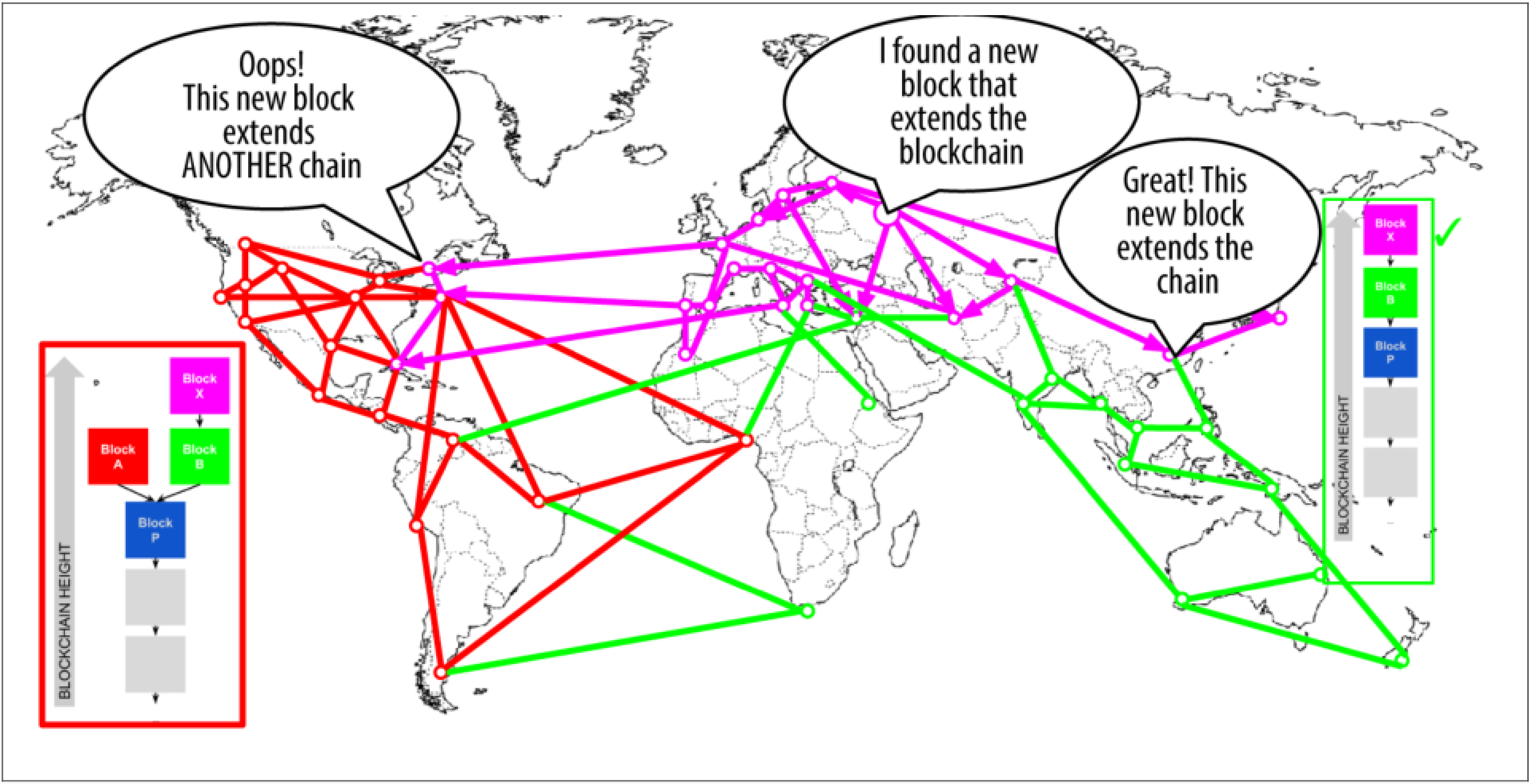
Block Assembly

- All nodes that had chosen "green" in the previous round
  - Simply extend the chain one more block
- Nodes that chose "red"
  - Will now see two chains: blue-green-pink and blue-red
  - The chain blue-green-pink is now longer (more cumulative difficulty) than the chain blue-red
  - Those nodes will set the chain blue-green-pink as main chain
    - Change the blue-red chain to being a secondary chain
      - This is a chain reconvergence: nodes are forced to revise their view of the blockchain to incorporate the new evidence of a longer chain
    - Any miners working on extending the chain blue-red will now stop that work
      - Their candidate block is an "orphan"
      - Its parent "red" is no longer on the longest chain

Block Assembly

- The transactions within "red" are queued up again for processing in the next block
  - Since that block is no longer in the main chain.
- The entire network re-converges on a single blockchain blue-green-pink
  - With "pink" as the last block in the chain
- All miners immediately start working on candidate blocks that reference "pink" as their parent
  - To extend the blue-green-pink chain

Block Assembly


Oops! This new block extends ANOTHER chain
I found a new block that extends the blockchain
Great! This new block extends the chain
BLOCKCHAIN HEIGHT
Block X
Block A
Block B
Block P
BLOCKCHAIN HEIGHT
Block X
Block B
Block P
✓

Block Assembly

- Theoretically possible for a fork to extend to two blocks
  - If two blocks are found almost simultaneously by miners
  - On opposite "sides" of a previous fork
    - Chance of that happening is very low
- One-block fork might occur every week
  - Two-block fork is exceedingly rare
- Bitcoin’s block interval of 10 minutes: design compromise
  - Between fast confirmation times (settlement of transactions)
  - Probability of a fork
- Faster block time: make transactions clear faster
  - But lead to more frequent blockchain forks
  - Slower block time would decrease the number of forks
    - But make settlement slower

Bitcoin Mining

- Extremely Competitive Industry
- Hashing power has increased exponentially every year
  - Total hashes per second across the network
- Sometimes driven by complete change of technology
  - 2010 and 2011: miners switched from using CPU to GPU mining and field programmable gate array (FPGA) mining
  - 2013: introduction of ASIC (application-specific IC) mining
    - By placing the SHA256 function directly on silicon chips
    - Specialized for the purpose of mining

Bitcoin Mining

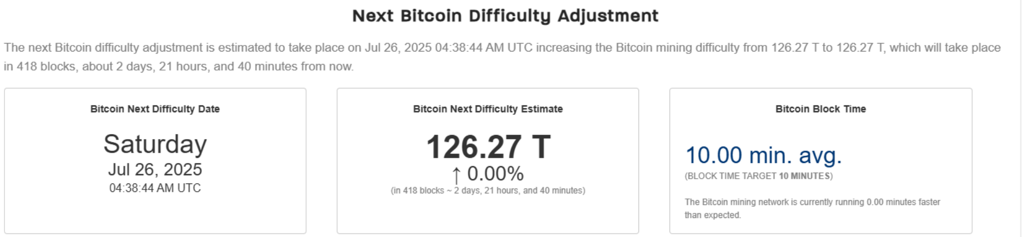
- Competition between miners, the growth of bitcoin
  - Exponential increase in the hashing power
  - Difficulty has risen to match it
- Difficulty metric: measured as a ratio of current difficulty over minimum difficulty
  - Difficulty of the first block
- Total hashing power of the bitcoin network:
  - Possible to estimate from the number of blocks being mined
  - Current block difficulty

2009	0.5 MH/sec	8 MH/sec	16×
2010	8 MH/sec	116 GH/sec	14,500×
2011	16 GH/sec	9 TH/sec	562×
2012	9 TH/sec	23 TH/sec	2.5×
2013	23 TH/sec	10 PH/sec	450×
2014	10 PH/sec	150 PH/sec	15×
2018 (02 Oct)		49K PH/s	
2022		173.78 EH/s	
2023		268.03 EH/s	
2024		510.60 EH/s	
2025		809.79 EH/s	

Current Bitcoin Hashrate (BTC Hashrate): 847.69 EH/s

- Block height 906,782
- Difficulty of 126.27 T

Bitcoin Hashrate All-Time High: Jul 23, 2024 at block 853,498 (879.51 EH/s)



Next Bitcoin Difficulty Adjustment

The next Bitcoin difficulty adjustment is estimated to take place on Jul 26, 2025 04:38:44 AM UTC increasing the Bitcoin mining difficulty from 126.27 T to 126.27 T, which will take place in 418 blocks, about 2 days, 21 hours, and 40 minutes from now.

Bitcoin Next Difficulty Date

Saturday
Jul 26, 2025
04:38:44 AM UTC

Bitcoin Next Difficulty Estimate

126.27 T
↑ 0.00%
(in 418 blocks ~ 2 days, 21 hours, and 40 minutes)

Bitcoin Block Time

10.00 min. avg.
(BLOCK TIME TARGET **10 MINUTES**)

The Bitcoin mining network is currently running 0.00 minutes faster than expected.

Bitcoin Mining

- Mining power of the network
  - Continues to advance at an exponential pace
  - Race for higher density chips is matched with a race for higher density data centres
    - Thousands of these chips can be deployed
    - No longer about how much mining can be done with one chip
  - How many chips can be squeezed into a building
    - While still dissipating the heat
    - Providing adequate power

Bitcoin Mining

- Mining Pools
  - Individual miners working alone (solo miners) don’t stand a chance
  - Likelihood of them finding a block to offset their electricity and hardware costs is very low
  - Even the fastest consumer ASIC mining system cannot keep up with commercial systems
    - Stacking tens of thousands of chips in giant warehouses near hydroelectric power stations
  - Miners now collaborate to form mining pools
    - Pooling their hashing power
    - Sharing the reward among thousands of participants
    - Miners get a smaller share of the overall reward
      - Typically get rewarded every day, reducing uncertainty

Bitcoin Mining

- Mining pools coordinate many hundreds or thousands of miners
  - Uses specialized pool-mining protocols
- Individual miners configure their mining equipment to connect to a pool server
  - After creating an account with the pool
- Their mining hardware remains connected to the pool server while mining
  - Synchronizing their efforts with the other miners
- The pool miners share the effort to mine a block
  - Then share the rewards

Consensus Attacks

- Consensus mechanism:
  - Depends on having a majority of the miners acting honestly out of self-interest
  - Vulnerable to attack by miners (or pools)
    - That attempt to use their hashing power to dishonest or destructive ends
  - If a miner or group of miners can achieve a significant share of the mining power
    - Can attack the consensus mechanism
    - So as to disrupt the security and availability of the bitcoin network

Consensus Attacks

- Consensus attacks can only affect future consensus,
  - At best the most recent past (tens of blocks)
- Bitcoin's ledger becomes more and more immutable as time passes
  - In theory, a fork can be achieved at any depth
  - In practice, the computing power needed to force a very deep fork is immense
    - Making old blocks practically immutable
- Consensus attacks also do not affect the security of the private keys and signing algorithm (ECDSA)
  - Consensus attack cannot steal bitcoins, spend bitcoins without signatures, redirect bitcoins, or otherwise change past transactions or ownership records
    - Can only affect the most recent blocks
    - Cause denial-of-service disruptions on the creation of future blocks

Consensus Attacks

- 51% attack
  - One attack scenario against the consensus mechanism
  - A group of miners controls a majority (51%) of the total network’s hashing power
    - Ability to mine the majority of the blocks
    - Can collude to attack bitcoin
  - The attacking miners can cause deliberate "forks" in the blockchain
    - Double-spend transactions
      - Attacker causes previously confirmed blocks to be invalidated
      - By forking below them and re-converging on an alternate chain
    - Execute denial-of-service attacks against specific transactions or addresses
  - With sufficient power, an attacker can invalidate six or more blocks in a row
    - Causing transactions that were considered immutable (six confirmations) to be invalidated
  - A double-spend can only be done on the attacker’s own transactions
    - For which the attacker can produce a valid signature

Consensus Attacks

- Other scenario for a consensus attack
  - To deny service to specific bitcoin participants (specific bitcoin addresses)
- An attacker with a majority of the mining power can ignore specific transactions
  - If they are included in a block mined by another miner
    - The attacker can deliberately fork and re-mine that block
    - Again excluding the specific transactions
  - This type of attack can result in a sustained denial of service against a specific address or set of addresses
    - for as long as the attacker controls the majority of the mining power
- 51% attack scenario
  - Doesn't actually require 51% of the hashing power
  - Can be attempted with a smaller percentage of the hashing power
  - The 51% threshold is the level at which such an attack is almost guaranteed to succeed

Consensus Attacks

- A consensus attack is essentially a tug-of-war for the next block
  - "Stronger" group is more likely to win
  - With less hashing power, the probability of success is reduced
    - Other miners control the generation of some blocks with their "honest" mining power
  - More hashing power an attacker has
    - The longer the fork he can deliberately create
    - The more blocks in the recent past he can invalidate
    - Or the more blocks in the future he can control
- Various types of consensus attacks are possible with as little as 30% of the hashing power

Altcoins – Merged Mining

- Altcoins - alternative cryptocurrency derived from Bitcoin
- Namecoin: first fork of Bitcoin
  - Intends to provide an alternative to the DNS
  - Possible to store arbitrary name-value pairs in its blockchain
  - Extends the Bitcoin protocol
    - Introduces transaction types: structured approach toward handling the storage and management of additional information in the blockchain (e.g., DNS entries)
- Litecoin:
  - Forked from Bitcoin by replacing its PoW
  - Uses the scrypt cryptographic hash function
  - The aim was to reduce the advantage of Bitcoin miners using hardware devices (ASICs) specifically built for high-performance SHA256 hash operations
  - Uses a reduced block interval of 2.5 minutes
- Dogecoin:
  - Started as an experiment but is now maintained by a vibrant community
  - Indirect fork of Litecoin
    - Smaller block interval of one minute
    - And a slightly adjusted difficulty and reward algorithm over time

Altcoins – Merged Mining

- Merged Mining
  - Process of allowing simultaneous mining
    - Of two different cryptocurrencies
  - Have to be based on the same algorithm
  - Originally conceived as a bootstrap technique
    - Aiming to increase the PoW difficulty
    - Therefore, increase the security of altcoins in their early stage
      - When they are more vulnerable to dishonest miners
  - Aims to improve the blockchain security
    - Rapidly increasing the number of nodes participating in the distributed consensus
  - Key idea: allow a blockchain (e.g., Namecoin) to accept valid PoW produced for another blockchain (e.g., Bitcoin)
    - Provided that they meet the hardness criteria of the receiving (child) blockchain
    - Even if they do not meet the criteria of the sending (parent) blockchain

Merged Mining

- Allows a miner to mine for more than one block chain at the same time
  - Benefit: every hash the miner does, contributes to the total hash rate of both (all) currencies
    - As a result they are all more secure
- The miner (or mining controller in the case of pooled mining) builds a block for both hash chains
  - In such a way that the same hash calculation secures both blocks
  - If a miner solves a block (at the difficulty level of either or both block chains) the block is re-assembled with the completed proof of work and submitted to the correct block chain (or both blocks are separately reassembled and each submitted to the corresponding network if it met both of their difficulty requirements)
- Benefits:
  - A lot of Bitcoin miners will probably do merged mining
    - It costs them nothing and gives them a greater return than mining Bitcoins alone
  - Namecoin will be more secure against a 51% attack

TOC

Thank you