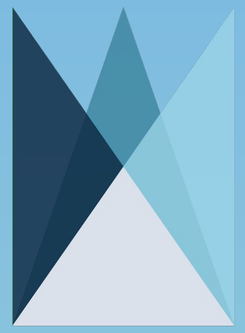
THE BITCOIN NETWORK



BLOCKCHAIN
@ COLUMBIA

Block in More Depth

Field	Description	Size
Blocksize	number of bytes following up to end of block	4 bytes
Block header	consists of 6 items	80 bytes
Transaction counter	positive integer	1-9 bytes
transactions	(non empty) list of transactions	<transaction counter> many transactions

BLOCKCHAIN
@ COLUMBIA

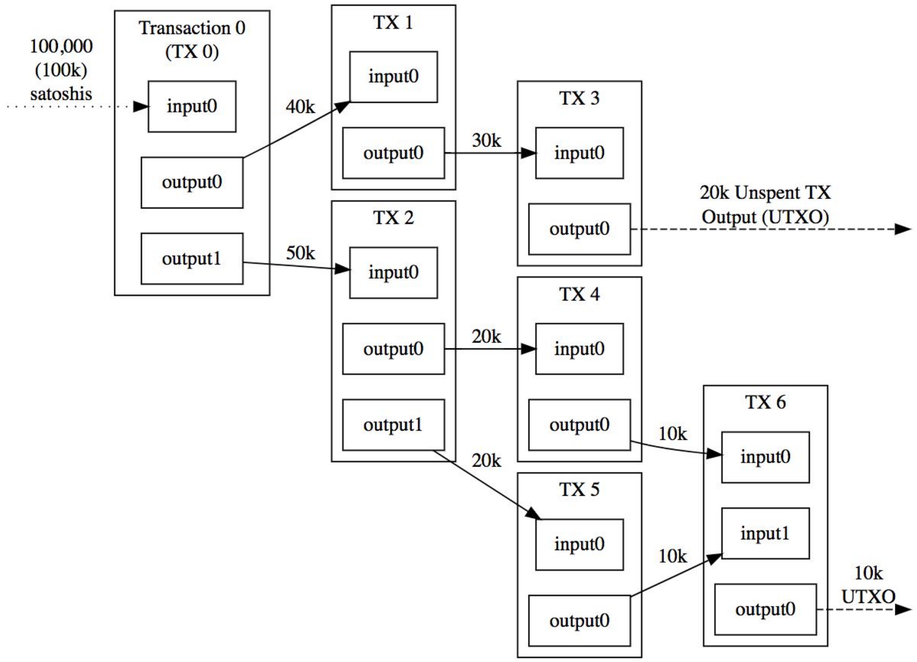
Block Header

Field	Description	Size
Version	Software version the node was running	4 bytes
hashPrevBlock	256-bit hash for the previous block header	32 bytes
hashMerkleRoot	256-bit hash for the root of the tree	32 bytes
Time	Current timestamp	4 bytes
Bits	Specifies current target	4 bytes
Nonce	32-bit number for hashing the next header	4 bytes

BLOCKCHAIN
@ COLUMBIA

Transactions

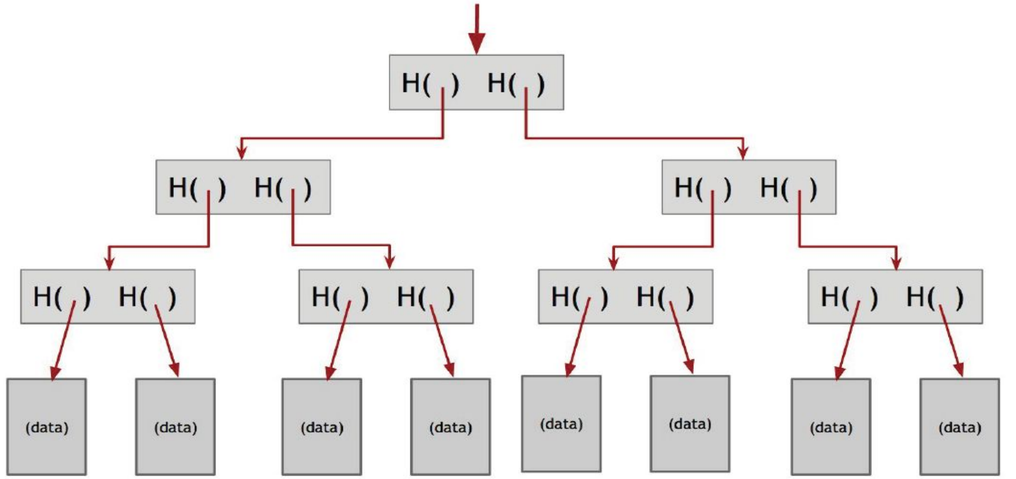
- All transactions are chained together
  - Bitcoins transfer from transaction to transaction, not from wallet to wallet
  - **Transaction Identifiers** (TXIDs)
    - Hashes of signed transactions
  - **Unspent Transaction Outputs** (UTXO)
    - Valid transactions only accept UTXOs as inputs

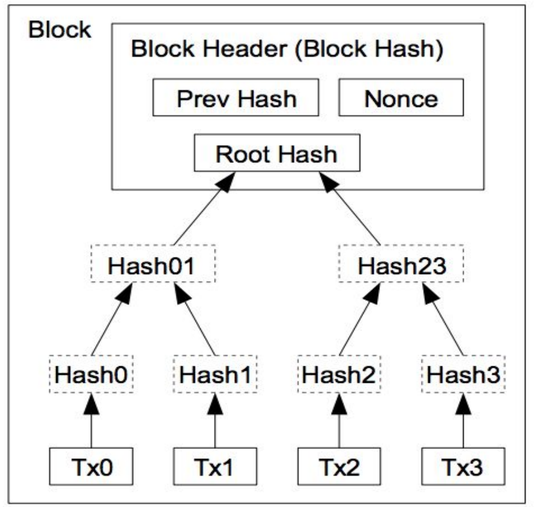


Triple-Entry Bookkeeping (Transaction-To-Transaction Payments) As Used By Bitcoin

Merkle Tree

- Transactions are stored in a tree of hash pointers
- Merkle trees allow for transactions to be verified securely and efficiently
- If a single transaction is changed then so does the Merkle root





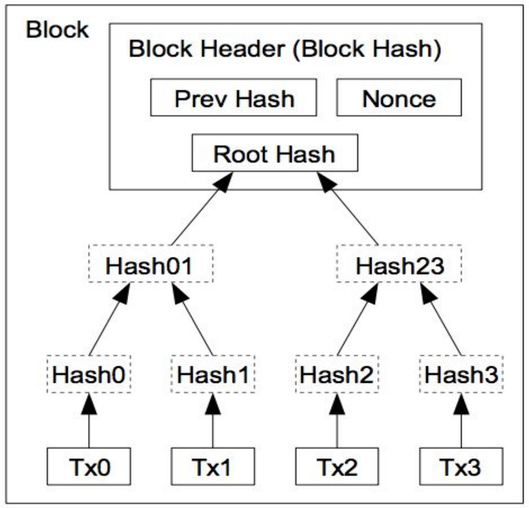
Transactions Hashed in a Merkle Tree

BLOCKCHAIN
@ COLUMBIA

How Large is the Entire Bitcoin Blockchain?

SPV Nodes

- **Simplified Payment Verification** (SPV) **Nodes**, also known as “thin clients” only download the block headers instead of the full blockchain
  - Each block header is 80 bytes
- If a SPV node wants to verify a transaction, it requests all the subsequent hashes in the merkle tree from a full node



Transactions Hashed in a Merkle Tree

BLOCKCHAIN
@ COLUMBIA

Nonce

- 32 bit arbitrary random number that is typically used once
- In the creation of a block, a hash is created from the concatenation of the nonce, previous hash, and list of transactions
    - Brute force all possible nonces to find a hash smaller than the target hash
    - The output hash should be below a target number calculated based on difficulty

Proof of Work

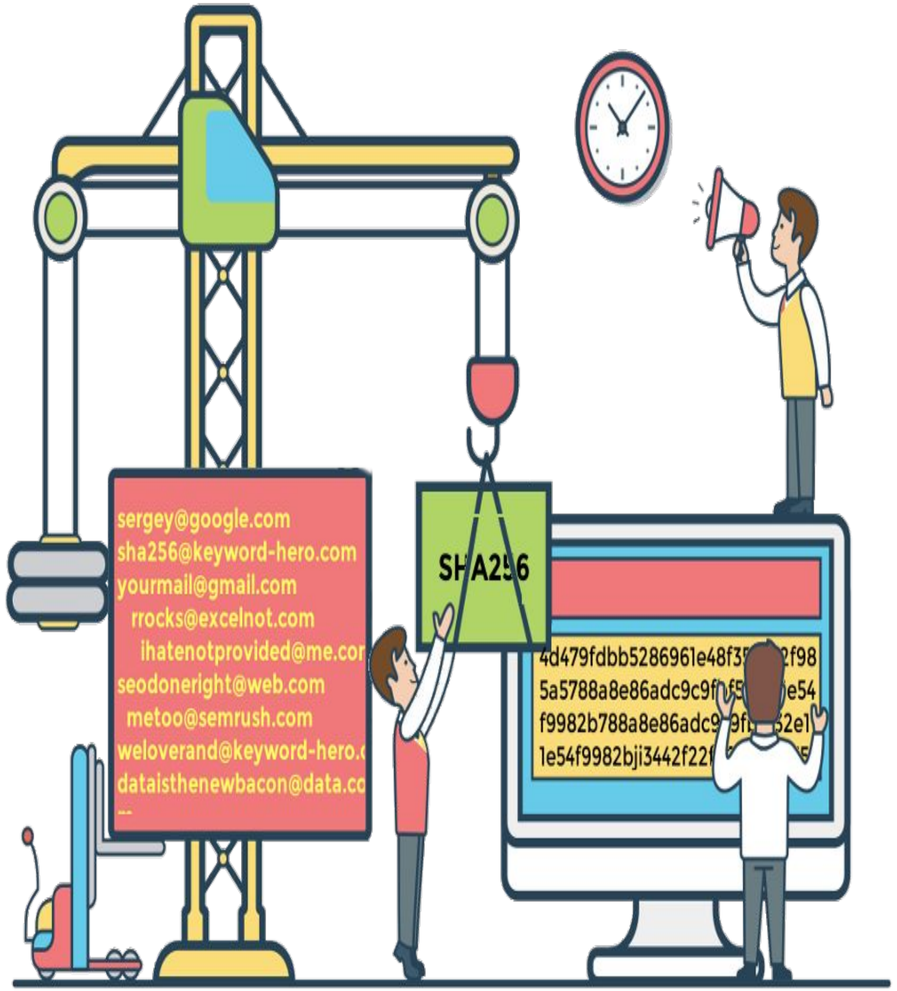
- Bitcoin requires that each block produced has had a significant amount of work invested in its creation
  - This ensures that peers are honest since malicious peers will have to do more work than the honest peers to modify existing blocks
- **The problem**: Find a number (*nonce*) such that
  - Hash(*nonce* + prev block hash + transactions) < target
    - The target is known as the mining difficulty
    - Mining difficulty is updated every 2,016 blocks
    - Time stamps in the block headers ensure that 1,209,600 seconds (2 weeks) elapsed during the creation of those blocks and the target is adjusted accordingly

BLOCKCHAIN
@ COLUMBIA

SHA-256 (Secure Hash Algorithm)

- Generates an unique, seemingly random fixed size **256-bit hash**
- Used in the creation of bitcoin addresses to improve security and privacy
- Maintain 256 bits of state, which splits into eight 32 bit words and gets added
- The result is wired over to the first word of the state and the entire state shifts over

https://www.movable-type.co.uk/scripts/sha256.html



BLOCKCHAIN
@ COLUMBIA

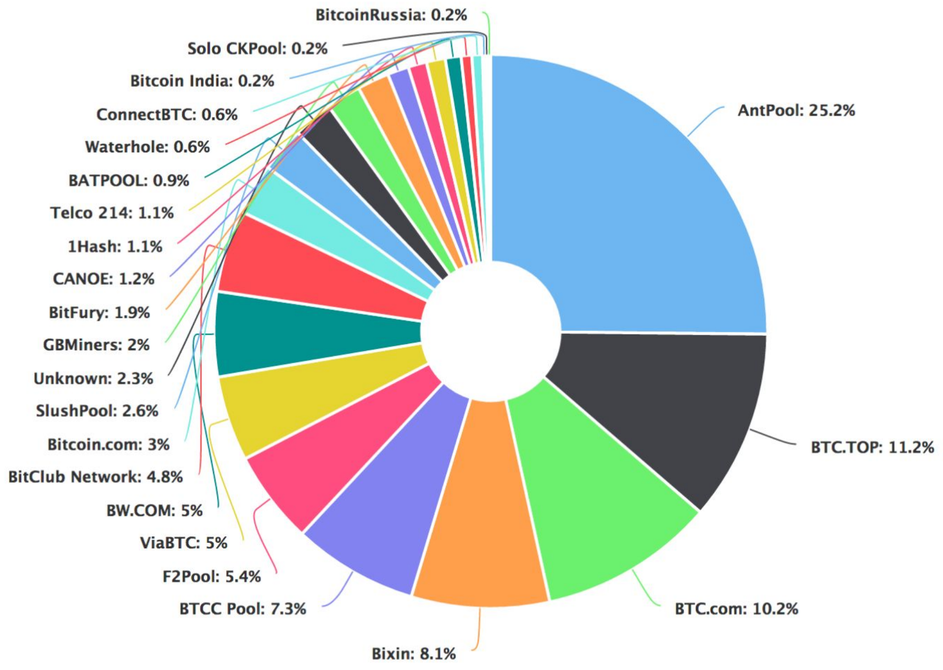
Application-Specific Integrated Circuit

- **CPU → GPU → FPGA → ASIC**
- **ASIC** machines mine at unprecedented speeds, consuming less power than previous mining rigs
  - Bitcoin mining hardware created solely to solve Bitcoin blocks
- ASIC machines are evaluated by 2 factors
  - Hash rate (Gh/s)
  - Efficiency (J/Gh)

BLOCKCHAIN
@ COLUMBIA

Mining Pools

- Group of cooperating miners who agree to share block rewards in proportion to their contributed mining hash power.
- Hashrate distribution amongst the largest mining pools chart
  - Antpool: mines 25% of all blocks
    - Nodes are located in US, EU, CH, etc.
    - No pool fee



BLOCKCHAIN
@ COLUMBIA

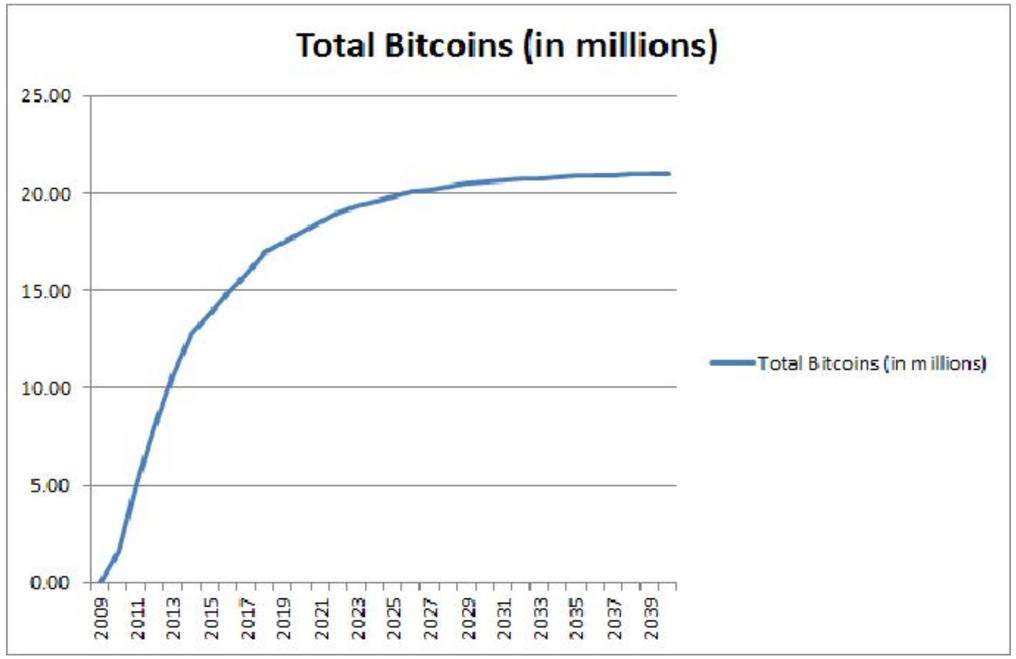
Incentives to Mine

- Receive block reward all the **transfer fees** collected in the block that was produced
  - Each transaction pays a fee that fluctuates depending on the current network congestion
- When the hash for a new block header is found, the node which produced that has is awarded newly minted coins created in the **coinbase transaction**
  - 2009: 50 btc awarded
    - Halved approx. every 4 years

BLOCKCHAIN
@ COLUMBIA

Bitcoin Supply

- Started with 50 BTC block reward
  - Halved approx. every 4 years
- Maximum 21 million bitcoin - last bitcoin will be mined in 2140
  - Supposed to mimic the scarcity of gold



BLOCKCHAIN
@ COLUMBIA

How Does Bitcoin Get Its Value?

Bitcoin's Solution to Financial Problems

- Transactions are recorded on ***a public ledger*** that is maintained by nodes connected to the payment network
- Invalid transactions can easily be identified by checking the transaction history
- Use a proof of work algorithm for randomly selecting node to update the chain
  - Produces financial incentive to not act maliciously
- We can reach consensus if at least 51 percent of the nodes are not malicious

BLOCKCHAIN
@ COLUMBIA

Bitcoin Script (1/4)

- **Script** is a small programming language for performing common operations e.g (verifying signatures and transactions, hashing, conditionals, arithmetic)
- Script is not Turing Complete - no loops!
  - Prevents infinite loops
  - Ensures finite memory requirements
- Each UTXO includes script that how the transaction can be spent
  - If the script included in the transaction runs to completion - it is considered valid

Bitcoin Script (2/4)

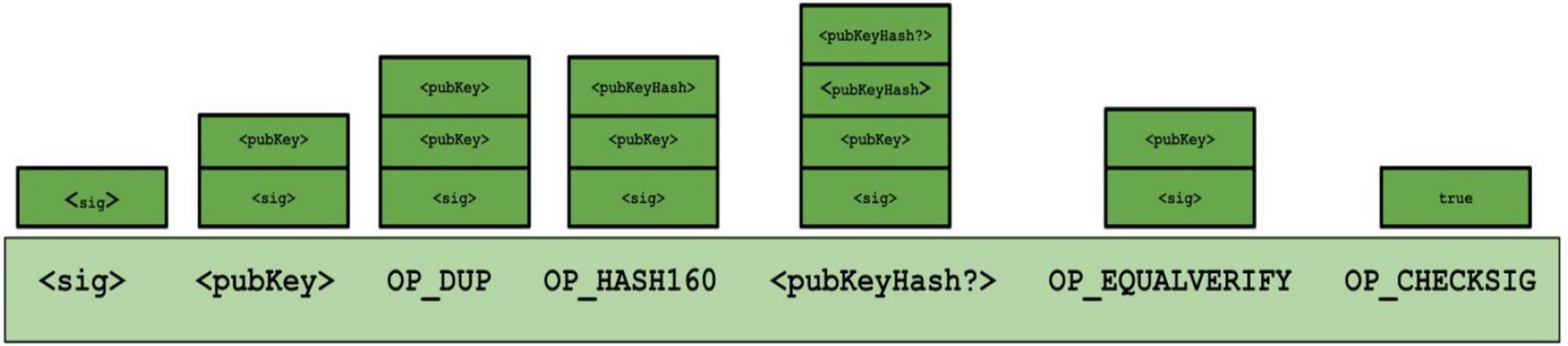
The most common script involves the simple spending a transaction output.

- Each instruction is known as opcode
  - There are 256 opcodes, 15 are disabled due to bugs, 75 are reserved but unused
- Data is pushed onto the stack, while opcode operations often remove data from the stack

OP_DUP	Duplicates the top item on the stack
OP_HASH160	Hashes twice: first using SHA-256 and then RIPEMD-160
OP_EQUALVERIFY	Returns true if the inputs are equal. Returns false and marks the transaction as invalid if they are unequal
OP_CHECKSIG	Checks that the input signature is a valid signature using the input public key for the hash of the current transaction
OP_CHECKMULTISIG	Checks that the *k* signatures on the transaction are valid signatures from *k* of the specified public keys.

Bitcoin Script (3/4)

Script execution for spending transaction output



Bitcoin Script (4/4)

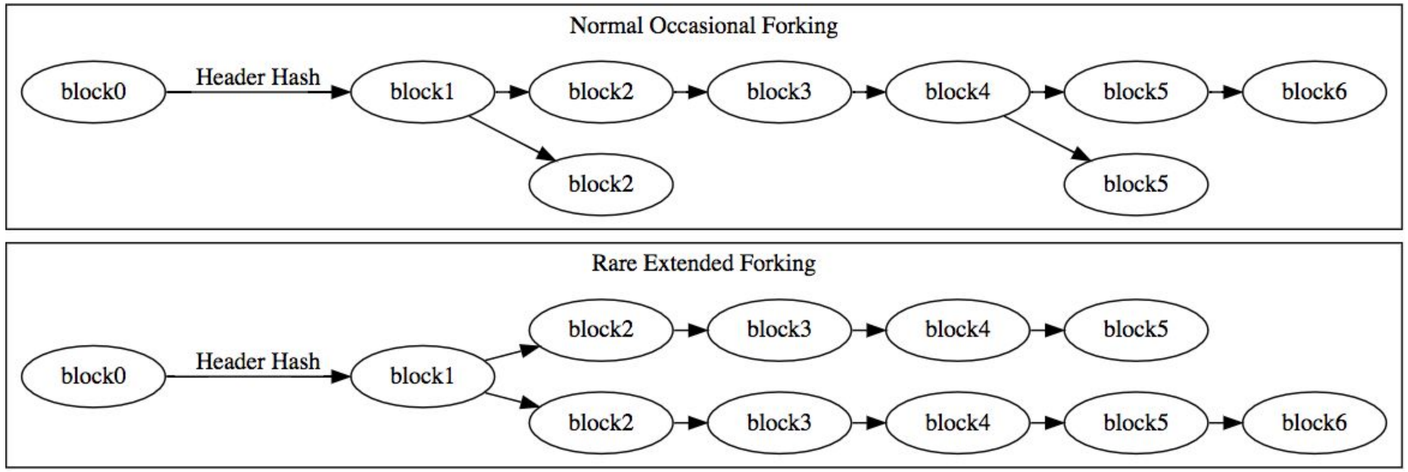
- Bitcoin script also includes the **Pay to Script Hash** (P2SH) transaction type
  - The sender specifies a script hash rather than a public key hash
  - The script hash is defined by the receiver
  - Allows sellers to receive money using complex scripts without confusing the payer e.g., 2-of-2 multisig
- To redeem coins, the receiver must specify the script that has the given hash, and the input data that is required for the script to evaluate to true

Security of the Bitcoin Network

BLOCKCHAIN
@ COLUMBIA

Chain Reorganizations

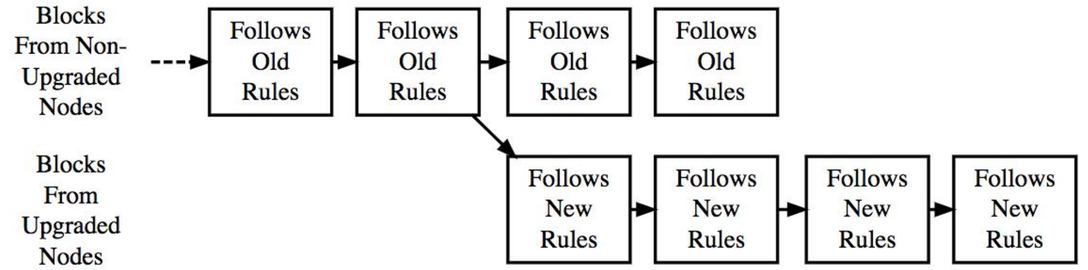
- **Forks** occur when two or more miners find the hash to next block header at roughly the same time.
- Each node accepts the block it sees first
- When another block is appended, the longest sequence of blocks known as the **longest chain** determines the **reorganization**
  - The blocks not part of the longest chain are known as **orphaned blocks**



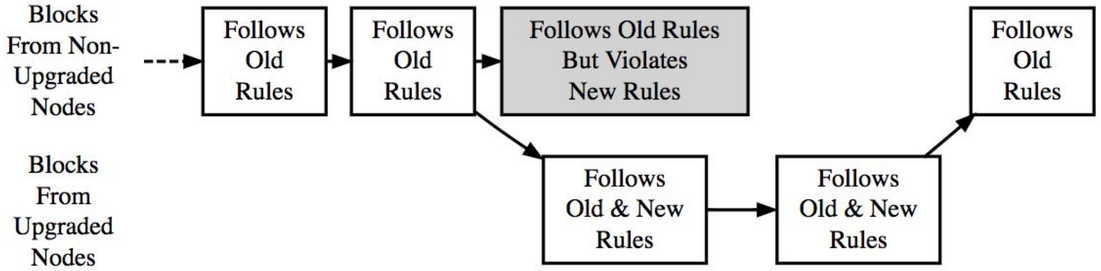
BLOCKCHAIN @ COLUMBIA

Soft Fork vs. Hard Fork

- Sometimes, changes in the consensus protocol are sent to the network to introduce new features or prevent network abuse
- However, not all the nodes receive the upgrade at the time of a new block creation creating either a ***hard fork*** or a ***soft fork***
- **Hard Fork**
  - Blocks with new consensus rules accepted by upgraded nodes and rejected by non-upgraded nodes
- **Soft Fork**
  - Blocks violating new consensus rules rejected by upgraded nodes but accepted by non-upgraded nodes



A Hard Fork: Non-Upgraded Nodes Reject The New Rules, Diverging The Chain



A Soft Fork: Blocks Violating New Rules Are Made Stale By The Upgraded Mining Majority

BLOCKCHAIN @ COLUMBIA

Attacks on the Network

- Different Types of Attacks
  - **51% Attack**
    - Miners control more than 50% of the network's mining hash rate
  - **Selfish Mining Strategy**
    - Mining pool of at least ~25% obtains a revenue larger than its mining power
  - **Sybil Attack**
    - Attacker fills the network with nodes that they control

BLOCKCHAIN
@ COLUMBIA

Future: Lightning Network

- A payment system on top of Bitcoin
- Bitcoin has a scalability issue:
  - cannot send more than 7 transactions per second
- Bitcoin transaction fee can be significant for low transactions
- Lightning network works on direct channels and a joint signature account on the blockchain
- Think of it like a bank with only two people
- Channel can exist indefinitely between the two people till someone pulls out