The Bitcoin Network

How Bitcoin works?

Bitcoin network

A standard p2p network:

- Nodes connect and disconnect
- New nodes enter the network
- There is no perfect connectivity
- Nodes listen to the messages by their neighbours
- Nodes transmit messages to their neighbours

Bitcoin network

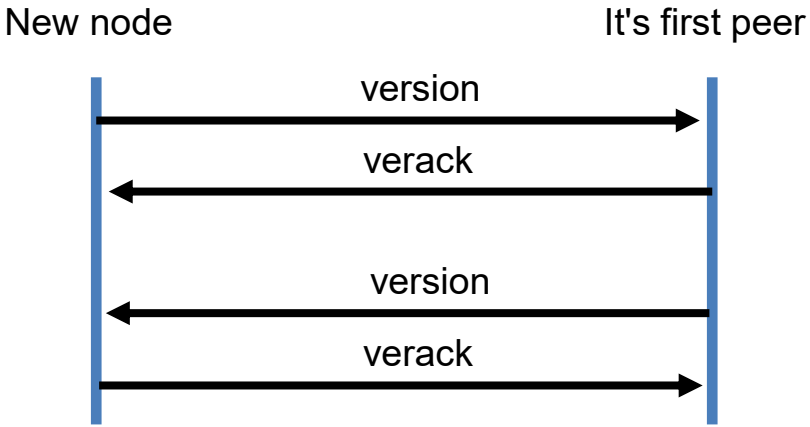
How does a new node get to have neighbours?

- Static DNS Seed
- DNS Seed BIND (Berkeley Internet Name Daemon)
- A specific IP of a node already known to the new node

Bitcoin network

Basic messages to establish a connection:

- *version*
- *verack*



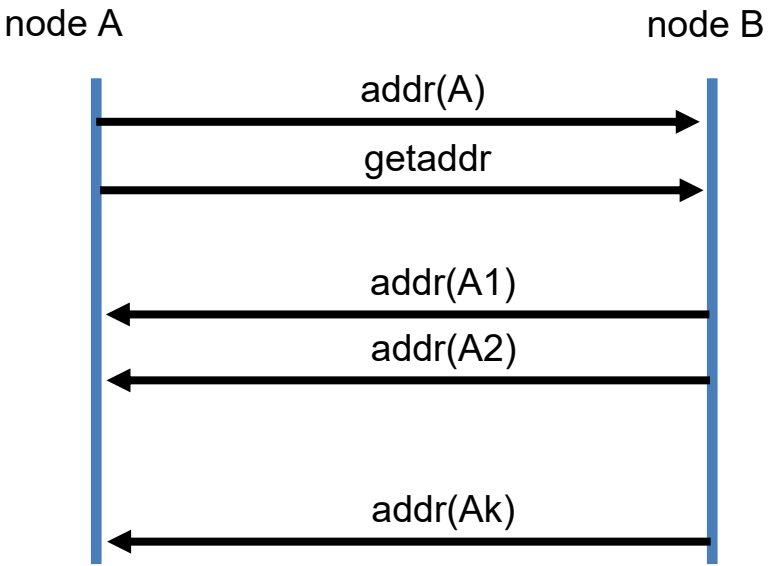
Bitcoin network

Discovering new peers:

- A connects to B
- B will pass on the address of A to its peers
- A can also ask B for the addresses of its peers
- B replies with these addresses

Bitcoin network

Discovering new peers:



Bitcoin network

Nodes enter and leave the network:

- A dynamic process
- Node stores its peers and upon enering again tries to connect to them
- If there are no messages from a node for 30 min, a message is sent
- If there are no messages for 90 min, the node is disconnected (one less edge)

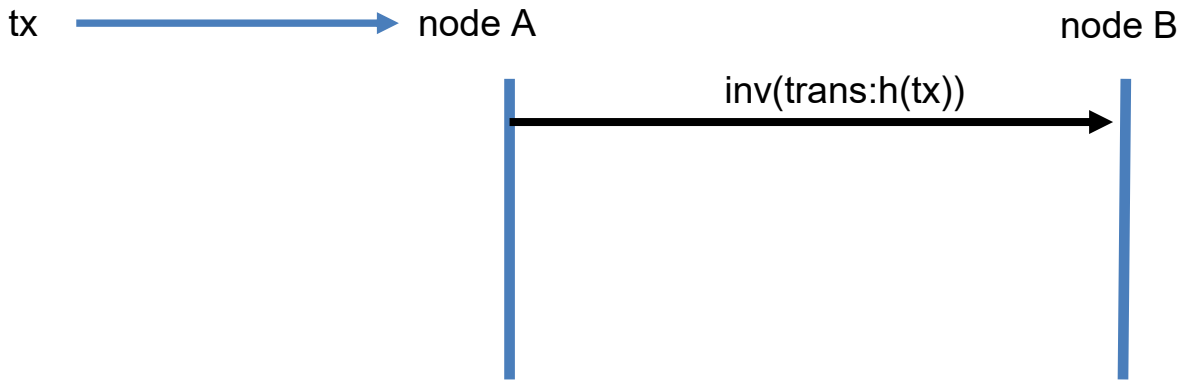
- It is recommended to have no more than 100 peers
- Too many connections put pressure onto the network

Bitcoin network

Basic communication:

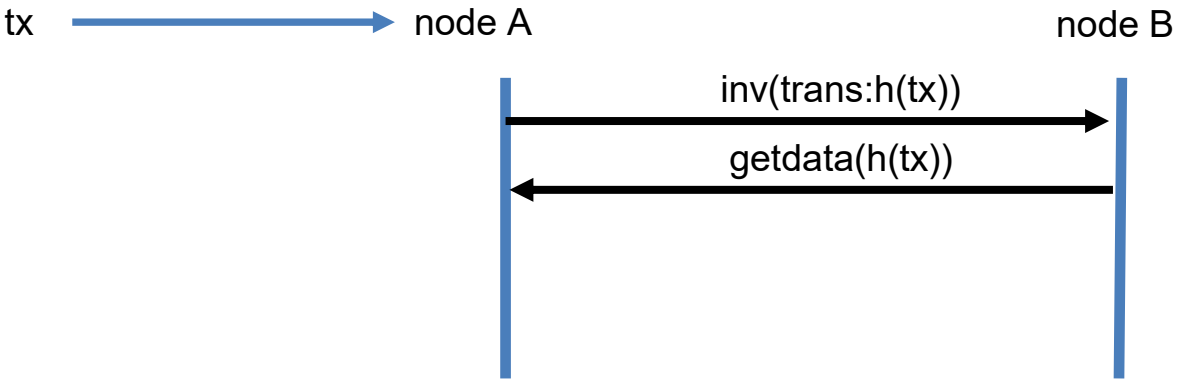
Bitcoin network

Basic communication:



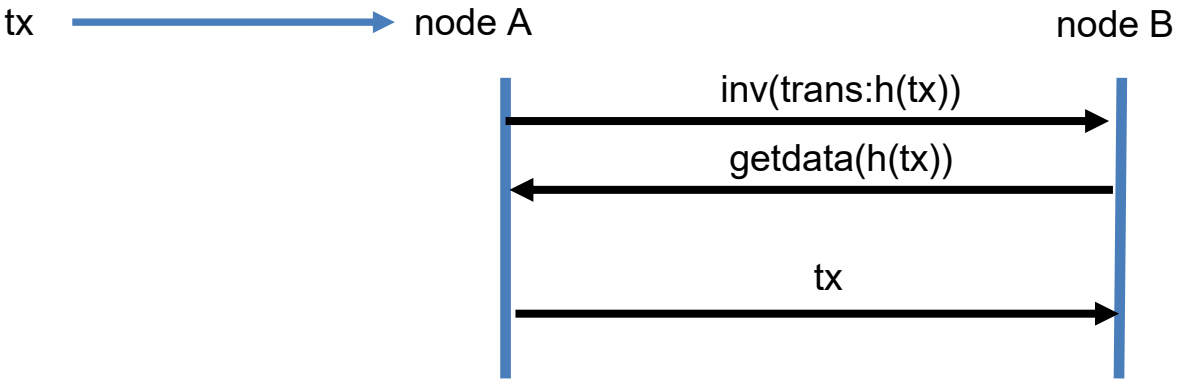
Bitcoin network

Basic communication:



Bitcoin network

Basic communication:

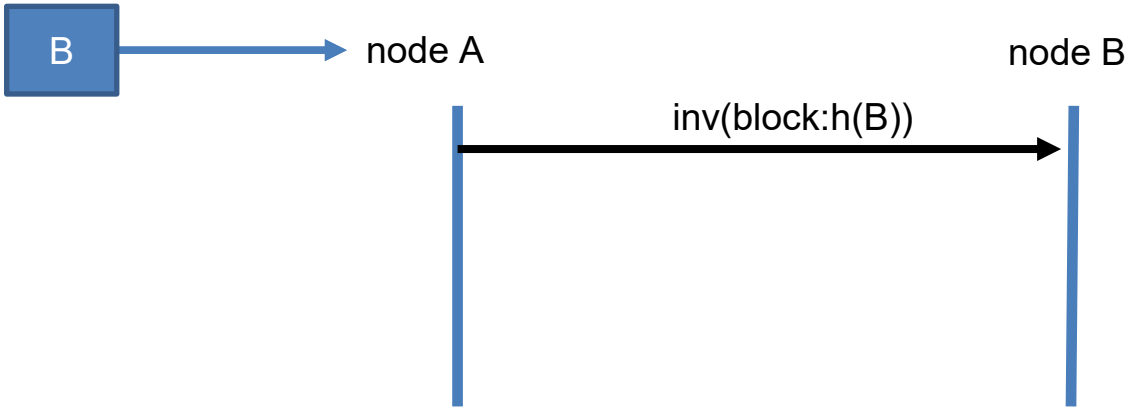


Bitcoin network

Basic communication (blocks):

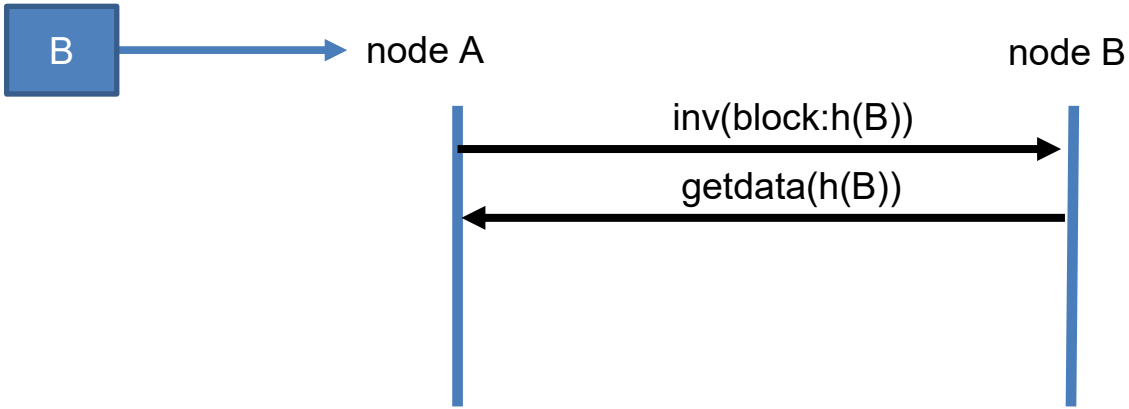
Bitcoin network

Basic communication (blocks):



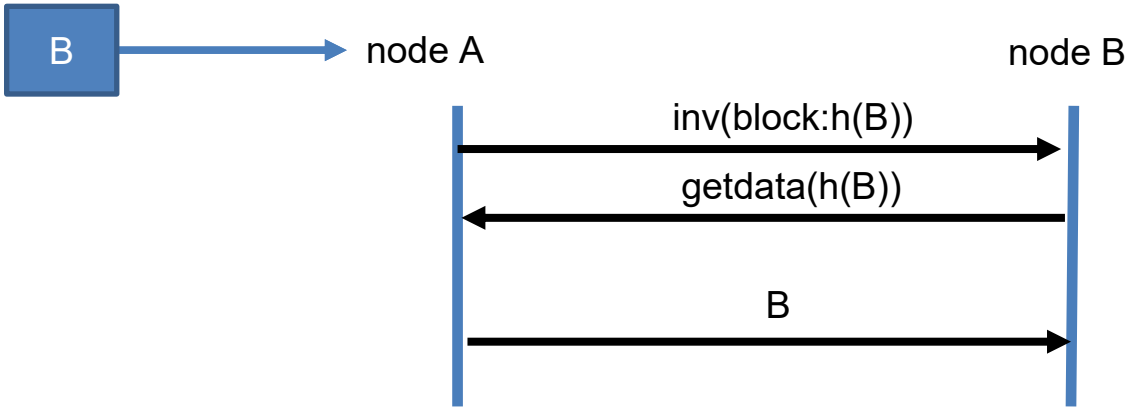
Bitcoin network

Basic communication (blocks):



Bitcoin network

Basic communication (blocks):



Bitcoin network

Full nodes:

- Store the entire blockchain
- Verify all the transactions and blocks

- How does a full node obtain the blockchain upon entering the network?

Bitcoin network

Up to version 0.9.3 of BitcoinCore

How does node X obtain the blockchain?

- Always starts with the genesis block (hardcoded in its software)
- Other blocks are received by its peers on the network

- First message sent/received by X: *version (contains the field BestHeight)*
- X sends *getblocks* to its peer with the highest BestHeight (call this node Y)
- The message *getblocks* has the hash of the last block in the blockchain of X
- Y identifies the first 500 blocks that X does not yet have
- Y sends the message *inv* with the hashes of these 500 blocks
- To obtain a block, X sends to Y *getdata(hash)* message
- Maximum of 500 *getdata* can be active per connection

Initial block download

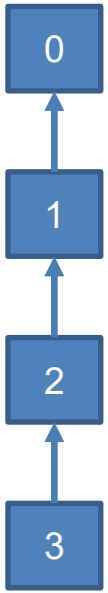
node A

node B

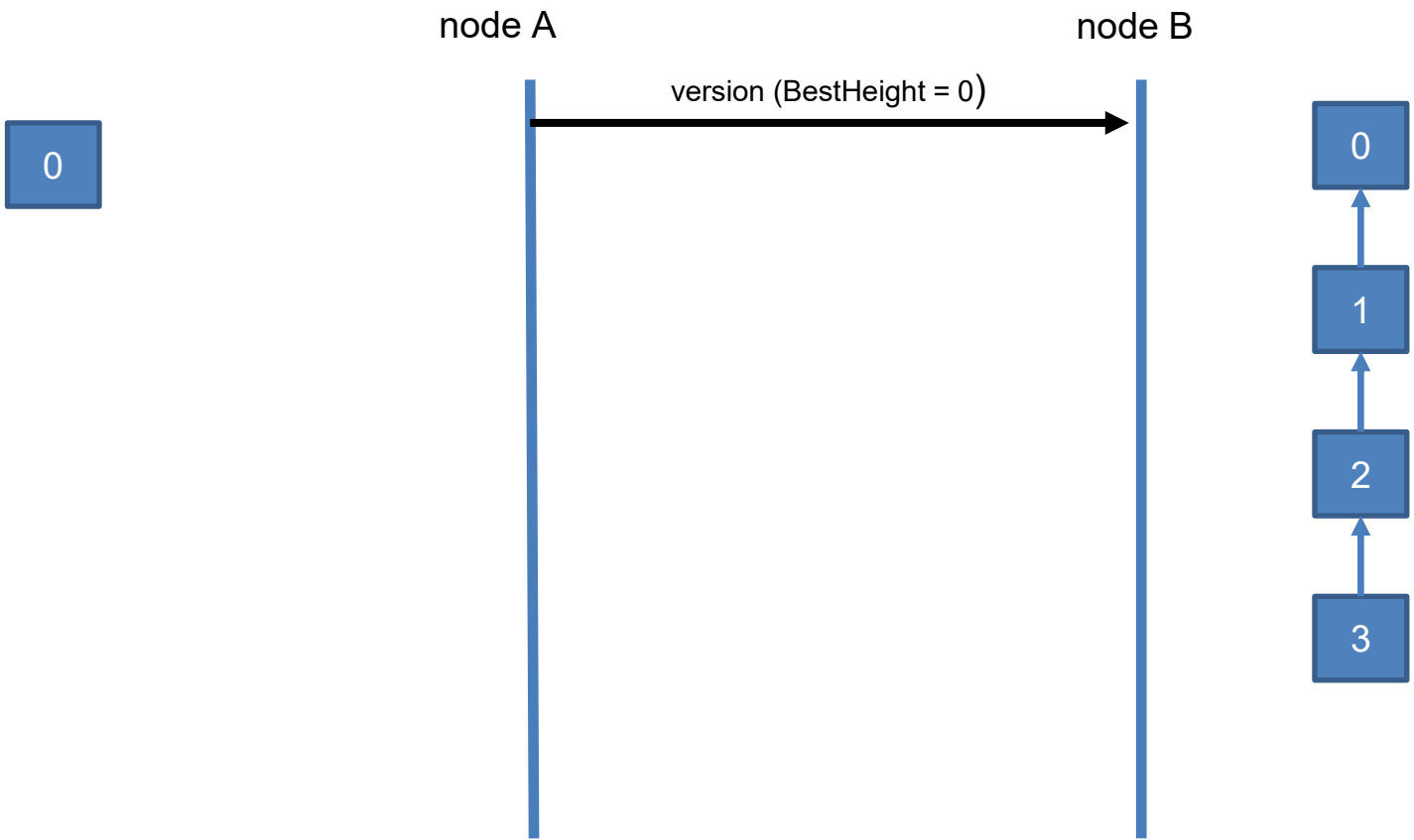
Initial block download

node A

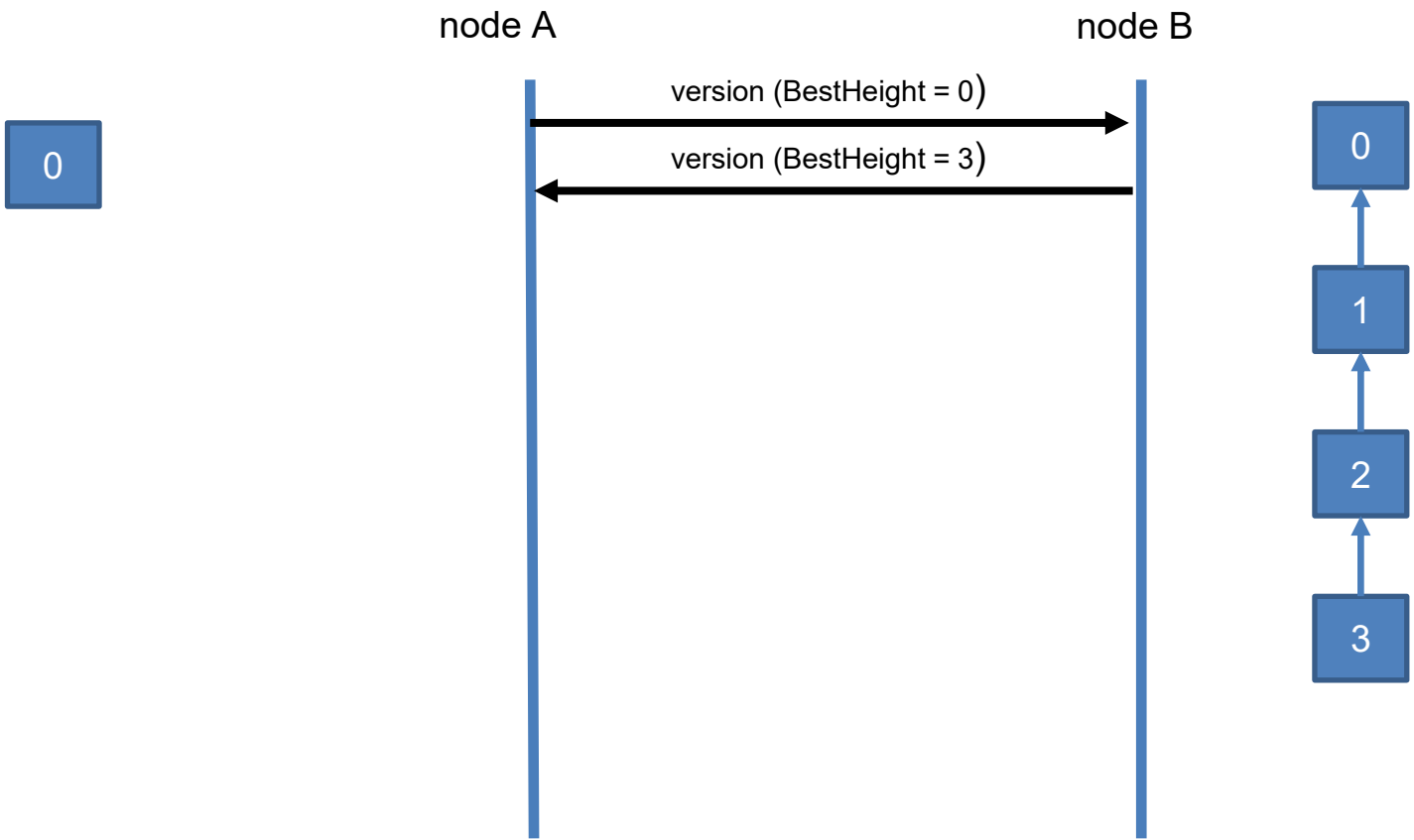
node B



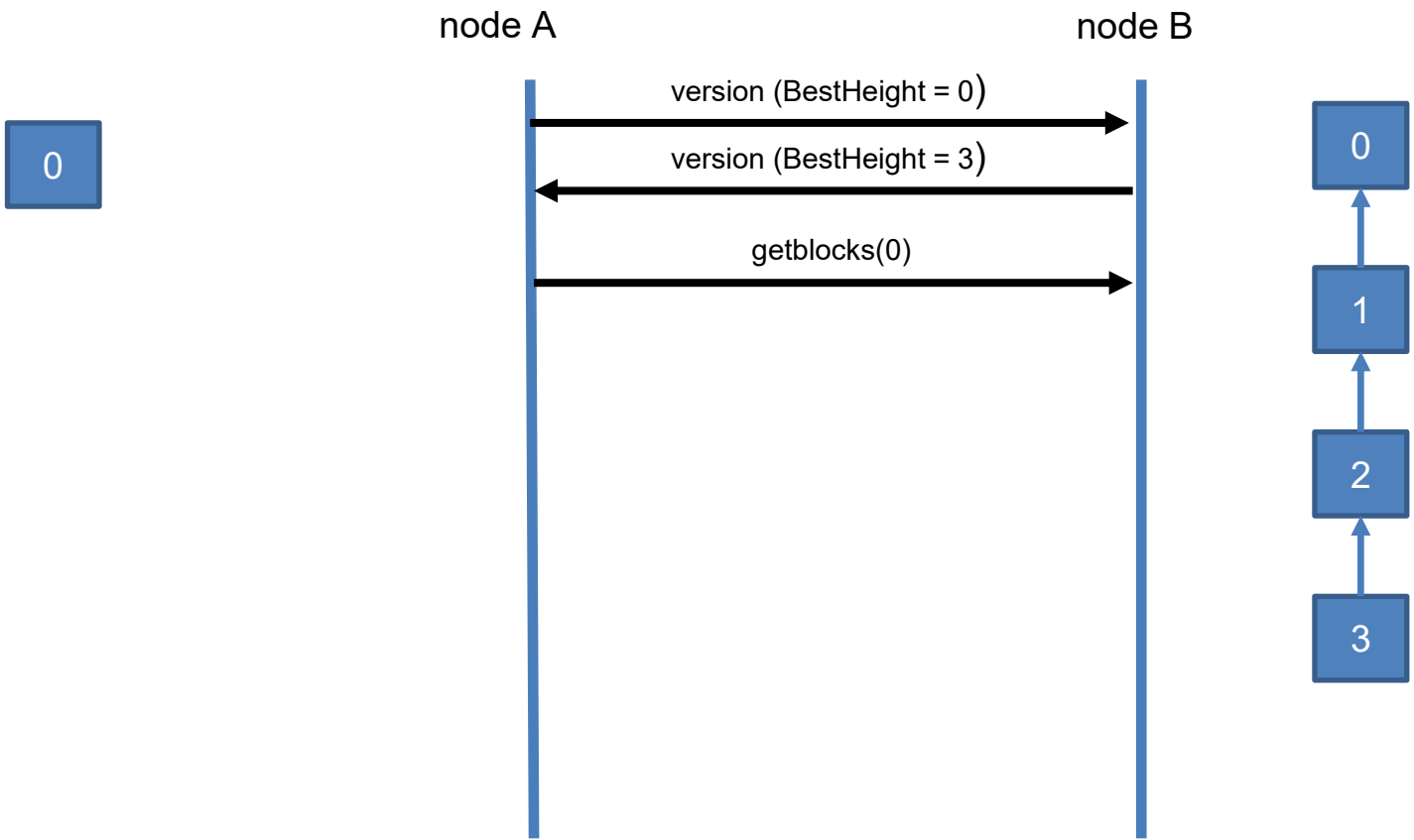
Initial block download



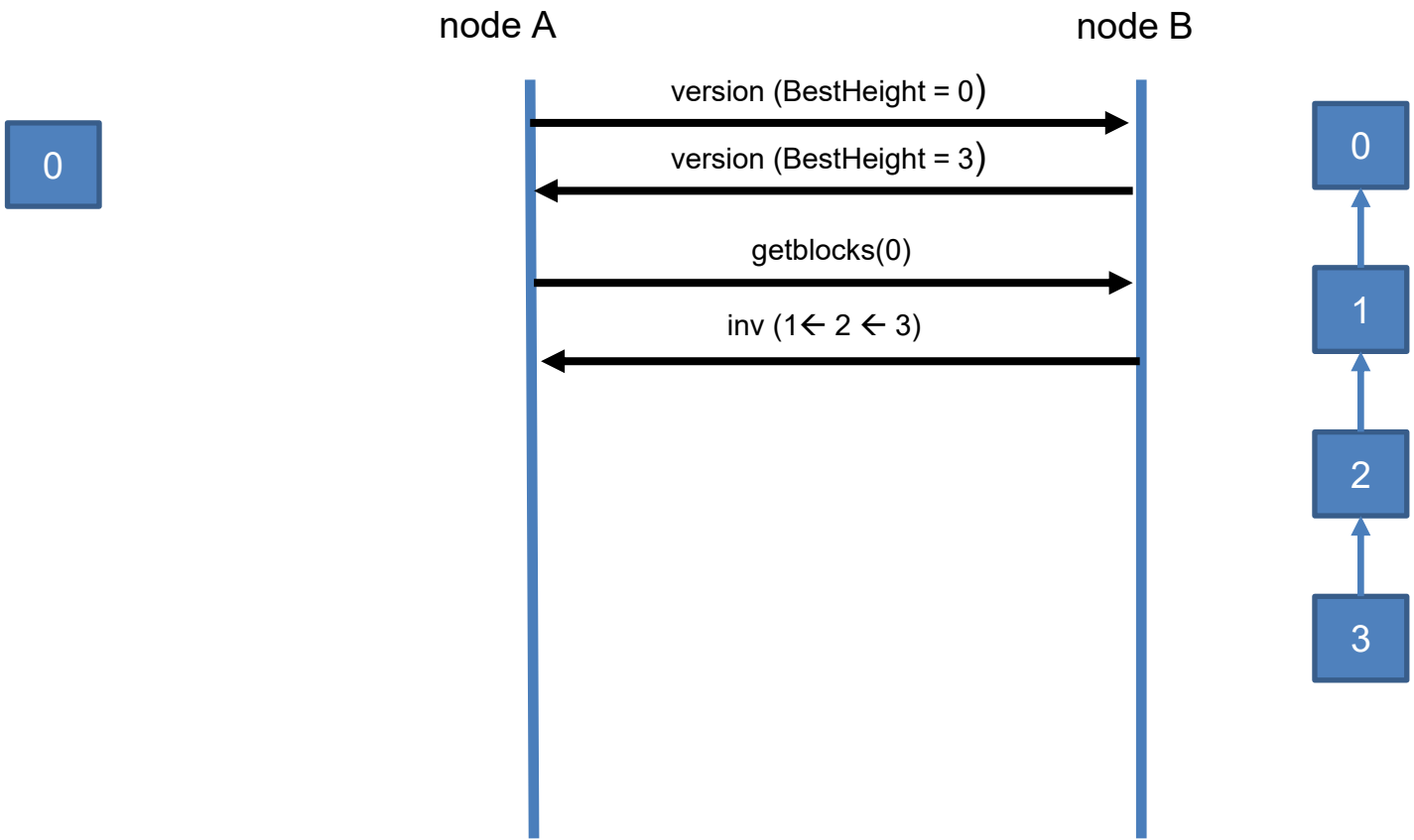
Initial block download



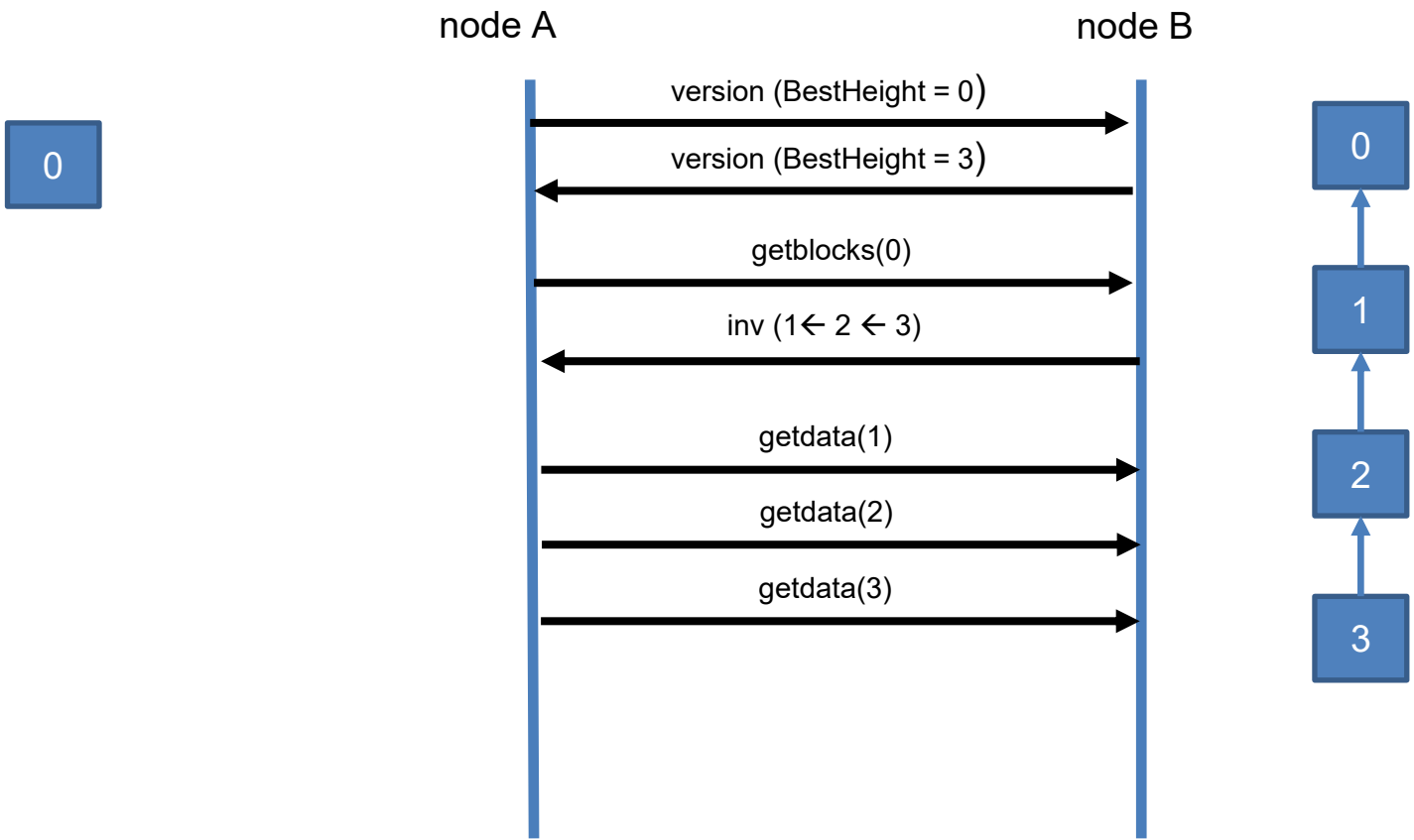
Initial block download



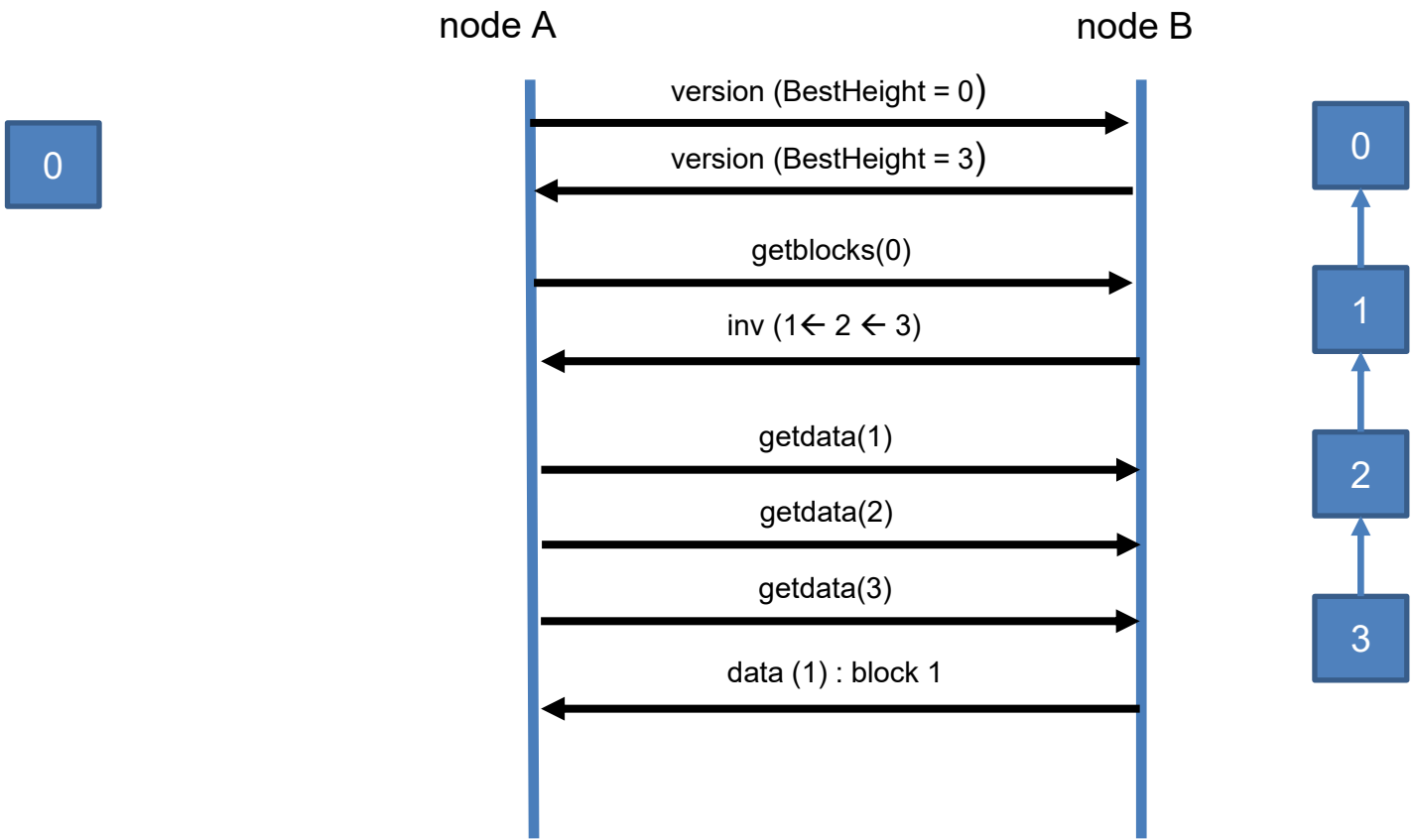
Initial block download



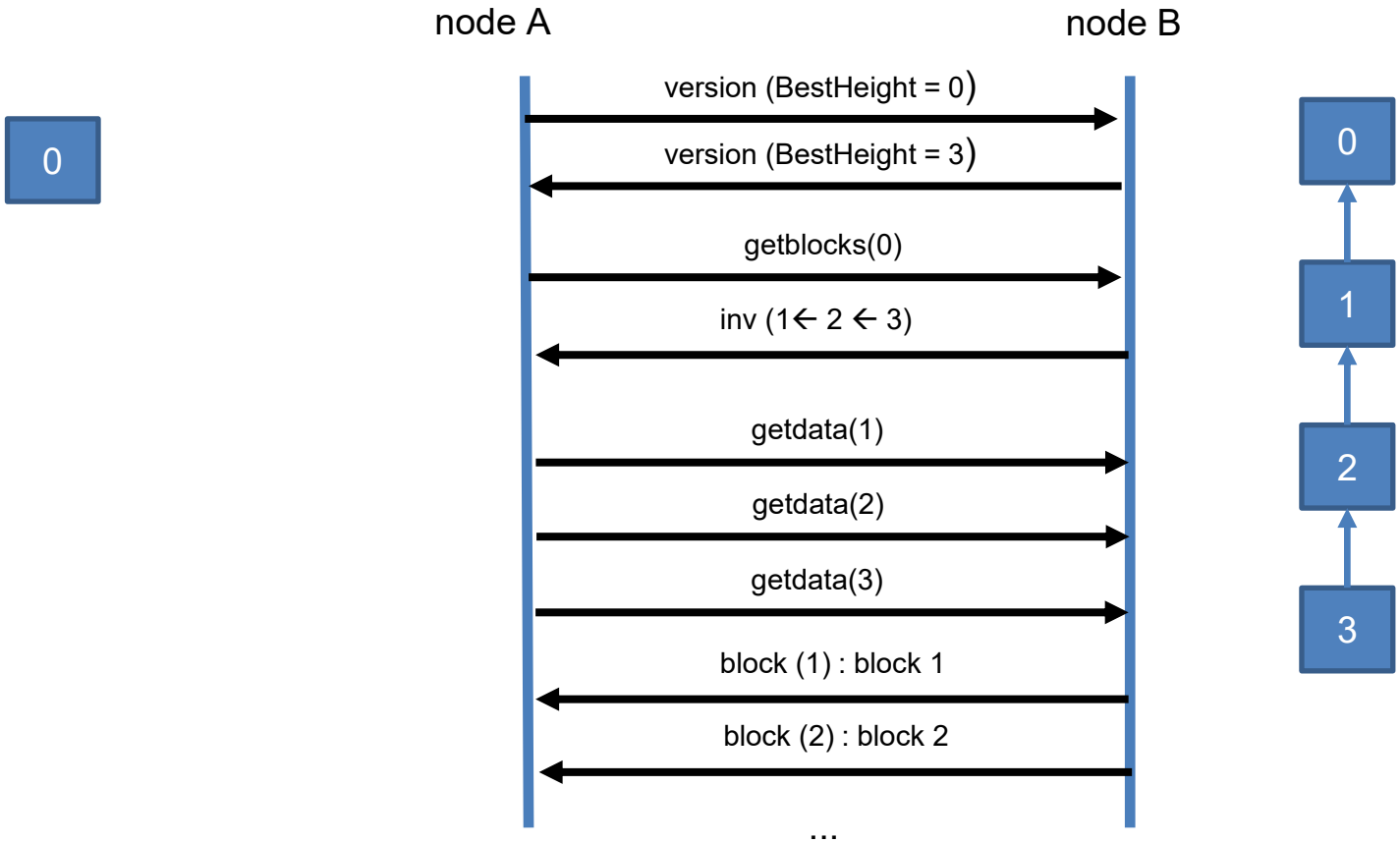
Initial block download



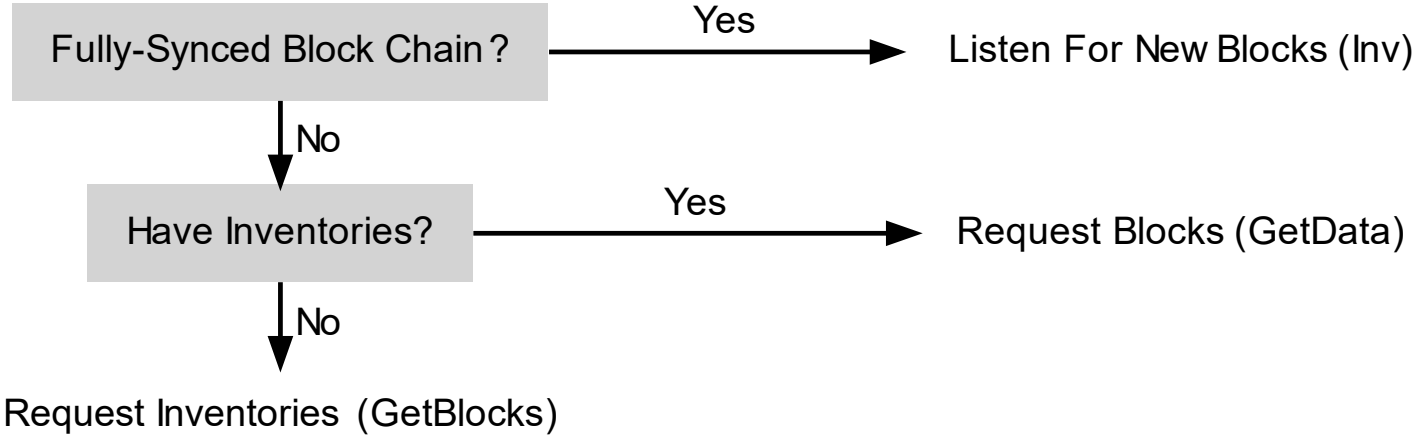
Initial block download



Initial block download



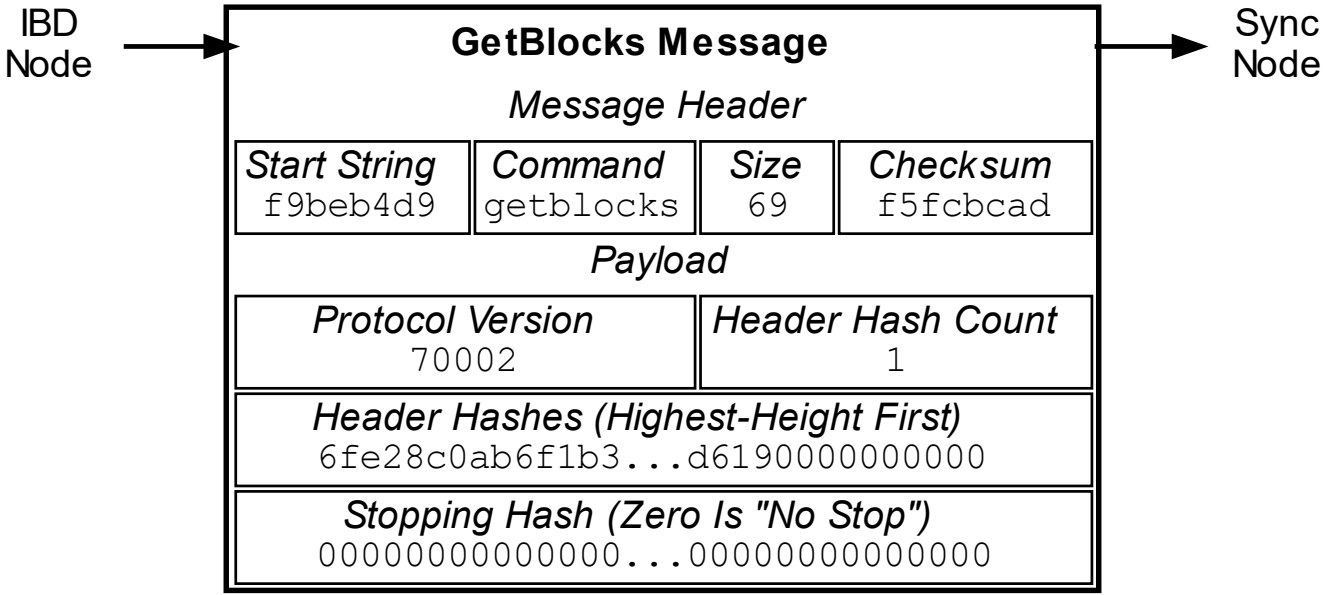
In reality (IBD)



Overview Of Blocks-First Initial Blocks Download (IBD)

https://bitcoin.org/en/developer-guide#peer-discovery

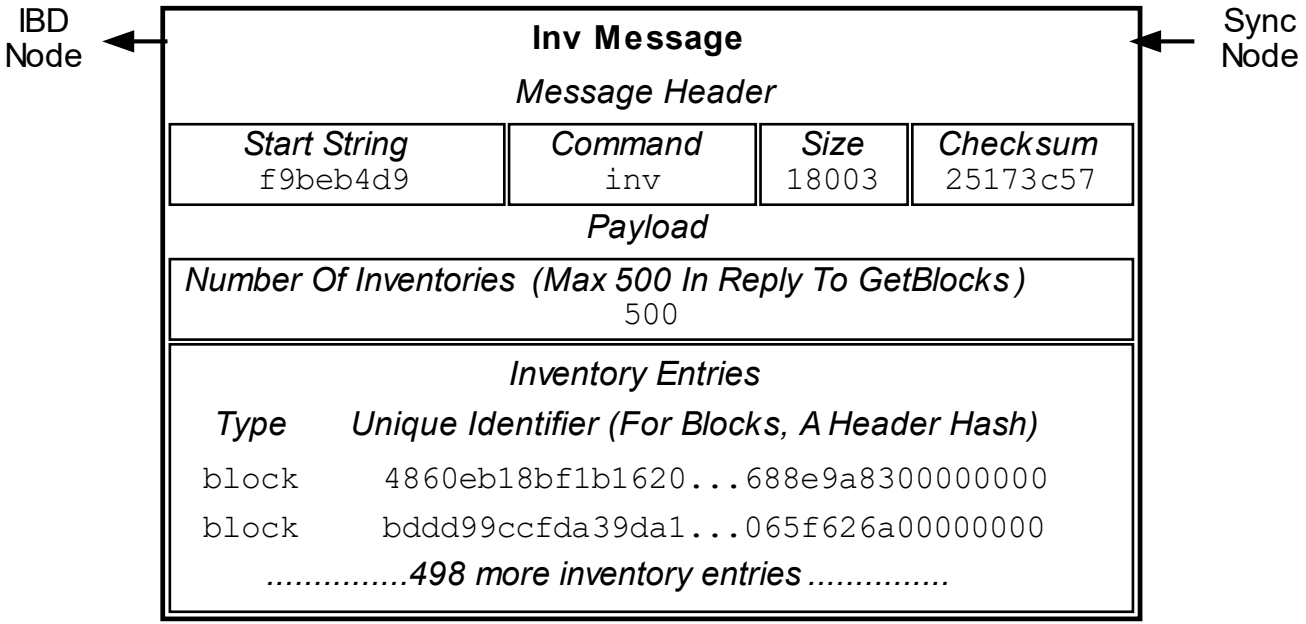
In reality (IBD)



First getblocks message sent from Initial Blocks Download (IBD) node

https://bitcoin.org/en/developer-guide#peer-discovery

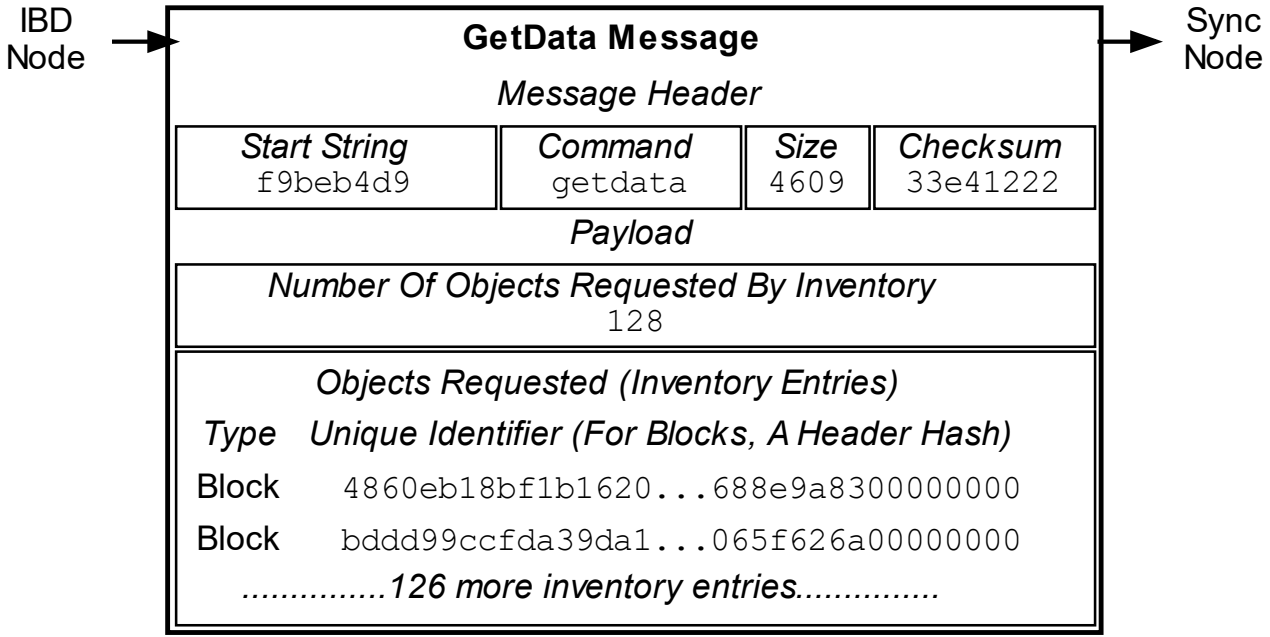
In reality (IBD)



First inv message reply sent to Initial Blocks Download (IBD) node

https://bitcoin.org/en/developer-guide#peer-discovery

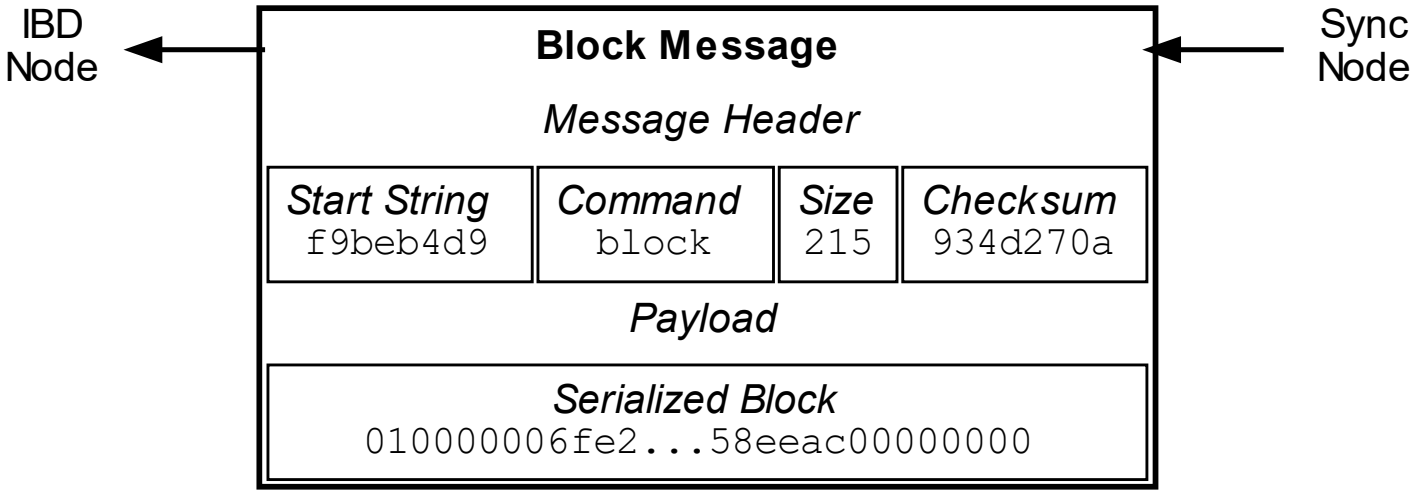
In reality (IBD)



First getdata message sent from Initial Blocks Download (IBD) node

https://bitcoin.org/en/developer-guide#peer-discovery

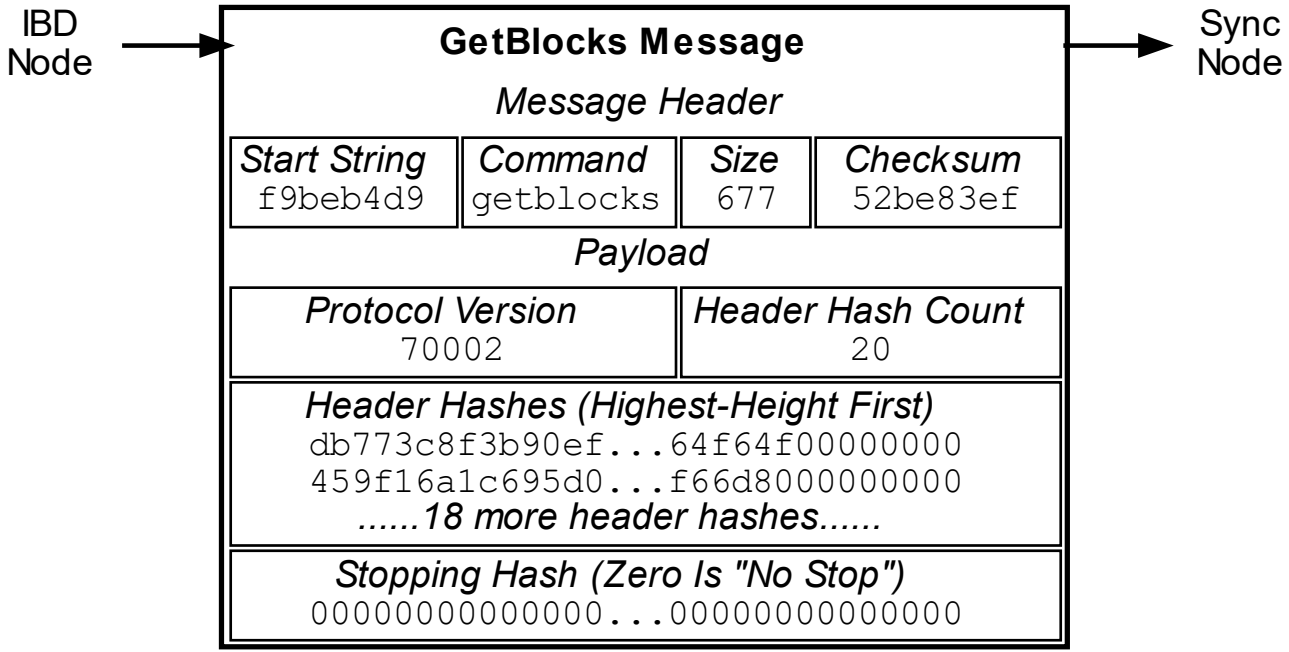
In reality (IBD)



First block message sent to Initial Blocks Download (IBD) node

https://bitcoin.org/en/developer-guide#peer-discovery

In reality (IBD)



Second getblocks message sent from Initial Blocks Download (IBD) node

https://bitcoin.org/en/developer-guide#peer-discovery

SPV nodes

Simple payment verification (SPV):

- A typical user is only interested in transactions using her own address
- A typical use of a Bitcoin wallet

- SPV nodes do not store the entire blockchain
- They store only the block headers (80 bytes per block = 4.2MB per year)
- Much smaller than a full node
- But they can not validate all the UTXOs

SPV nodes

An SPV node can verify only two things:

1. That a transaction *tx* belongs to the block *blockX*
2. That a block *blockX* has k confirmations

SPV nodes

How does an SPV node verify that tx belongs to blockX?

- An SPV node store the block headers

Field	Purpose	Updated when...	Size (Bytes)
Version	Block version number	You upgrade the software and it specifies a new version	4
hashPrevBlock	256-bit hash of the previous block header	A new block comes in	32
hashMerkleRoot	256-bit hash based on all of the transactions in the block	A transaction is accepted	32
Time	Current timestamp as seconds since 1970-01-01T00:00 UTC	Every few seconds	4
Bits	Current target in compact format	The difficulty is adjusted	4
Nonce	32-bit number (starts at 0)	A hash is tried (increments)	4

SPV nodes

How does an SPV node verify that tx belongs to blockX?

- An SPV node store the block headers
- An SPV node A asks a full node B for a certificate for *tx*

- And validates the certificate against *hashMerkleRoot*

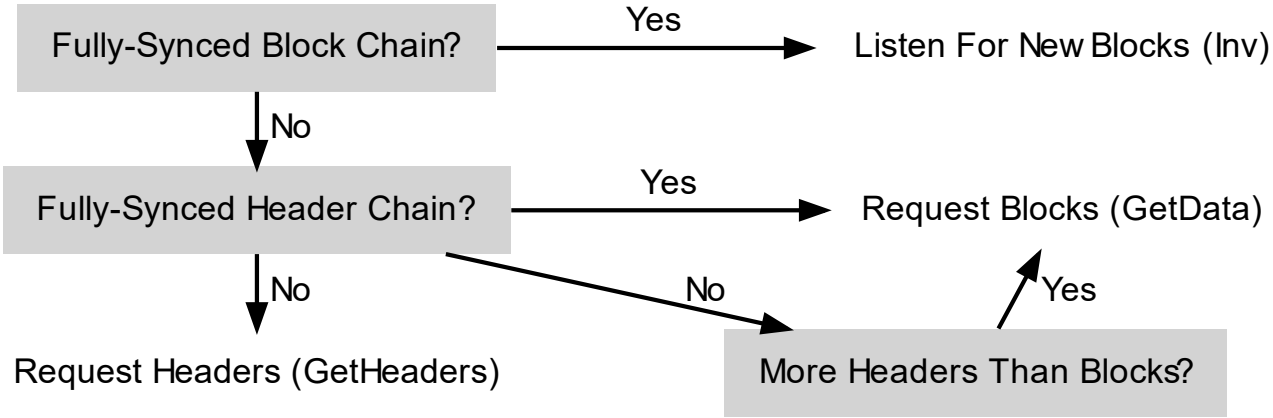
SPV nodes

How does an SPV node A verify that tx has k confirmations?

- A has the headers up until the current end of blockchain

- Can A verify that the headers are valid?

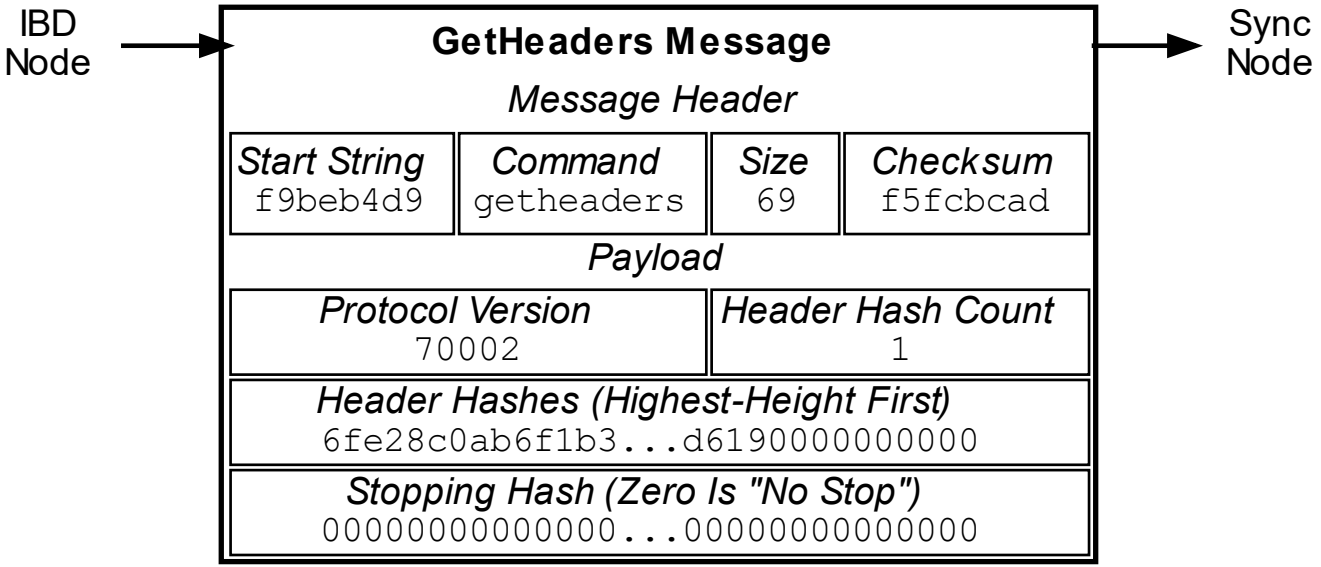
IBD for an SPV node



Overview Of Headers-First Initial Blocks Download (IBD)

https://bitcoin.org/en/developer-guide#peer-discovery

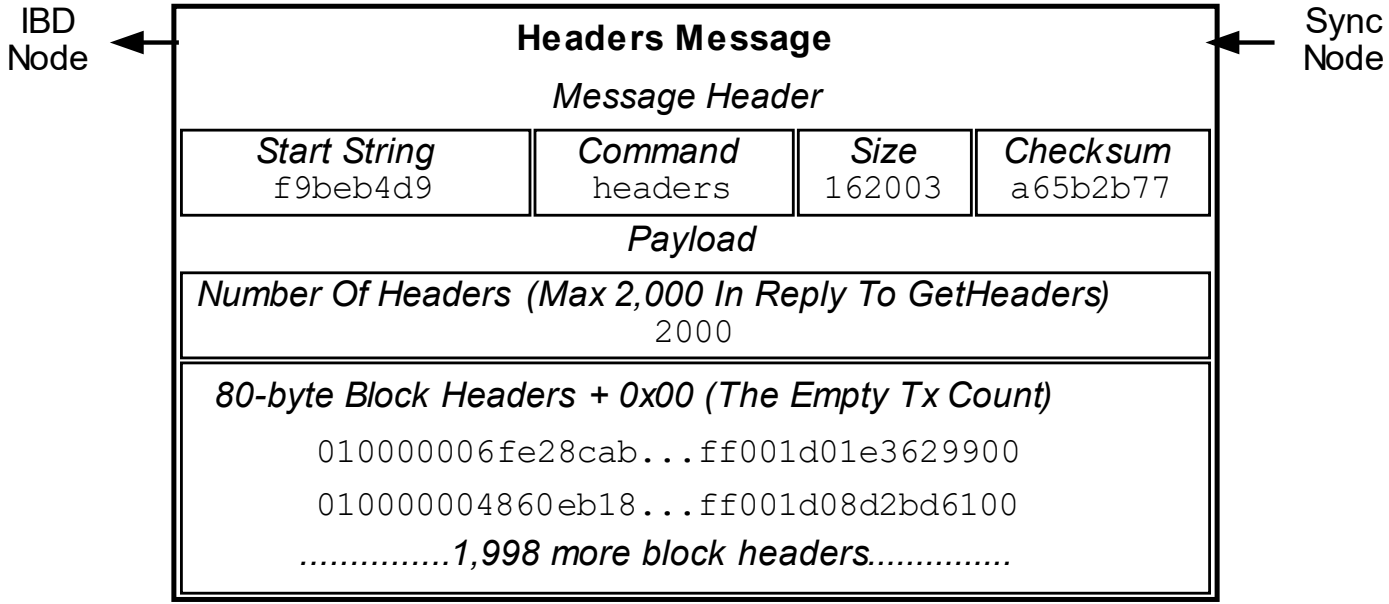
IBD for an SPV node



First getheaders message sent from Initial Blocks Download (IBD) node

https://bitcoin.org/en/developer-guide#peer-discovery

IBD for an SPV node



First headers message reply sent to Initial Blocks Download (IBD) node

https://bitcoin.org/en/developer-guide#peer-discovery

Vulnerabilities of SPV nodes

First vulnerability: partial DOS attack

- If a full node is malicious can it pretend that *tx* belongs to *blockX*?
- Can it pretend that *tx* does not belong to *blockX*?

Solution:

- The SPV node should use more than one full node
- Merkle trees should be ordered

Vulnerabilities of SPV nodes

Second vulnerability: loss of privacy

- If the SPV node queries node B a lot
- B will know all the addresses of A
- It can track A
- It can prohibit A to broadcast transactions

- How to solve this?

Bloom filters

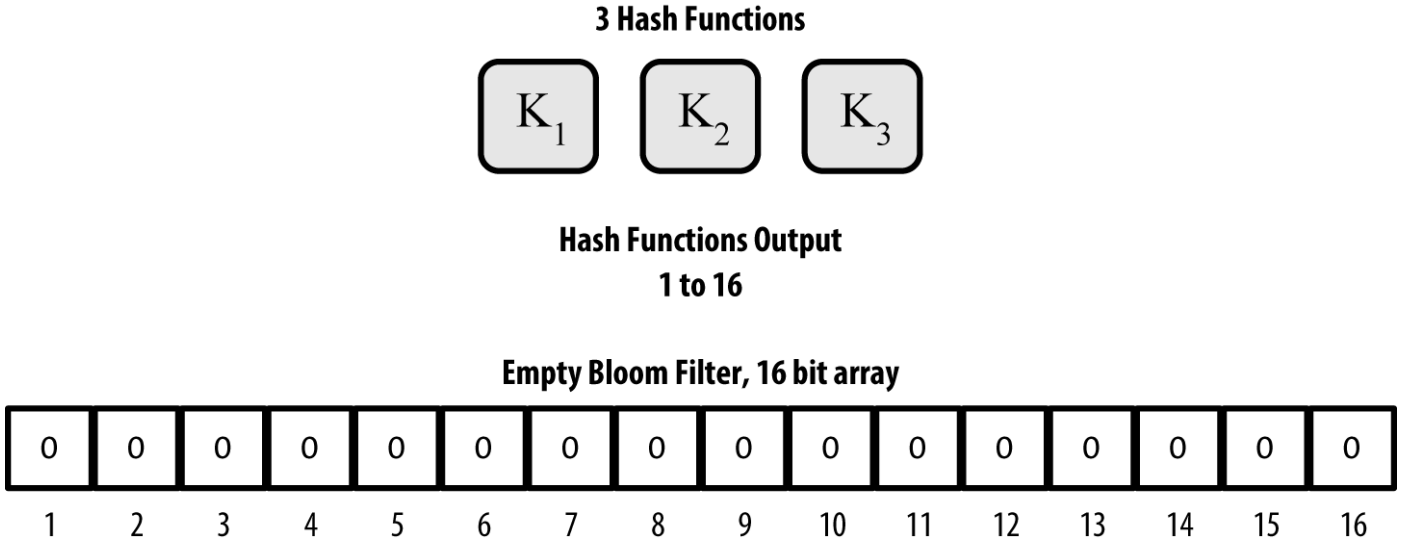
A probabilistic search filter:

- The filter starts empty
- Elements are added to the filter

- One wishes to check if an element was added to the filter

- Can give false positives
- Can not give false negatives

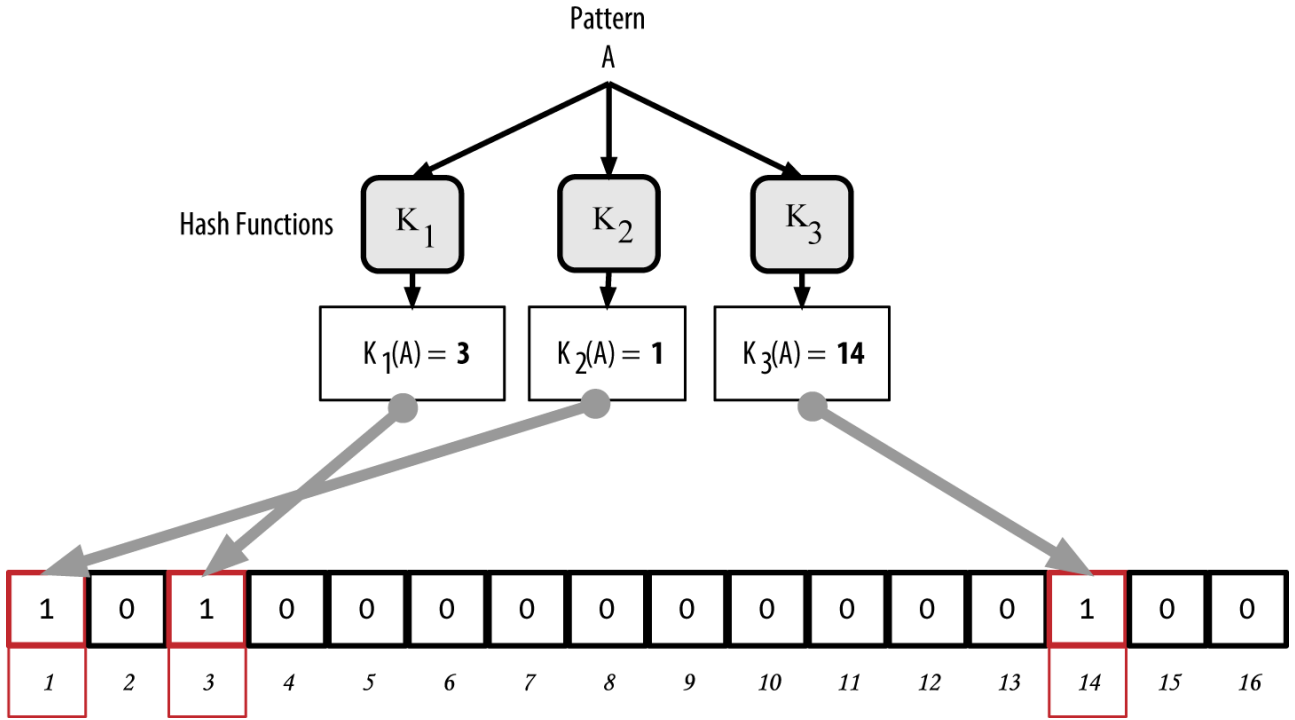
Bloom filters



https://github.com/bitcoinbook/bitcoinbook/blob/develop/ch08.asciidoc

Bloom filters

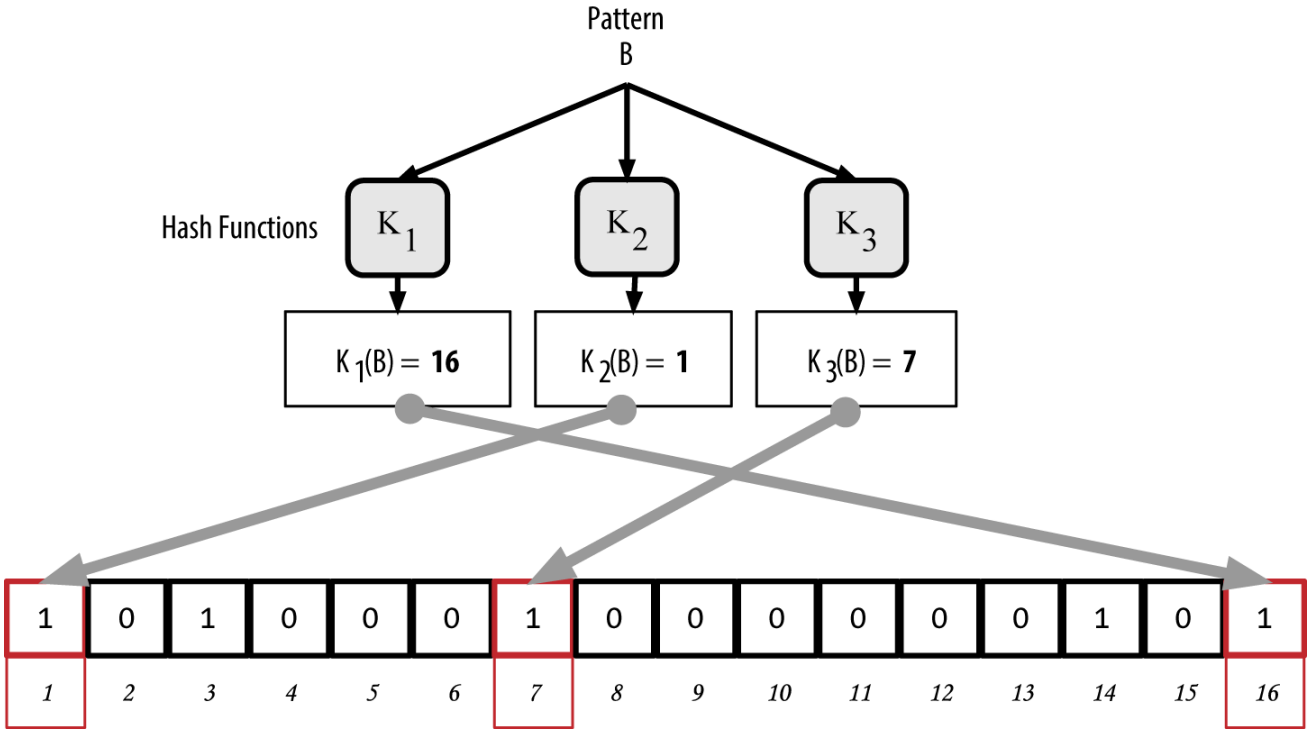
Inserción



https://github.com/bitcoinbook/bitcoinbook/blob/develop/ch08.asciidoc

Bloom filters

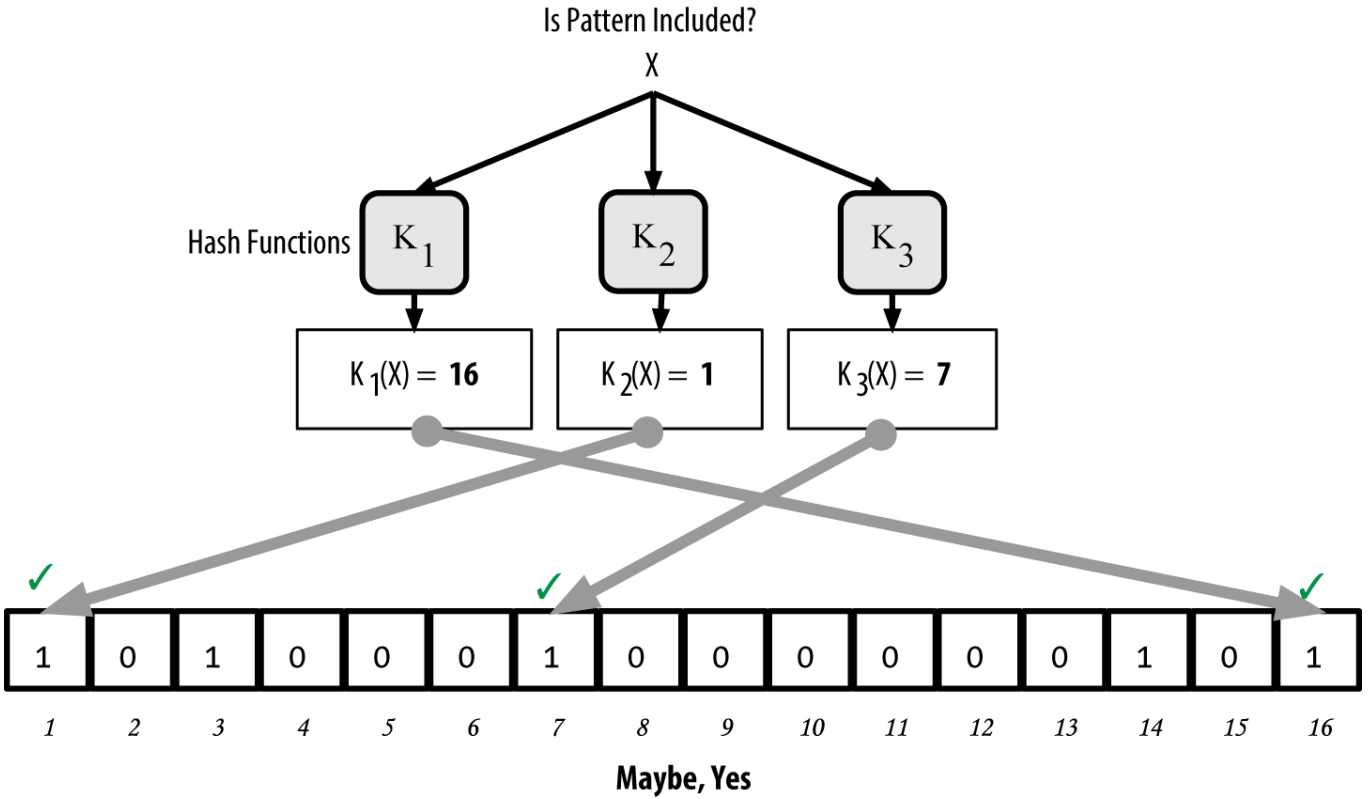
Insertion



https://github.com/bitcoinbook/bitcoinbook/blob/develop/ch08.asciidoc

Bloom filters

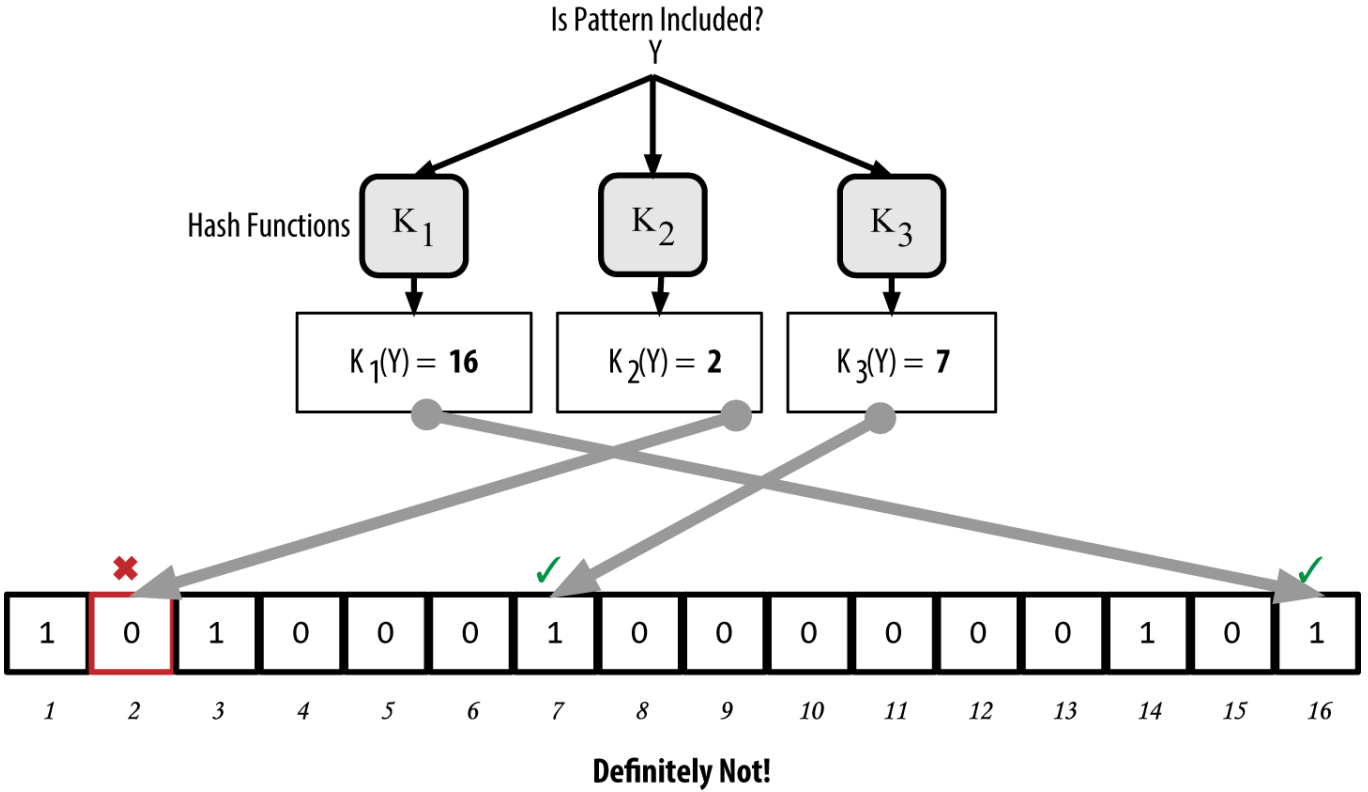
Search



https://github.com/bitcoinbook/bitcoinbook/blob/develop/ch08.asciidoc

Bloom filters

Search



https://github.com/bitcoinbook/bitcoinbook/blob/develop/ch08.asciidoc

SPV nodes and Bloom filters

SPV A and its full node B:

- A inserts the UTXOs and addresses of interst to a Bloom filter (BF)
- A inserts a bit of junk data into the BF
- A sends its BF to B
- B uses the BF in the communication channel with A
- For each *tx* that B receives, if *tx* matches the BF, *tx* is sent to A

- A validates whether *tx* was a false positive and in this case it discards *tx*

SPV nodes and Bloom filters

What B checks in a transaction:

- Hash of the transaction (the ID)
- Inputs
- Outputs
- All the addresses

Read more on this:

- https://github.com/bitcoin/bips/blob/master/bip-0037.mediawiki#filter-matching-algorithm

SPV nodes and Bloom filters

Numbers used in Bitcoin:

- The size of the filter is 36.000 bytes max
- 50 hash functions
- Full node sends the header + the certificate for the transactions matching the filter

Read more on this:

- https://github.com/bitcoin/bips/blob/master/bip-0037.mediawiki#filter-matching-algorithm

Bitcoin network

References:

1. Antonopoulos: https://github.com/bitcoinbook/bitcoinbook/blob/develop/ch08.asciidoc
2. Protocol specification: https://en.bitcoin.it/wiki/Protocol_documentation
3. Network: https://en.bitcoin.it/wiki/Network
4. P2P network: https://bitcoin.org/en/developer-guide#peer-discovery
5. SPV nodes: https://bitcoin.org/en/developer-guide#simplified-payment-verification-spv
6. Bloom filters: https://github.com/bitcoin/bips/blob/master/bip-0037.mediawiki#filter-matching-algorithm

Mining pools

What does a miner do on her own:

- Buys hardware
- Spends electricity to compute the hashes

- Gets Bitcoins (ocassionally)

- A huge variance on when it will discover the blocks
- It can happen it does not discover a block for two years

- Not a sustainable bussiness strategy

Mining pools

Solution:

- Several miners join a *mining pool*
- Similar to a producers collective (i.e. zadruga)
- They all solve the same mining puzzle
- *Pool manager* gives them the puzzle = block

- How does a pool manager know how much to pay each miner?
- Proportional to the work the miner did
- How can we measure this?

Mining pools

Mining shares:

- An incomplete solution to the mining puzzle
- Puzzle to mine the block has the difficulty of 70 zeros
- A share has a difficulty of 50 zeros

- Each time a miner mines a share, it sends it to the pool manager
- Manager pays in proportion with the number of mined shares

- It can happen that the miner that mined the block receives less than others

Mining pools

Mining shares:

- An incomplete solution to the mining puzzle
- Puzzle to mine the block has the difficulty of 70 zeros
- A share has a difficulty of 50 zeros

```
4AA087F0A52ED2093FA816E53B9B6317F9B8C1227A61F9481AFED67301F2E3FB
D3E51477DCAB108750A5BC9093F6510759CC880BB171A5B77FB4A34ACA27DEDD
00000000008534FF68B98935D090DF5669E3403BD16F1CDFD41CF17D6B474255
BB34ECA3DBB52EFF4B104EBBC0974841EF2F3A59EBBC4474A12F9F595EB81F4B
00000000002F891C1E232F687E41515637F7699EA0F462C2564233FE082BB0AF
0090488133779E7E98177AF1C765CF02D01AB4848DF555533B6C4CFCA201CBA1
460BEFA43B7083E502D36D9D08D64AFB99A100B3B80D4EA4F7B38E18174A0BFB
000000000000000078FB7E1F7E2E4854B8BC71412197EB1448911FA77BAE808A
652F374601D149AC47E01E777613845618 1FA4F9D0EEDD8C4FDE3BEF6B1B7ECE
785526402143A291CFD60DA09CC80DD066BC723FD5FD20F9B50D614313529AF3
000000000041EE593434686000AF77F54CDE839A6CE30957B14EDEC10B15C9E5
9C20B06B01A0136F192BD48E0F372A4B9E6BA6ABC36F02FCED22FD9780026A8F
```

Mining pools

Mining shares:

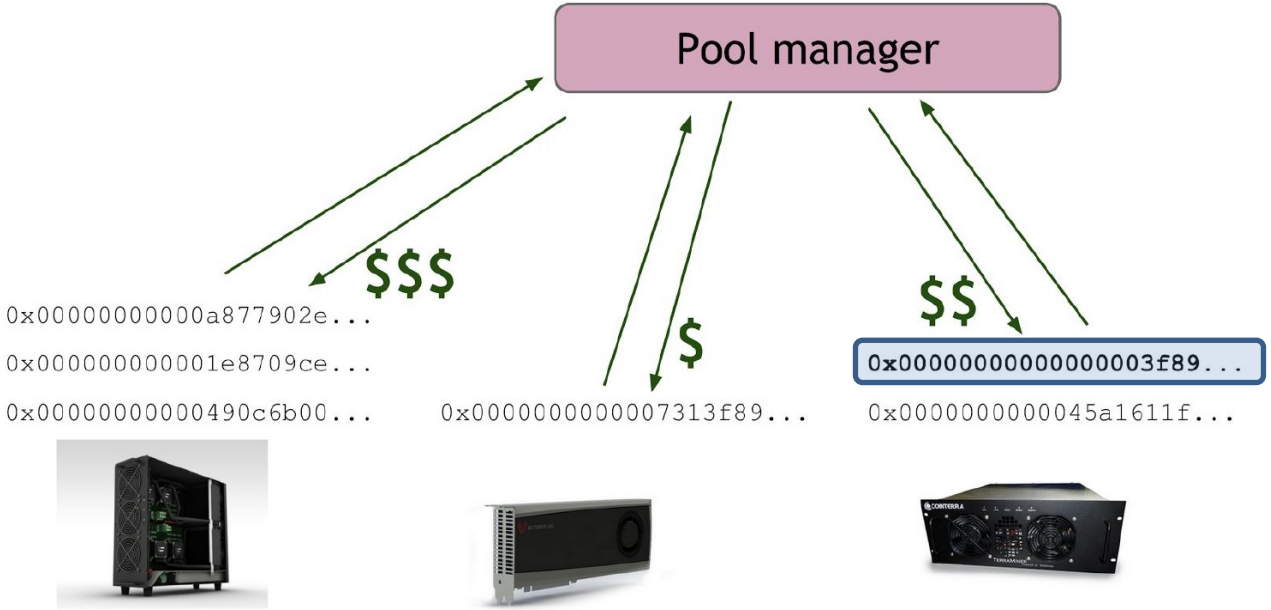
- An incomplete solution to the mining puzzle
- Puzzle to mine the block has the difficulty of 70 zeros
- A share has a difficulty of 50 zeros

- Each time a miner mines a share, it sends it to the pool manager
- Manager pays in proportion with the number of mined shares

- It can happen that the miner that mined the block receives less than others

Mining pools

It can happen that the miner that mined the block receives less than others



Mining pools

Operating a pool:

- How to pay the miners?
- Op1: flat fee – manager assumes the risk (pays even no block is mined)
- Op2: proportional – manager pays when a block is mined

- Miners can play with all these options

Mining pools

Mining pools and the 51% attack

- Ghash.io – in 2014 had 50% of the hash power

- The market autoregulates (at least that's what capitalism is based on)

Electricity

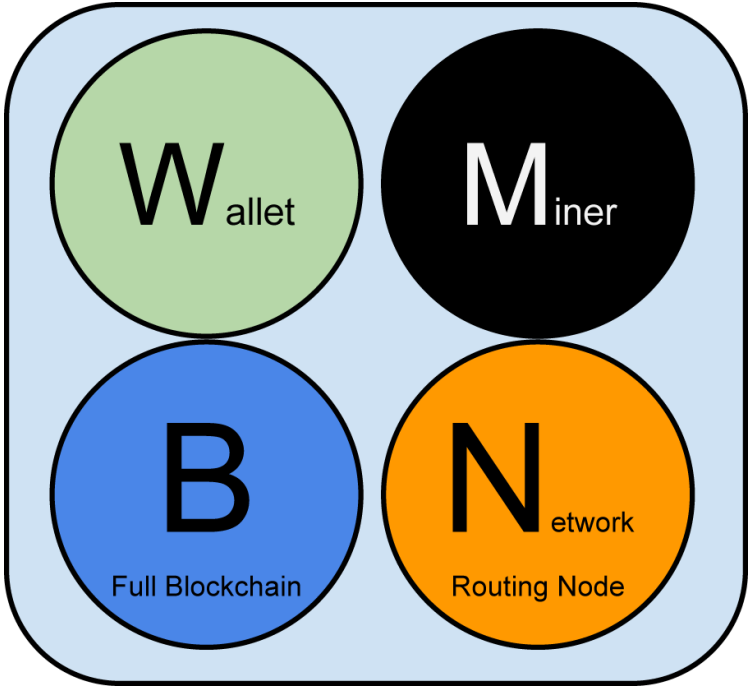
Two schools of thought:

1. Bitcoin mining is wasteful

2. Bitcoin mining is environmentaly friendly in the long term

Network recap

Node functionality



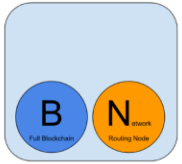
https://github.com/bitcoinbook/bitcoinbook/blob/develop/ch08.asciidoc

Network recap

Node type

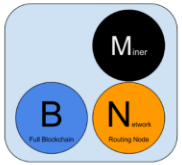
Reference Client (Bitcoin Core)

Contains a Wallet, Miner, full Blockchain database, and Network routing node on the bitcoin P2P network.



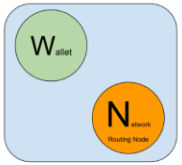
Full Block Chain Node

Contains a full Blockchain database, and Network routing node on the bitcoin P2P network.



Solo Miner

Contains a mining function with a full copy of the blockchain and a bitcoin P2P network routing node.



Lightweight (SPV) wallet

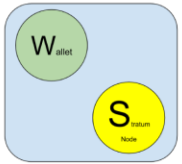
Contains a Wallet and a Network node on the bitcoin P2P protocol, without a blockchain.

Pool Protocol Servers

Gateway routers connecting the bitcoin P2P network to nodes running other protocols such as pool mining nodes or Stratum nodes.

Mining Nodes

Contain a mining function, without a blockchain, with the Stratum protocol node (S) or other pool (P) mining protocol node.

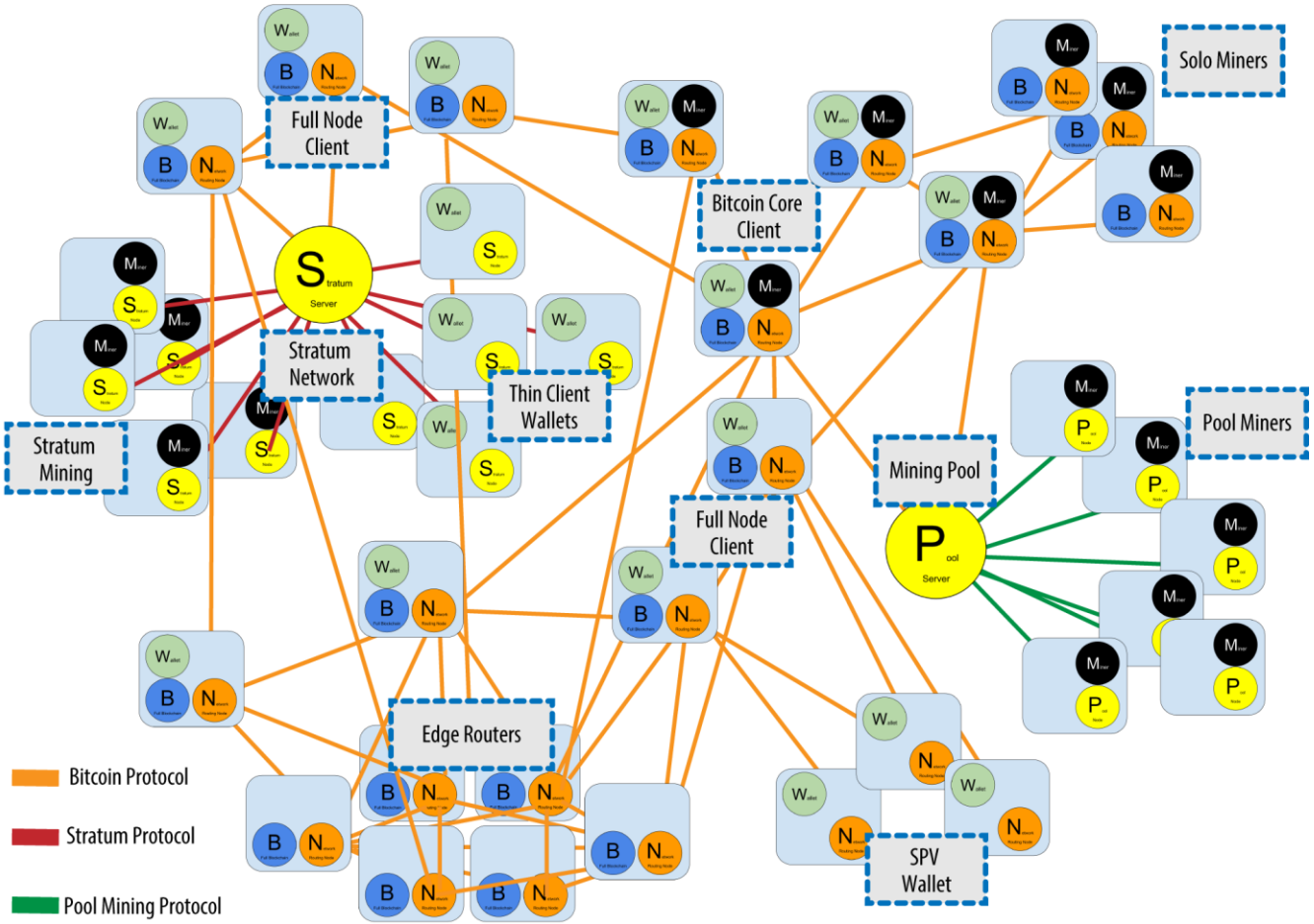


Lightweight (SPV) Stratum wallet

Contains a Wallet and a Network node on the Stratum protocol, without a blockchain.

https://github.com/bitcoinbook/bitcoinbook/blob/develop/ch08.asciidoc

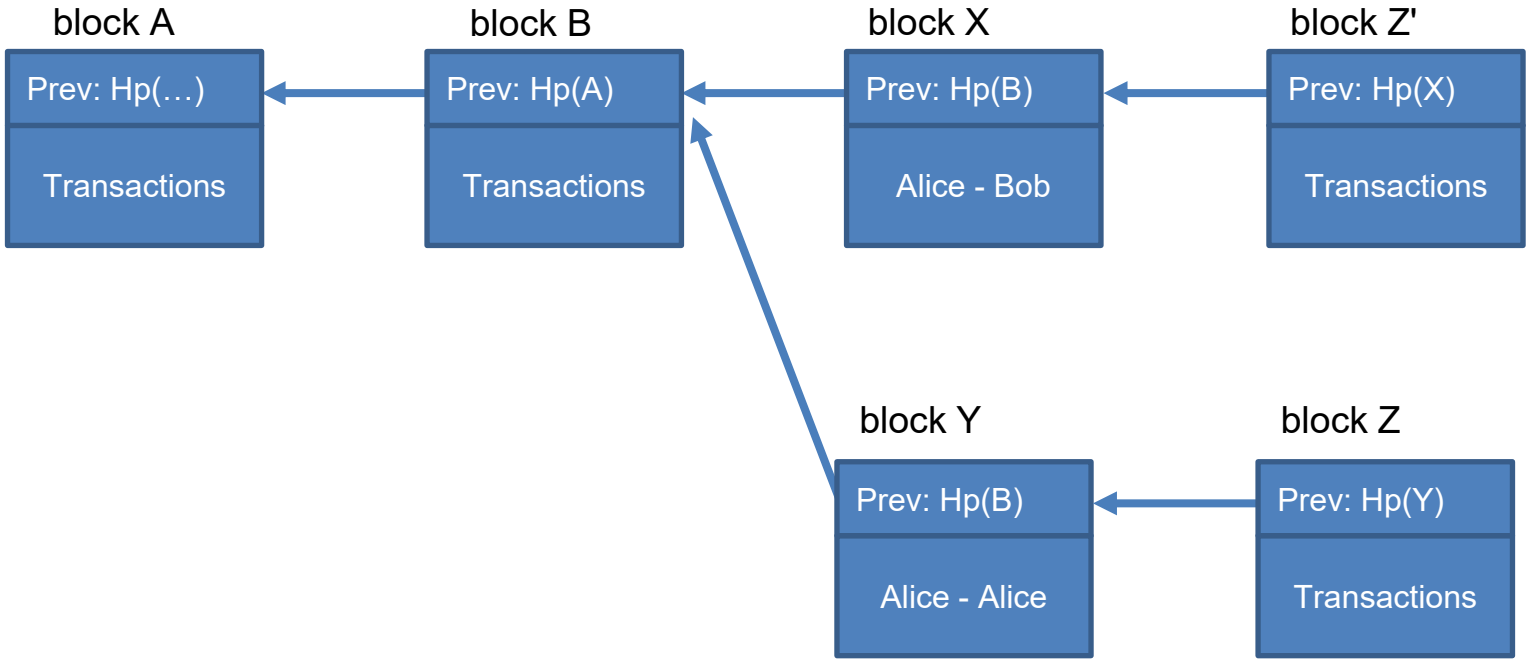
Network recap



https://github.com/bitcoinbook/bitcoinbook/blob/develop/ch08.asciidoc

Forks

Two (or more) branches in the blockchain:



Forks

A rule change

- All the nodes need to upgrade their software
- Does not happen naturally

- **Hard forks vs. Soft forks**

Forks

Hard fork:

- A change that is not forward compatible
- Introduces rules that were not valid previously

- E.g. you should sign double hash of a transaction H(H(tx)), and not only H(tx)

- Nodes that did the upgrade are OK, but the rest are not
- Blockchain divides

- Example: Bitcoin Cash

Forks

Soft fork:

- A change that is forward compatible
- Introduces stricter rules

- Nodes with the old software will accept all new blocks
- Nodes with the new software will reject some blocks that were previously valid
- Miners with the old software can realize that they need to upgrade

- Example: (almost) any BIP; e.g. P2SH