Red de Bitcoin

¿Cómo funciona Bitcoin?

Bitcoin network

Una red p2p estándar:

- Nodos se conectan y desconectan
- Nuevos nodos entran a la red
- No hay conectividad perfecta
- Nodos escuchan mensajes de sus vecinos
- Nodos transmiten mensajes a sus vecinos

Bitcoin network

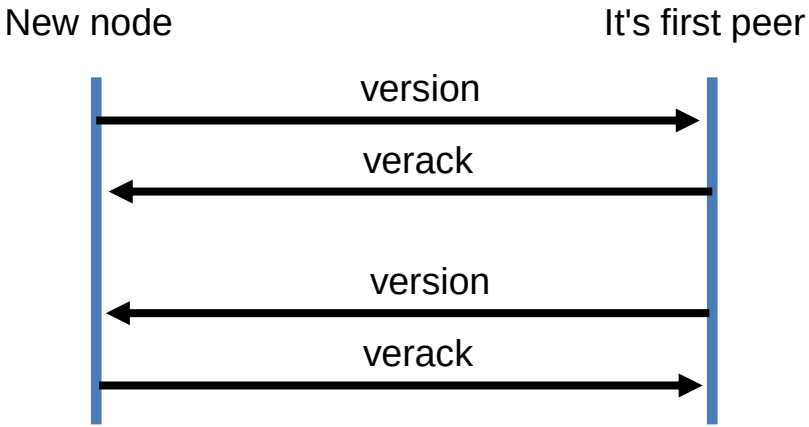
Como un nodo nuevo logra tener vecinos?

- DNS Seed estatico
- DNS Seed BIND (Berkeley Internet Name Daemon)
- Una IP especifica de un nodo que el nuevo nodo ya conoce

Bitcoin network

Mensajes básicos para establecer una conexión:

- *version*
- *verack*



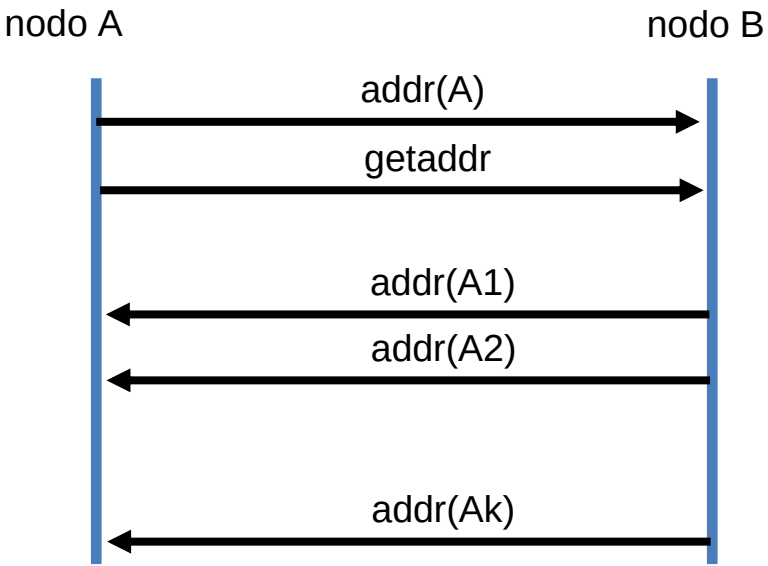
Bitcoin network

Descubriendo nuevos pares:

- A se conecta a B
- B va a pasar la dirección de A a sus papes
- A igualmente puede preguntar a B direcciones de sus pares
- B responde con estas direcciones

Bitcoin network

Descubriendo nuevos pares:



Bitcoin network

Nodos entran y salen de la red:

- Un proceso dinámico
- Nodo guarda sus pares e intenta conectarse a ellos al entrar de nuevo
- Si no hay tráfico con un nodo por 30 minutos, se manda un mensaje
- Si no hay tráfico en 90 minutos, el nodo se desconecta (una arista menos)

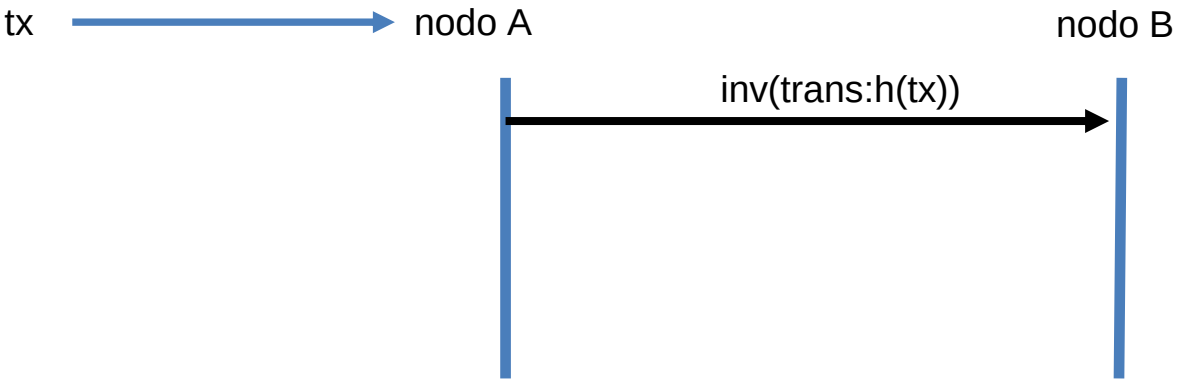
- Se recomienda tener no más de 100 pares
- Demasiada conexiones ponen presión a la red

Bitcoin network

Comunicación básica:

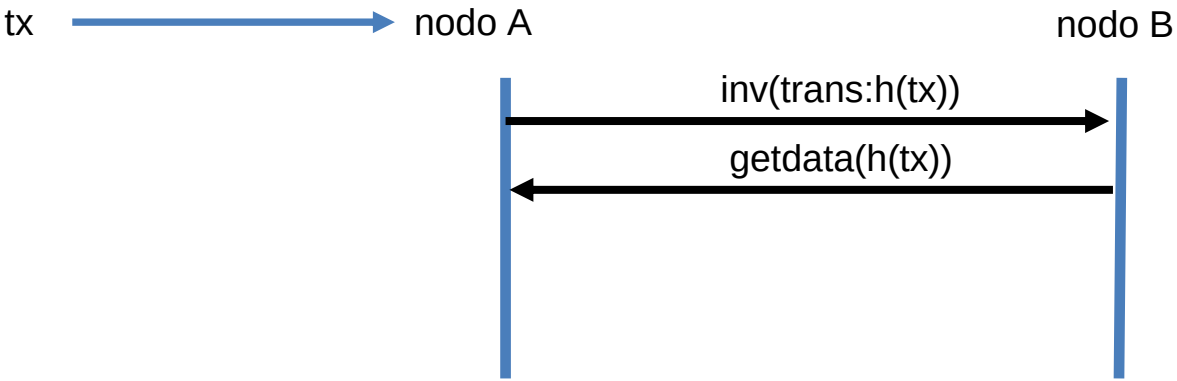
Bitcoin network

Comunicación básica:



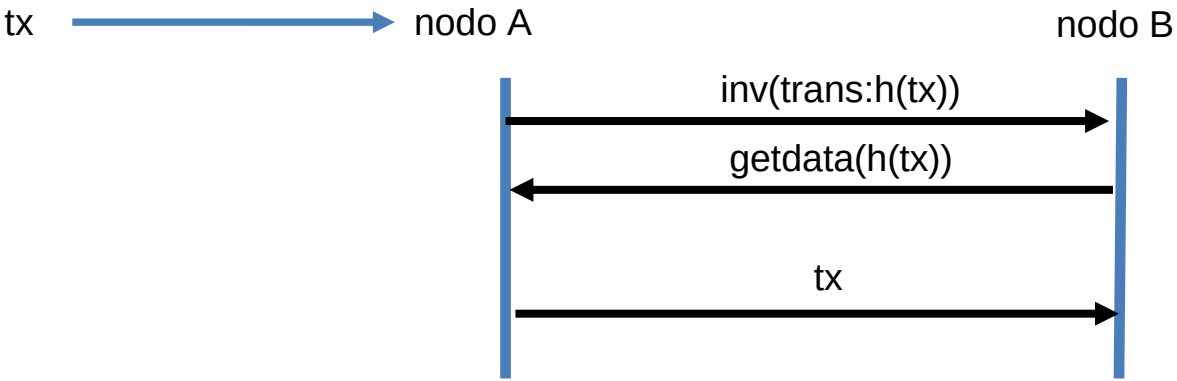
Bitcoin network

Comunicación básica:



Bitcoin network

Comunicación básica:

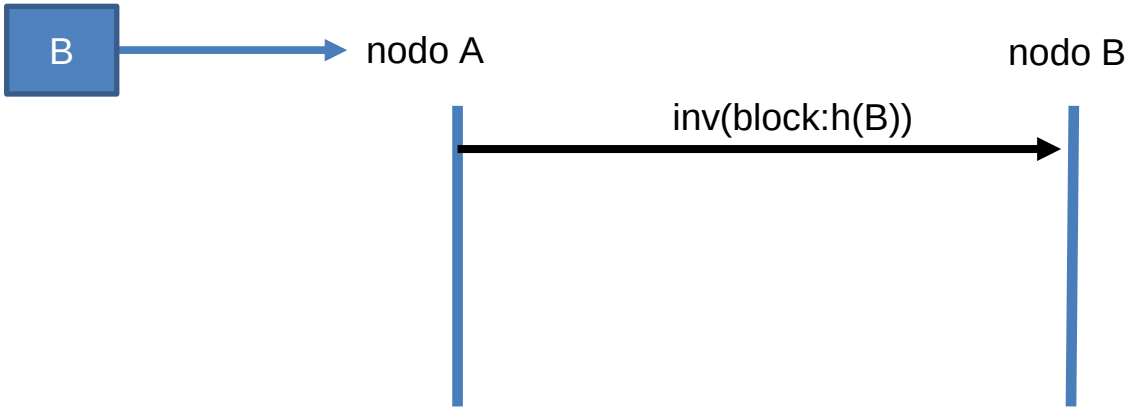


Bitcoin network

Comunicación básica (bloques):

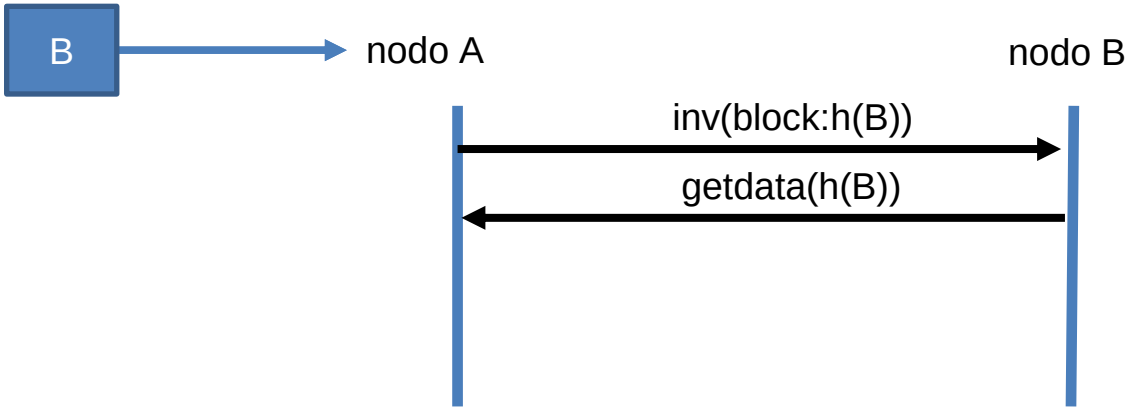
Bitcoin network

Comunicación básica (bloques):



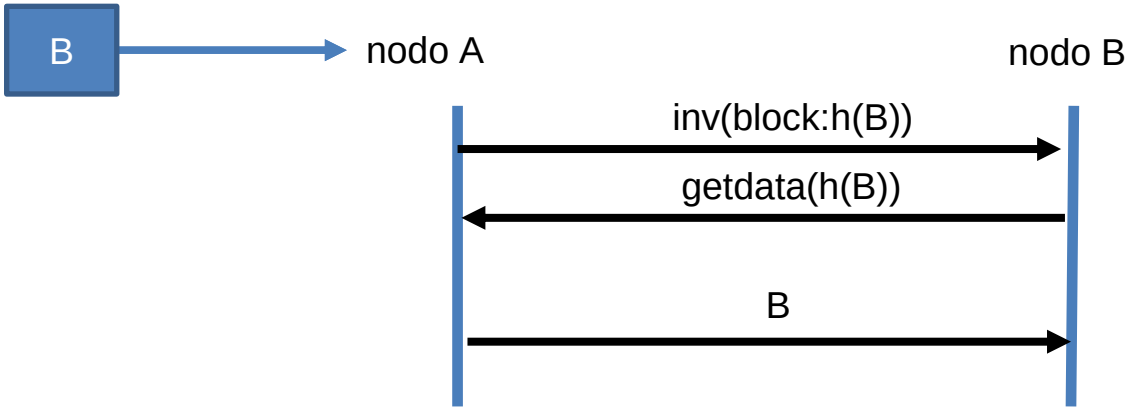
Bitcoin network

Comunicación básica (bloques):



Bitcoin network

Comunicación básica (bloques):



Bitcoin network

Full nodes (nodos completos):

- Guardan todo el blockchain
- Verifican todas las transacciones y bloques

- Como un nodo entero obtiene la copia del blockchain cuando entra a la red?

Bitcoin network

Up to version 0.9.3 of BitcoinCore

Como el nodo X obtiene el blockchain?

- Siempre parte con el genesis block (hardcodeado en su software)
- Otros bloques le pasan sus pares en la red

- Primer mensaje que manda/recibe X: *version (contiene el campo BestHeight)*
- X manda *getblocks* a su par con el BestHeight más grande (llamémoslo Y)
- El mensaje *getblocks* contiene el hash del último bloque en el blockchain de X
- Y identifica los primeros 500 bloques que faltan a X
- Y manda el mensaje *inv* con los hashes de los primeros 500 bloques
- Para conseguirse un bloque, X manda a Y *getdata(hash)*
- Máximo de 500 *getdata* activo por conexión

Initial block download

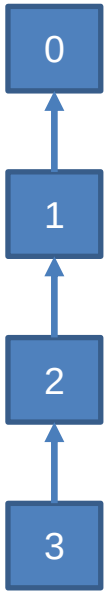
nodo A

nodo B

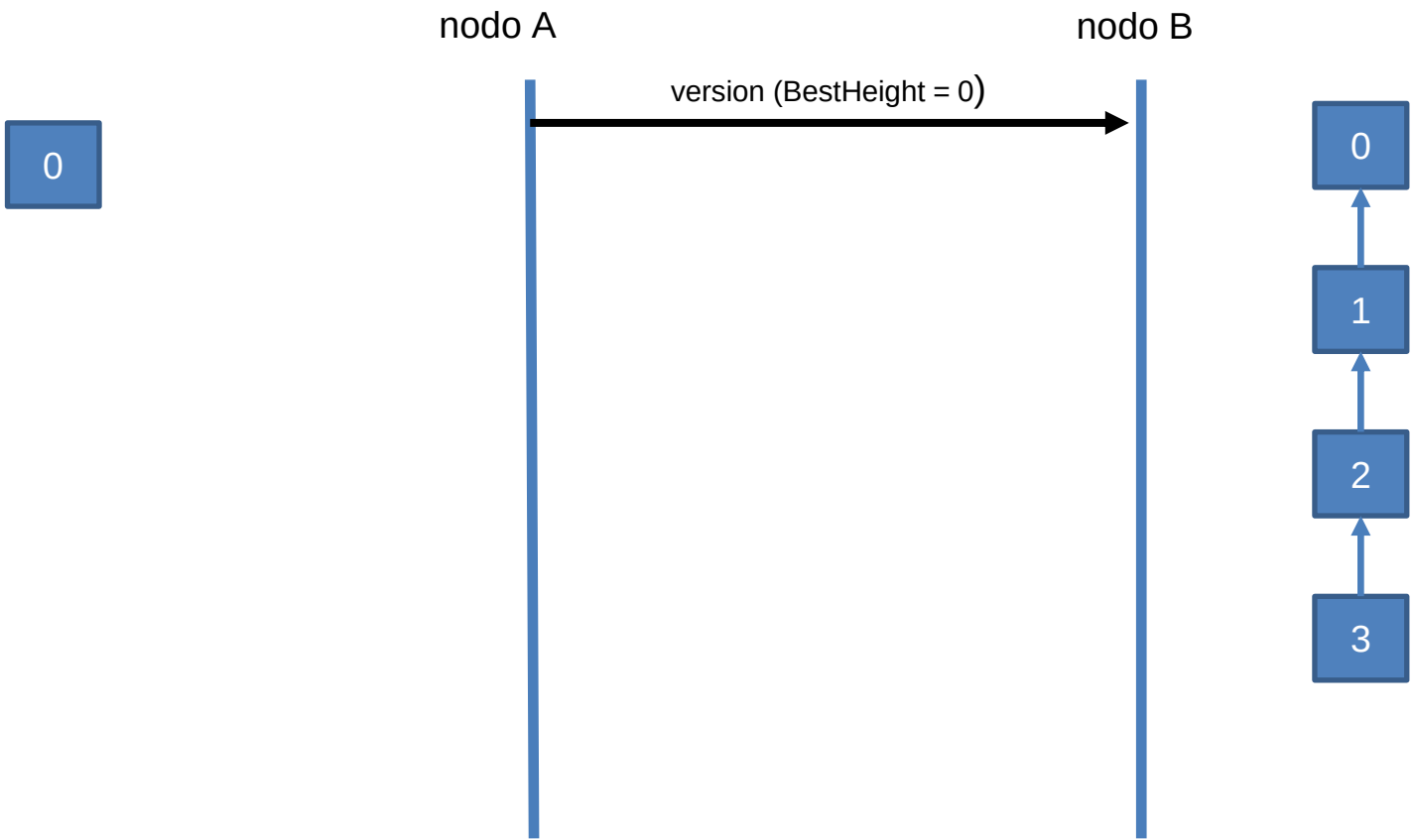
Initial block download

nodo A

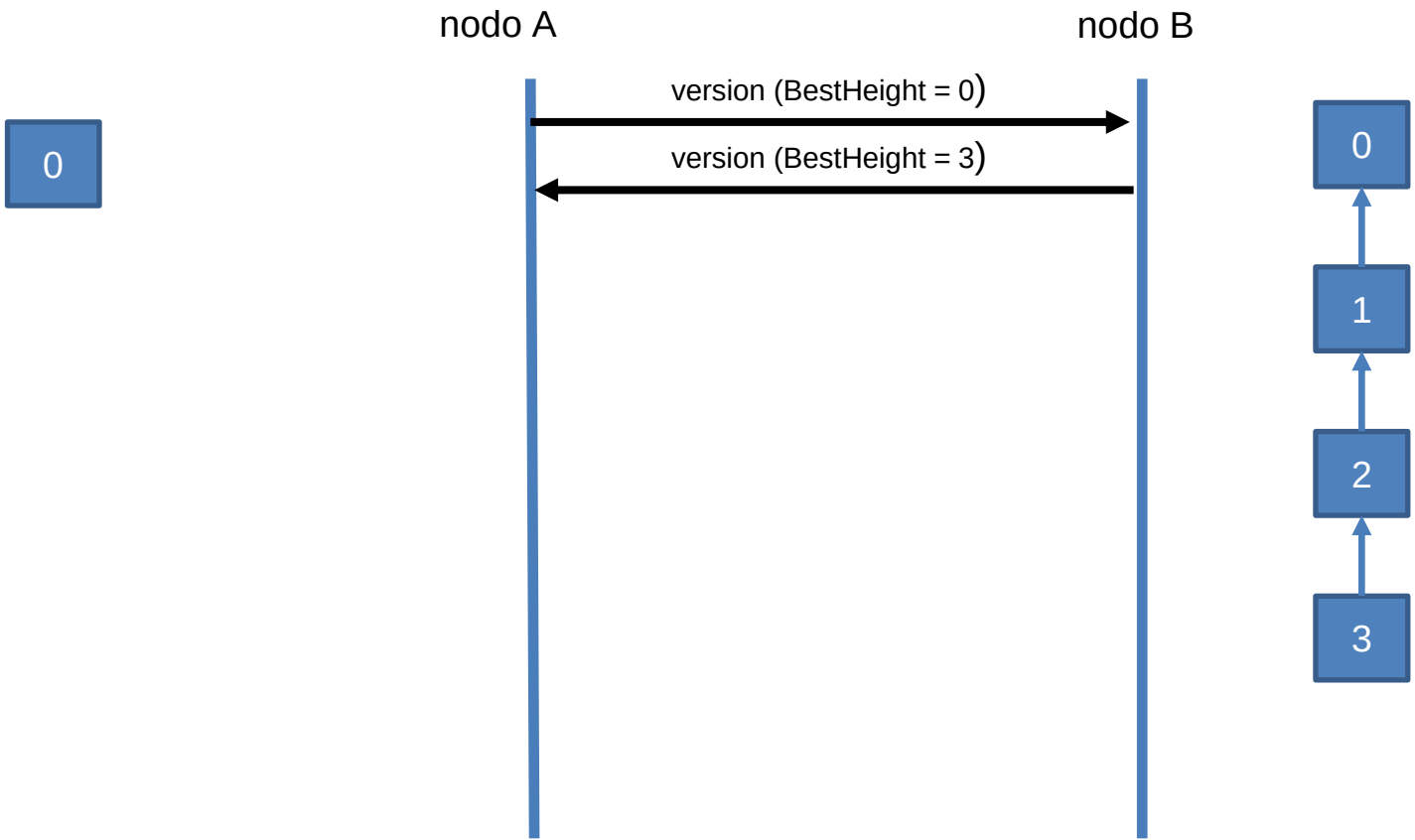
nodo B



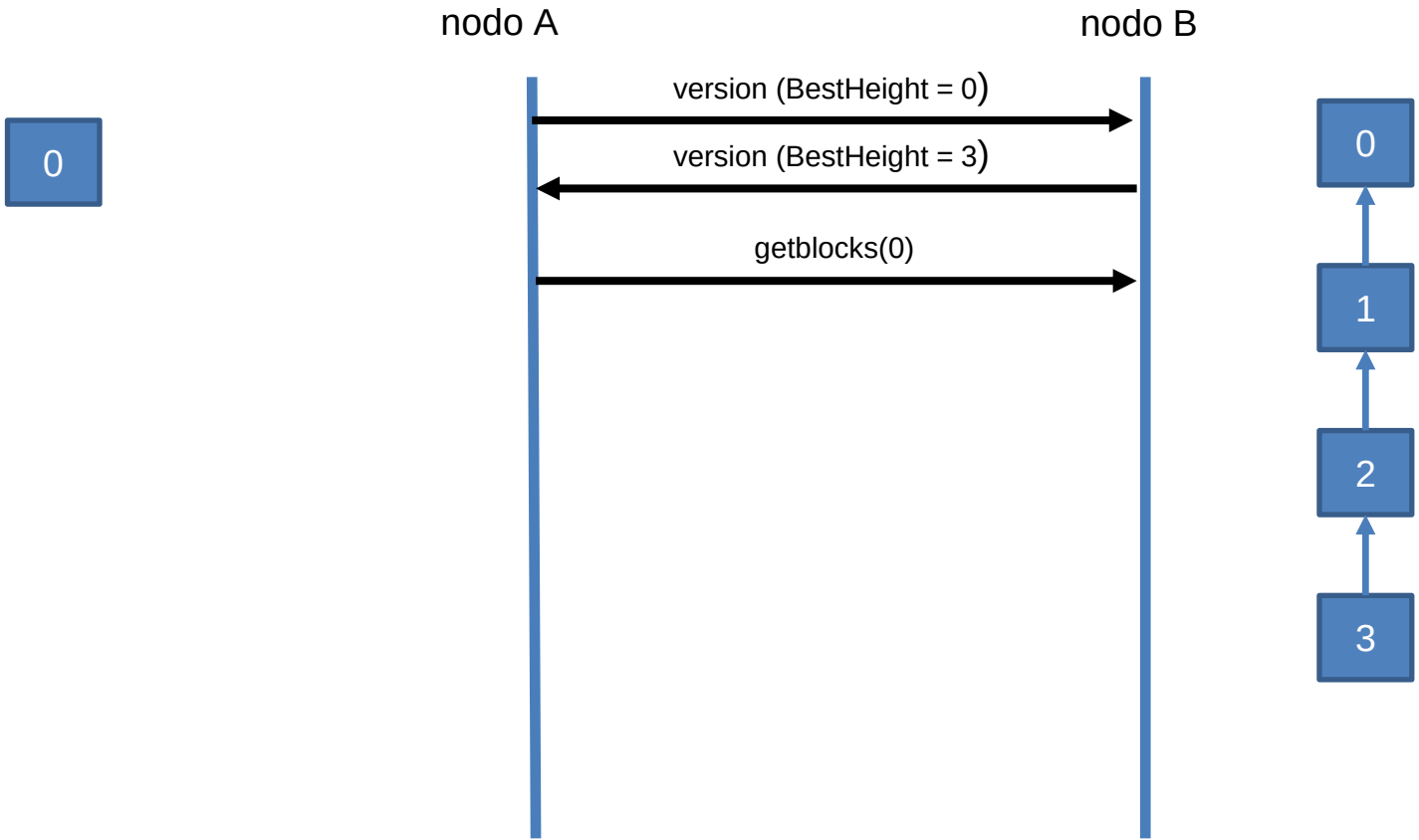
Initial block download



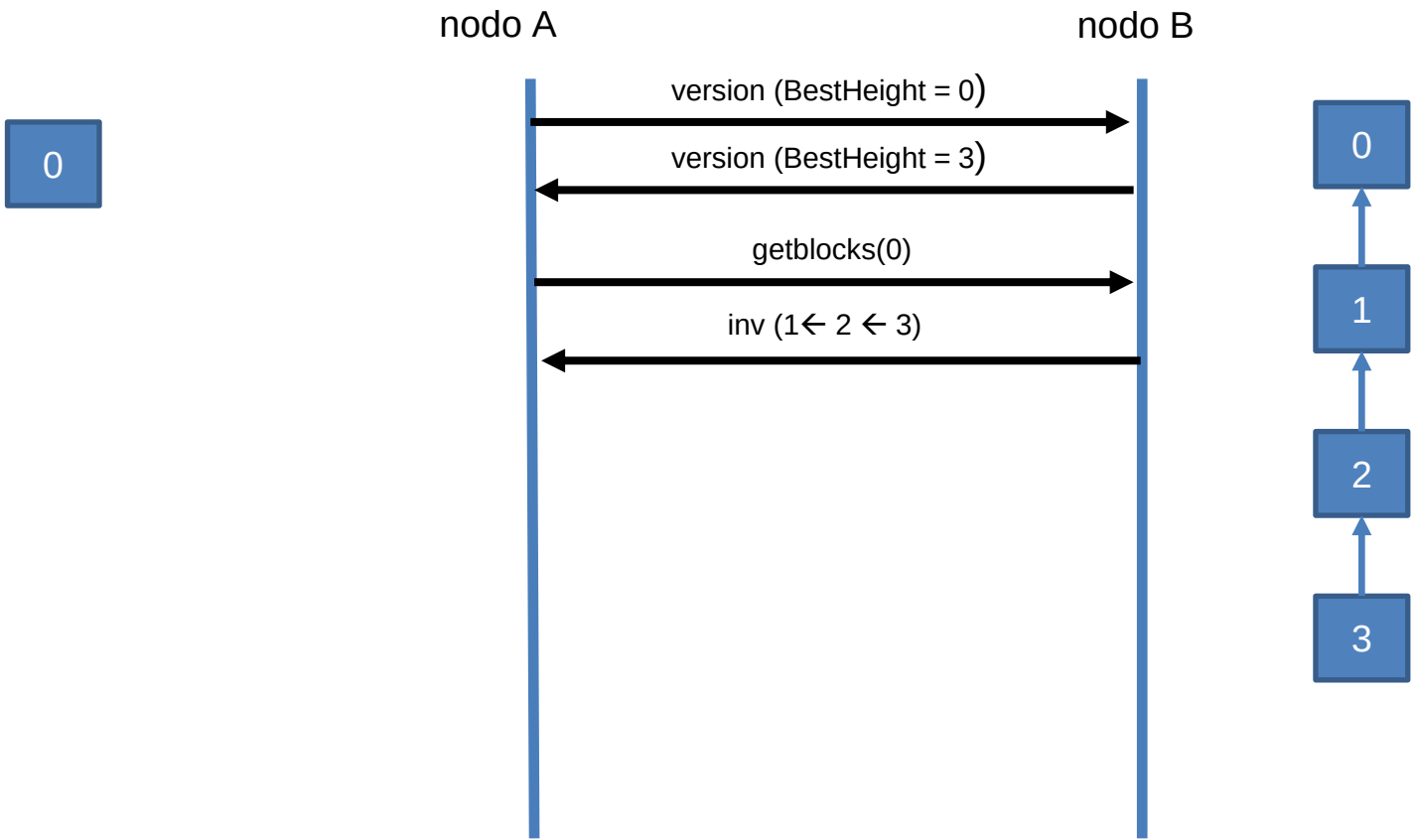
Initial block download



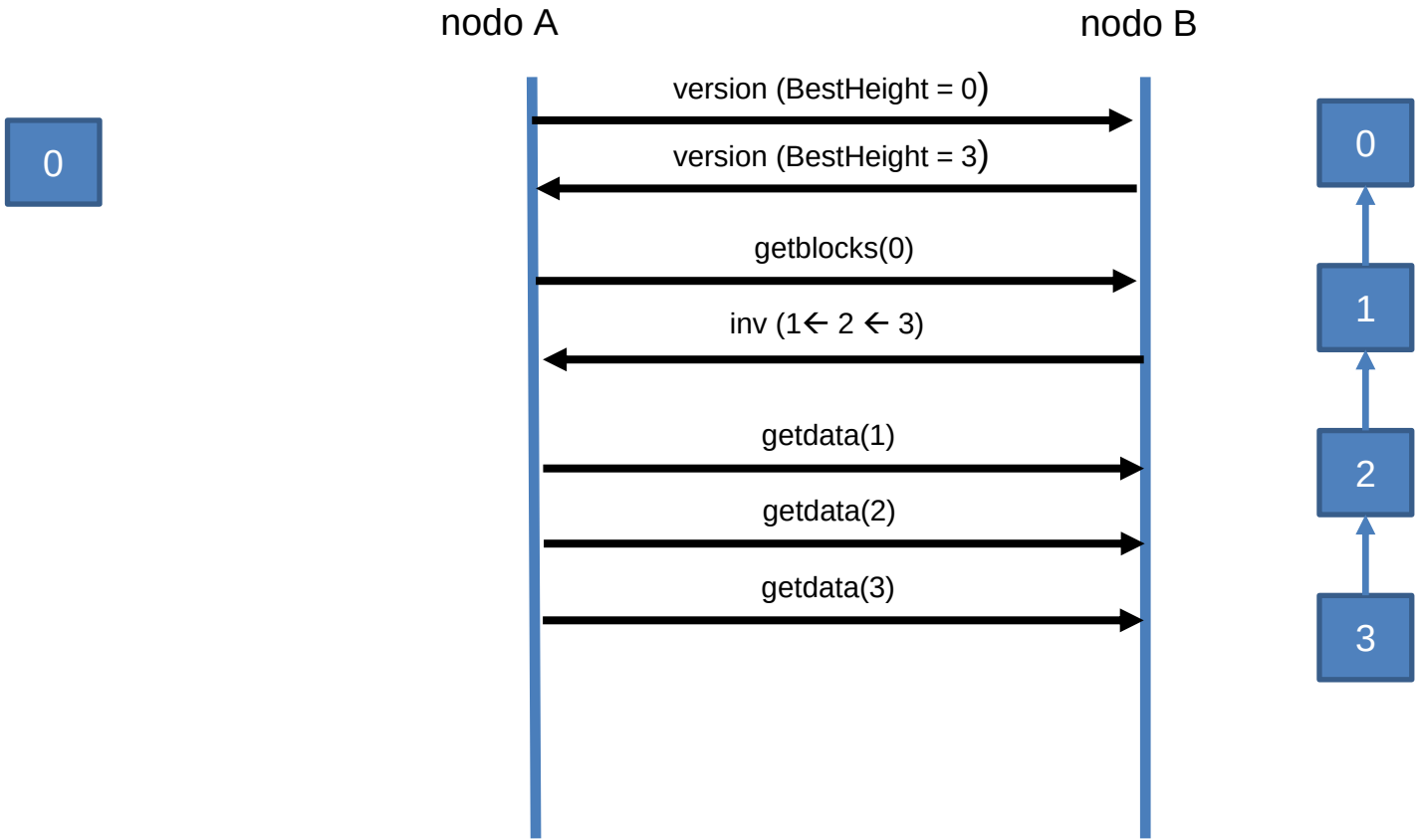
Initial block download



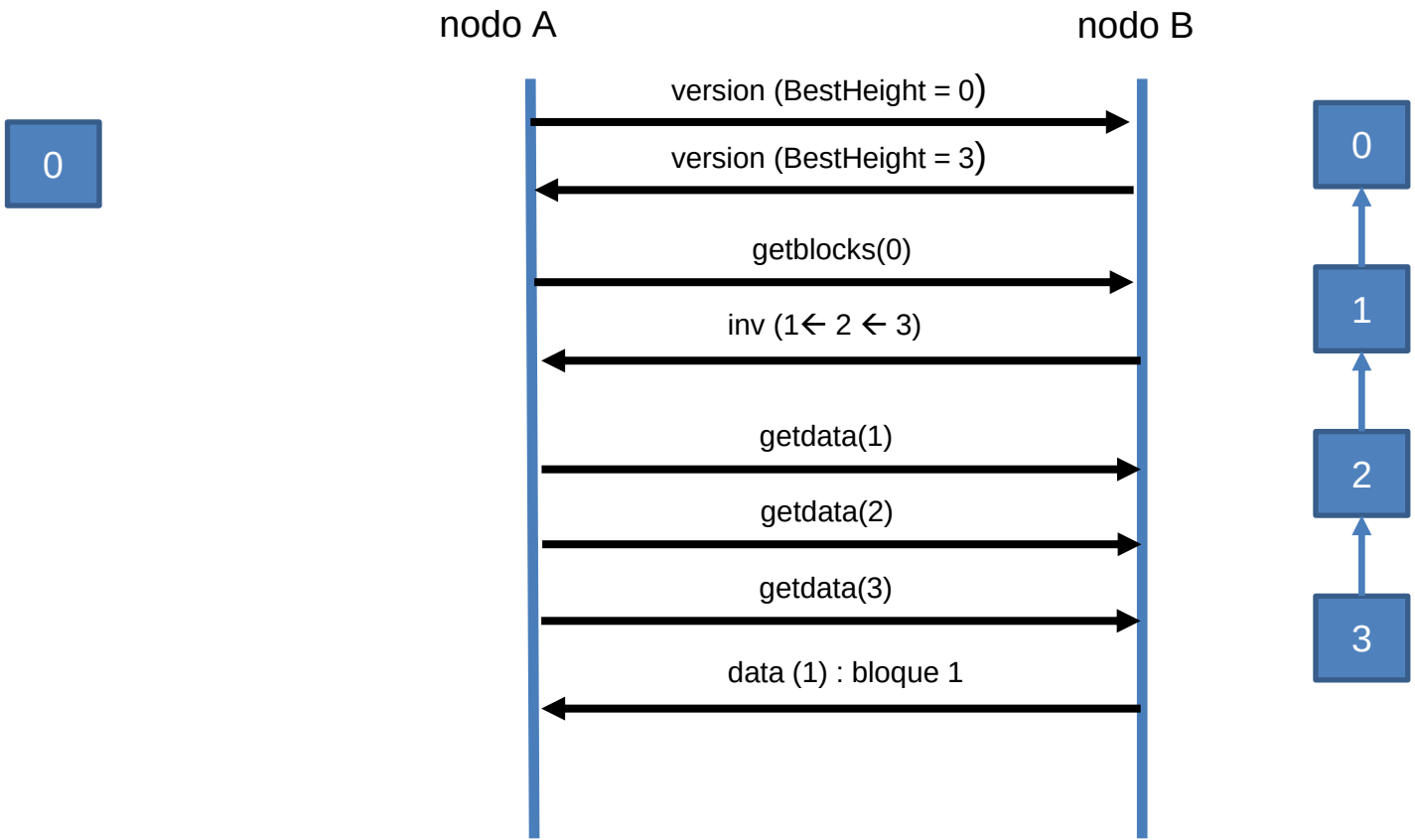
Initial block download



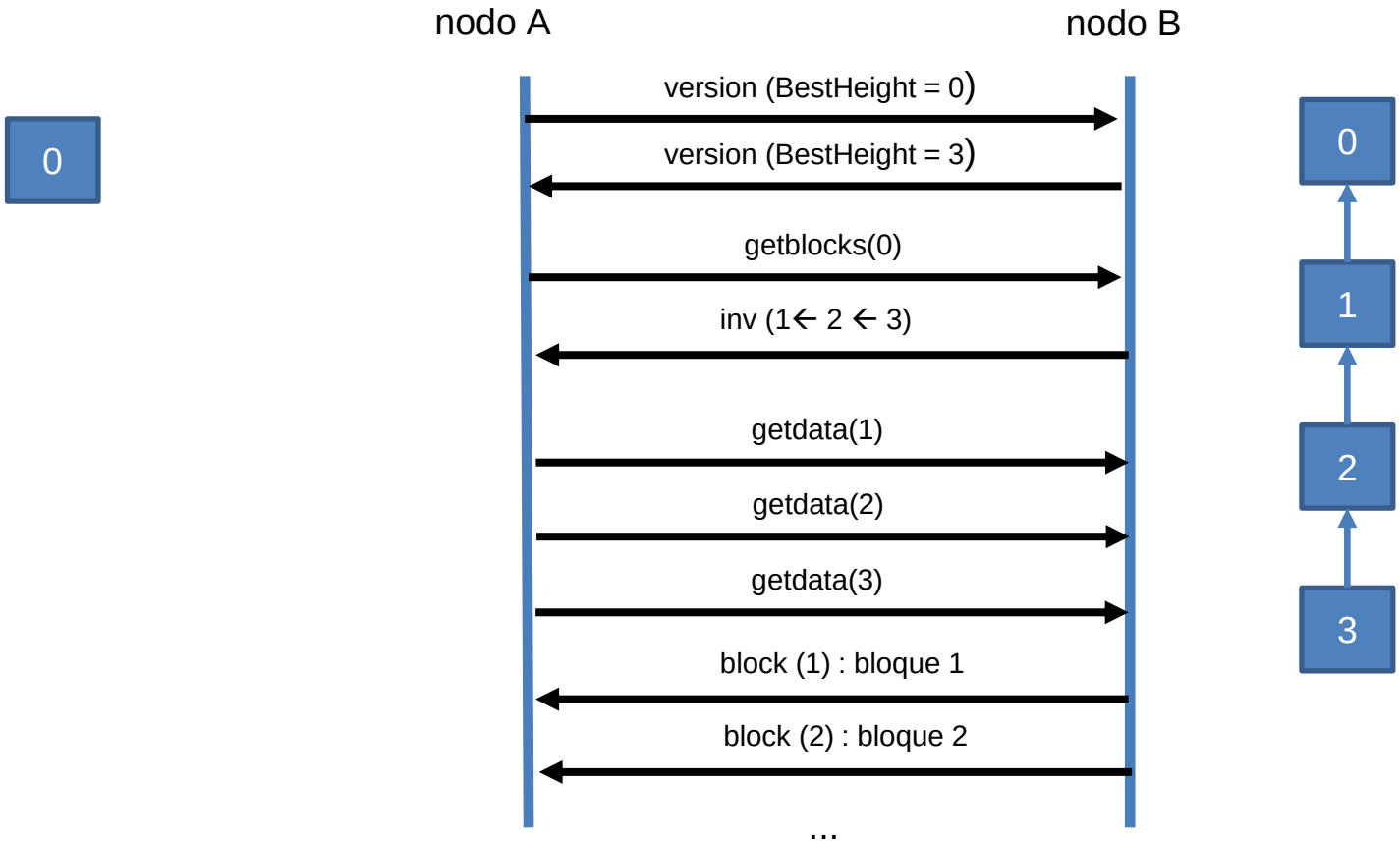
Initial block download



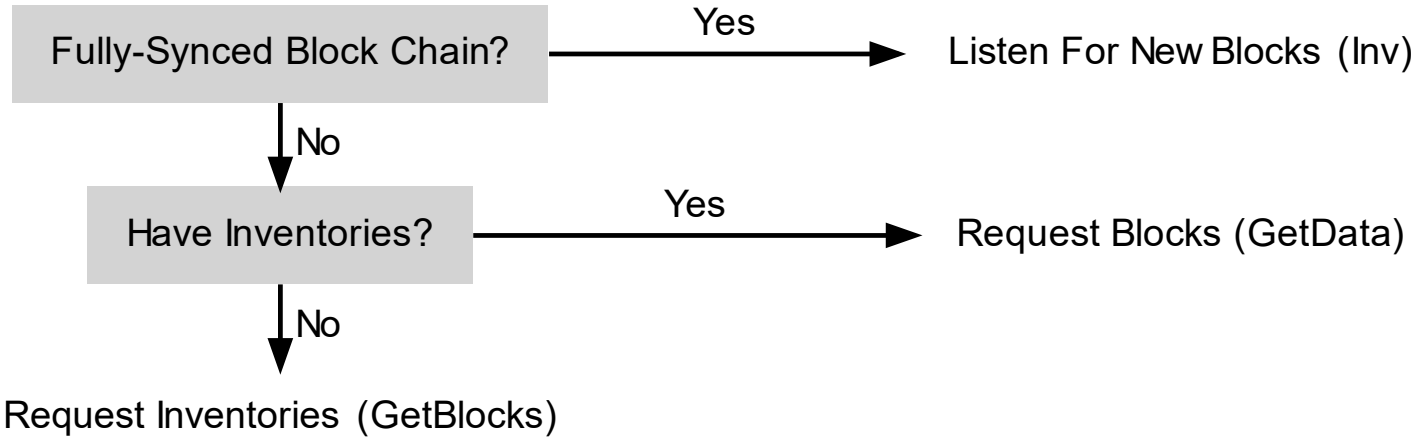
Initial block download



Initial block download



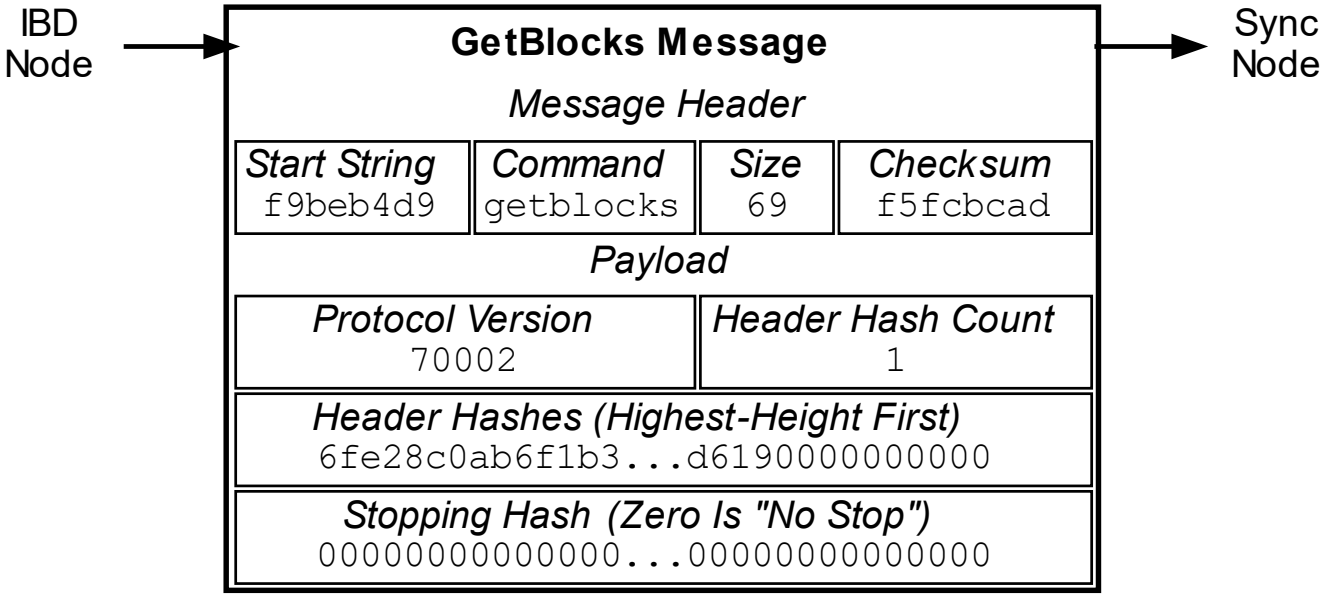
En realidad (IBD)



Overview Of Blocks-First Initial Blocks Download (IBD)

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

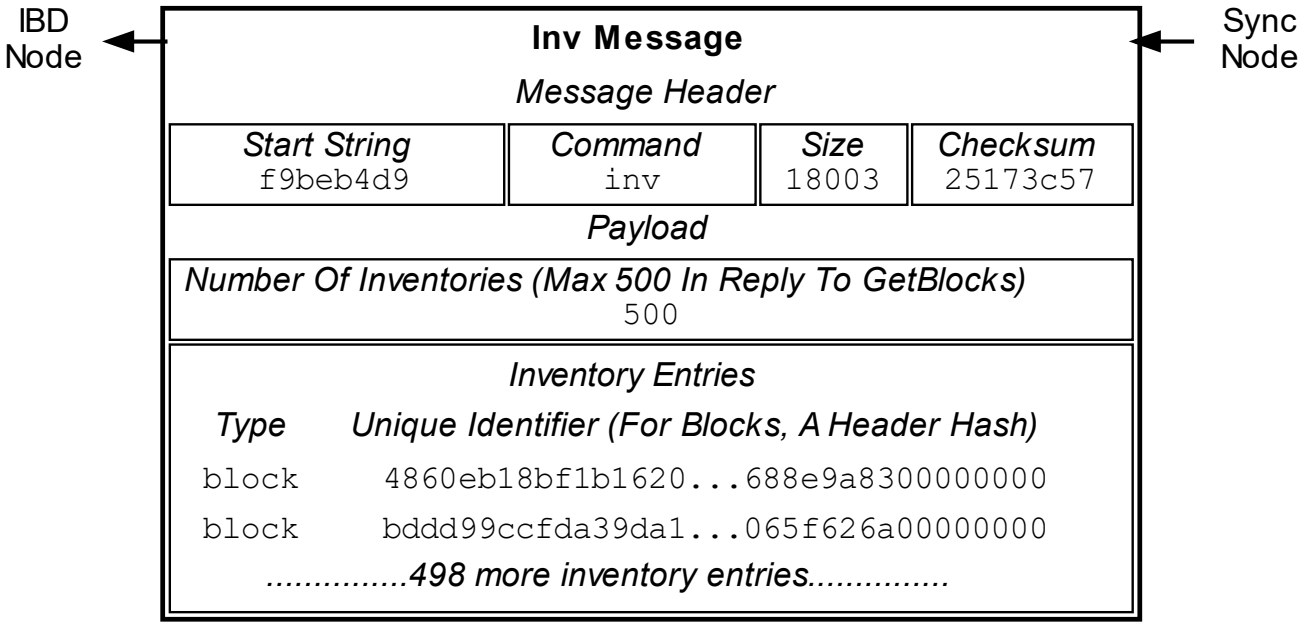
En realidad (IBD)



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

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

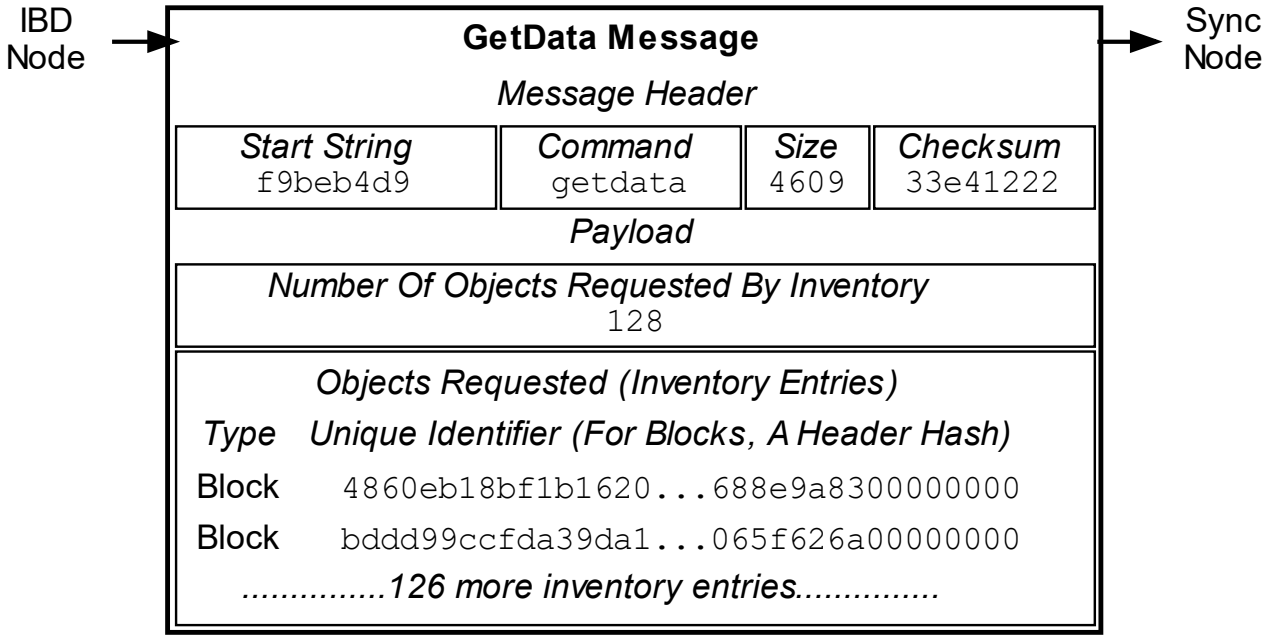
En realidad (IBD)



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

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

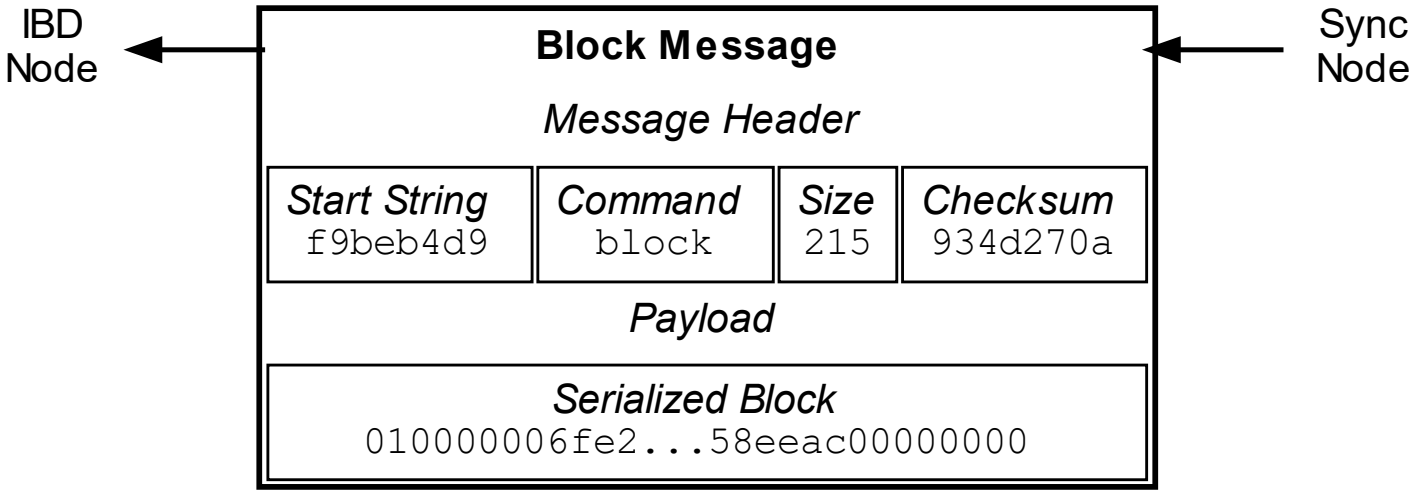
En realidad (IBD)



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

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

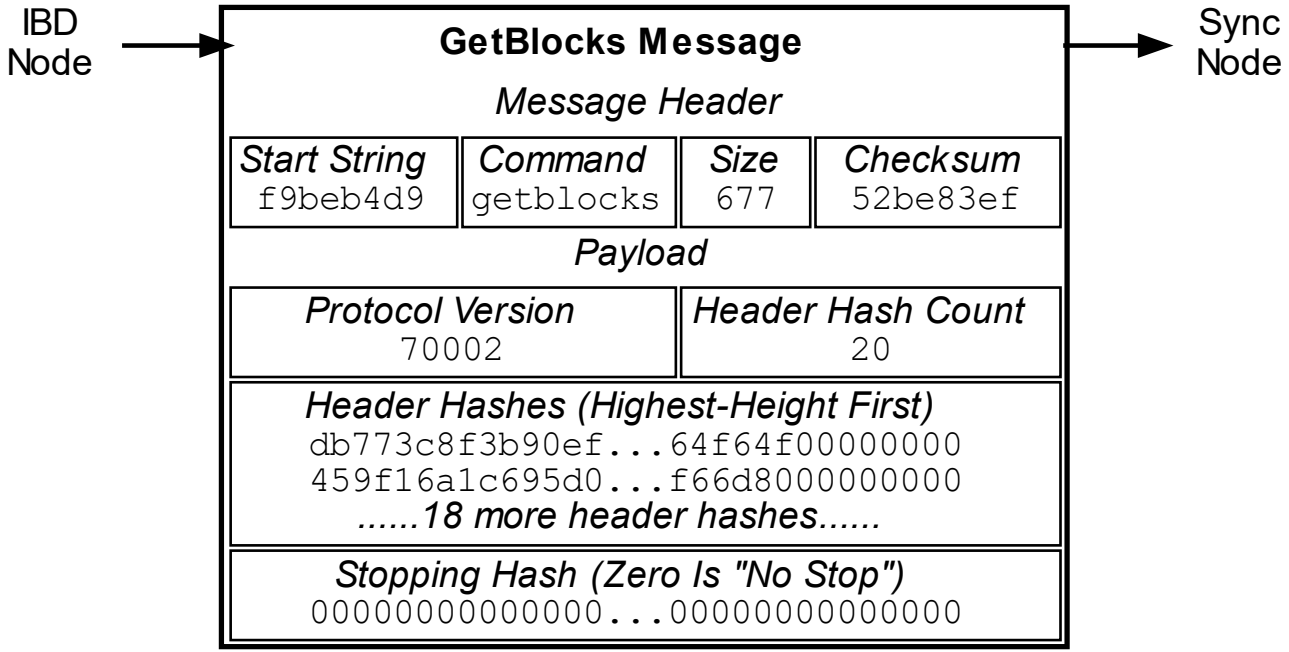
En realidad (IBD)



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

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

En realidad (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 un usuario le interesan solo las transacciones que involucrans a su dirección
- Uso típico de una billetera (wallet)

- Nodos SPV no guardan el blockchain entero
- Guardan solo block headers (80 bytes por bloque = 4.2MB por año)
- Mucho más liviano que un nodo entero
- Pero no pueden validar todos los UTXO válidos

SPV nodes

Nodo SPV puede verificar solo dos cosas:

1. Que una transacción tx pertenece a un bloque blockX
2. Que un bloque blockX tiene k-confirmaciones

SPV nodes

Como un nodo SPV verifica que tx pertenece a blockX?

- Un nodo SPV guarda a 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

Como un nodo SPV A verifica que tx pertenece a blockX?

- Un nodo SPV guarda a block headers
- El nodo SPV A pregunta a un nodo completo B al prueba de tx

- A valida la prueba contra el hashMerkleRoot

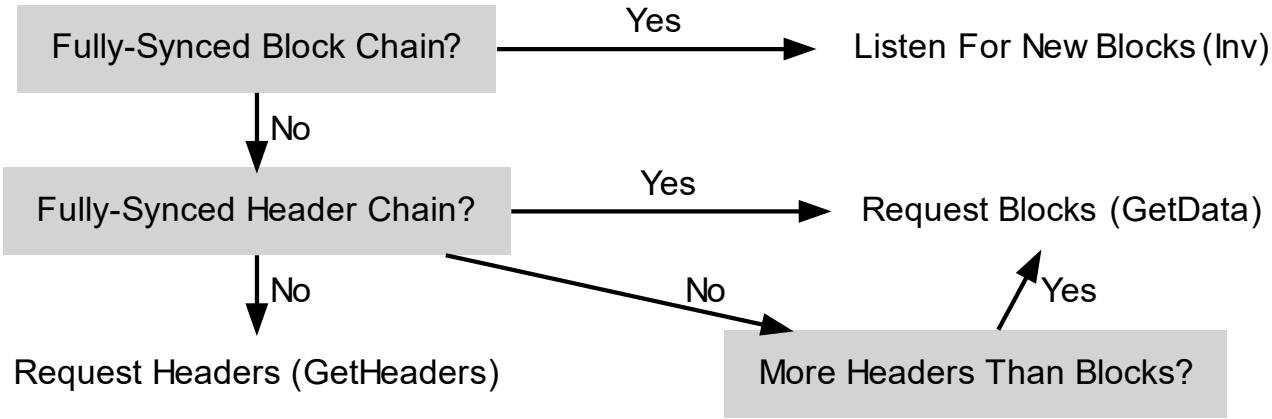
SPV nodes

Como un nodo SPV A verifica que tx tiene k confirmaciones?

- A tiene los headers hasta el final del blockchain actual

- Puede A verificar qué los headers son válidos?

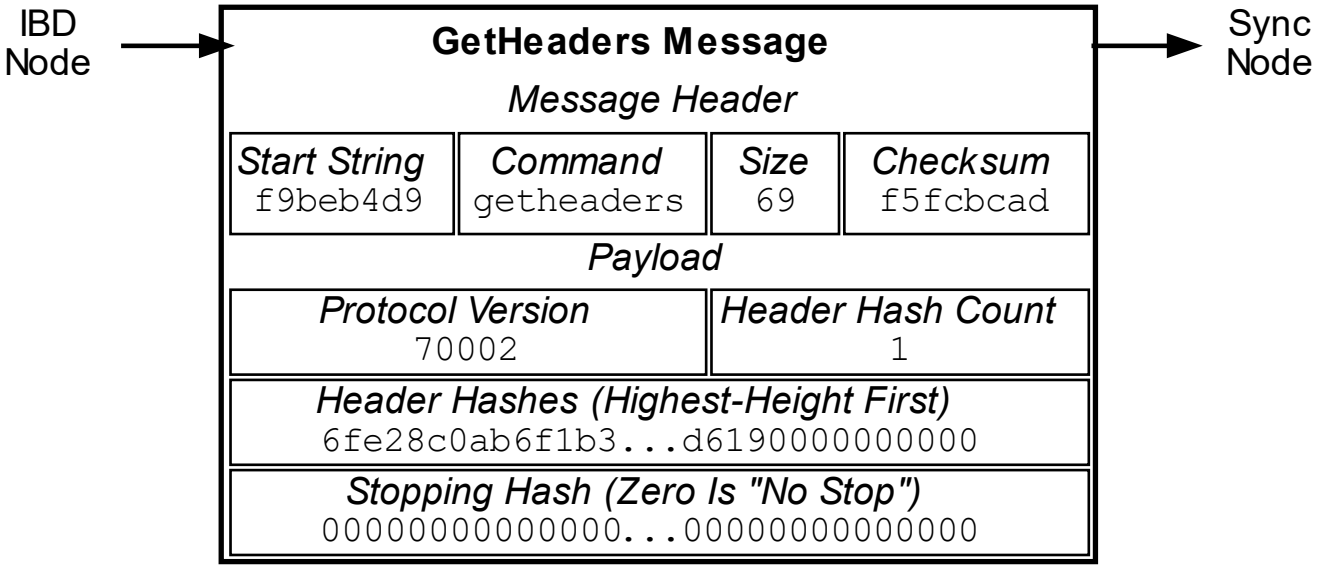
IBD para un nodo SPV



Overview Of Headers-First Initial Blocks Download (IBD)

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

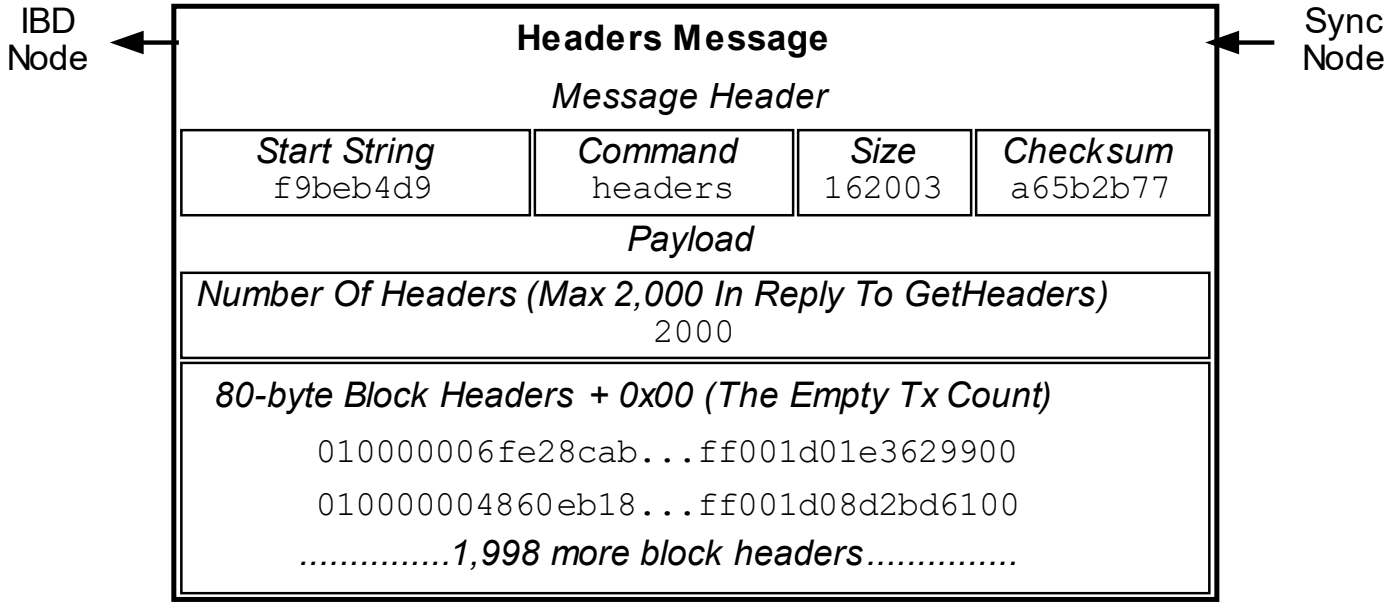
IBD para un nodo SPV



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

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

IBD para un nodo SPV



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

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

Vulnerabilidades de nodos SPV

Primera vulnerabilidad: DOS parcial

- Si el full nodo está malicioso puede pretender qué tx pertenece a blockX?
- Puede pretender qué tx no pertence a blockX?

Solución:

- El nodo SPV debería usar más de un nodo full
- Los arboles de Merkle deberían ser ordenados

Vulnerabilidades de nodos SPV

Segunda vulnerabilidad: perdida de privacidad

- Si el nodo SPV pregunta mucho al nodo B
- B va a conocer todas las direcciones de A
- Puede trackear a A
- Puede prohibir a A poner transacciones

- Como resolver?

Filtros de Bloom

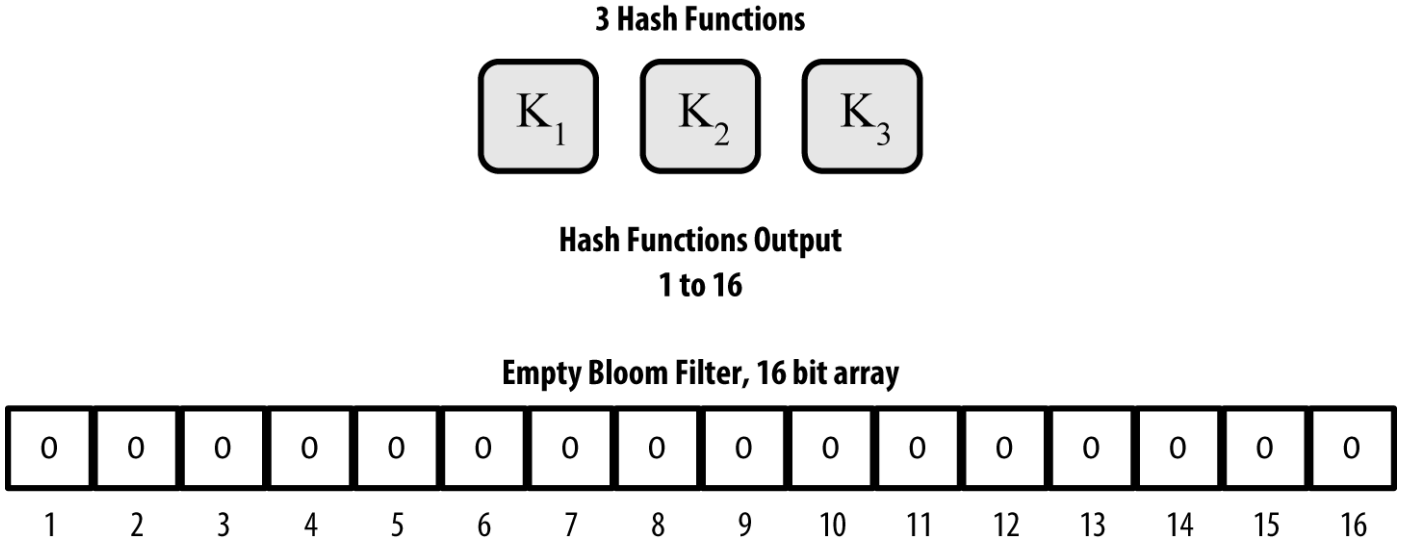
Un filtro de búsqueda probabilista:

- El filtro parte vacío
- Se agregan elementos al filtro

- Uno quiere verificar si el elemento fue agregado al filtro

- Puede dar falsos positivos
- No puede dar falsos negativos

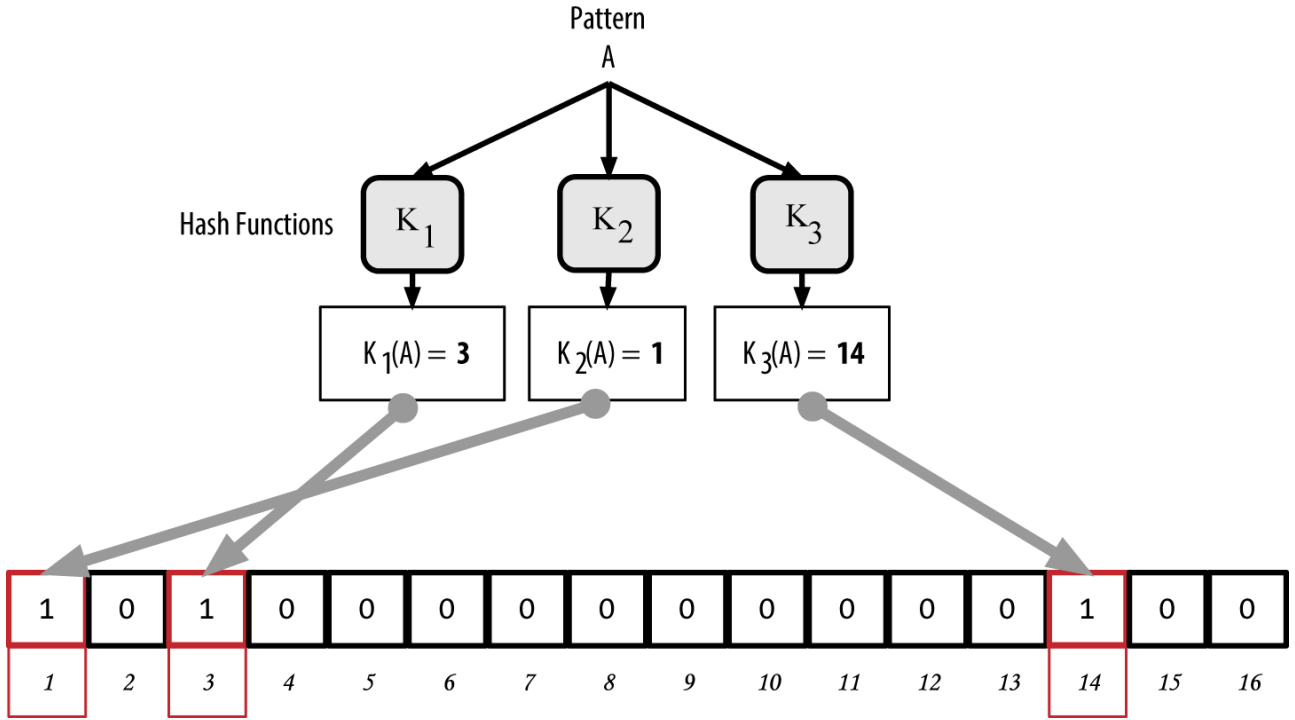
Filtros de Bloom



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

Filtros de Bloom

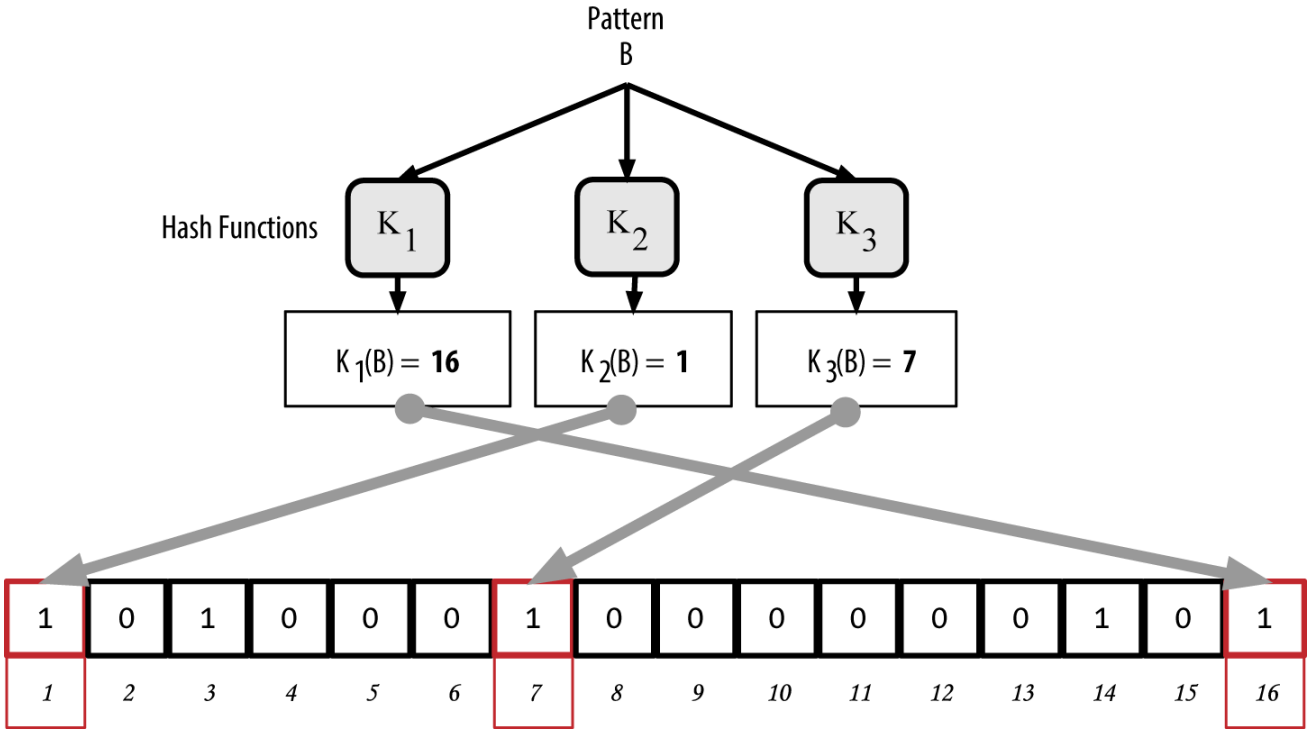
Inserción



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

Filtros de Bloom

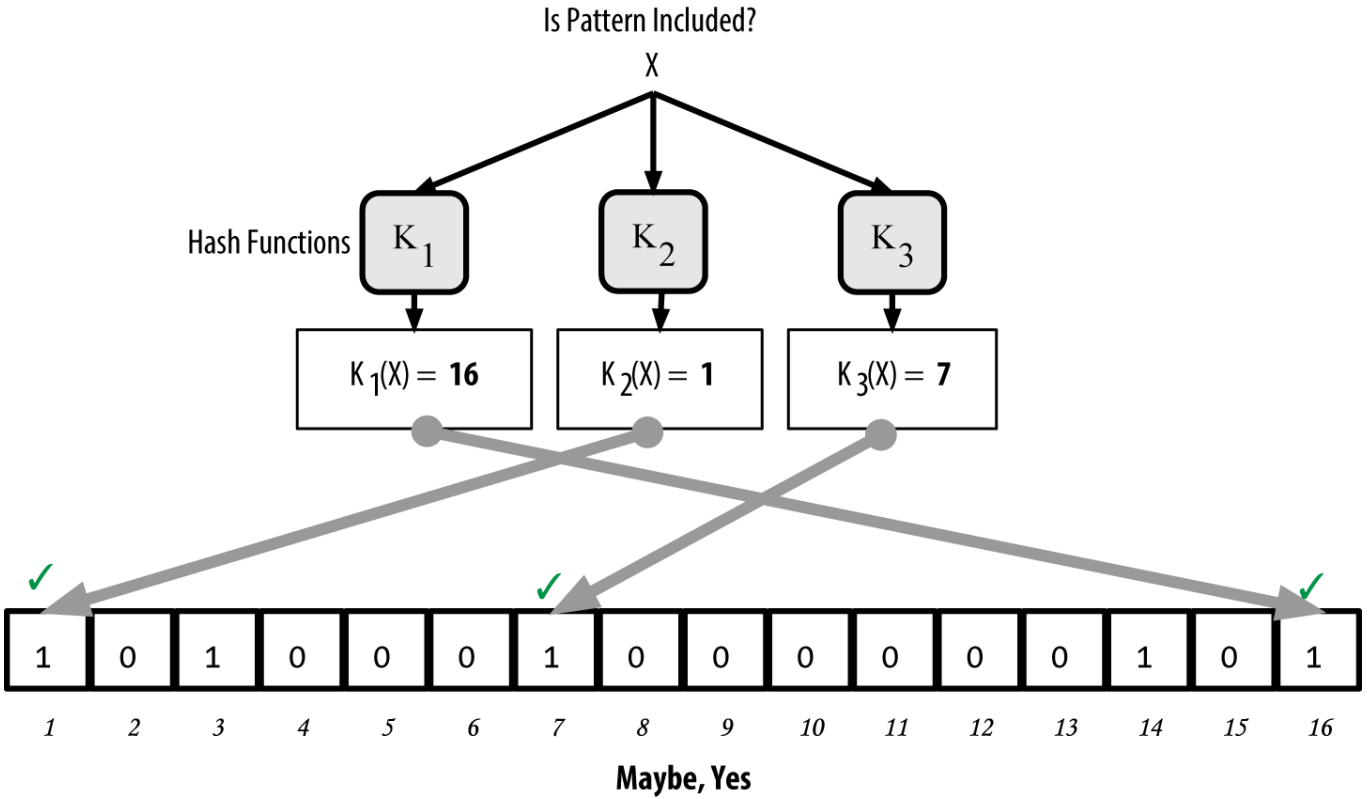
Inserción



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

Filtros de Bloom

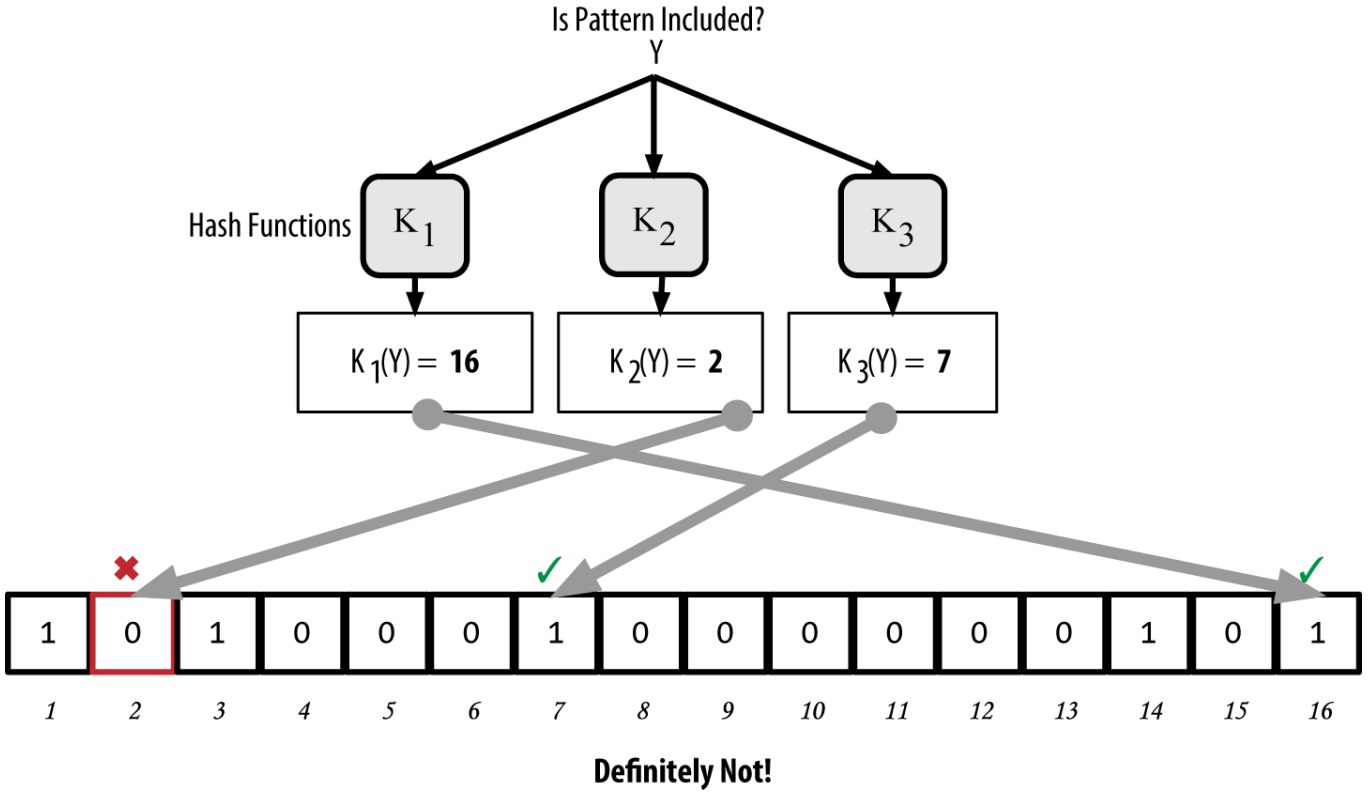
Busqeda



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

Filtros de Bloom

Busqeda



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

Nodos SPV y Filtros de Bloom

SPV A y su nodo completo B:

- A inserta las UTXOs y direcciones que le interesan a un BF
- A inserta un poco de basura en su BF
- A manda a B su BF
- B usa el BF en su canal de comunicación con A
- Cada transacción tx que B recibe, si tx hace match con BF, tx se manda a A

- A verifica si tx fue un falso positivo y en este caso descarta a tx

Nodos SPV y Filtros de Bloom

Qué revisa B en un transacción:

- Hash de la transacción
- Los inputs
- Los outputs
- Todas las direcciones

Leer más en:

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

Nodos SPV y Filtros de Bloom

Números usados el Bitcoin:

- Tamaño del filtro 36.000 bytes max
- 50 funciones de hash
- Full node manda el header + la prueba para las transacciones que hacen el match con el filtro

Leer más en:

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

Bitcoin network

Literatura:

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

Qué hace un minero solo:

- Compra hardware
- Gasta electricidad para computar los hashes

- Gana Bitcoins (de repente)

- Una varianza muy alta en lo bloques que descubre
- Puede pasar que no encuentra ningún bloque por dos años

- Un poco problemático en términos del costo

Mining pools

Solución:

- Muchos mineros se organizan en un mining pool
- Similar como una junta de granjeros/productores/inversionistas
- Todos resuelven el mismo puzle
- Pool manager le pasa el puzle = bloque

- Como el pool manager sabe cuánto debería pagar a un minero?
- Proporcional al trabajo que hizo
- Como medimos esto?

Mining pools

Mining shares:

- Una solución a mining puzle incompleta
- Puzle que saca el bloque tiene una dificultad de 70 zeros
- Un share tiene una dificultad de 50 zeros

- Cada vez que un minero encuentra un share, lo envía al manager
- Manager paga en proporción con el número de shares

- Puede pasar que el minero quien encontró el bloque gana menos que otros

Mining pools

Mining shares:

- Una solución a mining puzle incompleta
- Puzle que saca el bloque tiene una dificultad de 70 zeros
- Un share tiene una dificultad de 50 zeros

```
4AA087F0A52ED2093FA816E53B9B6317F9B8C1227A61F9481AFED67301F2E3FB
D3E51477DCAB108750A5BC9093F6510759CC880BB171A5B77FB4A34ACA27DEDD
00000000008534FF68B98935D090DF5669E3403BD16F1CDFD41CF17D6B474255
BB34ECA3DBB52EFF4B104EBBC0974841EF2F3A59EBBC4474A12F9F595EB81F4B
00000000002F891C1E232F687E41515637F7699EA0F462C2564233FE082BB0AF
0090488133779E7E98177AF1C765CF02D01AB4848DF555533B6C4CFCA201CBA1
460BEFA43B7083E502D36D9D08D64AFB99A100B3B80D4EA4F7B38E18174A0BFB
0000000000000000078FB7E1F7E2E4854B8BC71412197EB1448911FA77BAE808A
652F374601D149AC47E01E777613845618 1FA4F9D0EEDD8C4FDE3BEF6B1B7ECE
785526402143A291CFD60DA09CC80DD066BC723FD5FD20F9B50D614313529AF3
000000000041EE593434686000AF77F54CDE839A6CE30957B14EDEC10B15C9E5
9C20B06B01A0136F192BD48E0F372A4B9E6BA6ABC36F02FCED22FD9780026A8F
```

Mining pools

Mining shares:

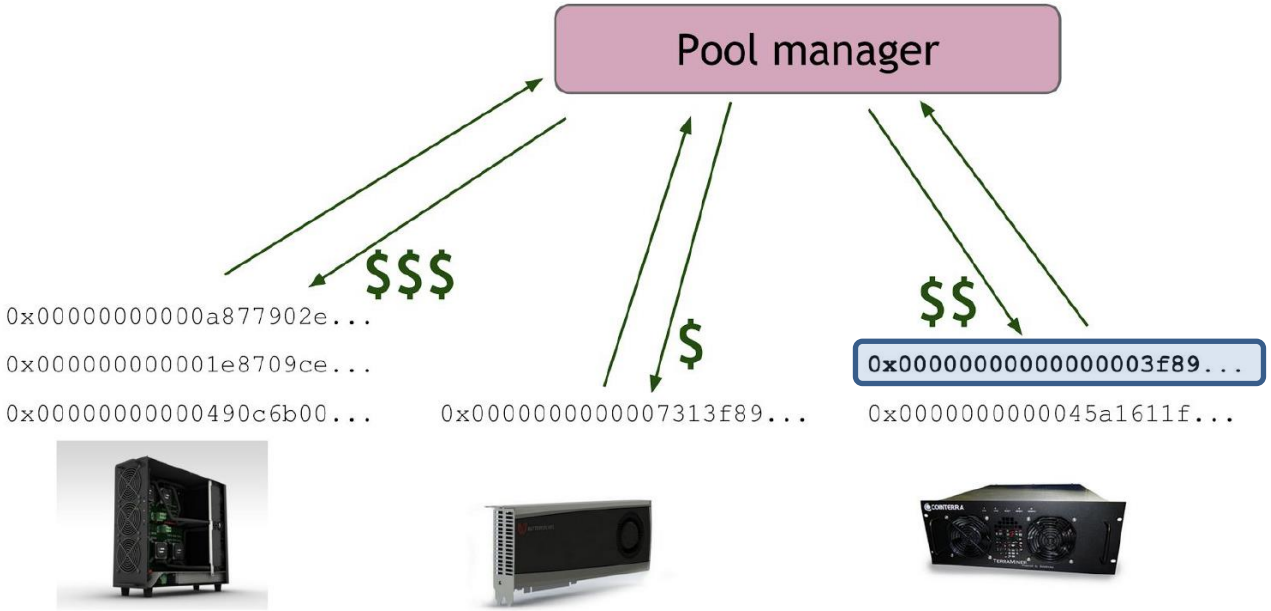
- Una solución a mining puzle incompleta
- Puzle que saca el bloque tiene una dificultad de 70 zeros
- Un share tiene una dificultad de 50 zeros

- Cada vez que un minero encuentra un share, lo envía al manager
- Manager paga en proporción con el número de shares

- Puede pasar que el minero quien encontró el bloque gana menos que otros

Mining pools

Puede pasar que el minero quien encontró el bloque gana menos que otros



Mining pools

Operando a un pool:

- Como pagar a los mineros?
- Op1: flat fee – manager asuma el riesgo (paga cuando no hay bloque)
- Op2: proporcional – manager paga cuando se encuentra un bloque

- Mineros pueden jugar con todo esto

Mining pools

Mining pools y ataque 51%

- Ghash.io – en 2014 tenía 50% de hash power

- El mercado se autoregula

Electricidad

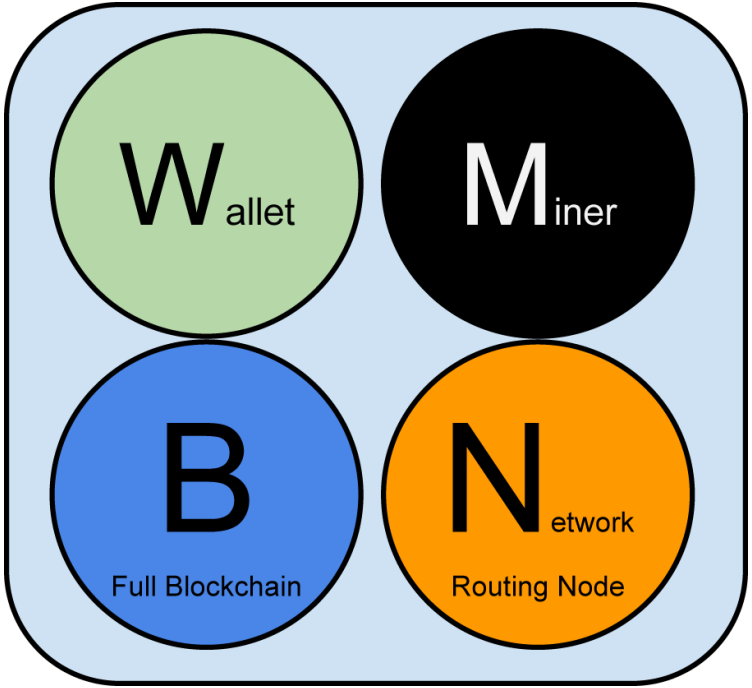
Dos escuelas de pensamiento:

1. Bitcoin mining is wasteful

2. Bitcoin mining es ecológico en largo plazo

Network recap

Node functionality



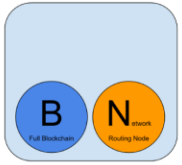
https://github.com/bitcoinbook/bitcoinbook/blob/develop/ch08.asciidoc

Network recap

Node type

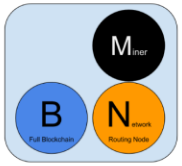
Reference Client (Bitcoin Core)

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



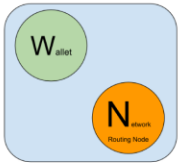
Full Block Chain Node

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

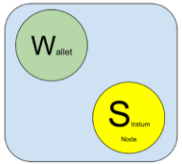
Contains a Wallet and a Network node on the bitcoin P2P protocol, without a blockchain.

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.

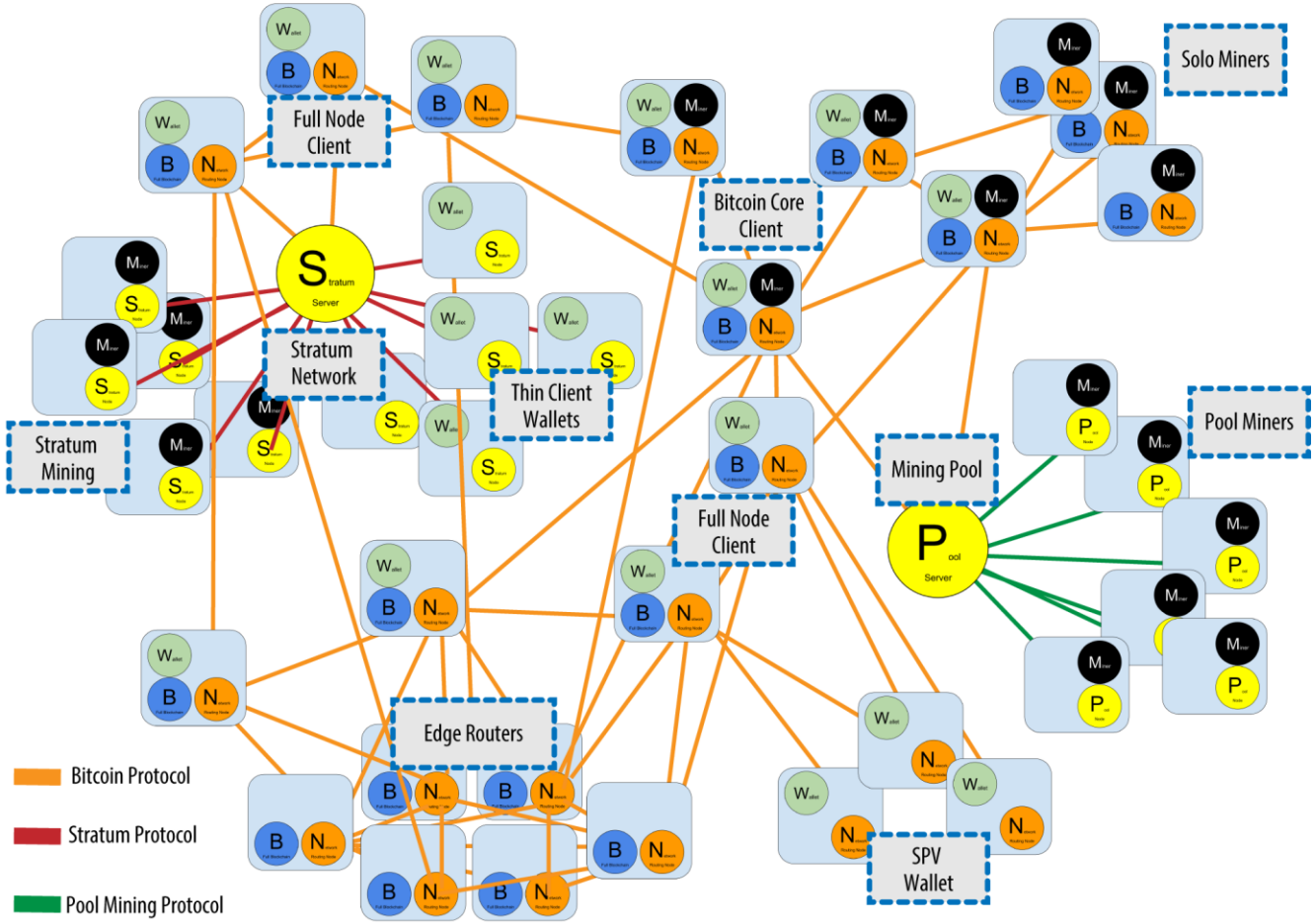


Lightweight (SPV) Stratum wallet

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

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

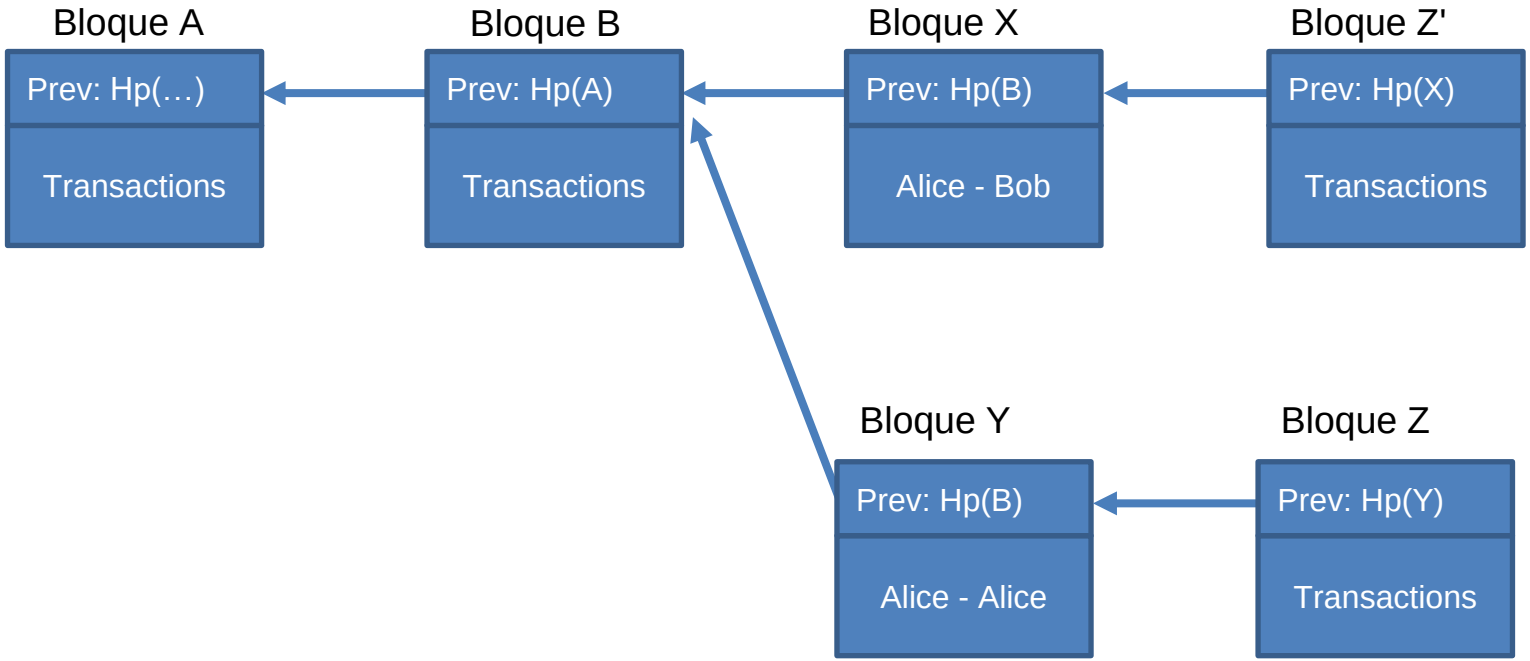
Network recap



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

Forks

Dos (o más ramas en el blockchain)



Forks

Un cambio de reglas:

- Todos los nodos deberían cambiar a su software
- No ocurre naturalmente

- **Hard forks vs. Soft forks**

Forks

Hard fork:

- Un cambio que no es forward compatible
- Se introducen reglas que antes no eran validas

- E.g. hay que firmar el doble hash de la transacción H(H(tx)), y no solo el H(tx)

- Nodos que hicieron el upgrade están bien, pero los antiguos no
- El blockchain se divide

- Ejemplo: Bitcoin Cash

Forks

Soft fork:

- Un cambio que es forward compatible
- Introduce reglas más estrictas

- Nodos con software antiguo van a aceptar todos los nuevos bloques
- Nodos con software nuevo van a rechazar algunos bloques antiguos
- Mineros antiguos pueden darse cuenta que hay que hacer el upgrade

- Ejemplo: (casi) cualquier BIP; e.g. P2SH