Presentation: OctupusTea

Ch.8
Bitcoin Network

P2P Network Architecture

- Inherently resilient
- Decentralized
- Open

Bitcoin Network

- Collections of nodes
- Bitcoin P2P Protocol

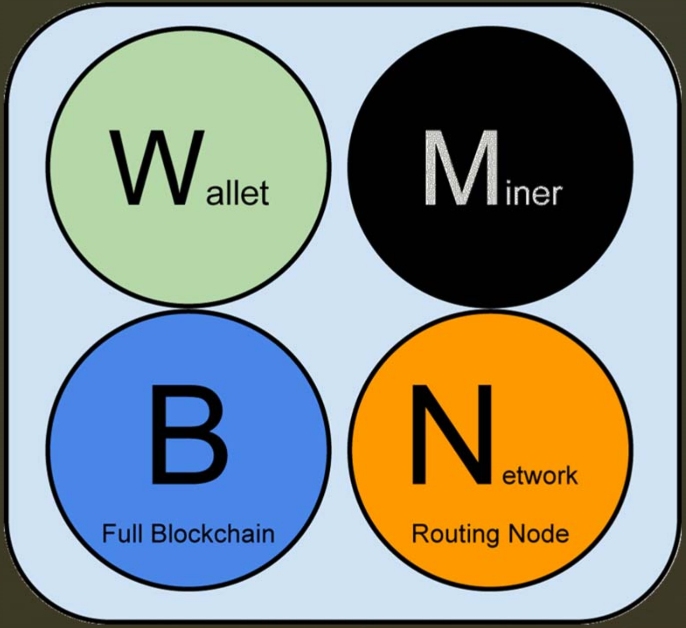
Extended Bitcoin Network

Extended Bitcoin Network

- Bitcoin P2P Protocol
- Pool-Mining Protocol
- Stratum Protocol
- Any other related protocols

Node Roles

- Collection of 4 functions:
  - Network (Routing)
  - BC DB
  - Miming
  - Wallet



The Ns: Network Routing

- Validate / Propagate TXs / blocks
- Discover Peers
- Maintain Connections

The Bs: Full Blockchain

- Full nodes
- Autonomously and authoritatively verify TXs.
- SPV nodes (not the Bs)

The Ms: Miner

- Mining nodes
- Create new blocks (PoW)
- Full node miner / Pool miner

The Ws: Wallets

- Bitcoin core / SPV wallets

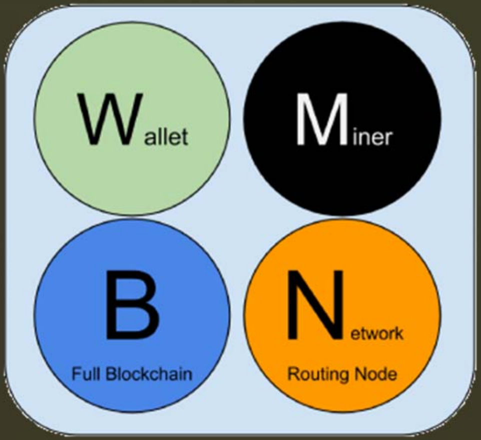
The Others

- Nodes running other protocols
  - Mining pool protocol
  - Light-weight client-access protocol

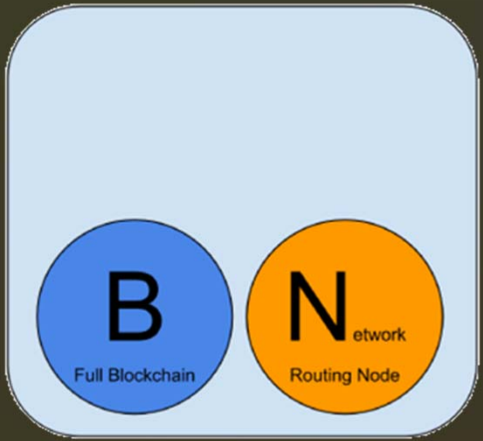
Common Node Types

Reference Client

(Bitcoin Core)

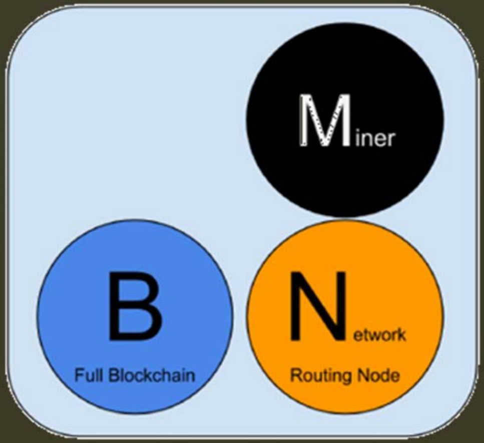


Full BC Node

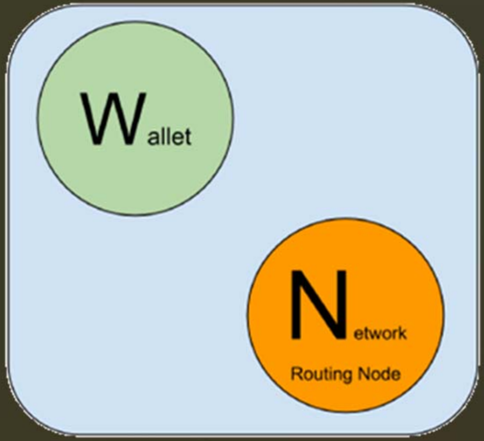


Common Node Types (ctd.)

Solo Miner

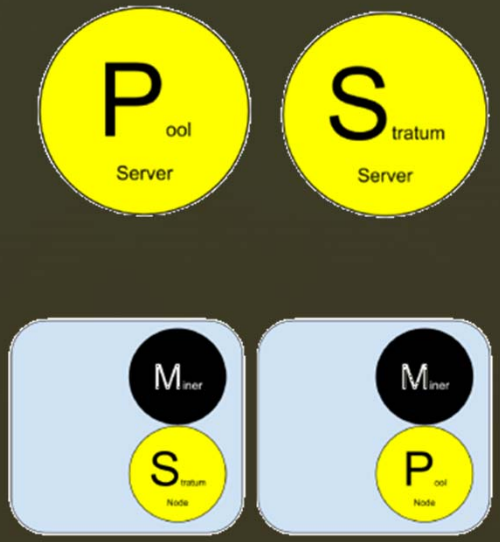


SPV Wallet

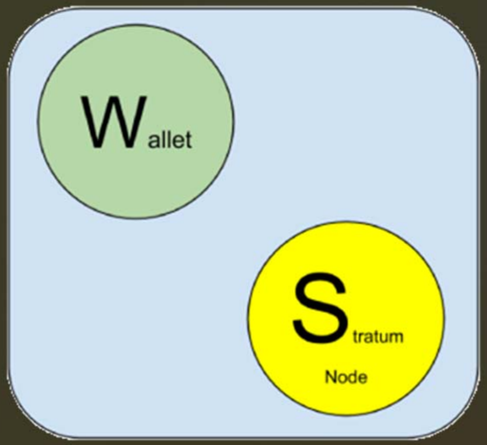


Common Node Types (ctd.)

Pool Protocol Servers & Mining Nodes



SPV Wallet

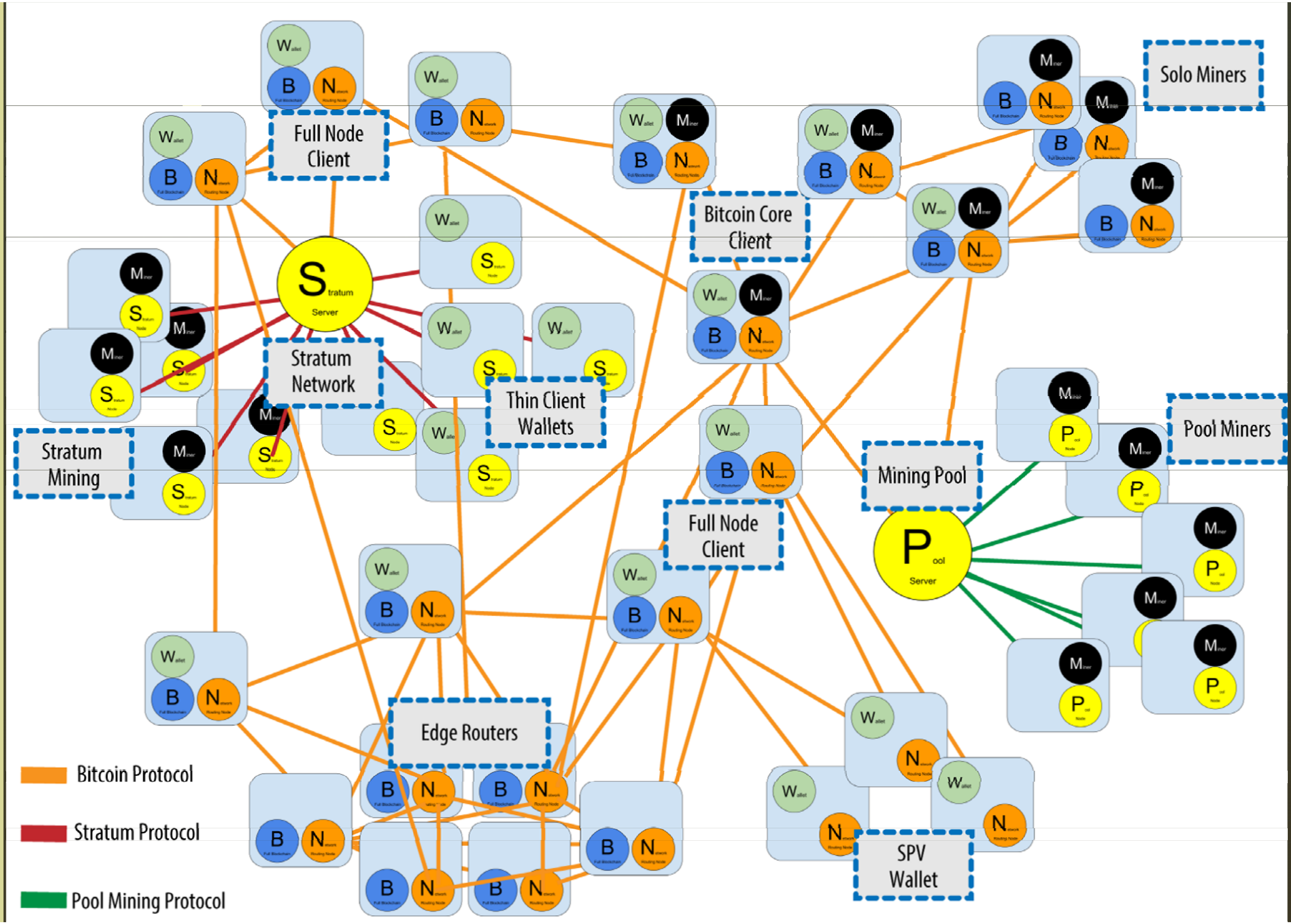


Full Nodes

- Maintain a full BC with all TXs
- In early years, all nodes are full nodes
- Independently and authoritatively verify TX
- Later, SPV nodes are introduced

SPV Nodes

- Most common form, especially wallets
- Only save block headers
- Special TX verification
- Not as safe as full nodes



Full Node Client

Solo Miners

Bitcoin Core Client

Stratum Network

Thin Client Wallets

Stratum Mining

Pool Miners

Mining Pool

Full Node Client

Edge Routers

SPV Wallet

🟧 Bitcoin Protocol

🟥 Stratum Protocol

🟩 Pool Mining Protocol

Bitcoin Relay Network

Bitcoin Relay Network

- Latency is related to profit.
- NOT replacements for P2P network
- But additional connectivity

Bitcoin Relay Network (ctd.)

- Original: Specialized host nodes
- Next original: FIBRE
- Falcon by Cornell University

Network Discovery

Protocol

- Booting
  - Connect to at least one existing node
  - Select at random
  - Generally use port 8333

Protocol (ctd.)

- Connection establishing
  - Handshake
  - `version` message is sent to remote
  - `verack` message is sent by remote if compatable

Protocol (ctd.)

- Peer finding
  - DNS query
    - DNS seeds
  - Given IP

Protocol (ctd.)

- Address exchanging
  - `getaddr` message is sent to remote
  - List of other peers is sent by remote

Protocol (ctd.)

- Path
  - No need to connect too many modes
  - Most-recent-success is remembered
  - Periodically message sending
    - Disconnection: 90-minute timeout

Exchanging “Inventory”

- Full nodes try construct a complete BC.
- Syncing
  1. Check `version` message for `BestHeight`
  2. Receive `version` message, compare to its own BC
  3. Exchange `getblocks` messages

Exchanging "Inventory" (ctd.)

- Share 500 hashes of blocks
  from the "the longer" BC
- Using an `inv` message

Exchanging "Inventory" (ctd.)

- SPV nodes use `getheader` message
  - Responding peer will send up to 2000 block headers in `header` message
  - TXs of interest are retrieved using `getdata` request
  - Privacy issue

Bloom Filter

Bloom Filter

- Problem: privacy risks of SPV nodes
- A probabilistic search filter
  - Can be tuned toward precision or privacy
  - More accurate at the expense of privacy

Implementation

- Foundation
  - *N*-bit array indexed 1 to *N*
  - *M* hash functions ranging [1, *N*]
  - Different *N* and *M* for accuracy or privacy.

Implementation (ctd.)

- Pattern recording
- Pattern matching

Application on SPV Nodes

- Requesting node
  - Bloom filter set to 0
  - List owning addresses, keys and hashes
    - Extracting PKH, SH, and TX ID from any UTXO
  - Send a `filterload` message to peers.

Application on SPV Nodes (ctd.)

- Peers
  - Check matching:
    - TX ID, TX inputs, input sig
    - Data components and / or witness scripts
  - Send back probably matchingTXs
    - `merkleblock` message

Application on SPV Nodes (ctd.)

- Back to requesting node
  - Discard false positives
  - Update UTXO and wallet balance
  - Modify the bloom filter for future matching

Filter modification

- `filteradd` message
- `filterclear` message
- Pattern removal via clear and resend.

Encrypted and Authenticated Connections

Encrypted and Authenticated Connections

- Originally, the network are entirely in the clear
  - Not a major concern for full nodes
  - But a big problem for SPV nodes.
- Solutions
  - BIP-150 and 151 / Tor transport

Tor Transport

- The Onion Routing network
- Encryption and encapsulation of data
- Randomized network paths
- Bitcoin Core supports Tor

P2P Authentication and Encryption

- BIP-150 (Peer authentication)
  - Optionsal
  - ECDSA
  - Requires BIP-151 communications

P2P Authentication and Encryption (ctd.)

- BIP-151 (P2P communication encryption)
- Overall benefits
  - Prevent MITM attack
  - Strengthen resistance of Bitcoin to surveillance.

TX Pools

Memory Pool (TX Pool)

- Per node temp list of unconfirmed TXs
  - Keep TXs known to the network but no yet on BC
  - Wallet nodes use TX pool to track incomings

Orphaned TX Pool

- Checked when a TX is added
- Matching orphans are then validated
  - Recursively find in the orphan pool

UTXO DB / Pool

- Not initialized empty
- Contains entries of UTXO
  - From all the way back to the genesis block
- May be a pool on local memory
- Or an indexed DB on mass storage

Differences

- TX / orphan pools represent local perspective
  - Might vary significantly between nodes
  - Only contains unconfirmed outputs
- UTXO pool represents the consensus
  - Vary little between nodes
  - Only contains confirmed outputs.