bitcoin

CS1699: Blockchain Technology and Cryptocurrency

9. The Bitcoin Network

Bill Laboon

Bitcoin Scripts

- Recall that all transactions actually include a script (written in the Bitcoin scripting language, Script)
- The majority of transactions on Bitcoin blockchain contain a simple script to transfer UTXOs
- But we can do more!

Green Addresses

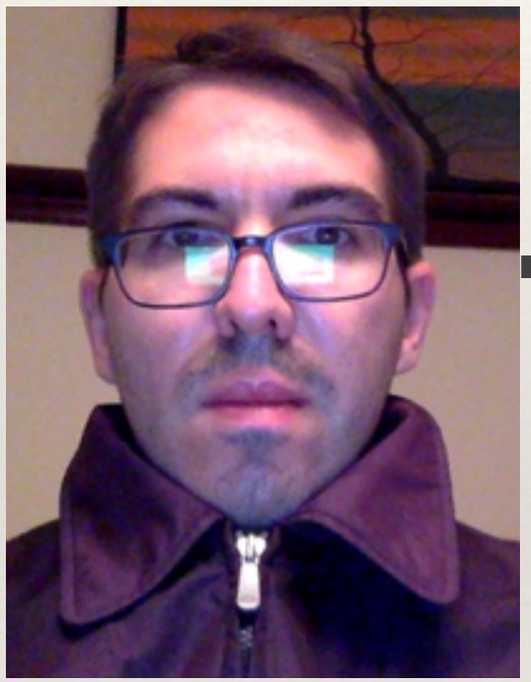
- Send bitcoin via third-party bank, even if both parties are offline
- Was exciting ten years ago… but turns out trusting third parties with bitcoin is a generally bad idea
- See: Mt. Gox - **NOT YOUR KEYS, NOT YOUR COINS**

Efficient Microtransactions

- Alice keeps signing transactions, each time adding more (e.g. 0.001 btc, 0.002 btc, 0.003 btc, etc.)
- Bob only signs final transaction - that is the only one that goes through
- Not used often - on-chain microtransactions have gone the way of the dodo
  - Microtransactions have turned out not to be used often in practice
  - Off-chain or second-tier mechanisms such as Lightning Network are taking over microtransactions

Escrow Transactions

Alice
(*buyer*)



bitcoin

goods / services

Bob
(*legitimate businessman*)

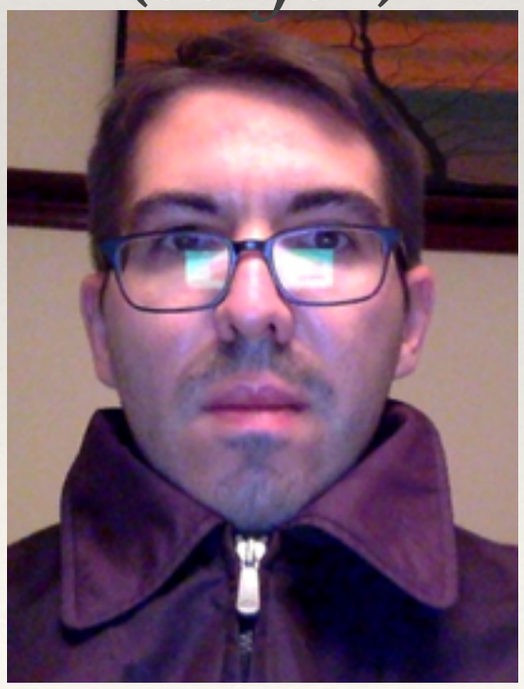
Remember: Bitcoin transactions are immutable once they are completed and on the blockchain!

Escrow Transactions w/ MULTISIG

Judy
(*third-party arbiter*)

m-of-n (2 of 3)

Alice
(*buyer*)



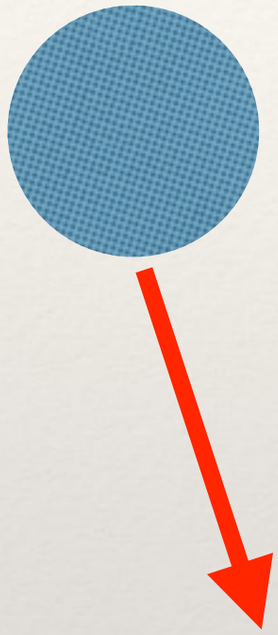
bitcoin (P2SH)

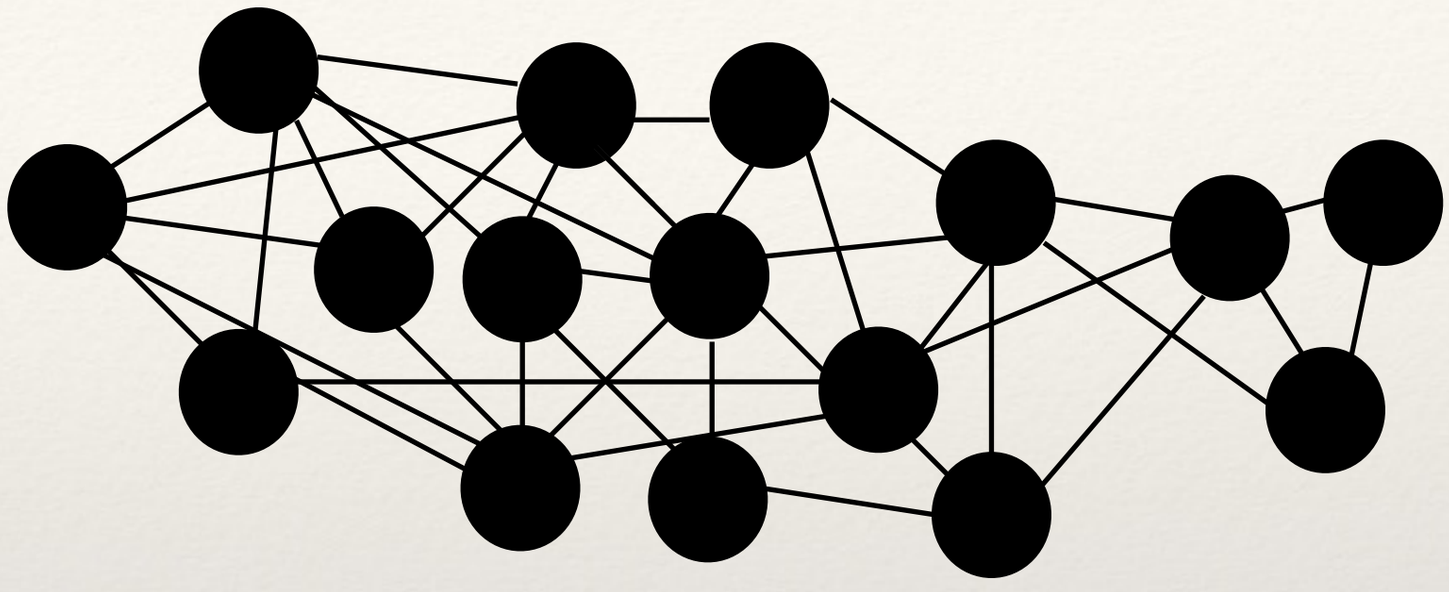
Bob
(*legitimate businessman*)

goods / services

Joining the Bitcoin Network

Where should I connect?



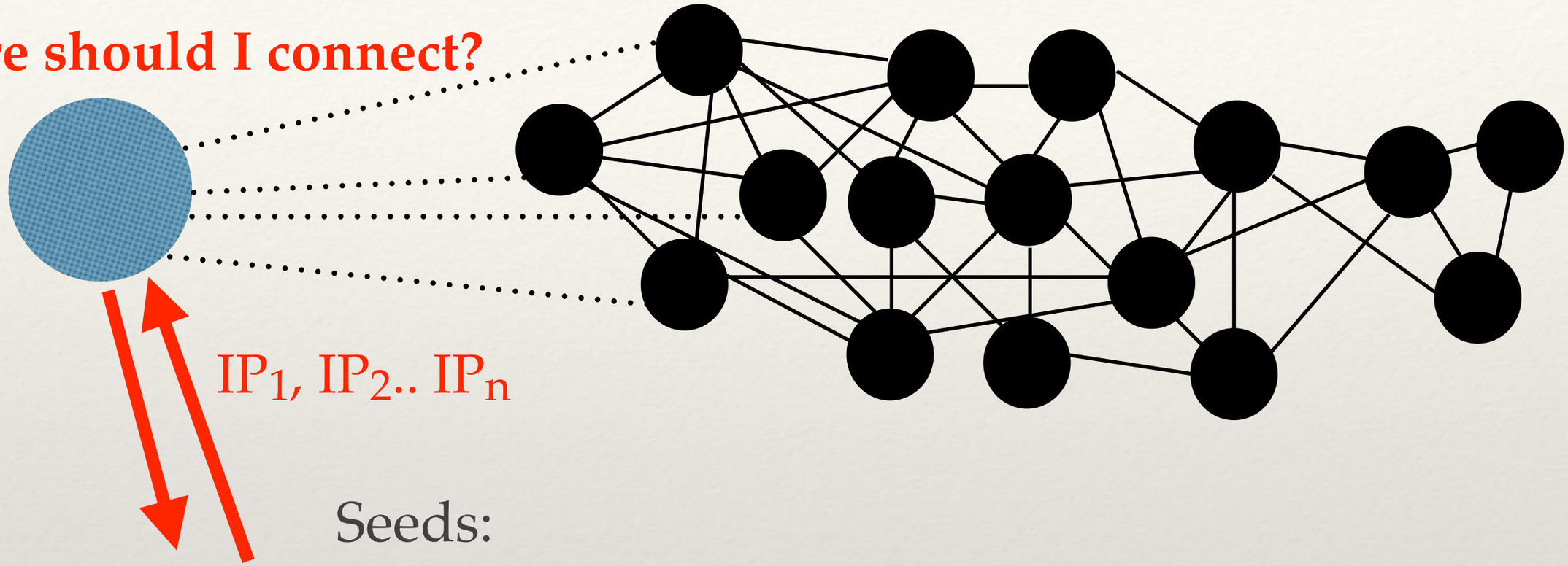


Seeds:

seed.bitcoin.sipa.be
dnsseed.bluematt.me
dnsseed.bitcoin.dashjr.org
seed.bitcoinstats.com
seed.bitcoin.jonasschnelli.ch
btc.petertodd.org
seed.bitcoin.sprovoost.nl

Joining the Bitcoin Network

Where should I connect?



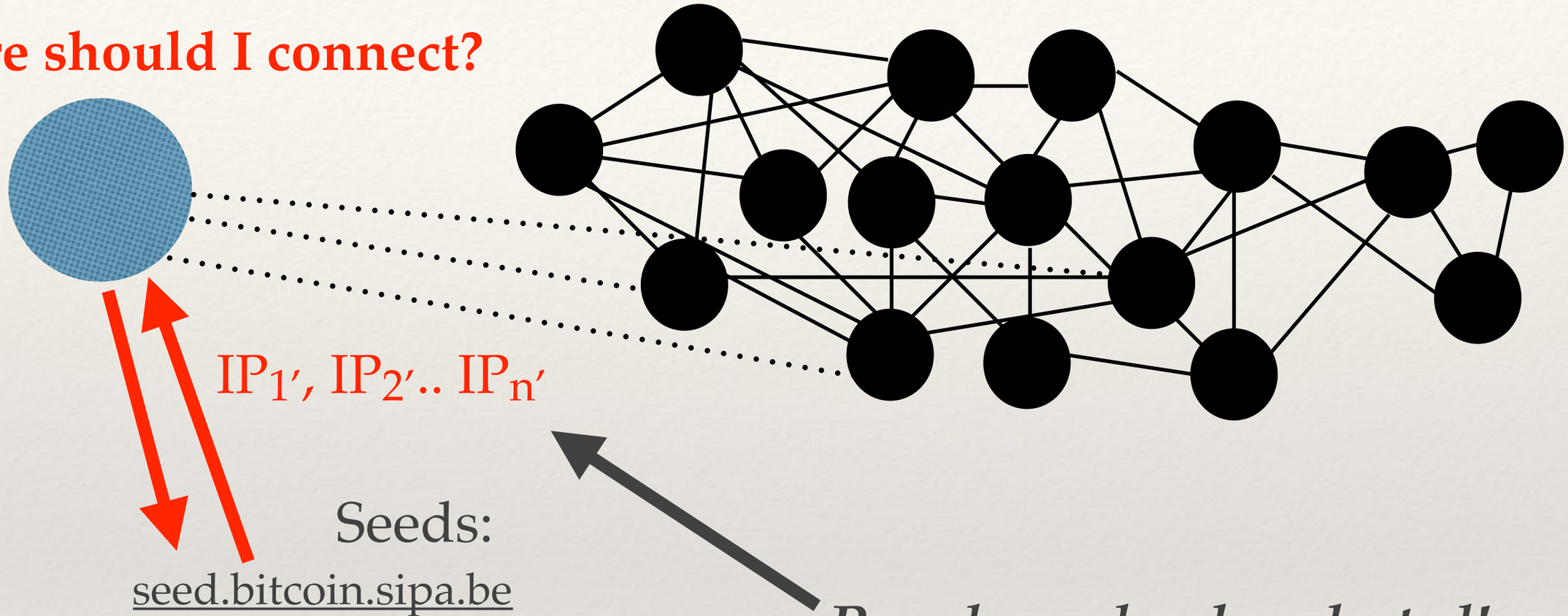
Seeds:

seed.bitcoin.sipa.be
dnsseed.bluematt.me
dnsseed.bitcoin.dashjr.org
seed.bitcoinstats.com
seed.bitcoin.jonasschnelli.ch
btc.petertodd.org
seed.bitcoin.sprovoost.nl

Seeds taken from: https://github.com/bitcoin/bitcoin/blob/master/src/chainparams.cpp

Dynamic Random Topology

Where should I connect?



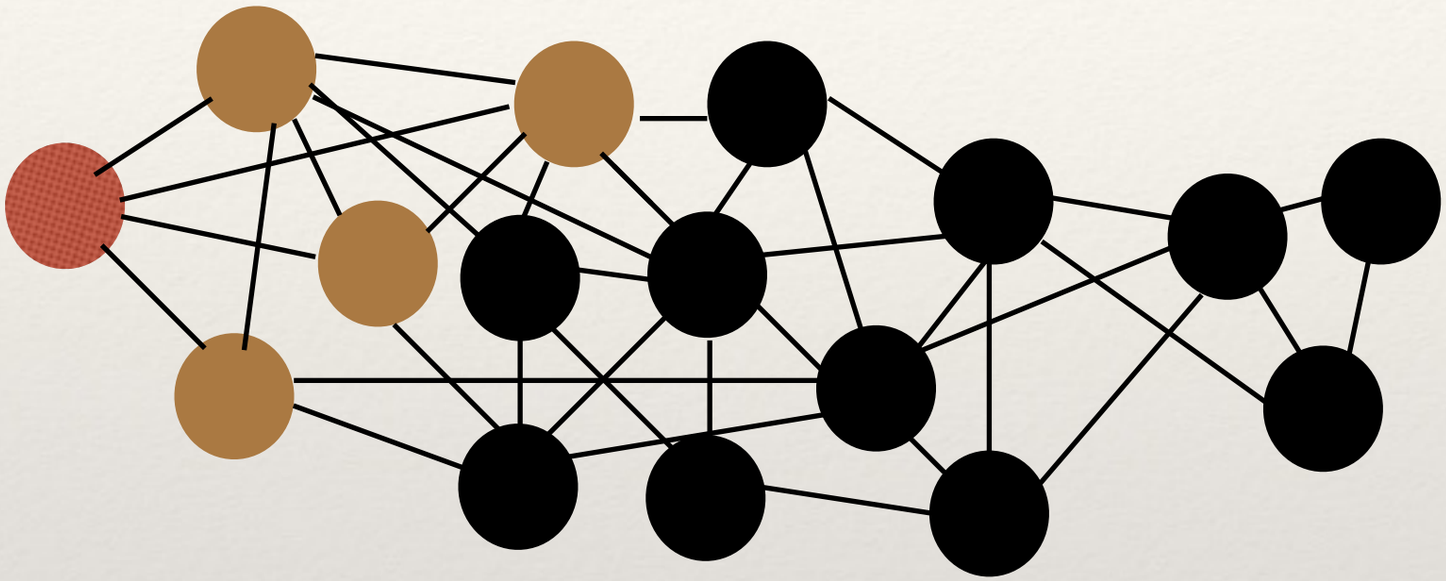
Seeds:

seed.bitcoin.sipa.be
dnsseed.bluematt.me
dnsseed.bitcoin.dashjr.org
seed.bitcoinstats.com
seed.bitcoin.jonasschnelli.ch
btc.petertodd.org
seed.bitcoin.sprovoost.nl

Pseudorandomly selected!

The Gossip Protocol

Initial node - Broadcasts transactions to all of its peers



Each receiving node checks for validity

- If valid, passes to its peers

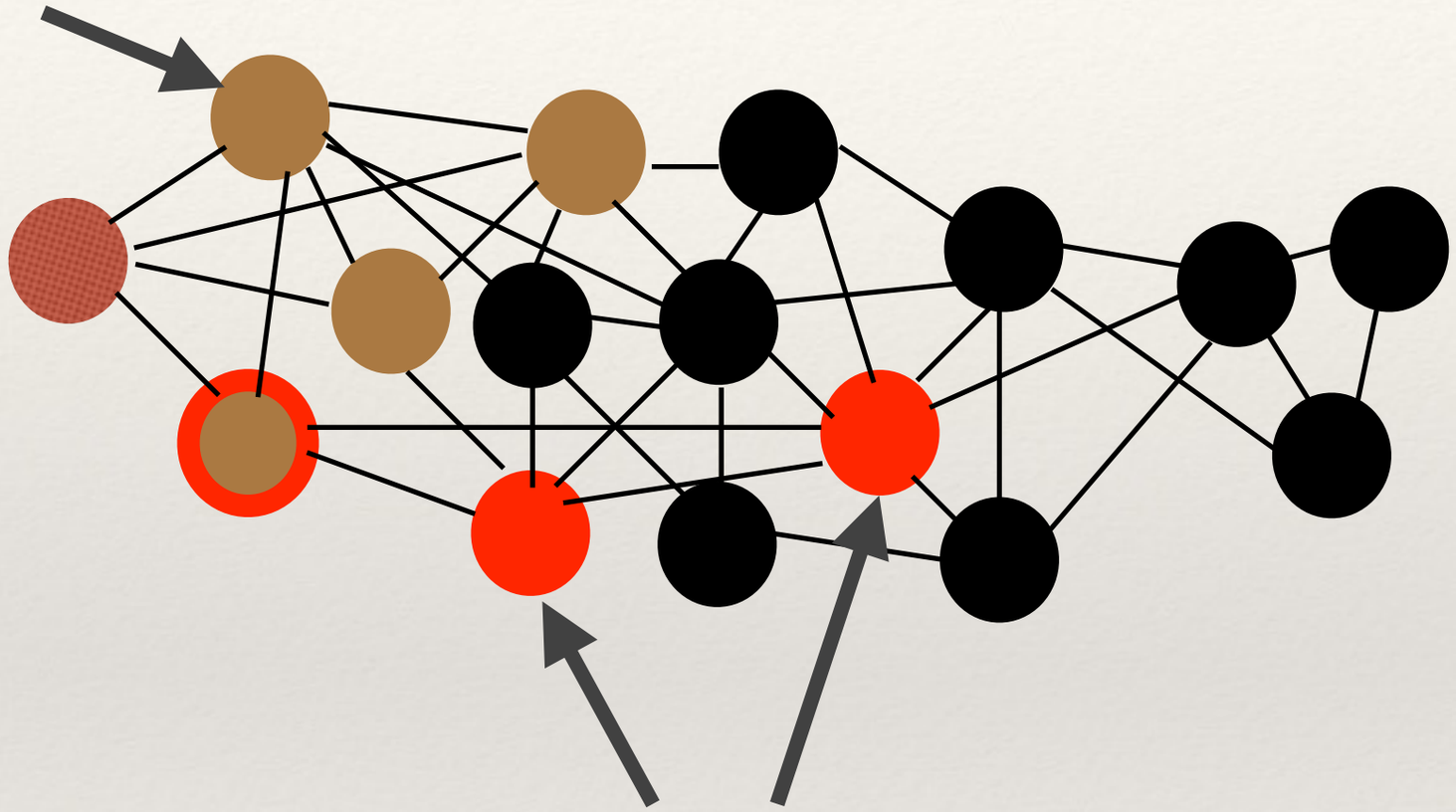
- If invalid, drops it

Valid Transactions at the Gossip Level

- Is the transaction valid within the current block chain? (i.e., does **`scriptSig || scriptPubKey`** return true?)
- Have the transaction outputs being used as inputs not already been redeemed (i.e., are they unspent?)
- Has this node not seen this transaction before?
- Is the script whitelisted?

The Gossip Protocol

"I already have this tx (checked against hash), I will ignore it"



"New valid transaction!
I will add it to the transaction pool, then pass it on."

Transaction Propagation

- Relatively slow process (decentralization and efficiency are often at loggerheads)
- Note that block propagation nowadays is much faster than in the book: https://bitnodes.earn.com/dashboard/?days=90
- Interestingly, though the number of fully validating nodes has remained about the same! (~10,000)

Storage (as of 26 September 2018)

- Current size of the Bitcoin blockchain:
  ~ 183,897 megabytes (monotonically increasing)
- https://www.blockchain.com/charts/blocks-size?
- Size of the Bitcoin transaction pool over the last week:
  192,629 bytes - 9,112,522 bytes (Max: 12 Jan 2018 - 126 MB)
- https://www.blockchain.com/charts/mempool-size
- If you really want to get down in the weeds:
  https://jochen-hoenicke.de/queue/#1,24h

Fully Validating Nodes

- Fully validating nodes:
  - Connect to the Bitcoin network and act as a full peer
  - Download and verify the entire blockchain (generally > 24 hours to do so)
  - Verify / propagate / drop transactions
  - Broadcast transactions

SPV ("Lightweight") Nodes

- Simple Payment Verification
- Not as secure as a fully validating node - only downloads headers of blocks, so checks that blocks and their hashes are valid, but not every single transaction in the block
- Can ONLY validate transactions that "affect them", not the entire network
- The majority of nodes on the Bitcoin network are SPV nodes - if you use a Bitcoin wallet, chances are it is an SPV node.

Why Run a Lightweight Node?

- Storage/CPU usage savings - ~ three orders of magnitude (1/1000th) compared to a fully validating node
- Very little “lag time” when spinning up a node

The Evolution of Bitcoin

- Want to change Bitcoin? File a BIP (https://github.com/bitcoin/bips/blob/master/README.mediawiki). Some notable active BIPs:
  - Already mentioned BIP-34: Changed coinbase attribute to include current height
  - BIP-11: Added MULTISIG support (https://github.com/bitcoin/bips/blob/master/bip-0011.mediawiki) - Used for escrow
  - BIP-13: Added P2SH (https://github.com/bitcoin/bips/blob/master/bip-0013.mediawiki)
  - BIP-141: Added SegWit support (https://github.com/bitcoin/bips/blob/master/bip-0141.mediawiki) - Prevent transaction malleability, allow second-tier scaling to take place

Kinds of Forks

- **Hard fork**: Introduce new features were previously considered invalid; previous versions of software will not accept new blocks
  - Can lead to chain splits (“coin forks”, e.g. Bitcoin Cash, Bitcoin Gold, Bitcoin Private)
- **Soft fork**: Make validation rules stricter; previous versions of software will still accept blocks produced under the stricter rules

Changes to Consensus

- How do people decide?
- If nodes accept changes, that change continues
- Other nodes can follow along or go off on their own - which has happened before!
  - Bitcoin Classic, Bitcoin XT, Bitcoin Unlimited - all are now subsumed under Bitcoin Cash

Limitations of Bitcoin Network

- Known implementation bugs (e.g. MULTISIG instruction popping multiple values off of the stack)
- Transactions per second (~ 7)
- Fixed cryptographic hashing algorithms (SHA-256, RIPEMD-160, ECDSA/secp256k1)
- Future/minor issues: divisibility smaller than satoshis, number of operations per block
- Not likely to ever be modified: number of bitcoin produced, mean time between blocks, block rewards/halvening
- Difficult/impossible to fix most of these without a hard fork