April 28, 2021

1 Bitcoin peer-to-peer protocol

Introduction to btc network, has to glue together all the parts of this section. Clients versions/ variants - how different clients interact - networking interface almost untouched. This thesis refers to latest version of bitcoin core.

1.1 Peer discovery, connection and behaviour upon connection

Nodes on fresh bootstrap make use of hardcoded *DNS seeds* as their first mean to discover other peers. DNS seed servers are maintained by community members and answer queries providing a list of addresses that can be either dynamically gathered through a periodic scan of the network or manually updated by server administrators. As a fallback option the user is able through command line to specify a list of addresses the client can connect to. Were both these two options to fail the client has a hardcoded list of peers it can directly connect to, though this is considered the last resort for bootstrapping peers. Nodes on startup that were previously in the network shall first lookup for peer names in their local address database, implemented as "peer.dat". The database contains the address of each peer the node has come to know during its lifetime in the network. If the node has disconnected for a time too long, many of the addresses in the database may have become outdated or unreachable. A node that cannot connect to any address in the peer database or has spent up to eleven seconds trying to connect unsuccessfully to at least one of the peers in the database behaves as on fresh bootstrap and resolves to query a DNS first. The use of a local address database, also called *peer cache*, provides for reconnecting peers a fully-decentralized way to join the network and is the first line of defense against *fake bootstrap attacks*; more can be found in Chapter $< num >$.
In order to establish a connection two peers need to exchange version messages in an handshake fashion. The node that first sends a `version` message is said to be establishing an *outgoing connection.* The receiving peer will eventually send back a `verack` message followed by another `version` message. This latter node is said to be establishing an *ingoing connection.* Once the second `version` message is answered with a `verack` the connection is set up. Each node is able to establish up to 8 outgoing connections and 117 ingoing connections, for a total of 125 connections. In order to keep its connections active, each peer sends a message to each neighbour at least once every thirty minutes. If more ninety minutes pass without receiving anything from a neighbour, the connection is dropped.

The peer discovery process after the first connection of a node is carried on through the exchange of `addr` messages.