Bitcoin node

Bitcoin Core application has reference client:

QT GUI or Daemon

CLI RPC interface

All in a singe application:
https://bitcoin.org/en/download
https://bitcoincore.org/en/download/

RPC Documentation:
https://bitcoincore.org/en/doc/0.21.0/

Bitcoin network

See https://en.bitcoin.it/wiki/Network and Core source code.

200GB Blockchain size for Full nodes.

https://en.bitcoin.it/wiki/Network

TCP on 8333 port is used by default, IPv4 and IPv6.

Default maximum 125 connections (https://github.com/bitcoin/bitcoin/blob/master/src/net.h#L71), 8 outbound (https://github.com/bitcoin/bitcoin/blob/master/src/net.h#L59) and 117 incoming.

Outgoing connection to nodes from other group, not in /16 IPv4 network (https://github.com/bitcoin/bitcoin/blob/master/src/net.cpp#L1713)

Bitcoin Peer Discovery

See https://en.bitcoin.it/wiki/Satoshi_Client_Node_Discovery

The Satoshi client discovers the IP address and port of nodes in several different ways.
- Nodes discover their own external address by various methods.
- Nodes receive the callback address of remote nodes that connect to them.
- Nodes makes DNS request to receive IP addresses.
- Nodes can use addresses hard coded into the software.
- Nodes exchange addresses with other nodes.
- Nodes store addresses in a database and read that database on startup.
- Nodes can be provided addresses as command line arguments
- Nodes read addresses from a user provided text file on startup

1) Peer addresses from previous connection.
2) IRC seeding from lfnet. Before 0.6 IRC irc.lfnet.org, joining a random channel between #bitcoin00 and #bitcoin99, nicknames were hosts. lfnet went down. Deprecated in 2011.
3) DNS seeding from bitseed.xf2.org, Texas, NS*.DNSPARK.NET, DNSPARK, LLC, USA.
https://github.com/bitcoin/bitcoin/commit/f684aec4f38d6a9e48e870ca5dae6bd65da516cf
https://github.com/bitcoin/bitcoin/commit/f684aec4f38d6a9e48e870ca5dae6bd65da516cf#diff-c61ed5adfba5354ced525ffbab3691f1
https://github.com/bitcoin/bitcoin/blob/1b2460bd5824170ab85757e35f81197199cce9d6/src/chainparams.cpp#L112

vSeeds.push_back(CDNSSeedData("bitcoin.sipa.be", "seed.bitcoin.sipa.be")); // Pieter Wuille
vSeeds.push_back(CDNSSeedData("bluematt.me", "dnsseed.bluematt.me")); // Matt Corallo
vSeeds.push_back(CDNSSeedData("dashjr.org", "dnsseed.bitcoin.dashjr.org")); // Luke Dashjr
vSeeds.push_back(CDNSSeedData("bitcoinstats.com", "seed.bitcoinstats.com")); // Christian Decker
vSeeds.push_back(CDNSSeedData("xf2.org", "bitseed.xf2.org")); // Jeff Garzik
vSeeds.push_back(CDNSSeedData("bitcoin.jonasschnelli.ch", "seed.bitcoin.jonasschnelli.ch")); // Jonas Schnelli

Bitcoin Network Performance

See https://www.researchgate.net/publication/260593322_The_Bitcoin_P2P_Network

As of 2014 …

10k public full nodes.

800k+ nodes estimate.

50% of blocks are broadcasted to 25% nodes in less then 22 seconds.

50% of transactions are broadcasted to 25% nodes in less then 17 minutes.

Bitcoin.conf

```
# See https://github.com/bitcoin/bitcoin/blob/master/share/examples/bitcoin.conf

# Run on the test network instead of the real bitcoin network.
testnet=1

# server=1 tells Bitcoin-Qt and bitcoind to accept JSON-RPC commands
server=1

rpcallowip=0.0.0.0/0
rpcuser=bitcoinuser
rpcpassword=bitcoinsomepass456

# Enable pruning to reduce storage requirements by deleting old blocks.
# This mode is incompatible with -txindex and -rescan.
# 0 = default (no pruning).
# 1 = allows manual pruning via RPC.
# >=550 = target to stay under in MiB.
prune=550

# Options only for testnet
[test]
# Only existing wallets can not create new wallet in new clients
#wallet=/home/sin/Univer/Blockchain/Bitcoin/wallet
rpcport=8332
```

Execution script

Linux/MacOS:

/home/sin/Univer/Blockchain/Bitcoin/bitcoin-0.19.1/bin/bitcoind
-conf=/home/sin/Univer/Blockchain/Bitcoin/bitcoin.conf
-datadir=/home/sin/Univer/Blockchain/Bitcoin/data

Windows:

Add .exe, fix path.

Bitcoin RPC cli

Use a helper script.

bitcoin-cli.sh (Linux, MacOS):

```
/home/sin/Univer/Blockchain/Bitcoin/bitcoin-0.19.1/bin/
bitcoin-cli
-conf=/home/sin/Univer/Blockchain/Bitcoin/bitcoin.conf $1 $2
$3 $4
```

bitcoin-cli.bat (Windows):

```
C:\Bitcoin\bitcoin-0.19.1\bin\bitcoin-cli.exe -conf=c:\
Bitcoin\bitcoin.conf %1 %2 %4 %4
```

Bitcoin RPC cli

Check connection:

```
./bitcoin-cli.sh getblockchaininfo
```

Output:

```
{
  "chain": "test",
  "blocks": 1723123,
  "headers": 1723123,
  "bestblockhash":
"000000003d19344bff5b4e28a992fd9bea16556461d2d8b11f951de3569ccedd",
  "difficulty": 1,
  "mediantime": 1588766109,
  "verificationprogress": 0.999999294498542,
  "initialblockdownload": false,
  "chainwork":
"0000000000000000000000000000000000000000000015981fc24cdcb94c473",
  "size_on_disk": 481001503,
  "pruned": true,
...
```

Bitcoin RPC cli

Get unspent transactions to check payments:

```
./bitcoin-cli.sh listunspent
```

Output:


```
[
  {
    "txid":
"1a36a7f4212b8748bb42d3ff367c4959d87edb988e6dbf1466873f76d32cb46a",
    "vout": 1,
    "address": "2N9xiUgX3tdqrK2C2sg6pDigZZARpYy91Kf",
    "redeemScript": "00148b4ff313796dd9dc0544f61d80b601d7f2fe0c12",
    "scriptPubKey": "a914b75ad95276977933ab6894b1627641696147l8a587",
    "amount": 0.00008830,
    "confirmations": 29725,
    "spendable": true,
    "solvable": true,
    "desc":
"sh(wpkh([6b9fc73d/0'/1'/1']036bc9f07dc1ddb1c2c6e066e6122089eceeead54
bfe547b2c87992b9db3640d9e))#mat8d9fq",
    "safe": true
  }
]
```

Bitcoin RPC cli

Get private key of an address:

```
./bitcoin-cli.sh dumpprivkey 2N9xiUgX3tdqrK2C2sg6pDigZZARpYy91Kf
```

Output:

```
cW2xV9kfELBaqQ8KFy6zxvRxKkzE7nfgmxgxDu3KQimMnYzmd9p8
```

Bitcoin RPC cli

Generate new address:

```
./bitcoin-cli.sh getnewaddress
```

Output:

```
2MyJfUK626P7ucw2QZcsZUYNVRyJft8zdiE
```

Generate new address 2:

```
./bitcoin-cli.sh getnewaddress
```

```
2Mt2Sx452xkHPCbuenRUvYJsho9Z3v136iS
```

Bitcoin RPC cli

Create transaction to send to two addresses, Outputs - Inputs = transaction fee (0.00001):

```
./bitcoin-cli.sh createrawtransaction '''
    [
      {
        "txid":
"'1a36a7f4212b8748bb42d3ff367c4959d87edb988e6dbf1466873f76d32cb46a'",
        "vout": '1'
      },
    ]
    ''' '''
    {
      "'2MyJfUK626P7ucw2QZcsZUYNVRyJft8zdiE'": 0.00004,
      "'2Mt2Sx452xkHPCbuenRUvYJsho9Z3v136iS'": 0.00003
    }'''
```

Output:

```
02000000016ab42cd3763f876614bf6d8e98db7ed859497c36ffd342bb48872b21f4a
7361a0100000000ffffffff02a00f000000000017a914427746624169765 8f07f4c
4dd4fcb13bd52be5d587b80b000000000017a914088d89bcb985efa0e89749d7c2d
a2294b97a8ca08700000000
```

Bitcoin RPC cli

Sign raw transaction:

```
./bitcoin-cli.sh signrawtransactionwithkey
"02000000016ab42cd3763f876614bf6d8e98db7ed859497c36ffd342bb48872b21f4
a7361a0100000000ffffffff02a00f000000000017a914427746624169765 8f07f4
c4dd4fcb13bd52be5d587b80b000000000000017a914088d89bcb985efa0e89749d7c2
da2294b97a8ca08700000000"
'["cW2xV9kfELBaqQ8KFy6zxvRxKkzE7nfgmxgxDu3KQimMnYzmd9p8"]'
'[{"txid":"1a36a7f4212b8748bb42d3ff367c4959d87edb988e6dbf1466873f76d3
2cb46a","vout":1,"scriptPubKey":"a914b75ad95276977933ab6894b162764169
614718a587","redeemScript":"00148b4ff313796dd9dc0544f61d80b601d7f2fe0
c12","amount":0.00008830}]'
```

Output:

```
{
  "hex":
"020000000001016ab42cd3763f876614bf6d8e98db7ed859497c36ffd342bb48872b21f4a7361a
0100000171600148b4ff313796dd9dc0544f61d80b601d7f2fe0c12ffffffff02a00f000000000
00017a914427746624169765 8f07f4c4dd4fcb13bd52be5d587b80b000000000000017a914088d89
bcb985efa0e89749d7c2da2294b97a8ca0870247304402201203dee4915b83ea39ac6fc3f1f4583
2978c29eb2f2874e7f50ec07d52272a8b0220683c0c3ba6244a58daa6426abb9852f8af5d5898a3
d4eadd88810977aac9a9180121036bc9f07dc1ddb1c2c6e066e6122089eceeead54bfe547b2c879
92b9db3640d9e00000000",
  "complete": true
}
```

Bitcoin RPC cli

Send raw transaction:

```
./bitcoin-cli.sh sendrawtransaction
02000000000101016ab42cd3763f876614bf6d8e98db7ed859497c36ffd342bb48872b2
1f4a7361a01000000171600148b4ff313796dd9dc0544f61d80b601d7f2fe0c12ffff
ffff02a00f000000000000017a914427746624169765 8f07f4c4dd4fcb13bd52be5d58
7b80b000000000000017a914088d89bcb985efa0e89749d7c2da2294b97a8ca0870247
304402201203dee4915b83ea39ac6fc3f1f45832978c29eb2f2874e7f50ec07d52272
a8b0220683c0c3ba6244a58daa6426abb9852f8af5d5898a3d4eadd88810977aac9a9
180121036bc9f07dc1ddb1c2c6e066e6122089eceeead54bfe547b2c87992b9db3640
d9e00000000
e16bc1c3bd7dfd3ce763afe048966e9f75ee43d8b6e9b1aa86f8064a008736cc
```

```
Output:

no
```

Check results in a couple of minutes later:

./bitcoin-cli.sh listunspent


```
[
  {
    "txid": "e16bc1c3bd7dfd3ce763afe048966e9f75ee43d8b6e9b1aa86f8064a008736cc",
    "vout": 0,
    "address": "2MyJfUK626P7ucw2QZcsZUYNVRyJft8zdiE",
    "label": "",
    "redeemScript": "001490eba93f448a74ea5d33f5cbdc821d1c1d09bb72",
    "scriptPubKey": "a914427746624169765 8f07f4c4dd4fcb13bd52be5d587",
    "amount": 0.00004000,
    "confirmations": 14,
    "spendable": true,
    "solvable": true,
    "desc": "sh(wpkh([6b9fc73d/0'/0'/9']0328a93231c0d81c2c14b87e5868fca48f0d96defd42eccfc80be4829aae2b343b))#y3zxcwrm",
    "safe": true
  },
  {
    "txid": "e16bc1c3bd7dfd3ce763afe048966e9f75ee43d8b6e9b1aa86f8064a008736cc",
    "vout": 1,
    "address": "2Mt2Sx452xkHPCbuenRUvYJsho9Z3v136iS",
    "label": "",
    "redeemScript": "0014af7384a5bf18c451c238813aa7d87d11986ba859",
    "scriptPubKey": "a914088d89bcb985efa0e89749d7c2da2294b97a8ca087",
    "amount": 0.00003000,
    "confirmations": 14,
    "spendable": true,
    "solvable": true,
    "desc": "sh(wpkh([6b9fc73d/0'/0'/10']03a1bd66bff8b0dbf879026e807a5f9b4543625cfadf167db3ca68ad7fc6406f72))#pwu4jgdt",
    "safe": true
  }
]
```

Bitcoin RPC cli

Check results:

```
./bitcoin-cli.sh getbalance
```

```
Output:
```

```
0.00007000
```

Bitcoin RPC programmatically

You can connect to bitcoind or bitcoin-qt via RPC
and run any API calls similar to CLI.

See details: https://en.bitcoin.it/wiki/API_reference_(JSON-RPC).

Bitcoin RPC Node JS Example

Install Bitcoin Core library:

> npm install bitcoin-core

Example code, fix connection parameters, put in in test.js file:

```
const Client = require('bitcoin-core');
const client = new Client({
  network: 'testnet',
  username: 'bitcoinuser',
  password: 'bitcoinsomepass456',
  port: 8332
});

client.getBlockchainInfo().then((help) => console.log(help));
```

Bitcoin RPC Node JS Example

Run it:

> node test.js

Output:

```
{ chain: 'test',
  blocks: 1723119,
  headers: 1723119,
  bestblockhash:
   '000000000000002668c623ed8c955b43196f65838e79831cd12e411e5e9e1bd2',
  difficulty: '13359759.54215726',
  mediantime: 1588763240,
  verificationprogress: '0.9999976636836265',
  initialblockdownload: false,
  chainwork:
   '0000000000000000000000000000000000000000000015980646e14ff46555f',
  size_on_disk: 480776805,
  pruned: true,
  pruneheight: 1668222,
  automatic_pruning: true,
  prune_target_size: 576716800,
  softforks:
   { bip34: { type: 'buried', active: true, height: 21111 },
     bip66: { type: 'buried', active: true, height: 330776 },
     bip65: { type: 'buried', active: true, height: 581885 },
     csv: { type: 'buried', active: true, height: 770112 },
     segwit: { type: 'buried', active: true, height: 834624 } },
  warnings: 'Warning: unknown new rules activated (versionbit 28)' }
```

Homework

1. Install Bitcoin official client.

2. Configure it to sync testing network, use “purge”.

3. Launch cli version and sync Bitcoin testnet.

References

- https://bitcoincore.org/en/segwit_wallet_dev/
- https://bitcoin.org/en/download or https://bitcoincore.org/en/download/
- https://bitcoincore.org/en/doc/0.21.0/