



BITCOIN-CORE P2P机制简介

建立连接

peers.dat & banlist.dat

```
root@i-o0z7gasq:~# ll .bitcoin/
total 12872
drwxr-xr-x 4 root root    4096 Jan  3 20:25 ./
drwx----- 7 root root    4096 Jan  3 20:26 ../
-rw----- 1 root root      37 Dec 23 12:24 banlist.dat
-rw-r--r-- 1 root root     57 Dec 18 16:54 bitcoin.conf
drwx----- 3 root root   69632 Dec 10 12:19 blocks/
drwx----- 2 root root   53248 Jan  3 20:25 chainstate/
-rw----- 1 root root         0 Dec  9 04:35 db.log
-rw----- 1 root root 12964170 Jan  3 20:25 debug.log
-rw----- 1 root root   22001 Jan  3 20:25 fee_estimates.dat
-rw----- 1 root root         0 Dec  9 03:42 .lock
-rw----- 1 root root   21231 Jan  3 20:25 mempool.dat
-rw----- 1 root root   22024 Jan  3 20:25 peers.dat
```

建立连接

dnsseed

src/chainparams.cpp

```
127      // Note that of those with the service bits flag, most only support a subset of possible options
128      vSeeds.emplace_back("seed.bitcoin.sipa.be", true); // Pieter Wuille, only supports x1, x5, x9, and xd
129      vSeeds.emplace_back("dnsseed.bluematt.me", true); // Matt Corallo, only supports x9
130      vSeeds.emplace_back("dnsseed.bitcoin.dashjr.org", false); // Luke Dashjr
131      vSeeds.emplace_back("seed.bitcoinstats.com", true); // Christian Decker, supports x1 - xf
132      vSeeds.emplace_back("seed.bitcoin.jonasschnelli.ch", true); // Jonas Schnelli, only supports x1, x5, x9, and xd
133      vSeeds.emplace_back("seed.btc.petertodd.org", true); // Peter Todd, only supports x1, x5, x9, and xd
```

建立连接

seeds

src/chainparamsseeds.h

```
1  #ifndef BITCOIN_CHAINPARAMSSEEDS_H
2  #define BITCOIN_CHAINPARAMSSEEDS_H
3  /**
4   * List of fixed seed nodes for the bitcoin network
5   * AUTOGENERATED by contrib/seeds/generate-seeds.py
6   *
7   * Each line contains a 16-byte IPv6 address and a port.
8   * IPv4 as well as onion addresses are wrapped inside a IPv6 address accordingly.
9   */
10 static SeedSpec6 pnSeed6_main[] = {
11     {{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xff,0xff,0x02,0xe4,0x46,0xc6}, 8333},
12     {{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xff,0xff,0x04,0x0f,0xb4,0x1d}, 8333},
13     {{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xff,0xff,0x04,0x0f,0xb4,0x1e}, 8333},
14     {{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xff,0xff,0x05,0x02,0x43,0x6e}, 8333},
15     {{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xff,0xff,0x05,0x27,0xe0,0x67}, 8333},
16     {{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xff,0xff,0x05,0x2b,0x7c,0x9a}, 8333},
17     {{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xff,0xff,0x05,0xbd,0xa5,0x66}, 8333},
18     {{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xff,0xff,0x05,0xe2,0x95,0x91}, 8333},
19     {{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xff,0xff,0x05,0xe4,0x07,0x92}, 8333},
20     {{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xff,0xff,0x05,0xe4,0x40,0x47}, 8333},
21     {{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xff,0xff,0x05,0xf9,0x98,0x65}, 8333},
22     {{0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xff,0xff,0x05,0xfe,0x7c,0x37}, 8333},
```

建立连接

命令行/配置文件 设置启动参数

-addnode=ip:port

-connect=ip:port

RPC接口添加节点

addnode ip add/remove

局域网隔离

防止网络分叉

/src/netaddress.cpp

```
bool CNetAddr::IsRoutable() const
{
    return IsValid() && !(IsRFC1918() || IsRFC2544() || IsRFC3927() || IsRFC4862() || IsRFC6598() || IsRFC5737() || (IsRFC41
```

```
101 bool CNetAddr::IsRFC1918() const
102 {
103     return IsIPv4() && (
104         GetByte(3) == 10 ||
105         (GetByte(3) == 192 && GetByte(2) == 168) ||
106         (GetByte(3) == 172 && (GetByte(2) >= 16 && GetByte(2) <= 31)));
107 }
108
109 bool CNetAddr::IsRFC2544() const
110 {
111     return IsIPv4() && GetByte(3) == 198 && (GetByte(2) == 18 || GetByte(2) == 19);
112 }
113
114 bool CNetAddr::IsRFC3927() const
115 {
116     return IsIPv4() && (GetByte(3) == 169 && GetByte(2) == 254);
117 }
118
119 bool CNetAddr::IsRFC6598() const
120 {
121     return IsIPv4() && GetByte(3) == 100 && GetByte(2) >= 64 && GetByte(2) <= 127;
122 }
123
124 bool CNetAddr::IsRFC5737() const
125 {
126     return IsIPv4() && ((GetByte(3) == 192 && GetByte(2) == 0 && GetByte(1) == 2) ||
127         (GetByte(3) == 198 && GetByte(2) == 51 && GetByte(1) == 100) ||
128         (GetByte(3) == 203 && GetByte(2) == 0 && GetByte(1) == 113));
129 }
130
```

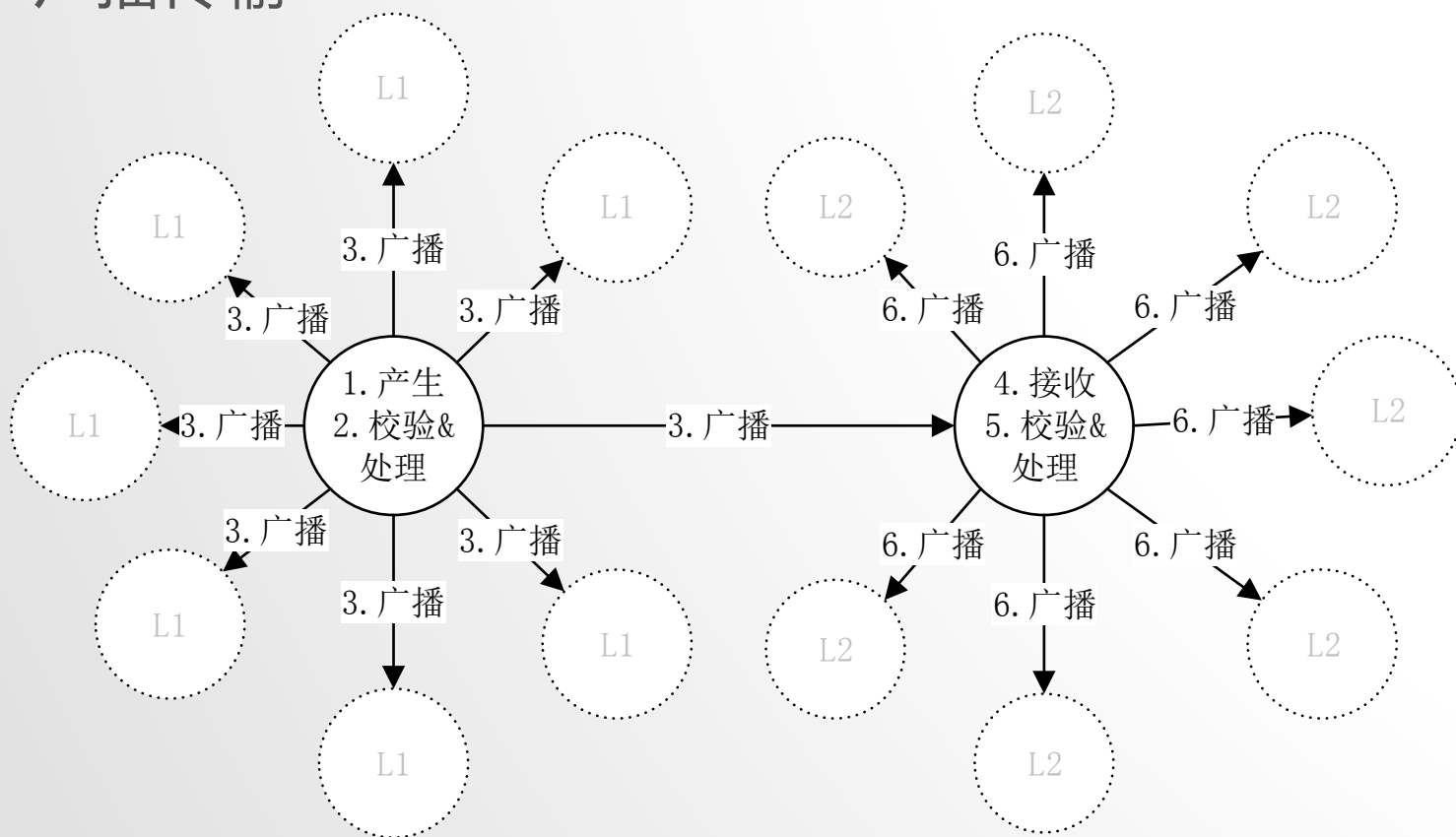
协议版本

协议版本号

```
root@ip-172-31-15-160:~# bitcoin-cli getinfo
{
  "version": 140200,
  "protocolversion": 70015,
  "walletversion": 130000,
  "balance": 22303775531.2500,
  "blocks": 544335,
  "timeoffset": 0,
  "connections": 3,
  "proxy": "",
  "difficulty": 61619.8243902439,
  "testnet": false,
  "keypoololdest": 1514185997,
  "keypoolsize": 101,
  "paytxfee": 0.0000,
  "relayfee": 0.1000,
  "errors": ""
}
```

P2P消息传输

广播传输



1.产生：
sendtoaddress
sendmany
sendrawtransaction
found new block
submitblock

2.校验：
checktransaction
checkblock
checkblockheader

3.处理
mempool & mining
update tip

TX广播

1. 检查交易合法性
2. 广播

BLOCK广播

1. 下载区块头 & 检查
2. 下载区块

1. 下载区块头 & 检查
2. 下载交易 & 拼凑区块