CJDNS support in Bitcoin Core

It is possible to run Bitcoin Core over CJDNS, an encrypted IPv6 network that uses public-key cryptography for address allocation and a distributed hash table for routing.

What is CJDNS?

CJDNS is like a distributed, shared VPN with multiple entry points where every participant can reach any other participant. All participants use addresses from the `fc00::/8` network (reserved IPv6 range). Installation and configuration is done outside of Bitcoin Core, similarly to a VPN (either in the host/OS or on the network router). See https://github.com/cjdelisle/cjdns#readme and https://github.com/hyperboria/docs#hyperboriadocs for more information.

Compared to IPv4/IPv6, CJDNS provides end-to-end encryption and protects nodes from traffic analysis and filtering.

Used with Tor and I2P, CJDNS is a complementary option that can enhance network redundancy and robustness for both the Bitcoin network and individual nodes.

Each network has different characteristics. For instance, Tor is widely used but somewhat centralized. I2P connections have a source address and I2P is slow. CJDNS is fast but does not hide the sender and the recipient from intermediate routers.

Installing CJDNS and finding a peer to connect to the network

To install and set up CJDNS, follow the instructions at https://github.com/cjdelisle/cjdns#how-to-install-cjdns.

You need to initiate an outbound connection to a peer on the CJDNS network before it will work with your Bitcoin Core node. This is described in steps "2. Find a friend" and "3. Connect your node to your friend's node" in the CJDNS documentation.

One quick way to accomplish these two steps is to query for available public peers on Hyperboria by running the following:

```
git clone https://github.com/hyperboria/peers hyperboria-peers
cd hyperboria-peers
./testAvailable.py
```

For each peer, the `./testAvailable.py` script prints the filename of the peer's credentials followed by the ping result.

Choose one or several peers, copy their credentials from their respective files, paste them into the relevant IPv4 or IPv6 "connectTo" JSON object in the

`cjdroute.conf` file you created in step "1. Generate a new configuration file", and save the file.

Launching CJDNS

Typically, CJDNS might be launched from its directory with `sudo ./cjdroute < cjdroute.conf` and it sheds permissions after setting up the TUN interface. You may also launch it as an unprivileged user with some additional setup.

The network connection can be checked by running `./tools/peerStats` from the CJDNS directory.

Run Bitcoin Core with CJDNS

Once you are connected to the CJDNS network, the following Bitcoin Core configuration option makes CJDNS peers automatically reachable:

`-cjdnsreachable`

When enabled, this option tells Bitcoin Core that it is running in an environment where a connection to an `fc00::/8` address will be to the CJDNS network instead of to an RFC4193 IPv6 local network. This helps Bitcoin Core perform better address management: - Your node can consider incoming `fc00::/8` connections to be from the CJDNS network rather than from an IPv6 private one. - If one of your node's local addresses is `fc00::/8`, then it can choose to gossip that address to peers.

Additional configuration options related to CJDNS

`-onlynet=cjdns`

Make automatic outbound connections only to CJDNS addresses. Inbound and manual connections are not affected by this option. It can be specified multiple times to allow multiple networks, e.g. onlynet=cjdns, onlynet=i2p, onlynet=onion.

CJDNS support was added to Bitcoin Core in version 23.0 and there may be fewer CJDNS peers than Tor or IP ones. You can use `bitcoin-cli -addrinfo` to see the number of CJDNS addresses known to your node.

In general, a node can be run with both an onion service and CJDNS (or any/all of IPv4/IPv6/onion/I2P/CJDNS), which can provide a potential fallback if one of the networks has issues. There are a number of ways to configure this; see doc/tor.md for details.

CJDNS-related information in Bitcoin Core

There are several ways to see your CJDNS address in Bitcoin Core: - in the "Local addresses" output of CLI `-netinfo` - in the "localaddresses" output of

RPC `getnetworkinfo`

To see which CJDNS peers your node is connected to, use `bitcoin-cli -netinfo 4` or the `getpeerinfo` RPC (i.e. `bitcoin-cli getpeerinfo`).

To see which CJDNS addresses your node knows, use the `getnodeaddresses 0 cjdns` RPC.