Ethereum underlying P2P Network

Amira BOUGUERA

ConsenSys

January 10, 2020

Your expert

Amira Bouguera

Cryptographer

Security & blockchain
engineer
At Consensys

My professional path

School:

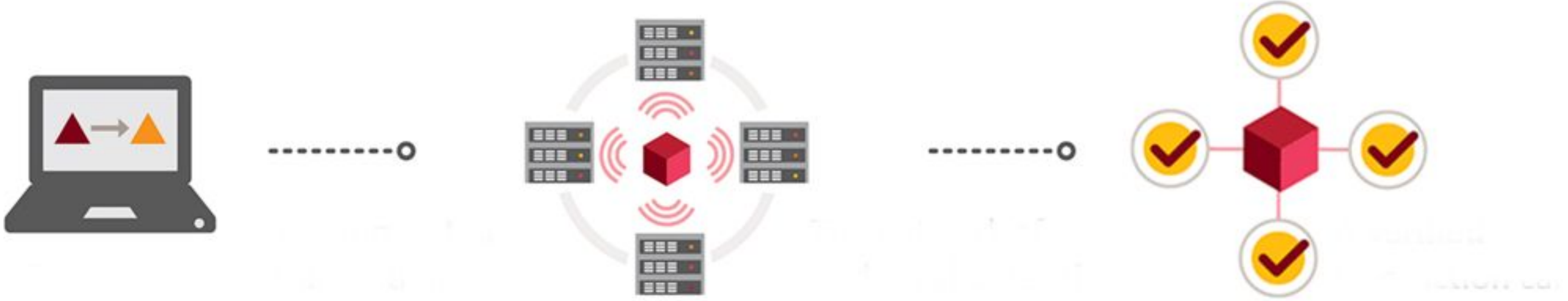
- **2010- 2013** : bachelor in mathematics and computer science and specialised in pure mathematics.
- **2013-2015** : master in applied mathematics (Algebra, Geometry and Cryptography).
- **2015-2017**: master in cyber security and cryptography.

Professional:

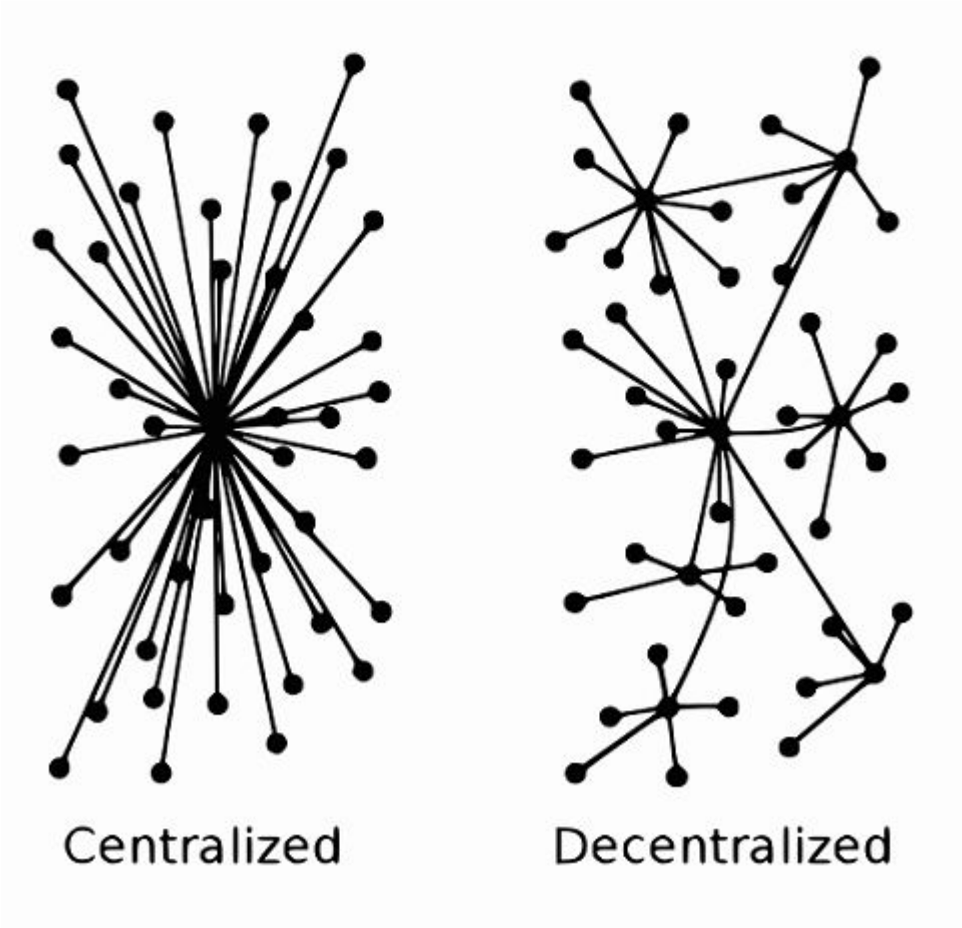
- **2016-2017**: Secure outsourced Cloud Computing using linear Algebra (LJK Laboratory Grenoble).
- **Jan 2017- Juin 2017:** Blockchain developer at Tessi Labs.
- **July 2017- December 2017**: Blockchain security Engineer at Stratumn.
- **Jan 2018- today**: Cryptographer at Consensys.

Blockchain

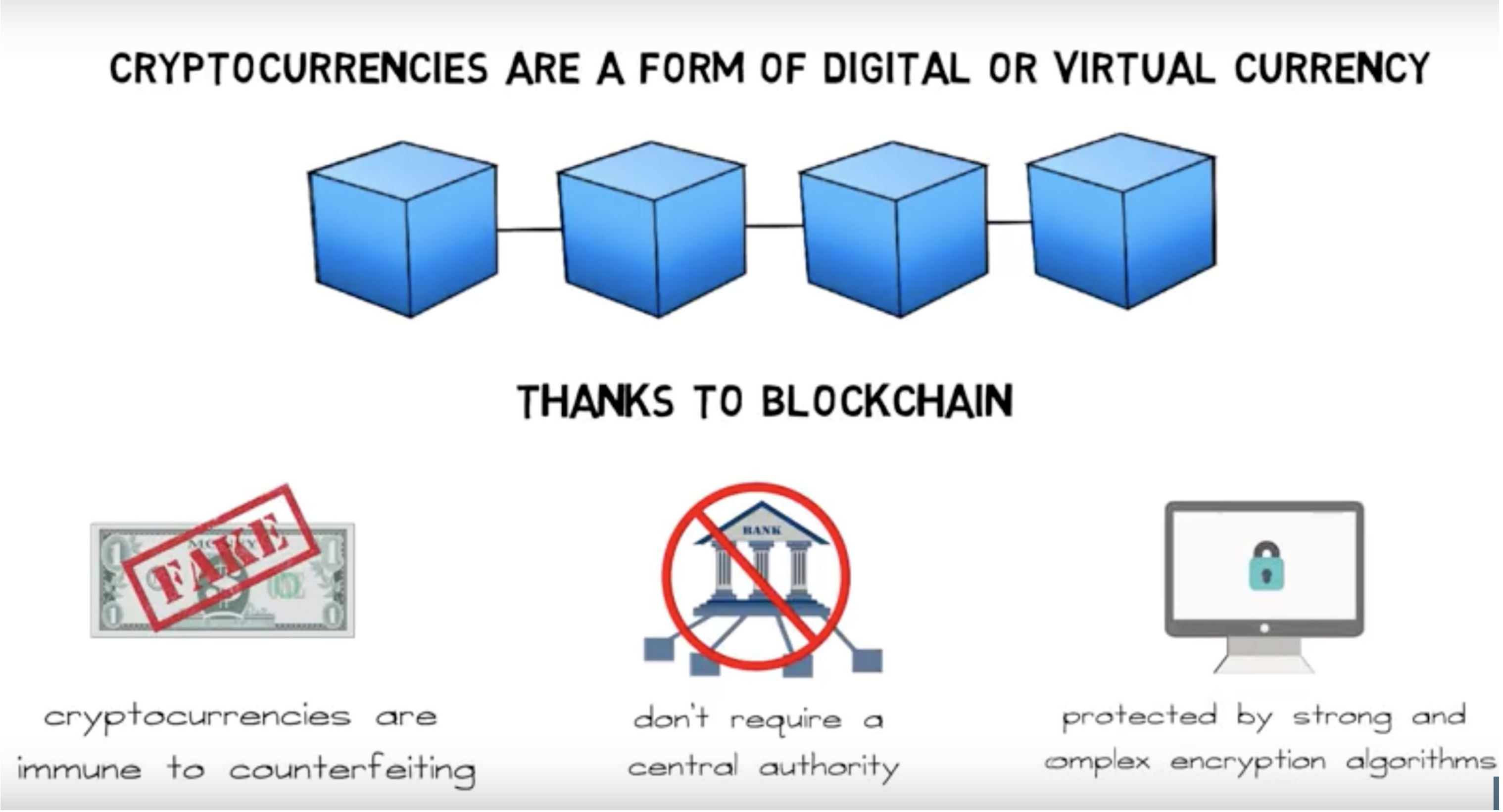
A blockchain is a type of distributed ledger, comprised of unchangeable, digitally recorded data in packages called blocks shared between nodes without any central point of control.



The difference between a centralized and a decentralized architecture



What makes blockchain technology such an advance?



CRYPTOCURRENCIES ARE A FORM OF DIGITAL OR VIRTUAL CURRENCY

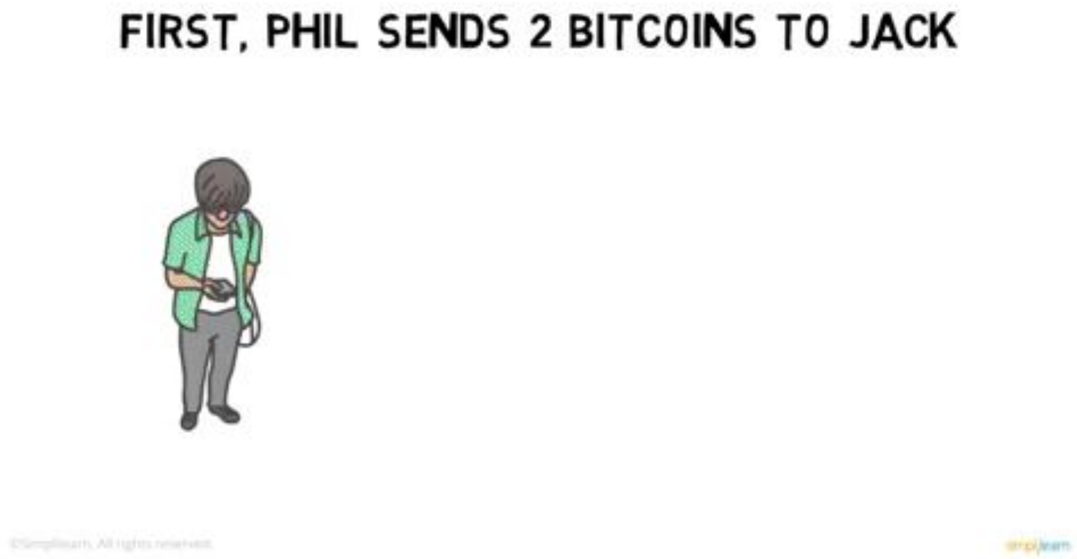
THANKS TO BLOCKCHAIN

cryptocurrencies are immune to counterfeiting

don't require a central authority

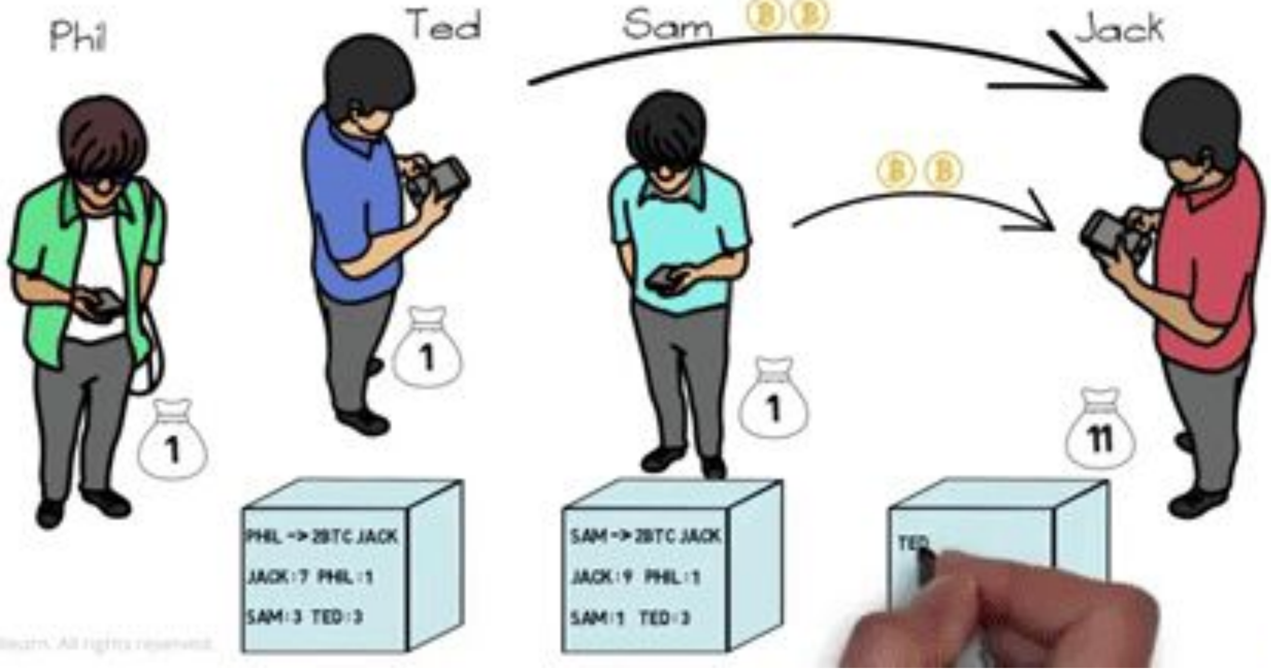
protected by strong and complex encryption algorithms

How does it work?



FIRST, PHIL SENDS 2 BITCOINS TO JACK

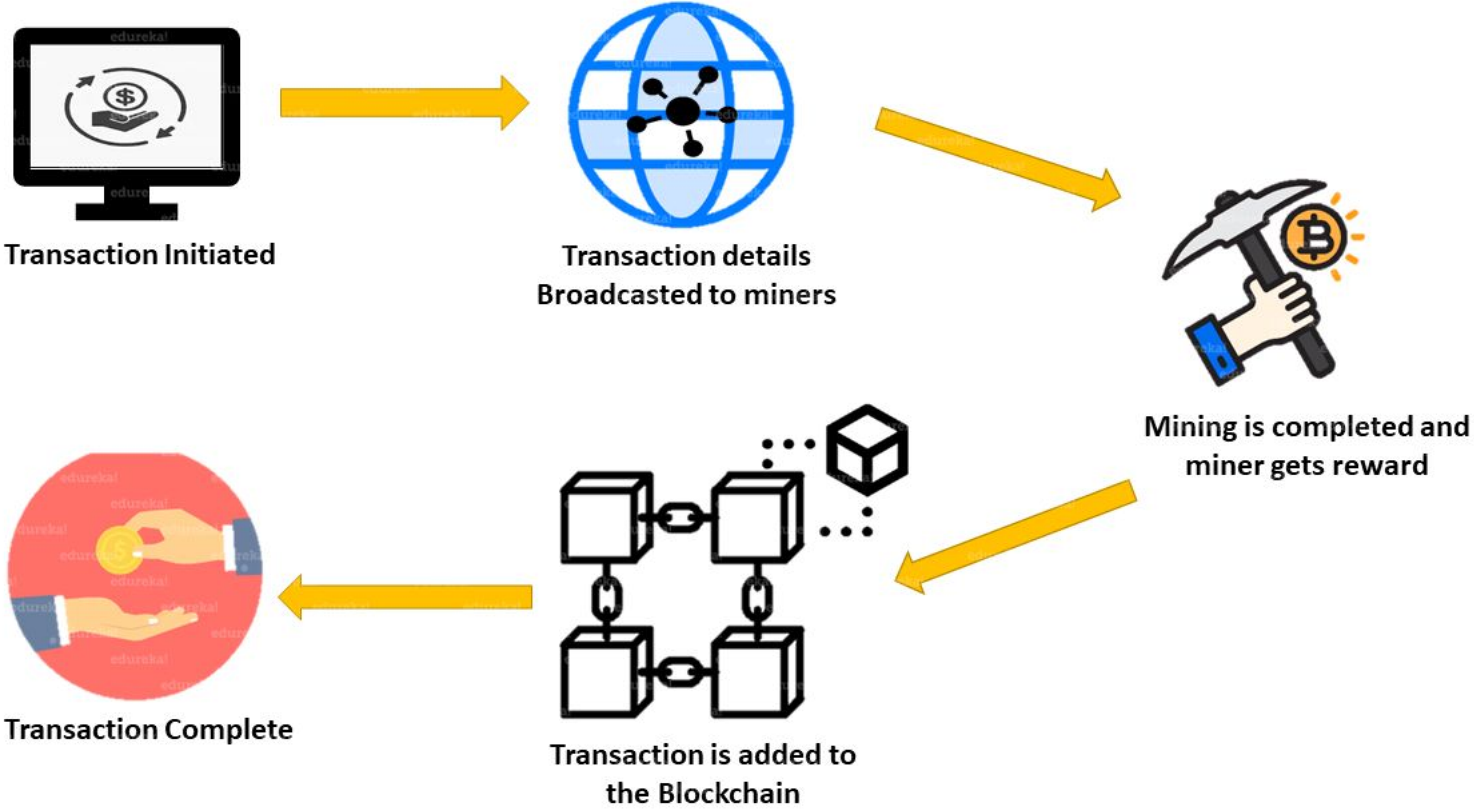
THIS RECORD ALSO HOLDS THE NUMBER OF BITCOINS EACH OF THE FRIENDS OWN



PHIL -> 2BTC JACK
JACK : 7 PHIL : 1
SAM : 3 TED : 3

SAM -> 2BTC JACK
JACK : 9 PHIL : 1
SAM : 1 TED : 3

TED



Transaction Initiated

Transaction details Broadcasted to miners

Mining is completed and miner gets reward

Transaction is added to the Blockchain

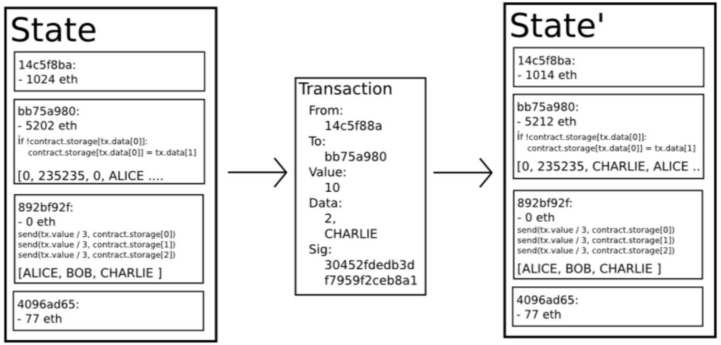
Transaction Complete

What is Ethereum?

Ethereum is a decentralized peer-to-peer network of untrusted nodes that execute transactions, and share and agree on the same view of data — the world state. It has the ability to perform any complex calculation using a fully trustless smart contract platform to create and work on decentralized applications.

Transaction-based state machine

Ethereum can be thought of as a transactional state machine (Transaction-based state machine) which begins with a state of genesis and ends with the current state. The state may include information such as account balances, reputations and trust agreements.



State

14c5f8ba:
- 1024 eth

bb75a980:
- 5202 eth

```
if !contract.storage[tx.data[0]]:
    contract.storage[tx.data[0]] = tx.data[1]
```

[0, 235235, 0, ALICE ....

892bf92f:
- 0 eth

```
send(tx.value / 3, contract.storage[0])
send(tx.value / 3, contract.storage[1])
send(tx.value / 3, contract.storage[2])
```

[ALICE, BOB, CHARLIE]

4096ad65:
- 77 eth

Transaction
From:
14c5f88a
To:
bb75a980
Value:
10
Data:
2,
CHARLIE
Sig:
30452fdedb3d
f7959f2ceb8a1

State'

14c5f8ba:
- 1014 eth

bb75a980:
- 5212 eth

```
if !contract.storage[tx.data[0]]:
    contract.storage[tx.data[0]] = tx.data[1]
```

[0, 235235, CHARLIE, ALICE ..

892bf92f:
- 0 eth

```
send(tx.value / 3, contract.storage[0])
send(tx.value / 3, contract.storage[1])
send(tx.value / 3, contract.storage[2])
```

[ALICE, BOB, CHARLIE]

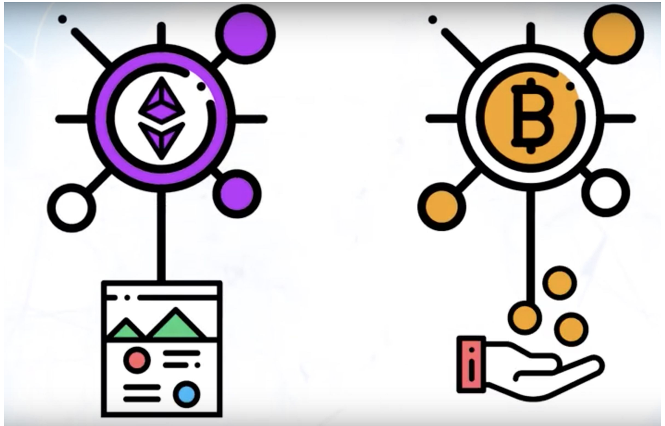
4096ad65:
- 77 eth

Bitcoin vs Ethereum

Bitcoin offers a particular application of blockchain technology, a peer to peer electronic money system allowing online payment by Bitcoin.

Ethereum focuses on running the programming code of any decentralized application.

Like Bitcoin, Ethereum is a distributed public blockchain network. While there are significant technical differences between the two, the most important distinction to note is that Bitcoin and Ethereum differ considerably in their purpose and capabilities.

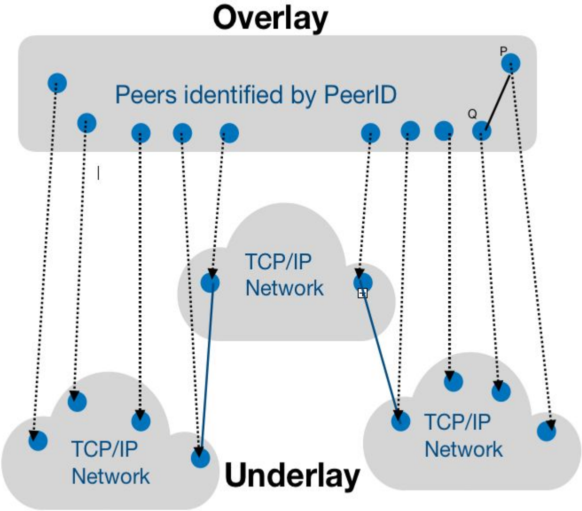


Ethereum p2p network

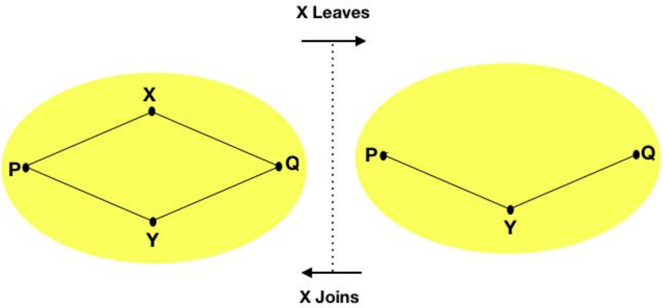
How does it work?

But first, **What is a P2P Network?**

- A Peer-to-Peer (P2P) network is an **overlay network** — that is, it's built on top of the public Internet.
- Each peer p has a unique identification number pID.
- The overlay depends on intermediate peers to forward messages to the correct regions of the overlay.



- A link **(p,q)** in E means that p has a direct path to send a message to q.
- The graph structure provides multiple paths between every pair of peers, and contributes to **resilience** by enabling connectedness despite peer node changes.



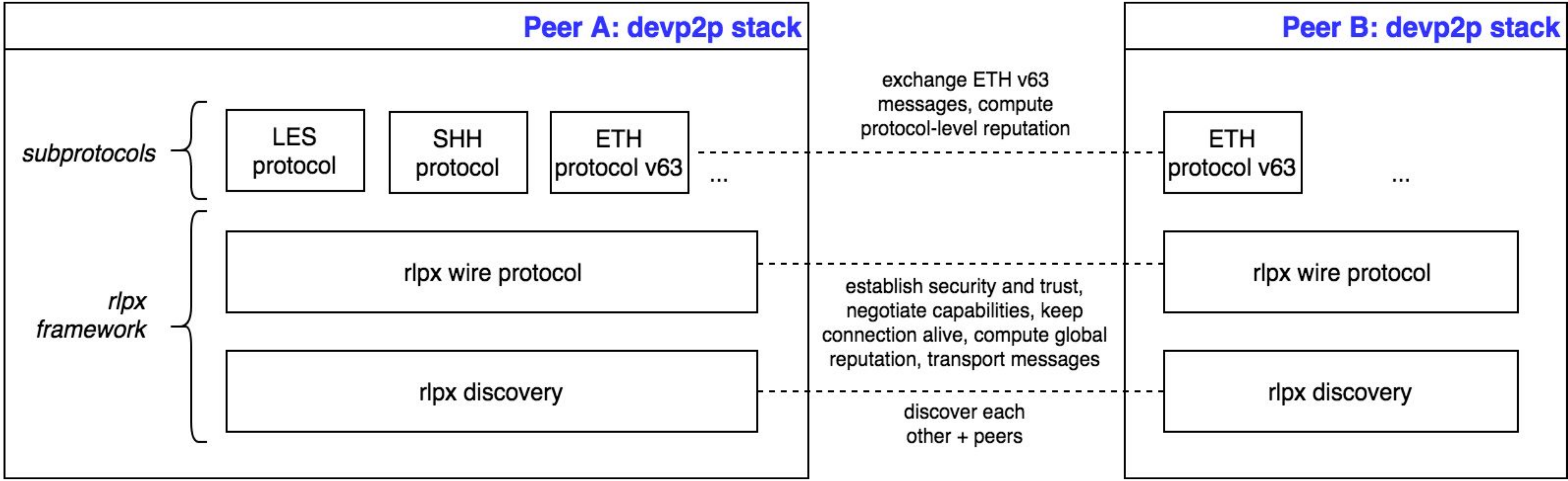
How does Ethereum’s P2P network work?

In Ethereum 1.0, nodes speak with each other through a framework of network protocols called ĐΞVp2p (devp2p), in order to discover peers, gossip about transactions, broadcast blocks, and share their status.

ĐΞVp2p is a layered stack, composed of:

1. **The RLPx framework:** responsible for the *plumbing* of communications. Split in two protocols: discovery and wire protocol.
2. **Pluggable user-land subprotocols,** such as ETHv63, SHHv1, LESv1, etc.

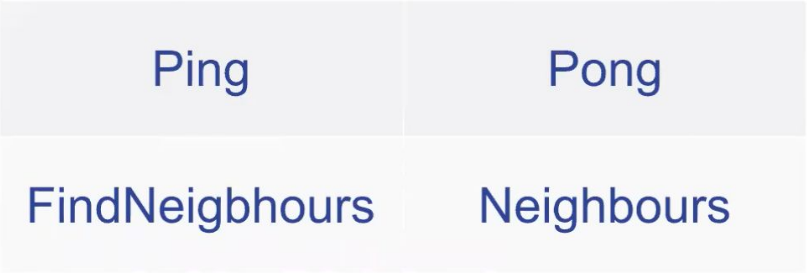
A schematic view of the devp2p network stack in Ethereum clients


Peer A: devp2p stack
subprotocols
LES protocol
SHH protocol
ETH protocol v63
...
rlpx framework
rlpx wire protocol
rlpx discovery
exchange ETH v63 messages, compute protocol-level reputation
establish security and trust, negotiate capabilities, keep connection alive, compute global reputation, transport messages
discover each other + peers
Peer B: devp2p stack
ETH protocol v63
...
rlpx wire protocol
rlpx discovery

Bonding

- Node must have successfully respond to ping/pong pair before neighbor messages work.
- After a ping/pong pair, the node may get added to the routing table
- If a node doesn’t get added to the routing table, it won’t respond to neighbor messages.

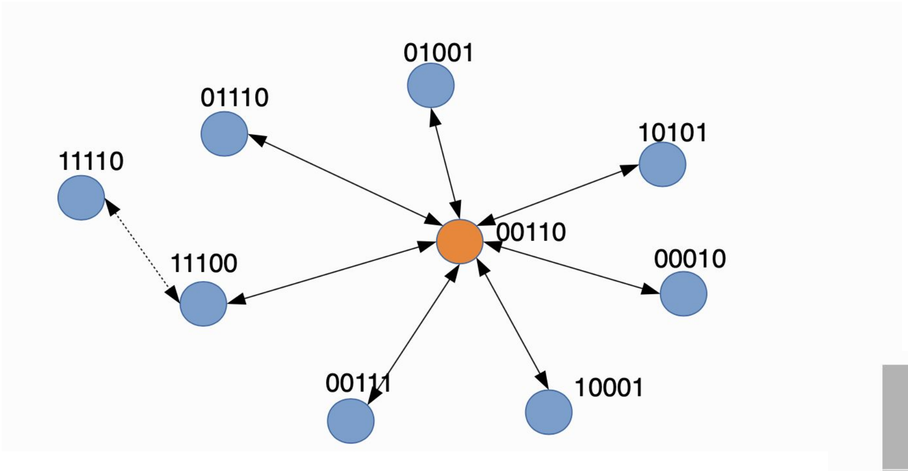
Two sets of paired messages:



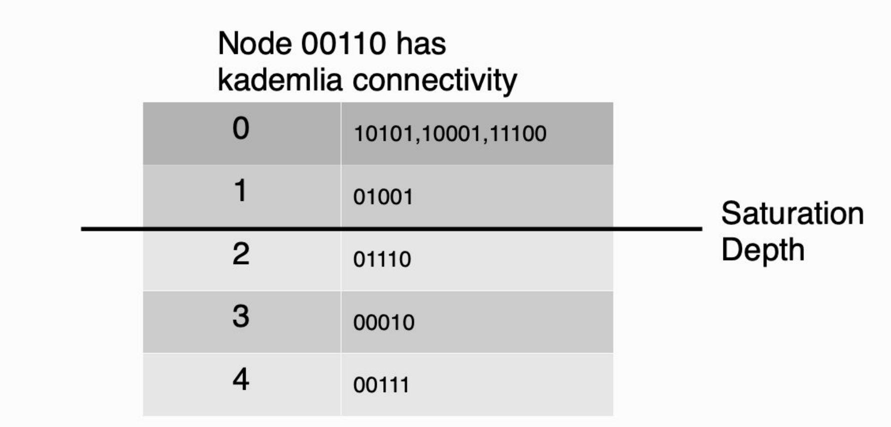
Ping	Pong
FindNeigbhours	Neighbours

Kademlia DHT Protocol

RLPx Handles peer discovery via a Kademlia DHT-based UDP protocol. It bootstraps from a set of seed nodes and performs iterative lookups on the network, filling up a k-bucket peer routing table where nodes take up positions based on their the XOR distance metric.



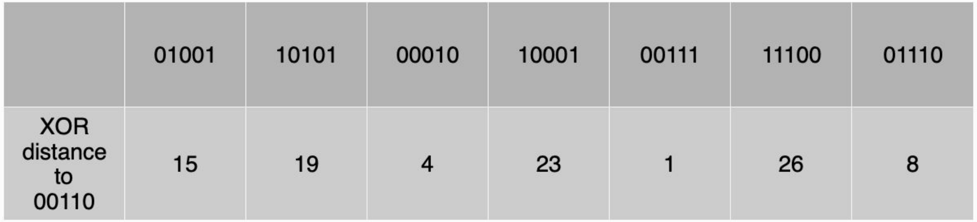
The **higher** the proximity number,
the **closer** to each other the two addresses are



Node 00110 has
kademlia connectivity

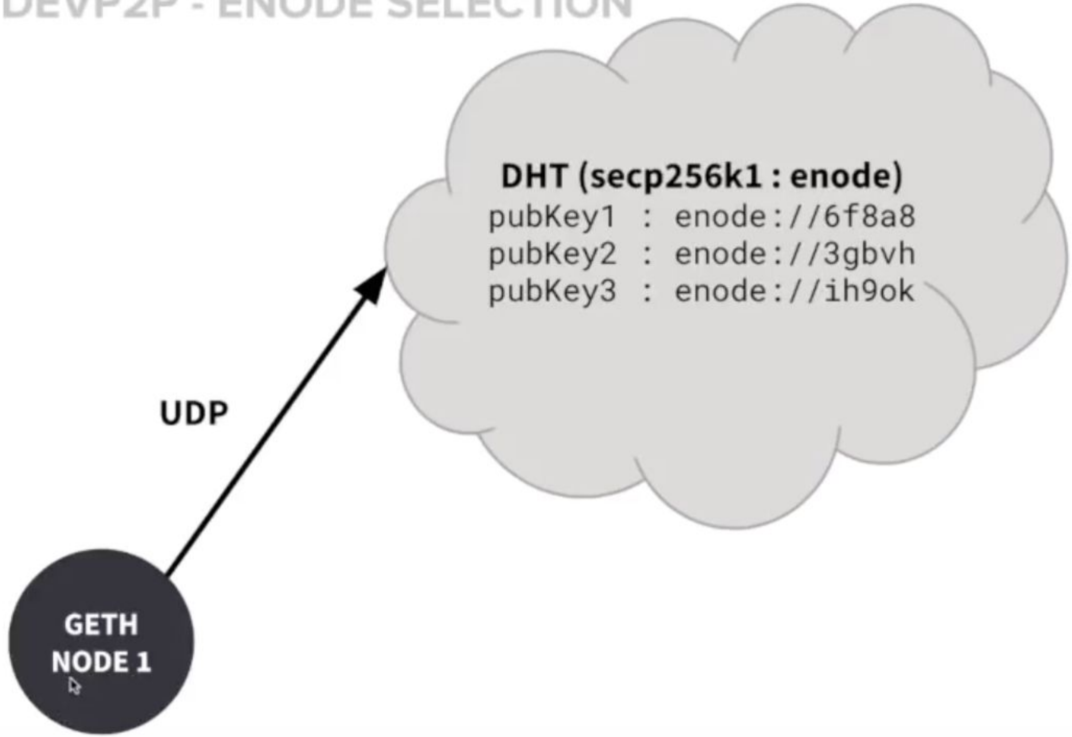
0	10101,10001,11100
1	01001
2	01110
3	00010
4	00111

Saturation Depth



	01001	10101	00010	10001	00111	11100	01110
XOR distance to 00110	15	19	4	23	1	26	8

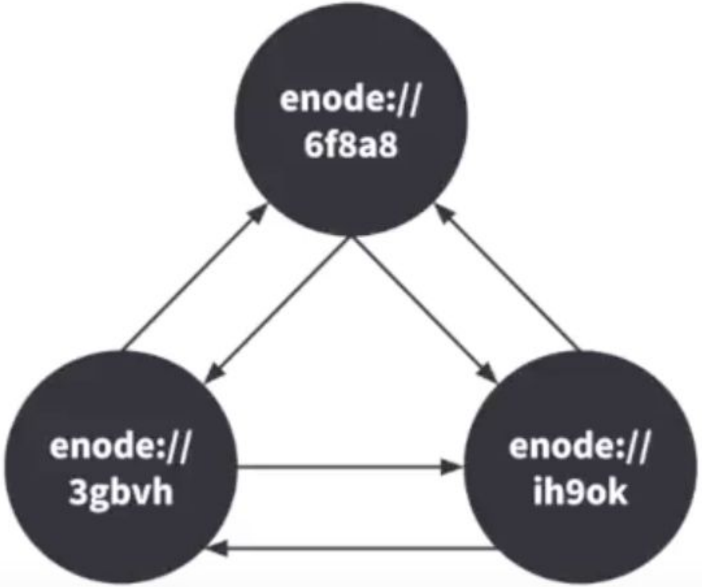
DEVP2P - ENODE SELECTION



DHT (secp256k1 : enode)

```
pubKey1 : enode://6f8a8
pubKey2 : enode://3gbvh
pubKey3 : enode://ih9ok
```

P2P Connected Nodes



RLPx wire protocol

- Establishes TCP-based encrypted and authenticated sessions with peers (exchange pubkeys and create a handshake using shared secret).
- Manages their lifecycle
- Performs keepalives (PING-PONG) to prevent DDOS.
- Agrees on mutually supported capabilities (subprotocols).

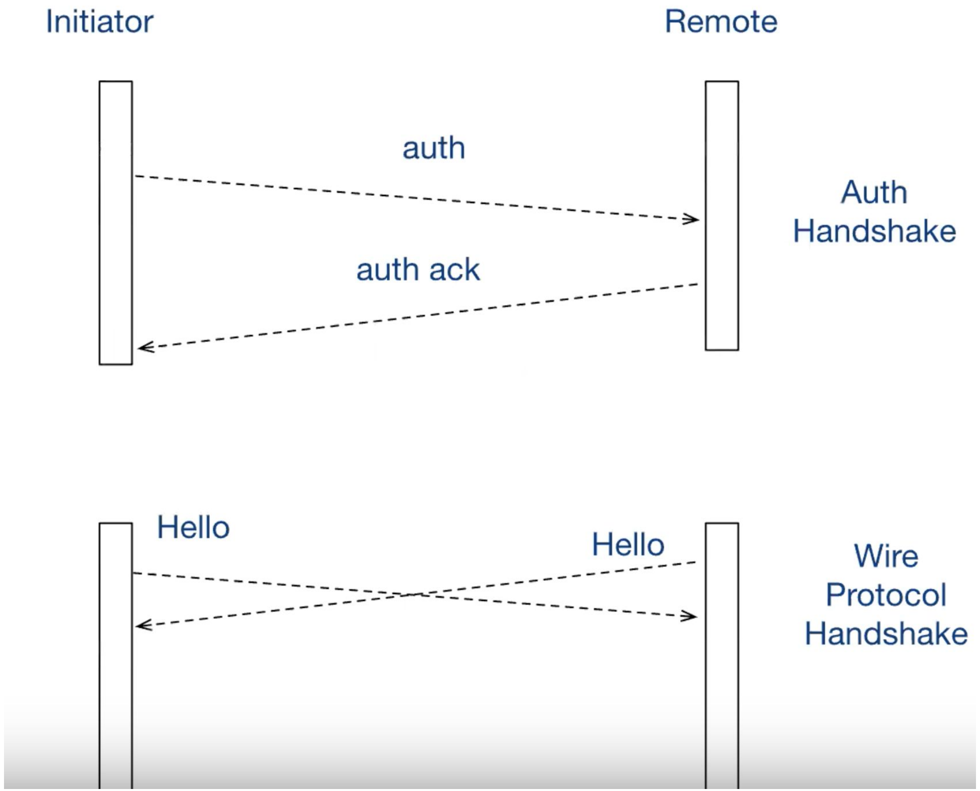
Session initialisation

AuthAck:

- Generates AuthAck from remote pubkey and nonce.
- Sends AuthAck

Handshake Protocol:

- Each node derives shared-secret, aes-secret, mac-secret,ingress-mac, egress-mac.
- Initiates wire protocol and sends Hello
- Authenticates protocol handshake (on receipt from other node).



Subprotocol negotiation and initiation

Subprotocols

Self-contained protocols that define a set of messages transported over the RLPx connections set up above.

For example, Ethereum data such as blocks, headers, transactions, etc. are propagated through the **ETH protocol (eth/63)**, used by the sync loop of Ethereum clients to synchronise the chain.

The Light Ethereum Subprotocol (LES) is the protocol used by “light” clients, which only download block headers as they appear and fetch other parts of the blockchain on-demand.

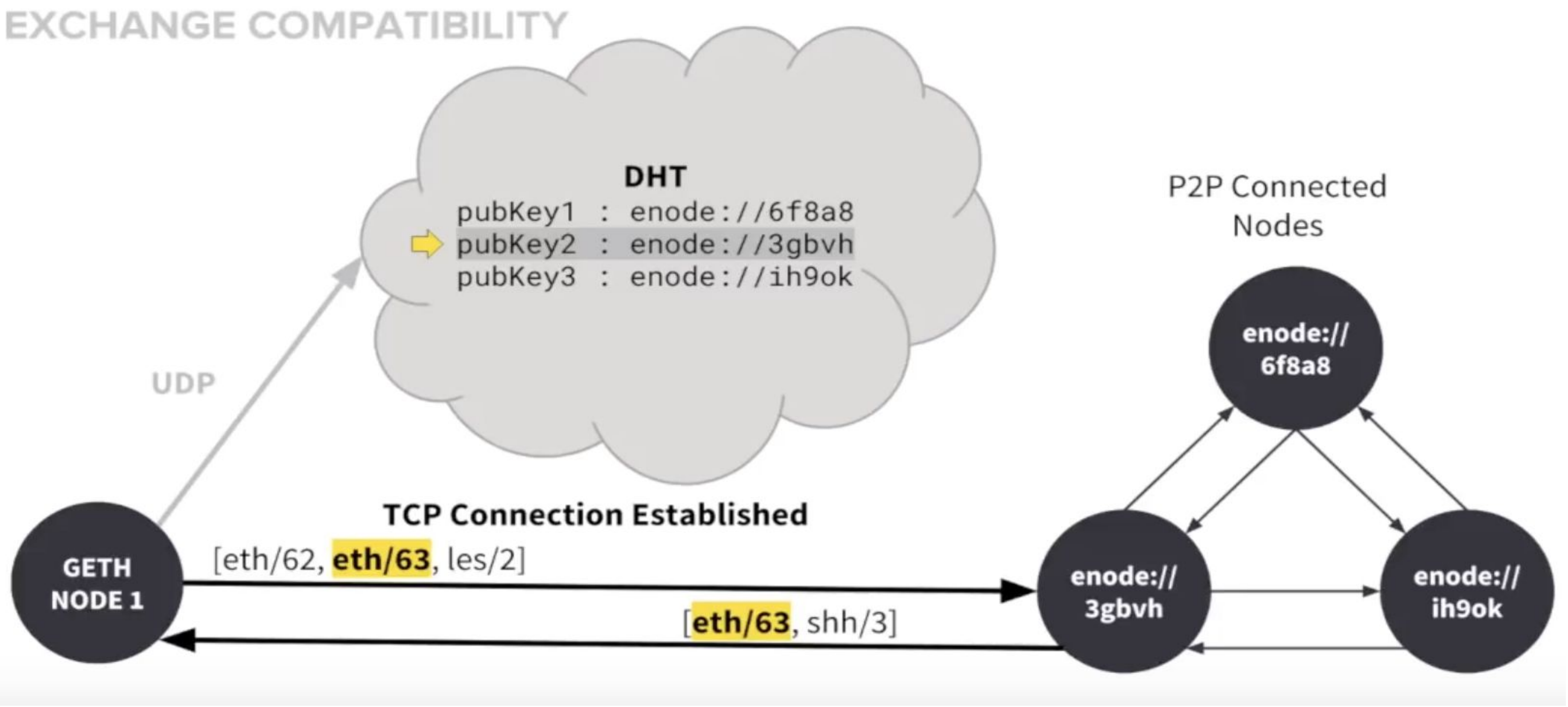
Other subprotocols include **Whisper (SHH)**, **Swarm (BZZ)** and even client-specific ones like **Parity’s Warp Sync (PAR)**.

Subprotocol negotiation

- Collect shared capabilities between initiator and remote.
- Pick highest version for each capability.
- Handshaking initiated for the subprotocol

  - Using status messages.

Nodes advertise supported subprotocols when they handshake (as tuples of [ID, version]), and agree on which ones will be used during the RLPx session.



Challenges facing devp2p

Discovery v4 does not attempt to differentiate between node *capabilities*. So, it is possible to 'accidentally' discover peers in other networks, such as an Ethereum node finding an Ethereum Classic node for example.

It is not until the more resource intensive devp2p/rlpx higher level handshake is established that the peer gets to learn that it is attempting to add an invalid peer to its own set of neighbours.

This common base layer **introduces noise**, which is shared across the networks.

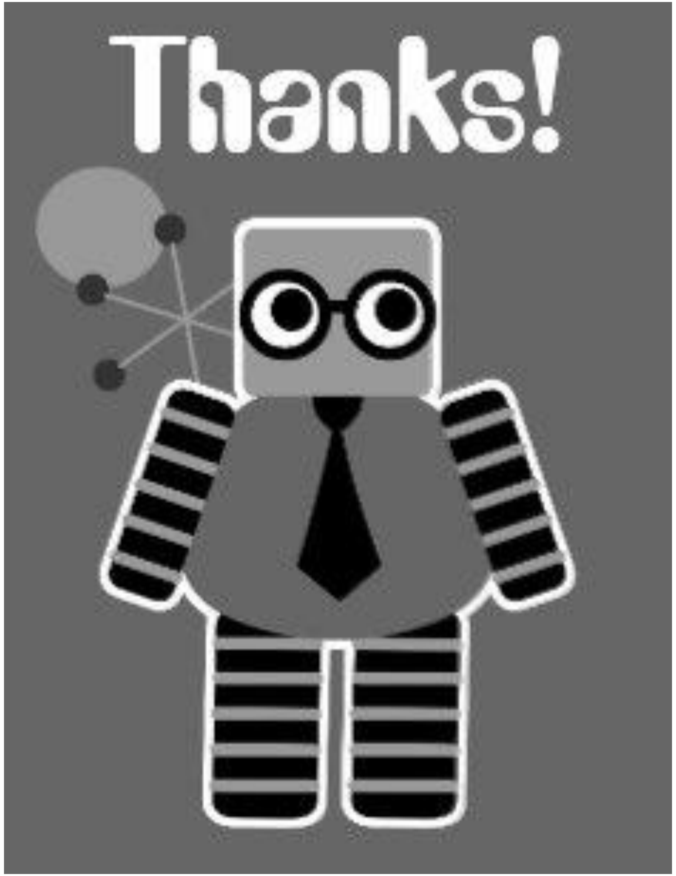
The fact that the peer *capability* is not known until after the devp2p/rlpx handshake is established also complicates the search for less common nodes, such as LES (light Ethereum) protocol serving peers.

Ethereum 2.0 improvements

The new improvements to *discovery* introduce mechanisms that help with the above issue.

The new "*discovery v5*" will be used for Ethereum 2.0 and will provide:

- Being independent of the clock
- Making traffic amplification prevention less weird
- Relaying more node metadata
- Indexing nodes by their capabilities
- Obfuscating discovery traffic


Thanks!