Distributed Hash Table edition

@gpestana

What's your superpower: DHT

What's your superpower: DHT

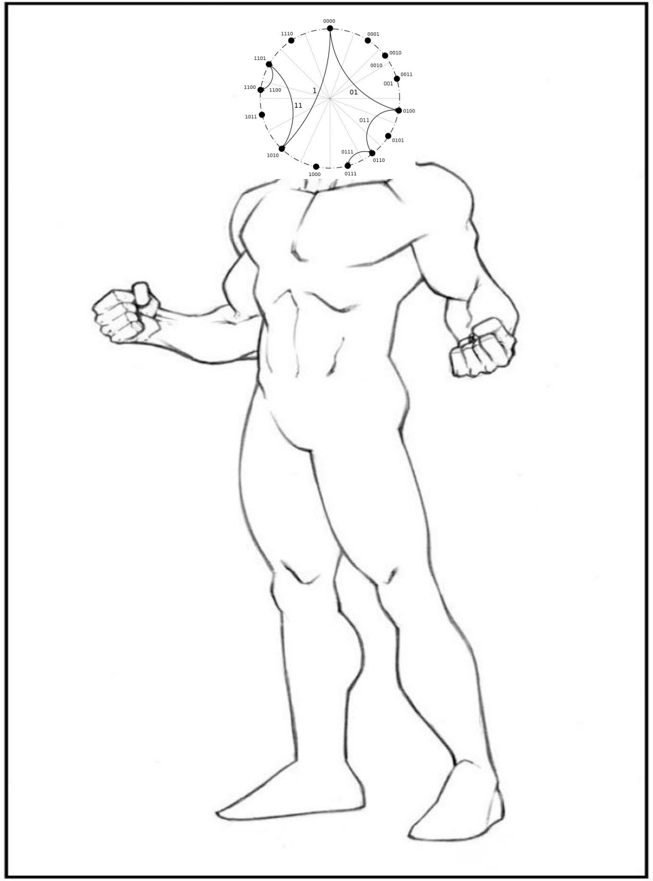
What's your superpower: DHT

What's your superpower: DHT

CREATOR(S) Dr X.

ROOM

DATE TOPSECRET



Hero Name: DHT

Real Name: Distributed Hash Table

Work: Logistics at IPFS, Freenet, and other projects

Place of birth: Interwebs

History

In the early 2000s started to be used and researched by many projects to decentralize the web. It's also used in distributed systems (e.g. riak).

Global Cause:

Fight against the centralization tyranny by making It easy for nodes to organize themselves, store and lookup state

Super powers

Efficient storage and lookup of data where all nodes are symmetric. No need for external coordinators.

Height: n

Weight: log(n)

Eyes: n/a

Hair: n/a

http://christianmale.blogspot.com/

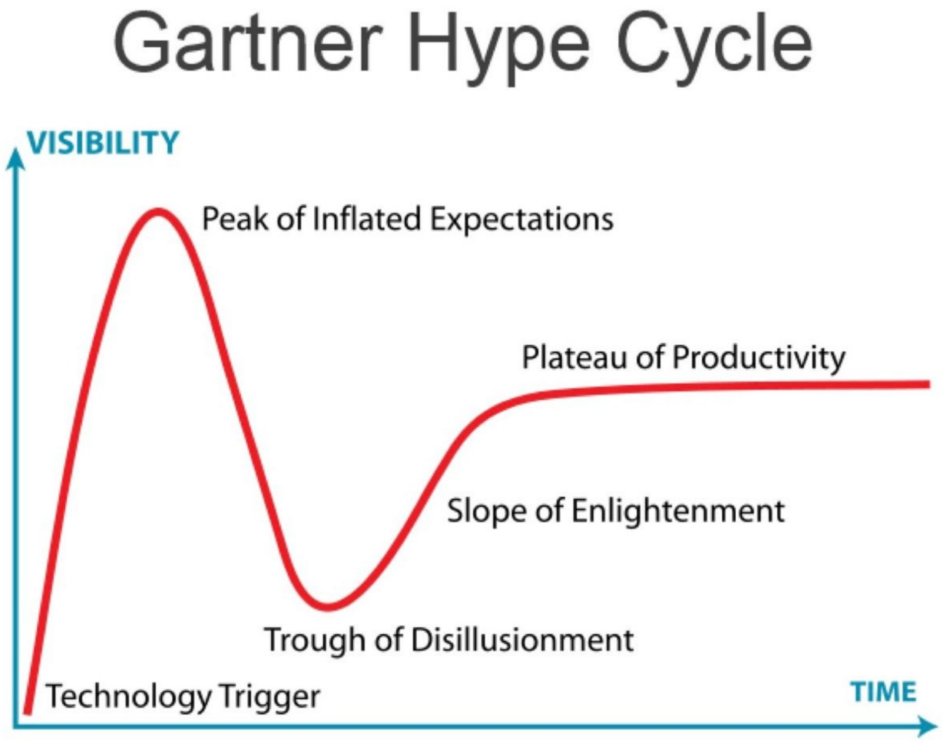
What's your superpower: DHT

History

Early 2000s, lots of research and applications on decentralization and DHTs

Napster, Freenet, Seti@Home, Bittorrent, Coral, IPFS, ...

History


Gartner Hype Cycle
VISIBILITY
Peak of Inflated Expectations
Plateau of Productivity
Slope of Enlightenment
Trough of Disillusionment
Technology Trigger
TIME

What's your superpower: DHT

What's your superpower: DHT

Decentralized P2P networks

Distributed systems in which all components are symmetric

No centralized control

Self organizing system

What's your superpower: DHT

Decentralized P2P networks

Resilience (nodes take roles of others organically)

Scalability/elasticity (new nodes use and **add** resources)

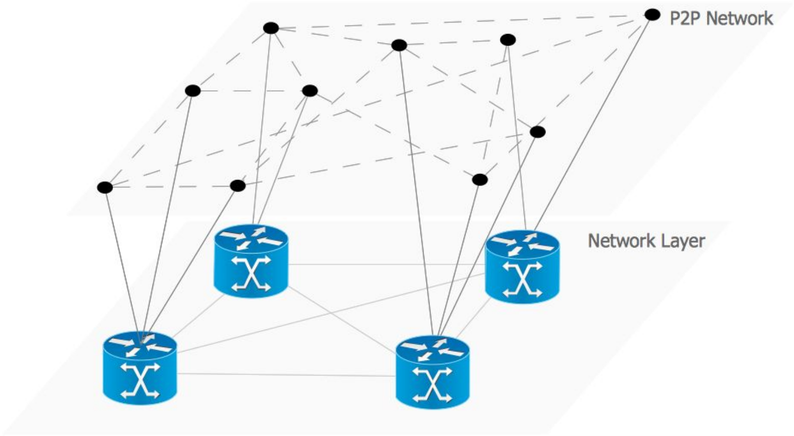
*(distribute the eggs in * baskets)*

Organic growth (barrier for deployment is **low**)

What's your superpower: DHT

Network overlays

Hosts connected through a topology which does not necessarily map to their physical connections



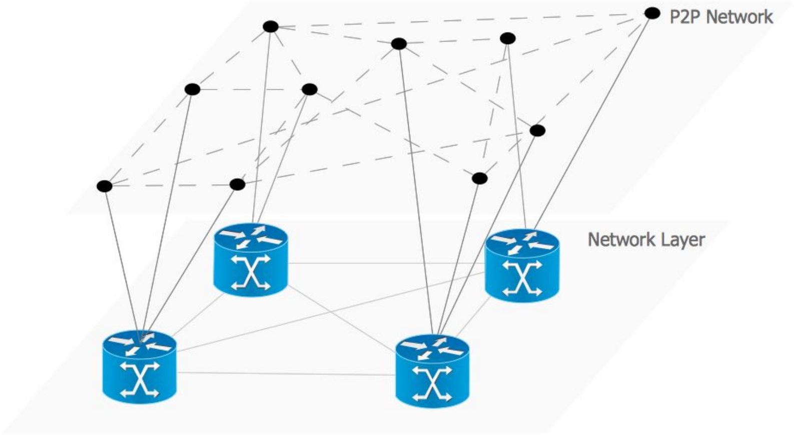
What's your superpower: DHT

Network overlays

Hosts maintain IPs of hosts they maintain connection

Structured topology

Unstructured topology



What's your superpower: DHT

💪 ⚡ ⚡ ⚡

What's your superpower: DHT

MAINTAIN AND LOOKUP STATE OBJECTS IN P2P NETWORK

Store data in hosts which are part of the network

Replicate state is important: nodes come and go

Also important to consider load balancing

No **coordinator**

What's your superpower: DHT

How do I get state objects from the network?

Overlay network needs to keep some sort of **index** and **primitives** for:

- storing state objects
- performing lookups

Network symmetry: there are no special nodes (e.g. leader)

What's your superpower: DHT

Data model abstraction of distributed hash tables

- storing state objects: `put(key, data)`
- performing lookups: `get(key)`

How to make this happen in a decentralized, leaderless network?

What's your superpower: DHT

DHT needs to:

1) Decide which nodes store which data
2) Allow peers to lookup data in the network
3) `put(key, value)` and `get(value)` primitives

Availability, replication, load balancing, etc.

What's your superpower: DHT

DHT deciding which nodes store which data

What's your superpower: DHT

Consistent Hashing

Mapping keyspace to nodes and limit re-hashing when nodes join/leave network

Each **node** in the network is dynamically associated with an ID

Each **state object** has a key (K:V) which is part of a large subset (keyspace)

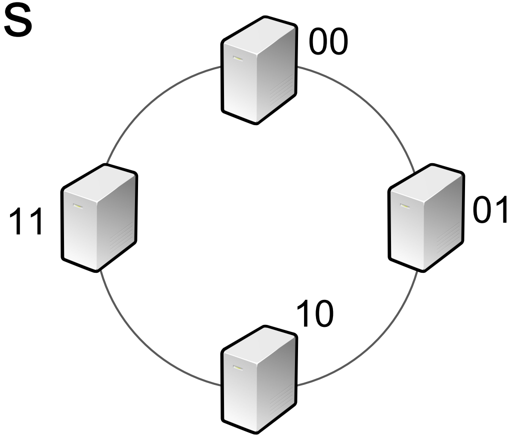
What's your superpower: DHT

Partitioning keyspace

Give addresses to nodes

2 bits key space: 4 positions for the nodes

```
hash_function(MAC_addr) → {00, 01, 10, 11}
```



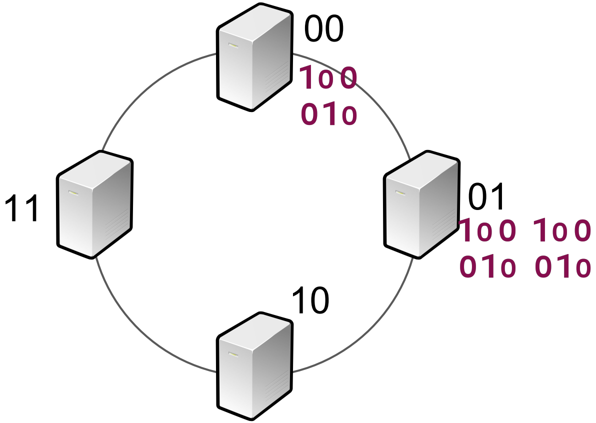
What's your superpower: DHT

Partitioning keyspace

```
hash_function(MAC_addr) → {00, 01, 10, 11}

hash_function(data_binary) → {00, 01, 10, 11}
```

Node and data addresses overlap!



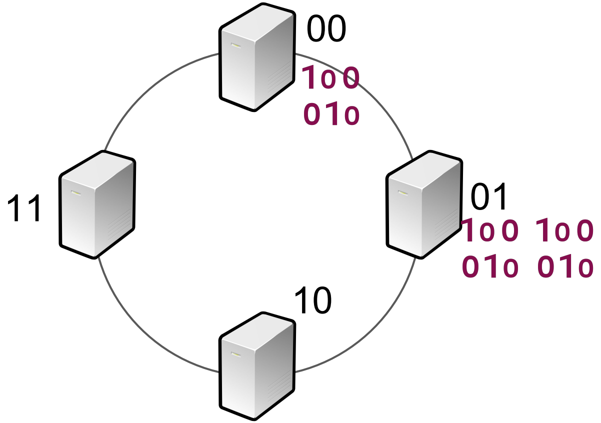
What's your superpower: DHT

Partitioning keyspace

No need for coordination

Load balanced by design

Rehashing when churn is controlled and localized



What's your superpower: DHT

DHT looking for data stored in the network

What's your superpower: DHT

Key-based routing (KBR)

Every node implements a primitive that routes lookup requests closest to the node storing the key

SOLUTION 1 Every node keeps a complete view of the network in their **routing table**

What's your superpower: DHT

Key-based routing (KBR)

Every node implements a primitive that routes lookup requests closest to the node storing the key

SOLUTION 1 Every node keeps a complete view of the network in their **routing table**

What's your superpower: DHT

Key-based routing (KBR)

Every node implements a primitive that routes lookup requests closest to the node storing the key

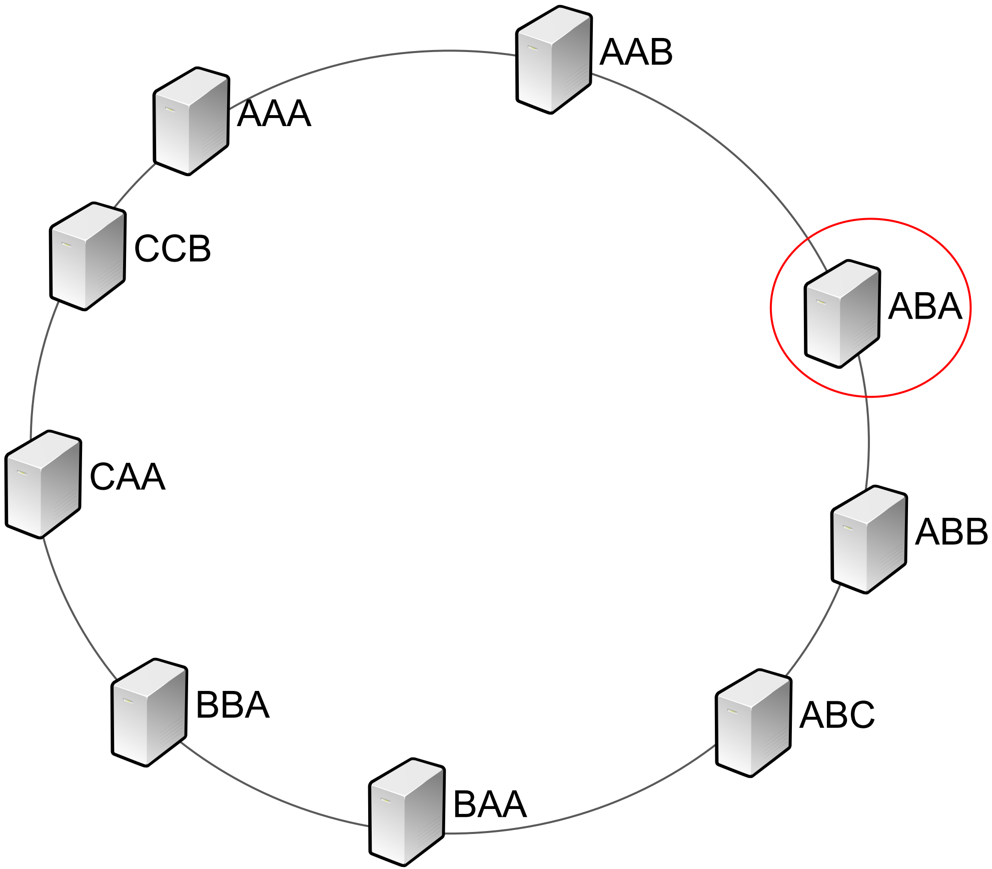
SOLUTION 2 Every node keeps a **reduced** view of the network in their **routing table** (finger table)

`log(n)`	for lookup hops
`const(n)`	for peers in local routing table

What's your superpower: DHT

DHT performing efficient routing

What's your superpower: DHT

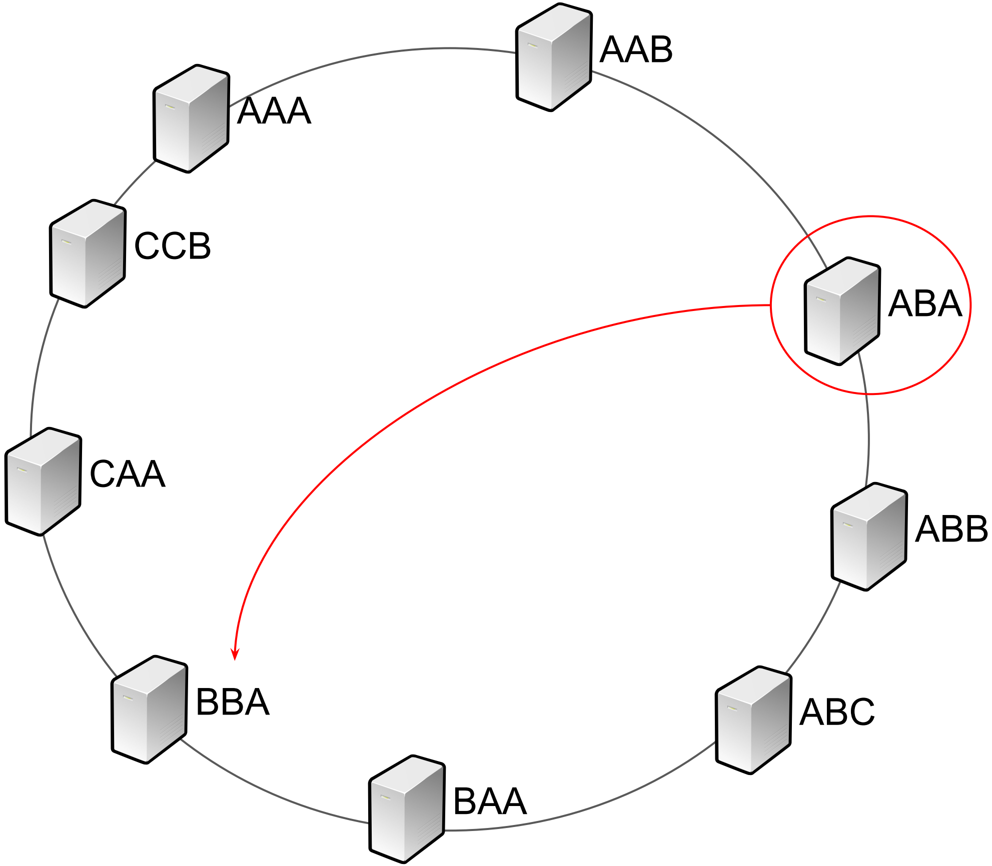


lookup(CCC)

RT:

ABB	*IP*
AAB	*IP*
BBA	*IP*

What's your superpower: DHT

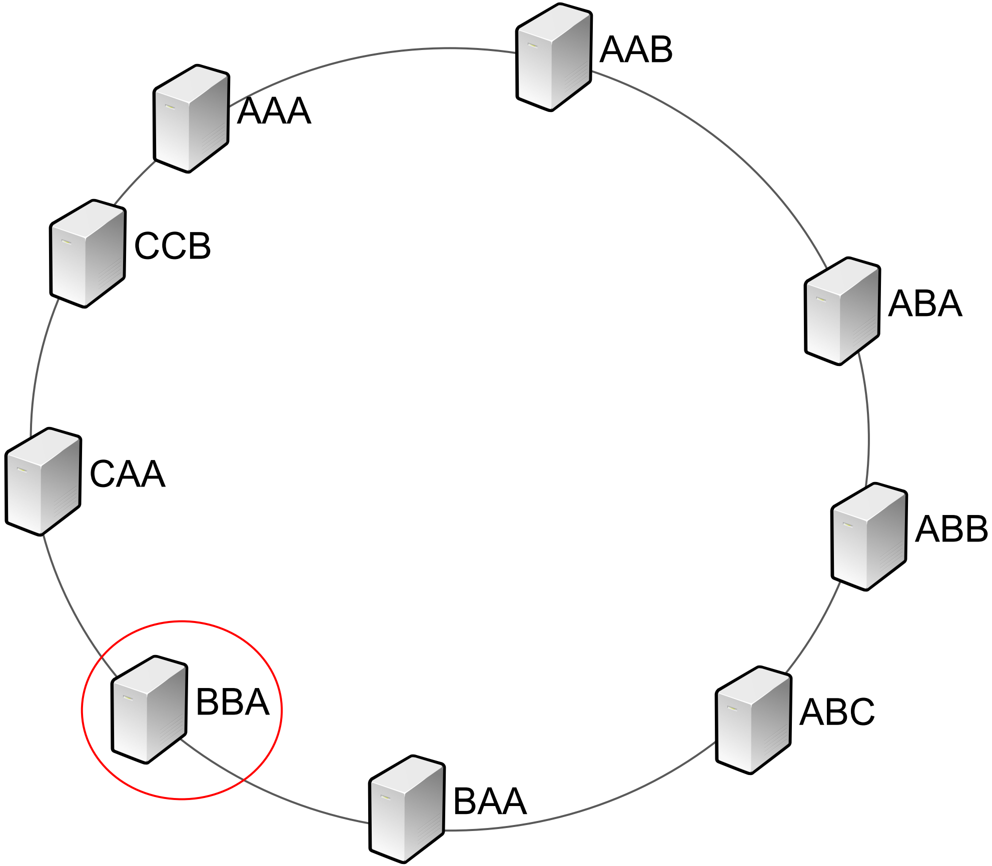


lookup(CCC)

RT:

ABB	*IP*
AAB	*IP*
BBA	*IP*

What's your superpower: DHT

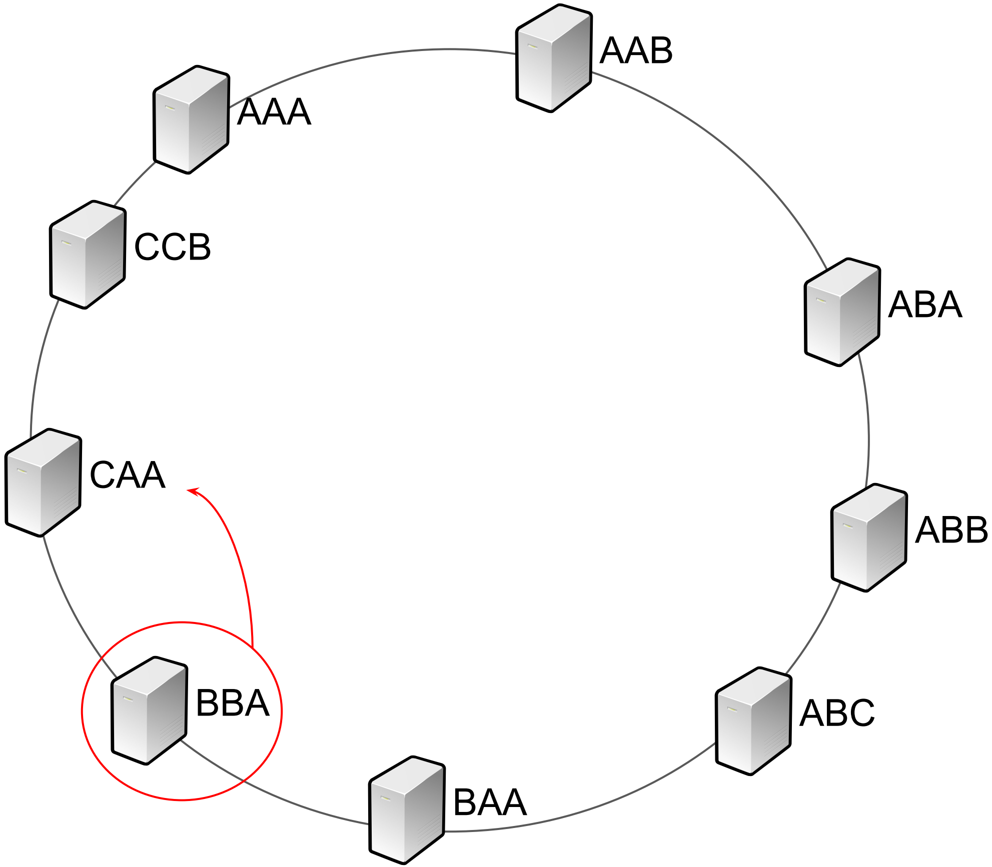


lookup(CCC)

RT:

CAA	*IP*
BBA	*IP*
AAB	*IP*

What's your superpower: DHT

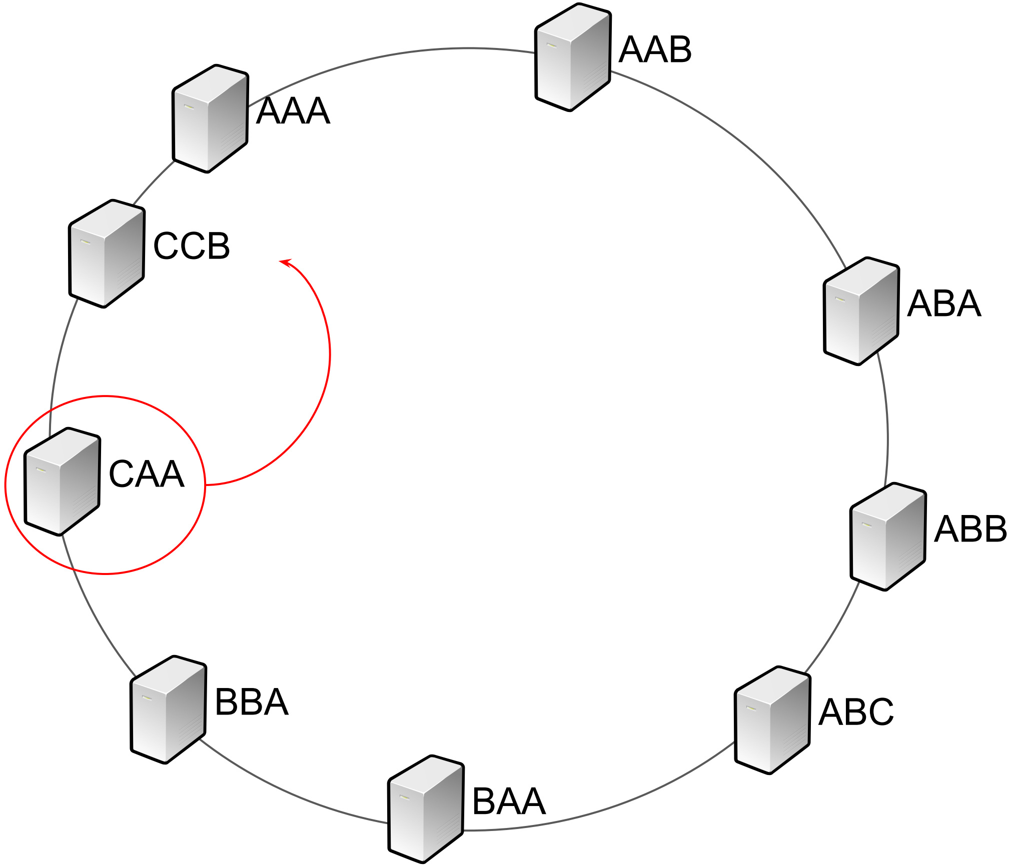


lookup(CCC)

RT:

CAA	*IP*
BBA	*IP*
AAB	*IP*

What's your superpower: DHT



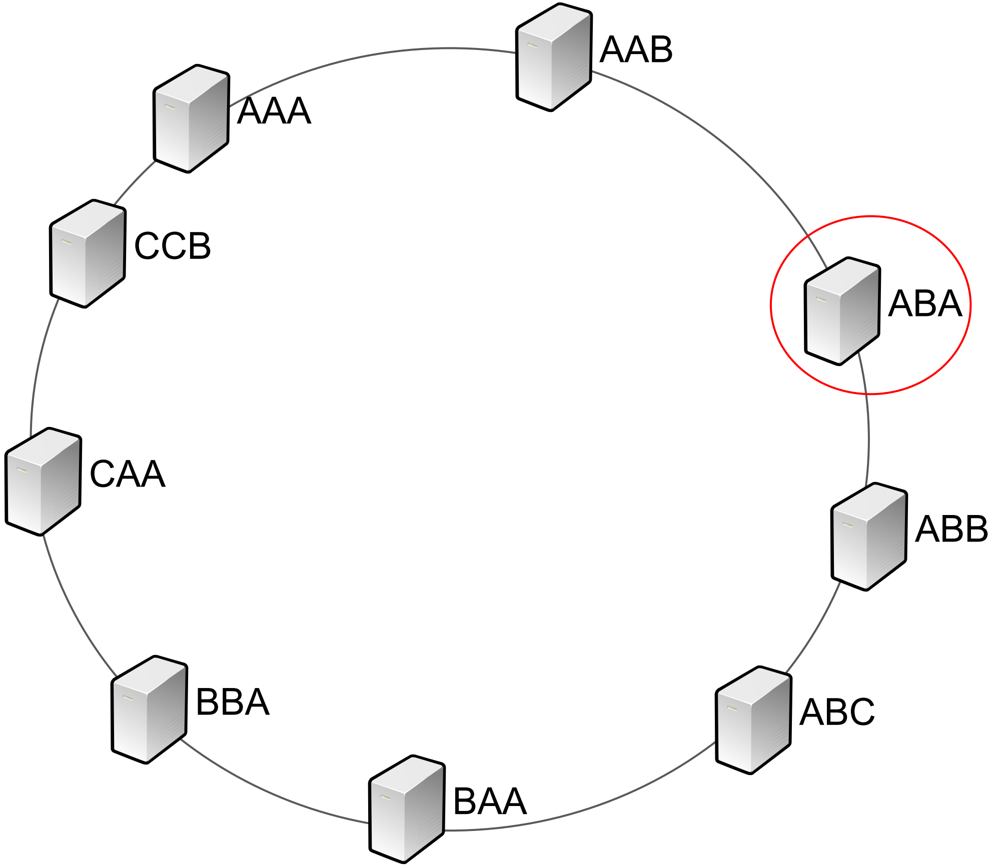
lookup(CCC)

RT:

CCB	*IP*
BBA	*IP*
ABB	*IP*

What's your superpower: DHT

What's your superpower: DHT

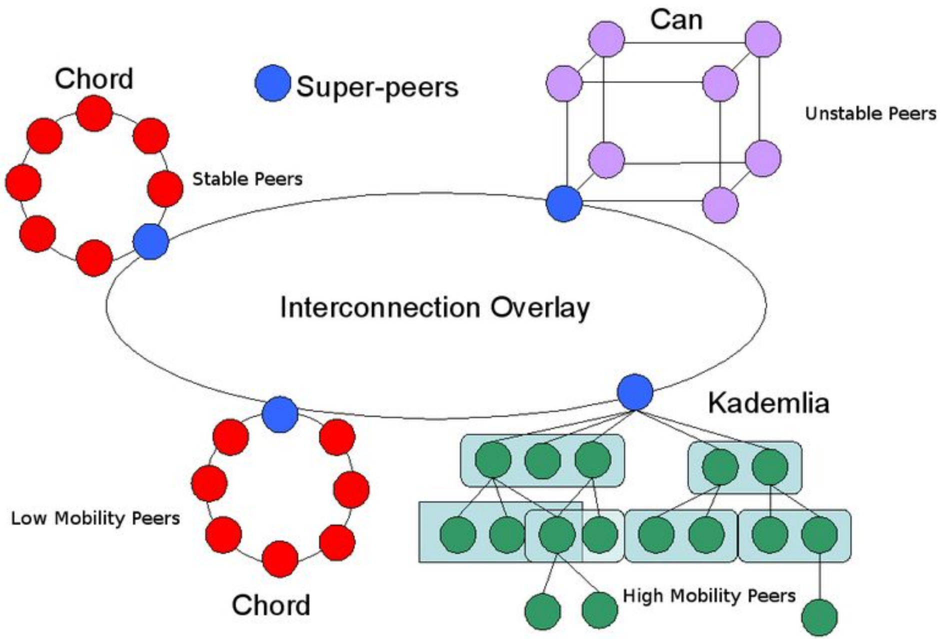


lookup(CCC)

RT:

ABB	*IP*
AAB	*IP*
BBA	*IP*

What's your superpower: DHT



Martinez-Yelmo, Isaias & Bikfalvi, Alex & Guerrero, Carmen. (2018). Benefits on using H-P2PSIP in mobile environments.

Routing tables define the **network overlay**

Different routing protocols have different superpowers!

What's your superpower: DHT

Key-based routing (KBR)

No need for coordination

Performant for large networks

Lookup is loop free

What's your superpower: DHT

Key-based routing (KBR)

Consistent Hashing

Performance

Latency with high churning

Sybil attacks

Anonymity/privacy

DHT maintain and lookup state in a P2P network

Metadata Resistant DHT

https://github.com/gpestana/notes/issues/8

https://pdht.hashmatter.com