Distributed Hash Table

A intro to DHT

2022.07.01

A brief history of networking

Some abstraction
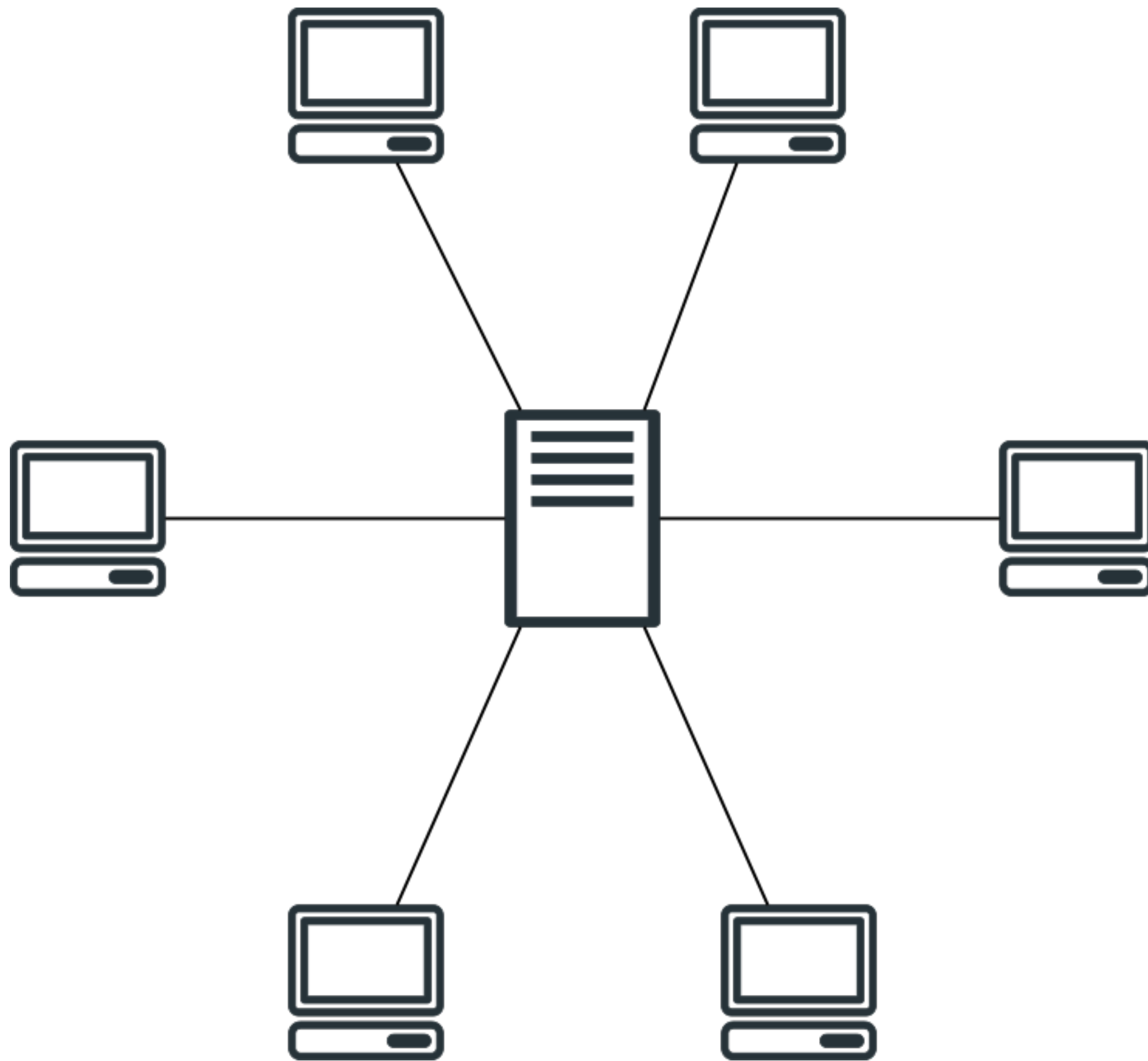## "Adding a new level of abstraction!"

- Some computers

- A service: get resource *v* for corresponding key *k*

- That is exactly why it is a "HashTable" !

- *k, v* are bijection: each computer will get same resource v_0 for same key k_0

- Certain level of security: keep working if some computer are shut down accidentally

CS networking
Client-Server

- A server handles all request

- Widely-used

- Failed when server was shut down

- Anonymous is impossible

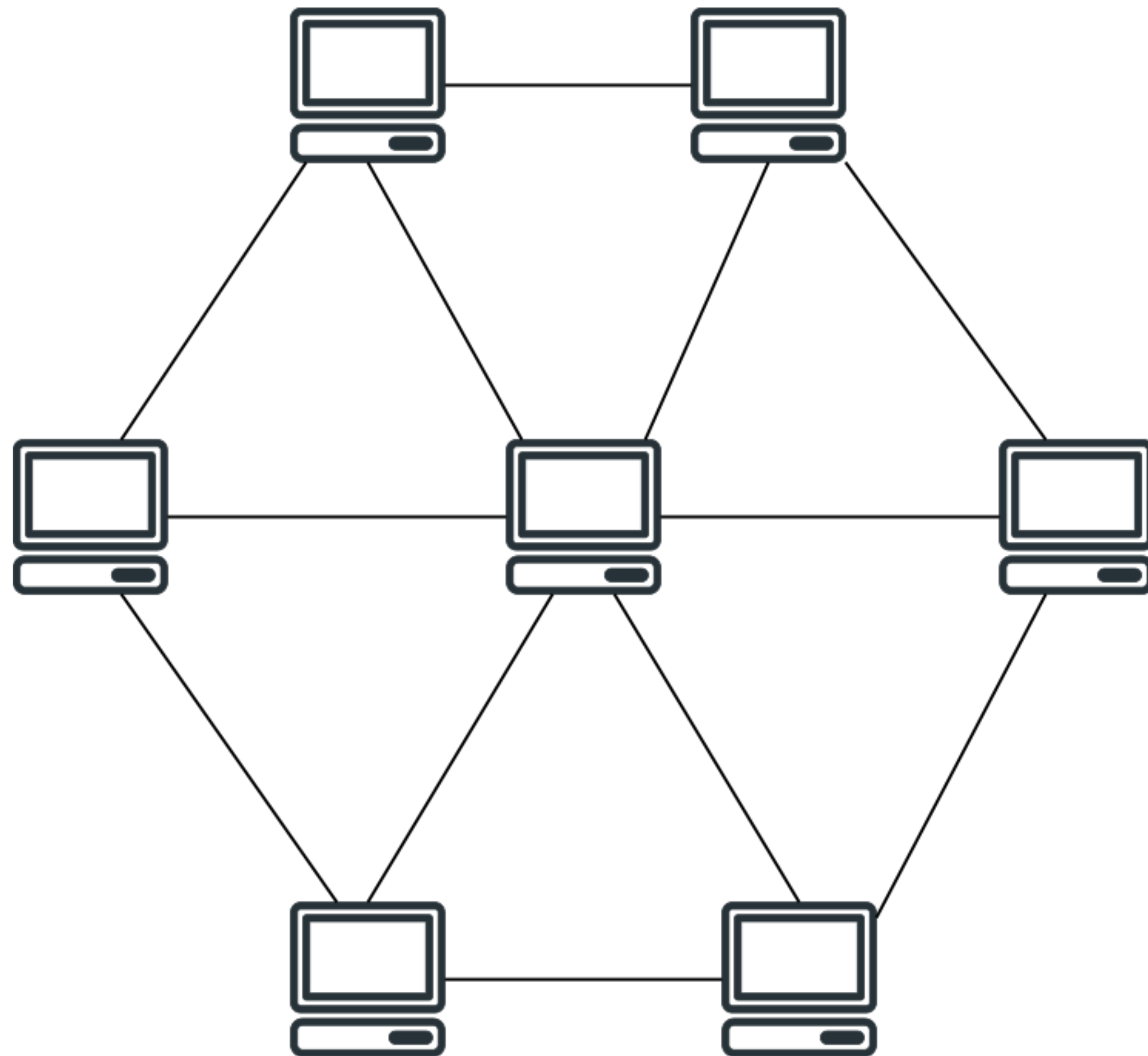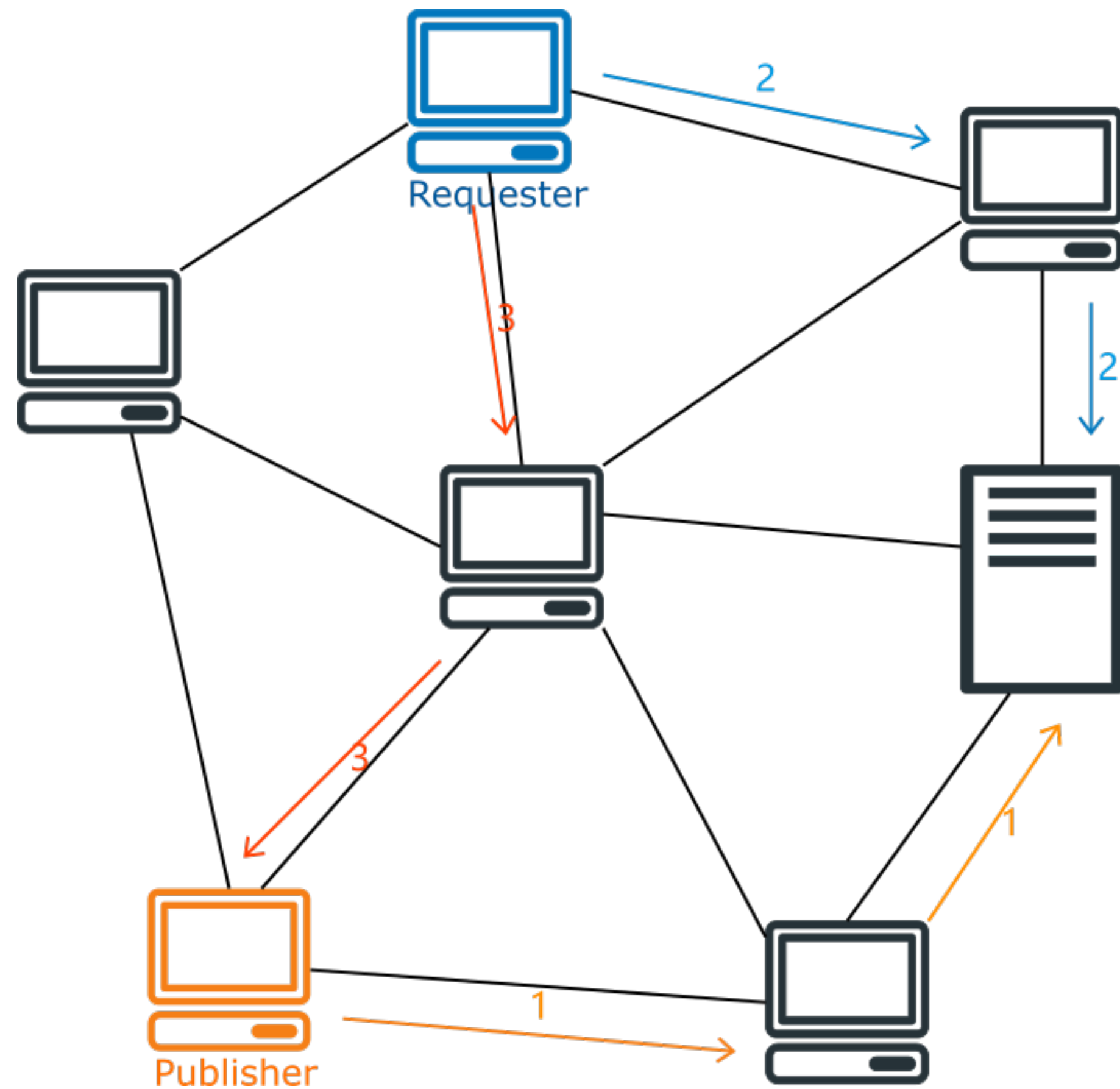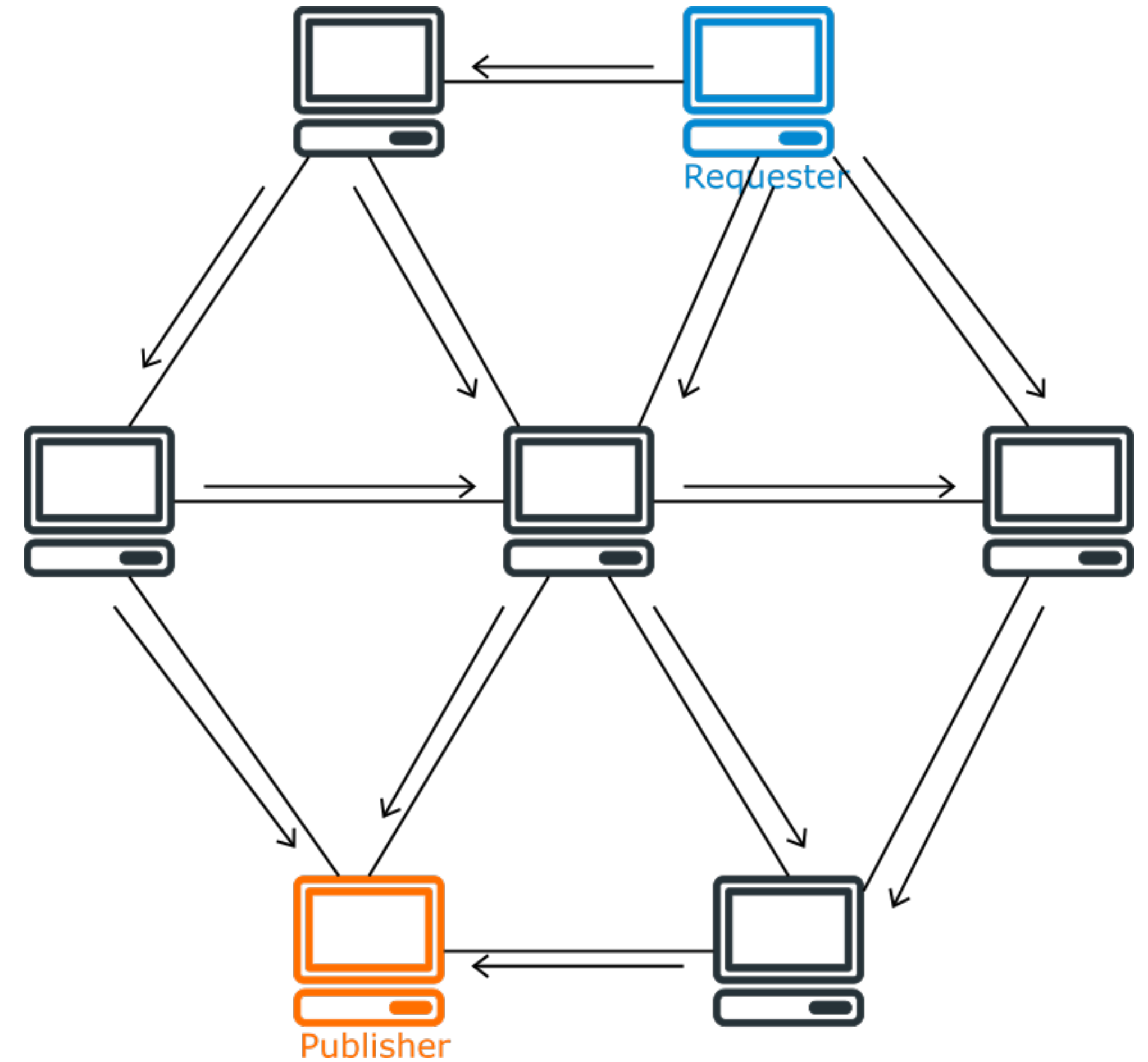|	微博热搜		↻ 每分钟更新	
↑	中国开展第12次北极科学考察			热
1	b站崩了	1186万		爆
2	港大不再承认学生会在…	175.5万		热
3	A站也崩了	129.8万		新

P2P networking
Peer-to-Peer

A **Peer-to-Peer System** is a self-organizing system of equal, autonomous entities (peers) which aims for the shared usage of distributed resources in a networked environment avoiding central services.

- Decentralized: no central server

- Self-organizing: manage itself

- DISTRIBUTED: resources were separated to all computers in the net

- Robustness: keep working if some computer offline accidentally

Some naïve implementations
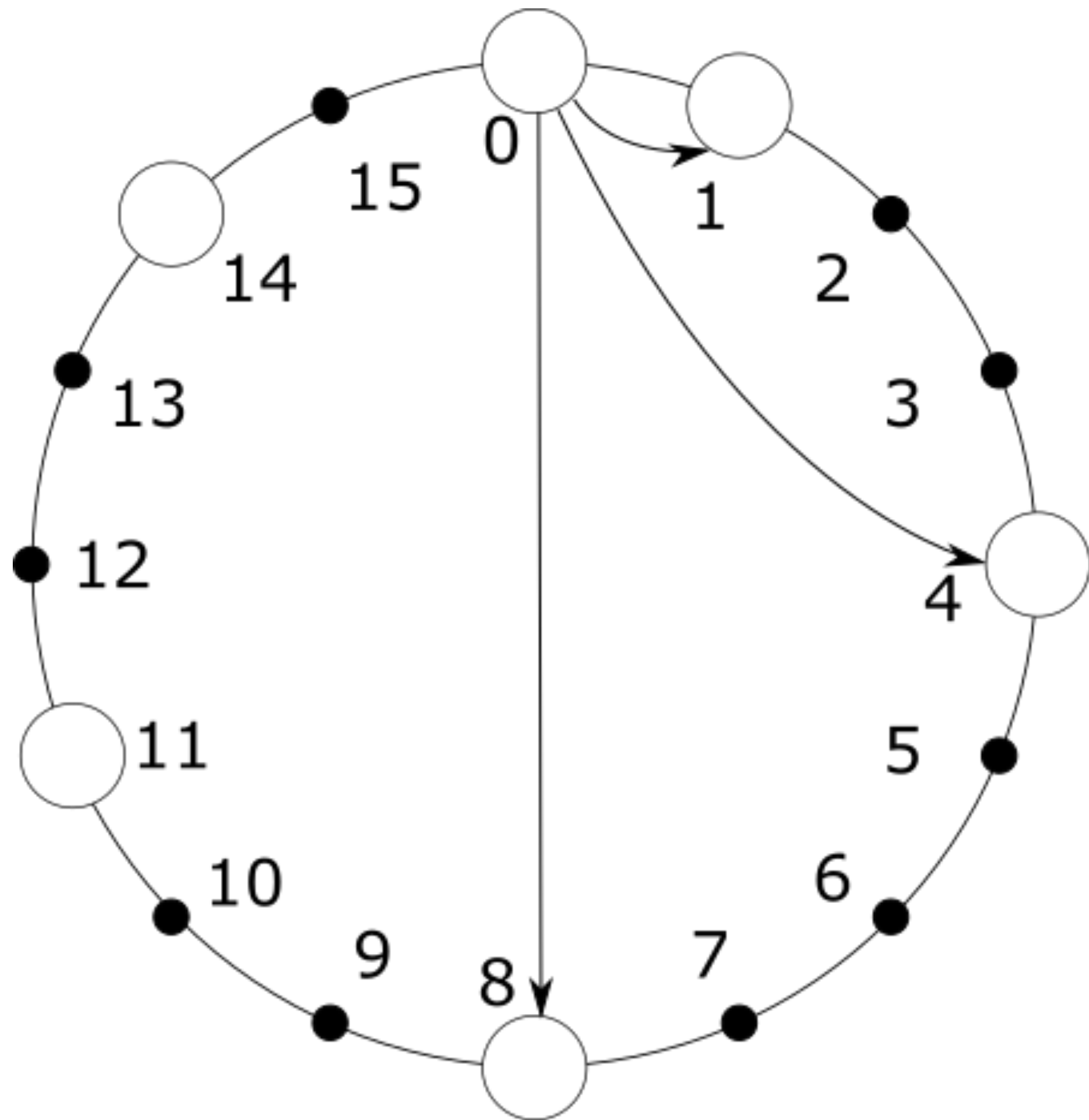

Location Manager

Flooding Search

What you will implement
Something advanced

• Intuition:

• Hash computers and keys to identifier

• Define "distance" between identifier

• Let "nearest" computer to store resources

• Using some router algorithm to find corresponding computer efficiently
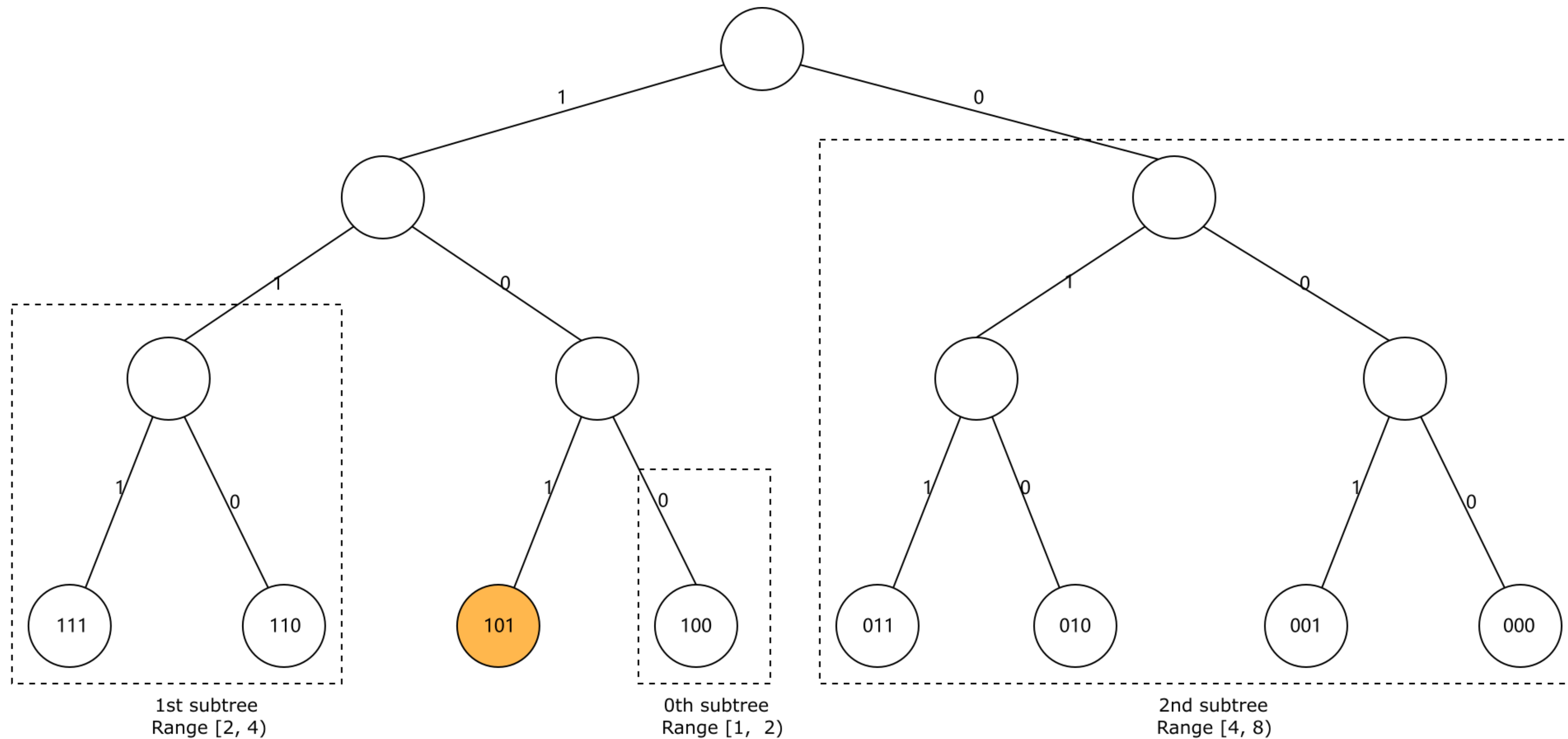
What you will implement
Chord Protocol

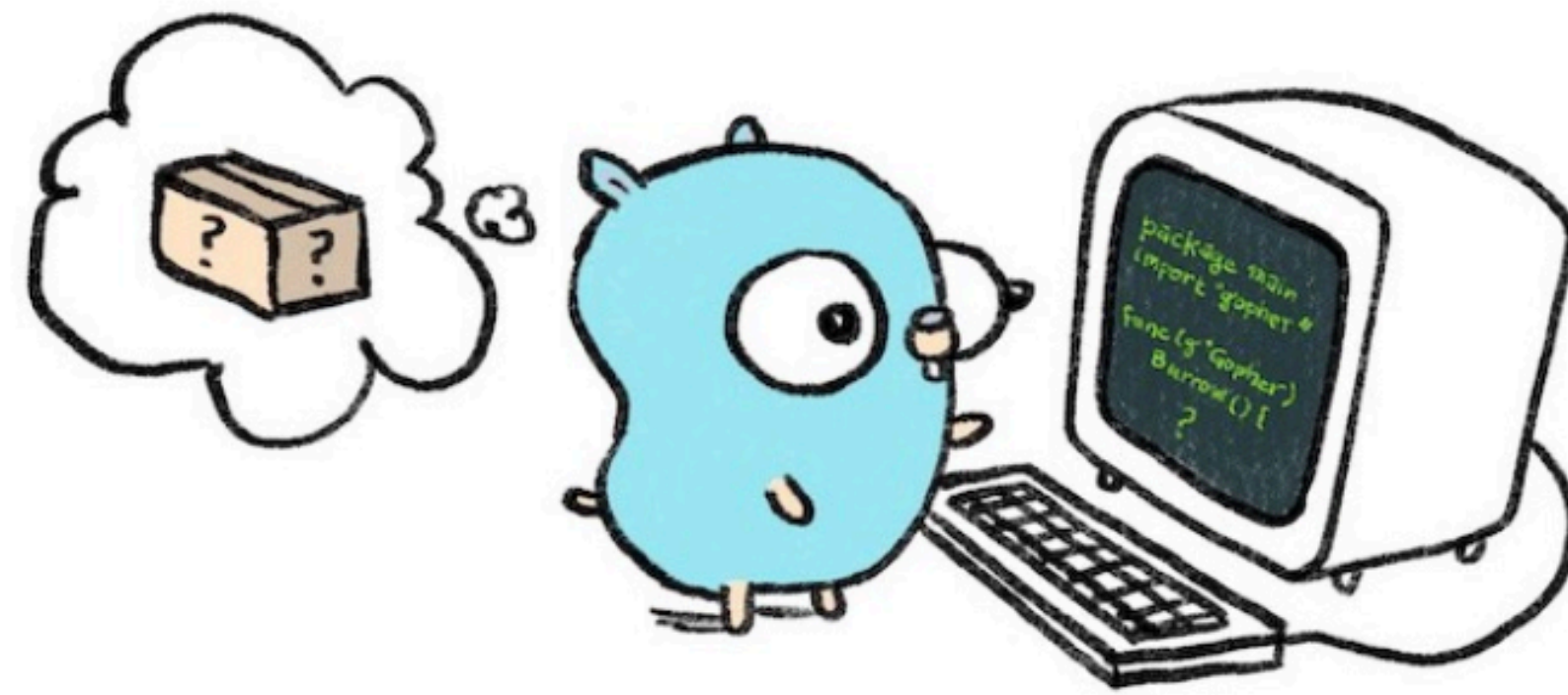| i	$n+2^{i-1}$	$successor(n+2^{i-1})$
1	1	1
2	2	4
3	4	4
4	8	8

What you will implement
Kademila Protocol

Programming language
go go go!

- Goroutine: "multiprocessing"

- Well-support for RPC (remote process call)

- Auto garbage collection

- (Self-)Learning a new language!

One more thing…

Create an application based on DHT!
Have fun :)

- Some previous work:

- Stupid Chord Peer-to-peer File Sharing System "SCPFSS"

- Naive Chord Peer-to-peer Instant Message Software "NCPIMS"

- ……

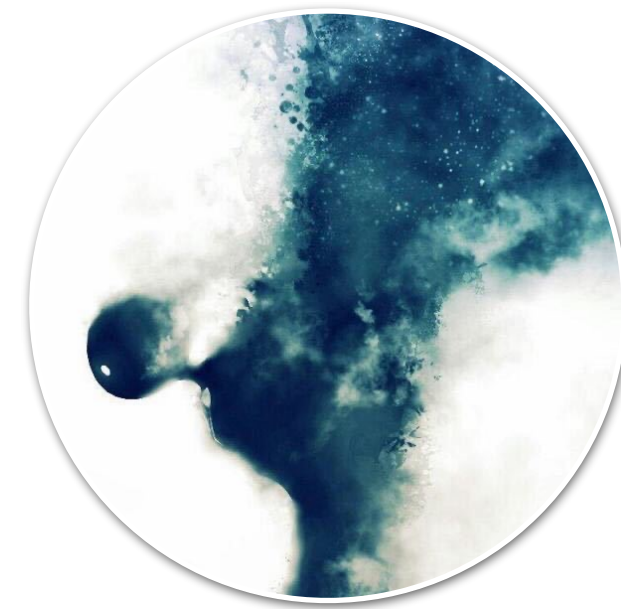- ~~It should have a cool name made up of 6 capital letters!~~

Score

~~35%+35%+20%+10%=100%! What a coincidence!~~

- 35% for Chord protocol

- 35% for Kademila protocol

- 20% for Application

- 10% for a short report and code review

Thanks and Welcome!

HappyPig

RainyMemory