Peer-to-Peer Computing: DHTs and Overlay Networks

Romaric Duvignau
Associate Professor, Chalmers University of Technology, Gothenburg, Sweden
https://www.cse.chalmers.se/~duvignau/
duvignau@chalmers.se

Today's lecture

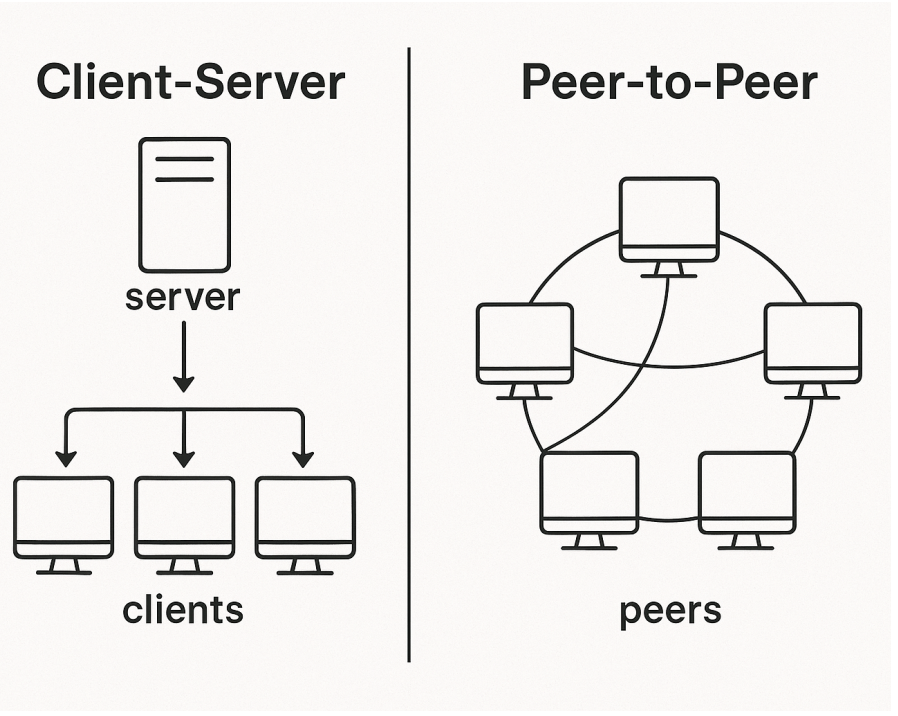
Example of unstructured overlays

- Napster 1999 vs. Gnutella 2000
- BitTorrent 2001

Structured Overlays / DHT

- How?
- Chord

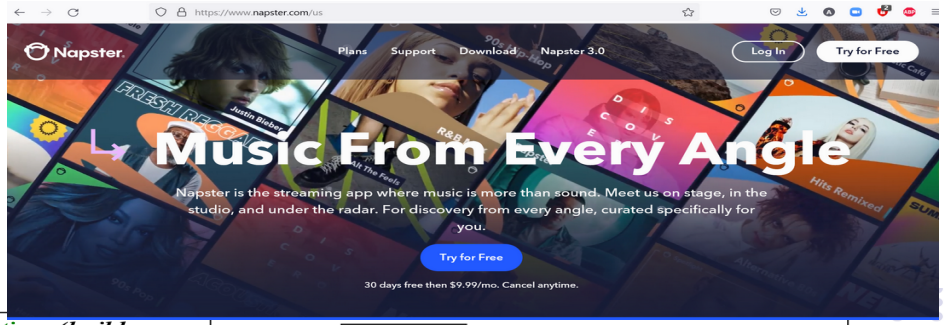
Overlay Network?

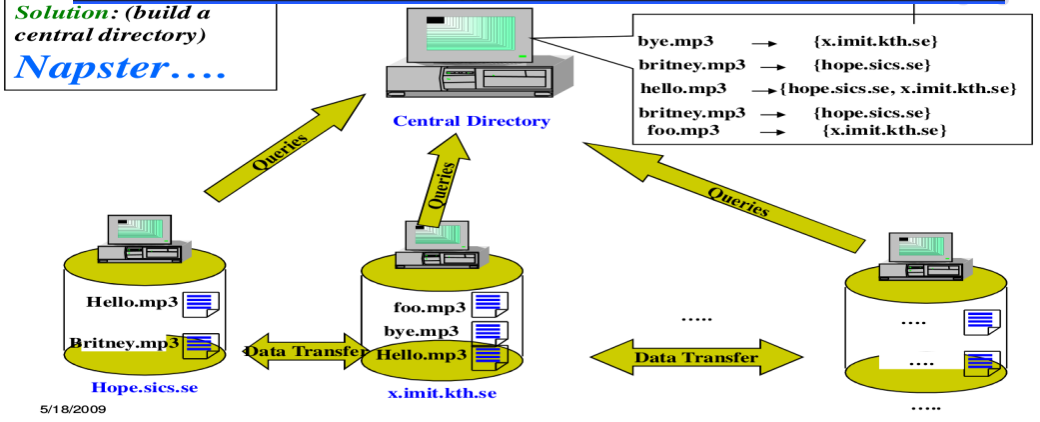


Unstructured overlays

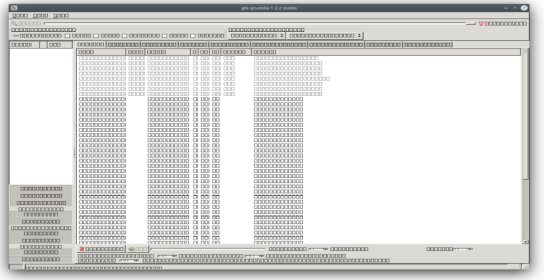
Unstructured Overlays

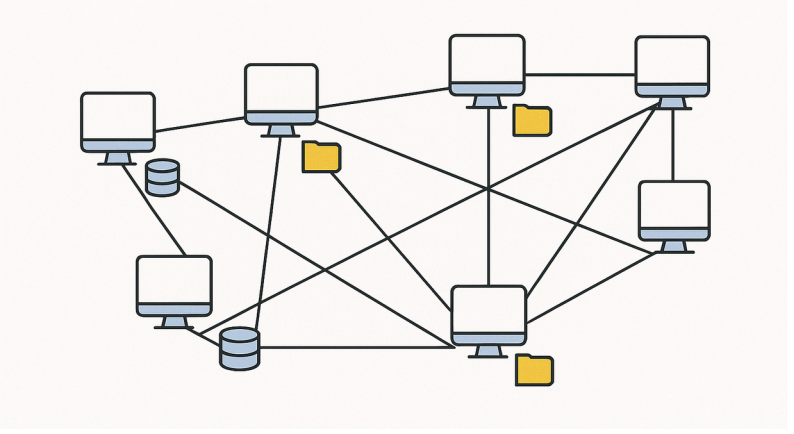
Napster





Gnutella





straightforward

Can be plugged off

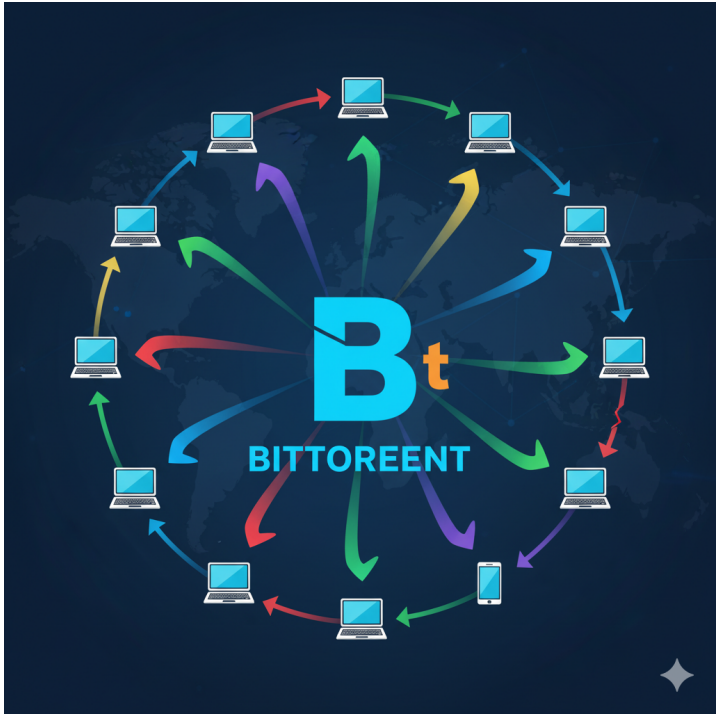
distributed

flooding

BitTorrent: Content distribution with a twist

- Large number of peers want to download a certain file as fast as possible.
- All previous techniques discussed **do not do that**!

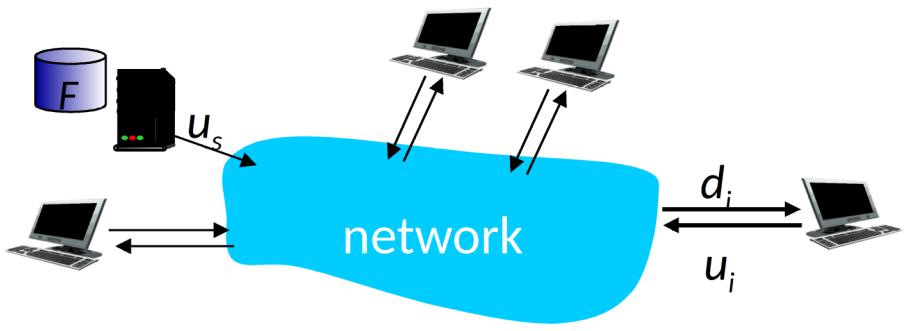
Bittorent's philosophy:
Use the bandwidth of everyone in the network to the maximum!



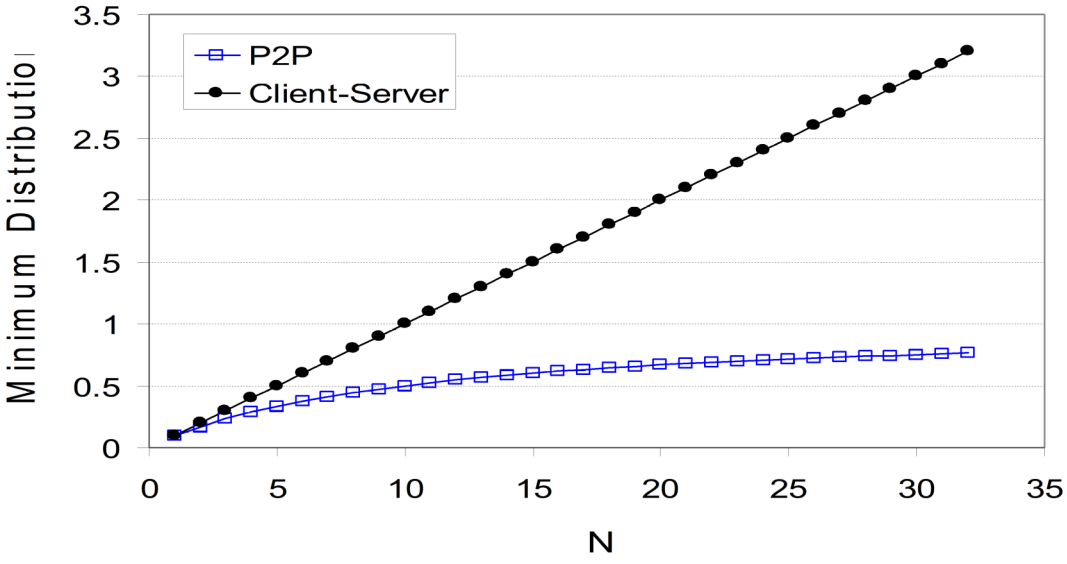
The advantage of P2P for file sharing

1 server = slow download for N peers

P2P = start sharing as soon as possible



client upload rate = u, F/u = 1 hour, $u_s = 10u$, $d_{min} \geq u_s$



Converges to F/u = **O(1)** instead of **O(N)**.

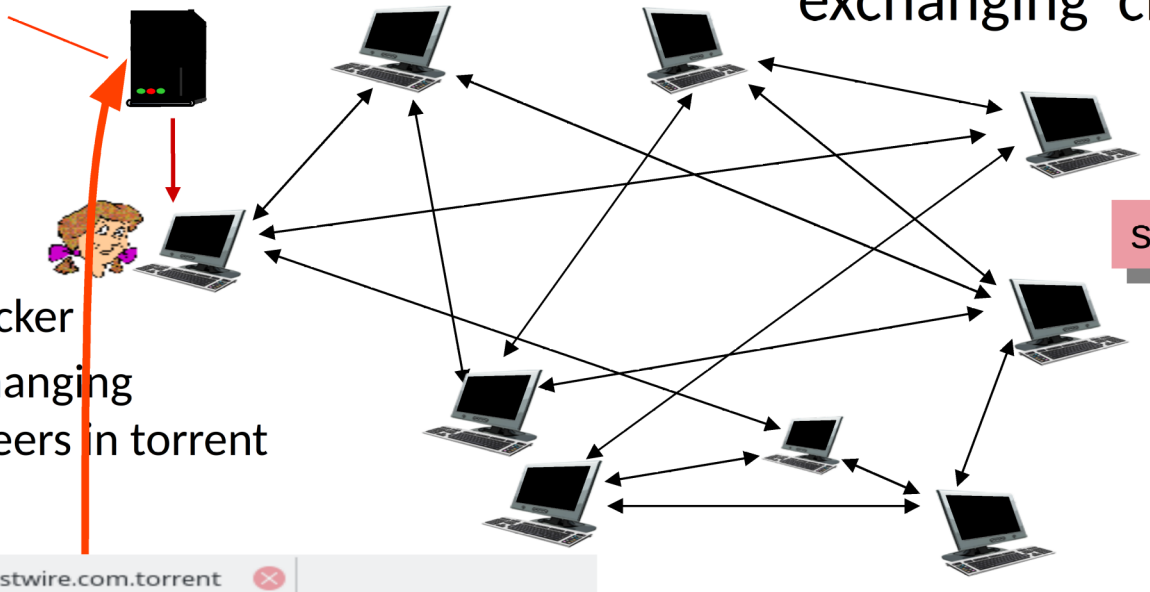
time to distribute F to N clients using P2P approach

$$D_{P2P} > max\{F/u_s, F/d_{min}, NF/(u_s + \Sigma u_i)\}$$

BitTorrent in action

tracker: tracks peers participating in torrent

swarm: group of peers exchanging chunks of a file



Alice arrives …

… obtains list of peers from tracker

… and begins exchanging file chunks with peers in torrent

some are leechers

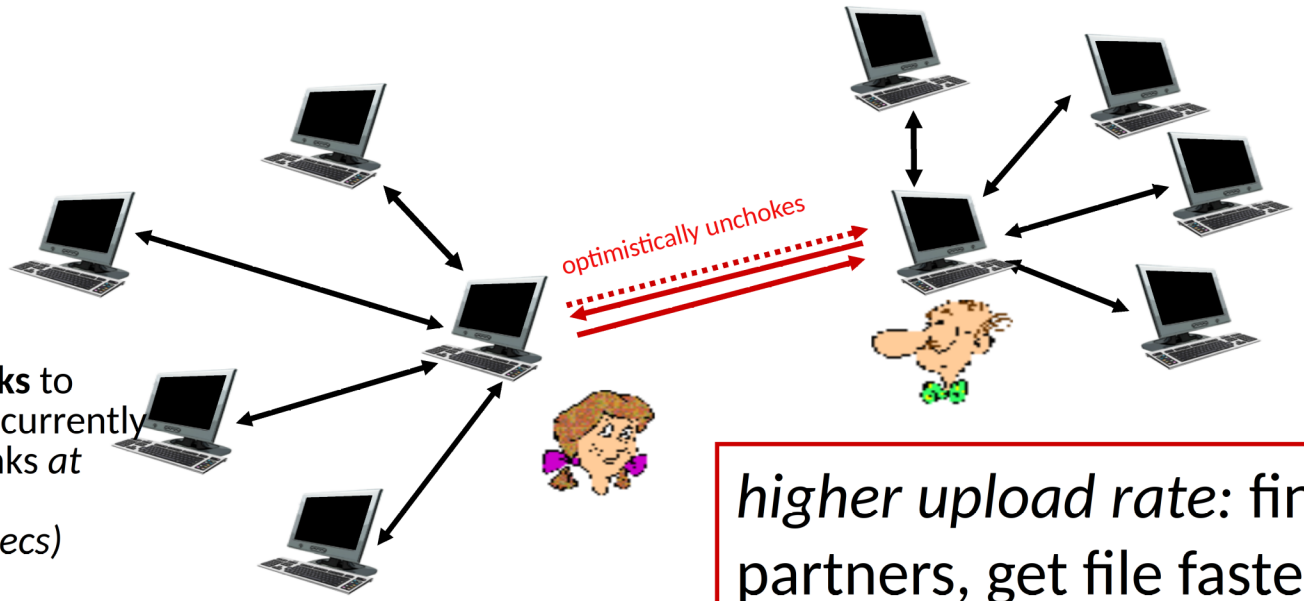
some are seeders

Big_Buck_Bunny_1080p_s..._frostwire.com.torrent

d8:announce42:http://tracker.frostwire.com:6969/announce13:creation datei1231188572e4:infod5:filesld6:lengthi928670754e4:pathl47:Big_Buck E_YOUR_CONTENT_ON_FROSTWIRE_01_06_09.txteed6:lengthi3456234e4:pathl39 e.txteee4:name58:Big_Buck_Bunny_1080p_surround_frostclick.com_frostwi

Choking algorithm: Mutual love or Tit-for-tat

Files are divided in **fixed-size chunks**, downloaded in rarest-first/random/streaming order.



other peers are choked by Alice (do not receive chunks from her)

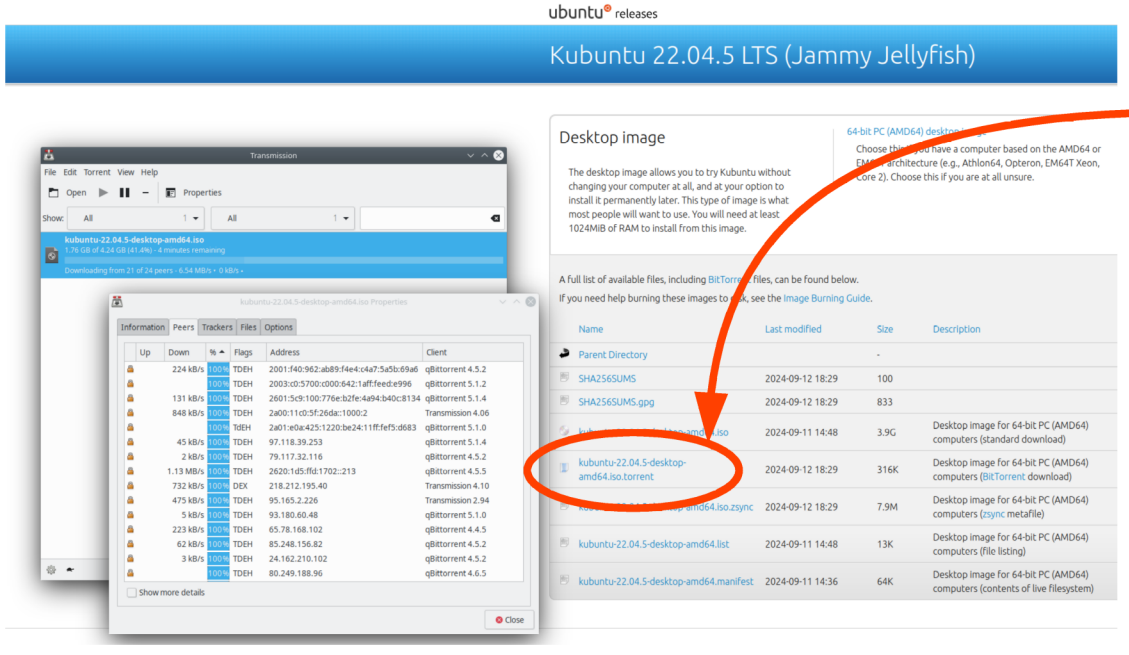
Alice sends **chunks** to those four peers currently sending her chunks *at highest rate* *(check every 10 secs)*

higher upload rate: find better trading partners, get file faster!

Reflections on BitTorrent (historical)

Hybrid P2P/centralized architecture, flexible and high performance!

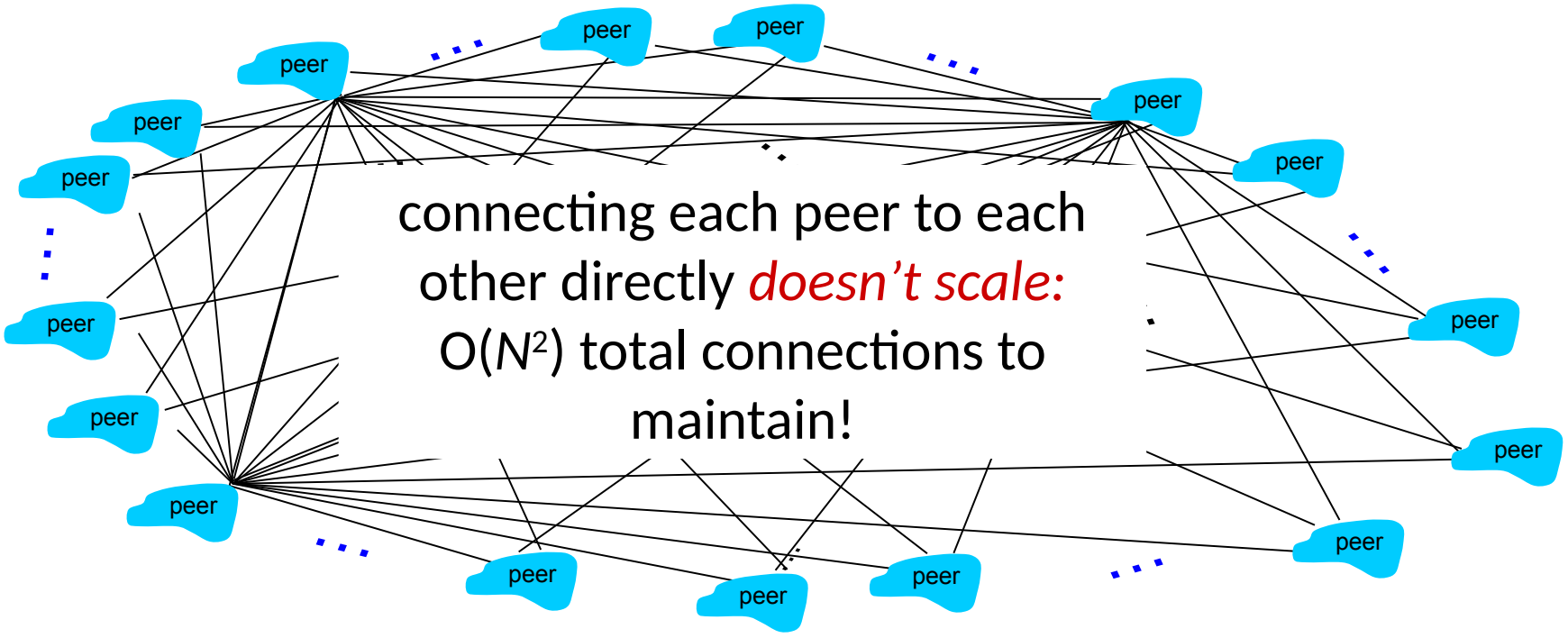
Dependence on a centralized component for the search function!



The rise of **DHTs**!

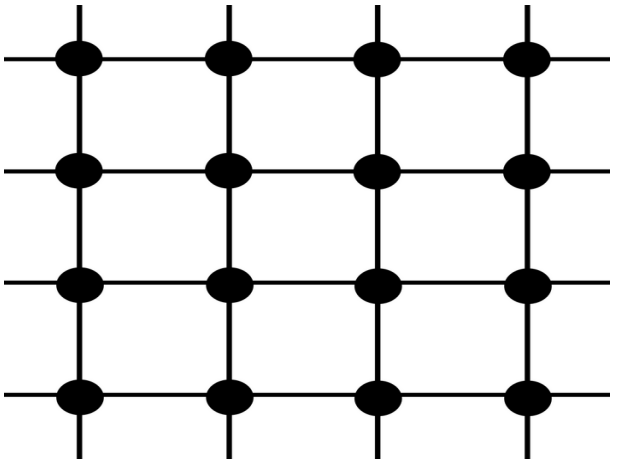
Structured Overlays

How to structure a P2P network?



Can we do better?

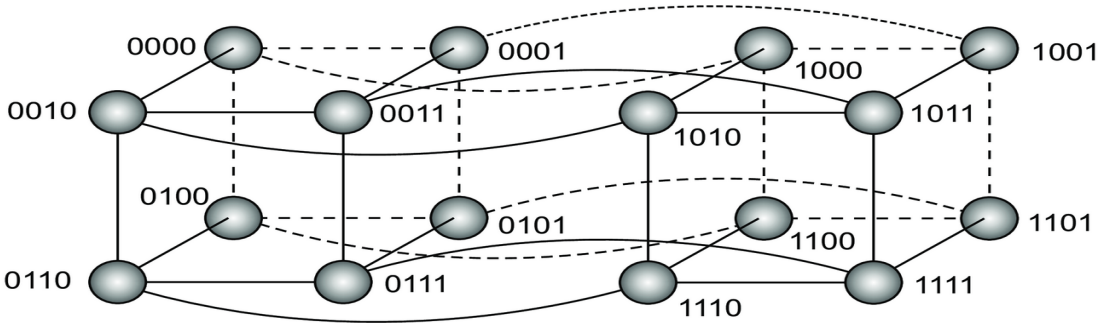
Lattice Graph



Degree: Constant!

Diameter: Linear!

Hypercube

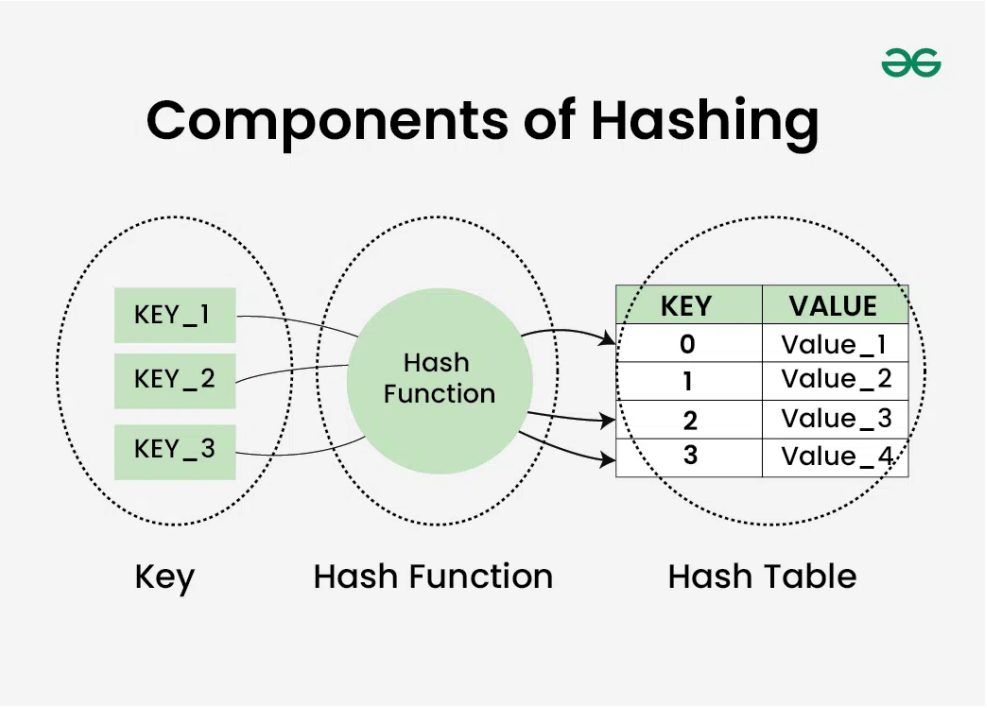


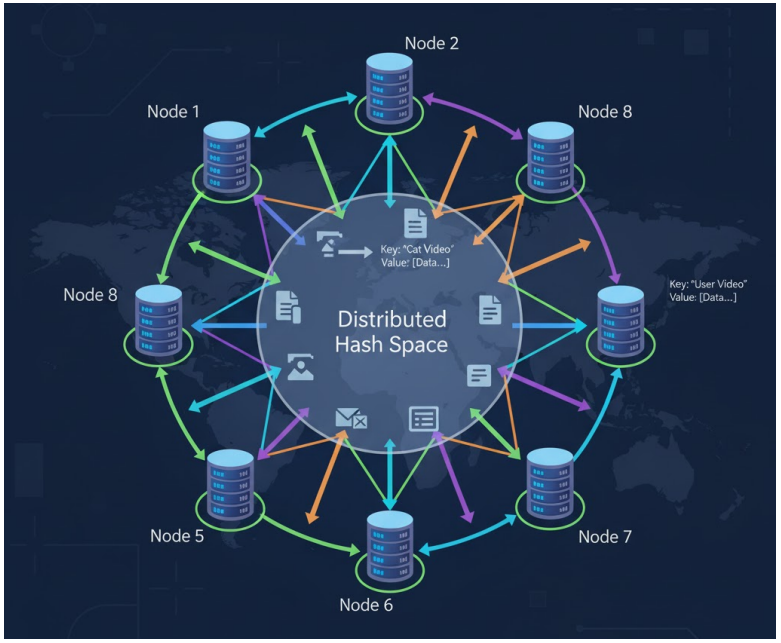
Degree: Logarithmic!

Diameter: Logarithmic!

Chord (2001) has similar properties to an hypercube!

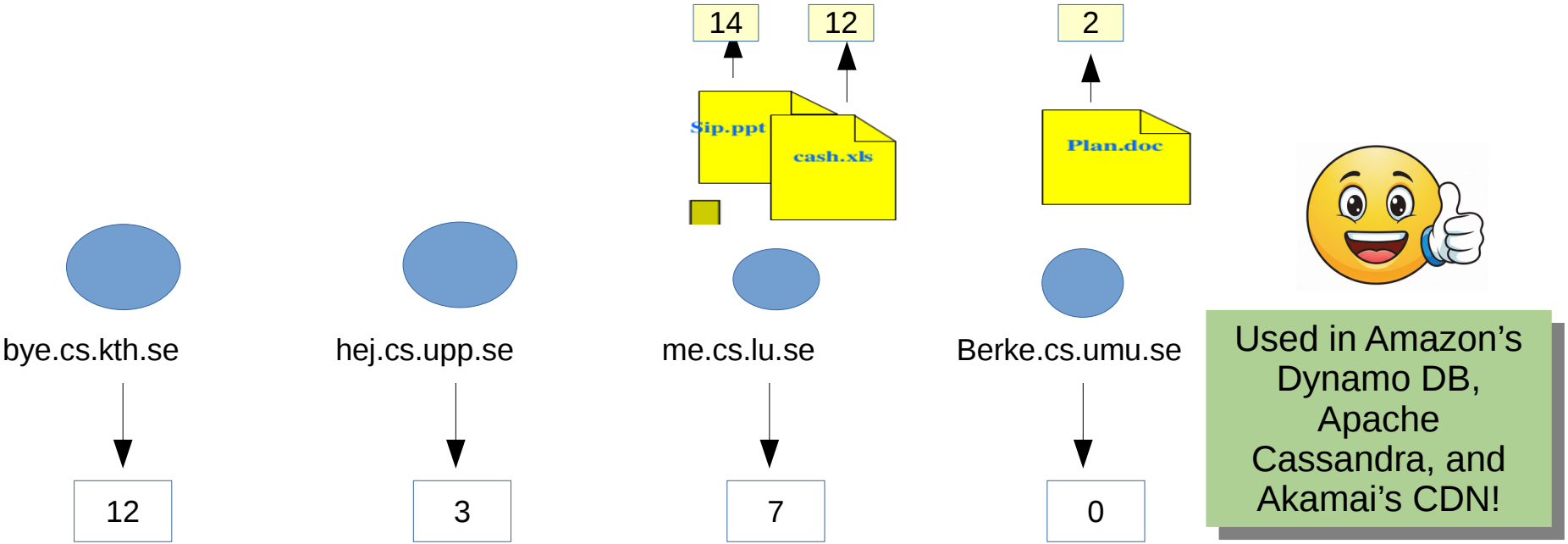
What is a DHT?





For example, **BitTorrent's DHT** (based on Kademlia) stores **infohash → <peer list>**

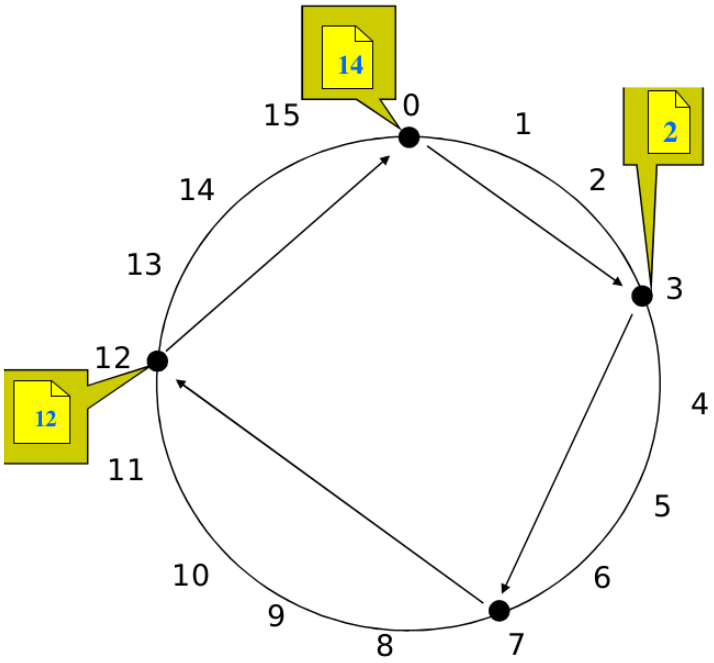
Simplest Example: Consistent Hashing



- 4 machines that need to share data
- Each machine will have an id based on the hash of its IP address
- Data will also have ids using the same function (SHA, MD5sum..)

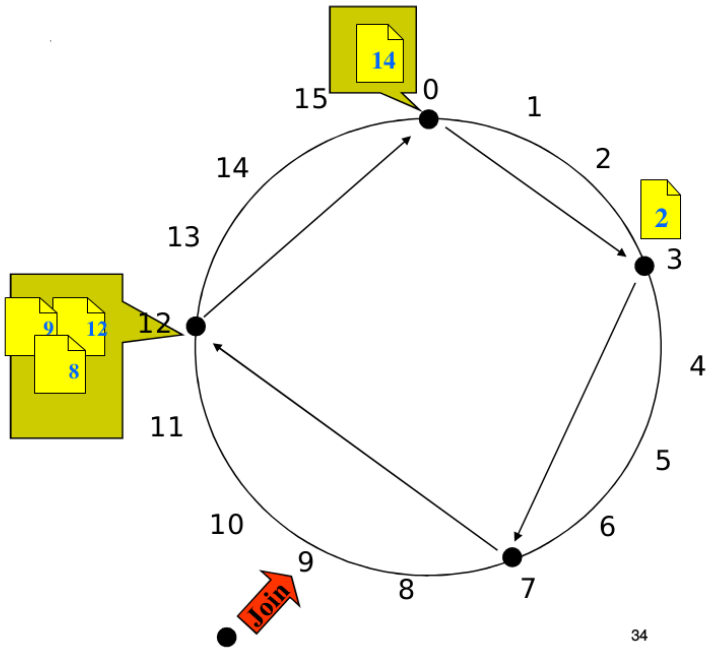
The **ID** of a data item determines the machine on which it is going to be stored.

Simple Example: build and update the ring



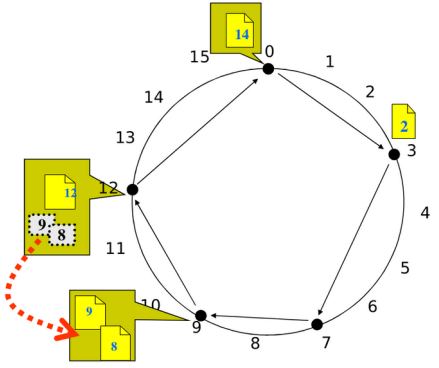
- Policy
  - A doc with **id y**, would be stored at **Succ(y)**
- Lookup
  - ask **node n** which you know about to find the successor of **id**.

New example: Handling Joins



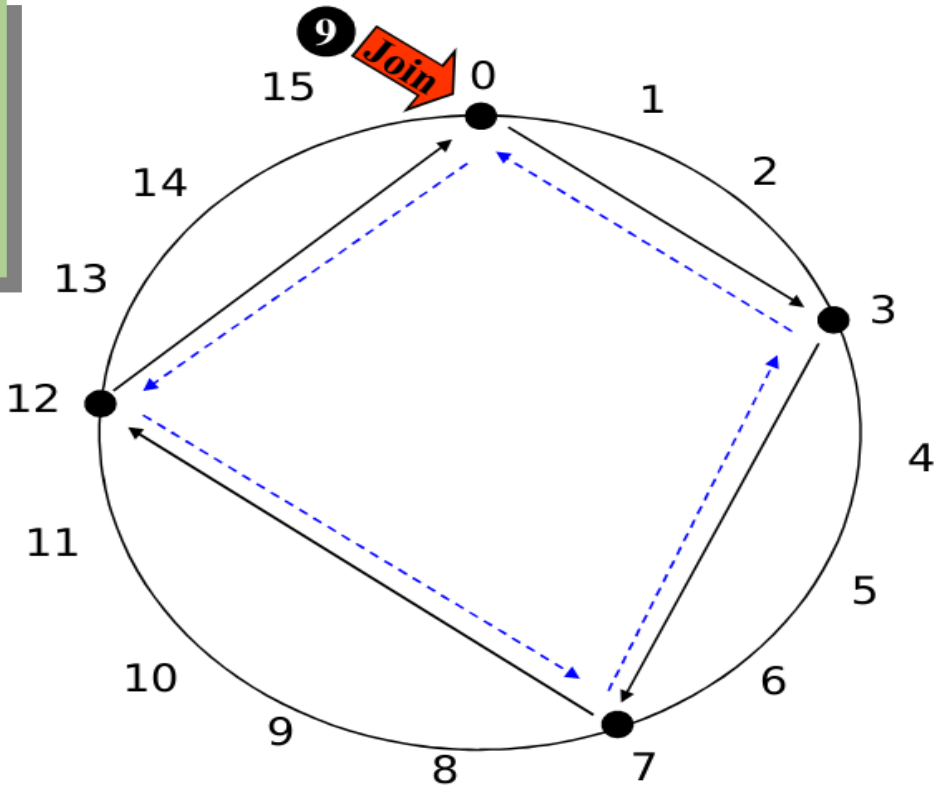
- You need to know the successor
- Your predecessor needs to know about you
- Example: A **node 9** joins the ring

Can you think what happens to data?



Chord & Chord's Join

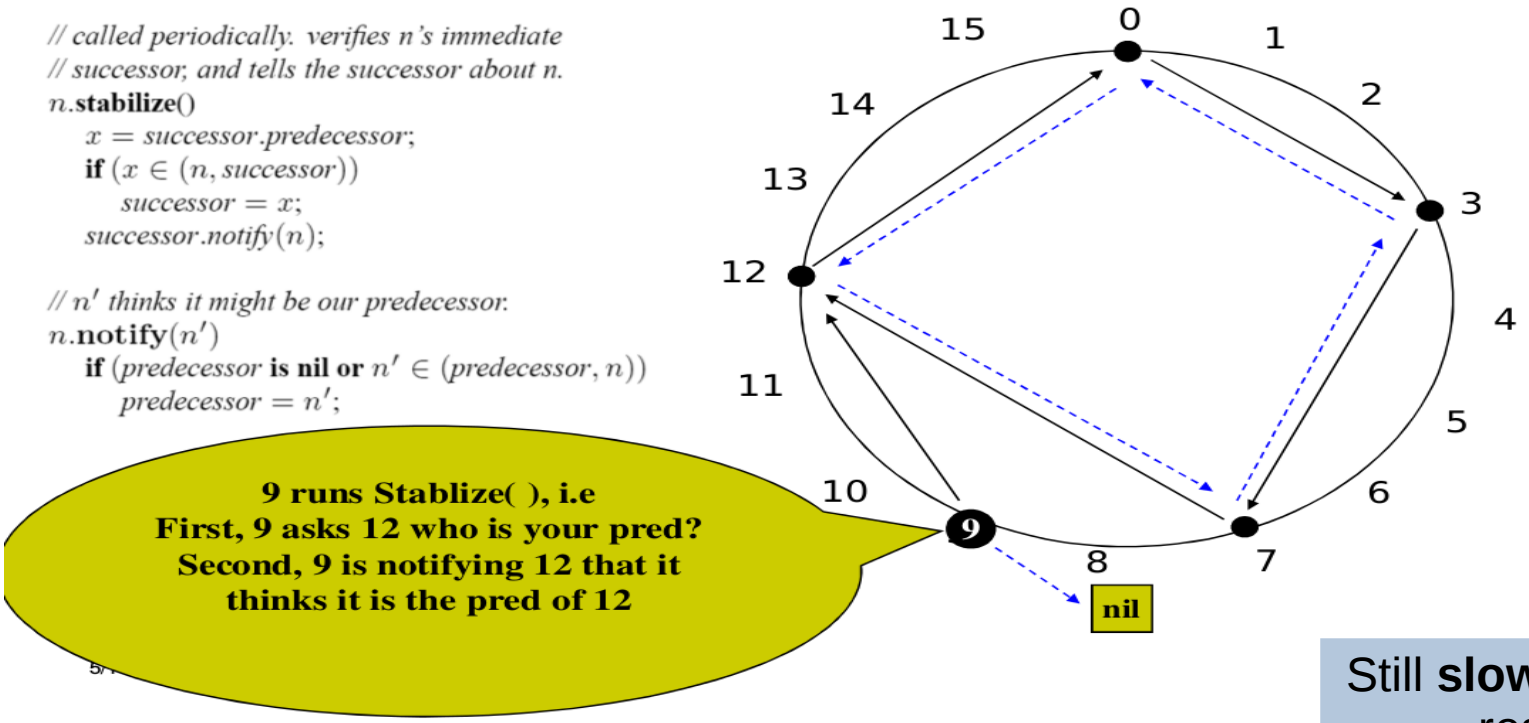
- Developed at MIT, published in 2001
- Very popular since it is very simple to understand
- In addition to the successor pointer, every node has a **predecessor pointer** as well



- 9 can join through any other node, take 0 for example.
- 9 will set its predecessor to nil
- 0 will help 9 to find its successo
- 0 will tell 9 that its successor should be 12

Chord Join Cont'd

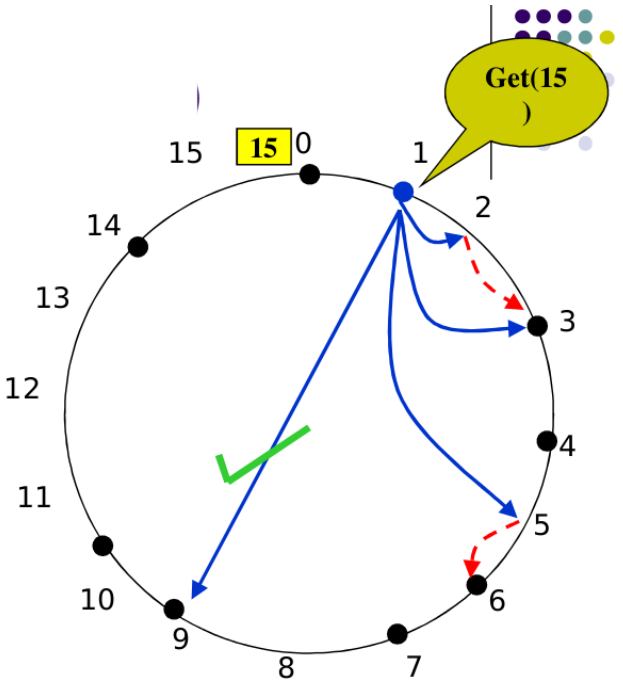
```
// called periodically. verifies n's immediate
// successor, and tells the successor about n.
n.stabilize()
    x = successor.predecessor;
    if (x ∈ (n, successor))
        successor = x;
    successor.notify(n);

// n' thinks it might be our predecessor.
n.notify(n')
    if (predecessor is nil or n' ∈ (predecessor, n))
        predecessor = n';
```



Still **slow** to retrieve a resource!

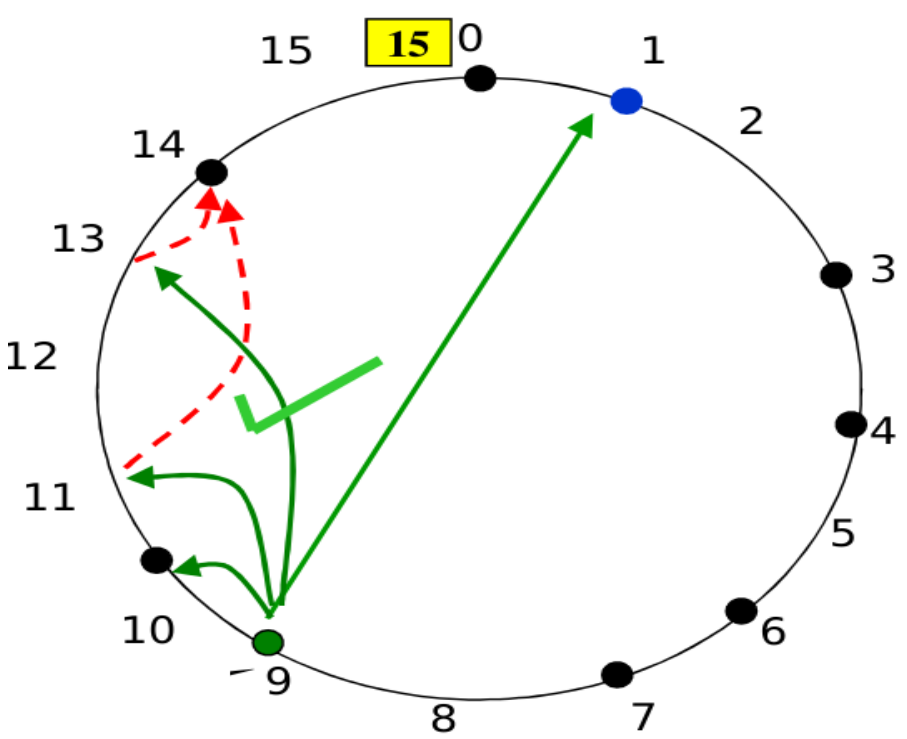
- 12 changes its predecessor to 9
- 7 will learn that 9 is it successor when it runs stabilize
  -12 will tell

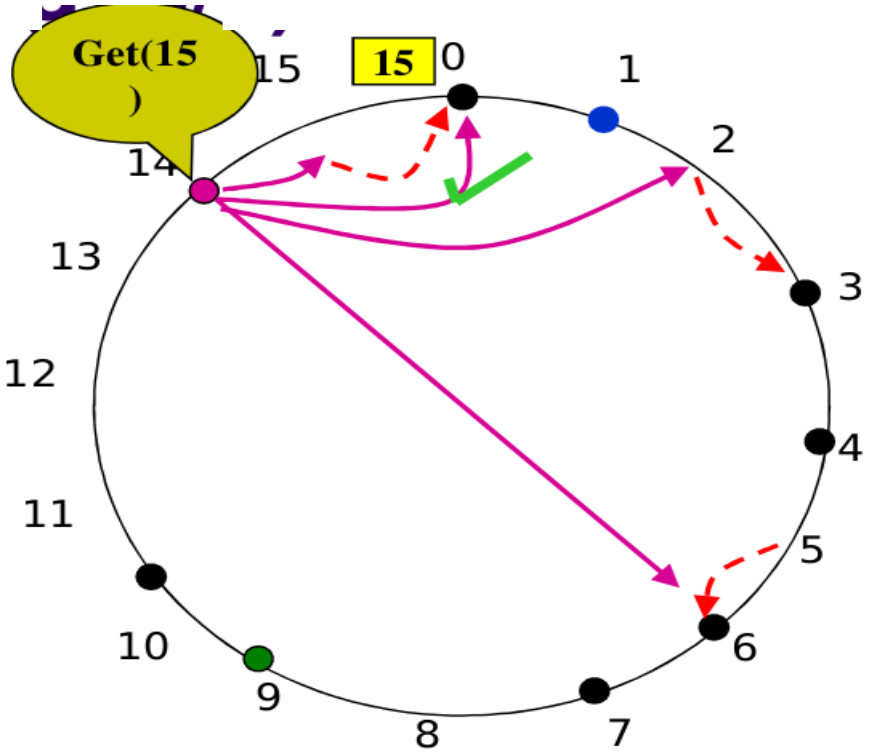
Routing: Fingers table



- Keep a routing table of M peers at each node
  - Where **$N=2^M$**
  - Maximum distance to any other node is then **$\log_2(N)$**, i.e., **M hops**

Routing: Fingers table





Perform a recursive search for the resource!

Chord: Routing

- In fact, if nodes are uniformly distributed, the maximum is l**og (# of nodes)**, i.e. log (8) hops between any two nodes
- The average complexity is: ***1⁄2* log(#nodes)**

If one successor fails, total ring collapses?

Would not it be nice to have something with locality?

Security of P2P networks: Sybil attacks

- Adversary introduces a large number of peers so to control the P2P system
  - Subverting the reputation system
  - Corrupting the network
  - Tricking the users
  - Controlling/influencing the network

Solution:
proof of work vs. **proof of stake**!

- Problem for consensus algorithms! (Blockchain, eg Bitcoin)

"One can have, some claim, as many electronic personas as one has time and energy to create."

Judith S. Donath (pioneer of sociable media / online identity research)

Conclusion: Drawbacks of P2P systems?

- Major drawbacks with P2P systems
  - Churn and performance guarantees
  - The power of **cloud computing**
    - **Spotify** abandoned their P2P streaming option (2014)
    - **Skype** abandoned their P2P architecture (2017)
  - Decreasing cost of content delivery / more legal download options

Modern P2P networks

1. Internal within companies and datacenters (eg Dynamo at Amazon)
2. Blockchains