中山大學
SUN YAT-SEN UNIVERSITY

区块链中的网络基础

Fundamentals of Blockchain Networks

吴迪

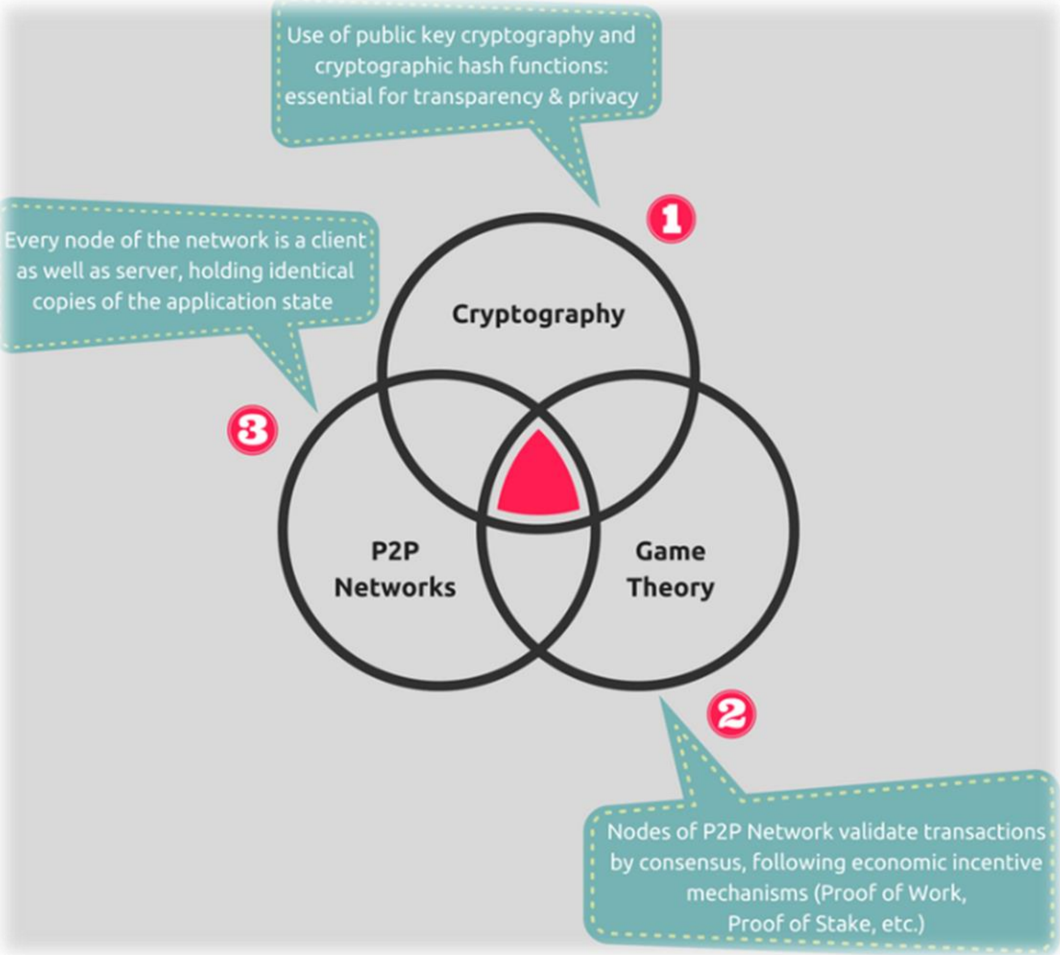
数据科学与计算机学院

2018年11月

Key Technologies of Blockchain

- Blockchain is built on top of three key technologies

Blockchain



Use of public key cryptography and cryptographic hash functions: essential for transparency & privacy

Every node of the network is a client as well as server, holding identical copies of the application state

Nodes of P2P Network validate transactions by consensus, following economic incentive mechanisms (Proof of Work, Proof of Stake, etc.)

Contents

1. Basics of P2P Networks
2. Bitcoin P2P Network
3. Ethereum P2P Network
4. Summary

Contents

1 **Basics of P2P Networks**

2 Bitcoin P2P Network

3 Ethereum P2P Network

4 Summary

1 P2P Basics

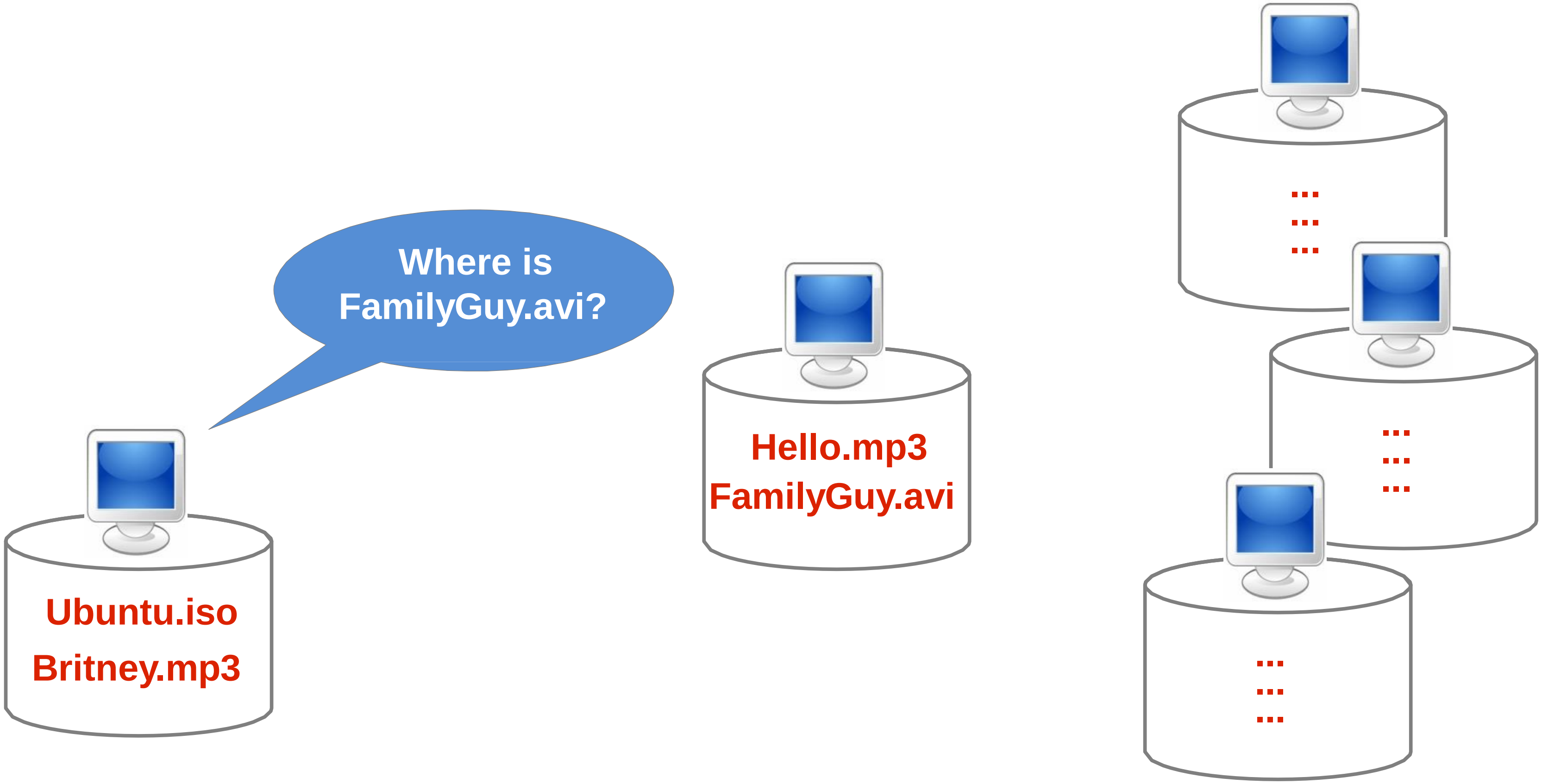
1 What is P2P Computing?

- P2P is a class of applications, that
  - Takes advantage of resources – (storage, cpu, etc,..) – available at the edges of the Internet
  - Unstable connectivity and unpredictable IP addresses, P2P nodes must operate outside the DNS system and have significant or total autonomy from central servers.

1 Let us see how did it all start ...

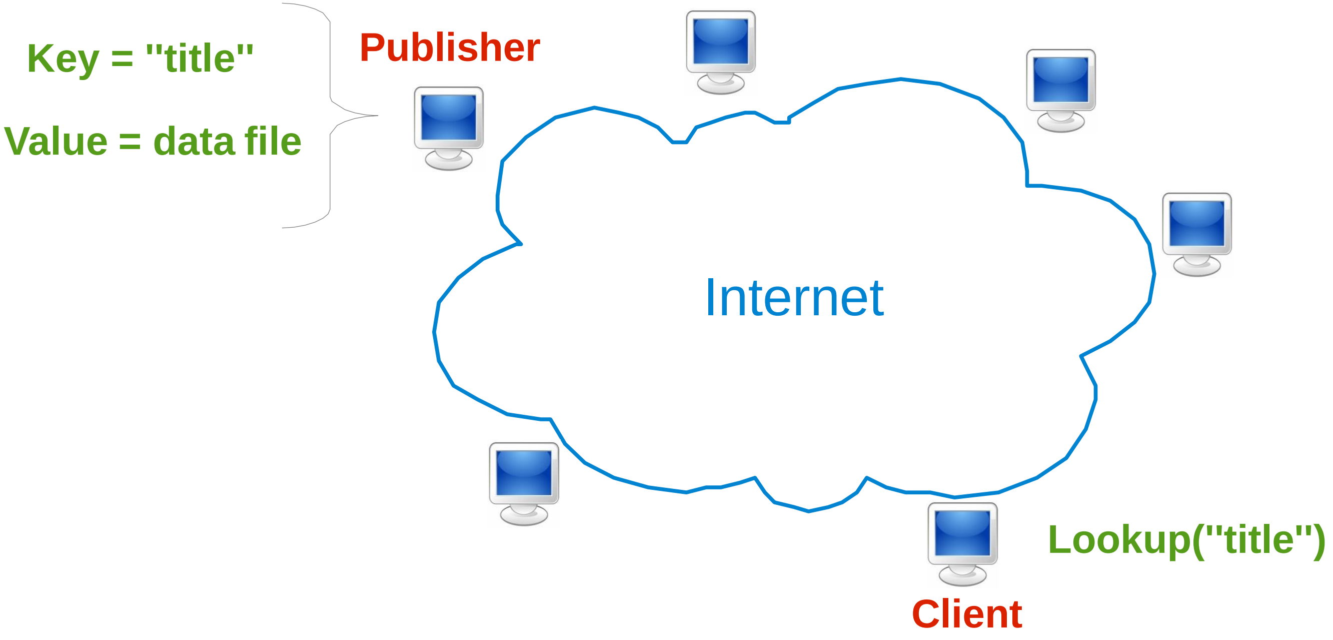
- Some users store data items on their machines.
- Other users are interested in this data.
- **Problem**: How does a user know which other user(s) in the world have the data item(s) that s/he desires?

1 Let us see how did it all start …



1 Example P2P Problem : Lookup

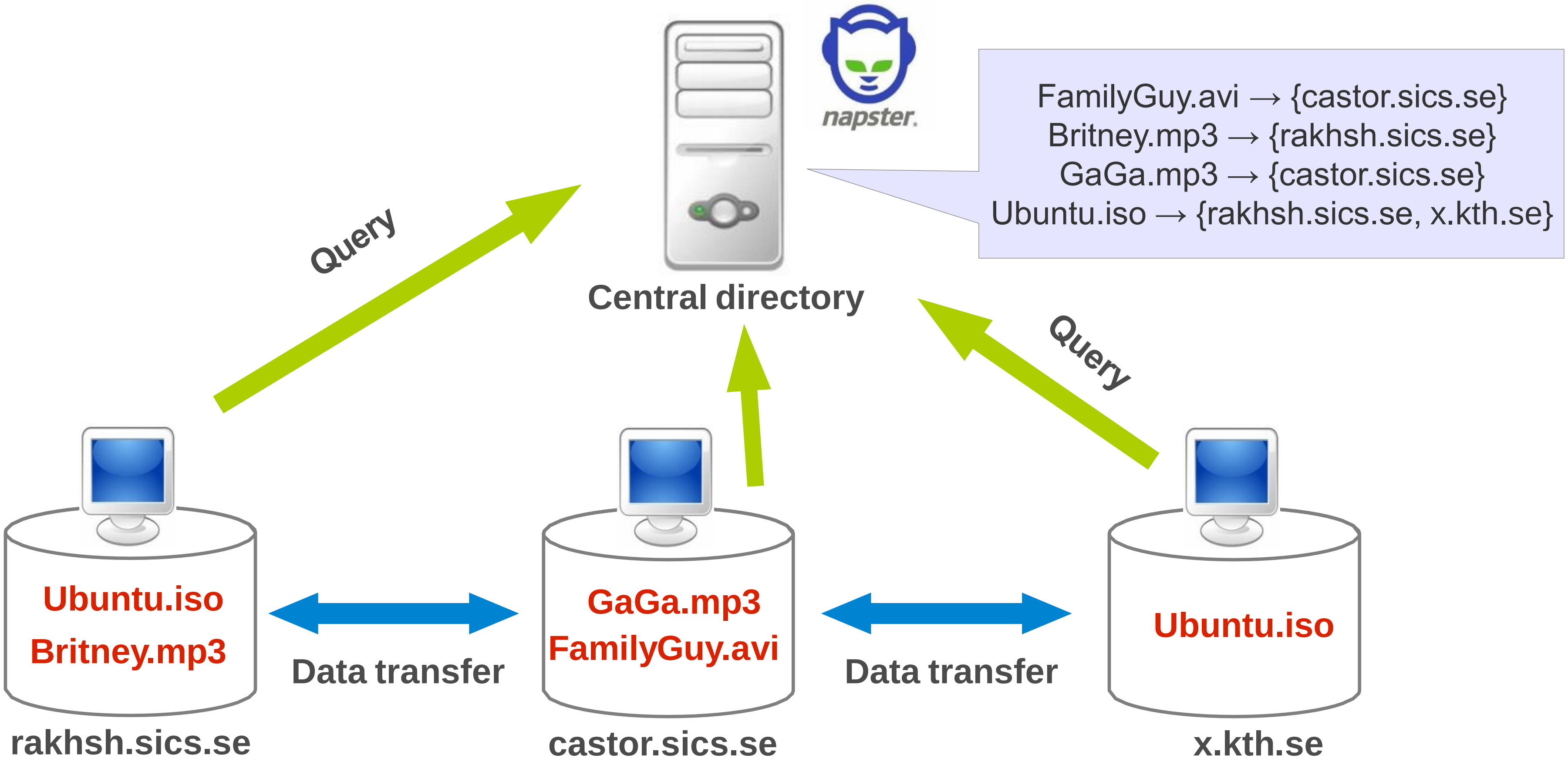
- At the heart of all P2P systems



First Generation

1 First Generation of P2P Systems

- Central directory + Distributed storage

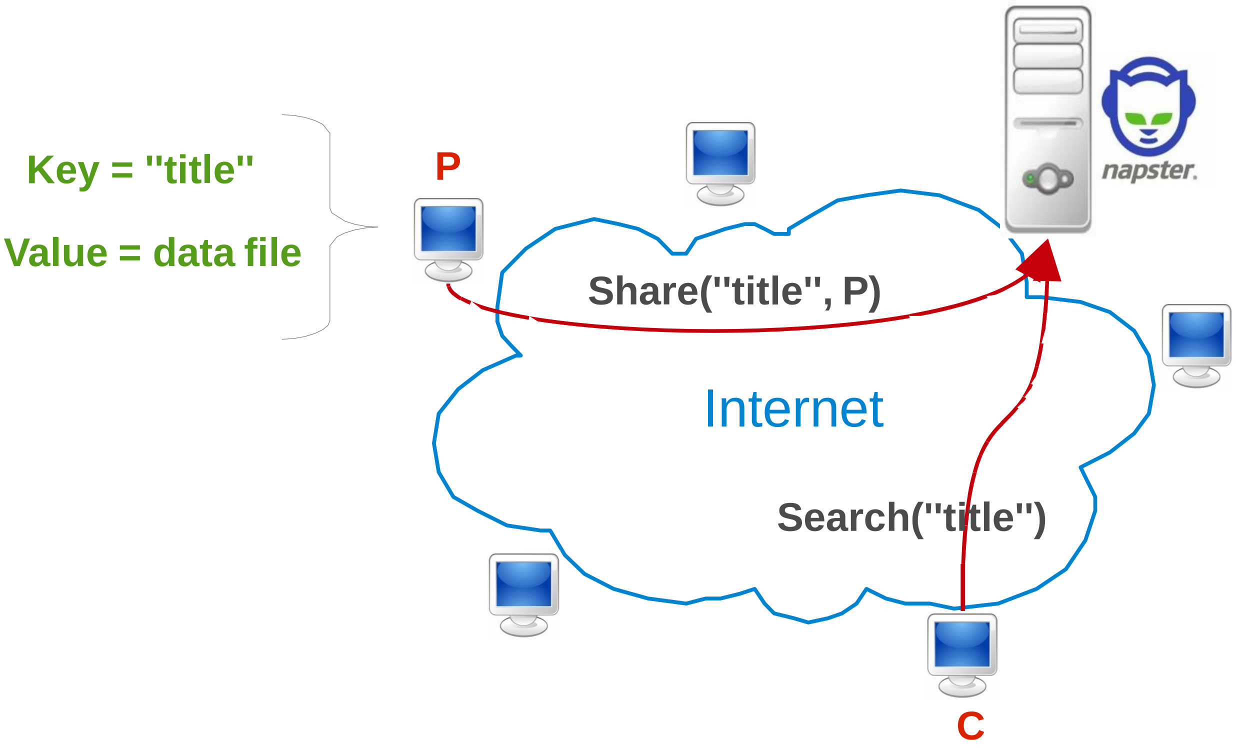


FamilyGuy.avi → {castor.sics.se}
Britney.mp3 → {rakhsh.sics.se}
GaGa.mp3 → {castor.sics.se}
Ubuntu.iso → {rakhsh.sics.se, x.kth.se}

1 Basic Operations in Napster

- Join
  - Connect to the central server (Napster)
- Share(Publish/Insert)
  - Inform the server about what you have
- Leave/Fail
  - Simply disconnect
  - Server detects failure, removes your data from the directory
- Search (Query)
  - Ask the central server and it returns a list of hits
- Download
  - Directly download from other nodes using hits provided by server

1 Centralized Lookup

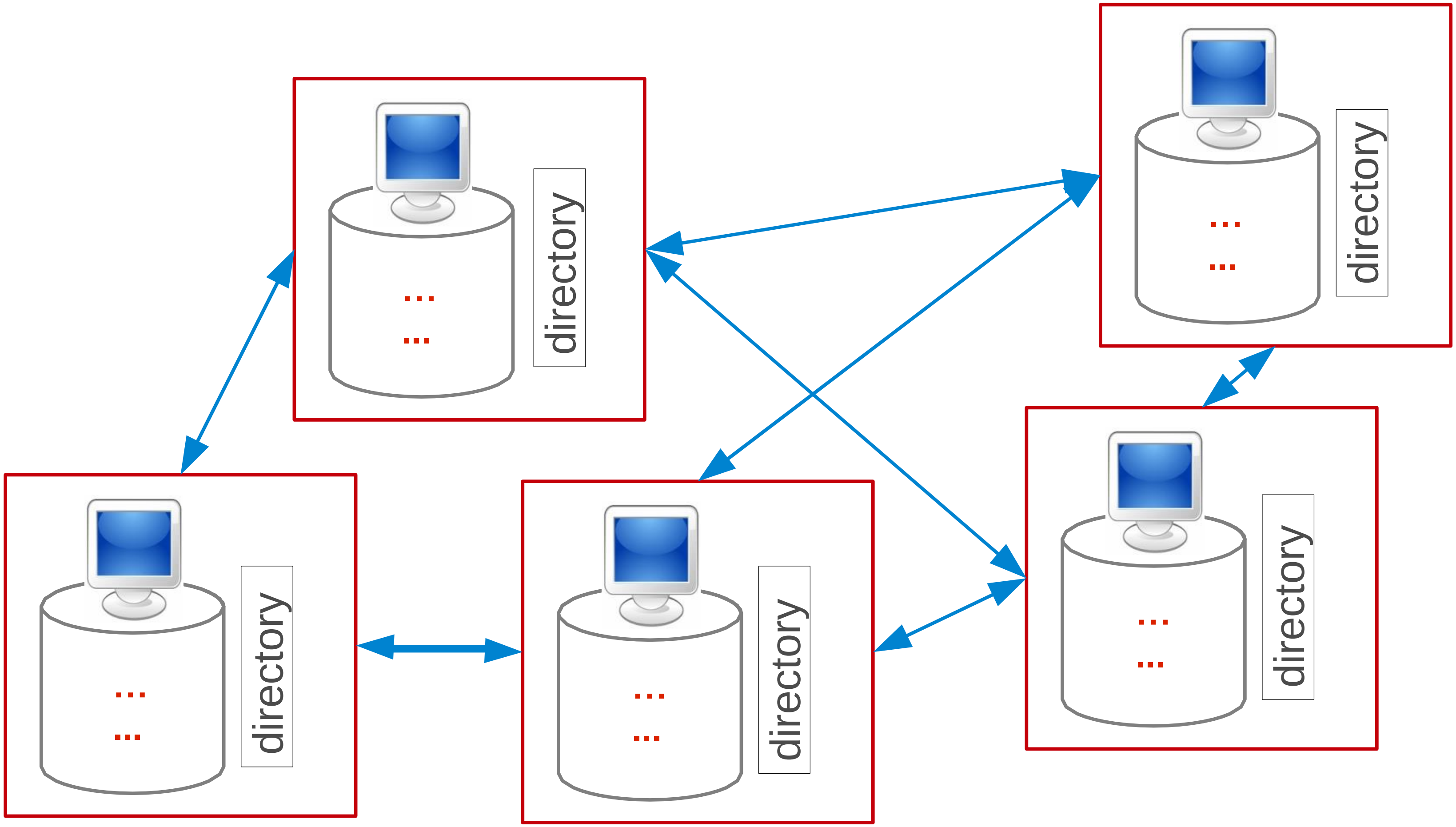


1 P2P Systems

Second Generation

1 Second Generation of P2P Systems

- Distributed Directory + Distributed Storage

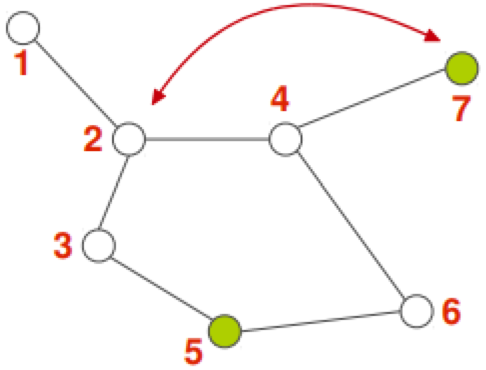


1 Gnutella Protocol Messages

- Broadcast Messages
  - Ping: initiating message ("I' m here") for overlay maintenance
  - Query: search pattern and TTL (timetolive)
- Back-Propagated Messages
  - Pong: reply to a ping, contains information about the peer
  - Query Hit: contains information about the computer that has the requested file
- Node-to-Node Messages
  - GET: return the requested file
  - PUSH: push the file to the requester node

1 Gnutella Search Mechanism

- Node 2 initiates search for file A
- Sends message to all neighbours
- Neighbours forward message
- Nodes that have file A initiate a reply message
- Query reply message is back propagated
- Nodes 2 directly connects to node 7 and downloads file A



Third Generation

1 Distributed Hash Tables(DHT)

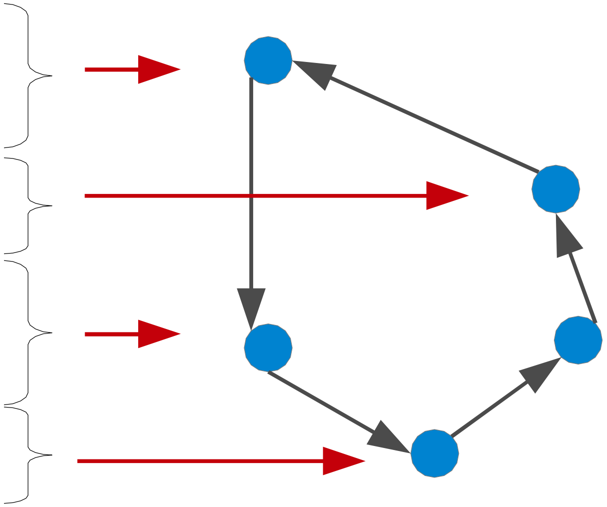
- An ordinary hashtable, which is …

Key	Value
Fatemeh	Stockholm
Ali	California
Tallat	Islamabad
Cosmin	Bucharest
Seif	Stockholm
Amir	Tehran

1 Distributed Hash Tables(DHT)

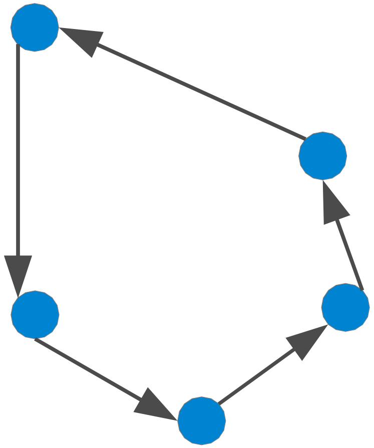
- An ordinary hashtable, which is distributed.

Key	Value
Fatemeh	Stockholm
Ali	California
Tallat	Islamabad
Cosmin	Bucharest
Seif	Stockholm
Amir	Tehran



1 Distributed Hash Tables(DHT)

- put(key,value), get(key) interface.
- The neighbors of a node are well defined and not randomly chosen.
- Values are no longer stored at their owners, instead the network chooses at which node a data item will be stored.
- Every node provides a lookup operation.
- Nodes keep routing pointers
  - If item not found, route to another node

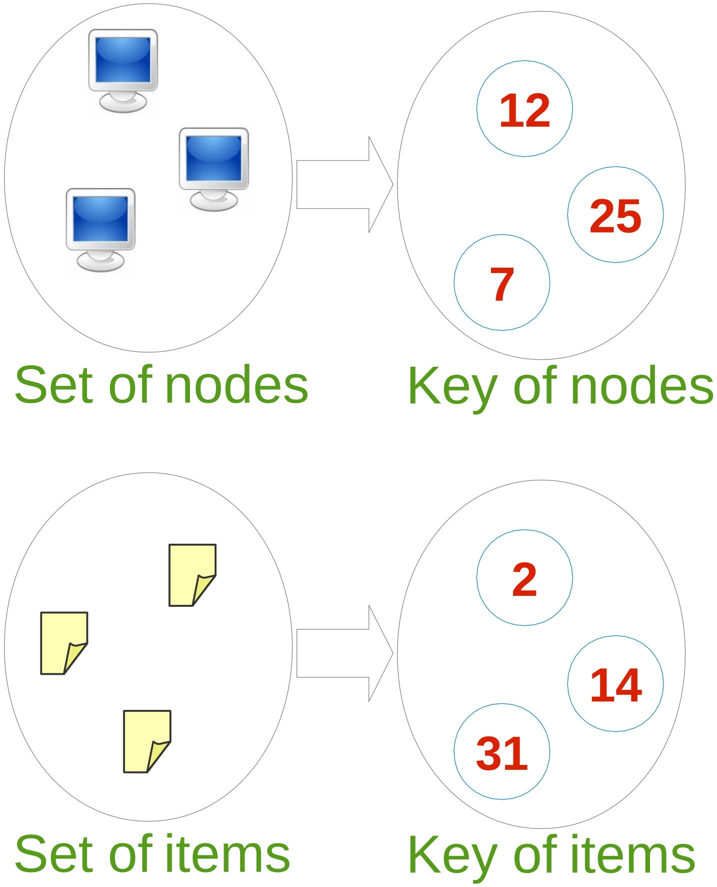


1 The Key Idea in DHTs

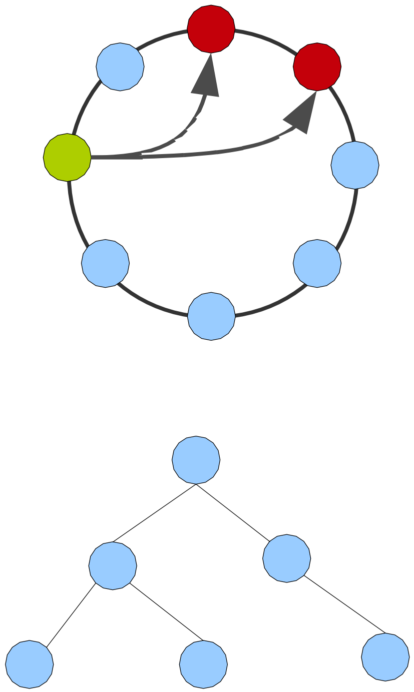
- 1st and 2nd Generation:
  - Each data item is stored in the machine of its creator/downloader
- 3rd Generation (DHTs):
  - The ID of a data item determines the machine on which it is going to be stored

1 Distributed Hash Tables (DHT)

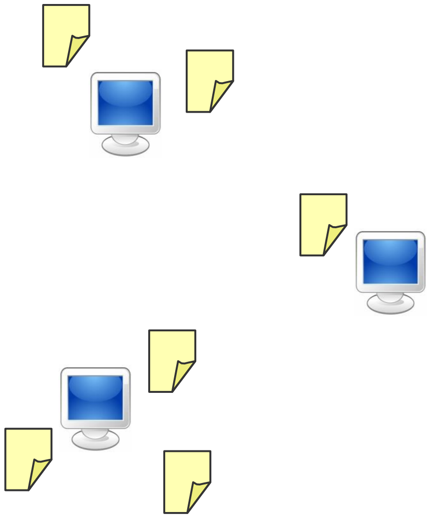
1. Decides on common key space for nodes and values



2. Connects the nodes smartly



3. Make a strategy for assigning items to nodes



1 Consistent Hashing using a Ring

- Identifier space of size 16, [0, 15].

rakhsh.sics.se

H(rakhsh.sics.se)=12

castor.sics.se

H(castor.sics.se)=3

x.kth.se

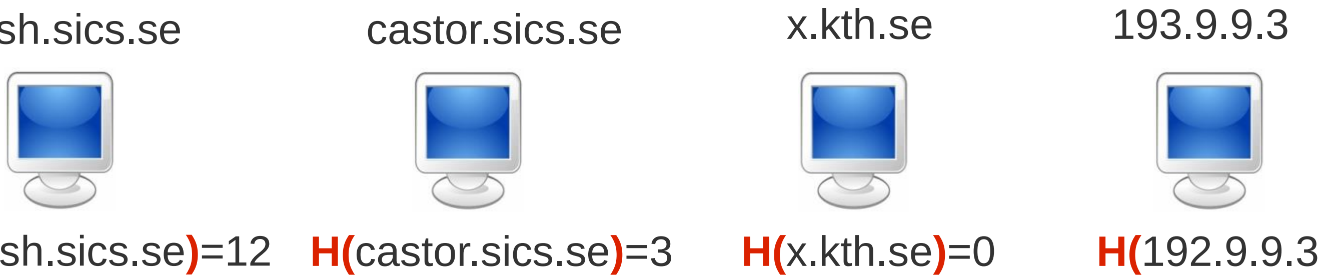
H(x.kth.se)=0

193.9.9.3

H(192.9.9.3)=7

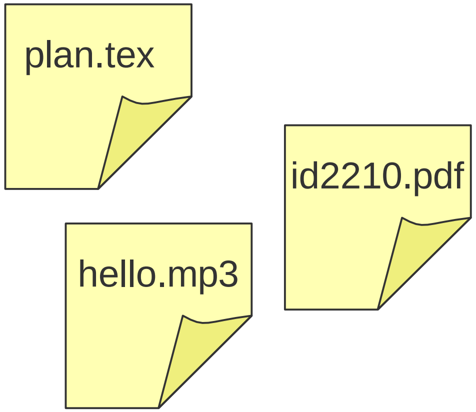
1 Consistent Hashing using a Ring

- Identifier space of size 16, [0, 15].



rakhsh.sics.se castor.sics.se x.kth.se 193.9.9.3

H(rakhsh.sics.se)=12 H(castor.sics.se)=3 H(x.kth.se)=0 H(192.9.9.3)=7



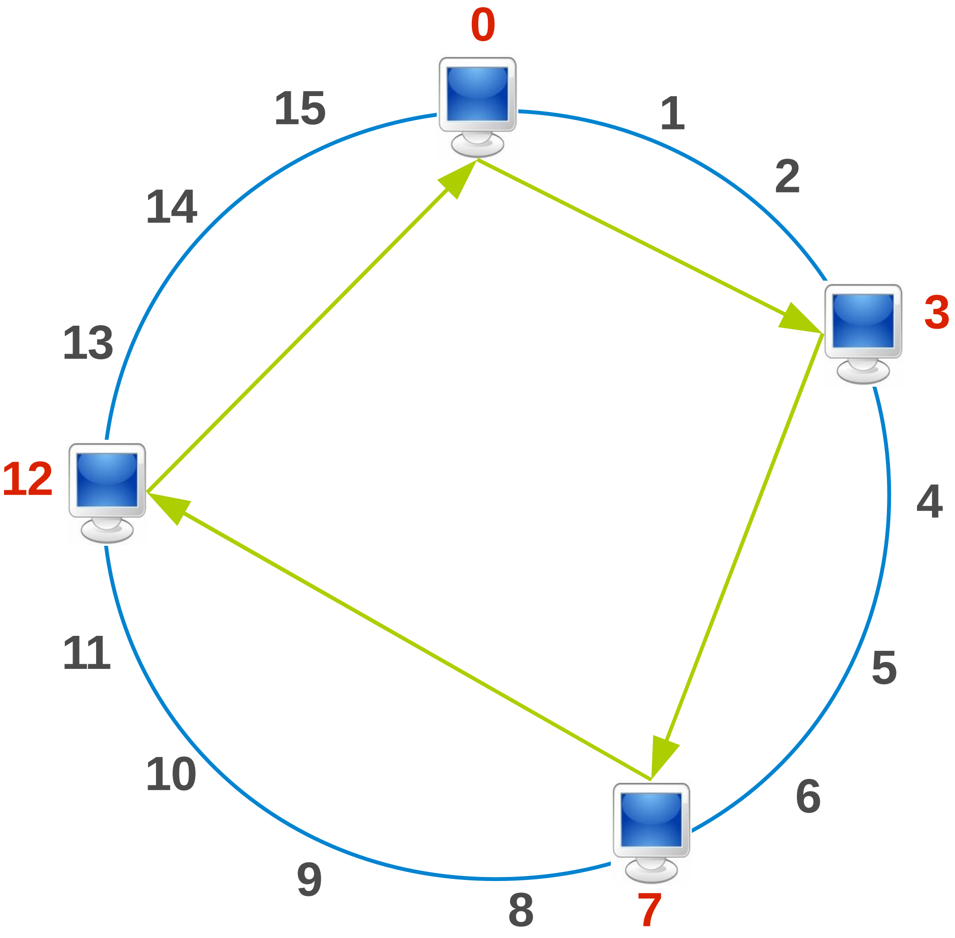
H(plan.tex)=2

H(id2210.pdf)=12

H(hello.mp3)=14

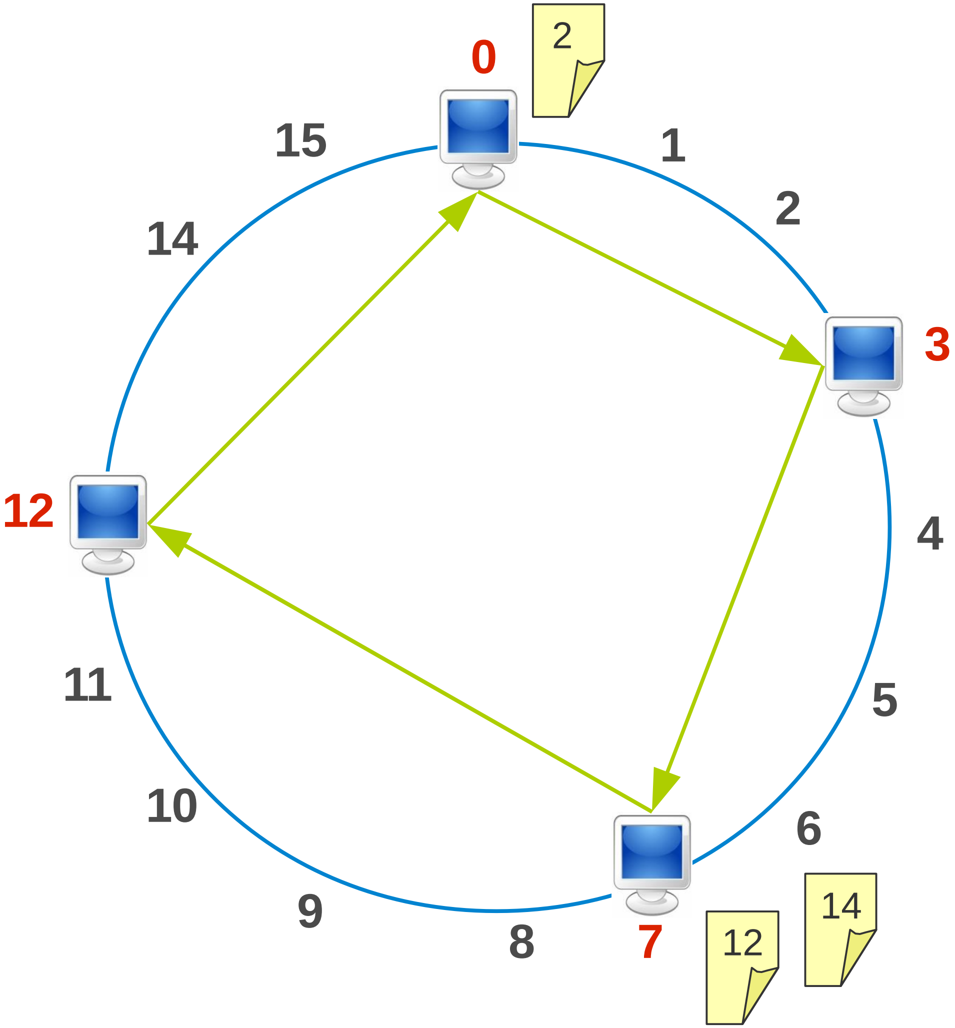
1 Consistent Hashing using a Ring

- Assume the ID space is [0, 15], i.e. a maximum of 16 nodes.
- We treat this range as a circular id space.
- succ(x): is the first node on the ring with id greater than or equal x, where x is the id of a document or node.



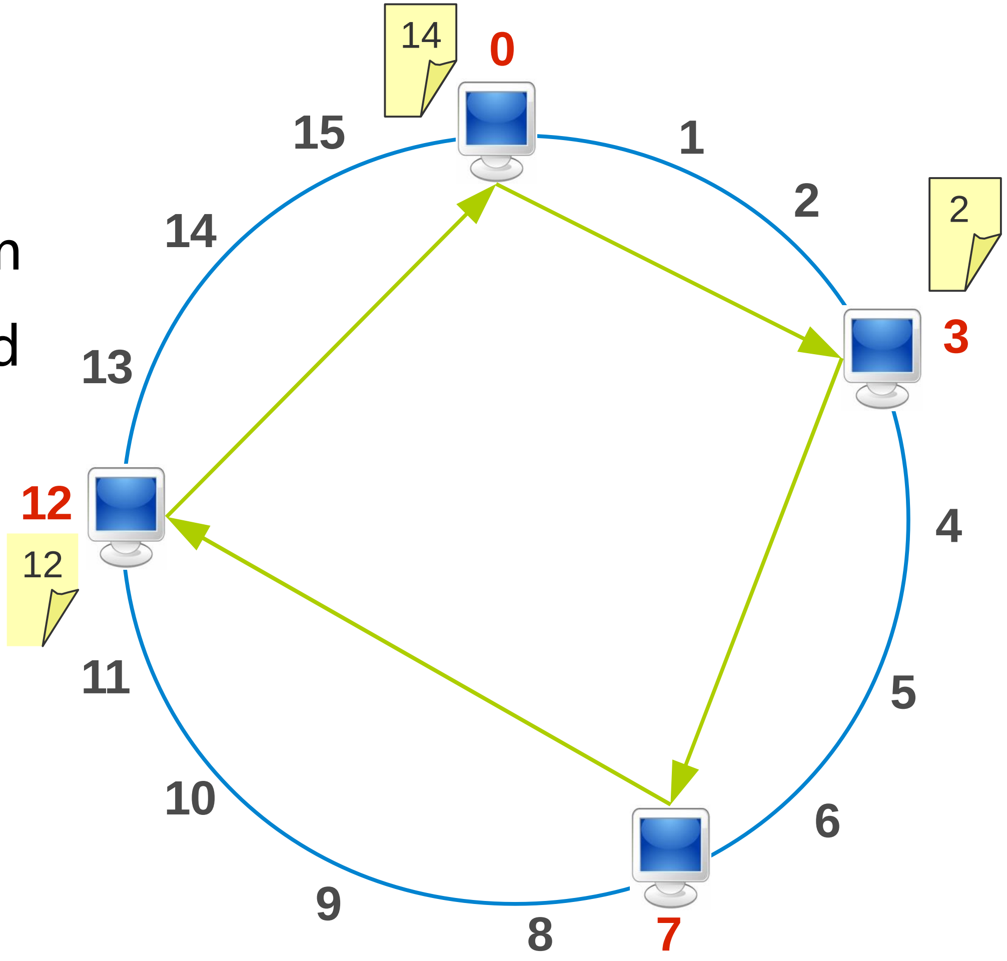
Consistent Hashing using a Ring

- Initially, node 0 stored item 2 and node 7 stored items 12 and 14.
- The policy is: An item with ID x, would be stored at the node with id succ(x).



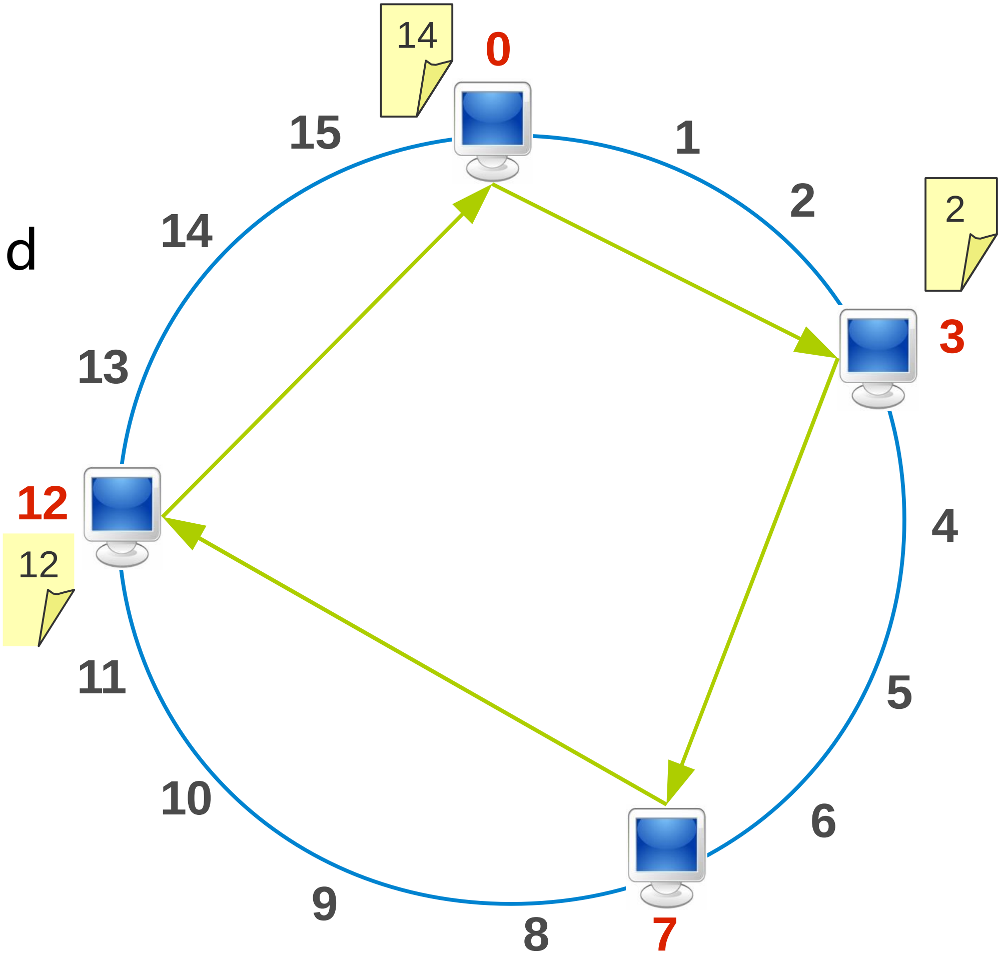
1 Consistent Hashing using a Ring

- The policy is: An item with ID x, would be stored at the node with id succ(x).
- So, node 0 gets to store item 14, node 3 to store item, and node 12 to store item 12.
- But how can we do this?



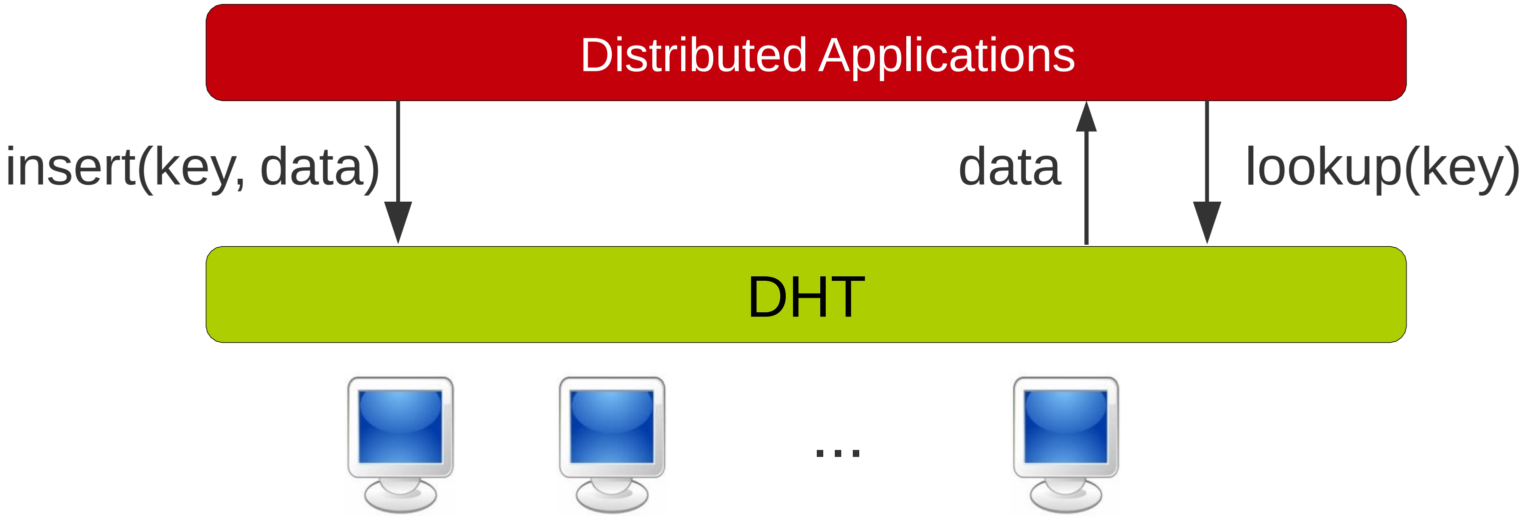
1 Consistent Hashing using a Ring

- **But how can we do this?**
- If the successor pointers are already there, the two operations, get and put would be simply done by following them sequentially.
- From any node, you can do: put(hash(item), item)
- From any node, you can do: get(hash(item))



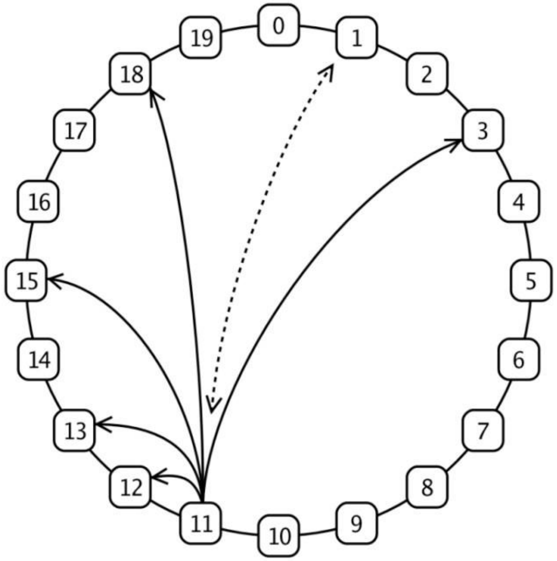
1 Distributed Hash Tables(DHT)

- Nodes are the hash buckets
- Key identifies data uniquely
- DHT balances keys and data across nodes
- DHT replicates, caches, routes lookups, etc



1 Category of P2P network structure

- **Unstructured networks:** Random connections between peers
- **Structured networks:** Nodes assigned to specific positions in a structure
  - Positions have a meaning, e.g. node at position 17 stores content with hash values starting with “17”
  - Nodes with certain responsibilities can be found efficiently



Structured network: Chord

Contents

1 Basics of P2P Networks

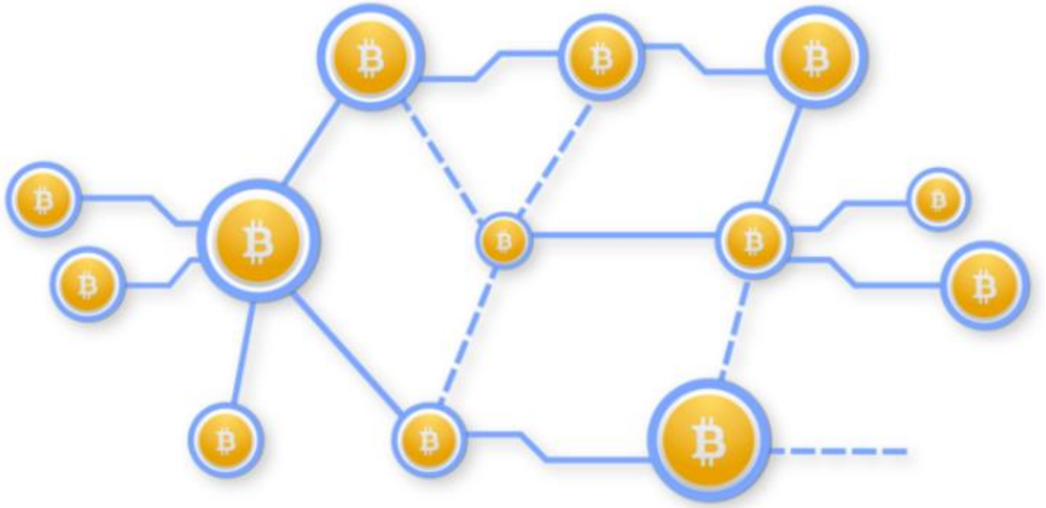
2 Bitcoin P2P Network

3 Ethereum P2P Network

4 Summary

2 The Bitcoin P2P Network

- Bitcoin
  - pure Peer-to-Peer principle
  - Bitcoin clients have to agree on account balances
  - Bitcoin clients need information about transactions
  - Goal: Consistent view in the whole network

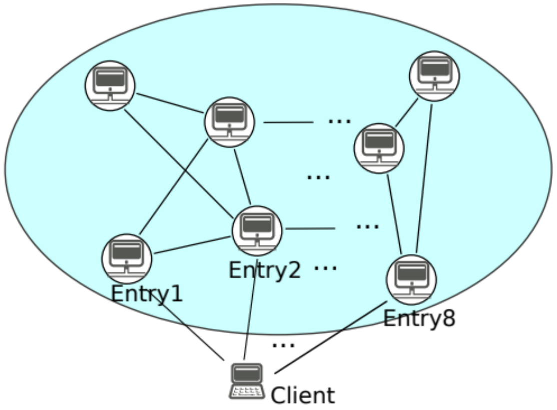


2 P2P Network Structure of Bitcoin

- Bitcoin: Unstructured Peer-to-Peer network
  - No overhead for maintaining the structure
  - Simple implementation
    - Main advantage of structured networks – quick finding of specific information
    - Not applicable to Bitcoin: All nodes need (more or less) complete information

2 Joining the Peer-to-Peer network

- First step: Finding some other peers
  - Requires “cheating”: Finding peers without some central system is difficult
  - Bitcoin’s approaches
    - Use pre-configured IP addresses
    - Get IP addresses from an IRC channel (no longer used in the default setting)
    - Get IP addresses via the Domain Name System (DNS servers run by volunteers)



2 Connections in the Bitcoin network

- Node knows some IP addresses of other nodes
  - Node connects to a certain number (default: 8) of these nodes
  - Node accepts incoming connections beyond that limit (not always possible, e.g. due to firewalls)
  - On average: About 30 connections per node that accepts incoming connections
- Inactive nodes deleted from lists after timeout (several hours)

2 Bitcoin transactions and blockchain

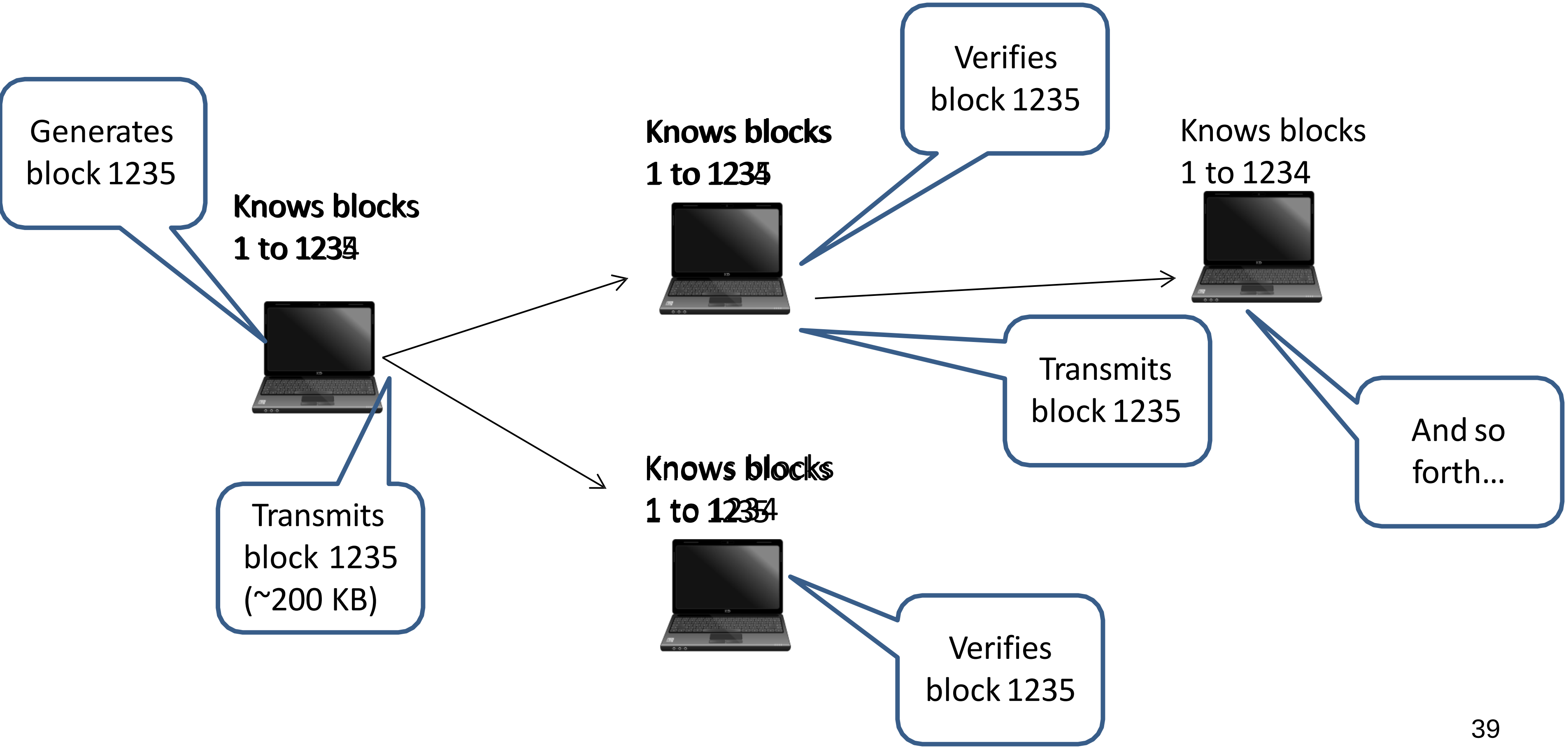
- Individual transaction from A to B
  - A signs the transaction using the private key of his address
  - A broadcasts the transaction to the whole Bitcoin network
- Confirmation of transactions
  - Nodes (miners) collect transactions in a "**block**"
  - Miners append block to blockchain and compute a proof of work
  - Successful miner broadcasts the block to the whole Bitcoin network

2 Broadcasting

- Sender informs all connected Bitcoin nodes about availability of a new transaction / new block
  - Invite message
- On receipt of an invite message
  - Node requests the transaction / block if it does not know it
  - Node verifies the transaction / block based on local blockchain copy
  - Node informs all connected Bitcoin nodes about availability of a new transaction / new block

2 Consistent view?

- Goal of the Peer-to-Peer network: Consistent view
  - Network becomes inconsistent once a new block is generated



Generates block 1235

Knows blocks 1 to 1234

Knows blocks 1 to 1235

Transmits block 1235 (~200 KB)

Knows blocks 1 to 1234

Knows blocks 1 to 1235

Verifies block 1235

Knows blocks 1 to 1234

Knows blocks 1 to 1235

Verifies block 1235

Knows blocks 1 to 1234

Transmits block 1235

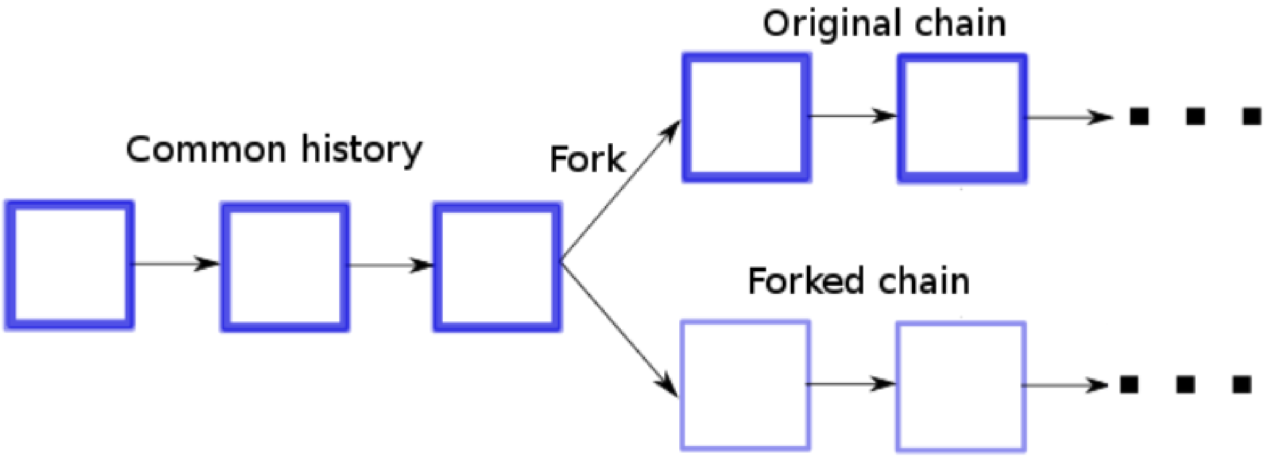
And so forth…

2 Information propagation

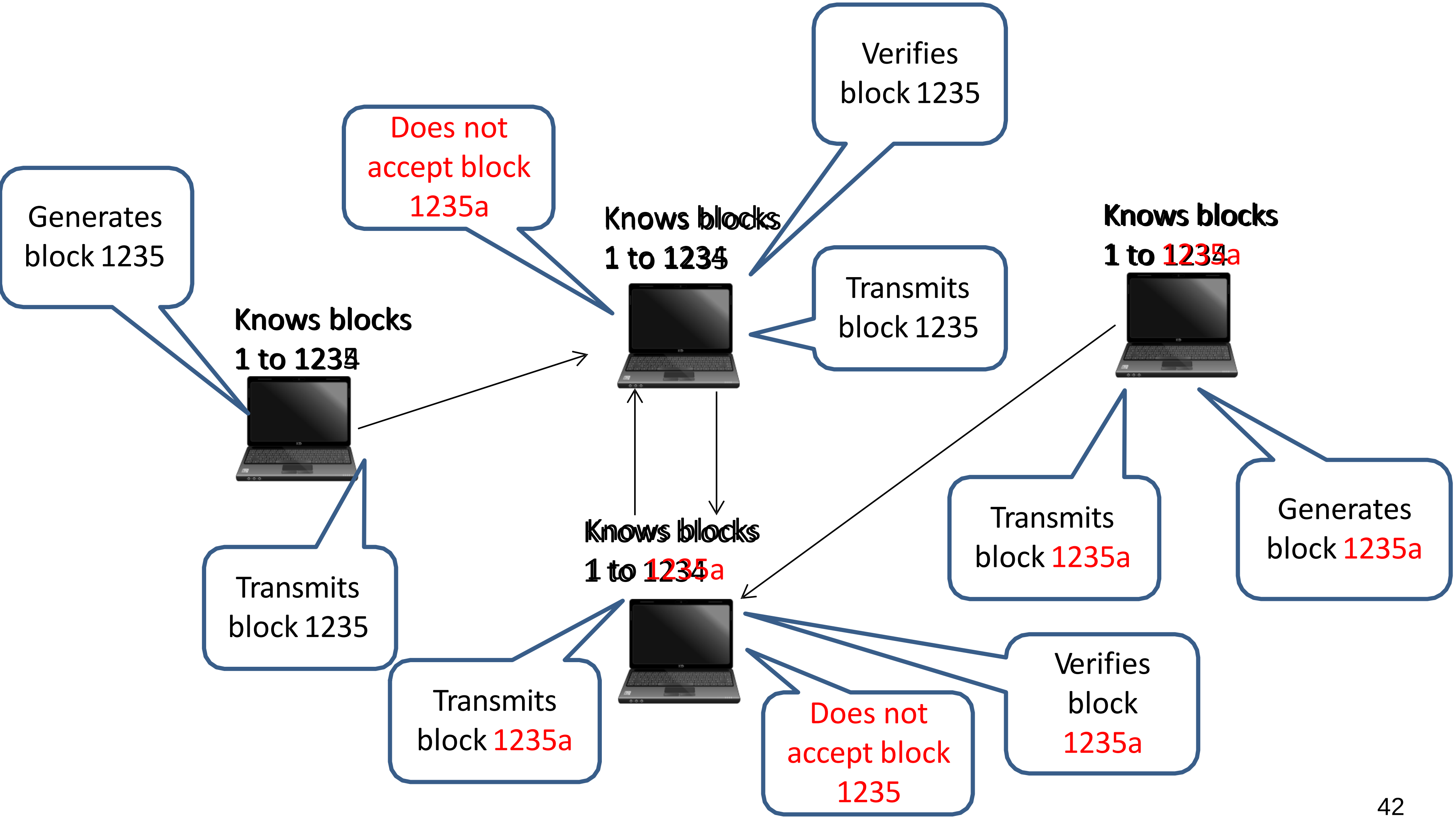
- Investigation by Decker and Wattenhofer (Proc. IEEE P2P ‘13)
  - Connection to a large number of nodes, observation of information propagation
  - Average time till a node receives a new block: 12.6 seconds
  - Long tail: 5% of nodes do not have the new block after 40 seconds

2 Information propagation

- Problem of propagation time: Other miner may find a new block within that time
  - **blockchain fork:** two inconsistent versions of the blockchain
  - Decker and Wattenhofer observe 169 blockchain forks during a period of 10,000 generated blocks



2 Inconsistency example



Generates block 1235

Knows blocks 1 to 1234 / 1 to 1235

Transmits block 1235

Does not accept block 1235a

Knows blocks 1 to 1234 / 1 to 1235

Verifies block 1235

Transmits block 1235

Knows blocks 1 to 1234 / 1 to 1235a

Transmits block 1235a

Generates block 1235a

Knows blocks 1 to 1234 / 1 to 1235a

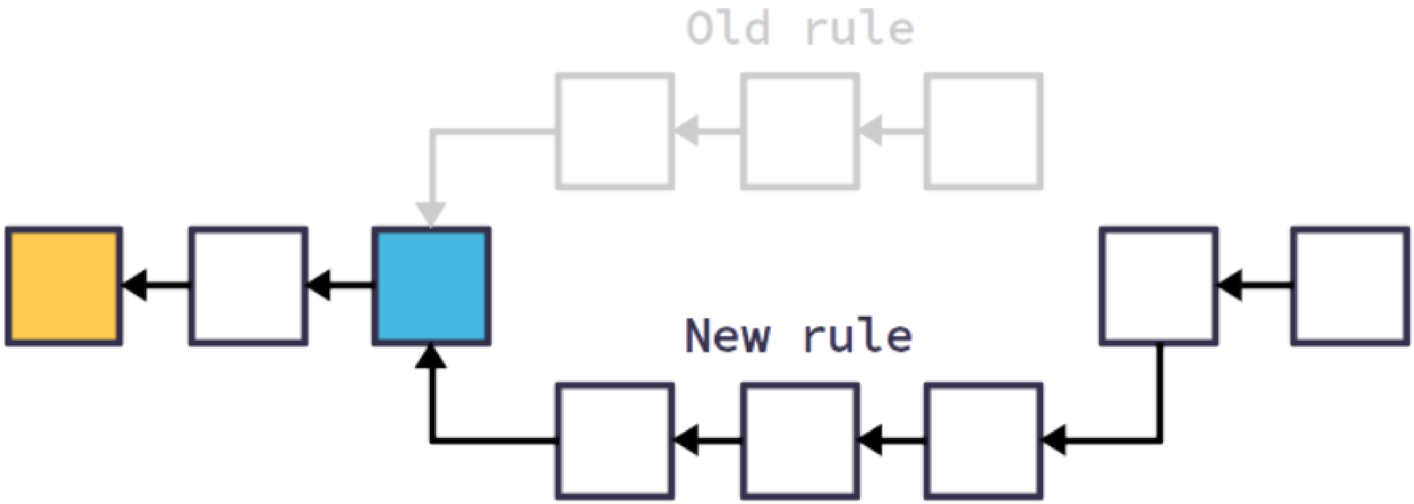
Transmits block 1235a

Does not accept block 1235

Verifies block 1235a

2 Dealing with inconsistency

- Each miner continues with one version of the blockchain
  - First newly generated block leads to longest chain
  - All nodes switch to longest chain once that block has been received
- Transaction only present in the shorter version: Not lost, but integrated into the next block



2 Privacy issues

- Bitcoin privacy research concerning the transaction graph
  - linking different Bitcoin addresses of a user
- Concerning the Peer-to-Peer network
  - Figure out origin (IP address) of a transaction by finding the first node that broadcasts it
  - Try to get connections to as many nodes as possible
    - Connect to nodes in the network that accept incoming connections
    - Join the network under many fake identities to get many other nodes to connect to you

Contents

1. Basics of P2P Networks
2. Bitcoin P2P Network
3. **Ethereum P2P Network**
4. Summary

3 The Ethereum P2P Network

- Ethereum
  - Ethereum is an open-source, public, blockchain-based distributed computing platform and operating system featuring smart contract (scripting) functionality.
- The Ethereum's P2P network is based on a structured networks called Kademlia.

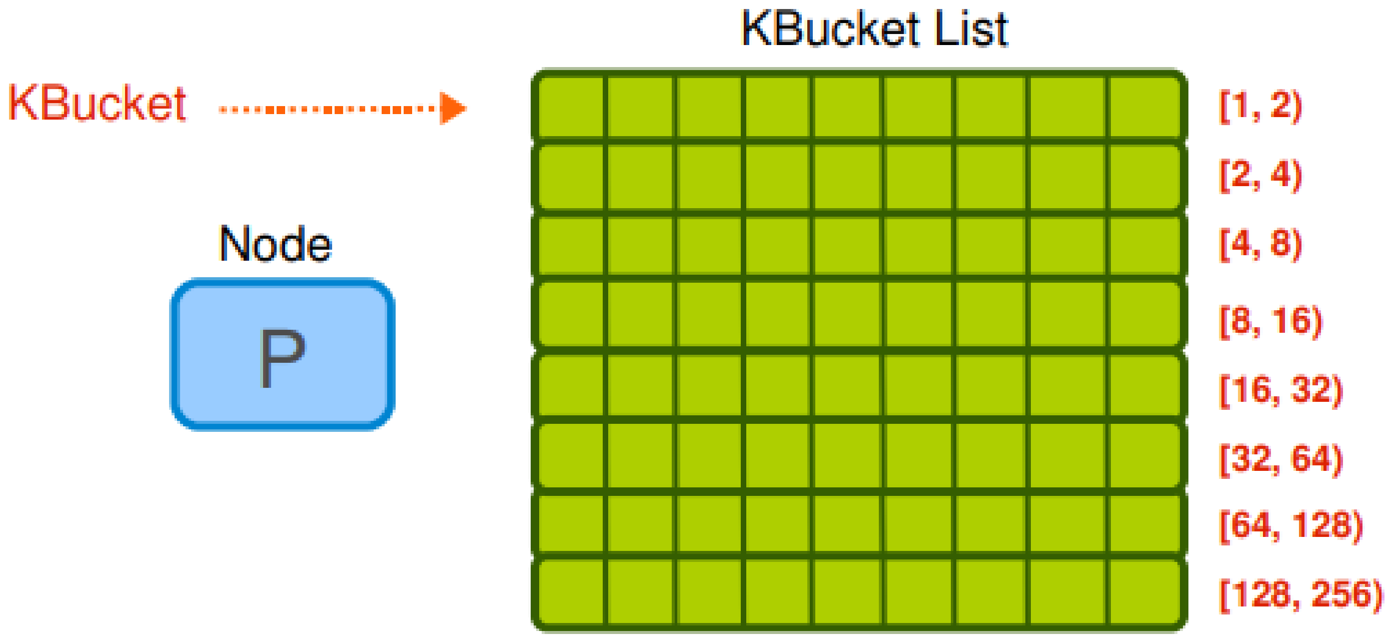
3 Kademlia Network

- Each object is stored at the k closest nodes to the object's ID.
- Distance between id1 and id2: d(id1, id2) = id1 XOR id2
  - If ID space is 3 bits:

$$
\begin{aligned}
d(1, 4) &= d(001_2, 100_2) \\
&= 001_2 \text{ XOR } 100_2 \\
&= 101_2 \\
&= 5
\end{aligned}
$$

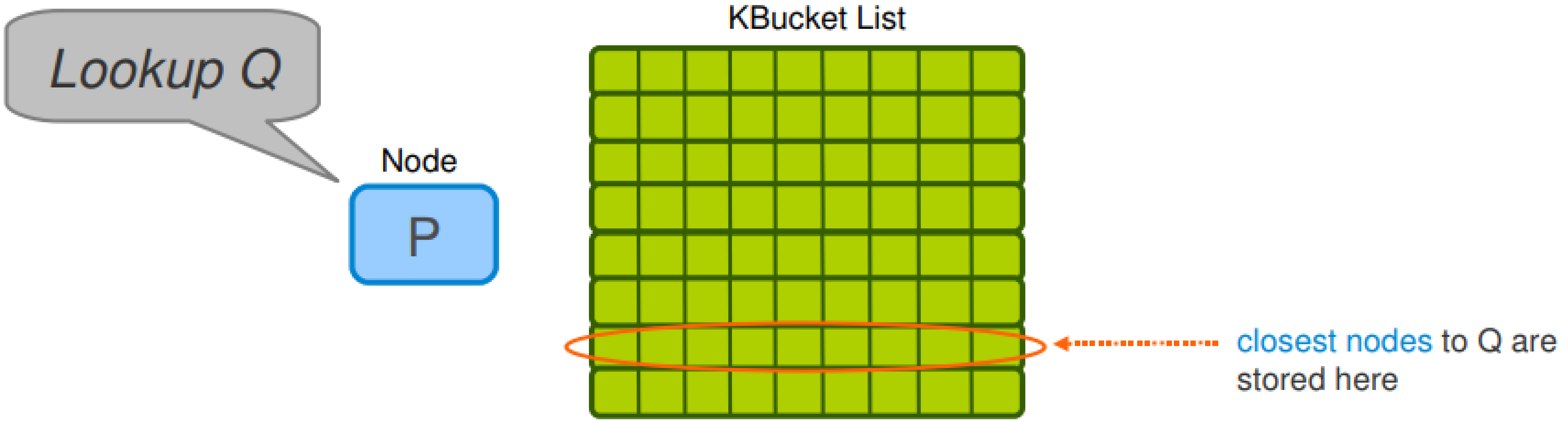
3 Kademlia Network

- Kbucket: each node keeps a list of information for nodes of distance between 2^i and 2^{i+1}.

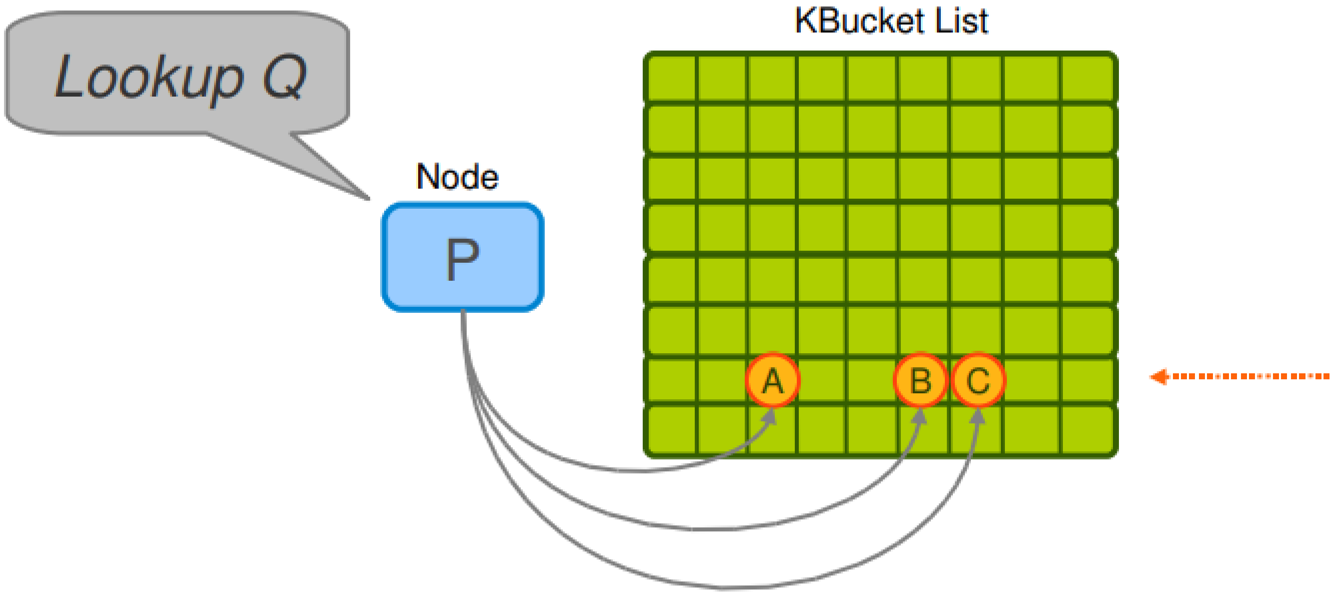


3 Kademlia Network

- Closest nodes in ID space



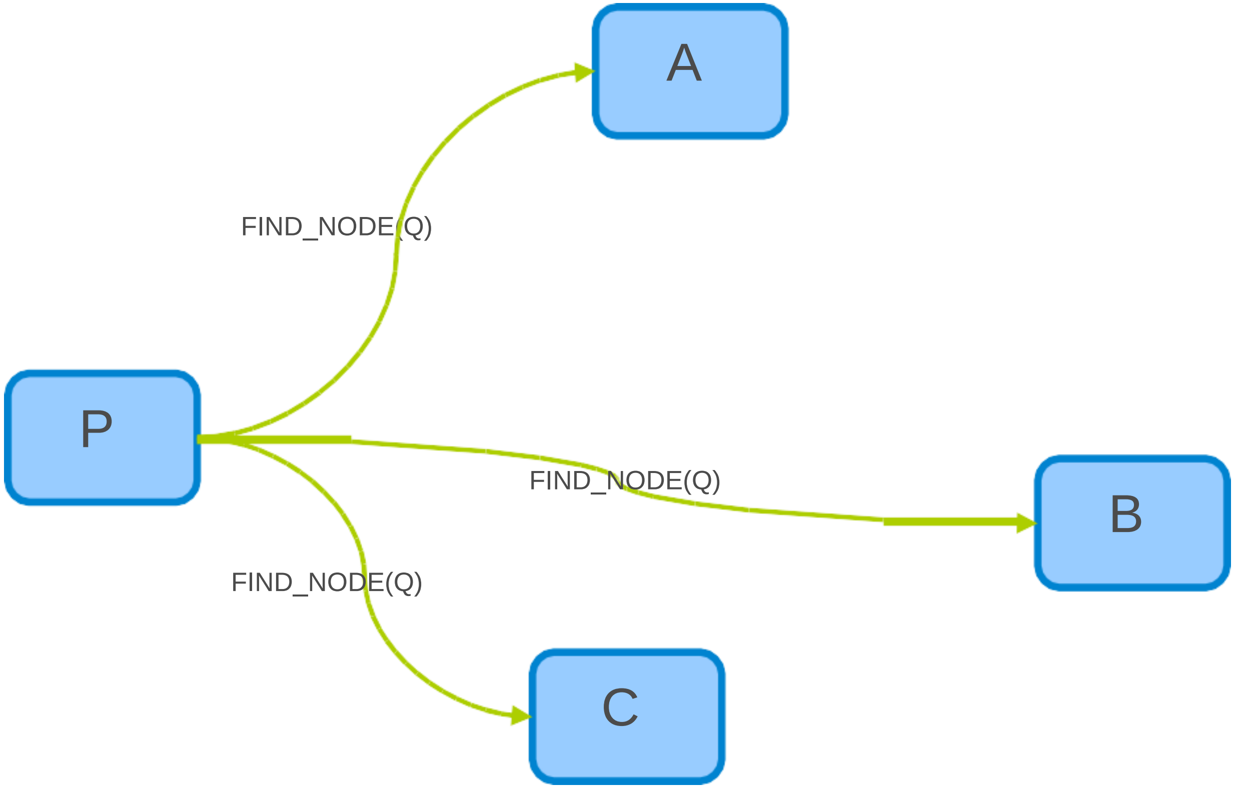
3 Kademlia Network



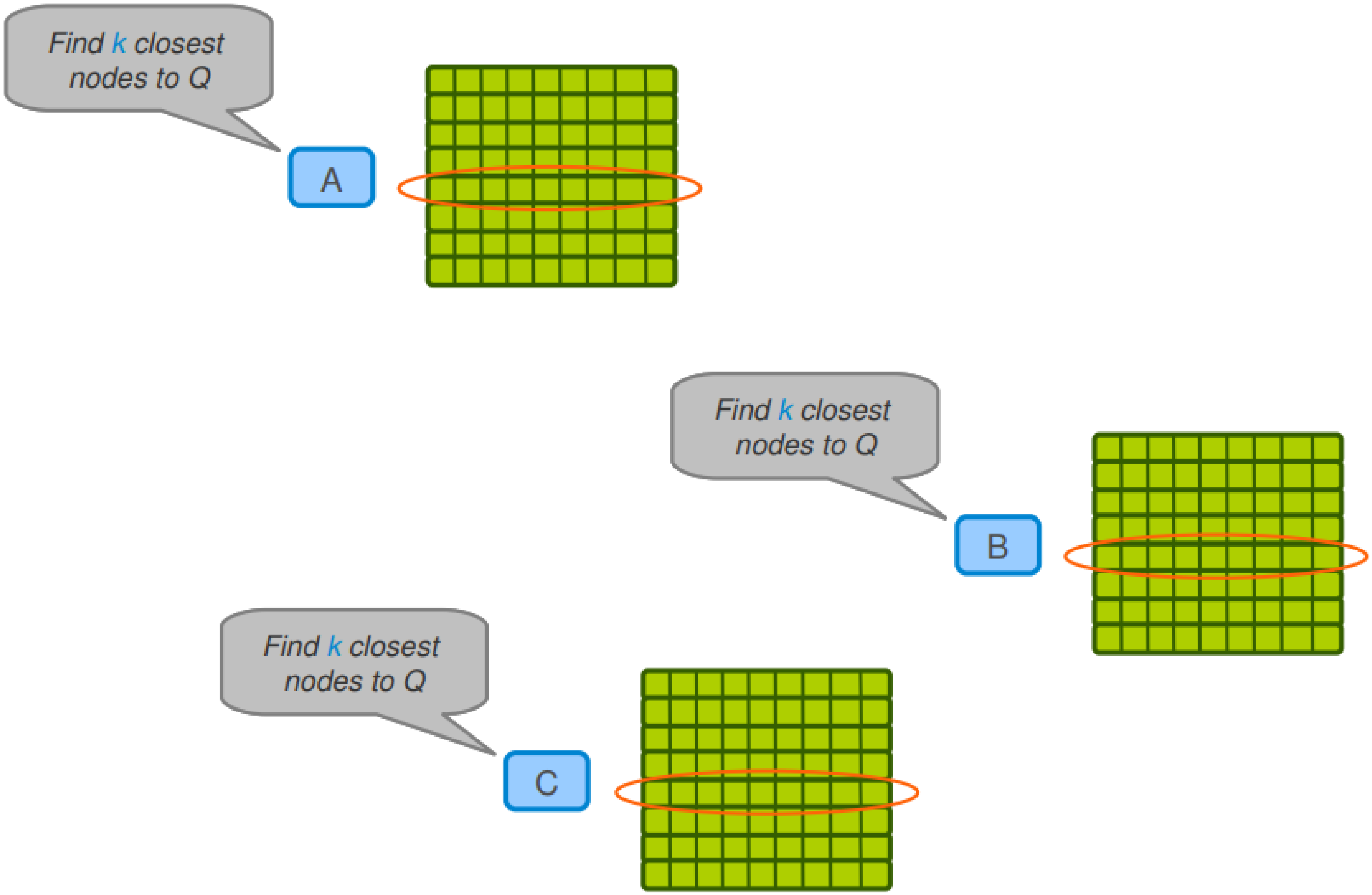
closest nodes to Q are stored here

... and select α nodes from the appropriate kbucket

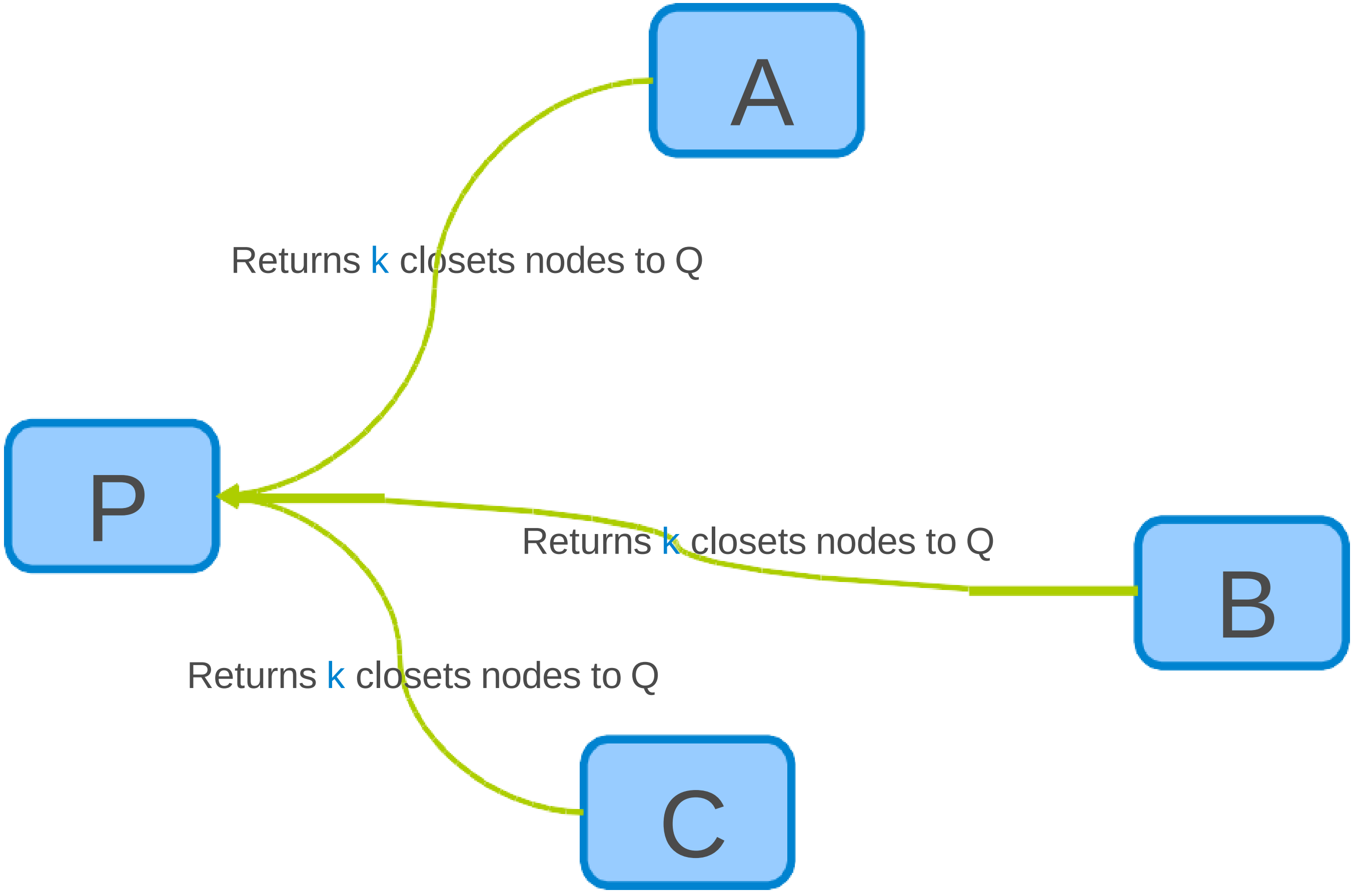
3 Kademlia Network



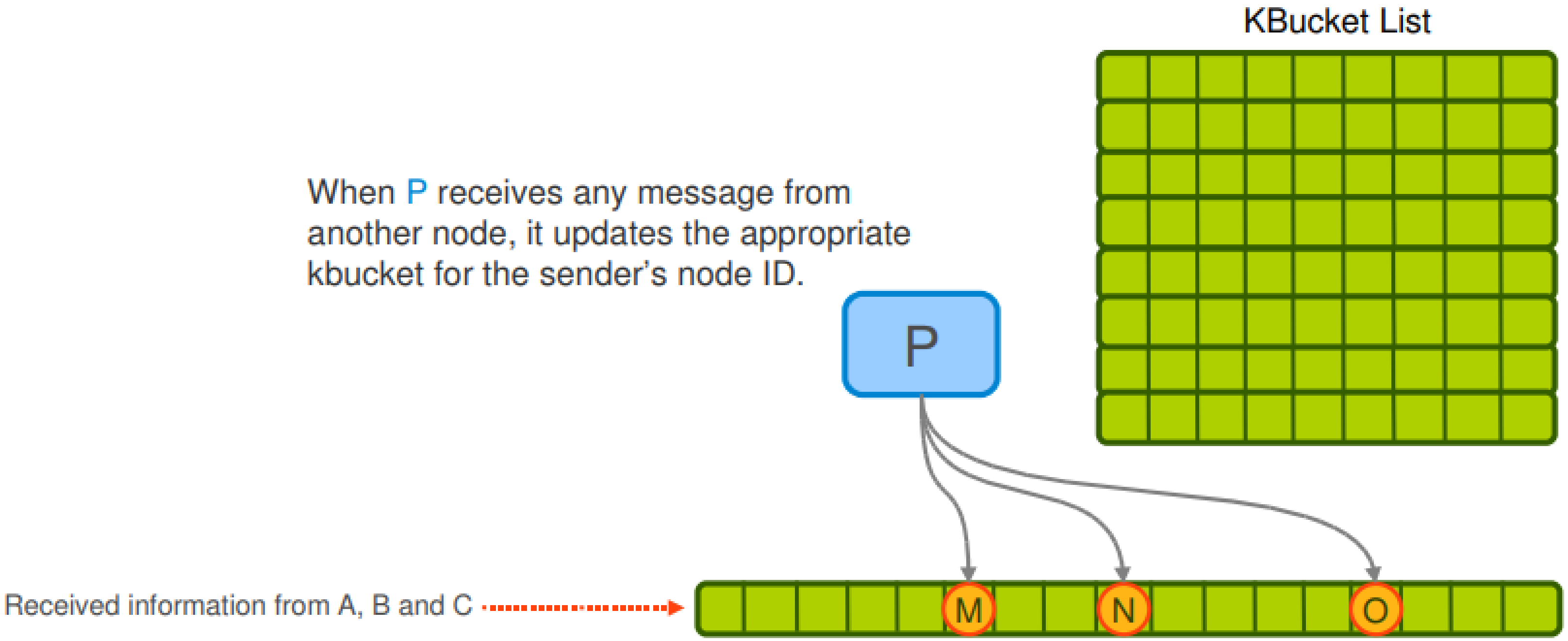
3 Kademlia Network


Find k closest nodes to Q
A
Find k closest nodes to Q
B
Find k closest nodes to Q
C

3 Kademlia Network


A
Returns k closets nodes to Q
P
Returns k closets nodes to Q
B
Returns k closets nodes to Q
C

3 Kademlia Network

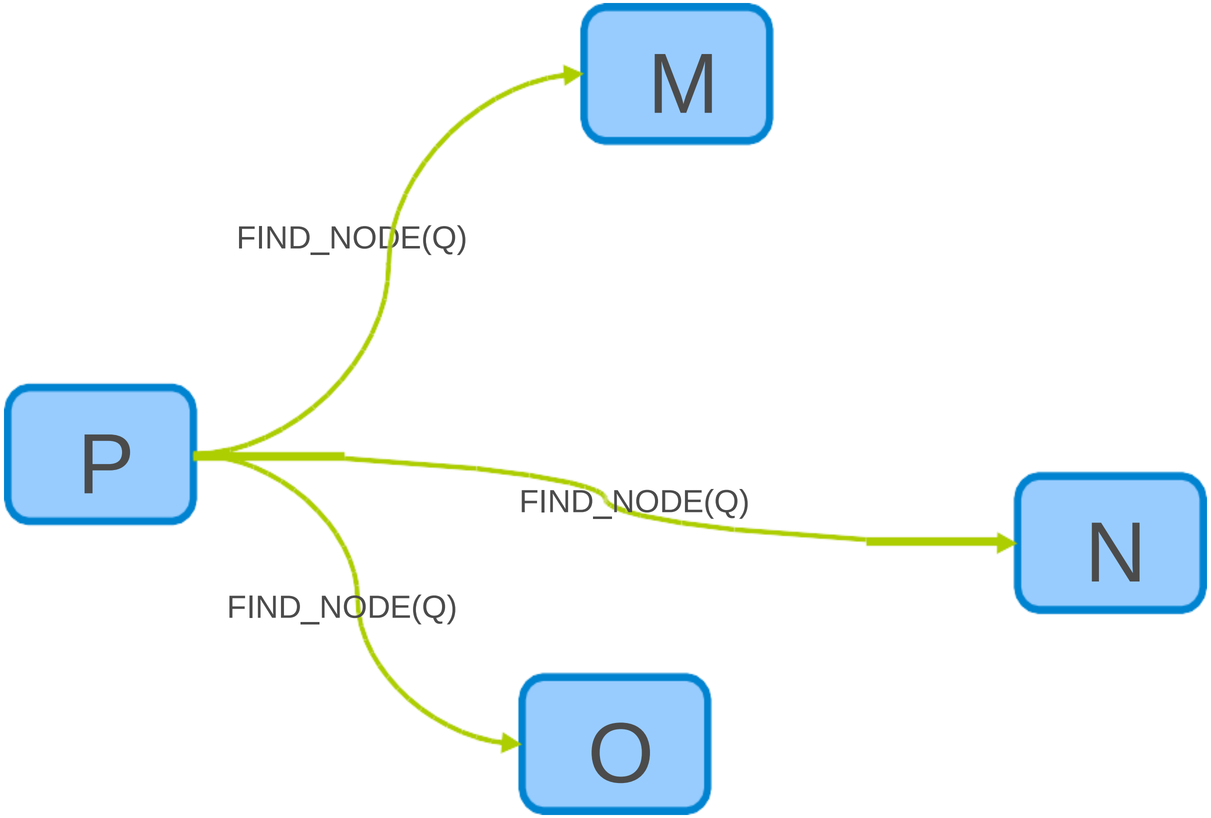


When P receives any message from another node, it updates the appropriate kbucket for the sender's node ID.

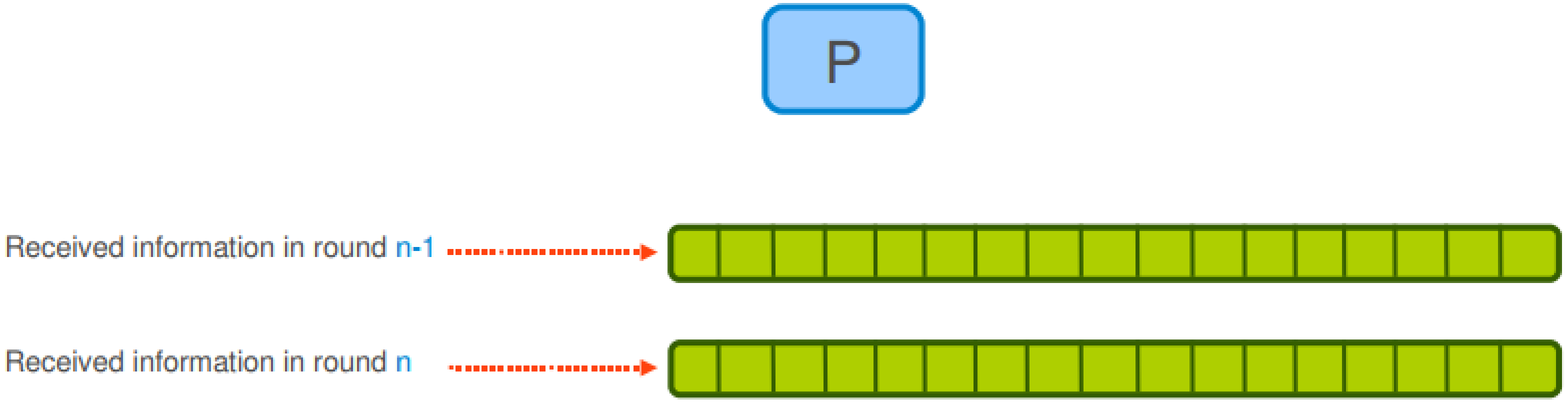
Received information from A, B and C

... again select α nodes from the received information

3 Kademlia Network



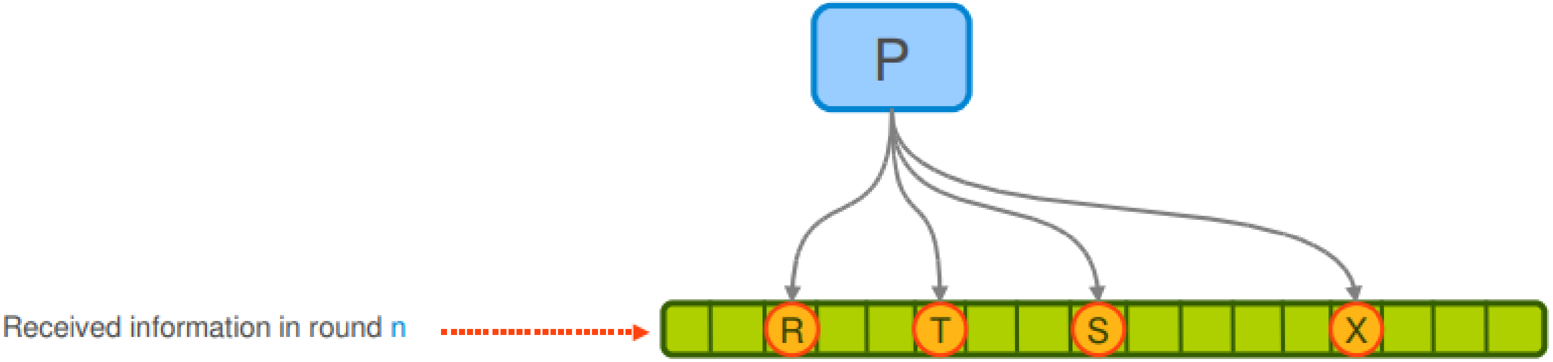
3 Kademlia Network


P
Received information in round n-1
Received information in round n

Repeats this procedure iteratively until received information in round n-1 and n are the same.

3 Kademlia Network

P resends the FIND_NODE to k closest nodes it has not already queried ...



3 Kademlia RPCs

- **PING** : Probes a node to see if it is online.
- **STORE** : Instructs a node to store a <key, value> pair.
- **FIND_NODE** : Returns information for the k nodes it knows about closest to the target ID.
- **FIND_VALUE**
  - Like FIND_NODE, …
  - But if the recipient has stored they <key, value>, it just returns the stored value

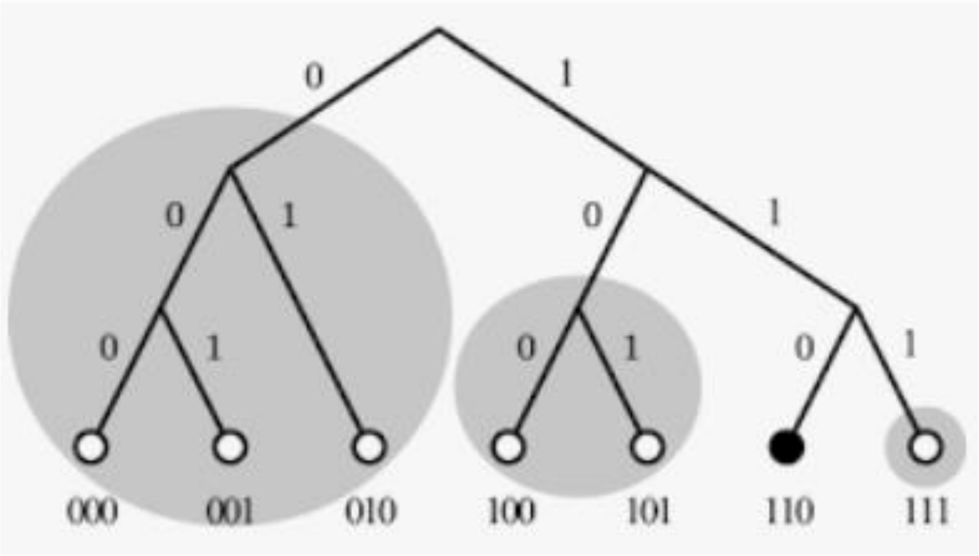
3

Maintaining Kbucket List

- When a Kademlia node receives any message from another node, it updates the appropriate kbucket for the sender’s node ID.
- If the sending node already exists in the kbucket: Moves it to the tail of the list.

3 Join

- Node P contacts to an already participating node Q.
- P inserts Q into the appropriate kbucket.
- P then performs a node lookup for its own node ID.



3 Leave and Failure

- No action!
- If a node does not respond to the PING message, remove it from the table.

Contents

1 Basics of P2P Networks

2 Bitcoin P2P Network

3 Ethereum P2P Network

4 **Summary**

4 Summary

- P2P network is one of three most important technologies of Blockchain.
- Bitcoin’s Peer-to-Peer network is simple, but works. It use unstructured network and simple broadcast scheme
- The Ethereum’s P2P network is based on a structured networks called Kademlia. It is more complex but more effective when finding nodes.

Q & A