HELSINKI UNIVERSITY OF TECHNOLOGY
Networking laboratory

Peer-to-Peer Architectures and Signaling

Nicklas Beijar
nbeijar@netlab.tkk.fi

Slides based on presentations by
Marcin Matuszewski (2005) and Juuso Lehtinen (2006)

Agenda

- **Introduction**
- P2P architectures
- Skype
- P2P-SIP
- Mobile P2P
- Summary

Introduction

- There are various definitions of peer-to-peer

  "A distributed network architecture may be called a Peer-to-Peer (P-to-P, P2P, . . .) network, if the participants share a part of their own hardware resources (processing power, storage capacity, network link capacity, printers, . . .). These shared resources are necessary to provide the service and content offered by the network (e.g. file sharing or shared workspaces for collaboration). They are accessible by other peers directly, without passing intermediary entities. The participants of such a network are thus resource (service and content) providers as well as resource (service and content) requesters (servent-concept)." (Schollmeier, 2002)

  "A peer-to-peer (or "P2P") computer network uses diverse connectivity between participants in a network and the cumulative bandwidth of network participants rather than conventional centralized resources where a relatively low number of servers provide the core value to a service or application. Peer-to-peer networks are typically used for connecting nodes via largely ad hoc connections. Such networks are useful for many purposes. Sharing content files (see file sharing) containing audio, video, data or anything in digital format is very common, and realtime data, such as telephony traffic, is also passed using P2P technology." (Wikipedia, ref. 3.4.2008)

Introduction

- Generally in peer-to-peer communication
  - Participants share resources
  - Each communicating node (*peer*) has both server and client capabilities (*servent* concept)
  - Applications directly connect with each other
  - Users can search for resources (files, services, users)
  - The capacity of the network edge is utilized

Peer-to-Peer popularity

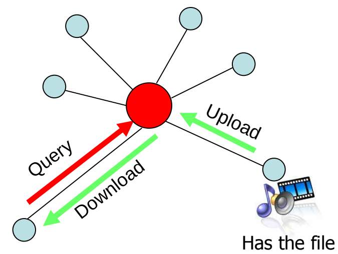
- P2P accounts for 60–80% of all Internet traffic
- File-sharing applications are the most popular form of P2P – at least traffic wise – e.g. BitTorrent, Kazaa, Direct Connect
- Peer-to-peer paradigm has many uses
  - File-sharing
  - Internet telephony (e.g. Skype, P2P-SIP)
  - Distributed computing (e.g. Seti@home)
  - Collaboration
  - Games
  - …
- Mobile P2P

Agenda

- Introduction
- **P2P architectures**
- Skype
- P2P-SIP
- Mobile P2P
- Summary

Traditional Client-Server Architecture

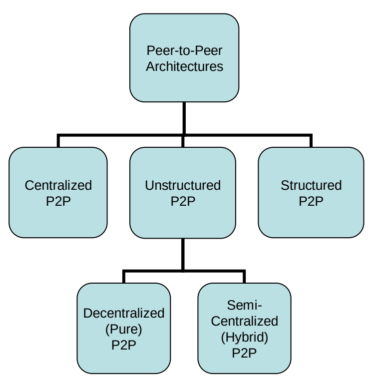
- One high-performance server (or cluster of servers) holds all the content in the network
  - Owner of the server has full control of the content
- Multiple clients share content via the centralized server
- No communication between clients
- Limited scalability
  - The server must store all contents
  - The server must serve all clients
- E.g. web servers, FTP servers



This is **not** peer-to-peer!

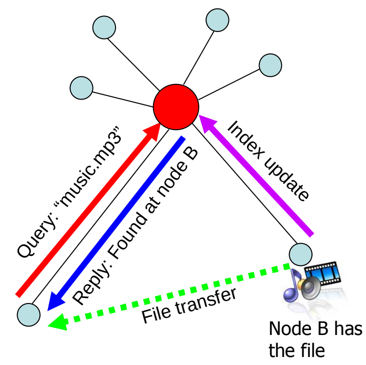
P2P architectures

Architectures can be categoritzed according to
- Distribution
  - Centralized peer-to-peer
  - Decentralized peer-to-peer (pure peer-to-peer)
  - Semi-centralized peer-to-peer (hybrid peer-to-peer)
- Structure
  - Unstructured peer-to-peer
  - Structured peer-to-peer



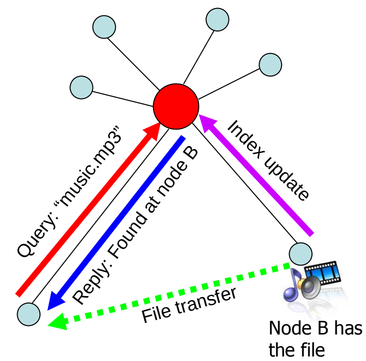
Centralized P2P architecture

- A centralized server (or a cluster of servers) stores *index information* (metadata) about the available files
  - The owner of the server has high control on the shared content
- The files are stored at the clients (not at the server)
- The clients transfer content directly without the server's involvement
  - The server is only used for content searches
- Example: Napster, BitTorrent trackers



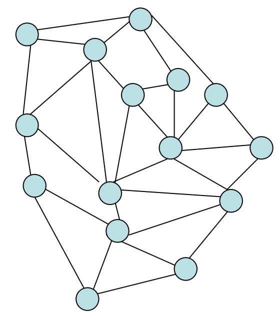
Centralized P2P architecture

- Advantages
  - Quick searches
  - Low bandwidth requirement
- Disadvantages
  - Server represents a single point of failure for the entire system
  - Can be easily attacked
  - Capacity of server (bandwidth, memory, processing power) limits scalability



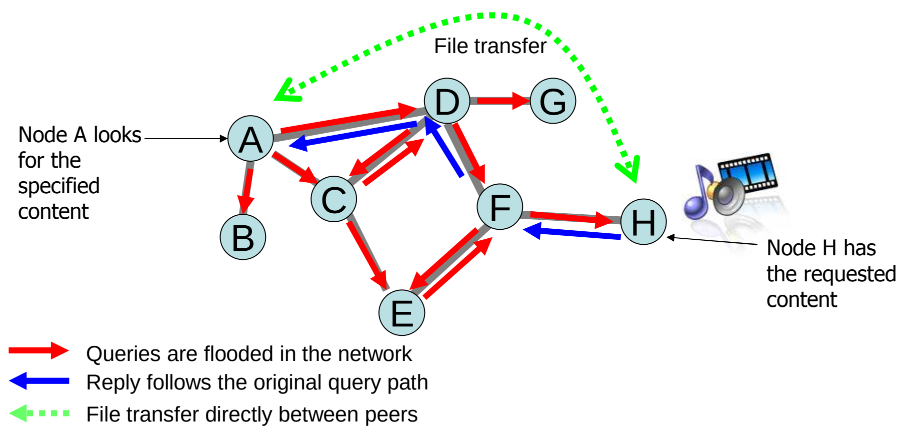
Decentralized P2P architecture

- Control and data are completely distributed
  - No centralized server
- All nodes in the network are equal
  - No single node has control over the content shared by the other users
- Searches are done by flooding search requests in the network
- Downloads are executed directly between the peers
- Example: Gnutella v0.4



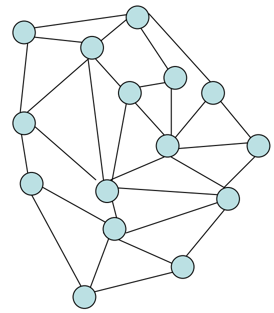
Decentralized P2P architecture

- Requests are flooded to all nodes
- Reply path may follow query path or be direct



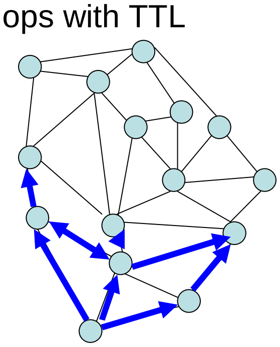
Decentralized P2P architecture

- Advantages
  - Resilience to node failures and attacks
- Disadvantages
  - Searching is based on flooding, which is inefficient in terms of bandwidth.
  - TTL is used to limit the scope of flooding → not all resources are found
  - Long search delays
- Connections tend to form a power-law graph
  - Most peers have low number of connections, small number of highly connected peers

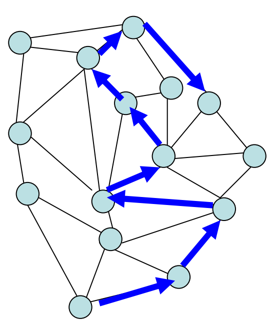


Reducing messages of flooding reduces the number of hits

- Fixed TTL or iterative deepening
  - limit the number of hops with TTL

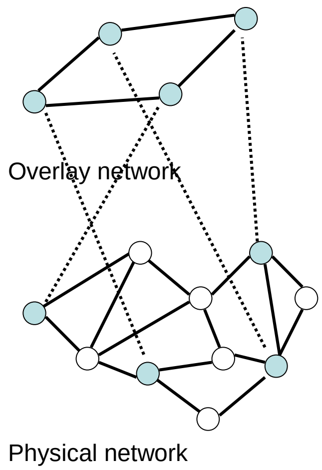


- Random walk
  - forward the search to only one neighbor



Overlay networks

- Every peer cannot be connected to all other peers → select a subset of neighbors
- An overlay network is formed
- Direct communication with neighbors, indirect communication with other peers
- The overlay is independent of the physical network
  - Nodes that are neighbors in the overlay network may be far away physically
  - Makes flooding even more inefficient

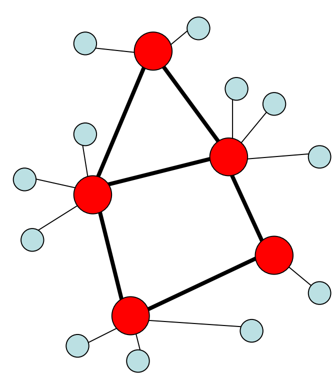


Semi-centralized P2P architecture

Combines the centralized and distributed architectures

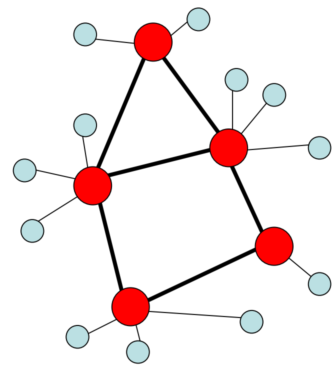
Two types of peers:

1. Super-peers (ultrapeers, core nodes, gateways)
   - Usually more powerful peers
   - Form a decentralized P2P network between themselves
   - Handle search requests on behalf of clients
2. Ordinary-peers (clients, edge nodes)
   - Usually less powerful peers (e.g. behind firewall/NAT)
   - Connect as clients to super-peers
   - Upload index information about shared files to super-peer



Semi-centralized P2P architecture

- Searching is based on flooding between the superpeers
- Downloads are executed directly between the peers
- Advantages:
  - Improved scalability
  - Stability, higher success rate
- Disadvantages:
  - Searching is still based on flooding
  - Loss of distribution, more centralized control
- Examples: FastTrack (Kazaa), Gnutella v0.6



Semicentralized P2P – KaZaA

- KaZaA is an example of semi-centralized P2P network
- Uses the FastTrack protocol
- Super-Peers (SP) are normal peers that have been automatically elected as the super-peers based on their uptime, bandwidth, connectivity, CPU power, IP address (public vs. private)
- Super-peers maintain a database with metadata (file name, file size, content-hash, file descriptors) of shared files and the corresponding IP addresses
- SP maintain large number of long-lived TCP connections with other SPs
- KaZaA peers frequently exchange list of super-peers
  - An ordinary-peer (OP) maintains list of 200 super-peers

Semicentralized P2P – KaZaA

- File searching
  - OP sends a query with a keyword to its SP
  - SP returns IP addresses and related metadata that correspond to the match from its database
  - SP may forward query to one or more SPs to which it is connected
  - Query visits only a small subset of SPs so the result represent only a small subset of all files stored in KaZaA network
- All signaling traffic between peers is encrypted
- File transfer between nodes is not encrypted
- TCP is used for both file transfer and signaling traffic

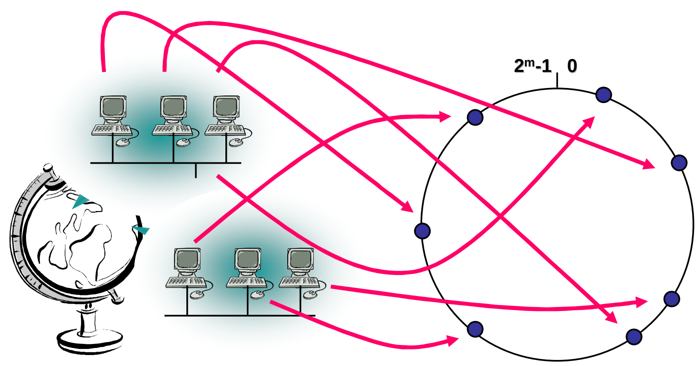
Structured P2P architectures

- Also decentralized – there is no single point of control
- Based on Distributed Hash Tables (DHT)
- Location of index information is strictly determined
  - Fast searching
  - Very bandwidth efficient
  - Maintenance of structure and routing tables causes additional traffic
- Only exact-match single-key lookups are supported
  - Exact name for the searched content must be known
  - Does *not* support substrings (e.g. file names containing a given word), value ranges (e.g. mp3-files with bitrate larger than a given value), multiple attributes (e.g. mp3-files with given name AND given bitrate), etc

  → not popular in traditional file sharing

Structured P2P - Chord

- Distributed hash tables (DHT) map IP-addresses into a circular logical address space: id = SHA1(IP address)

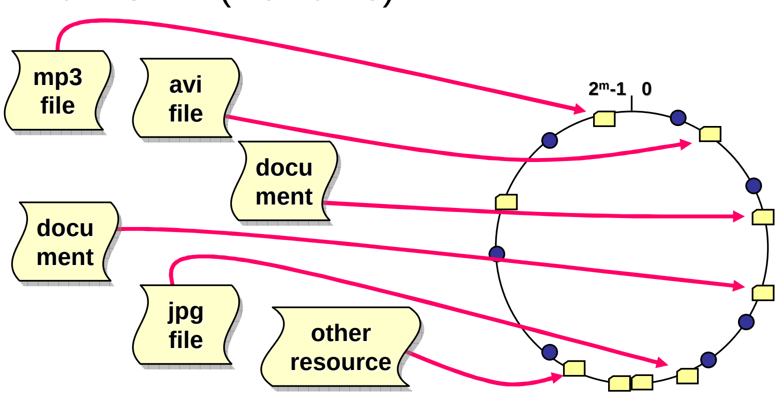


S-38.3155 / N Beijar, M Matuszewski and J Lehtinen / 4.4.2008

21

Structured P2P - Chord

- Also the resources (e.g. files) are mapped into the circular logical address space: id = SHA1(file name)

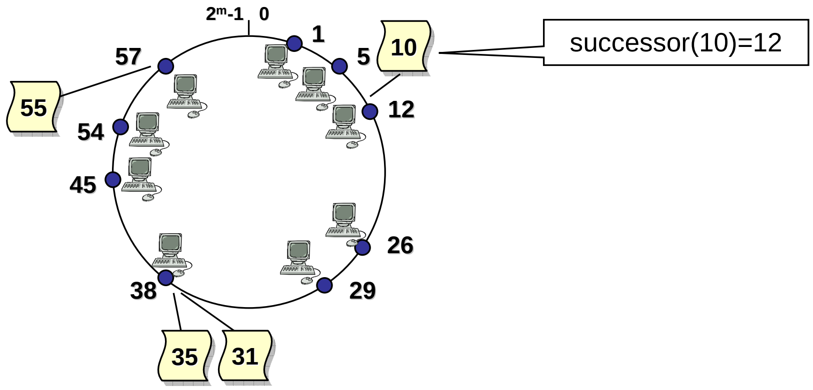


S-38.3155 / N Beijar, M Matuszewski and J Lehtinen / 4.4.2008

22

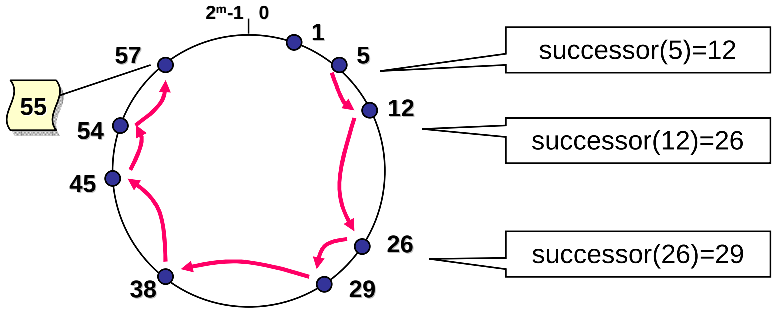
Structured P2P - Chord

- A resource with key *k* is assigned to the first node whose identifier is equal to or follows *k* in the identifier space (=the successor of *k*)



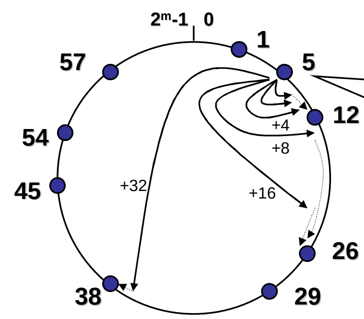
Structured P2P - Chord

- The *minimum* requirement for correct operation is that every peer knows its successor
- Simple (but inefficient) search is then possible:



Structured P2P - Chord

- Every node *n* maintains a finger table
  - finger(*i*) = successor($n + 2^{i-1}$), $1 \le i \le m$



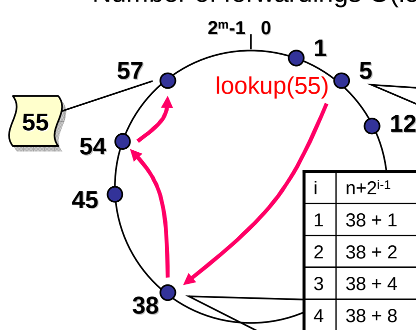
i	$n+2^{i-1}$	finger(i)
1	5 + 1	12
2	5 + 2	12
3	5 + 4	12
4	5 + 8	26
5	5 + 16	26
6	5 + 32	38

S-38.3155 / N Beijar, M Matuszewski and J Lehtinen / 4.4.2008

25

Structured P2P - Chord

- The finger table is used for *efficient* searching
  - Precicion increases on each hop
  - Number of forwardings O(log N)



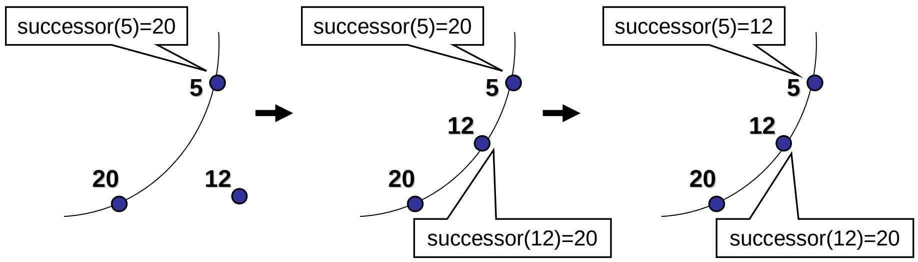
i	$n+2^{i-1}$	finger(i)
1	5 + 1	12
2	5 + 2	12
3	5 + 4	12
4	5 + 8	26
5	5 + 16	26
6	5 + 32	38

i	$n+2^{i-1}$	finger(i)
1	38 + 1	45
2	38 + 2	45
3	38 + 4	45
4	38 + 8	54
5	38 + 16	54
6	38 + 32	12

S-38.3155 / N Beijar, M Matuszewski and J Lehtinen / 4.4.2008

26

Chord – joining node



Node 12 joins the network. It performs a lookup on its ID and learns that successor(12)=20.
On the following update node 12 informs node 20 that successor(12)=20.

Node 5 has incorrect information about its successor. On the following update, node 5 asks its current successor (node 20) if it still is its successor. Node 20 tells that successor(12)=20. Thus, node 5 learns that successor(5)=12.

The network is again stable.

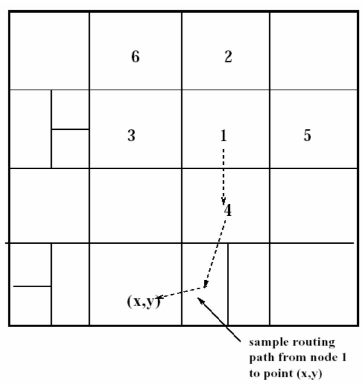
Properties of Chord

- Availability
  - Protocol functions very well even if the system is in a continuous state of change
- Scalability
  - Lookup grows logarithmically with the number of nodes, O(log N)
- Load balancing
  - Keys are spread evenly over the nodes
  - But no control over where the information is stored
- Maintenance of finger table causes traffic
  - Check that successor and predecessor are consistent
  - Update fingers
- No complex queries
  - Chord only supports exact-match single-key queries

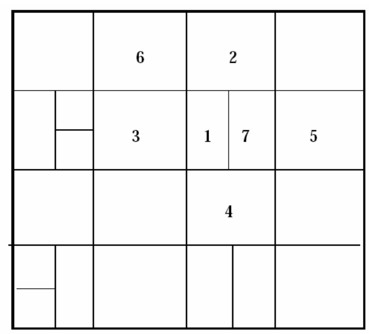
Other structured peer-to-peer algorithms

- Chord
  - circular key space
  - simple, popular in research
- Kademlia
  - k-bit integer key space with distance defined as XOR between two keys
  - used in eDonkey, eMule, BitTorrent, etc
- Content addressable network (CAN)
  - multi-torus key space
- Tapestry
  - hexadecimal key with prefix-based routing
- Pastry
  - similar to Tapestry with support for proximity

CAN example



1's coordinate neighbor set = {2,3,4,5}
7's coordinate neighbor set = { }



1's coordinate neighbor set = {2,3,4,7}
7's coordinate neighbor set = {1,2,4,5}

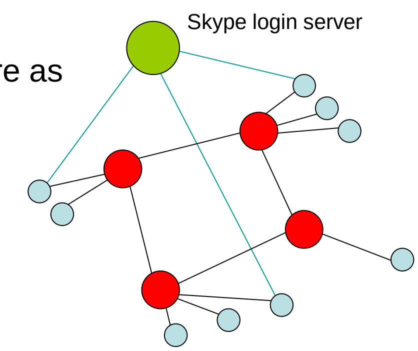
Agenda

- Introduction
- P2P architectures
- **Skype**
- P2P-SIP
- Mobile P2P
- Summary

Skype

- Skype uses the peer-to-peer concept
  - to locate users
  - to relay traffic for bypassing firewalls/NAT
- Provides internet telephony, instant messaging and file transfer services
- Skype is a proprietary protocol in contrast to SIP and H.323
  - No official specifications available
  - Some info acquired by reverse engineering the protocol
    - Salman A. Baset, Henning Schulzrinne, ”An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol”

Skype



- Skype has a similar architecture as its predecessor KaZaA
- There are three types of nodes in the Skype network:
  - Ordinary-peers
  - Super-peers
  - Central login server
- The login server stores all of user names and passwords and ensures that names are unique across the Skype name space

Skype - some facts

- Uses TCP for signaling and both UDP and TCP for transporting media traffic
- Uses GlobalIPSound's iLBC and iSAC codecs (and a third party unknown voice codec)
- All user communication is encrypted using AES 256-bit (Advanced Encryption Standard)
- Uses a variation of STUN and TURN for NAT and firewall traversal
- Normal login
  - The Skype client connects to a super-peer
  - The Skype client authenticates the user name and password with the Login Server

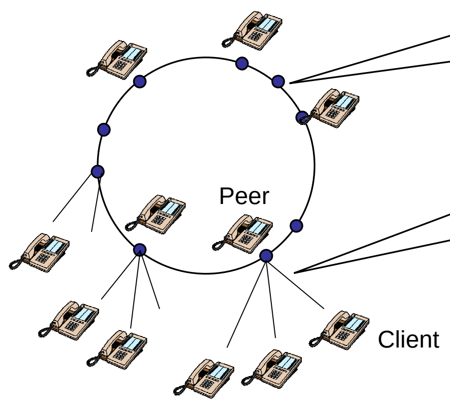
Agenda

- Introduction
- P2P architectures
- Skype
- **P2P-SIP**
- Mobile P2P
- Summary

P2P-SIP

- Ongoing standardization work in IETF
  - P2PSIP working group
- Replaces servers with a structured P2P network
- Uses a DHT (e.g. Chord) to locate users
- Why P2P-based SIP?
  - Small deployments
  - Limited/no internet connectivity
  - Ad-hoc group
  - Infrastructure independence, no servers
  - Simple setup
  - Privacy, lack of central control
  - Scalability

P2P-SIP



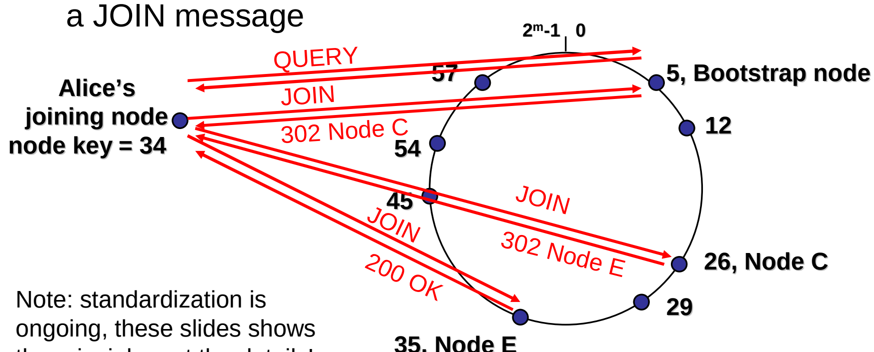
Peers communicate using a lightweight binary P2P-SIP peer protocol, e.g. P2PP

Clients communicate with peers using a P2P-SIP client protocol. Several possibilities.

Several candidate peer-protocols have been proposed:
- P2PP, ASP, Reload, XPP & XPP-PCAN, HIP HOP, With HIP, SEP

P2P-SIP: Joining

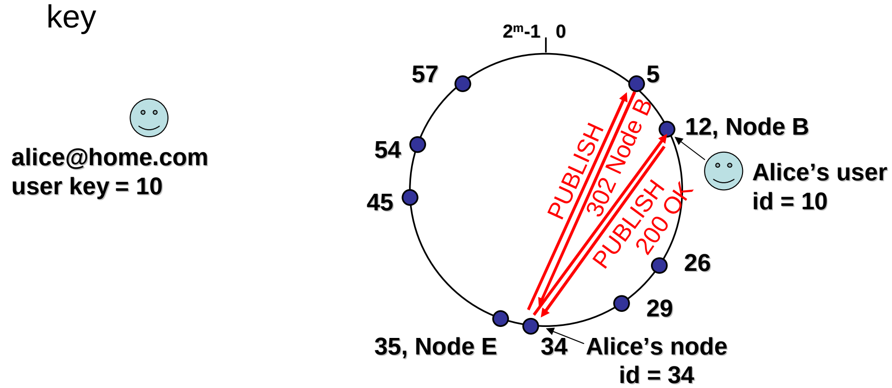
- The node key is calculated from the IP address
- The node asks information about the network with QUERY
- The node joins the DHT based on the node key by sending a JOIN message



Note: standardization is ongoing, these slides shows the principle, not the details!

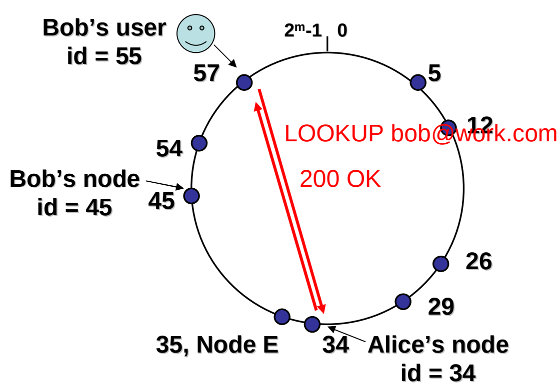
P2P-SIP: Registration

- The user's key is calculated by hashing the user's screen name, e.g. alice@home.com
- A PUBLISH is sent to the node responsible for the user's key



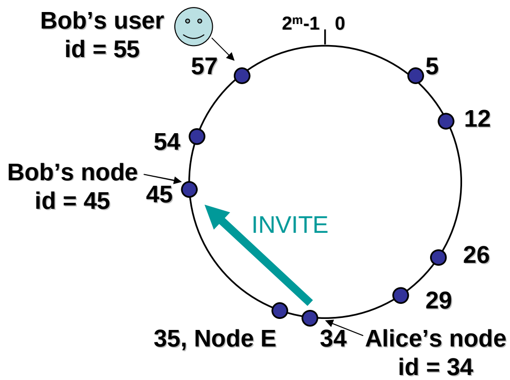
P2P-SIP: User search

- Alice searches for id 55 (hash of bob@work.com), which is managed by node 57. Alice sends a LOOKUP to node 57, who returns Bob's contact.



P2P-SIP: Call establishment

- Alice sets up a call to Bob using normal SIP signaling



Agenda

- Introduction
- P2P architectures
- Skype
- P2P-SIP
- **Mobile P2P**
- Summary

Mobile P2P (MP2P)

- Faster residential Internet connections, powerful computers, and cheaper storage were the main drivers stimulating P2P growth
- We can observe a similar technological change in mobile networks
- Mobile device becomes a platform for producing and consuming digital media
  - Camera, mp3-player, TV, etc
  - Both user-created and commercial contents
- Real-time access important
- Mobility and location-based services
- P2P is more than file sharing
  - General location of services, users and information
  - Integrates well with many typical mobile applications

Technical challenges

- Device capabilities
  - Memory size
  - CPU performance
  - Screen and keyboard size
  - Battery capacity
- Network
  - Limited and expensive bandwidth
  - Firewalls and NATs
- Widely used P2P applications/protocols have to be redesigned
- Bandwidth must be minimized
  - Bringing search traffic to the terminal does not scale!

Business and user challenges

- Special Needs of Mobile Environment
  - Operator control
  - Users are authenticated
  - Charging models (currently paying both for uploading and downloading, flat rate fills all available capacity)
  - Understand and analyze the impact of peer-to-peer services on the mobile market and its value chain
- User Requirements
  - Quick response times, rapid downloads
  - Group management features for sharing private content
  - Lot of content is probably self-created, like pictures/videos taken with camera-phone

MP2P seems to require a support from the fixed network

- Semi-centralized architecture for generic sharing
  - Minimizes signaling load on the air interface
  - Allows operator to have control on content by controlling the super-peer
  - Multiple operators can network super-peers in peer-to-peer fashion still retaining quite high autonomy
  - Super-peer can be also operated by private entity, e.g. family or sports club
- Two-layer architecture (clients and peers) for P2P-based VoIP
  - Do structured architecture offer low enough traffic for phones to participate in the DHT?

Agenda

- Introduction
- P2P architectures
- Skype
- Mobile P2P
- P2P-SIP
- **Summary**

Summary

- P2P offers scalability, robustness, fault tolerance and decentralized control
- File sharing is a dominant P2P application
  - Other applications, such as Internet telephony, are emerging
- P2P-SIP concept aims to improve scalability and usability of standardized SIP applications – no need for centralized SIP nodes
- In the near future we will see P2P services in the mobile domain
- Can P2P replace DNS? Search engines?