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 relies primarily on the computing power and bandwidth of the participants in the network rather than concentrating it in a relatively low number of servers. P2P 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. 19.2.2007)

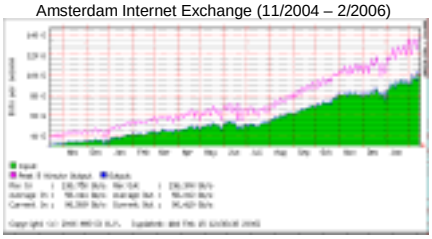
Introduction

- Generally in peer-to-peer communication
  - Each communicating node (peer) has both server and client capabilities
  - Any party can initiate a communication session
  - Applications connect with each other directly
  - Users can search for resources (files, services, users)
- Peer-to-peer paradigm has many uses
  - File-sharing
  - Internet telephony
  - Distributed computing
  - Collaboration
  - …

Peer-to-Peer popularity

- File-sharing applications are the most popular form of P2P – at least traffic wise – e.g. BitTorrent, Kazaa, Direct Connect
- P2P accounts for 60 – 80% of all Internet traffic
- Other peer-to-peer applications are also gaining popularity, e.g. Skype - Voice over P2P, Mobile P2P

Amsterdam Internet Exchange (11/2004 – 2/2006)

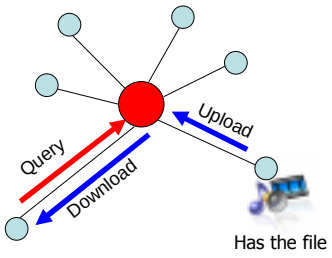


Agenda

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

Traditional Client-Server Architecture

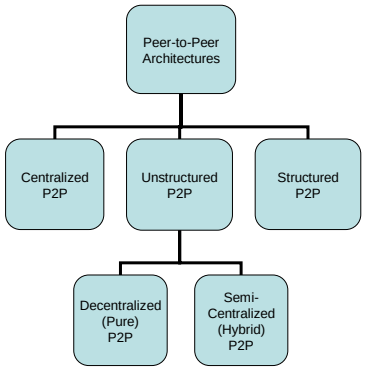
- One high-performance server (or cluster of servers) holds all the content in the network
  - Owner of the server has full control of the content
- Multiple clients share content via the centralized server
- No communication between clients
- Limited scalability
  - The server must store all contents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



This is <u>not</u> peer-to-peer!

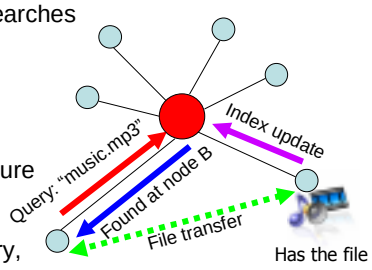
P2P architectures

- Three main architecture types
  - Centralized peer-to-peer
  - Unstructured peer-to-peer
    - Decentralized peer-to-peer (a.k.a. pure p2p)
    - Semi-centralized peer-to-peer (a.k.a. hybrid p2p)
  - Structured peer-to-peer
    - Always decentralized



Centralized P2P architecture

- A centralized server (or a cluster of servers) stores index information 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
- 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
- Example: Napster

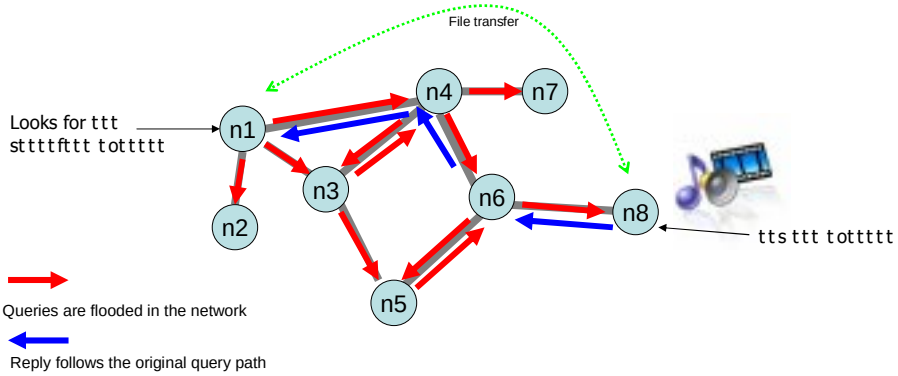


Pure P2P architecture

- 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
- Advantage
  - Resilience to node failures and attacks
- Disadvantage
  - 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
- Inter-peer connections tend to form a power-law graph (most peers have low number of connections, small number of highly connected peers)
- Example: Gnutella

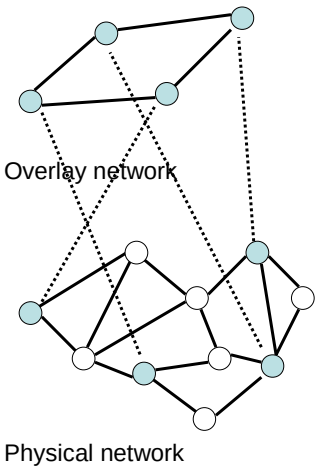
Search in a pure P2P architecture

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



Overlay networks

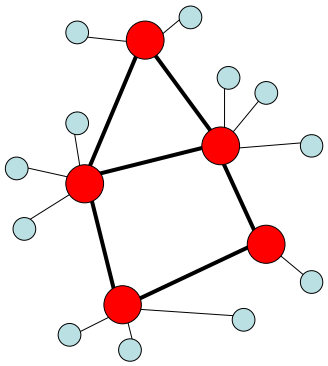
- Every peer cannot be connected to all other peers -> select a subset of neighbors
- An overlay network is formed
- Direct communication with neighbors, indirect communication with other peers
- The overlay is independent of the physical network
  - Nodes that are neighbors in the overlay network may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



Hybrid (semi-centralized) P2P architecture

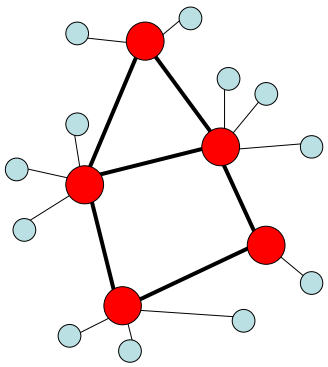
Two types of peers:

1. Super-peers (Gateways)
   - more powerful peers become gateways to a network
   - form a pure P2P network between themselves
   - handle search requests on behalf of clients
2. Ordinary-peers (Clients)
   - less powerful peers connect as clients to the super-peers peers
   - upload metadata information about shared files to super-peer



Hybrid (semi-centralized) P2P architecture

- Searching is based on flooding between the superpeers
- Downloads are executed directly between the peers
- Advantages:
  - Scalability
  - Stability, higher success rate
- Disadvantages:
  - Searching is still based on flooding
  - Loss of distribution, more centralized control



Hybrid 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

Hybrid P2P - KaZaA

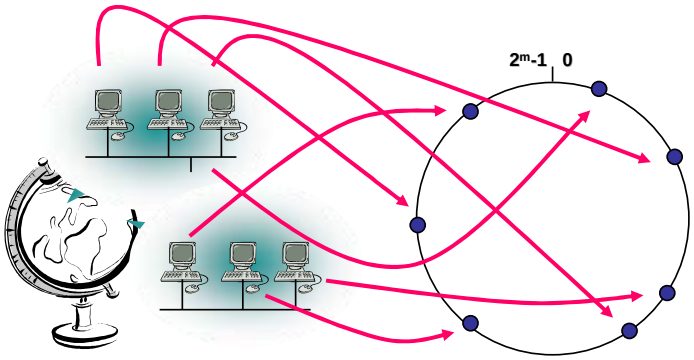
- File searching
  - OP sends a query with a keyword to its SP
  - SP returns IP addresses and related metadata that correspond to the match from its database
  - SP may forward query to one or more SPs to which it is connected
  - Query visits only a small subset of SPs so the result represent only 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 totally decentralized – there is no single point of control
- Based on Distributed Hash Tables (DHT)
- Location of information is strictly determined
  - Fast searching
  - Maintenance of structure (routing) causes traffic
- Wildcard searches are not possible
  - Exact name for the searched content must be known, thus unusable for traditional file sharing

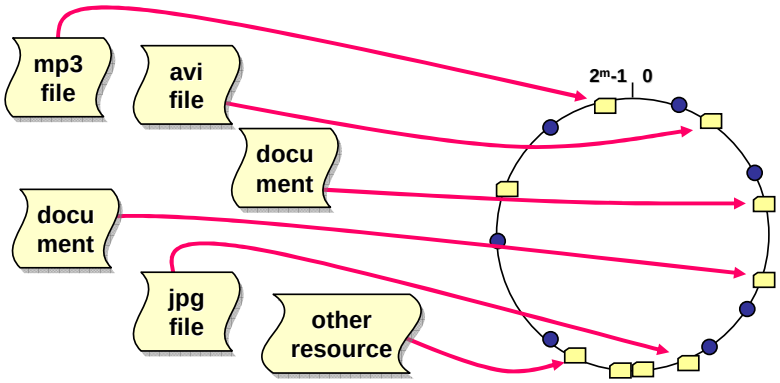
Structured P2P - Chord

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



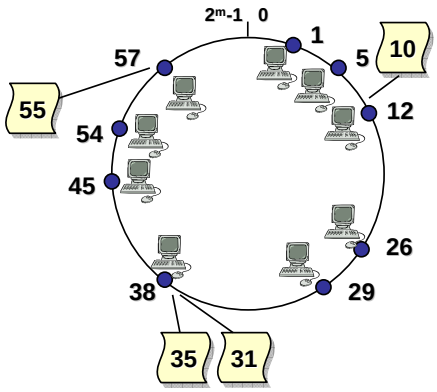
Structured P2P - Chord

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



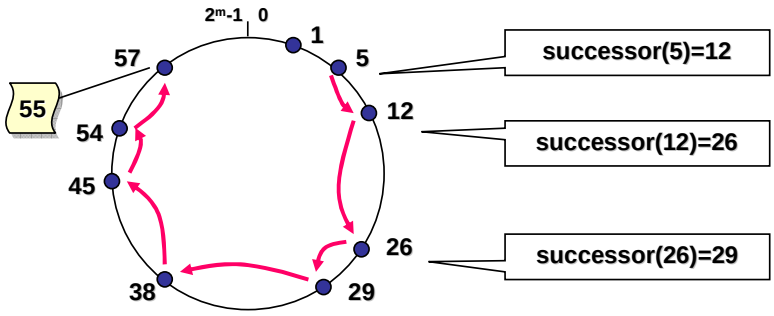
Structured P2P - Chord

- Key *k* is assigned to the first node whose identifier is equal to or follows *k* in the identifier space (=*successor*)
- Example Chord ring (m=6):



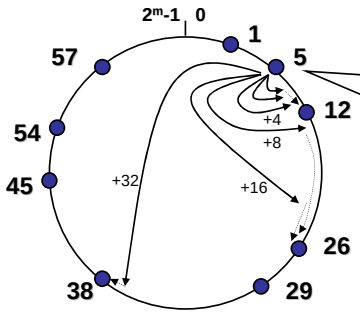
Structured P2P - Chord

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



Structured P2P - Chord

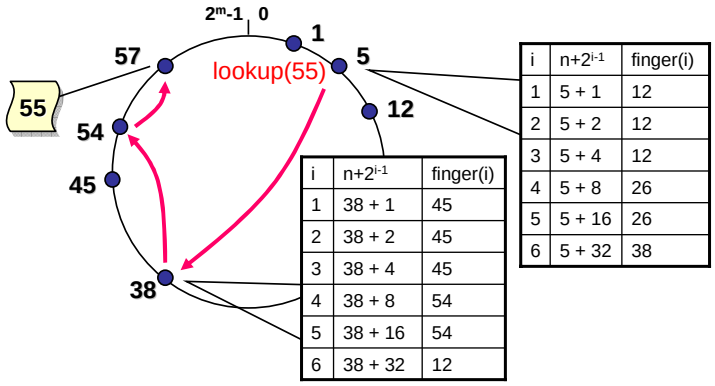
- Every node *n* maintains a finger table
  - finger(*i*) = successor($n + 2^{i-1}$), $1 \le i \le m$



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

Structured P2P - Chord

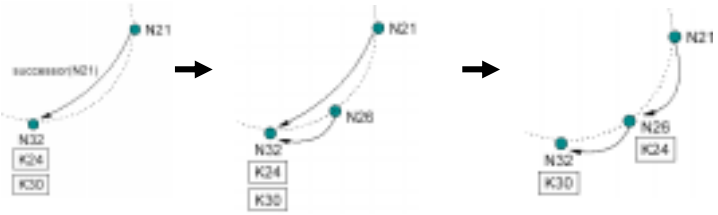
- The finger table is used for efficient searching
  - Number of forwardings O(log N)



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

Chord – joining node



- Node 21 asks its successor the following question: “Am I your predecessor?”
- If node 26 joins the network an answer to this question is: “NO, node 26 is my predecessor”

Properties of Chord

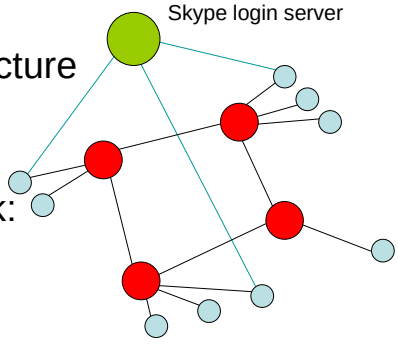
- Availability
  - Protocol functions very well even if the system is in a continuous state of change
- Scalability
  - Lookup grows logarithmically with the number of nodes, O(log N)
- Load balancing
  - Keys are spread evenly over the nodes
  - But no control over where the information is stored
- Maintenance of finger table causes traffic
  - Check that successor and predessor are consistent
  - Update fingers
- Flexible naming
  - No constraints on a key structure
- No complex queries
  - Chord supports “exact match“, cannot handle queries similar to one or more keys

Agenda

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

- Skype uses the peer-to-peer concept to locate users and relay traffic for bypassing firewalls/NAT
- Provides internet telephony, instant messaging and file transfer services
  - Over 250 million downloads
- Skype is a proprietary protocol in contrast to SIP and H.323
  - No official specifications available
  - Some info acquired by reverse engineering the protocol

Skype: Architecture



- 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
- Buddy list is signed digitally encrypted and is local to the machine (not stored on the central server like in MSN Messenger)

Skype: Login

- After installation, a client connects to some bootstrap super-peers, since its Super-Peer list is empty, and acquires the address of the Login Server
- Normal login
  - The Skype client (OP) connects to a Super-Peer
  - OP authenticates the user name and password with the Login Server

Skype: User search

- The client sends an user name to the SP and receives four IP addresses and port numbers
  - Subsequently the client contacts these four nodes
  - If it cannot find the user it sends the request to its SP once again and as a result receives eight IP addresses and port numbers
  - The process continues until the user is found
- If the user is behind a NAT and an UDP-restricted firewall, the SP searches the user on behalf of the client
- Search results are cashed in the intermediate nodes

Skype: Call establishment

- If both a caller and a callee have public IP addresses, a caller sends signaling information over TCP to a callee
- If a callee is behind a port-restricted NAT, the caller sends signaling information over TCP to an online Skype node that forwards it to a callee
- If both a callee and a caller are behind a port-restricted NAT and an UDP-restricted firewall, both exchange the information with an online Skype node

Agenda

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

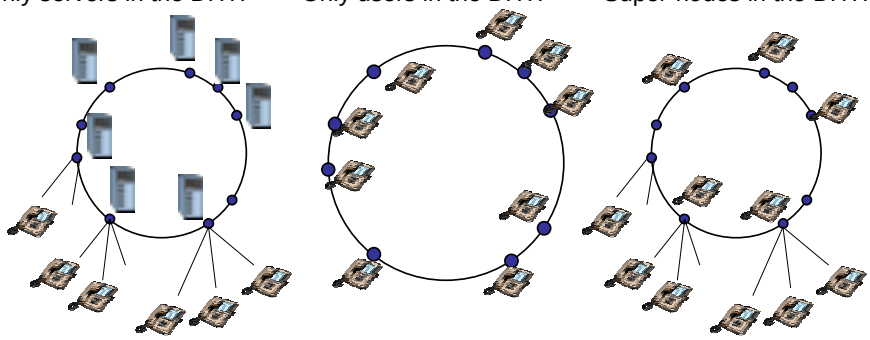
P2P-SIP

- Ongoing standardization work in IETF
  - P2PSIP, SIPPING working groups
- 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

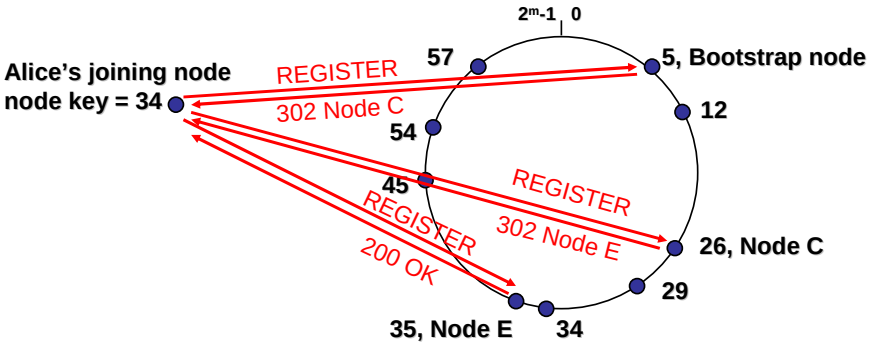
- Uses a Chord-based approach to locate users
- Ongoing work – different approaches considered

Only servers in the DHT: Only users in the DHT: Super-nodes in the DHT:



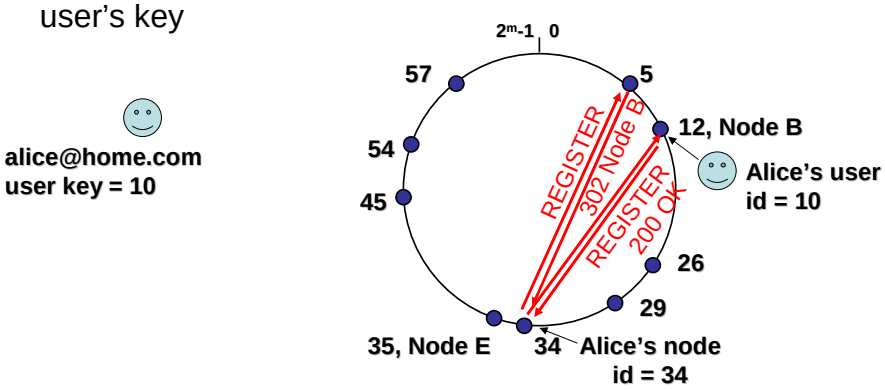
P2P-SIP: Joining

- The node key is calculated from the IP address
- The node joins the DHT based on the node key by sending a REGISTER message



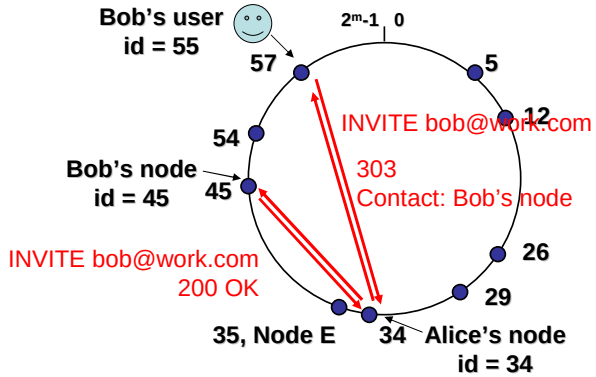
P2P-SIP: Registration

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



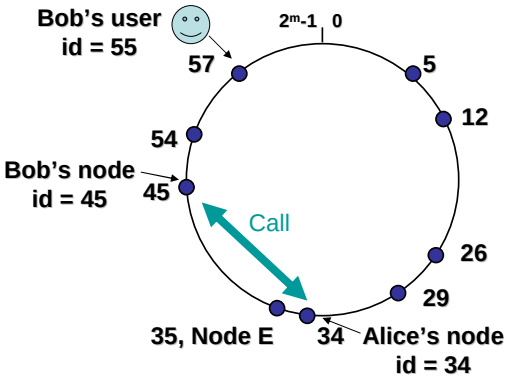
P2P-SIP: User search

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



P2P-SIP: Call establishment

- Alice sets up a call to Bob



Agenda

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

Mobile P2P (MP2P)

- Faster residential Internet connection, more powerful desktop 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

Technical challenges

- Shortage of resources
  - Memory size
  - CPU performance
  - Screen and keyboard size
  - Battery capacity
  - Access Network Parameters
    - Limited bandwidth shared between multiple users in the same cell
- Widely used P2P applications/protocols have to be redesigned

Business and user challenges

- Special Needs of Mobile Environment
  - Support for various access networks
  - Operator control
  - Feasible bandwidth pricing
  - 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 super-peer based architecture

- Hybrid 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
- Mobile Voice over P2P
  - Super-peers

SIP-based MP2P application

- SIP based mobile peer-to-peer application has been developed in the Networking Laboratory
  - Hybrid peer-to-peer architecture
  - All signaling is in SIP
    - Searching with INVITE
    - File-list update with MESSAGE
    - Download initialization with INVITE
  - Use of SIP as the signaling protocol allows easy integration with IMS and other SIP aware networks
  - Search and file-list update messages have content information encoded in XML to enable easy parsing and future extensions

Agenda

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

Summary

- P2P offers scalability, robustness, fault tolerance and decentralized control
- Content 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?

Thank you!