Peer-to-Peer Networking

Introduction

Introduction

- This course introduces you to many different aspects of P2P technology
- We will especially consider them from the networking point of view
- First we need to define and clarify the characteristics and concepts, background, etc. of P2P
- After this course you should be able to *understand* and *recognise* what is P2P and what is not

Lecture Content

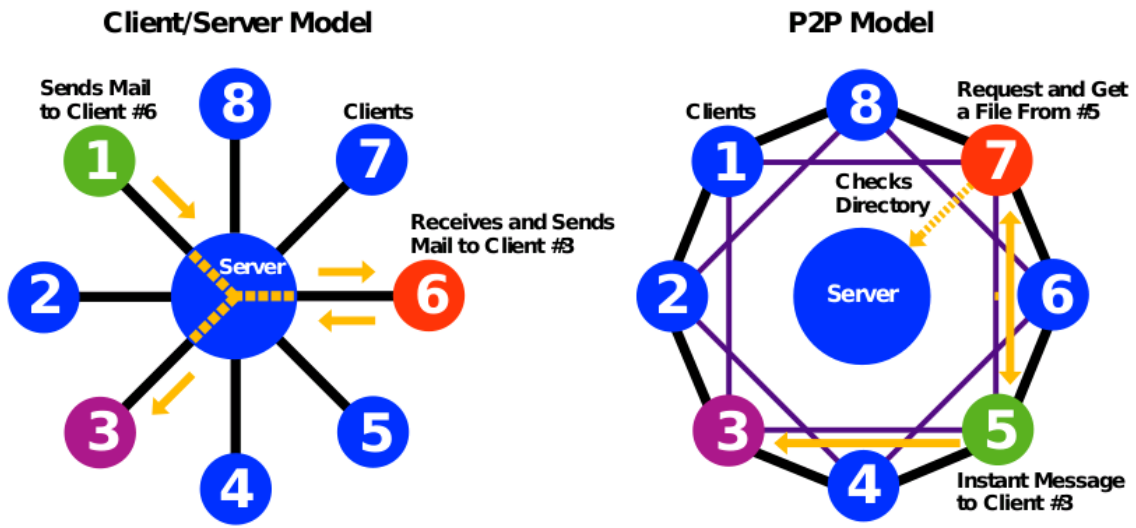
- What is a **peer** and **P2P model**
- **Definitions** and **characteristics**
- How and why did P2P **emerge** in the first place
- **Terminology:** computing, communication, network
- In the heart of P2P: **sharing** and **collaboration**
- **Benefits** and **challenges**
- Implications for **user, developer,** and **IT industry**

What Is a Peer

- Remember the client-server model?
  - client (user) makes requests to the networked server (hidden in a back room), server responds and acts on the requests
- P2P: each *peer*, i.e. participating computer, can act both as a client and as a server *in the context of some application*
- "peer" describes this new relationship

P2P Model

- Client-server model: communication goes trough a server (mail server)
- P2P model: communication and computing towards the edges of the net (file transfer and IM)



[David Barkai et al. An Introduction to Peer-to-Peer Computing]

P2P Model

- The client-server duality is a central feature of P2P
- *Pure* P2P corresponds to the above description, where all peers are equal in the client-server sense
- In *hybrid* P2P model there are dedicated servers which act only as servers (Napster directory service)
- In *structured* P2P systems the overlay network structure is defined and controlled by the P2P system

More Formal Definitions of P2P

- *"Peer-to-peer computing is a network-based computing model for applications where computers share resources via direct exchanges between the participating computers"*
- *"P2P is a network architecture in which nodes are relatively equal, in the sense that each node is in principle capable of performing each of the functions necessary to support the network, and in practice many nodes do perform many of the functions"*

Characteristics of P2P Architecture

- Peers act both as client and server
- Peers contribute services or resources to others
  - content, storage, CPU, ...
- Interaction between peers direct without servers
- Nodes are autonomous (no centralised administration)
- Network is dynamic: peers join and leave frequently, services may become temporarily unavailable
- P2P systems are self-organising

The Emergence of P2P

- First, there was the Internet (HTTP, Telnet, FTP, ...)
- The 2nd generation: graphical-interface browsers
- Since the late 1980s, client-server architectures have been the norm
- Common working-tools as client-server: accessing web pages with a browser, downloading emails, ...
- Client-server architecture is mature and provides centralised control and coordination

The Emergence of P2P

- Aggregate “power at the edge of the net” growing
- Technically P2P enables the use of huge amounts of untapped resources, eliminates single source bottleneck and single point of failure
- Many P2P systems based on well-known protocols, like TCP/IP, HTTP, XML, UDDI, SOAP
- Nowadays: P2P, decentralization, self-organizing online communities collaborating, web becomes “writeable”

…And Why Now?

- The ubiquity of connected computers
- Broadband communication
- Increased popularity of wireless devices
- Advances in software agent technologies
- We could say that Napster was the first *killer app* for P2P computing model

The Popular View of P2P

- In 1999-2000, P2P emerged to popularity
- Napster, Scour: direct exchange of MP3 music files
- Napster: hybrid of P2P and client-server architecture
  - File repository highly distributed among peers
  - Directory centralised on the company's own servers
- Servers: single point-of-failure and bottle-neck
- Napster closed in 2002 due to copyright breaches

The Popular View of P2P

- Plenty of others similar and different service types
- The Freenet Project: completely decentralized, anonymous way to distribute information
- Gnutella: Decentralized, any-type files, anonymity, resilience of service
- SETI@home: “cycle sharing”, sharing computing resources and collaboration
- Skype for real-time two-way audio/telephone

Terminology

- P2P Computing, Network and Communications?
- P2P network allows every computer in the network to act as a server to every other user on the network
- P2P communication enables both participants to initiate, manage and terminate the session
- While P2P network implies P2P communication, the P2P communications can also occur between two computers in a non-P2P network

Terminology

- P2P computing does not imply that every node can be a server, the relationship is negotiated somehow
- The peers in P2P computing are likely to exercise P2P communications, although not necessarily
- In short: P2P network and P2P communication refer to the physical network and a mode of communications, P2P computing defines end-user applications environment

P2P Computing

- Can refer to the model, or it can refer to the action of computing within that framework
- The P2P framework allows peers to interact with each other directly: the computing environment becomes *decentralized*
- Computing is moved to the *edges of the net*
- Still some the connections can remain to the centre

P2P Framework

- End-user applications include storage, computations, messaging, security, distribution, etc...
- Common to all is that they involve *sharing* of resources and a form of *collaboration* among peers
- To the end user:
  - No dependence on central servers
  - Enables users to organize in self-governed communities
- Production use, business-to-business, e-Commerce...

Share and Collaborate

- What to share in P2P?
- Files, like music, video or any other type of file
- compute-cycles (SETI@home)
- Storage space for another peer or community
- Services and their management: instant messaging and security
- Information: make available or transfer to others

Share and Collaborate

- Next, how the peers should collaborate?
- Direct exchange between the peers
- Transactions taking place with the help of an intermediate (Napster)
- Granting access to resources (SETI@home)
- Any kind of interaction formed through policies that are implied when a peer joins an online community

Benefits of P2P

- Avoidance of bottlenecks and single-points-of-failure
- Reduced dependence on individual devices and sub-networks
- Improved resilience, scalability and ability to service highly-peaked demand
- Resistance to denial of service attacks
- Easy sharing of resources and services
- Low cost of deploying the network

Interoperability

- Many developers create their own underlying services and tools
- Applications cannot “talk” to each other
- Different hardware architectures, operating systems, devices such as servers, desktop and mobile PCs, wireless and even sensor-like appliances
- Effective solution: *a common infrastructure*
- Interoperability is just the first of big P2P challenges

Other P2P Challenges

- Dealing with firewalls, NATs and dynamic IPs
- Security and local autonomy
- Trust in P2P environment, need for an authority?
- Efficient searching and locating wanted services and resources
- Dynamic nature of P2P networks, peers joining and leaving
- The list continues...

The User's Perspective

- Online identity that is not attached to a device: using home/lab/mobile computer, tablet etc. in changing locations
- Multiple communities and profiles: family, work, ... in profiles access permission, content interests, ...
- A trust-space within peers? local autonomy while entrusting others with one's own resources
- Applications: interoperable or integrated with each other

Practical Requirements for P2P

- A physical network of appropriate topology, reach and scale (Internet, extranet, intranet?)
- Standard protocols for communication among nodes
- Naming conventions: devices need to be identified
- Naming conventions for digital objects (e.g. hash)
- Service and resource metadata (e.g. for searches)
- Discovery mechanisms: locate service or resource

The Implications for IT

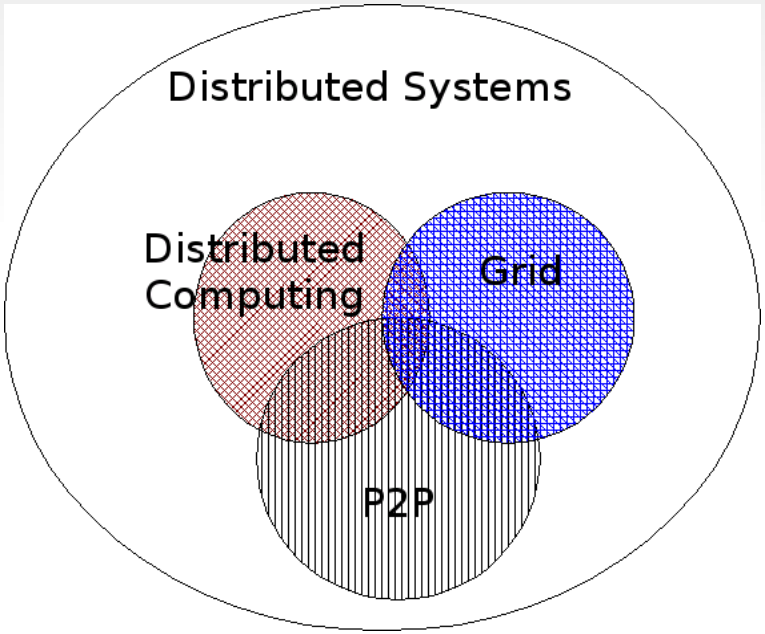
- Distributed storage and computing
- Data backup
- Remote maintenance: cascading software or files from peer to peer thus balancing network traffic
- Edge services: getting data where it is to be used
- Virus protection: access negotiation and intelligence software agents, distribute warning peer-to-peer

Application Development

- Interoperability (as talked earlier)
- Use of common standard infrastructure: can focus on applications features rather than low-level system
- Re-configurable environment: dynamic environment, peers join and leave, data migration, ...
- Security and fault tolerance
- These requirements favour object-based and modular programming practices

What Is P2P and What Is Not?

- How to separate P2P systems from, e.g., distributed and grid computing?
- Loosely defined: distributed systems cover all systems with multiple nodes connected by a network
- Distributed computing: solve a large problem by giving small parts of the problem to many computers to solve and then combining the solutions
- P2P and grid computing are two approaches to distributed computing, both concerned with the organization of resource sharing in large-scale computational societies



Grids and P2P

- Grids are *coordinated* and *high-performance* oriented forming some virtual organisation
- Grid provides access to pooled resources as a platform, initially P2P apps were vertically integrated
- P2P systems focus on dealing with self-adaptation, transient populations, instability, and fault tolerance
- One can say that “Grid computing addresses infrastructure, but not yet failure, while P2P addresses failure, but not yet infrastructure”

Summary

- P2P is a novel way of solving problems
- Idea of sharing and collaborating among peers
- Utilises huge amounts of unused resources on the edge of the net
- Great benefits: scalability, easy data and service distribution for end-user, no single point failure, ...
- But also great challenges: interoperability, security, dynamic nature, intermittent connectivity...

To Think Over

- *Decentralisation:* what does it mean to end-user? Consider, for example, the services, autonomy and security management?
- In general, why is the Internet a difficult network to apply efficient and interoperable P2P system?
- Are there any other issues, e.g. governmental or concerning the traffic (what happened to Napster, why P2P services are blocked on campus networks)
- What separates P2P from distributed computing?

The Rest of the Course

- Lecture 2: Overview of P2P applications and systems
- Lecture 3: P2P overlay networks and topologies
- Lecture 4: Case studies of selected P2P systems
- Lecture 5: Issues: connectivity, naming, searching
- Lecture 6: Security issues, fairness and trust
- Exercises are given per lecture
- Project work
- Exam