
Teaching Teamwork

- Jussi Kangasharju
- Rolland Vida
- Klaus Wehrle

Agenda

- Experiences from past courses
- Topics of a Peer-to-Peer course
- Lab sessions

Experiences from past courses

- **Jussi: Peer-to-Peer and GRID Computing**
 - ▶ 2h+2h
 - ▶ 2 times, 7th+ semester, 70 students
- **Rolland: Peer-to-Peer-Systems**
 - ▶ 4h (no exercises)
 - ▶ 5 times, higher semesters, 100-130 students
- **Klaus: Peer-to-Peer Systems and Applications**
 - ▶ 2h+1h
 - ▶ 3 times, 7th+ semester, 40-90 students
- **Very popular courses**
 - ▶ Buzzword “Peer-to-Peer” attracts students

Aims/Goal of a course on „Peer-to-Peer“

- **Aims/Goals of a course on „Peer-to-Peer“**
 - ▶ Discussing: „What’s behind this buzzword?“
 - ▶ Demystifying the popular view of „Peer-to-Peer“ (not only file-sharing)
 - ▶ Designing massively scalable systems
 - Which implies decentralized control/coordination/resources
 - P2P is a valid paradigm (Client/Server does not solve everything)
 - ▶ Showing the applicability/possibilities of P2P approaches
 - Link to reality, e.g. application-layer multicast

Topics of a Peer-to-Peer course (examples) – I

- **Jussi's course:**

- ▶ Currently deployed P2P systems
 - file-sharing, eMail, skype, SETI, FreeNet, Wikipedia
- ▶ DHTs (usual suspects)
- ▶ Storage, file-systems
- ▶ P2P-Applications
 - P2P-DNS, P2P-eMail
- ▶ Replication: Reliability, performance
- ▶ Misc. stuff (security, anonymity, discussion of Napster-case)

- ▶ ... then GRID Computing (1/3 of the course)

Topics of a Peer-to-Peer course (examples) – II

- **Klaus' course:**

- ▶ Motivation

- (Examples: What is P2P, what is not P2P → Definition of P2P)

- ▶ Unstructured P2P systems:

- Napster, Gnutella, FastTrack

- Random Graphs, Small Worlds, Scale-Free Networks

- ▶ Structured P2P systems

- Comparison of approaches: C/S, Flooding, DHTs

- Usual DHT suspects

- ▶ Applications

- Multicast, anonymity, RON, SOS, i3

- ▶ Misc (legal issues, skype)

Topics of a Peer-to-Peer course (examples) – III

- **Rolland's course:**

- ▶ Introduction (Internet was P2P, migrated to a C/S system, Usenet)
- ▶ Unstructured P2P systems
 - Napster, Gnutella, Kazaa, edonkey, bittorrent, Freenet
- ▶ Structured P2P systems
 - CAN, Chord, Tapestry, Kademlia
- ▶ Security issues
 - SOS, Index poisoning, sybil attacks, fair sharing
- ▶ Multicasting
 - IP Multicast (for comparison), ALM (Narada, TBCP, DHT-based)
- ▶ Ad-hoc networking:
 - Why AHN is P2P, usual suspects
- ▶ P2P-based network management
- ▶ P2P platforms (JXTA)
- ▶ Misc (skype, ...)

Recommended Structure (to be extended by audience!)

- **Structure of P2P course is pretty obvious:**
(Topics that should not appear in other courses)
 - ▶ Motivation, Definition, Examples
 - ▶ Unstructured P2P systems
 - ▶ Structured P2P systems
 - ▶ Applications / Services
 - ▶ Misc.

- **Should be included (if course is P2P only):**
 - ▶ Overlay networks
 - Cross-layer issues
 - ▶ Small Worlds, etc.

Further Issues (to be extended by audience!)

- **Ad-hoc networking:**
 - ▶ Is it P2P or not? (depends on definitions of P2P)
 - ▶ Topics is definitely worth to be discussed (if room)
- **Algorithms**
- **Performance issues**
 - ▶ P2P traffic, P2P caches
 - ▶ Queueing theory
- **Legal issues**
- **Security**

Exercises (to be extended by audience!)

- **Written exercises**

- ▶ Read a paper and answer questions
- ▶ Applying approaches,
 - e.g. filling a DHT with nodes-ids, data, ..
- ▶ Text questions, questioning knowledge
- ▶ Design a certain application and discuss different approaches

Labs and Projects on Peer-to-Peer

- **Examples:**
 - ▶ Programming tiny P2P systems (e.g. simple Chord, gnutella, ...)
 - ▶ Design and (program) fancy i3-applications
 - ▶ Design and program simulation models of P2P systems
 - ▶ (Audience?: P2P simulators for visualization)

Ideas for Labs

- **Design, Implementation and Evaluation of a Peer-of-Peer-Applications**
 - ▶ Design phase
 - ▶ Presentation of approaches
 - Feedback (especially feasibility)
 - ▶ Implementation and deployment (e.g. PlanetLab)
 - Audience? Experience
 - Alternative: Simulation
 - ▶ Analyzing performance, ...
- **Should all groups get the same application?**
 - ▶ + Groups can be compared (award for best design, best performance)
 - ▶ + Discussion of different approaches (learning by doing)
 - ▶ + Interoperability may increase the students' attention
 - ▶ - ...but it is hard to achieve, and the students may lose the big picture
 - ▶ -

Example

- **Students should implement an application**

- ▶ Design is open but should follow what they learned in the course

- ▶ E.g.:

- Design a system that streams a soccer championship game from one streaming server (UDP stream) to up to 10 millions of clients

- Streaming of a live game

- Time difference between real game and the clients is at most 20 seconds

- No infrastructure service can be used

- Only application-level programming, no change of Internet routers

Open questions

- **Is Planet Lab a valid platform for student labs?**

Seminar Book: Experiences

- Is the book useful for courses?
- Are enough slides available? 😞