

IoT/P2P Project



Niels Olof Bouvin

The Project part of the course

- **Form**

- Fewer lectures (at most one per week)
- Most of the time spent working on your project
- Your project is self determined, pending my approval
 - so start thinking about cool systems!
- Two weeks before course end: Code freeze and Show & Tell
- Last two weeks spent writing the Report
- Exam: An oral, individual defense of your system and report, supplemented with a randomly chosen topic (topics being known in advance)

Remaining course plan

- **Week 39: Introduction to the project part**
 - You develop a *smashing* idea and get it approved
- **Week 40-48: Planning, coding, writing, coding...**
 - groups have weekly meetings with Michal and/or me at Doodle determined times
 - there will be milestones for report chapters with peer-review between groups
- **Week 48: Code freeze & Show and tell**
 - in front of everybody—mandatory participation
- **2018-12-7: Turn report and system in**
 - report *quite* important at the exam

Project Work

- **Starts as soon as you finish the mandatory assignment**
- **You are free to choose any topic, provided that**
 - there *must* be a strong element of IoT or P2P in your proposal (and that I approve it)
 - you should incorporate both your Raspberry Pi and the Cloud in your system
 - no restrictions on technology or choice of frameworks (as long as you make a Δ)
- **You will be expected to build a system, posit hypotheses, perform experiments, and reflect and conclude upon them**
 - in the form of a written report using the provided template and an oral defence
- **Show'n'tell: Demonstration of your system before all**

Suggestions for topics for P2P projects

- **Communication (text, audio, video, ...)**
 - multi-casted?
- **Backup**
- **Collaborative “spaces”**
- **File sharing**
- **(Domain specific) software frameworks**
- **Security?**
- **Games (cheat prevention etc.)**

Suggestions for topics for IoT projects

- Sensing phenomena
- Actuating devices and actions
- Tracking of persons and entities
- Situated interaction through devices
- Using open data in an interesting fashion
 - <http://www.opendata.dk>
- Applying machine learning to the IoT
- Coordination between devices
- Continuing the WoT work well beyond Chapter 9

Earlier projects

- **Locality aware BitTorrent**
 - always prefer pieces from local network rather than WAN
- **BitTorrent for streaming**
 - time critical piece selection
- **Synchronised playback over mobile devices**
 - for a discreet dance club
- **Secure social networking**
 - conspirators confiding confidentially
- **Cheat prevention in a MMOG**
 - no central authority here

More earlier projects

- **Multicast music streaming**
 - experiments with WebRTC
- **IoT visual programming**
 - making home configuration easy
- **Maintaining constant indoor climate**
 - more sensors, actuators, and machine learning
- **Mobile P2P photo sharing**
 - no need for central storage
- **Secure Chord**
 - like S/Kademlia, but for Chord

Additional earlier projects

- **Live score keeping for judges at sporting events**
 - RPis with touch displays arranged in mesh-network and with security
- **Zeroconf for IoT**
 - making adding a new device trivial
- **Adding HUDs (on phones) to Hardball**
 - like paintball, but with less mess and more pain
- **Tracking a cat in a domestic setting**
 - using BT beacons and mobile phones
- **Distributed card or dice game**
 - how is cheating prevented?

A Streaming Tree Approach To WebRTC Video Chatting

- Using the browser-based WebRTC protocol as a starting point, the project involved creating a decentralised social network based on a tree topology

Decentralised Backup for Everyone

- The project created a P2P solution for distributed backup between users based on an extensive related work investigation, and did involved performance testing on tuning efficient ways of storing and distributing files

Location Aware Audio Streaming

- The project involved building a distributed system using Bluetooth LE to track users in a home and to offer synchronized playback of music on distributed speakers

Sblocktify: Distributed Shared Playlists

- The project used a consensus-by-bet blockchain to implement a shared Spotify playlist between users, where voting and popularity affects the ability of the users to add new songs

A Smart Home Room Detection System & An implementation performing Music Synchronization

- Using ultra sound detectors to determine users entering and leaving rooms, as well as developing techniques to synchronise music playback across different rooms

IntelliTune: A Context-Aware Internet of Things Music System

- **Adapting the music selection and volume to the users present across a home**
- **Users' position determined using Bluetooth LE and ultrasound**

A S-Kademlia Simulation System and File Storage Implementation

- S-Kademlia was implemented as well as a system to simulate different types of attack to evaluate the effectiveness of the counter-measures of S-Kademlia

RDMP: A Music Player for Parties with a Distributed Library

- RDMP was a decentralised system that could dynamically discover music on participants' devices, storing the metadata in a Kademlia network supplemented with gossiping. Playlists were created collaboratively, and playback occurred at a designated device
- The system was shown to scale up to 50 participants

Romeo: Distributed Backup on a Local Area Network

- The Romeo system supported P2P backup with a special eye on implementing and evaluation policies to handle peers with wildly available disk space fairly

Choosing a fundamental

- You are free to choose any base technology
 - of course, you still have produce a sizeable Δ in your system
- This includes programming language, system platform, and framework
 - I should still be able to run your system
- Does the framework actually do what you need?
 - you need to know this ASAP, not just believe it based on the framework's website
- Using a framework: pro and con
 - it might give you what you need quickly and easily
 - you might end up having to debug somebody else's code (and try to get them to fix it)

The JXTA framework

- Do not under any circumstances use this

FreePastry

- **An officially sanctioned Java implementation of Pastry**
 - <http://www.freepastry.org/>
 - (last updated 2009)
 - Has been successfully used by students

Open Chord

- An open source Java implementation of Chord
 - <https://sourceforge.net/projects/open-chord/>
 - (last updated 2013-04-23)
 - Has been successfully used by students

Kademlia

- **Several implementations in diverse languages**
 - MaidSafe: <https://github.com/maidsafe/MaidSafe-Routing> (C++) (2015-3-25)
 - Entangled: <http://entangled.sourceforge.net> (Python) (2012-5-18)
 - (TomP2P: <http://tomp2p.net> (Java) (2015-5-27) Has led to much frustration)
- **Plenty more, see**
 - <http://en.wikipedia.org/wiki/Kademlia#Libraries>
 - search on GitHub
- **But... why not use your own?**

Serval Project

- Enables Android phones to use Ad-Hoc WiFi with each other, while also using 3G
 - <https://www.servalproject.org>

Ongoing, OSS, P2P & other Projects

- **IPFS: Interplanetary File System**
 - a system to create the distributed Web, sort of BitTorrent, Kademlia & git in one
 - <https://ipfs.io>
- **Dat Project**
 - a data sharing protocol with similar characteristics
 - <https://datproject.org>
- **TensorFlow is available on the Raspberry Pi**
 - machine learning toolkit—could be useful to make sense of sensed data
 - <https://www.tensorflow.org/install/>
- **OpenCV (camera/image analysis) also works on RPi**

BitTorrent

- **MonoTorrent (C#)**
 - <https://github.com/mono/monotorrent/> (2018-8-9)
- **Ttorrent (Java)**
 - <https://mpetazzoni.github.io/ttorrent/> (2018-9-24)
 - highly rated by students

P2P through the Web Browser

- **WebRTC — supported in many modern browsers**
 - designed for direct audio/video streaming but can be used for other things as well
 - <http://www.openwebrtc.org>
- **WebTorrent — BitTorrent in the browser**
 - <https://webtorrent.io>

Other systems and frameworks...

- **NS3**
 - a very large, highly complicated, and powerful MANET simulator (still under development)
 - experience suggests that 7 weeks is a short time to get to grips with it...
- **See this discussion on Hacker News:**
 - <https://news.ycombinator.com/item?id=11144921>

IoT platforms

- You can, of course, just continue with Raspberry Pi
- Alternatives are plentiful, including Arduino, ESP, etc.
- Don't forget about mobile devices!
 - Android/IOS devices certainly qualify as interesting platforms
 - There has to date been no successful use of WiFi-Direct, so beware

IoT software platforms

- **<https://home-assistant.io> (Python 3)**
- **<http://www.openhab.org> (Java)**
- **<http://nodered.org> (Node.js)**
- **<https://www.cesanta.com/products/smart-js> (C++,js)**

Cloudy places to run your system

- **Students have previously used Digital Ocean**
 - it is often free for students—there is also a GitHub student package that includes it

ORBIT Lab

<https://orbitlab.au.dk>

- Will be presented next Tuesday by the Orbit Lab Manager, Simon Christensen
- Can provide a space to work, plenty of equipment, and lots of inspiration
- Situated just across on the block on Finlandsgade

Smart Aarhus

<https://www.smartaarhus.eu>



About Smart Aarhus

News

Projects

Sharing economy

Contact

Danish

Eduroam

Aarhus City Lab

The purpose of the Aarhus City Lab is to be a test facility for Smart City solutions and a showroom for new smart city initiatives.

Groups

- You do your project in groups
- All email communication must include group id

Your project proposal

- **Your project proposal should consist of at least**
 - a use case – a short story to illustrate the why & how
 - a description of how and why IoT/P2P will be used
 - a sketch of the envisioned architecture
 - some rough and realistic weekly milestones
- **All this in 1-2 pages (PDF, please!)**
- **Due: The sooner, the better. Preferably early next week. I'll then approve the approvable ASAP**
- **I'll be at my office from 12-17, Thursday, so drop by**

Your system

- **You have a limited time, so you can't polish your system too much**
 - a fancy GUI is not a priority
- **Functionality is paramount; form less so**
- **Reproducible functionality is a requirement**
 - ensure that you are creating a system where you can run repeated, quantifiable tests (like graphs with error bars)
 - your system should be easy to setup and its intended use well-documented
- **Statistics and proper graphs are important tools**
 - use "Think Stats: Probability and Statistics for Programmers" (it's free!)
 - <http://www.greenteapress.com/thinkstats/> (you don't have to read the whole book)

Your project: Preparing for the Report

- Given the project nature of the course, it is easy to focus solely on coding, coding, coding
 - this would be a *grave* mistake
- Even though there will be no official syllabus for the course, it will go hard for a report without a respectable list of references (at least 10 references)
- Therefore: Research your chosen project topic and find related work as you develop your ideas
 - who knows? Could turn out to be a help along the way...
 - <http://scholar.google.com/>

Report requirements

- **There is a template, and you *must* use it *and* follow it**
- **Your report should contain the standard elements**
 - introduction, related work, analysis thereof, system design & implementation, experiments/evaluation, conclusion, (future work), references
- **A description of intended use & a link to a video demo**
- **A comprehensive manual in an appendix**
 - your system is not self-explanatory...
- **A clear distinction between planned and actual functionality**
 - this should be in the introduction

So what makes a great report?

- Innovative/exciting use case
- Well documented positioning with regards to related work
- Impressive system with a good description
- Thought-out and systematic evaluation/experiments
- Well reasoned analysis of results

Exam

- **30 minutes per person**
- **Prepare your personal presentation of the report**
 - 12 minutes, then a few minutes for questions
 - 10 minutes to present a randomly chosen topic from the first part of the course
- **Slides are welcome (for the project part)—demos OK, if you think you have the time**
- **Your grade is based on the work, the report, and your presentation**