Introduction to
Building the **IoT** with **P2P** & ☁️

Niels Olof Bouvin

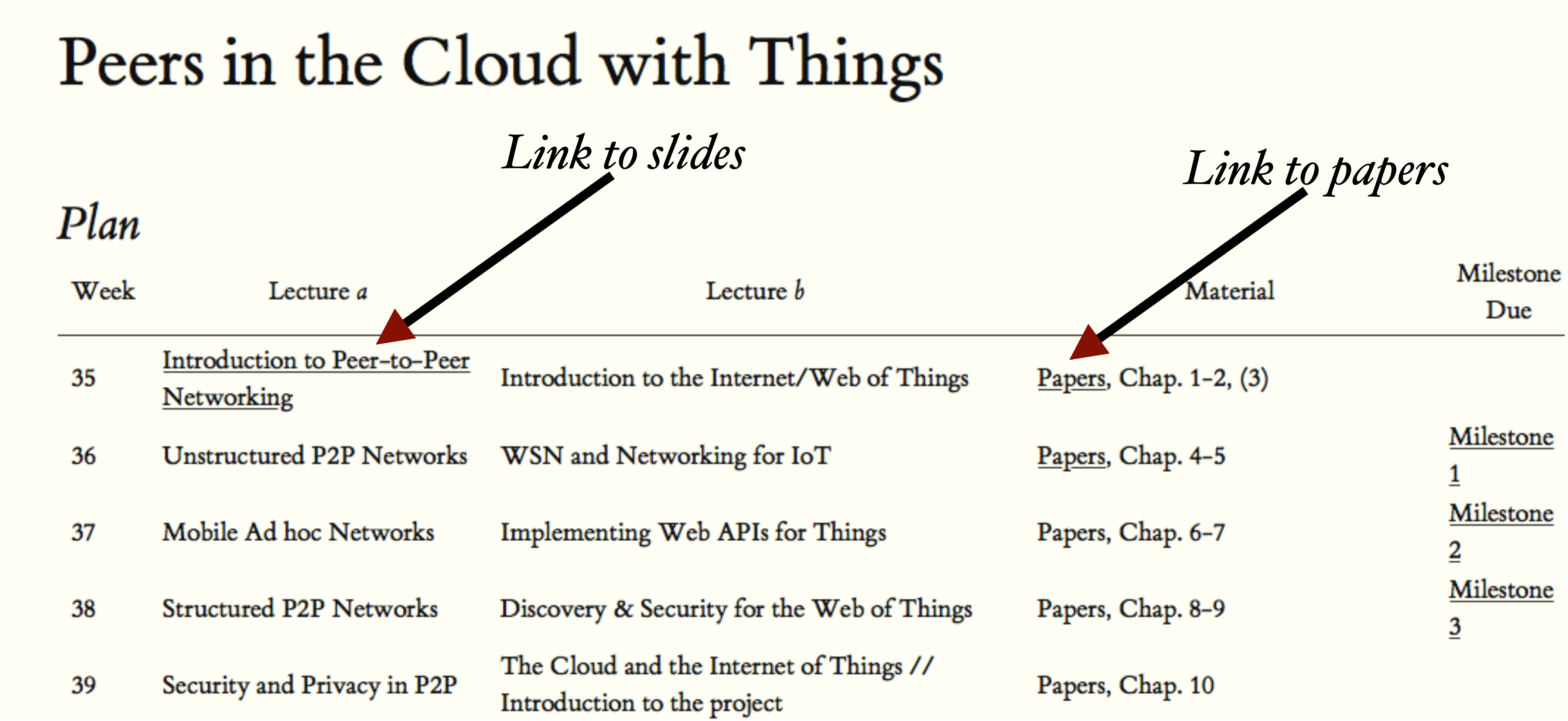
Overview

- Introduction to the course
- Introduction to Peer-to-Peer networking
- Client/server compared to Peer-to-Peer
- P2P characteristics
- Typical P2P applications
- Gnutella
- Summary

Course Overview

- **Blackboard (¯_(ツ)_/¯)**
  - (send me an email <bouvin@cs.au.dk>, if you are not on the list)
- **Course material**
  - papers & technical reports found in the BB system
  - *Building the Web of Things* by Dominique D. Guinard & Vlad M. Trifa
- **Group work (2-3 persons)**
  - First four weeks of the course: Mandatory assignment (first milestone next week)
  - Rest of the course: Self-determined IoT/P2P project (more about this on later)
- **Exam: Oral, 30 minutes, known questions & project**
  - based on written report documenting and reflecting upon project work

Course Homepage



Peers in the Cloud with Things

Link to slides

Link to papers

Plan

Week	Lecture *a*	Lecture *b*	Material	Milestone Due
35	Introduction to Peer-to-Peer Networking	Introduction to the Internet/Web of Things	Papers, Chap. 1–2, (3)	
36	Unstructured P2P Networks	WSN and Networking for IoT	Papers, Chap. 4–5	Milestone 1
37	Mobile Ad hoc Networks	Implementing Web APIs for Things	Papers, Chap. 6–7	Milestone 2
38	Structured P2P Networks	Discovery & Security for the Web of Things	Papers, Chap. 8–9	Milestone 3
39	Security and Privacy in P2P	The Cloud and the Internet of Things // Introduction to the project	Papers, Chap. 10	

Purpose of Course

- To familiarise you with decentralised sensing systems
- To introduce a number of design criteria for P2P as well as Web-based Internet of Things networks
- To teach you to assess the strengths and weaknesses of a given system, based on these criteria
- To establish practical knowledge of IoT/P2P networking by constructing a Web based sensing system with a resilient cloud based delivery & storage mechanism from scratch, and, based on these gained skills, create your own project system

Topics

- Introduction to P2P
- Unstructured P2P systems
- MANET
- Structured P2P systems
- Security & Privacy
- P2P Applications
- BitTorrent

- Introduction to IoT & WoT
- Embedded systems
- Networks for IoT
- Discovery & Security
- Cloud, IoT, and P2P
- IoT Applications
- P2P Streaming
- The Blockchain
- Distributed Web Platforms

Time Overview

- **First half of course**
    - Two lectures every week to give you a crash course on P2P/IoT
- **Second half of course (after autumn break)**
    - One lecture every week to give you more time to focus on project work
- **First four weeks:**
    - a mandatory assignment of three milestones
- **Remainder of course**
    - a self-determined project with a written report and weekly milestone meetings

Mandatory Assignment

- **Follows the WoT book, so more about it on Friday**
- **Will require you have a Raspberry Pi + sensors**
- **This will, hopefully, be made available for sale Friday**
  - announcement will follow
- **Three milestones, using the RPi as well as the cloud**
- **Sadly, no P2P component this year**
  - on the bright side, this leaves you with more time for the project
  - and I do have some suggestions about adding a P2P component as a project. . .

Project Work

- **Starts as soon as you finish the mandatory assignment**
  - but ideally well before the autumn break (week 42)
- **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 and an oral defence
- **Show’n’tell: Demonstration of your system before all**

People

- **Niels Olof Bouvin**
  - lectures (Tuesday & Friday)
  - and the occasional milestone meeting
- **Michal Ratajský**
  - TA
  - milestone meetings

- **TA: Thursday, 15-18**
- **Milestone meetings: Thursday, 11-17**
  - during the project part of the course

Administratrivia

the creation of groups

- **Divide yourself into groups (2-3 persons)**
  - create a matching group using the magic of Blackboard
- **The mandatory assignment consists of three milestones (and starts next week)**
- **The project work occupies the remainder**
- **Progress on project work to be presented to Michal**
  - Thursday 11-17
  - I’ll create a Doodle for scheduling

On the Nature of Groups

- **Finding a group: there is a subreddit for the course**
- **Finding the right members**
  - your project work is a significant part of your final, individual grade, so it is important that a group agrees on a plan and the level of ambition from the start
- **All group members are expected to contribute equally**

Overview

- Introduction to the course
- **Introduction to Peer-to-Peer networking**
- Client/server compared to Peer-to-Peer
- P2P characteristics
- Typical P2P applications
- Gnutella
- Summary

What are P2P Networking Systems?

- **There are a fair number of different opinions:**
  - "A server-less networking technology that allows several network devices to share resources and communicate directly with each another"
  - "Deployable in an ad-hoc fashion, without requiring centralised management or control"
  - "Peer-to-peer is a class of applications that take advantage of resources—storage, cycles, content, human presence—available at the edges of the Internet"

Defining Characteristics for P2P

- Resources are shared directly between peers
- Activities are (largely) coordinated between peers
- The peers are capable of handling contingencies

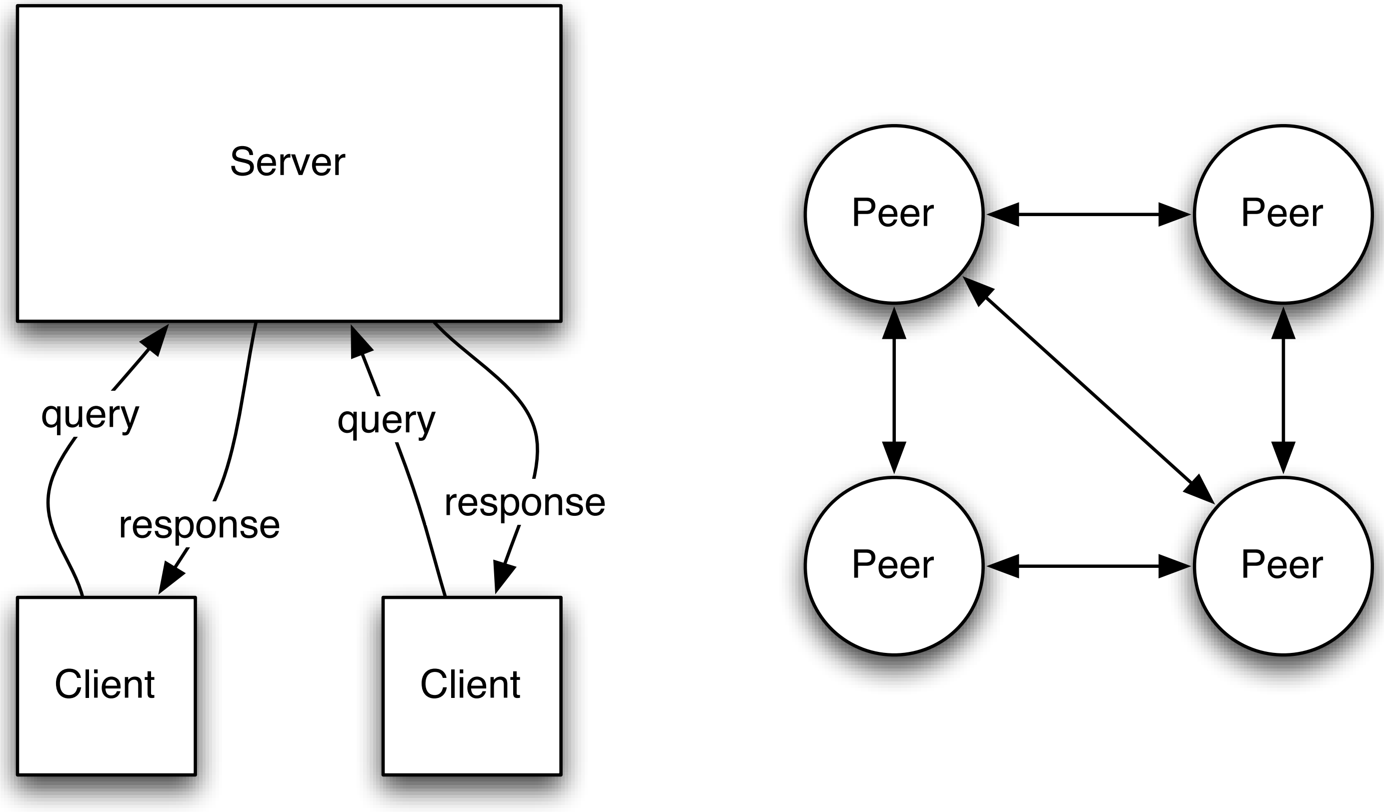
A Brief History of P2P Computing

- **1969-1995: The original Peer-to-Peer Internet**
  - No firewalls, most services widely available
  - Usenet: based on Unix-to-Unix-CoPy. DNS: hosts are clients and servers, cache replies
- **1995-2001: The Internet explosion (and implosion)**
  - Movement away from P2P to client/server models
  - Web, firewalls, ADSL, asymmetric connections, NAT, ...
- **2001-...: New wave of peer-to-peer**
  - separating authoring from publishing; (Web) service oriented Internet; distributed media publishing; BitTorrent; P2P streaming; tor; federation of social networks
- **Now and onwards:**
  - The rise of the internet connected device/sensor: Will the edge overwhelm the center?

Overview

- Introduction to the course
- Introduction to Peer-to-Peer networking
- **Client/server compared to Peer-to-Peer**
- **P2P characteristics**
- **Typical P2P applications**
- **Gnutella**
- **Summary**

Client/Server vs. P2P



Client/Server

Advantages

- Centralised
- Increased security
- Control
- Easy to maintain
- Static topology
- State kept in one place

- Simple architecture
- Scalable (only few resources on client)
- Well known and well supported
- Loose coupling between client/client

Client/Server
Disadvantages

- **Single point of failure**
  - much less of an issue with a well designed cloud service
- **Scalability is costly**
- **Large bandwidth requirements at server**
- **State kept in one place**
- **Central control (and sometimes, central *lack* of control)**
- **Does not take advantage of the resources of the clients**
- **Collaboration between clients involves the server**

Peer-to-Peer
Advantages

- Robust
- Scalability
- More clients = more available resources
- Dynamic (self configuring)
- Replication
- Decentralised (autonomy)
- Peers can collaborate directly
    - if designed well with low latency due to closeness

Peer-to-Peer
Disadvantages

- Architectural complexity
- Churn: Peers joining and leaving
- Resources are distributed and not always available
- Harmful content/systems can be hard to eradicate
- More demanding of peers
- New technology: abstractions, techniques, etc., are not as mature

Client/Server vs P2P vs Cloud

In Practice

- **No need to pick only one, when you can use all**
- **Most successful P2P incorporate client/server elements, often with a server hosted in the Cloud**
  - often for bootstrapping purposes
- **Cloud computing permits high scalability**
  - at a cost depending on your requirements
  - but it is much simpler to achieve scalability here than with P2P systems
  - Cloud solutions often involve P2P computing at the hosting center level

Overview

- Introduction to the course
- Introduction to Peer-to-Peer networking
- Client/server compared to Peer-to-Peer
- **P2P characteristics**
- **Typical P2P applications**
- **Gnutella**
- **Summary**

Classes of P2P Architectures

- **Purely decentralised architectures**
  - All peers have the same basic capabilities and offer similar services
- **Partially centralised architectures**
  - Some, usually more powerful, peers will accept more demanding roles on an ad hoc basis
- **Hybrid decentralised architectures**
  - Some central servers facilitate coordination

Degrees of P2P Structure

- **Unstructured networks**
  - Peers connect in a more or less haphazardly way – resulting in a network graph either power-law or random. Routing/searching is ad-hoc or based on heuristic
- **Semi-structured networks**
  - While the network is still relatively random, resources are placed so that efficient routing works
- **Structured networks**
  - Peers and resources are placed according to a rigidly defined schema, which is maintained over the lifetime of the network

P2P Characteristics

Scalability

- **The ability of a system to support an increasing use**
- Pro: Network, storage, computational power of peers may be leveraged
- Con: Routing, location, synchronising may not scale; “fat” clients needed; peers must contribute

P2P Characteristics
Performance

- **The time it takes for a system to react to a stimulus**
- Pro: Data and computation may be close to peers, high degree of distribution
- Con: Replicated, distributed state and computation; complex architectures

P2P Characteristics
Availability

- **The part of the deployment period during which a system can deliver the services it implements**
- Pro: No single point of failure/robustness; system may be self-configuring, replicated, autonomous
- Con: Ensuring consistent availability; having knowledge of network state

P2P Characteristics
Fairness

- **Distributing work equally across the peers according to their needs and abilities**
- Pro: Necessary in order to maintain the good performance of P2P
- Con: Difficult to ensure

P2P Characteristics

Integrity and authenticity

- **The ability of a system to maintain correct state**
- **Pro: State is distributed, so it can not *all* be corrupted**
- **Con: Cryptographically authenticated security difficult to establish without central authority**

P2P Characteristics

Security

- **The degree to which a system can withstand attacks**
- Pro: Robustness against Denial of Service attacks; anonymity
- Con: Complex, decentralised security architecture

P2P Characteristics

Anonymity, deniability, censorship resistance

- **Being able to retrieve or publish information without risk of discovery**
- **Pro: Adds security, difficult to suppress information**
- **Con: Not easy to ensure, what if running the system becomes a crime? Should *all* information be freely and anonymously available?**

Overview

- Introduction to the course
- Introduction to Peer-to-Peer networking
- Client/server compared to Peer-to-Peer
- P2P characteristics
- **Typical P2P applications**
- **Gnutella**
- **Summary**

Typical P2P Applications

- Content sharing and distribution
- Communication
- Collaboration
- Distributed computation

P2P Content Sharing

- Users want to share information. The bandwidth/ storage of users' computers are often partly unused
- Napster
  - MP3 file sharing and searching service
  - Hybrid P2P solution: Files on peers, metadata on server
- Gnutella
  - The original "proper" P2P application for file sharing

P2P Content Distribution

- **A secure way of distributing content – focus on creating a “fabric” for distribution**
- **Freenet**
  - Anonymous censor resistant information distribution,
  - Completely decentralised P2P solution
  - Routing based on key “closeness” to target; routing replicates data
- **BitTorrent**
  - Popular system for legal (and illegal) content sharing
  - Designed to encourage fairness
- **IPFS**
  - combines Kademlia (a structured P2P network), BitTorrent with Merkle tree IDs

P2P Communication

- **Communication is essential in collaboration and communication need only be between peers**
- **ICQ (and many others)**
  - Pioneered instant messages in 1996
  - ICQ server maps from ICQ-specific addresses to IP addresses; communication is P2P
- **Skype (original version)**
  - Voice over IP & instant messagingbased on P2P routing
  - Supported API for third-party development
- **Mastodon**
  - Federated social networking (“decentralised Twitter”)

P2P Collaboration

- **Human collaboration is intrinsically peer-to-peer, this can be reflected in software architectures**
- **Groove**
  - Commercial peer platform for hosting peer-to-peer applications and business solutions
  - Very elaborate security model
  - Now owned by Microsoft
- **Distributed Knight (local PhD project)**
  - P2P collaboration on (UML) diagrams
  - Tight integration with context-aware instant messaging system

P2P for Distributed Computation

- **Powerful desktop machines most often under-utilised**
- **Typical model:**
  - Central server divides "embarrassingly parallelisable" problem into manageable chunks and distributes these to clients which perform computations and return results
  - In principle similar to clusters
- **True P2P?**
  - Sort of... most projects are strictly hierarchical in architecture
- **Cyber foraging**
  - letting (mobile) devices use locally available computing resources on the fly

Overview

- Introduction to the course
- Introduction to Peer-to-Peer networking
- Client/server compared to Peer-to-Peer
- P2P characteristics
- Typical P2P applications
- **Gnutella**
- **Summary**

Gnutella

- **The first major truly distributed P2P file sharing system – a counterpoint to the SPoF of Napster**
  - Gnutella is fully distributed and cannot be easily be taken out by an attack (legal or otherwise)
- **Invented by Justin Frankel & Tom Pepper of Nullsoft**
  - most famous for creating WinAmp
- ***Very* quickly pulled by AOL/Time Warner**
  - at that point the source was “in the wild”, and a number of Gnutella variants have since developed
- **Quite primitive system, yet hugely successful**

Gnutella protocol:
5 commands are all you need

- **Ping**
  - used for discovery
- **Pong**
  - the response to a Ping
- **Query**
  - used for searching
- **QueryHit**
  - the response to a successful query
- **Push**
  - used to get fire-walled servents to reach outside the firewall

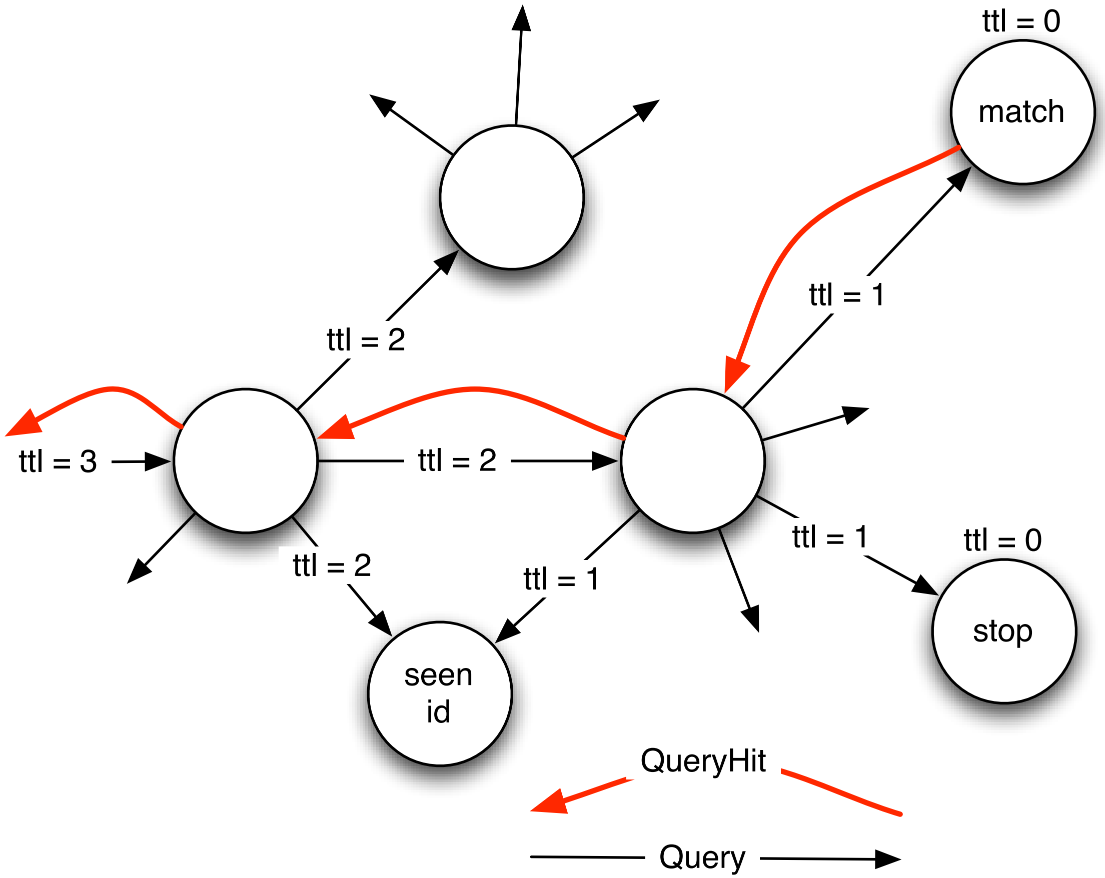
How does it work?

- A Gnutella peer starts out with a number of peers from “somewhere” (perhaps found on a Web page)
- It can ping these peers to receive information about them, and thus build a list of potential peers to contact in the future
- Pings and queries are sent to all known peers who in turn call all their peers and so on (flooding)
- Queries has a unique ID (128 bit) and a TTL (Time To Live). This ensures that peers do not retransmit the same query twice and that queries eventually die out

How does it work?

- Peers remember (for a limited time) received and transmitted queries and whence they came
- If a query match is found, the response (containing the query and the host address) is returned following the query route back to the originator
- The originator receives (presumably) a number of hits and can then contact a host directly for downloading (usually through HTTP)

A Gnutella Example



Ranking of Gnutella peers

- **Peers report**
  - amount of shared data
  - available bandwidth
- **Self-reporting is problematic**
  - claim your bandwidth is low, and you will be left alone

Gnutella is inefficient

- Flooding ensures that all peers within TTL horizon are contacted
- However, flooding generates a tremendous amount of (duplicate) network traffic
- Gnutella is so inefficient, that swamping the network becomes quite likely, *even without any data traffic*

Gnutella Calculations

	TTL=1	*TTL=2*	*TTL=3*	*TTL=4*	*TTL=5*	*TTL=6*	*TTL=7*	*TTL=8*
N=2	332	664	996	1328	1660	1992	2324	2656
N=3	498	1494	3486	7470	15438	31374	63246	126990
N=4	664	2656	8632	26560	80344	241696	725752	2177920
N=5	830	4150	17430	70550	283030	1132950	4532630	18131350
N=6	996	5976	30876	155376	777876	3890376	19452876	97265376
N=7	1162	8134	49966	300958	1806910	10842622	65056894	390342526
N=8	1328	10624	75696	531200	3719728	26039424	182277296	1275942400

Traffic (in bytes) generated by search for the string 'Grateful Dead Live' in a perfectly balanced Gnutella graph with variable TTL and #Neighbours per peer

Gnutella Experiences

- **Flooding hardly the most efficient use of network resources**
- **Downloads the whole file from a single peer**
    - So if that peer goes missing in the middle of your download… so does your data
- **Advantage of Gnutella: So abysmal performance, it spurred the development of a lot of improvements – more about this next time!**

Overview

- Introduction to the course
- Introduction to Peer-to-Peer networking
- Client/server compared to Peer-to-Peer
- P2P characteristics
- Typical P2P applications
- Gnutella
- **Summary**

Summary

- **The strength of P2P is in numbers**
  - Great number of unused processors
  - Large amount of unused bandwidth
  - Whole lot of storage

- **P2P systems can be built to increase**
  - Computing power
  - Data availability
  - Free speech

- **This involves significant challenges**
  - Routing
  - Searching
  - Churn
  - Security