Peer-to-Peer Systems and Security

Introduction

Christian Grothoff

Technische Universität München

April 7, 2014

"No one shall be subjected to arbitrary interference with his privacy, family, home or correspondence, nor to attacks upon his honour and reputation. Everyone has the right to the protection of the law against such interference or attacks." – Universal Declaration of Human Rights, Article 12

Peer-to-Peer Systems

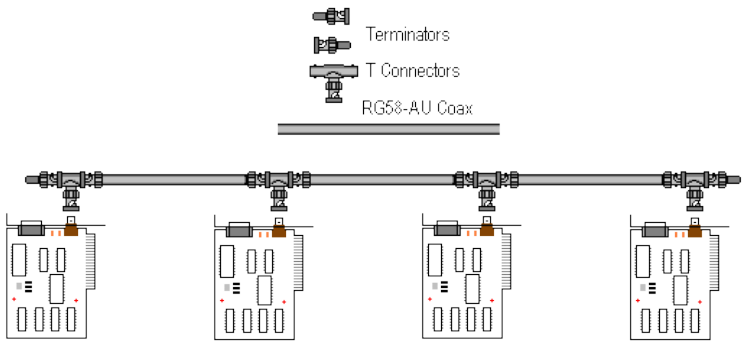
Definition:

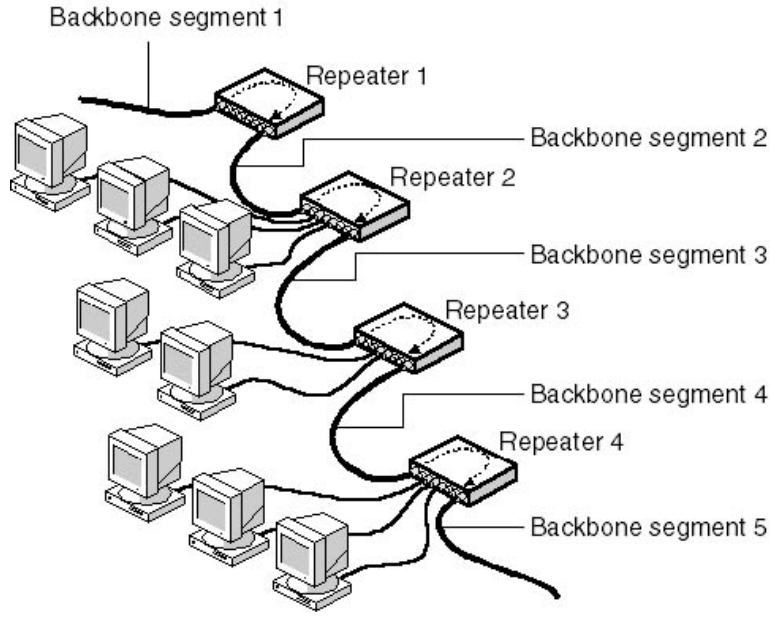
- A Peer-to-Peer (P2P) system is a system where participants work together as equals, with symmetric roles, rights and responsibilities.
- A *pure* P2P system is a P2P system where *all* (important) services are realized by peers.

This course is about P2P systems that use the Internet for communication between peers (also known as *overlay* networks).

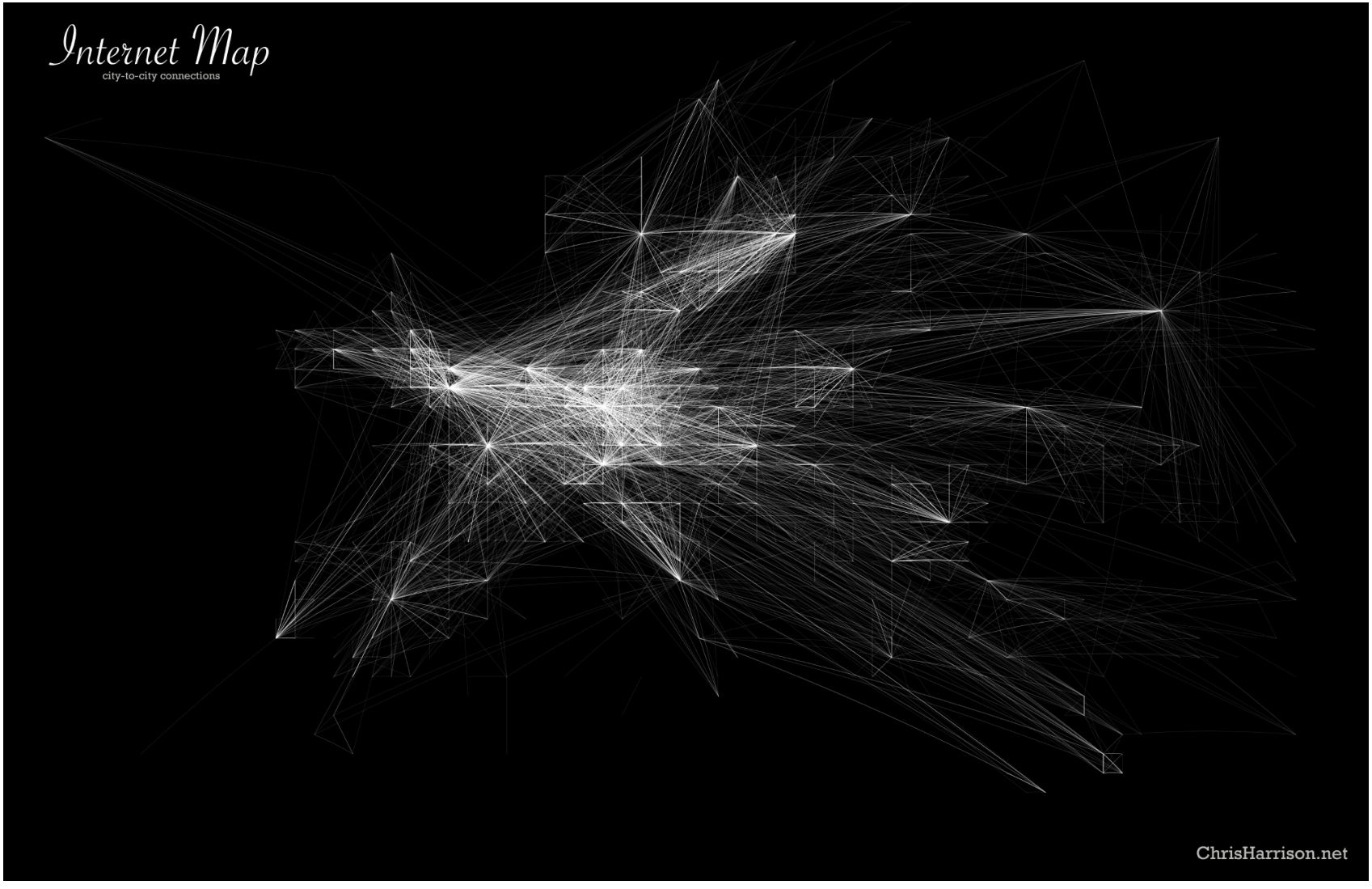
Famous P2P Systems: Democracy

Famous P2P Systems: Ethernet

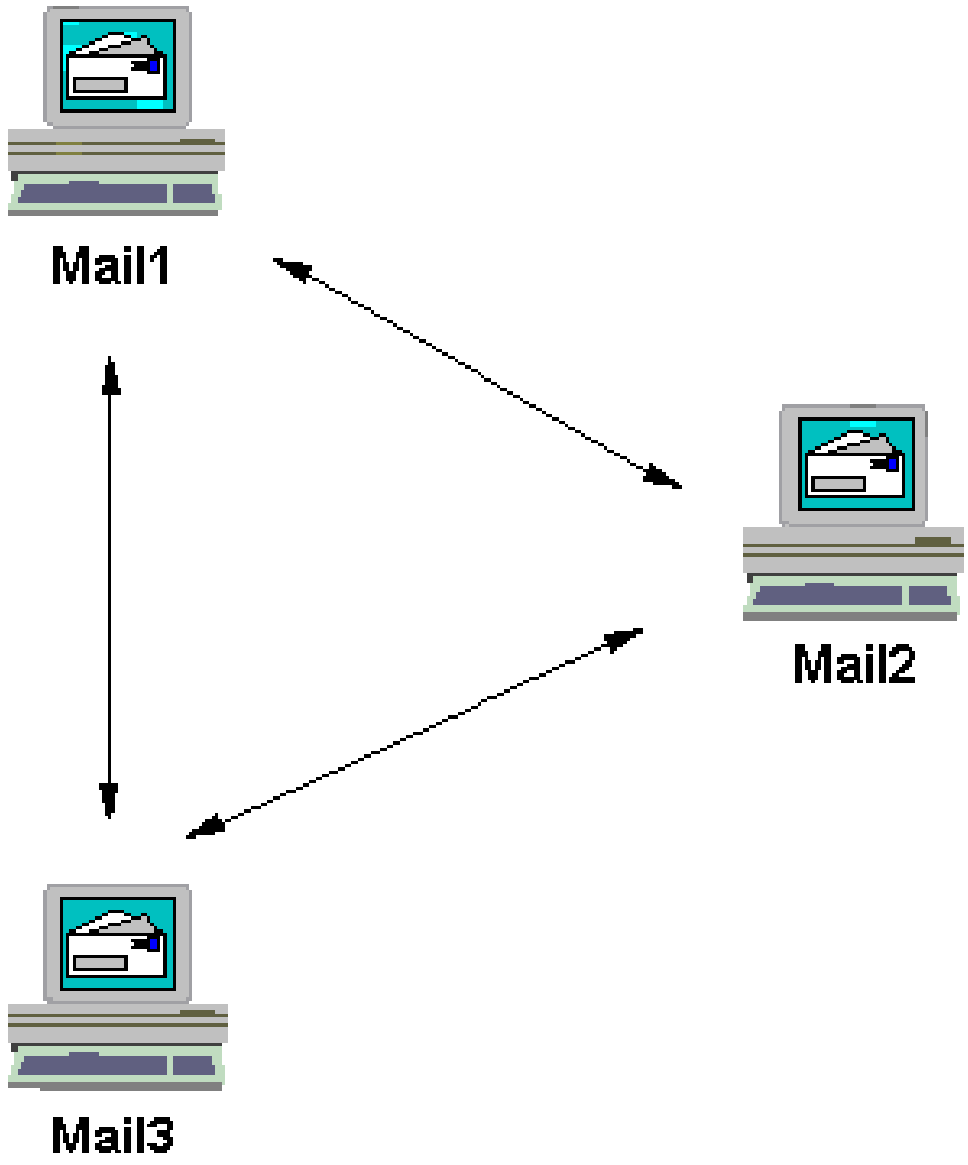




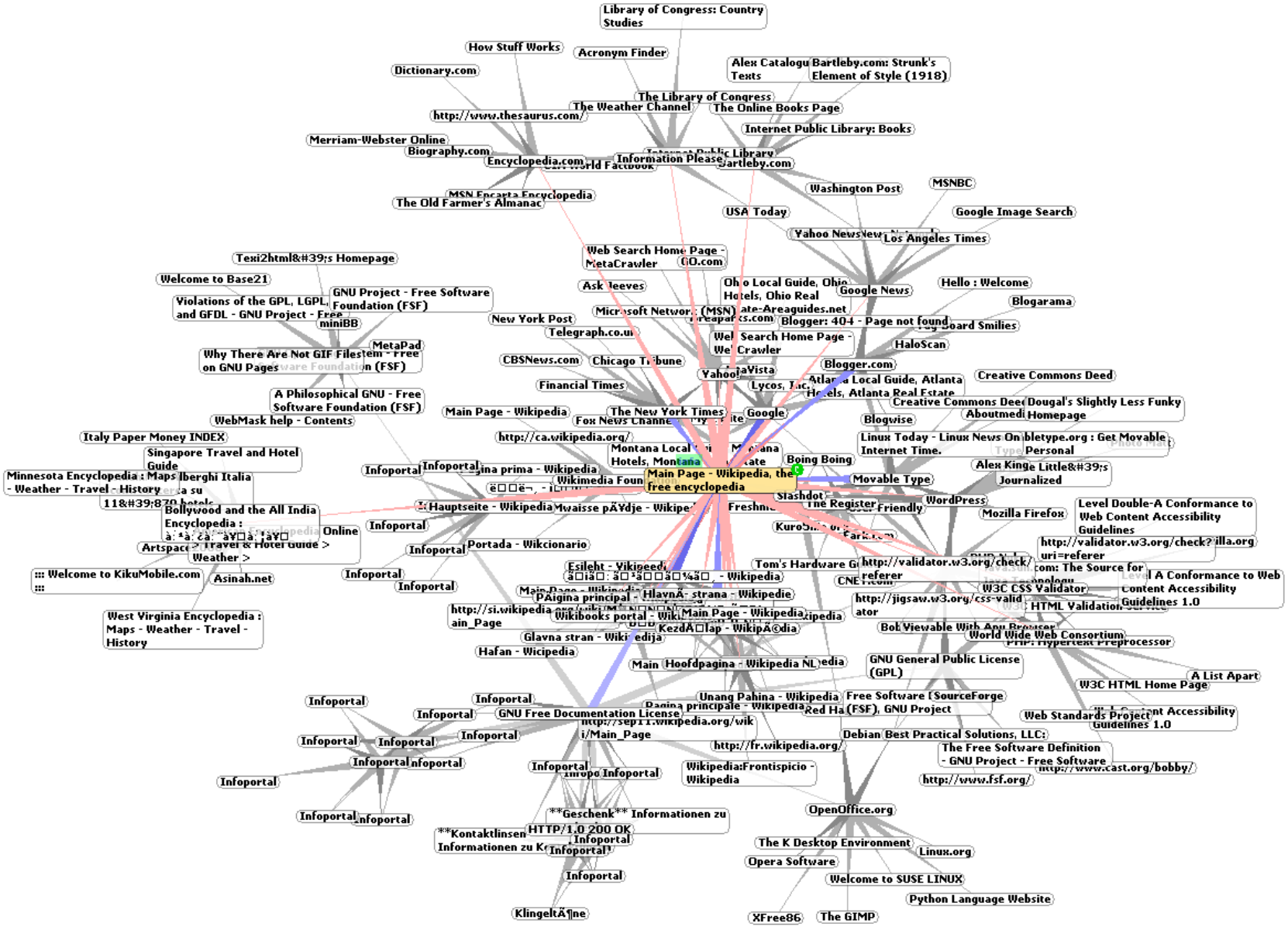
Famous P2P Systems: Internet (IP/BGP)



Famous P2P Systems: SMTP



Famous P2P Systems: World Wide Web



Famous P2P Overlay Systems

- Napster
- Gnutella
- Freenet
- Bittorrent
- Tor

Client-Server Benefits

Client-server architectures make it easy to:

- establish trust, and restrict access
- manage resources, and charge fees
- deploy updates, and remove features
- collect data, and sell it

Why not use Client-Server architecture?

If you use a server, you give up control of your:

- data
- computation
- free software

Why study *overlay* P2P Systems?

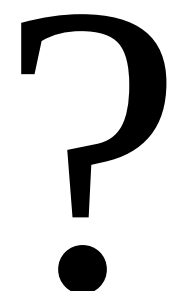
- Easier to develop and deploy
- Layered architecture: make use of existing abstractions
- Envision the future of the Internet!

Distributed Systems

An overlay P2P network is a distributed system. Deutsch formulated "The Eight Fallacies of Distributed Computing":

- The network is reliable
- Latency is zero
- Bandwidth is infinite
- The network is secure
- Topology does not change
- There is one administrator
- Transport cost is zero
- The network is homogeneous

Questions?



Learning Goals

In this course, you will learn about:

- Protocol design
- Distributed algorithms & data structures
- System programming
- Game theory / Reputation Systems
- Network security & privacy

Learning Methods

- Lectures on existing designs and implementations
- Study current research papers
- Present (and discuss) your own ideas
- Implement your own protocol / extension

Deliverables

- Quizzes
- Written reports (design document, progress report, final report)
- Individual presentation on group project
- Project code
- Final individual interview
- **NO** final exam

Details at
`http://grothoff.org/christian/teaching/2014/2194/`.

The Project

- Webiste gives *suggestions*
- Teams of one or two students
- One project-related presentation per student
- Joint project reports
- Individual interview

Using GNUnet for the project is a suggestion, not a requirement.

Schedule

- Introduction & GNUnet architecture
- Security & unstructured protocols
- Structured Routing Algorithms & NAT traversal
- Game theory & Anonymity
- Attacks & Evil P2P networks
- Visions for the future

Schedule

- Introduction & GNUnet architecture
- Security & unstructured protocols
- Structured Routing Algorithms & NAT traversal
- Game theory & Anonymity
- Attacks & Evil P2P networks
- Visions for the future

- Presentations

Schedule

- Introduction & GNUnet architecture
- Security & unstructured protocols
- Structured Routing Algorithms & NAT traversal
- Game theory & Anonymity
- Attacks & Evil P2P networks
- Visions for the future

- Presentations

- GNU Hacker Meeting (August 15-17)

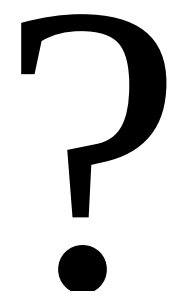
Project Ideas

- Random Peer Sampling (`https://gnunet.org/brahms`)
- Tor-like OR in GNUnet
- Cubit DHT [3] (or other "interesting" DHT [2])
- P2P over DNS, SMTP [5], SCTP, Satellite, ...
- NAT traversal [1]
- Socialist Millionair Problem protocol
- Distributed search engine [4]
- Asynchronous messaging
- Enhanced voice-over-IP (gnunet-conversation)
- M2M applications
- Distributed constraint optimization [6]
- ...

Remember

- Study assigned reading before each class
- Review previous lectures before each class
- Form teams, e-mail team information to Andreas Korsten
- Prepare design documents, first presentation due in 6 weeks!

Questions?



References

A. Müller and A. Klenk and G. Carle.

Behavior and Classification of NAT devices and implications for NAT-Traversal.

IEEE Special issue on Middleboxes, pages 14–19, September 2008.

Ioannis Aekaterinidis and Peter Triantafillou.

PastryStrings: a comprehensive content-based publish/subscribe DHT network.

In *Proc. 26th IEEE Int. Conf. on Distributed Computing Systems (ICDCS '06), Lisboa, Portugal*, page 23, 2006.

Aleksandrs Slivkins Bernard Wong and Emin Gn Sirer.

Approximate matching for Peer-to-Peer overlays with Cubit.

Technical report, Cornell University, Computing and Information Science, 2008.

Michael Christen.

Yacy.

`http://yacy.net/`, 2013.

Ronaldo A. Ferreira, Christian Grothoff, and Paul Ruth.

A Transport Layer Abstraction for Peer-to-Peer Networks.

In *Proceedings of the 3rd International Symposium on Cluster Computing and the Grid (GRID 2003)*, pages 398–403. IEEE Computer Society, 2003.

Pragnesh Jay Modi, Wei-Min Shen, Milind Tambe, and Makoto Yokoo.

Adopt: asynchronous distributed constraint optimization with quality guarantees.

Artificial Intelligence, 161(1–2):149–180, January 2005.