Peer-to-Peer Systems and Applications

Lecture 1: Introduction to P2P Systems

TECHNISCHE UNIVERSITÄT DARMSTADT

Chapter 2, 4, and 5:

Part I: P2P: Notions, Areas and
Part II: Unstructured P2P Systems

*Original slides for this lecture provided by David Hausheer (University of Zürich, Switzerland), Detlef Schoder (University of Cologne, Germany), Rüdiger Schollmeier and Jörg Eberspächer (Technische Universität München, Germany).

0. Lecture Overview

TECHNISCHE UNIVERSITÄT DARMSTADT

1. Introduction
   1. Definitions and Characteristics
   2. Classification and Properties
2. Application Areas
   1. Definition
   2. Classification
3. Centralized P2P Networks
   1. Definition
   2. Characteristics, and Topology
   3. Example: Napster
   4. Discussion
4. Decentralized P2P Networks
   1. Definition
   2. Characteristics, and Topology
   3. Example: Gnutella
   4. Discussion

TECHNISCHE UNIVERSITÄT DARMSTADT

1. Introduction

What Is This *"Peer-to-Peer"* About?
Definitions and Characteristics

1.1. Definitions and Characteristics (1)

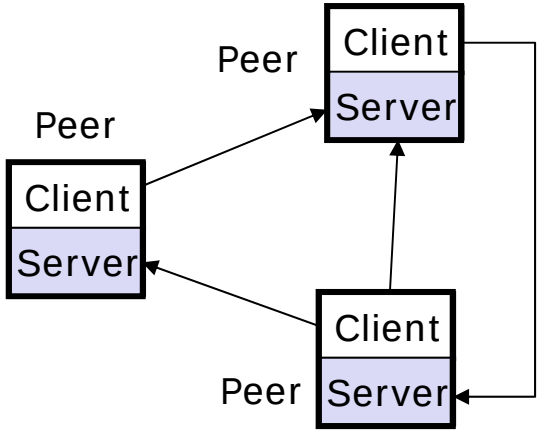
TECHNISCHE UNIVERSITÄT DARMSTADT

- Oram et al.:
  - A Peer-to-Peer (P2P) system is *„a self-organizing system of equal, autonomous entities (peers)* [which] *aims for the shared usage of distributed resources in a networked environment avoiding central services.“*
  - *„A system with completely decentralized self-organization and resource usage*.“
- Derived key characteristics of a P2P system:
  - Equality – All peers are equal (peer = gleichgestellt)
  - Autonomy – No central control
  - Decentralization – No centralized services
  - Self-organization – No coordination from outside
  - Shared resources – Peers may use resources provided by other peers

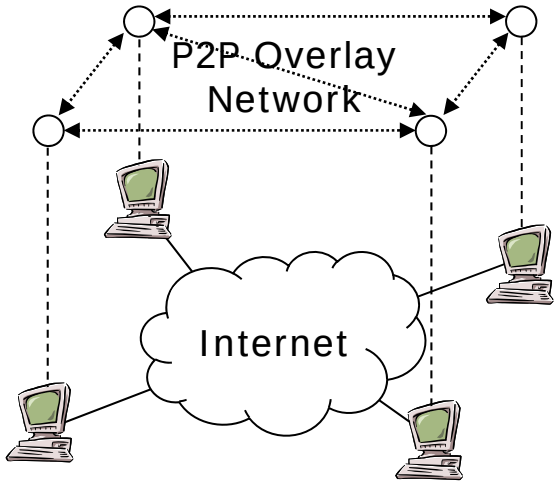
1.1. Definitions and Characteristics (2)

TECHNISCHE UNIVERSITÄT DARMSTADT

- Peers
  - Nodes actively participating in the P2P network (e.g. PC running BitTorrent)
  - Have all the same capabilities (can act as ”clients“ and “servers“ at the same time)
  - Typically located at the edges of the network (end-to-end principle)
  - Identifiable by a General Unique ID (hash-value of “unique” or random ID)



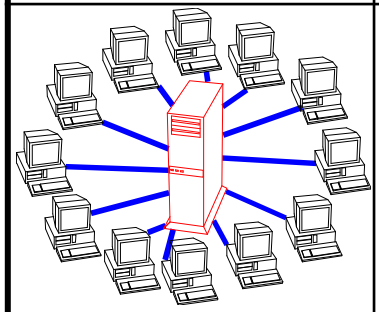
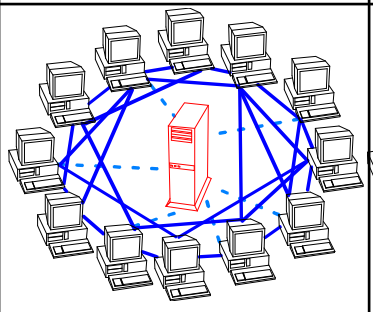
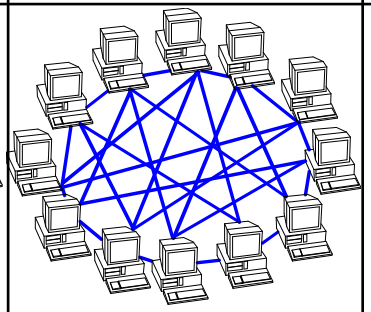
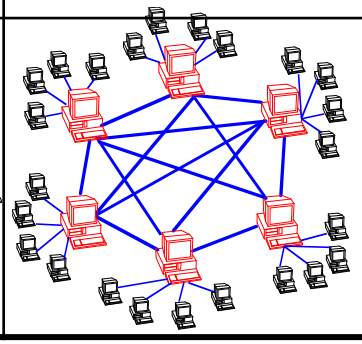
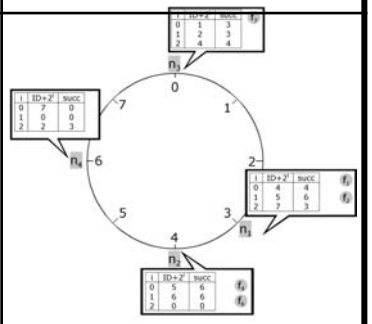
- Peer-to-Peer (Overlay-) Network
  - “Virtual” signaling network, composed of direct connections (e.g. TCP) between peers
  - Typically an “overlay“ network on top of a network (e.g., the Internet)
  - Different topologies: hierarchical, centralized, random



1.2. Classification of P2P Systems

TECHNISCHE UNIVERSITÄT DARMSTADT

<table>
<tr><th>Client-Server</th><th colspan="4">Peer-to-Peer</th></tr>
<tr><td rowspan="3">1. Server is the central entity and only provider of service and content.
→ Network managed by the Server
2. Server as the higher performance system.
3. Clients as the lower performance system

Example: WWW</td><td colspan="4">1. Resources are shared between the peers
2. Resources can be accessed directly from other peers
3. Peer is provider and requestor (Servent concept)</td></tr>
<tr><td colspan="3">Unstructured P2P</td><td>Structured P2P</td></tr>
<tr><td>Centralized P2P</td><td>Pure P2P</td><td>Hybrid P2P</td><td>DHT-Based</td></tr>
<tr><td>1. All features of Peer-to-Peer included
2. Central entity is necessary to provide the service
3. Central entity is some kind of index/group database
Example: Napster</td><td>1. All features of Peer-to-Peer included
2. Any terminal entity can be removed without loss of functionality
3. → No central entities
Examples: Gnutella 0.4, Freenet</td><td>1. All features of Peer-to-Peer included
2. Any terminal entity can be removed without loss of functionality
3. → dynamic central entities
Example: Gnutella 0.6, JXTA</td><td>1. All features of Peer-to-Peer included
2. Any terminal entity can be removed without loss of functionality
3. → No central entities
4. Connections in the overlay are “fixed”
Examples: Chord, CAN</td></tr>
<tr><td>  </td><td>  </td><td>  </td><td>  </td><td>  </td></tr>
</table>

1.2. Properties – Client/Server versus P2P Systems

TECHNISCHE UNIVERSITÄT DARMSTADT

Properties	Description	C/S	P2P
Manageability	How hard is it to keep the system working?	+	-
Information coherence	How authoritative is information in the system?	+	-
Extensibility	How easy is it to grow the system, to add new resources to it?	-	+
Fault-tolerance	How well can the system handle failures?	-	+
Security	How hard is it to subvert the system?	+/-	-
Resistance to lawsuits	How hard is it for an authority to shut down the system?	-	+
Scalability	How large can the system grow?	+/-	+

Source: N. Minar: *Distributed Systems Topologies: Part 2*; O'Reilly Open P2P, December 2001.

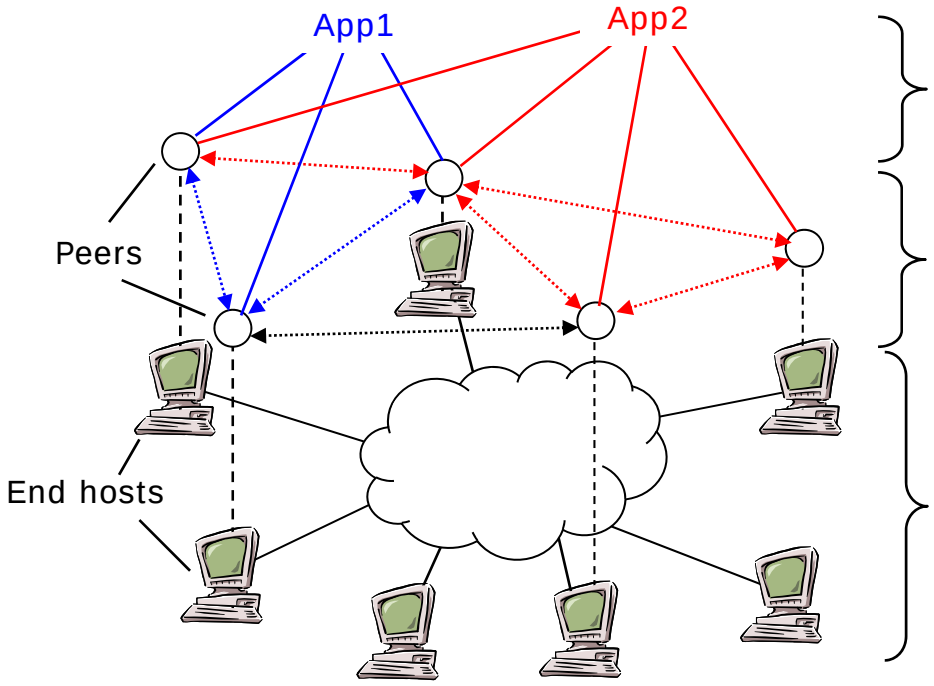
2. Application Areas

Definition, Classification

2.1. Definition of P2P Applications

TECHNISCHE UNIVERSITÄT DARMSTADT

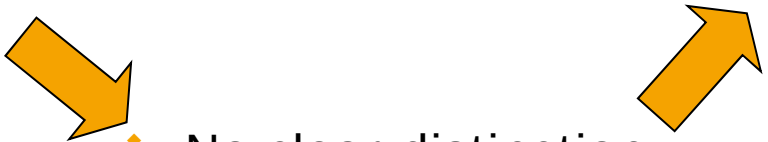
- P2P Applications
  - Application of a P2P network for a specific purpose
  - Draw on the cooperation of peers in the form of services these peers provide each other in order to achieve a common goal



2.2. P2P Application Classifications (1)

TECHNISCHE UNIVERSITÄT DARMSTADT

- Conventional Classification of P2P Applications
  - File Sharing (e.g., Napster, Gnutella, eMule, BitTorrent)
  - Grid Computing (SETI@home)
  - Instant Messaging (Skype)
  - Collaboration (Groove)
- Classification by Means of Shared Resources
  - Information
  - Files
  - Bandwidth
  - Storage space
  - Processor cycles



- No clear distinction
- Some cases even misleading

2.2. P2P Application Classifications (2)

TECHNISCHE UNIVERSITÄT DARMSTADT

- Information
  - Presence Information, Instant Messaging, Document Management, Collaboration
  - *Mobile*: location aware services
- Files
  - File sharing: Music, Movies, Pictures, Software
  - File storage, search, and retrieval
  - Most widespread P2P application
- Bandwidth
  - P2P Video Streaming, Application Layer Multicast, P2P Telephony (Skype)
- Storage space
  - P2P Storage Networks (e.g. PAST, Pasta, OceanStore, Wuala)
- Processor cycles
  - Sharing of idle CPU (e.g. SETI@home)

TECHNISCHE UNIVERSITÄT DARMSTADT

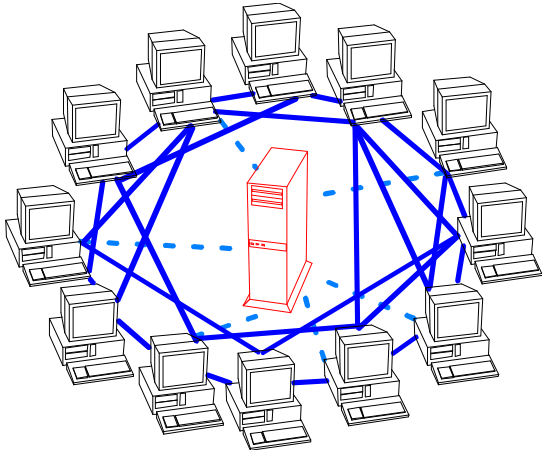
3. Centralized P2P Networks

Definition, Basic Characteristics,
Signaling Characteristics, and Discussion

3.1. Definition of Centralized P2P

TECHNISCHE UNIVERSITÄT DARMSTADT

- All peers are connected to central entity
- Peers establish connections between each other on demand to exchange user data
  - *E.g.*, mp3 compressed data
- Central entity is necessary to provide the service
- Central entity is some kind of index/group database
- Central entity is lookup/routing table



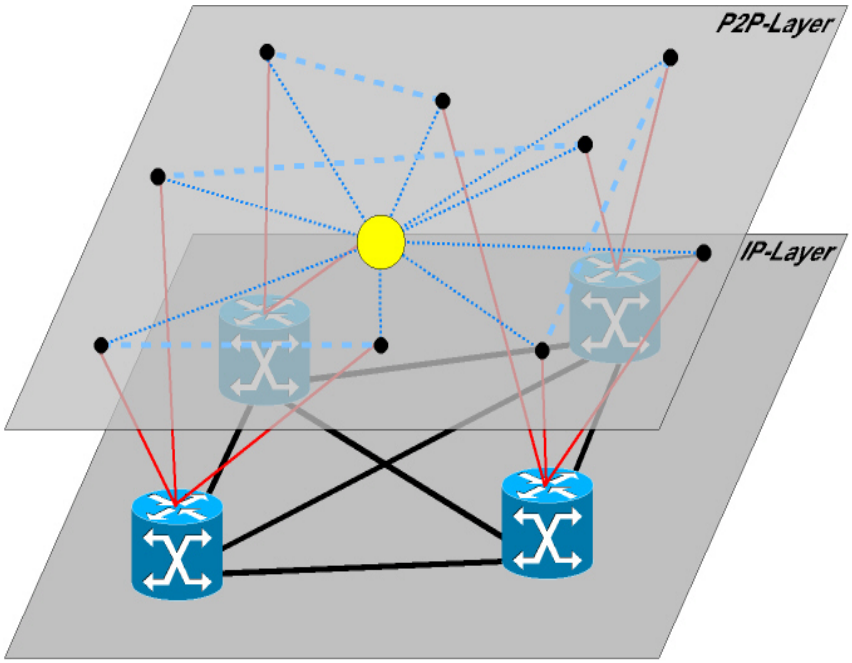
3.2. Basic Characteristics of Centralized P2P

TECHNISCHE UNIVERSITÄT DARMSTADT

- Bootstrapping: Bootstrap-server = central server
- Central entity can be established as a server farm, but one single entry point = single point of failure (SPOF)
- All signaling connections are directed to central entity
- Peer ↔ central entity:
  P2P protocol, *e.g.*, Napster protocol
  - To find content
  - To log on to the overlay
  - To register
  - To update the routing tables
  - To update shared content information
- Peer ↔ Peer: HTTP
  - To exchange content/data

3.2. Topology of Centralized P2P

TECHNISCHE UNIVERSITÄT DARMSTADT



Servant
Connection between 2 servents (TCP)
Connection between router and servant
Connection between routers (Core)

3.3. Centralized P2P — Example: Napster

TECHNISCHE UNIVERSITÄT DARMSTADT

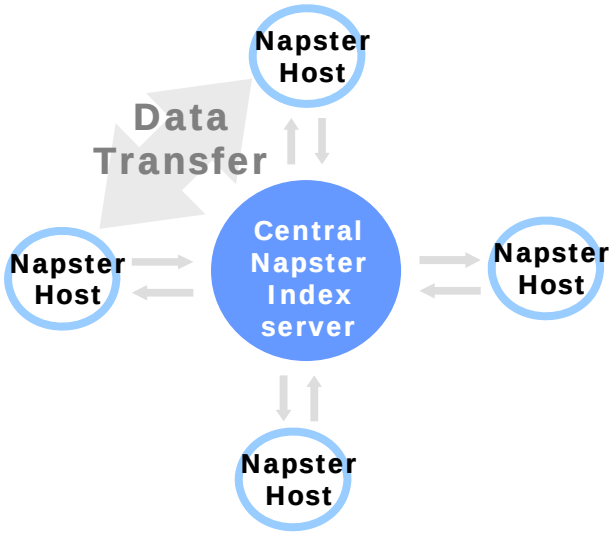
- Napster – A program for sharing files over the Internet

- Brief History
  - **May 1999**: Shawn Fanning (freshman, Northeastern University) founds Napster Online music service
  - **December 1999**: First Lawsuit
  - **March 2000**: University of Wisconsin reports that 25% of its IP traffic is Napster traffic
  - **December 2000**: estimated 60 million users
  - **February 2001**: US Circuit Court of appeals: Napster knew users violating copyright laws
    → Shut down of the service

3.3. Napster: How Does it Work?

TECHNISCHE UNIVERSITÄT DARMSTADT

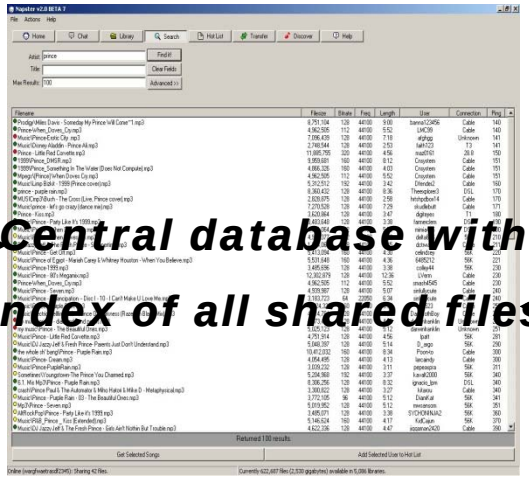
- Application-level, client-server protocol over point-to-point TCP
- Participants
  - Napster Hosts/peers
  - Client Service
    - Login
    - Data-requests
    - Download-requests
  - P2P Service
    - Data-transfer
  - Napster Index Server
    - Pure Server
- Five steps
  - Connect to Napster Server
  - Upload your list of files (push) to server
  - Query Index Server with a list of keywords to search the full list with
  - Select “best” of correct answers
  - Connect to providing host/peer



3.3. Napster — Example

Joe shares on his client several MP3 files.

Melissa shares on her client several MP3 files, too.



Central database with index of all shared files

? Prince purple rain ?

! Prince purple rain !

@ Mr. Arayama

! Prince purple rain !

? Prince purple rain ?

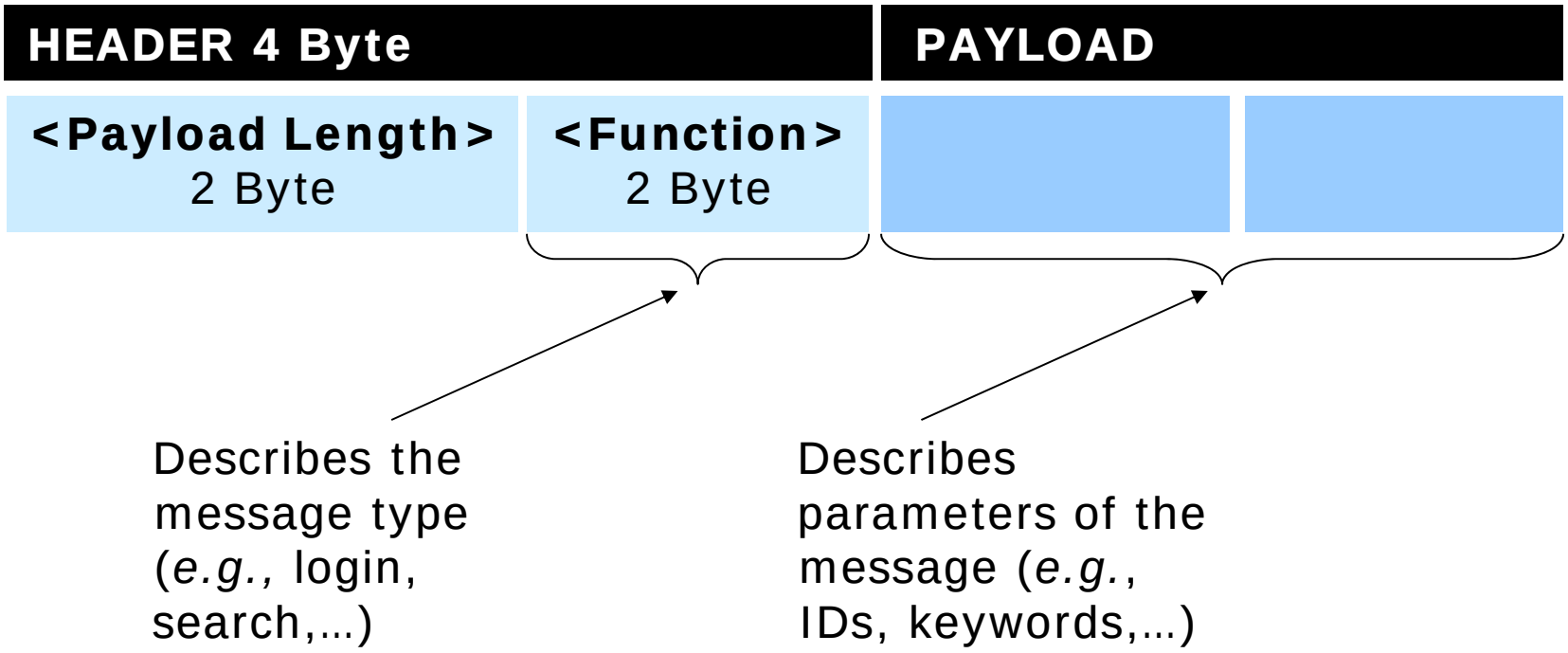
Prince - purple rain

Prince - purple rain

3.3. Napster Message Structure

TECHNISCHE UNIVERSITÄT DARMSTADT

General Header Structure:



3.3. Napster Initialization

TECHNISCHE UNIVERSITÄT DARMSTADT

Client/Server Service

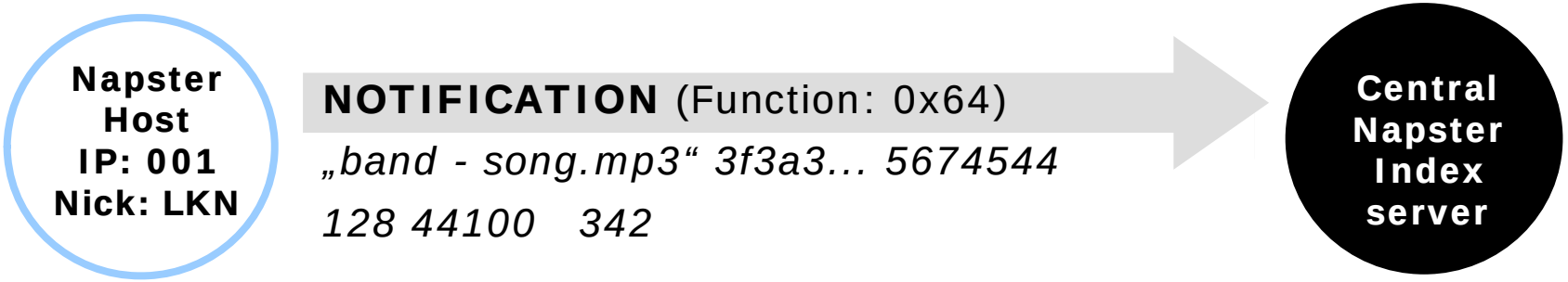
1: LOGIN (Function: 0x02)

<Nick>	<Password>	<Port>	<Client-Info>	<Link-type>

2: LOGIN ACK (Function: 0x03)

3: NOTIFICATION OF SHARED FILE (Function: 0x64)

"<Filename>"	<MD5>	<Size>	<Bitrate>	<Freq>	<Time>



3.3. Napster File Request Procedure

TECHNISCHE UNIVERSITÄT DARMSTADT

1: SEARCH (Function: 0xC8)

[FILENAME CONTAINS ”Search Criteria“] [MAX_RESULT <Max>]

[LINESPEED <Compare> <Link-Type>]

[BITRATE <Compare> “<Bitrate>”] [FREQ <Compare> “<Freq>”]

2: SEARCH RESPONSE (Function: 0xC9)

”<Filename>“ <MD5> <Size> <Bitrate> <Freq>

<Time> <Nick> <IP> <Link-Type>

Central Napster Index server

SEARCH (Function: 0xC8)

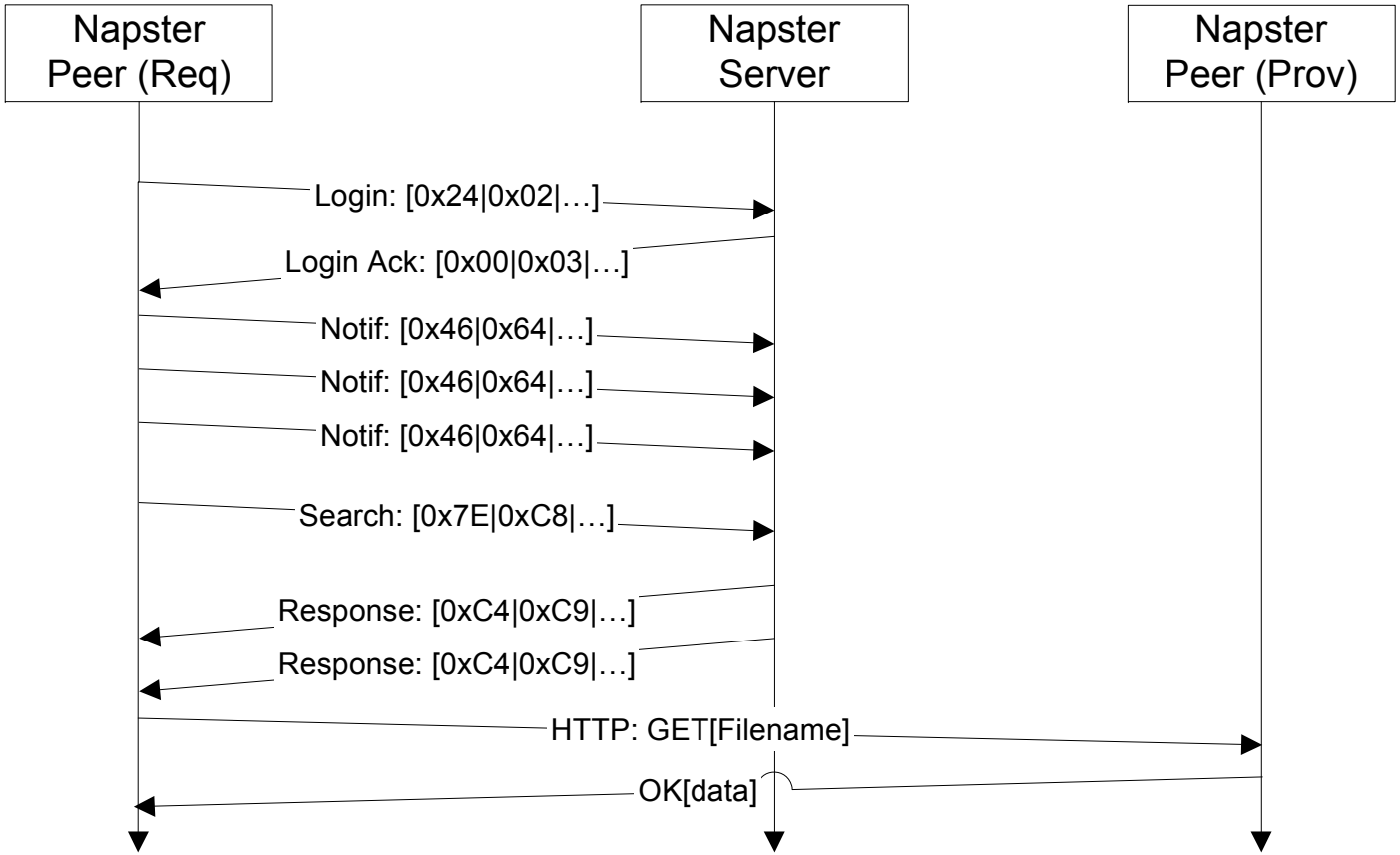
FILENAME CONTAINS ”song“ MAX_RESULTS 100

LINESPEED ”AT LEAST“ 6 BITRATE ”AT LEAST“ “128“

FREQ ”EQUAL TO“ ”44100“

Napster Host IP: 002 Nick: MIT

3.3. Napster Signaling (Summary)



Sample message sequence chart for one Napster server with one requesting and one providing peer

3.4. Discussion

TECHNISCHE UNIVERSITÄT DARMSTADT

- Drawbacks
  - Single Point of Failure → Easily attackable
  - Bottleneck
  - Potential of congestion
  - Central server in control of all peers
- Advantages
  - Fast and complete lookup (one hop lookup)
  - Central managing/trust authority
  - No keep alive necessary, beyond content updates
- Application areas (examples)
  - VoIP (SIP, H.323)
  - Auctioning (Ebay)

TECHNISCHE UNIVERSITÄT DARMSTADT

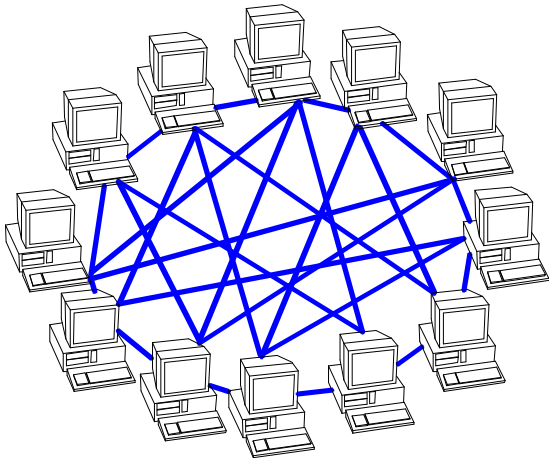
4. Decentralized P2P Networks

Definition, Basic Characteristics,
Signaling Characteristics, and Discussion

4.1. Definition of Decentralized P2P (Pure P2P)

TECHNISCHE UNIVERSITÄT DARMSTADT

- Any terminal entity can be removed without loss of functionality
- No central entities at all employed in the overlay
- Peers establish connections between each other randomly
  - To route request and response messages
  - To insert request messages into the overlay



4.2. Basic Characteristics of Decentralized P2P (1)

TECHNISCHE UNIVERSITÄT DARMSTADT

- Bootstrapping
  - Via bootstrap-server (host list from a web server)
  - Via peer-cache (from previous sessions)
  - Via IP multicast / broadcast
- Routing
  - Completely decentralized
  - Reactive protocol: routes to content providers are only established on demand, no content announcements
  - Requests: flooding (limited by TTL and GUID)
  - Responses: routed (Backward routing with help of GUID)

4.2. Basic Characteristics of Decentralized P2P (2)

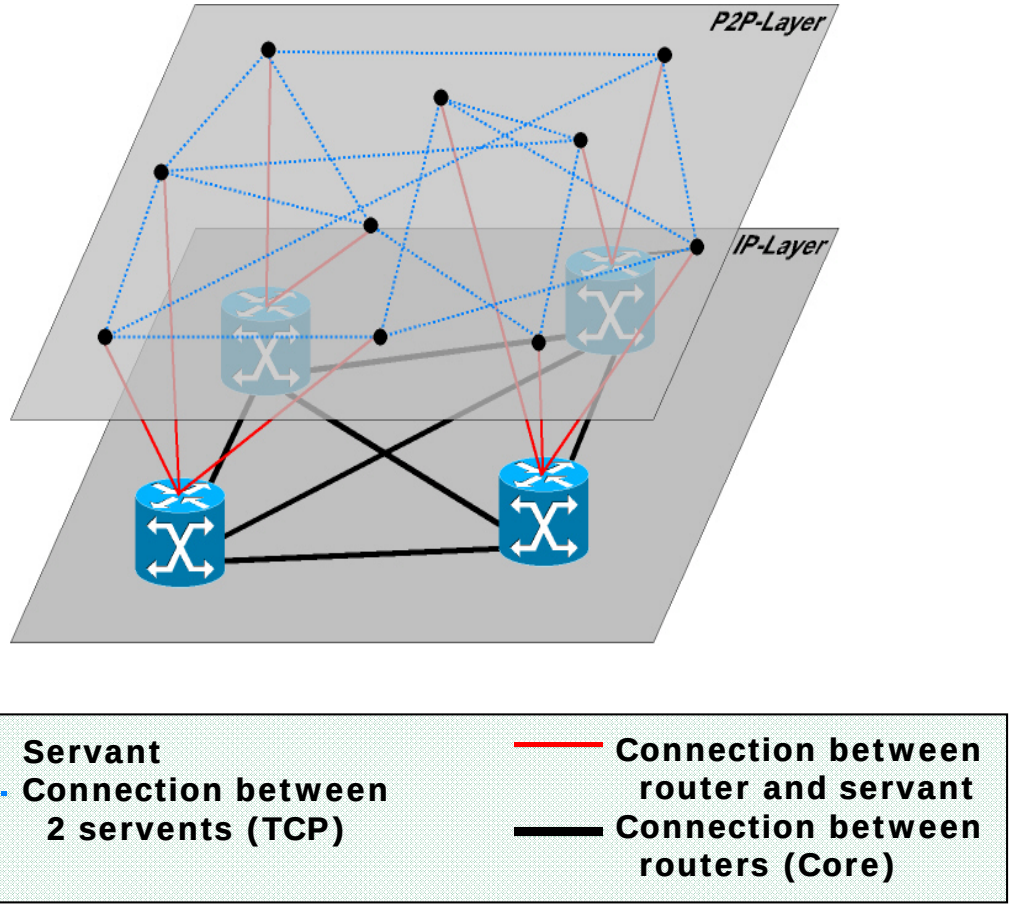
TECHNISCHE UNIVERSITÄT DARMSTADT

- Signaling connections
  (stable, as long as neighbors do not change)
  - Based on TCP
  - Keep-alive
  - Content search

- Content transfer connections (temporary)
  - Based on HTTP
  - Out of band transmission

4.2. Topology of Decentralized P2P

TECHNISCHE UNIVERSITÄT DARMSTADT



4.3. Example: Gnutella 0.4

TECHNISCHE UNIVERSITÄT DARMSTADT

- Focus: decentralized method of searching for files

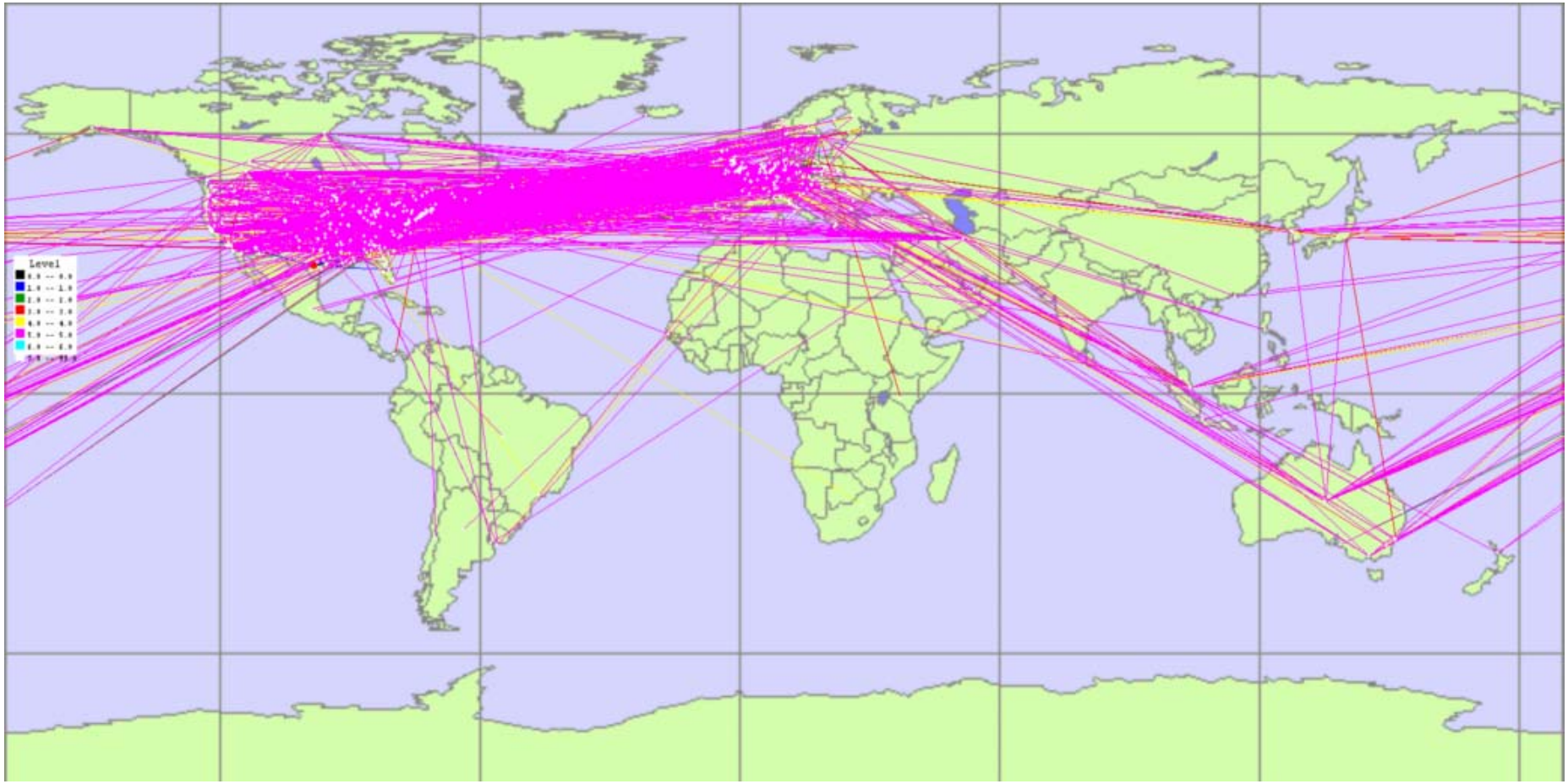
- Brief History
  - **March 2000**: open source release by by Justin Frankel and Tom Pepper of Nullsoft, a division of AOL, and almost immediately withdrawn

  - **Spring 2001**: further developments to improve scalability
    → Gnutella 0.6 (Hybrid P2P)

  - Since then:
    - available in a lot of implementations (Limewire, bearshare,…)
    - Developed further on (privacy, scalability, performance,…)

4.3. The Gnutella Network

TECHNISCHE UNIVERSITÄT DARMSTADT

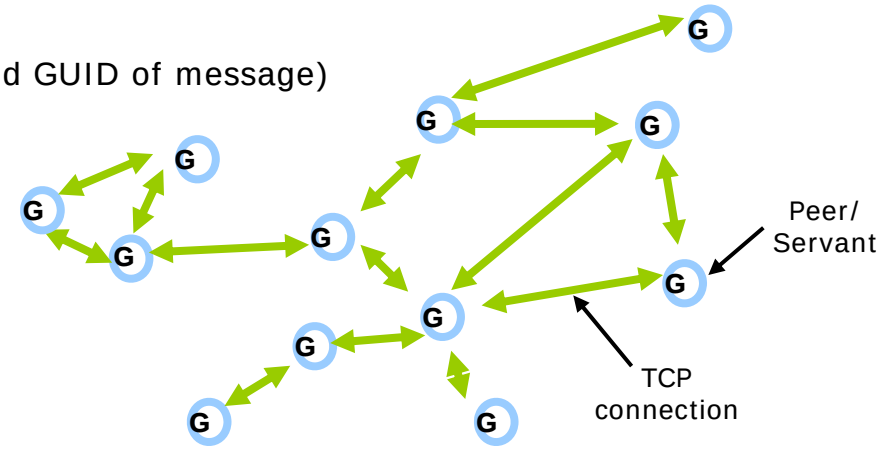


Measurements taken at LKN in May 2002

4.3. Gnutella: How Does it Work?

TECHNISCHE UNIVERSITÄT DARMSTADT

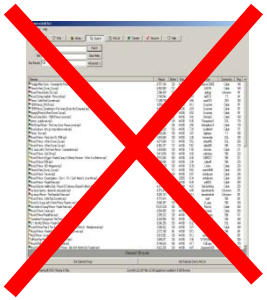
- Application-level, peer-to-peer protocol over point-to-point TCP
  Participants:
  - Gnutella peers/servants
  - Router Service
    - Flood incoming requests (regard TTL!)
      - Keep alive
      - content
    - Route responses for other peers (regard GUID of message)
      - Keep alive (PING/PONG)
      - Content (QUERY/QUERYHIT)
    - Data-requests
    - Download-requests
  - Lookup Service
    - Initialize Data requests
    - Initialize keep alive requests
  - "Server"-Service
    - Serve Data-requests (HTTP)



- Five steps:
  - Connect to ate least one active peer (address received from bootstrap)
  - Explore your neighborhood (PING/PONG)
  - Submit Query with a list of keywords to your neighbors (they forward it)
  - Select "best" of correct answers (which we receive after a while)
  - Connect to providing host/peer

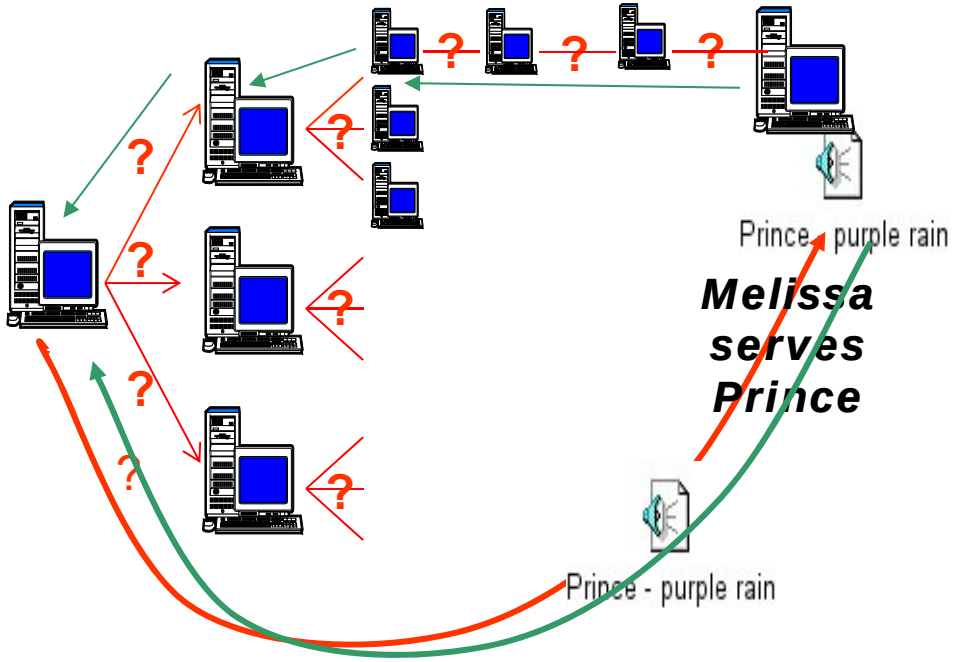
4.3. Gnutella — Example

http://www.gnutelliums.com/



No central Database

Joe is searching for Prince

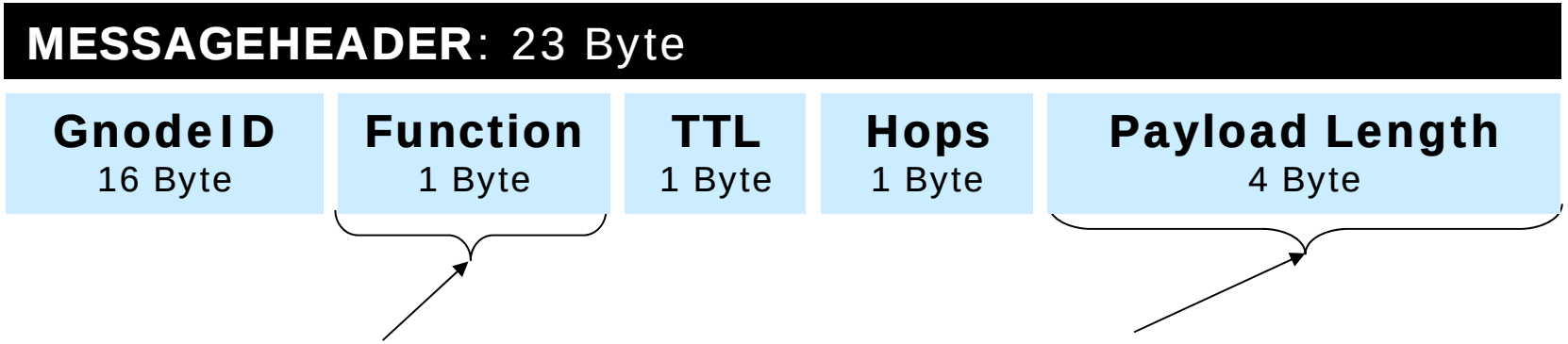


Melissa serves Prince

4.3. Gnutella Message Structure

TECHNISCHE UNIVERSITÄT DARMSTADT

General Header Structure:



- **GnodeID**: Unique 128 bit ID of any host
- **TTL** (Time-To-Live): Number of servants, a message may pass before it is killed
- **Hops**: Number of servants a message already passed

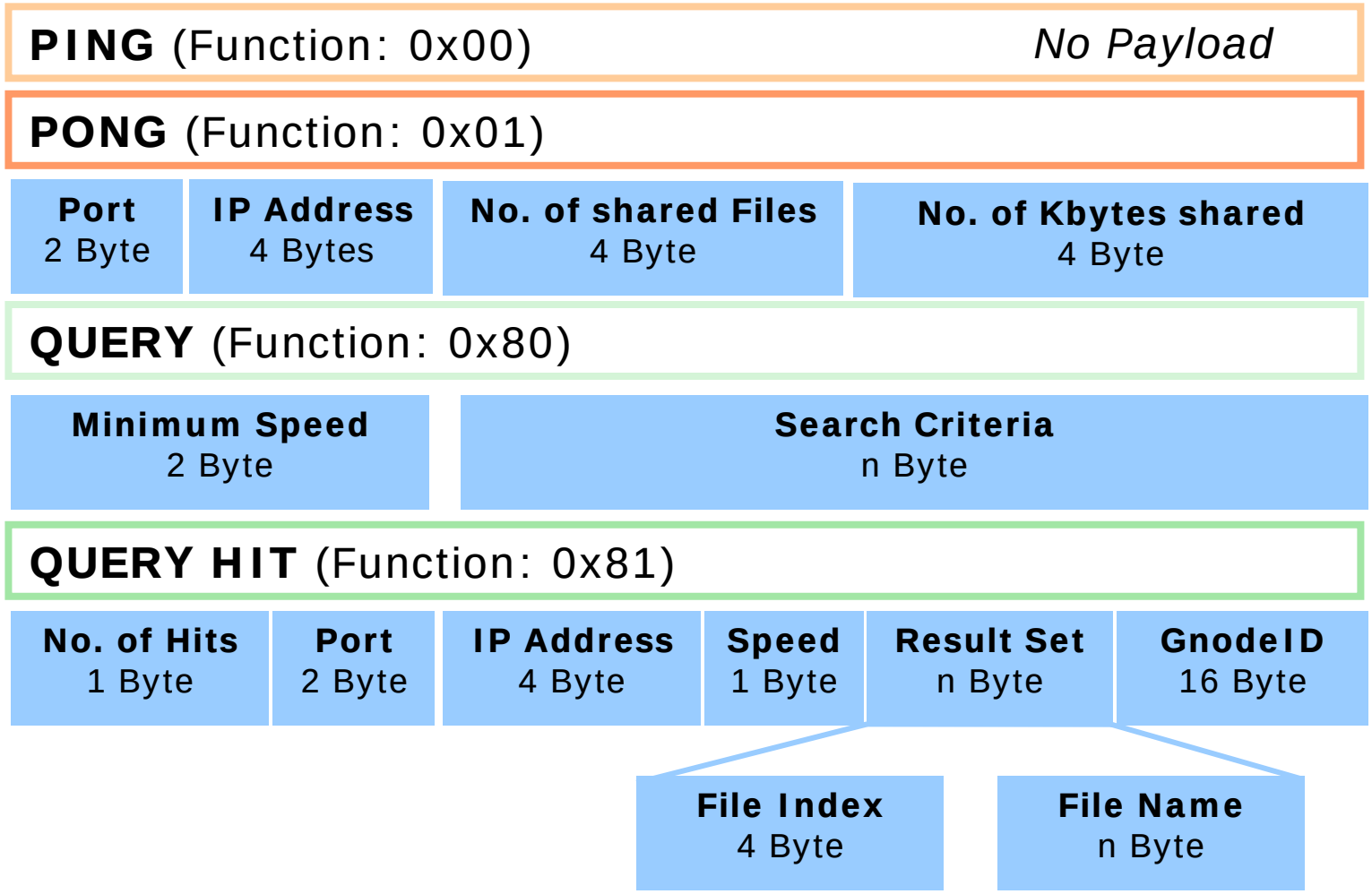
4.3. Gnutella TTL and Hops Counter

TECHNISCHE UNIVERSITÄT DARMSTADT

- TTL
  - Time To Live. The number of times the descriptor will be forwarded by Gnutella servents before it is removed from the network. Each servent will decrement the TTL before passing it on to another servent. When the TTL reaches 0, the descriptor will no longer be forwarded.
- Hops
  - The number of times the descriptor has been forwarded. As a descriptor is passed from servent to servent, the TTL and Hops fields of the header must satisfy the following condition:
  - TTL(0) = TTL(i) + Hops(i)
  - Where TTL(i) and Hops(i) are the value of the TTL and Hops fields of the header at the descriptor's i-th hop, for i >= 0.

[Source: Clip2 Distributed Search Services: The Gnutella Protocol Specification v0.4, 2001 https://web.archive.org/web/20081221081801/http://www9.limewire.com/developer/gnutella_protocol_0.4.pdf]

4.3. Gnutella Messages



4.3. Gnutella Routing

TECHNISCHE UNIVERSITÄT DARMSTADT

- **Basic Routing Principle: „Enhanced“ Flooding**

- Save Origin of received **PINGs** and **QUERIEs**
- Decrease **TTL** by 1
- If **TTL** equals 0, kill the message

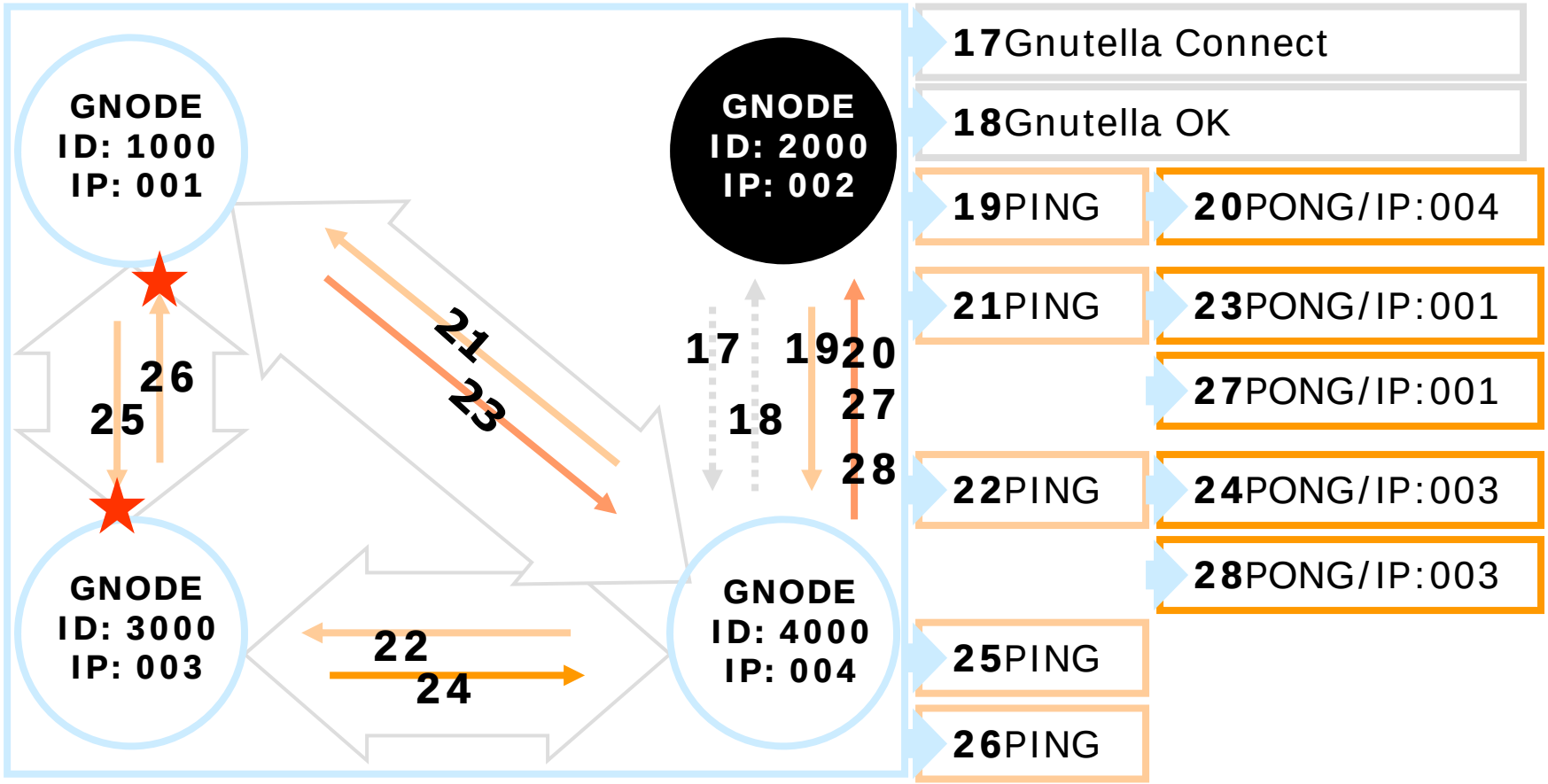
- **Flooding**: Received **PINGS** and **QUERIES** must be forwarded to all connected Gnodes
- **PINGS** or **QUERYS** with the same FUNCTION ID and GNODE ID as previous messages are destroyed (avoid loops)

- **PONG** and **QUERY HIT are forwarded to the** origin of the according PING or QUERY

4.3. Gnutella Connection Setup

TECHNISCHE UNIVERSITÄT DARMSTADT

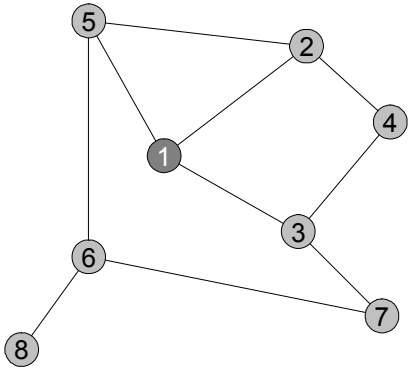
Gnode 2000 establishes a connection to 4000



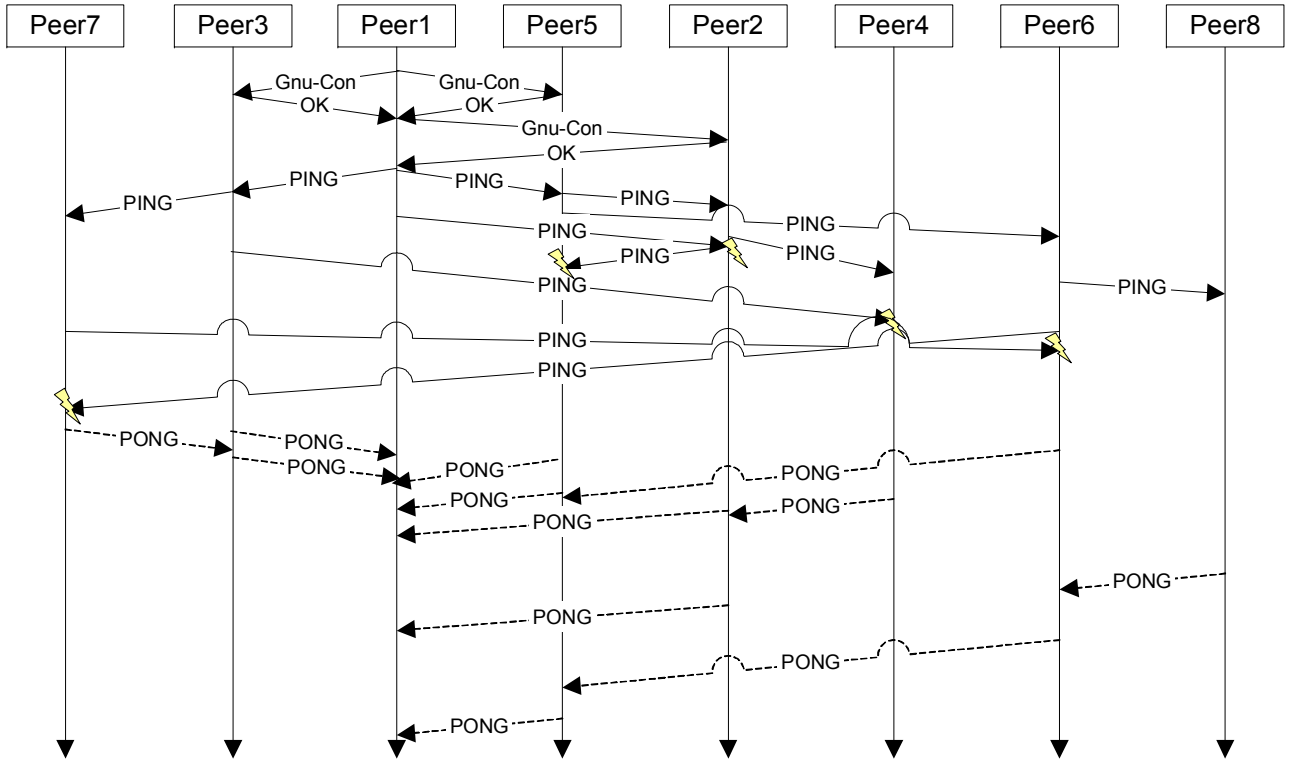
4.3. Summary of Signaling (Gnutella 0.4)

TECHNISCHE UNIVERSITÄT DARMSTADT

Sample Gnutella 0.4 network:



Sample message sequence chart according to the sample network:



4.4. Discussion

TECHNISCHE UNIVERSITÄT DARMSTADT

- Drawbacks
  - High signaling traffic, because of decentralization
  - Modem nodes may become bottlenecks
  - Overlay topology not optimal, as
    - No complete view available,
    - No coordinator
  - If not adapted to physical structure delay and total network load increases
    - Zigzag routes
    - Loops
- Advantages
  - No single point of failure
  - Can be adapted to physical network
  - Can provide anonymity
  - Can be adapted to special interest groups
- Application areas
  - File-sharing
  - Context based routing (see chapter about mobility)