

## Willkommen bei Verteilte Systeme!

Von Datenbanken  
über Webdienste  
bis zu p2p und Sensornetzen.



Heute: **Peer-to-peer in der Praxis – wo und wie sich Verteilung lohnt(-e).**

Wer nicht aus der Vergangenheit lernt, ist verdammt ihre Fehler wiederholen, mit weniger Zeit, denn „die Probleme sind ja schon gelöst“.

## Darum ich

- Seit 2004 in p2p-Entwicklung
- Seit 2013 mit Kompetenz :-)
- Aktuell Release-Manager des Freenet/Hyphanet Projektes

## Vorträge

- 5 Minuten pro Person, Gruppen möglich.
- Weitere Ideen: Übersicht über einen [FOSDEM 2023-Vortrag](#).

## Fallacies of distributed Systems

- The network is reliable
- The network is secure
- The network is homogeneous
- Topology does not change
- Latency is zero
- Bandwidth is infinite
- Transport cost is zero
- There is one administrator
- Hard disks don't fail
- Files stay intact
- Power is stable
- IPs are reachable
- Constant factors are negligible
- APIs stay compatible
- Textfiles are simple

## Ziele heute

- Sie kennen die zentralen Herausforderungen der Praxis:
  - Einstieg
  - Suche
  - Inhalte verbreiten
  - Kommunikation
- Sie können in Bezug auf zwei Herausforderungen die Eigenschaften von je zwei p2p-Systemen beschreiben, die sich in der echten Welt bewährt haben.
- Sie können einschätzen, ob ein bestimmtes Konzept diese Herausforderungen bestehen könnte.

## Grundprobleme in Peer-to-Peer-Netzen

- Einstieg:** Wie finde ich meinen Platz im Netz?



- Suche:** Wo gibt es, was ich brauche?



- Störungsresistenz:** Wie skaliert Gewünschtes besser als Unerwünschtes?

- Verbreitung:** Wie vermeide ich Flaschenhälse?



- Kommunikation:** Wie fließen Informationen durchs Netz?



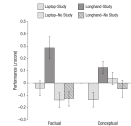
## Darum praktisch erprobte p2p-Netze verstehen

after a few days (and especially nights) of nervous full-site tinkering, it turned a 40 minute deploy process into one that lasted just 12 seconds!

- Bittorrent-Deployment: <https://vimeo.com/11280885>

Spoiler: Cut-through routing.

## Laptops: Eigenverantwortlich



“even when laptops are used solely to take notes ... their use results in shallower processing.”  
— Mueller and Oppenheimer (2014)

### Laptops

- Möglichst zu

## Wiederholung 1

- Sammlung autonomer Knoten, die als ein kohärentes System erscheinen.
- Ziele:** Ressourcen, Verteilungstransparenz, Skalierbarkeit
- Skalierung:** Größe, Geographie, Administration
- Latenz, Partitionierung, Replikation, Caching
- Fallacies!**
- Cluster, Grid, Cloud, Ubiquitous, Mobile, DIS, Sensornetze

## Ablauf heute

- Grundprobleme
- Gnutella (das Erste verbreitete, komplett verteilte p2p-Netz)  
--- PAUSE 14:30 ---
- Kademlia (das am weitesten verbreitete DHT)
- BitTorrent
- Freenet/Hyphanet
- Weiteres (Aktuelles, WebRTC, ...)

## Welche p2p-Netze kennen Sie

### am FlipChart sammeln

- 
- 
- 
- 
- 

## Warum p2p?

**Skalierbarkeit** Ein einzelner Server bricht bei etwa 100k Anfragen pro Sekunde ein. *dwd bei Sturm Sabine 2020?*

**Mit Nutzung wachsen** Ähnliche Infrastruktur für 1000 Leute oder 10 Millionen Leute

**Infrastrukturkosten** 100k€ pro Jahr = Entwickler oder Entwicklerin

## Mein Ziel

Ich will, dass Sie die Fähigkeiten erwerben, unter denen zu sein, die die Deployment Zeit um Größenordnungen verringern, ohne dabei die Kosten dafür zu zahlen, Torrents als Blackbox zu sehen.

**Torrent** Bezeichnung für eine BitTorrent-Datei oder eine von BitTorrent verwaltete Datei.

**BitTorrent** Ein p2p-System zum Verteilen großer Datenmengen, bei dem die Verwaltung auf zentralisierten Trackern läuft

## Projekte

- Notieren Sie bitte Ideen
- Modulplan: 39h Selbststudium
- Projekt interessanter und nützlicher als Klausur
- Projektideen sammeln.

Doing X with *libp2p* oder *libresilient*?

Auf einem der aktuellen **NL.net-Projekte** aufbauen?

## Wiederholung 2

- Architektur:** Layered, Object, Resource, Event
- Schichten und Overlay Netze
- Prozesse sind isoliert, Threads teilen Speicher.
- Middleware** als Schicht: Übernimmt Verteilung, gibt Garantien.
- Messaging:** Request-Reply, Pub-Sub, Pipeline.
- Overlay** metriken: Link Stress und Stretch

## Was und warum?

Was?  
**peer-to-peer (p2p)** peers (gleichberechtigte Partner) arbeiten zusammen, um sich gegenseitig einen Dienst zu erbringen.

Warum?  
*Sie haben ein unerwartet beliebtes Programm geschrieben. Jetzt wollen es 100 Millionen Leute herunterladen. Größe: 50GiB. Wie viel kostet die Verteilung?*

## Ziele für Grundprobleme

Sie können die Grundprobleme beschreiben, die Peer-to-Peer-Netze lösen müssen:

### Fundamente

- Einstieg
- Suche
- Verbreitung

### Aufbauend

- Kommunikation
- Störungsresistenz

## Warum nicht p2p?

- Gestiegende Leistung von Servern. *Sturm: dwd<sup>2</sup> hielt größtenteils Stand (durch vereinfachte Seite<sup>3</sup>)*
- Handies sind durch Batterie und Netz begrenzt → keine kontinuierliche Leistung. (Nachts möglich?)
- Viele der einfachen Lösungen unmöglich, z.B. Geld auf Probleme werfen.

<sup>2</sup>dwd: Deutscher Wetterdienst.  
<sup>3</sup>=> gibt es eine einfachere Lösung?



## Einstieg: Wie finde ich meinen Platz im Netz?

- **Erste Adressen:** Wie finde ich Adressen anderer Knoten?
- **Wahl der Verbindungen:** Mit wem sollte ich mich verbinden?
- **Routing-Informationen:** Welche Daten brauchen die Knoten?



## Suche: Wo gibt es, was ich brauche?

Zwei Konzepte:

- **Pfade zu existierenden Daten finden:** Gnutella
- **Daten an die richtigen Orte bringen:** Kademia, BitTorrent VHT, Freenet



## Kommunikation

- **Vier-Augen Gespräch (PM/DM/msg/Anruf/...)**
- **Unterhaltung in Gruppen (Chat, Forum, Videokonferenz, ...)**
- **Öffentliche Unterhaltung**
- **Von neuen Inhalten erfahren**
- **Informationen über Inhalte (Kommentare, Bewertung, ...)**



## Störungsresistenz: Wie skaliert Gewünschtes besser als Unerwünschtes?

### Störung

"Disruption", alles, was die Qualität des Dienstes für die Nutzer verringert

### Auf jeder Ebene nötig

- **Knotenauswahl:** Verbindung mit Angreifern
- **Suche:** Spam, Falschinformationen
- **Verbreitung:** Dateien korrumpieren
- **Kommunikation:** Spam, Belästigung und Zensur<sup>6</sup>

<sup>6</sup> Das Web betrachtet Zensur als Störung und lenkt Anfragen darum herum.  
— The Internet treats censorship as a malfunction and routes around it. – John Perry Barlow



## Weitere Eigenschaft: Grad der Verteilung

Serverkoordinierte Teilgruppen bis vollständig dezentrale Interaktion.



## Ziele

Sie verstehen die grundlegende Funktionsweise von Gnutella als Beispiel einer effizienten, dezentralen Schlüsselwort-Suche.

Sie erkennen, wo die für Gnutella entwickelten Techniken sinnvoll genutzt werden können.

Sie wissen, welche Probleme ungelöst blieben.



## Strukturiert vs. Unstrukturiert

### Strukturiert

- **Erste Adressen:** Braucht Topologie<sup>4</sup>
- **Wahl der Verbindungen:** Nur bestimmte sinnvoll
- **Routing-Informationen:** Durch Auswahl der Partner (Peers)

### Unstrukturiert

- **Erste Adressen:** Einfache Liste
- **Wahl der Verbindungen:** Beliebige Andere
- **Routing-Informationen:** Explizit austauschen

Kann ich alle direkt erreichen?

<sup>4</sup>Topologie: Struktur des Netzes.



## Schlüssel zum Licht



## Störungsresistenz: Wie skaliert Gewünschtes besser als Unerwünschtes?

### Störung

"Disruption", alles, was den die Qualität des Dienstes für die Nutzenden verringert



## Störquellen

Sammeln am Flipchart

<sup>7</sup>Werbung ist Spam durch die genutzte Plattform.



## Zusammenfassung

- **Einstieg:** Erste Adressen und Routing-Info
- **Suche:** Schlüsselwort, Inhalt, Public Key
- **Störung:** Parasiten, Trolle, Spammer, Konkurrenten und Angreifer
- **Verbreitung:** Tracker, Download-Mesh, Fragmente mit Caching
- **Kommunikation:** Privatnachricht, Forum, Neuigkeiten, Kommentare



## Inhalt

- **Nutzersicht:** Das war Gnutella
- **Einstieg:** GWebCaches
- **Suche:** Slow-Start + Keyword-Multicast
- **Verteilung:** Download-Mesh
- **Kommunikation:** Neues und Sammlung zeigen
- **Störungsresistenz:** Heuristik oder Inhalts-Matrizen



## Suche: Wonach suchen?

- **Schlüsselwort:** Gnutella, Skype (vor MS)
- **Inhalts-Hash:** Kademia, BitTorrent VHT, Freenet
- **Öffentlicher Schlüssel:** Freenet

Glossar:



**BitTorrent VHT** Verteilte Hashtabelle, ein DHT

**DHT** Distributed Hashtable

**Öffentlicher Schlüssel** public key, das Gegenstück zum privaten Schlüssel in asymmetrischer Verschlüsselung.



## Verbreitung: Wie vermeide ich Flaschenhälse?

- **Zentralisiert:** Streaming im Provider-Netz via Multicast
- **Swarming:** Nutzer übernehmen einen Teil der Verteilung
  - Koordiniert von zentraler Stelle: BitTorrent (Tracker)
  - Koordiniert durch die Nutzer: Gnutella (Download-Mesh)
  - Unabhängig verteilte Fragmente: Freenet<sup>5</sup>

Glossar:



**Download-Mesh** Name des Protokolls

**Tracker** Ein Server der den BitTorrent-Schwarm koordiniert

<sup>5</sup>Reduziert Swarming auf Download einzelner Dateien, braucht aber caching: Zeitlich begrenzte Zwischenspeicherung.



## Störungsresistenz: Wie skaliert Gewünschtes besser als Unerwünschtes?

### Störung

"Disruption", alles, was den die Qualität des Dienstes für die Nutzenden verringert

### In Gruppen sammeln

- **Knotenauswahl:**
- **Suche:**
- **Verbreitung:**
- **Kommunikation:**



## Störquellen

Sammeln am Flipchart

### Quellen

- **Parasiten:** Bessere Leistung auf Kosten Anderer (leecher).
- **Trolle:** Kein Finanzinteresse, minimale Ressourcen, nutzen jegliche Lücke.
- **Spammer:** Erfolg durch Verbreitung eigener Inhalte.<sup>7</sup>
- **Konkurrenten:** Erfolg durch verringerte Qualität des Systems.
- **Angreifer:** Erfolg durch Schädigung von Nutzern.

<sup>7</sup>Werbung ist Spam durch die genutzte Plattform.



## Gnutella

### Erinnert mich bitte an die Pause

On March 14th, 2000, ... an early version ... with a note: "Justin and Tom work for Nullsoft, makers of Winamp and Shoutcast. See? AOL can bring you good things!" ... AOL ordered him to take the program down immediately ... calling Gnutella an "unauthorized freelance project." ... hackers had gone ... to reverse-engineer it ... into the hands of the open-source community ... — *The World's Most Dangerous Geek; Interviewed by David Kushner; RollingStone.com; January 13, 2004.*

Ursprung der ersten Tauschbörse.<sup>8</sup> Sie verschwand nach verlorenen Urheberrechtsklagen der Entwicklungsfirmer in der Obskurität und seine technischen Errungenschaften gerieten in Vergessenheit.

<sup>8</sup>Frankreich: Ein Pionier in dem Musikmarkt und Absatz labels schützte

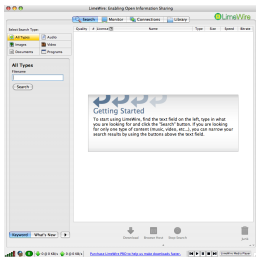


## Sicht der Nutzer/-innen

- 50 Millionen Knoten
- Globale Suche nach Dateinamen und ID3-Tags
- Filter für Creative-Commons-Lizenzen
- Suche nach den neusten Dateien (What's New?)
- Downloads von vielen Quellen ohne zentrale Koordination
- Audio-Streaming um 2004 („Dateivorschau“)
- LimeWire, Bearshare, Shareaza, Phex, gtk-gnutella, ...



## Praktisch



Arne Babenhauserheide					
Verteilte Systeme 1: peer-to-peer					
Einstieg	Grundprobleme	Gnutella	Kademia	BitTorrent Downloads	Freenet Abschluss
000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000

## Weitere Knoten finden: X-Try

Beim Handshake (wie HTTP):

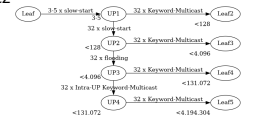
When rejecting a connection, a server MUST, if possible, provide the remote host with a list of other Gnutella hosts, so it can try connecting to them. This SHOULD be done using the X-Try header.

An X-Try header can look like:

X-Try:1.2.3.4.1234,3.4.5.6:3456

Arne Babenhauserheide					
Verteilte Systeme 1: peer-to-peer					
Einstieg	Grundprobleme	Gnutella	Kademia	BitTorrent Downloads	Freenet Abschluss
000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000

## Suche im Netz



Nicht existente Datei durchschnittlich:  $4 \times 32 \times 32 = 4.096$  Knoten.

Last (empirisch):  $<1\text{kiB/s}$  Leaf,  $<14\text{kiB/s}$  Ultrapeer

**Ultrapeer (UP)** Ein Hub, über den Kommunikation läuft.

Leaf Ein Randknoten, sucht über Ultrapeers.

Knoten Ein Peer oder Ultrapeer.

Arne Babenhauserheide					
Verteilte Systeme 1: peer-to-peer					
Einstieg	Grundprobleme	Gnutella	Kademia	BitTorrent Downloads	Freenet Abschluss
000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000

## Größe der Query Routing Tabellen in Gnutella

- Hashes: Normalisierte Suchwörter in der Suchanfrage oder im Dateinamen
- Größe: Variabel, Default in LimeWire 128kiB, interpolation auf größere und kleinere Tabellen möglich.
- Aktuell verfügbare Quelle: [BitSetQRTTableStorage.java](#)
- Hash-Funktion pro Suchwort: [HashFunction.java](#)

Arne Babenhauserheide					
Verteilte Systeme 1: peer-to-peer					
Einstieg	Grundprobleme	Gnutella	Kademia	BitTorrent Downloads	Freenet Abschluss
000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000

## Verteilung: Download-Mesh

- Standard HTTP Range-Requests
- Content-Addressed: `HOST/uri-res/raw/urn:sha1:HASH`<sup>14</sup>
- 5 zusätzliche Header:<sup>15</sup>

X-Alt Bestätigte Quelle für die Datei, IP/Port

X-NAlt Unerkennbare Quelle oder Quelle mit Korruptierten Daten. IP/Port

X-Gnutella-Content-URN Merkle-Tree Root-Hash

X-Text-URI /uri-res/N2X?urn:sha1:HASH;MERKLE\_TREE\_ROOT

X-Available-Ranges bytes 0-10,20-30 (Beispiel)

<sup>14</sup> <https://www.ietf.org/rfc/rfc2169.txt> und <http://www.muke24.net/doc/2015/Hash000a.html>  
<sup>15</sup> [http://rfc-gnutella.sourceforge.net/src/Partial\\_File\\_Sharing\\_Protocol\\_1.0.txt](http://rfc-gnutella.sourceforge.net/src/Partial_File_Sharing_Protocol_1.0.txt)

Arne Babenhauserheide					
Verteilte Systeme 1: peer-to-peer					
Einstieg	Grundprobleme	Gnutella	Kademia	BitTorrent Downloads	Freenet Abschluss
000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000

## Störungsresistenz: Objektvertrauen via Credence

- Jede korrekt bezeichnete Datei: 1.0
- Jede inkorrekt bezeichnete Datei: -1.0
- Wertungen anderer mit Korrelation der gemeinsamen Wertungen multipliziert.

→ <http://credence-p2p.org>

Wurde nie in ein Mainstream-Programm übernommen.

Arne Babenhauserheide					
Verteilte Systeme 1: peer-to-peer					
Einstieg	Grundprobleme	Gnutella	Kademia	BitTorrent Downloads	Freenet Abschluss
000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000

## Implementierung (Grundlagen)

- **Adressen:** Webcaches
- **Verbindung:** HTTP-Handshake, dann Binär über TCP socket + out of band Antworten via UDP
- **Verbreitung:** HTTP + swarming
- **Kommunikation:** Via Suche

Arne Babenhauserheide					
Verteilte Systeme 1: peer-to-peer					
Einstieg	Grundprobleme	Gnutella	Kademia	BitTorrent Downloads	Freenet Abschluss
000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000

## Weitere Knoten finden: Pong

Pong messages contains information about a Gnutella host. The message has the following fields

Bytes: Description:  
0-1 Port number. The port number on which the responding host can accept incoming connections.  
2-5 IP Address. The IP address of the responding host. Note: This field is in big-endian format.  
...  
\* When a Ping message is received (TTL>1 and it was at least one second since another Ping was received on that connection), a server MUST, if possible, respond with a number of Pong Messages. These pongs MUST have the same message ID as the incoming ping, and a TTL no lower than the hope value of the ping.

→ [http://rfc-gnutella.sourceforge.net/src/rfc-0\\_6-draft.html](http://rfc-gnutella.sourceforge.net/src/rfc-0_6-draft.html)

Arne Babenhauserheide					
Verteilte Systeme 1: peer-to-peer					
Einstieg	Grundprobleme	Gnutella	Kademia	BitTorrent Downloads	Freenet Abschluss
000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000

## Suche 1: Slow-Start

„Dynamic Querying“ (DQ)

- Leaf fragt einen UP nach dem anderen. Stoppt nach „genug“ Ergebnissen (um die 100).
- UP fragt Leafs und andere UPs. Stoppt nach „genug“ Ergebnissen.

Arne Babenhauserheide					
Verteilte Systeme 1: peer-to-peer					
Einstieg	Grundprobleme	Gnutella	Kademia	BitTorrent Downloads	Freenet Abschluss
000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000

## Suche 4: Dateien nach Hash finden

- Zugriff auf Magnet-Links<sup>11</sup> brauchte exakte Dateisuche.<sup>12</sup>
- Angepasstes Kademia ⇒ im Abschnitt zu Kademia.

<sup>11</sup> Magnet-Links liefern Infos für Downloads in leicht kopierbarem Link.  
<sup>12</sup> kt...: Suchanfrage, wurde kaum genutzt. Weiteres: [https://en.wikipedia.org/wiki/Magnet\\_URI\\_scheme#Design](https://en.wikipedia.org/wiki/Magnet_URI_scheme#Design)

Arne Babenhauserheide					
Verteilte Systeme 1: peer-to-peer					
Einstieg	Grundprobleme	Gnutella	Kademia	BitTorrent Downloads	Freenet Abschluss
000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000

## Kommunikation: Schwachstelle

- Chat nie wirklich verlässlich
- Kein bleibender Kontakt zu anderen
- Funktionierend:
  - Was gibts Neues? (via LimeWire: Neueste Dateien sehen)
  - Sammlung durchsuchen (Alle freigegebenen Dateien sehen)



Arne Babenhauserheide					
Verteilte Systeme 1: peer-to-peer					
Einstieg	Grundprobleme	Gnutella	Kademia	BitTorrent Downloads	Freenet Abschluss
000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000

## Verbleibende Schwächen 2008

- 10-20% Spam-Ergebnisse trotz 50 Millionen Nutzer.
  - Credence nie weiterverbreitet.
- Ein Schrott Flooding: Windows begrenzte Verbindungszahl.
- Parameter-Anpassungen beim Wachstum.
- Keine Kommentare, Peer-Chat wurde nie gut.

Arne Babenhauserheide					
Verteilte Systeme 1: peer-to-peer					
Einstieg	Grundprobleme	Gnutella	Kademia	BitTorrent Downloads	Freenet Abschluss
000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000

## Einstieg: Webcaches

Ursprünglich

- Liste langlebiger Hosts einkompiliert

Final

- Liste der letzten guten Knoten
- UDP Host-Caches: Minimalserver, die IP-Listen sammelten und die neuen weitergaben
- Beispiel: GhostWhiteCrab<sup>9</sup>



<sup>9</sup> gwc resource: <https://github.com/gtk-gnutella/gwc>

Arne Babenhauserheide					
Verteilte Systeme 1: peer-to-peer					
Einstieg	Grundprobleme	Gnutella	Kademia	BitTorrent Downloads	Freenet Abschluss
000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000

## Suche abschicken

<15 bytes GUID>0x00  
0x80 ; message type: Query  
0x07 ; TTL: 7  
0x00 ; Hops 0  
0x00,0x00,0x09 ; payload length, max: 4kiB  
0x00,0x00 ; min speed  
test foo ; payload: search criteria  
0x00 ; null-terminator, begins extensions

**GUID** Globally Unique ID. Zufällig erstellt, um Schleifen zu vermeiden.

Arne Babenhauserheide					
Verteilte Systeme 1: peer-to-peer					
Einstieg	Grundprobleme	Gnutella	Kademia	BitTorrent Downloads	Freenet Abschluss
000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000

## Suche 2: Keyword-Multicast

Query Routing Protocol (QRP)

- Suchwörter normalisiert:<sup>10</sup> lowercase, keine Akzente, ...
- Query Routing Table (QRT): Set mit schwachen Hashes von normalisierten Suchwörtern
- Automatisch hochskaliert für gewünschten Füllgrad

Intra-Ultrapeer-QRP:

- Vereinigung der Tabellen

Ähnlich: Bloom-Filter

<sup>10</sup> ungelöst: Japanische oder Chinesische Zeichen.

Arne Babenhauserheide					
Verteilte Systeme 1: peer-to-peer					
Einstieg	Grundprobleme	Gnutella	Kademia	BitTorrent Downloads	Freenet Abschluss
000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000

## Verteilung in Gnutella: Out-of-Band

Ursprünglich auf dem Suchpfad zurückgereicht, aber:

- 5 Schritte
- Durchschnittliche Lebensdauer eines Knotens:<sup>13</sup> 2h
- => Abbruch nach durchschnittlich 24 Minuten

Daher: Download-Mesh, unabhängig von der Suche



<sup>13</sup> 2h als Lebensdauer sind erstaunlich persistent. Aktuelles bei Freenet.

Arne Babenhauserheide					
Verteilte Systeme 1: peer-to-peer					
Einstieg	Grundprobleme	Gnutella	Kademia	BitTorrent Downloads	Freenet Abschluss
000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000

## Störungsresistenz: Heuristiken als Spam-Filter

Ähnlich wie E-Mail-Spamfilter.

Hat Spam auf 10-20% der Ergebnisse reduziert.<sup>16</sup>

<sup>16</sup> Ginge so ein Webshop? Wer braucht welche Garantien?

Arne Babenhauserheide					
Verteilte Systeme 1: peer-to-peer					
Einstieg	Grundprobleme	Gnutella	Kademia	BitTorrent Downloads	Freenet Abschluss
000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000

## Das Ende von Gnutella

2010: Die Zeit von Gnutella endete nicht durch technische Grenzen, sondern durch Klagen der Musikindustrie. Der Besitzer von LimeWire haftete mit seinem Privatvermögen. LimeWire bleibt freie Software, aber ohne große Verbreitung.<sup>17</sup>

Die entwickelte Technologie gerät in Vergessenheit.<sup>18</sup>

<sup>17</sup> Die Geschichte von LimeWire: <https://melmagazine.com/en-us/story/an-oral-history-of-limewire-the-little-app-that-changed-the-music-industry>

<sup>18</sup> Teile der Spezifikation: <http://rfc-gnutella.sourceforge.net/rfc-gnutella.rtf> und [https://web.archive.org/web/20070429042424/http://www.the-gsf.org/index.php?title=Wala\\_Page](https://web.archive.org/web/20070429042424/http://www.the-gsf.org/index.php?title=Wala_Page)

Arne Babenhauserheide					
Verteilte Systeme 1: peer-to-peer					
Einstieg	Grundprobleme	Gnutella	Kademia	BitTorrent Downloads	Freenet Abschluss
000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000



## Gnutella Routing Experiment

- Peers: Tisch + davor + dahinter
- Letzte 2 Hops
- Suche nach Namen
- Hash = 1. Buchstabe
- QRT<sup>19</sup>: Hash der Namen der Peers
- Intra-UP QRT: QRTs der Peers, zusammengefasst

Was müsst ihr vorher austauschen?

<sup>19</sup>QRT: Query Routing Table.



## PAUSE

--- PAUSE ---



## Sicht der Nutzer/-innen

### Werkzeuge

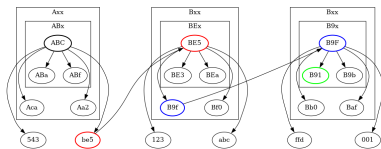
Ursprünglich Tauschbörsen: Kad in aMule, VHT in Torrent clients  
Amazon Dynamo verwendet das sehr ähnliche Chord.

### Anwendung

- Suche nach exakten Dateien
- Löst Magnet-links auf
- Server-Auswahl zum Schreiben; eventual consistency



## Suche nach b91



## Kademlia Routing-Experiment, Einstieg

IDs nach Sitzplatz:

11 - - - - - (Fenster)

10 - - - - -

01 - - - - -

00 - - - - -

000 001 010 011 100 101 110 111

(Tür) (Pult)

Präfix-Buckets (E=Eigener Bitwert):

1 4 4 2 1

EEEEx EEExx EEExx Exxxx xxxxx



## Zusammenfassung

- Distanz: key-hash XOR node-ID
- Suche: Nächstgelegenen bekannten Knoten nach besseren Knoten fragen
- Kennt mehr nahe als entfernte Knoten
- Speichern wie Suchen
- Einstieg:
  - Suche nach eigener ID
  - Erreichte Knoten nutzen Adresse und ID



## Zusammenfassung Gnutella

- Effiziente Suche nach Schlüsselworten
- TCP-basiertes Binärprotokoll, 50 mio Nutzer, 1kiB/s Leaf, 14kiB/s Ultrappeer
- Einstieg: WebCache-Server + Austausch QRT (wie Bloom-Filter)<sup>20</sup>
- Suche: Slow-Start + QRT Routing
- Verteilung: Download-Mesh
- Störungsresistenz: Heuristik oder Objektbasiert

<sup>20</sup>Set von schwachen Hashes der Suchwörter, Anzahl keys dynamisch skaliert und interpoliert



## Kademlia

Lookup in einer Verteilten Hash-Tabelle (DHT) mit xor-Metrik.

- Nutzersicht
- Suche
- Einstieg (nutzt die Suche)

## Suche in Kademlia

- Jeder Knoten hat eine zufällige ID
- Suche nach Hash → Distributed Hash Table
- Distanz zwischen Hash und ID via **xor-Metrik**<sup>22</sup>
- Schritt für Schritt in  $O(\log(N))$  zum richtigen Server

Ähnlich: Chord, Pastry.

<sup>22</sup>xor-Metrik:  $4 \text{ xor } 2 \Rightarrow 100 \text{ xor } 010 \Rightarrow 110 \Rightarrow 6$ .



## Speichern

- Suche nach Knoten nahe Hash.
- STORE: Hash + Wert.



## Kademlia Routing-Experiment, Suche

- ID berechnen (vereinfacht: zufällig<sup>23</sup>)
- Name in ID speichern.
- Andere Person: Name abfragen

<sup>23</sup>→ shared state, global; in Realität stattdessen: Hash.



## BitTorrent

- Verbreitetste Lösung für Swarming
- BitTorrent, IPFS, Blizzard-Updater
- Upload für schnelleren Download
- Koordiniert durch Tracker
- Keine Suche



## Projektideen

- Download-Mesh implementieren
  - Nur Range-Requests + magnet für Quellen
  - Quellen-Gossip via XAlt<sup>21</sup>
  - Mit Merkle-Tree oder hashliste für chunks und mit XNalt
- Suche über WebRTC in Javascript
  - flooding über vereinfachtes Binärprotokoll
  - QRP / QRT
  - Sharing als Upload in local storage
  - GGEP: Generic Gnutella Extension Protocol; Binarprotokoll für beliebige Daten.

<sup>21</sup>XAlt/XNalt: Header, der gute / kaputte Quellen beschreibt.

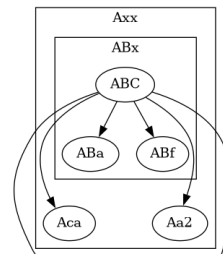


## Ziele

Sie verstehen die grundlegende Funktionsweise von Kademlia als Beispiel einer effizienten, dezentralen Hash-Suche.

Sie erkennen, wo die in Kademlia entwickelten Techniken sinnvoll genutzt werden können.

## Präfix-Buckets



## Einstieg in Kademlia

- Kontakt zu mindestens einem bestehenden Knoten.
- Suche nach eigener ID: FIND\_NODE (nah = zuständig für ID)
- Erhält Adressen + IDs der am nächsten liegenden Knoten
- Angefragte Knoten behalten auch die eigene Adresse und ID.



## Projektideen

- 
- 
- 
- 
- 

## Ziele

- Sie kennen die grundlegende Funktionsweise von BitTorrent.
- Sie verstehen, wo BitTorrent durch teilweise Zentralisierung Komplexität vermeidet.
- Sie können erklären, warum Torrent für Twitter keine optimale Wahl war, trotzdem aber Faktor 100 schneller, als die vorherige Lösung.









## Freenet als Datenbank

- Suche nach Public Key + Pfad
- → persönlicher Keyspace
- → tarballs für strukturierte Daten
- → pub-sub-Protokolle auf dezentraler Datenbank
- → Webseiten, Foren, Chat, ...

### 1 Minute Round-Trip-Time

Optimierung: Schlüssel Abonnieren, um 10k Schlüssel zu beobachten und Updates schnell zu sehen.



## Mutability: O(1) Zugriff auf neuste Version

- Nutzende: SSK@.../meine-seite-1/... → SSK@.../meine-seite-2/activelink.png
- Optimierte: USK@.../meine/seite/1
  - SSK@[key]/[sitename]-DATEHINT-[year]

HINT

46

2013-7-5

DATEHINT-[year], DATEHINT-[year]-WEEK-[week], DATEHINT-[year]-[month], DATEHINT-[year]-[month]-[day]



## Spam-Abwehr

WoT (Web of Trust): Eine von zwei praktisch genutzten Möglichkeiten. Die andere ist FMS (Freenet Message System).

- ID = USK
- Trust -100 bis 100
- Rank: Distanz → capacity
- Score: Summe über alle Wertungen: trust \* rank
- Skaliert bei 22 Nachrichten pro Tag und Person<sup>28</sup>

<sup>28</sup><https://www.draketo.de/english/freenet/deterministic-load-decentralized-spam-filter>



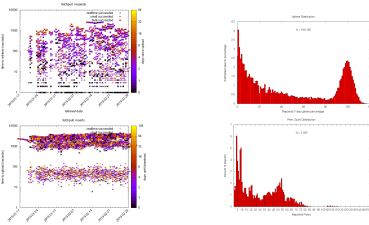
## Schnittstelle: FCP

Async für Programme:

- Put/Putdir/Get
  - Password: KSK@...
  - Inhalt: CHK@.../datei.endung
  - Schlüssel: SSK@pubkey/ordner/datei
  - Updatable: USK@pubkey/ordner/version/datei
- Subscribe to key
- Plugins kontrollieren



## Stats



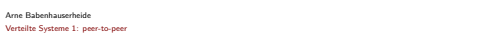
## Verschiedenes

- Queuing
- Magnet-Links
- WebRTC
- Verstreutes
- Aktuelles
- Zusammenfassung



## WebRTC

- Läuft im Browser (Javascript)
- Liefert Audio, Video, ... und Peer-Socket
- Erste Verbindung vom Server moderiert – erspart praktische Probleme
- p2p-Systeme, die nicht installiert werden müssen
- Beispiel: WebTorrent <https://webtorrent.io/>



## Capacity



- Rank 1 40 % rank 1: 100 trust, 40 Punkte als Score.
- Rank 2 16 %
- Rank 3 6 %
- Rank 4 2 %
- Rank 5 und niedriger: 1 %

Integer-Mathematik:  $2 * 6 / 100 = 0$ .



## Latenz in der Praxis

- Bis zu 1kiB, raw, realtime mode: <30s
- Große Dateien, im Manifest: ~5 min

```
Realtime
PriorityClass . 2 ;; high
MaxRetries . 0 ;; default: 10
RealTimeFlag . false
DontCompress . true
ExtraInsertsSingleBlock . 0
ExtraInsertsSplitfileHeaderBlock . 0

Bulk
PriorityClass . 3 ;; medium
RealTimeFlag . false
DontCompress . false
```



## Projektideen

- 
- 
- 
- 
- 



## Queuing in p2p-Netzen

- Alle Upload-Warteschlangen sind immer voll
- Queuing-Strategien bestimmen, welche Dateien gut funktionieren
- FIFO Dateien: Große blockieren kleine (wie Alben in Musik-Spiellisten)
- FIFO Chunks: Overhead durch Ständigen Auf- und Abbau von Verbindungen
- HRRN: Große Dateien müssen warten, stört Preview
- Komplexer: Torrent-Superseed u.ä.



## Verstreutes

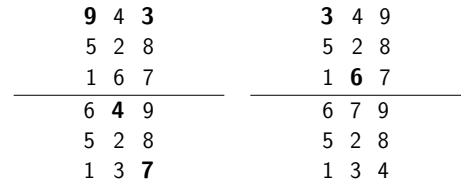
- Optimierung für Netzbetreiber oft angedacht: Clients im gleichen (Sub-)Netz bevorzugen Bei Gnutella „p4p“. Bei Pastry (Windows) laut Ghosh umgesetzt.
- Beispiel für non-greedy routing<sup>31</sup>: Random Walk in ants (Programm). Nicht durchgesetzt.
- Geld auf Probleme werfen: MaidSafe hatte 2000\$ Hardwarekosten pro Monat. 2019 abgeschaltet.<sup>32</sup> Freenet hat <20\$ pro Monat Kosten.

<sup>31</sup>Greedy-Routing: Anfragen mit rein lokaler Information an den am besten passenden Knoten weiterleiten.

<sup>32</sup>Quelle: [https://www.reddit.com/r/safesetwork/comments/erprve/dumb\\_question\\_is\\_safe\\_live/](https://www.reddit.com/r/safesetwork/comments/erprve/dumb_question_is_safe_live/)



## Swapping: Friend-to-Friend wird Small World



## Schnittstelle: Web

- Browse
- Hoch-/Herunterladen
- Plugins
- Freund-zu-Freund-Nachrichten
- Lesezeichen mit Update-Info (5 min Latenz)



## Kommunikation über Freenet

- Einstieg: Seed-keys + Captcha<sup>29</sup>-Queue: KSK-Prefix
- Suche: Nutzerspezifische Seiten mit Links, Update-Infos
- Verteilung: Gossip<sup>30</sup> keys, Dateien einfach hochladen
- Störungsresistenz: Web of Trust mit langsam steigender Sichtbarkeit

Autospawn node => Freenet als Backend, unsichtbar

<sup>29</sup>CAPTCHA: Meist Bilder, auf denen Zeichen erkannt werden müssen, um automatische Systeme auszuschließen.

<sup>30</sup>Gossip: Informationen während normaler Kommunikation von Knoten zu Knoten verteilen.



## Zusammenfassung

- Einstieg: Suche nach meiner ID bei Seednode
- Suche: Greedy Hash auf Small World
- Verteilung: Chunk-Tree mit Redundancy
- Kommunikation:
  - Einstieg: Seed-keys + CAPTCHA-Queue
  - Suche: Index-Seiten, Update-Erkennung
  - Verteilung: Dateien hochladen, Webseiten
  - Propagating Trust mit langsam steigender Sichtbarkeit



## Magnet-Links

```
magnet:?xt=urn:bitprint:TIGER_TREE.SHA1
&xt=urn:btih:BITTORRENT_INFO_HASH
&xt=urn:sha1:HASH
&xl=LENGTH
&dn=NAME
&as=LINK_WITHOUT_HASH
&xs=LINK_WITH_HASH
&kt=SEARCH_STRING
```

Netzwerk-unabhängig, Link zu HTTP und p2p-Quellen, weitverbreitet



## Aktuelles

Was gerade jetzt passiert:

- Sritely Golem: p2p distributable content for the fediverse<sup>33</sup>
- Decentralized Internet and Privacy at FOSDEM<sup>34</sup>
  - DAT, GUNet, Fediverse, Tor, ...
- In Karlsruhe: 21. Gulaschprogrammarnacht: <https://entropia.de/GPN21> 8. bis 11. Juni 2023

<sup>33</sup><https://gitlab.com/spritely/golem/blob/master/README.org>

<sup>34</sup><https://fosdem.org/> — viele Vorträge zu decentralization, privacy, ...

