peer-to-peer networks that worked

core challenges, routing, and privacy.

Dr. Arne Babenhauserheide

<2025-01-24 Fr>

peer-to-peer: where decentralization works/worked

From Gnutella and Kademlia
Over BitTorrent
To Freenet/Hyphanet

Those who don't learn from the tech of the past are doomed to repeat its mistakes — with less time and brain share, since "the problems are already solved".

DO NOT OBEY IN ADVANCE

- You are the ones who build the rules of the future.
- You are the last line of defense against building horrors.
- If you do not build it, it will not be built (well enough to work).
- Whatever you are told: **you are not replaceable**.

Do not obey in advance.

https://media.ccc.de/v/38c3-opening-ceremony#t=1343

You knew whom you invited.

Understand p2p-networks that worked in real life

> *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!*[1]

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

Understand why deployment improved but did not match expectations.

Spoiler: Cut-through routing.

[1] https://web.archive.org/web/20120807165933/http://engineering.twitter.com/2010/07/murder-fast-datacenter-code-deploys.html

Me

- Since 2004 in p2p-Development
- Since 2013 with competence
- Since 2017 Release-Manager of the Freenet/Hyphanet project
- Until 2017 PhD and PostDoc in physics
- Since 2017 Software Developer as job

Topics today

- Core challenges
- Gnutella (the first widespread, fully distributed p2p network)
- Kademlia (the most widespread DHT)
- BitTorrent
- Freenet/Hyphanet

p2p?

peer-to-peer (p2p) peers (equal partners) cooperate to grant a service to each other.

Why?

Your programs popularity exceeded your wildest dreams. 50 million people want to download it. Size 100 GiB. How much do you pay?
Relax: the game of the year in almost every award 2023/2024 launched with less than one million players.

p2p?

peer-to-peer (p2p) peers (equal partners) cooperate to grant a service to each other.

Why?

Your programs popularity exceeded your wildest dreams. 50 million people want to download it. Size 100 GiB. How much do you pay? *Relax: the game of the year in almost every award 2023/2024 launched with less than one million players.*

2025 quick checks: 5 million TB

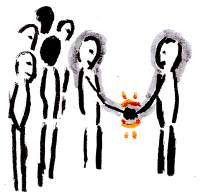
- AWS CloudFront: $100 million ($0.02/GB)
- Hetzner: 6 million € (1.19€/TB) in 137 years (2000 dl/day)
- Gnutella worked with a handful of servers for 50 million users

Topics

The core challenges p2p systems have to solve or circumvent.

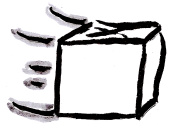
Core problems in p2p networks

- **Entry**: How to find the right place?



- **Search**: Where do I find what I need?

- **Distribution**: How to avoid bottlenecks?

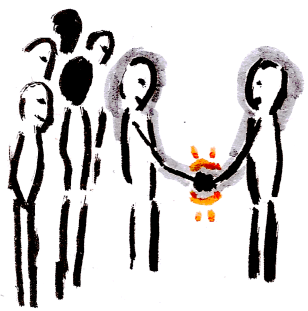


- **Communication**: How can information flow?

- ***Resistance against disruption***: *How can desired scale better than unwanted?*

Entry: How to find the right place?

- **First addresses**: How to find addresses of other nodes?
- **Choose connections**: With whom should I connect?
- **Routing-Information**: Which data do nodes need?



Structured vs. unstructured

Unstructured

- **First addresses**: A Simple list
- **Choose connections**: Choose at will
- **Routing-Information**: Exchange explicitly

Structured

- **First addresses**: Needs topology[2]
- **Choose connections**: Only some are useful
- **Routing-Information**: By choosing peers

Can I reach all directly?

[2] Topology: structure of the network.

Search: What to look for?

- **Keyword**: Gnutella, Skype (before MS)
- **Content-Hash**: Kademlia, BitTorrent VHT, Freenet/Hyphanet
- **Public Key**: Freenet/Hyphanet

BitTorrent VHT Distributed Hash Table, a DHT

DHT Distributed Hash Table

Public Key The mirror of the private key in asymmetric cryptography

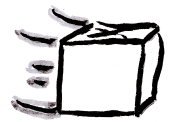
Search: Where do I find what I need?

Two concepts:

- Search Paths to existing data: Gnutella
- Put the data in the right place: Kademlia, BitTorrent VHT, Freenet/Hyphanet

Distribution: How to avoid bottlenecks?

- Centralized: Streaming by the ISP via multicast
- Swarming: users take part in the distribution
  - Coordinated by a central place: BitTorrent (Tracker)
  - Coordinated by other users: Gnutella (Download-Mesh)
  - Independently distributed chunks: Freenet/Hyphanet[3]



Download-Mesh: name of the protocol

Tracker: A server coordinating a BitTorrent-Swarm

[3] Reduces Swarming to downloading many single files but requires caching: saving transported files temporarily.

Communication: How can information flow?

- One-to-one (PM/DM/msg/Anruf/...)
- Group discussion (Chat, Forum, Video call, ...)
- Public discourse
- Learn about new content
- Metadata about content (comments, rating, ...)

Resistance against disruption

Disruption

Anything that reduces the quality of the service for users

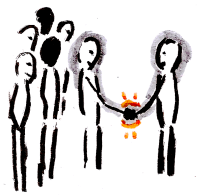
Required on every level

- **Entry**: connecting to attackers
- **Search**: spam, misinformation
- **Distribution**: poison files
- **Communication**: spam, harassment and censorship[4]

[4] "The Internet treats censorship as a malfunction and routes around it." – John Perry Barlow

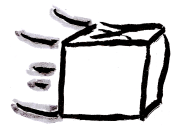
Summary

- **Entry**: First addresses and routing info



- **Search**: Keyword, Content, Public Key

- **Distribution**: Tracker, Download-Mesh, Cached fragments



- **Communication**: One-to-one, Forum, News, Comments

- ***Disruption***: *connect to attackers, spam results, misinformation, poison files, harassment.*

Gnutella

> *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.*

The simple TCP based protocol improved. By 2008 about 50 million people used it every day. It mostly disappeared after the main development companies lost in court. The tech is almost forgotten.

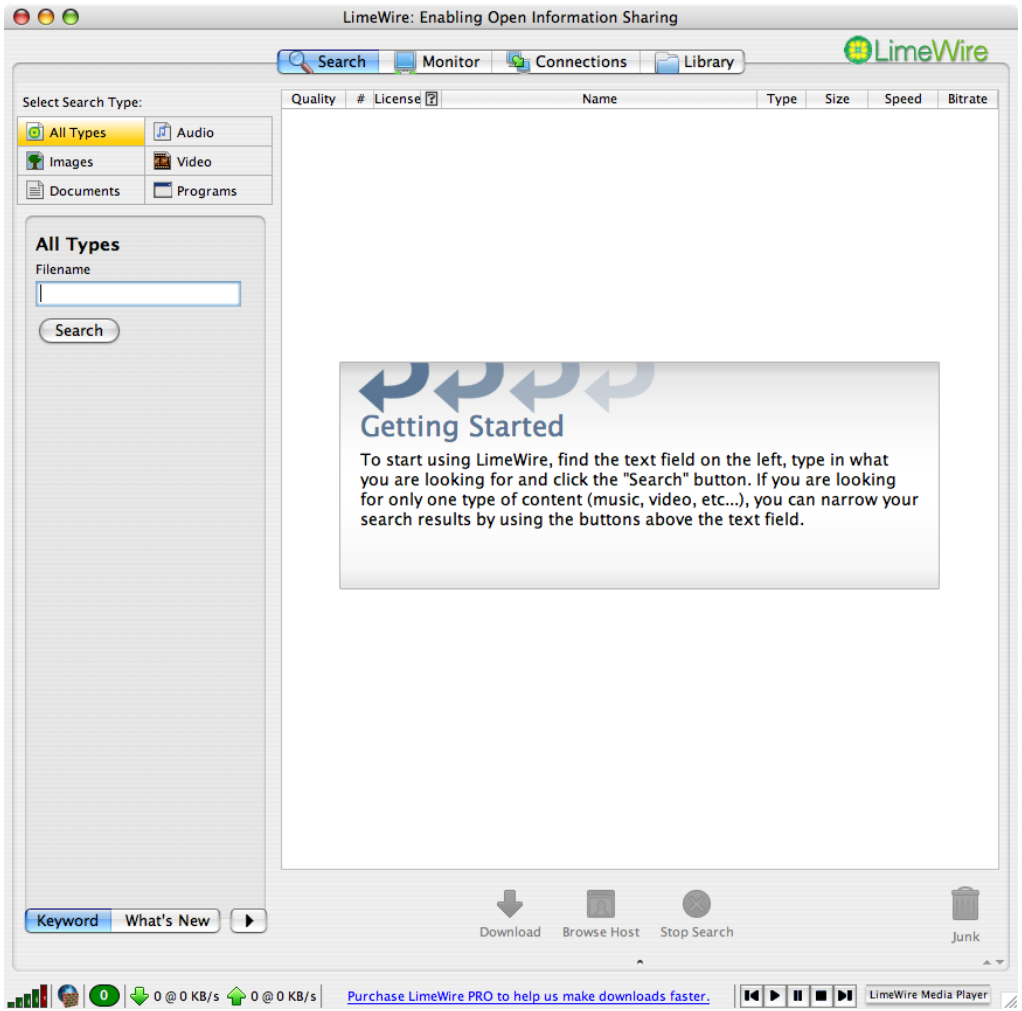
Topics

- **Users perspective**: this was Gnutella
- **Entry**: GWebCaches
- **Search**: Slow-Start + Keyword-Multicast
- **Distribution**: Download-Mesh
- **Communication**: See new files and brows the collection
- **Resistance against disruption**: Heuristics or rating matrices

Users perspective

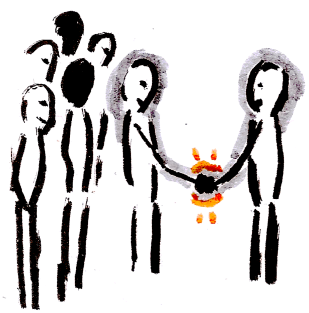
- 50 million nodes
- Search by filename and ID3 tags
- Filter to see only creative commons licensed contentnp
- Search by newest files
- Download from many sources without central coordination
- Audio streaming around 2004 (“preview”)
- *LimeWire, Bearshare, Shareaza, Phex, gtk-gnutella, . . .*

How it looked



Entry: addresses

- List of the recently connected good nodes
- UDP Host-Caches: Tiny servers that collected IP-lists and provided the most recent ones
- Example: GhostWhiteCrab[5]

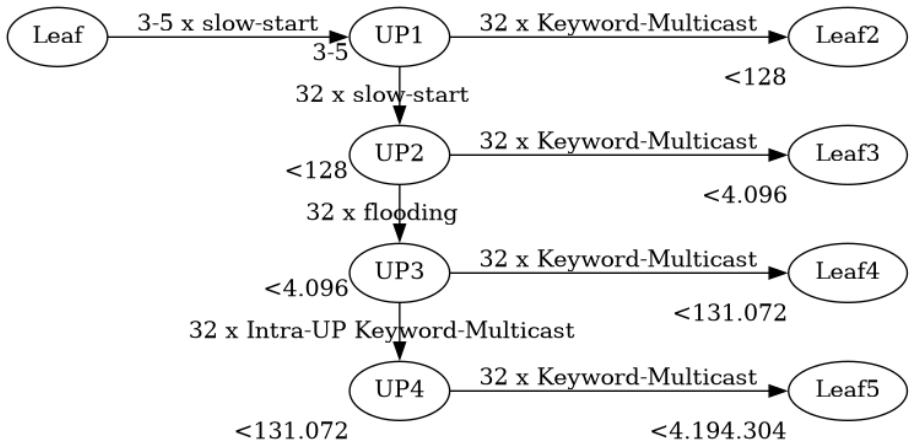


[5] gwc resource: https://github.com/gtk-gnutella/gwc

Entry: Connecting

HTTP-Handshake with feature negotiation, then binary over TCP socket + out of band answers via UDP.

Search process



Not existing file: 4 x 32 x 32 = 4.096 nodes

Load (empirical): <1kiB/s Leaf, <14kiB/s Ultrapeer

Ultrapeer (UP) A communication hub (32 peers)

Leaf An edge node, searches via Ultrapeers

Knoten A Peer or Ultrapeer.

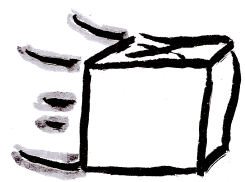
Distribution in Gnutella: Out-of-Band

Originally sent back via the search path, but:

- 5 Steps
- Average life time of a node:[6] 2h
- $=>$ Half the connections broke after 24 minutes

Solution: Download-Mesh, independent of the search.

Disadvantage: All IPs always visible.



[6] 2h lifetime are surprisingly persistent.

Distribution: Download-Mesh

- Standard HTTP Range-Requests
- Content-Addressed: HOST/uri-res/raw/urn:sha1:HASH[7]
- 5 additional Headers:[8]

X-Alt Checked source for the file, IP/Port

X-NAlt Unreachable source or node with corrupted data. IP/Port

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

X-Thex-URI /uri-res/N2X?urn:sha1:HASH;MERKLE_TREE_ROOT

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

[7] https://www.ietf.org/rfc/rfc2169.txt und http://www.nuke24.net/docs/2015/HashURNs.html

[8] http://rfc-gnutella.sourceforge.net/src/Partial_File_Sharing_Protocol_1.0.txt

Communication: Weakness

- Chat never worked well
- No lasting contact to others
- Working:
  - "What's new?" (via LimeWire: See new files)
  - Search collection (see all shared files)

Resistance against disruption: Heuristics as Spam-Filter

Similar to E-Mail spam filters, partially user-configurable.

Reduced Spam to 10-20% of the results.[9]

[9] Could this be used in a webshop? Who needs which guarantees?

Resistance against disruption: object trust via Credence

- Every correctly named/good file: 1.0
- Every falsely named/bad file: -1.0
- Ratings of others multiplied with the correlation of files both rated.

With proof that spammers would have to rate enough files correctly to spread false ratings that multiple spamming parties would cancel each other out.

→ http://credence-p2p.org

Never added to a mainstream program; LimeWire already in court.

Privacy

- IP visible in searches and downloads (originally not; direct downloads as optimization)
- Ultrapeers know their leafs
- Downloader of a file knows all other downloaders (but not how much all have)

Remaining weaknesses 2010

- 10-20% Spam-Results despite 50 million users
  - Credence was never widespread
- One step flooding: Windows (home)limited the connection count
- Required parameter-adaption during growth
- No comments, Peer-Chat never worked well
- Bad support for Asian fonts

Summary Gnutella

- Efficient Search for keywords
- TCP-based binary protocol, 50 mio users, 1kiB/s Leaf, 14kiB/s Ultrapeer
- Entry: WebCache-Server + Exchange QRT (like Bloom-Filters)[10]
- Search: Slow-Start + QRT Routing
- Distribution: Download-Mesh (mini-network per download)
- Resistance against disruption: Heuristics or object trust

[10] Set of weak hashes of the search words, number of keys scaled and interpolated dynamically

Kademlia

Lookup in a distributed Hash-Table (DHT) with xor-metric.

- Users' perspective
- Search
- Entry *(uses search)*

Users' perspective

Tools

Filesharing: Kad in aMule, VHT in Torrent clients

Amazon Dynamo uses Chord, that works very similar.

Usage

- Searches for exact files or exact keyword
- Resolves Magnet-links
- Choose the server to write to; eventual consistency

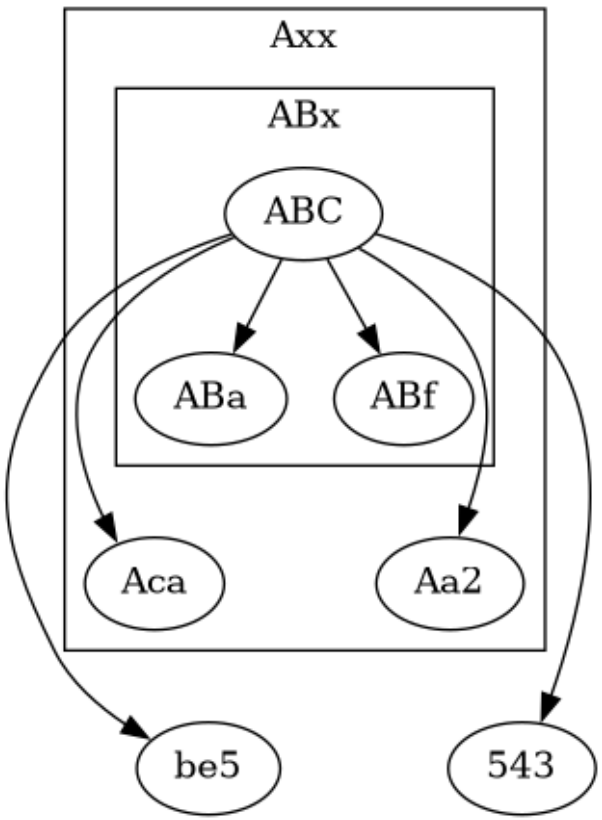
Search in Kademlia

- Every node has a random ID
- Search by Hash $\rightarrow$ Distributed Hash Table
- Distance between Hash und ID via **xor-Metric**[11]
- Step by step in O(log(N)) to the nearest node

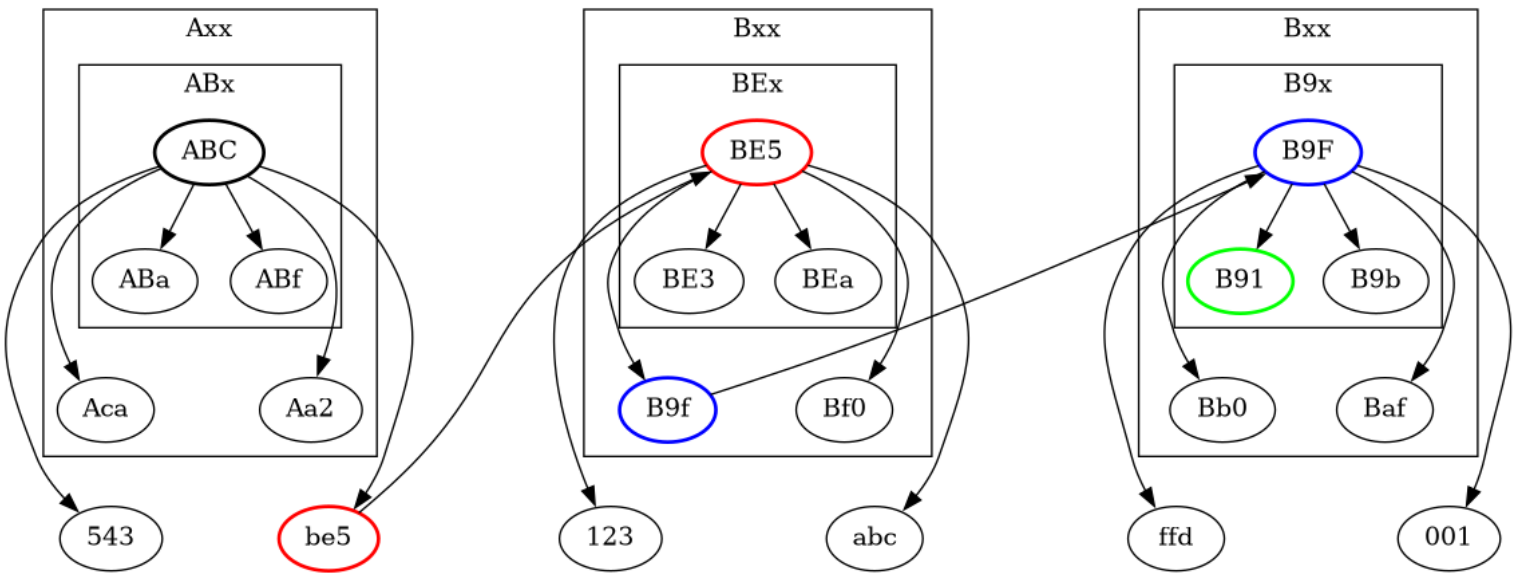
Simlar: Chord, Pastry.

[11] xor-Metric: 4 xor 2 $\Rightarrow$ 100 xor 010 $\Rightarrow$ 110 $\Rightarrow$ 6.

Prefix-Buckets



Search for b91



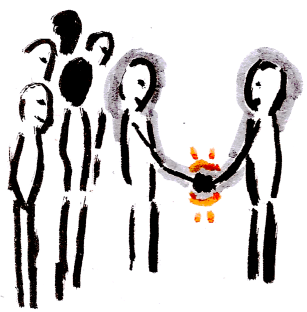
Store

- Search for node nearest to the hash.
- STORE: Hash + Value.

Think `put(key, value)` $\Rightarrow$ *"Distributed Hash Table".*

Entry in Kademlia

- Needs contact to at least one existing node.
- Search your own ID: `FIND_NODE`
  (near = responsible for ID)
- Response contains addresses and IDs
  of nodes touched
- Contacted nodes also keep your address and ID.



Privacy

- IPs of downloaders and uploaders known to responsible nodes
- Hashes of searches & searcher IP known to nodes on the path
- All addresses must be reachable globally

Summary

- Distance: key-hash XOR node-ID
- Search: ask closest known node for better nodes
- knows more close nodes than remote nodes
- Store works like searching: store where a search would land
- Entry:
  - Search for own ID
  - Touched nodes use address and ID

BitTorrent

- Most widespread solution for swarming
- BitTorrent, IPFS, Blizzard-Updater
- Upload to get faster Downloads
- Coordinated by centralized Tracker: avoids complexity
- No decentralized search

Users' perspective

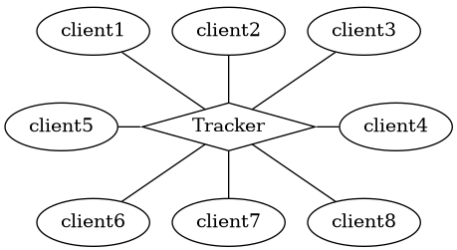
- Information from Tracker-sites
- Download with torrent-file or Magnet-Link
- Supports folders
- Today: ipfs: Websites via BitTorrent
- Supports NAT-Traversal
- Can hide the IP via Tor (as SOCKS5 proxy)

Concept of BitTorrent

Tracker: Website

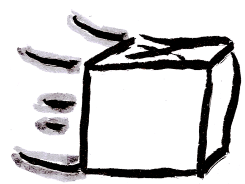
- Coordinates Swarms
- Search, Forum, Rating, Validation, Community
- Statistics: Seeder, Leecher
- Does not provide Data
- Aggregates how much nodes upload
  $\rightarrow$ Incentive

Structure



Torrent-File

- Tracker URL(-s)
- Hashes for Chunks
- Names of the File(-s)
- Can contain folders[12]



[12] http://www.bittorrent.org/beps/bep_0003.html

Incentive to Upload

- Fraction Upload/Download is checked
- Freeloaders[13] are throttled by other clients (choked: lower download rate)
- Published research gave strong focus on the incentive, in practical use the forums make more of a difference
- Tracker with login: private groups

[13] Freeloader: People who don't upload. Also: „Leech". Inverse: „Seed".

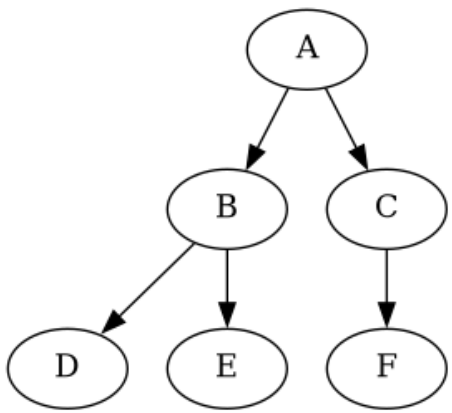
Related

- VHT in addition/instead of Tracker possible (Kademlia)
- Free and Open Protocol with many implementations
- Development within the Community
- IPFS uses Torrents as decentralized cache for Websites

Torrent for Twitter-Deployment

- Cost at Twitter: Transport over many steps
- Torrent transmits in fragments.

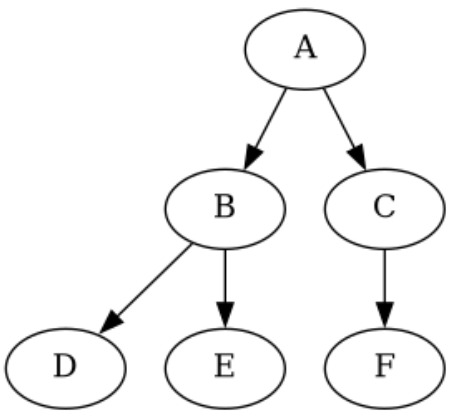
Wish



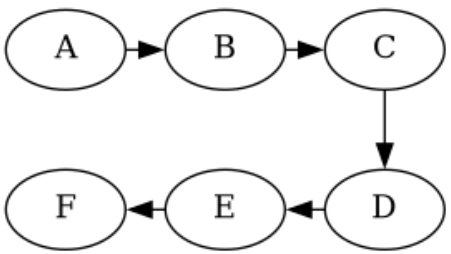
Torrent for Twitter-Deployment

- Cost at Twitter: Transport over many steps
- Torrent transmits in fragments.

Wish



Reality



cat . . . ssh tee . . .
Cut-through / streaming.

Privacy

- Tracker knows
  - who downloads
  - how much they have
  - how much they upload
- Users know what the tracker tells them
- No search, except by Hash: private trackers stay private
- IP hiding via Tor is possible but blocked by most trackers

Summary

- Tracker and Clients
- Tracker: statistics and coordination
- Torrent-File with Piece-Info

Freenet/Hyphanet

Censorship-resistant, privacy respecting communication on friend-to-friend network

Decentralized database with pubkey-access.

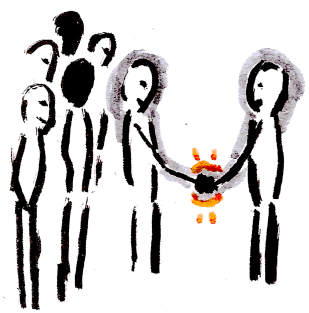
- Users' perspective
- Entry
- Small-World
- Search
- Distribution
- Communication
- Usage
- Privacy

Users' perspective

- Web-interface: decentralized websites
- Plugins with E-Mail and Microblogging
- External programs like Chat and forums with Freenet/Hyphanet as database using HTTP-like API (FCP)

Entry in Freenet/Hyphanet

- Opennet:
  - Similar to Kademlia: choose known Seednode[14], Seednode searches ID $\rightarrow$ node references
  - Difference to Kademlia: Not just IP: referenz with keys
- Friend-to-Friend:
  - Fixed connections
  - Knoten swap their IDs, to reconstruct the social Small-World-Network
    $\Rightarrow$ minimize Overlay-costs.



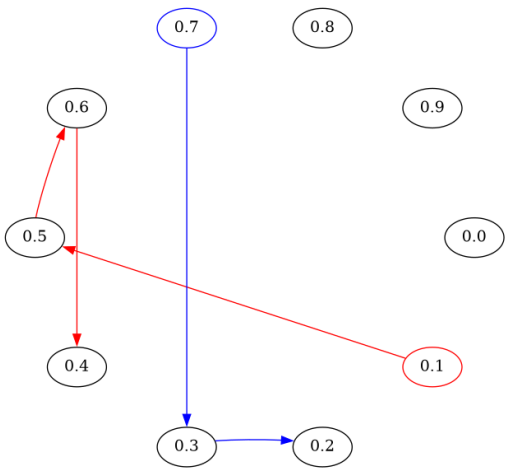
[14] Seednode: known node that arranges connections to others.

Small-World-Network (scale free network)

- Many short and few long connections.
- 6 deegrees of separation via snail mail: Our acquaintances formm a small-world network
- Kleinberg-Network: Probability to be connected: $\frac{1}{d^x}$, d = distance, x = dimension.
- Freenet/Hyphanet: $x = 1$

Freenet/Hyphanet Search

- Like Kademlia, but forwarded hop by hop → no global reachability or visibility
- Can search by public key
- Keyspace: [0.0 : 0.1)



Types of Keys

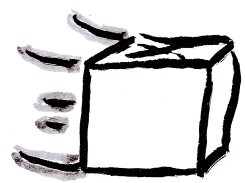
- CHK: Content Hash
- KSK: Keyword Subspace: Password
- SSK: Signed Subspace: Public Key
- USK: Updatable Subspace: SSK with version

Format:
`XXK@routing,encryption/tarball-name/path/to/file.ext`

Can omit path and name (smaller $\rightarrow$ optimization).

Distribution in Freenet/Hyphanet

- Network saves content → distributed Cache
- Files saved encrypted, as 32 kiB fragments with 100% redundancy
- Manifest contains keys of the fragments as CHKs
- Limited lifetime: Effectively LRU-Cache:[15]
  - Saving overwrites randomly chosen fragments
  - Access restores overwritten fragments
- Upload to exsting key+path: collision
  → Effectively immutable



[15] LRU: Least Recently Used. Delete oldest first.

Freenet/Hyphanet as Database

- Search by Public Key + Path
- $\rightarrow$ personal keyspace
- $\rightarrow$ tarballs for structured data (i.e. website)
- $\rightarrow$ pub-sub-protocols on decentralized database
- $\rightarrow$ Websites, Forums, Chat, ...

Optimization: Subscribe to keys to be able to watch 10k keys and see updates quickly.

Getting low latency

- Up to 1kiB, raw, realtime mode: <20s; 30s – 90s RTT
- Larger files, in manifest (folder): ~5 min

Must do it exactly right or it will be slow. Like 14kB websites.

Realtime (small, interactive)

```
PriorityClass . 2 ;; high
MaxRetries . 0 ;; default:
↪  10
RealTimeFlag . true
DontCompress . true
ExtraInsertsSingleBlock .
```

Bulk (large, slower)

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

Spam-protection

WoT (Web of Trust): One of two possibilities in actual use. The other is FMS (Freenet/Hyphanet Message System).

- ID = USK
- Trust -100 to 100
- Rank: Distance → capacity
  - Rank 1 40 % to 1 %
  - rank 1: 100 trust = +40 points score.
- Score: Sum of all ratings: trust * rank
- Can scale to arbitrary size at 22 messages per day and person[16]

[16] https://www.draketo.de/english/freenet/deterministic-load-decentralized-spam-filter

Communication via Freenet/Hyphanet

- Entry: Seed-keys + Captcha[17]-Queue: KSK-Prefix
- Search: User-specific pages with links, update-infos
- Distribution: Gossip[18] keys, just upload files
- Resistance against disruption: Web of Trust with visibility increasing by interaction

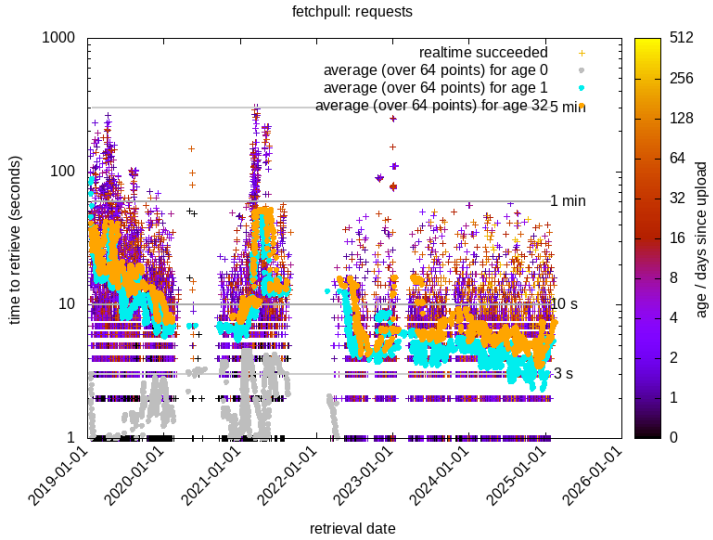
Autospawn node => Freenet/Hyphanet transparently

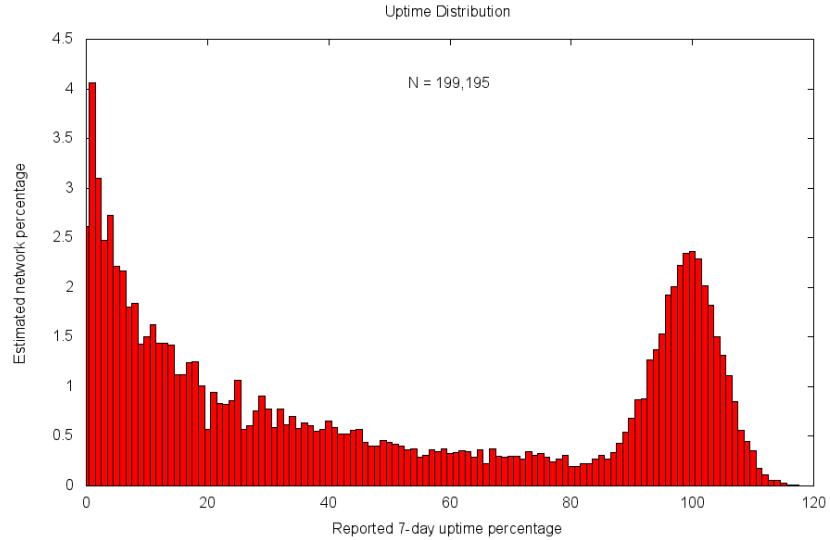
[17] CAPTCHA: Usually images with letters that are part of a watched key.

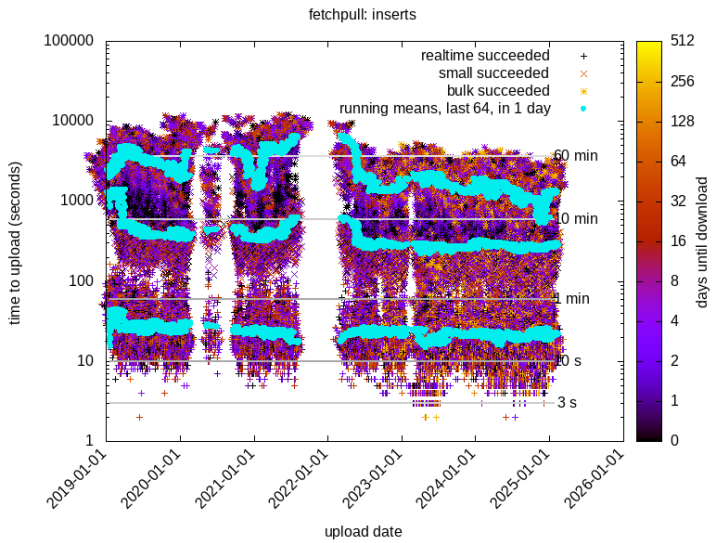
[18] Gossip: attach information to normal communication to distribute it.

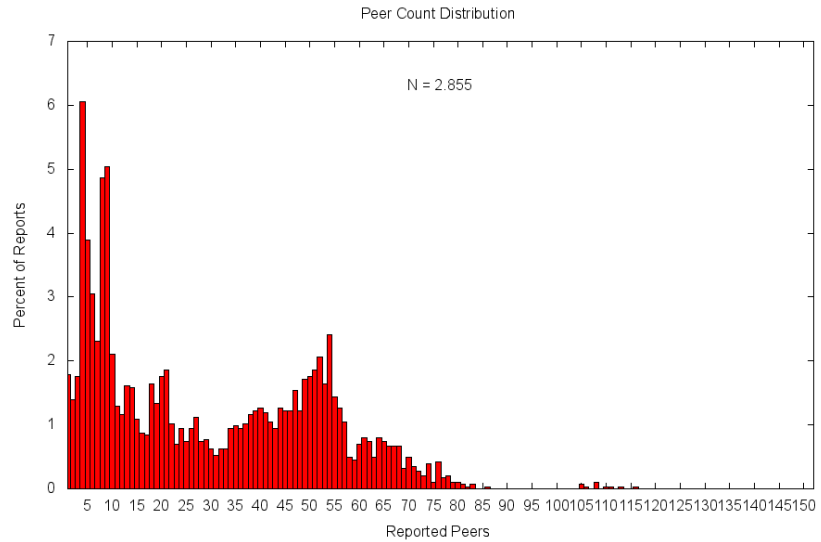
Usage

Stats









Dr. Arne Babenhauserheide

peer-to-peer networks that worked

Privacy

- Seednodes can enumerate all opennet nodes.
- With pure Friend-to-Friend mode, ISPs may recognize a high volume of encrypted UDP packets.
- Directly connected peers can see that a search was **started by or forwarded by** a node (HTL with randomized decrement).
- With friend-to-friend mode, only trusted nodes can be directly connected. Needs key exchange with friends.
- Nodes can see whether chunks for which they are the best location are accessed but not by whom or how often.
- IDs in communication are stable pseudonyms: rotated session keys — ratchet — only in custom implementations.

Summary

- Entry: Search at a seednode for my ID
- Search: Greedy Hash search on Small World Network
- Distribution: Chunk-Tree with Redundancy
- Communication:
  - Entry: Seed-keys + CAPTCHA-Queue
  - Search: Index-Pages, Subscribe for Updates
  - Distribution: Upload files, websites
  - Propagating Trust with slowly rising visibility

Aside

- Guarantees
- Magnet-Links
- WebRTC
- Lessons learned

Guarantees

They scale by giving few guarantees. From strong to weak:

- Tampering: All networks here prevent tampering with a file being downloaded.
- Access control: You need the keys to files to gain access: hashes or public keys.
- Availability: None of the p2p networks here guarantee it:
  - Data may not exist
  - Connections may break
  - Names may be wrong

To give availability guarantees, take CRDTs or similar as starting point for proofs.

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

WebRTC

- Runs in the Browser (Javascript)
- Provides Audio, Video, . . . , and **Peer-Socket**
- First connection moderated by server – avoids many problems
- p2p-Systems, that don't need installation
- Example: WebTorrent `https://webtorrent.io/`

Thoughts

- All Upload Queues are always full. Like all disks are always full. Ask the Large Scale Data Facility at KIT/SCC.
- Optimization for ISPs often thought about: Prefer clients in same (sub-)net. Gnutella: „p4p“. Pastry (Windows) uses it according to Ghosh.
- Example for non-greedy routing[19]: Random Walk in ants (programm). Did not gain traction.
- Throwing money on problems: MaidSafe had 2000$ hardware cost per month. Shut that down 2019.[20] Freenet/Hyphanet has <20$ cost per month.

[19] Greedy-Routing: forward requests with local information to the best node.

[20] Source: https://www.reddit.com/r/safenetwork/comments/erpvee/dumb_question_is_safe_live/

Fallacies of distributed Systems, extended version

1. The network is reliable
2. The network is secure
3. The network is homogeneous
4. Topology does not change
5. Latency is zero
6. Bandwidth is infinite
7. Transport cost is zero
8. There is one administrator

1. Hard disks don't fail
   Files stay intact
2. Power is stable
3. IPs are reachable
4. Constant factors are negligible
5. APIs stay compatible
6. Textfiles are simple

Current developments

What's happening today:

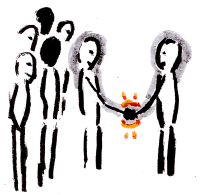
- Spritely Golem: p2p distributable content for the fediverse[21]
- Decentralized Internet and Privacy at FOSDEM[22]
  - DAT, GNUnet, Fediverse, Tor, ...
- In Karlsuhe: 23. Gulaschprogrammiernacht: https://entropia.de/GPN23/en 19. bis 22. Juni 2025

[21] https://gitlab.com/spritely/golem/blob/master/README.org

[22] https://fosdem.org/ — viele Vorträge zu decentralization, privacy, ...

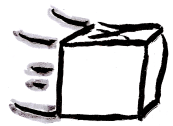
Summary: Core challenges

- **Entry**: First addresses and routing info



- **Search**: Keyword, Content, Public Key

- **Distribution**: Tracker, Download-Mesh, Cached fragments



- **Communication**: One-to-one, Forum, News, Comments

- ***Disruption***: *connect to attackers, spam results, misinformation, poison files, harassment.*

Summary: Implementations

	Einstieg	Suche
Gnutella	WebCache	Slow-Start + Keyword-Multicast
Kademlia	Search own ID	xor-Hash-hierarchy
BitTorrent	Tracker-URL	Kademlia / Tracker / Web
Freenet/Hyphanet	Seed-Nodes search ID	Greedy on Small World
WebRTC	WebRTC Server	-

	Verteilung	Störung
Gnutella	Alt+NAlt, Range, Merkle-Tree	Heuristik/Credence
Kademlia	*various*	-
BitTorrent	Torrent	Rating on Tracker
Freenet/Hyphanet	Chunk-Tree with Redundancy	Propagating Trust
WebRTC	-	-

Good luck!

My wish is that 5 years from now
some of you look back and say:

*"What I learned in the p2p lecture
was one of the pillars of my success."*

Verweise I

Bilder:

Merkle Tree Patent 1982
https://worldwide.espacenet.com/patent/search/family/022107098/publication/US4309569A?q=pn%3DUS4309569
Eingereicht 1979 als Methode Diffie-Authentication günstiger zu machen.

Weitere Knoten finden: X-Try

Beim Handshake (wie HTTP):

```
When rejecting a connection, a servent 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
```

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
      servent 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 hops value of the
      ping.
```

→ http://rfc-gnutella.sourceforge.net/src/rfc-0_6-draft.html

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

Suche 4: Dateien nach Hash finden

- Zugriff auf Magnet-Links[23] brauchte exakte Dateisuche.[24]
- Angepasstes Kademlia ⇒ im Abschnitt zu Kademlia.

[23] Magnet-Links liefern Infos für Downloads in leicht kopierbarem Link.

[24] kt=. . . : Suchanfrage, wurde kaum genutzt. Weiteres:
https://en.wikipedia.org/wiki/Magnet_URI_scheme#Design

Weiteres

Gnutella Routing Experiment

- Peers: Tisch + davor + dahinter
- Letzte 2 Hops
- Suche nach Namen
- Hash = 1. Buchstabe
- QRT[25]: Hash der Namen der Peers
- Intra-UP QRT: QRTs der Peers, zusammengefasst

Was müsst ihr vorher austauschen?

[25] QRT: Query Routing Table.

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

Warum nicht p2p?

- Gestiegende Leistung von Servern. *Sturm: dwd*[26] *hielt größtenteils Stand (durch vereinfachte Seite*[27] *)*
- Handies sind durch Batterie und Netz begrenzt $\rightarrow$ keine kontinuierliche Leistung. (Nachts möglich?)
- Viele der einfachen Lösungen unmöglich, z.B. Geld auf Probleme werfen.

[26] dwd: Deutscher Wetterdienst.

[27] $\Rightarrow$ gibt es eine einfachere Lösung?

Weiteres

Schlüssel zum Licht

Störquellen

Sammeln am Flipchart

[28] Werbung ist Spam durch die genutzte Plattform.

Störquellen

Sammeln am Flipchart

Quellen

- **Parasiten**: Bessere Leistung auf Kosten Anderer (leecher).
- **Trolle**: Kein Finanzinteresse, minimale Resourcen, nutzen jegliche Lücke.
- **Spammer**: Erfolg durch Verbreitung eigener Inhalte.[28]
- **Konkurrenten**: Erfolg durch verringerte Qualität des Systems.
- **Angreifer**: Erfolg durch Schädigung von Nutzern.

[28] Werbung ist Spam durch die genutzte Plattform.

Weitere Eigenschaft: Grad der Verteilung

Serverkoordinierte Teilgruppen bis vollständig dezentrale Interaktion.

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.

Suche 2: Keyword-Multicast

Query Routing Protocol (QRP)

- Suchwörter normalisiert:[29] 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

[29] ungelöst: Japanische oder Chinesische Zeichen.

Weiteres

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.

Weiteres

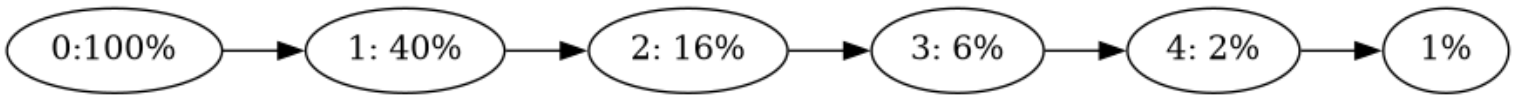
Mutability: O(1) Zugriff auf neuste Version

- Nutzende: `SSK@.../meine-seite-1/...` → `SSK@.../meine-seite-2/activelink.png`
- Optimiert: `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]

Capacity

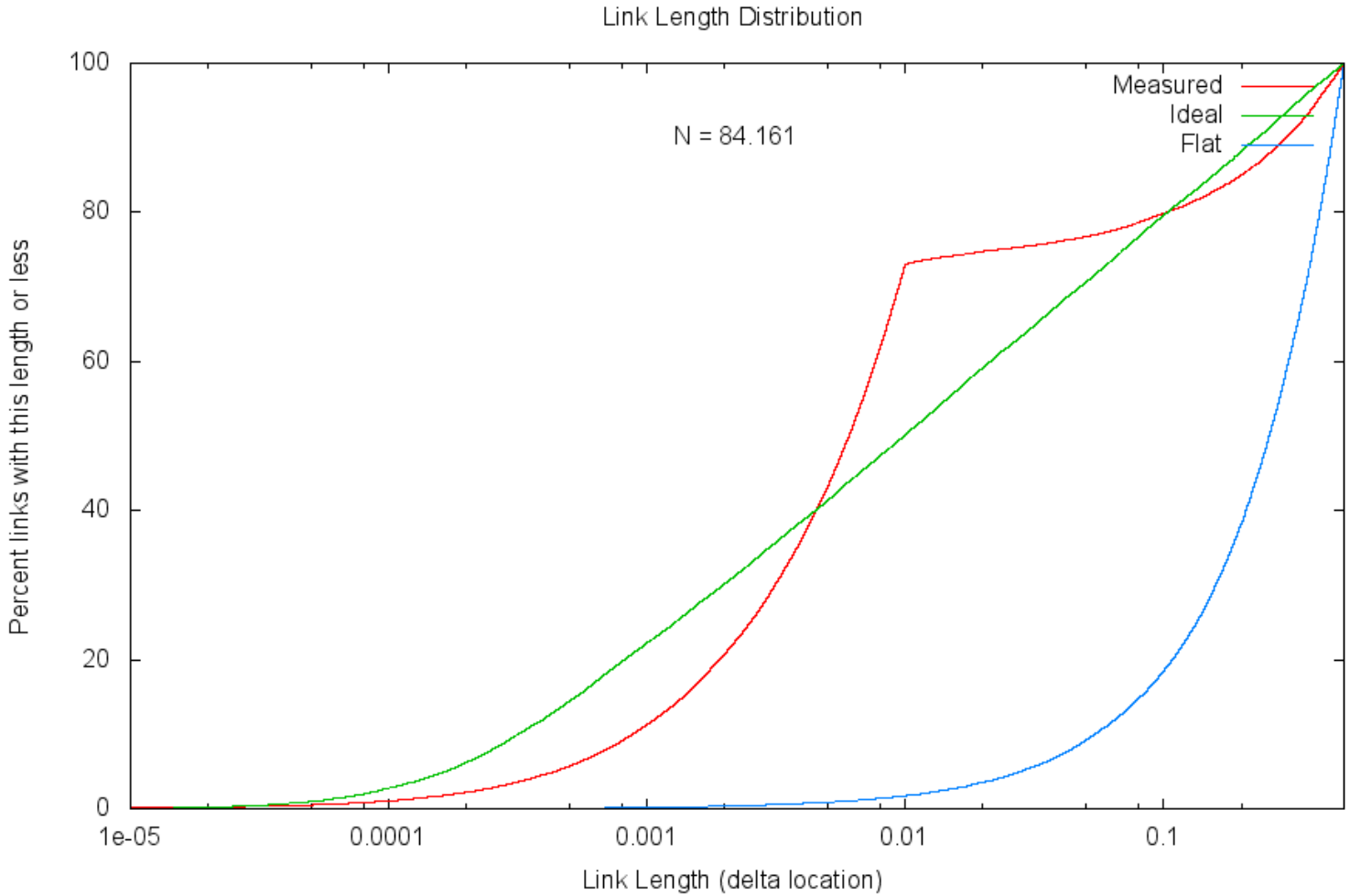


- Rank 1 40 %. rank 1: 100 trust, 40 Punkte als Score.
- Rank 2 16 %
- Rank 3 6 %
- Rank 4 2 %
- Rank 5 und niedriger: 1 %

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

Weiteres

Theoretische und gemessene Link-Längen



Weiteres

Swapping: Friend-to-Friend wird Small World

9	4	**3**
5	2	8
1	6	7

6	**4**	9
5	2	8
1	3	**7**

3	4	9
5	2	8
1	**6**	7

6	7	9
5	2	8
1	3	4

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; Verwaltung läuft auf zentralisierten Trackern

Projektideen

- Download-Mesh implementieren
  - Nur Range-Requests + magnet für Quellen
  - Quellen-Gossip via XAlt[30]
  - 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.

[30] XAlt/XNalt: Header, der gute / kaputte Quellen beschreibt.