

Measurement Study of Wuala, a Distributed Social Storage Service



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Thomas Mager - Master Thesis



Advisors:

Prof. Ernst Biersack
Prof. Thorsten Strufe
Prof. Pietro Michiardi



Agenda



TECHNISCHE
UNIVERSITÄT
DARMSTADT

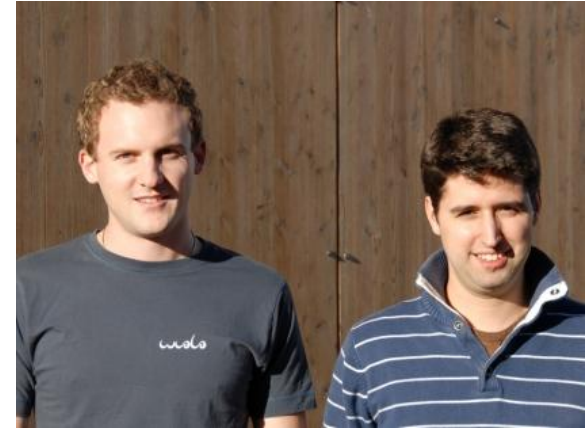
1. Introduction
2. Motivation
3. Basics
4. General Experimental Setup
5. Experiments
6. Conclusion



Introduction

What is Wuala?

- On-line storage solution
- Leverages resources of peers to store files
- Development started at ETH Zürich 2004
- Company founded 2007
- Launched in 2008
- Acquired by LaCie 2009
- Pioneer in this area - first widely-used system!



Dominik Grolimund, Luzius Meisser

ETH

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich



Features



TECHNISCHE
UNIVERSITÄT
DARMSTADT

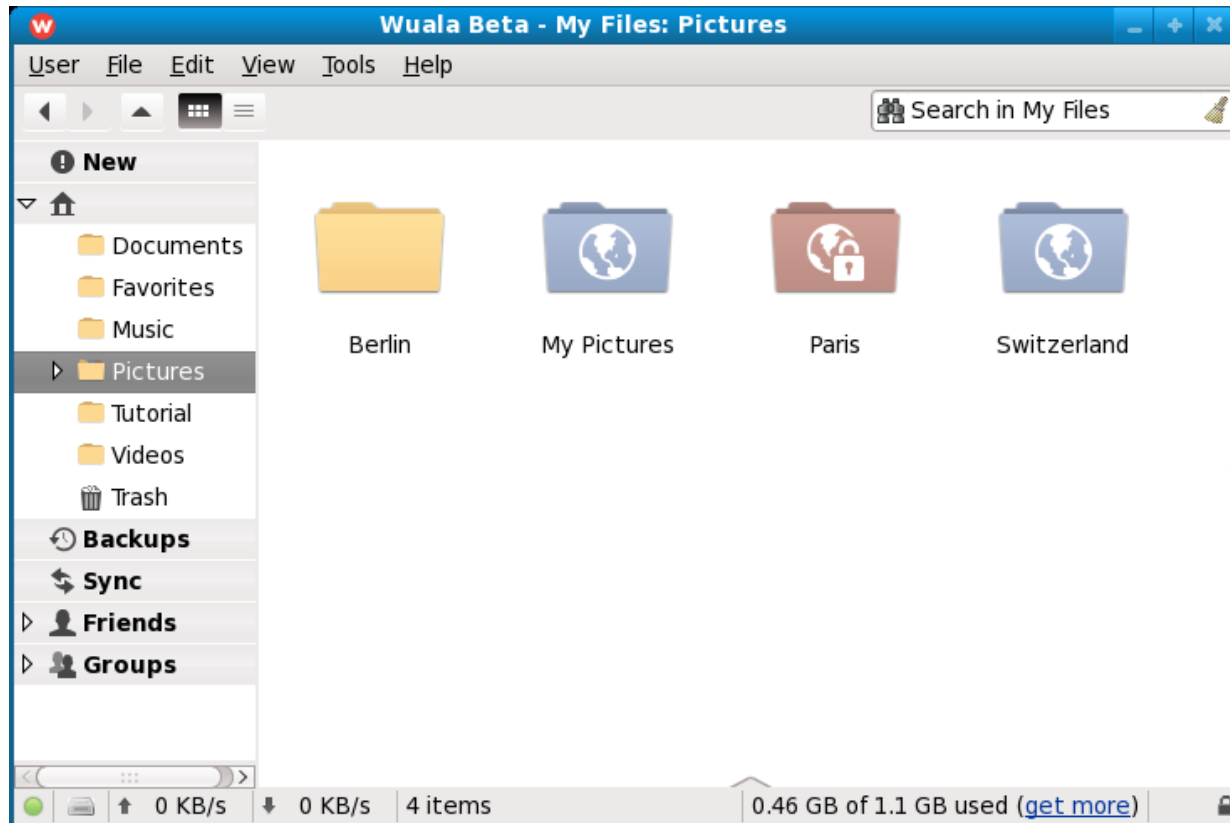
- 1 GiB storage for free, else trade local storage or pay
- Automatic backup
- Share files with friends
- All files encrypted locally
- Mount on-line storage into file system hierarchy
- File versioning



Wuala Application



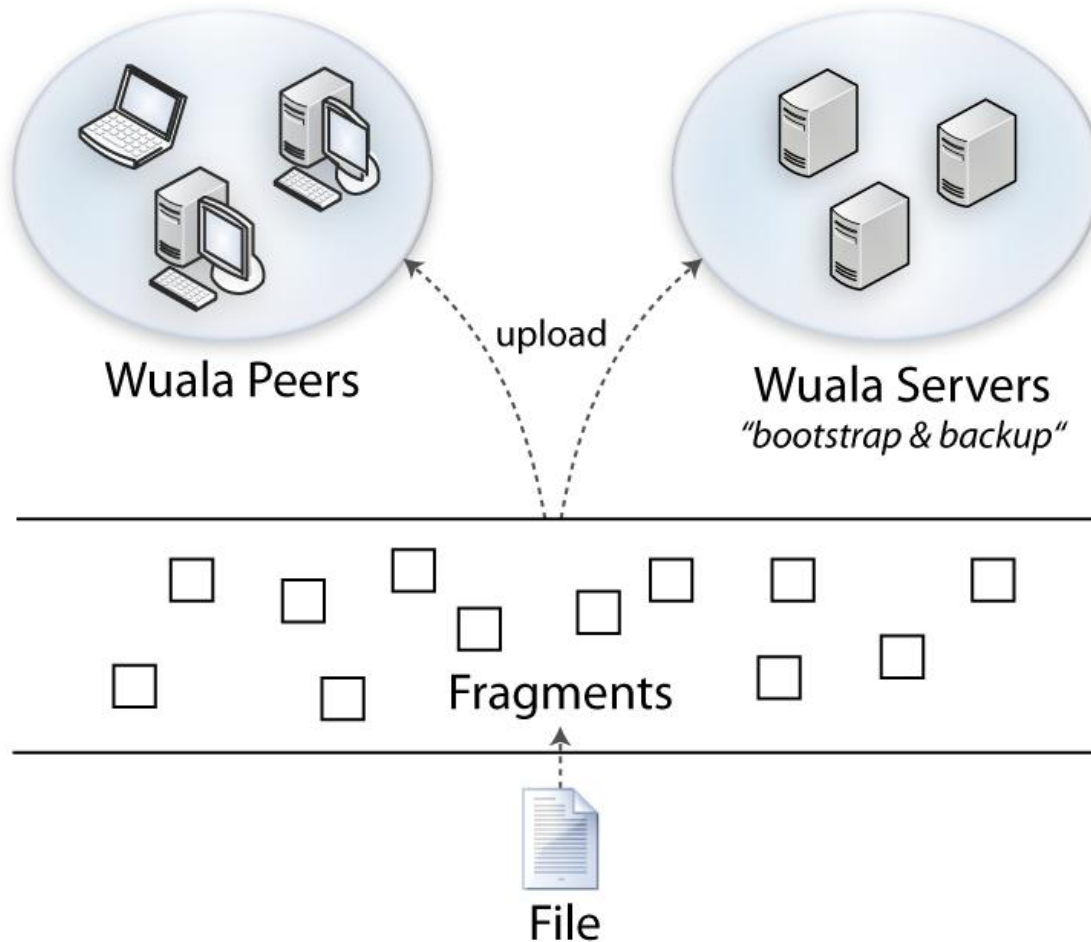
TECHNISCHE
UNIVERSITÄT
DARMSTADT



Wuala Application



General Architecture



Motivation

How to make sure that files are always accessible?

→ Crucial because of high churn rate

Requirements:

- Availability
- Reliability
- Consistency

How do peers / servers contribute to achieve this?

Redundancy Required

Two main approaches:

- Replication
- Erasure Coding

Replication



TECHNISCHE
UNIVERSITÄT
DARMSTADT

$$A_{\text{replication}} = 1 - (1 - p)^r$$

p = node availability

r = redundancy factor



$$A_{\text{replication}} = 1 - (1 - p)^r$$

p = node availability

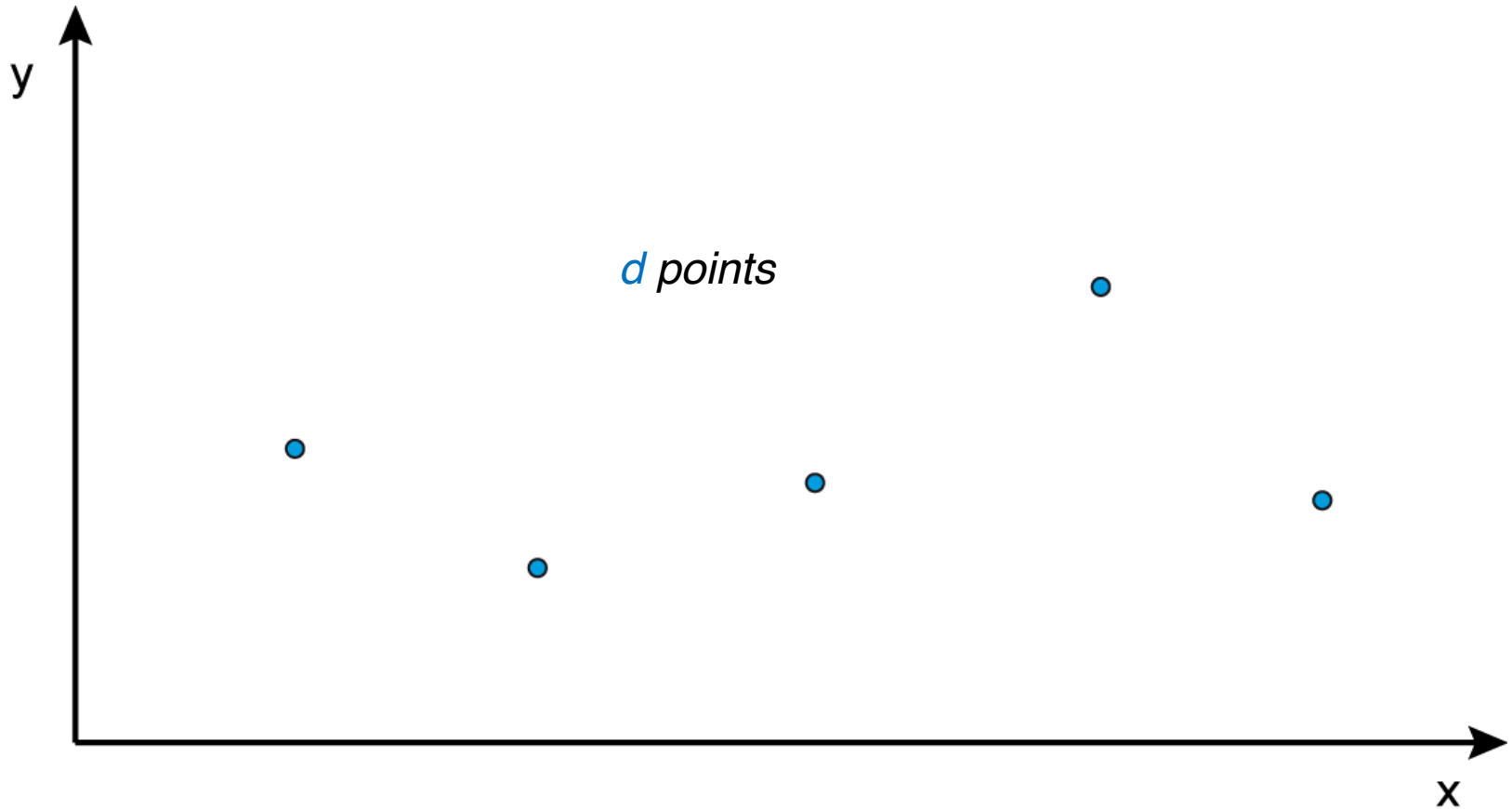
r = redundancy factor

Availability target of 99.9%

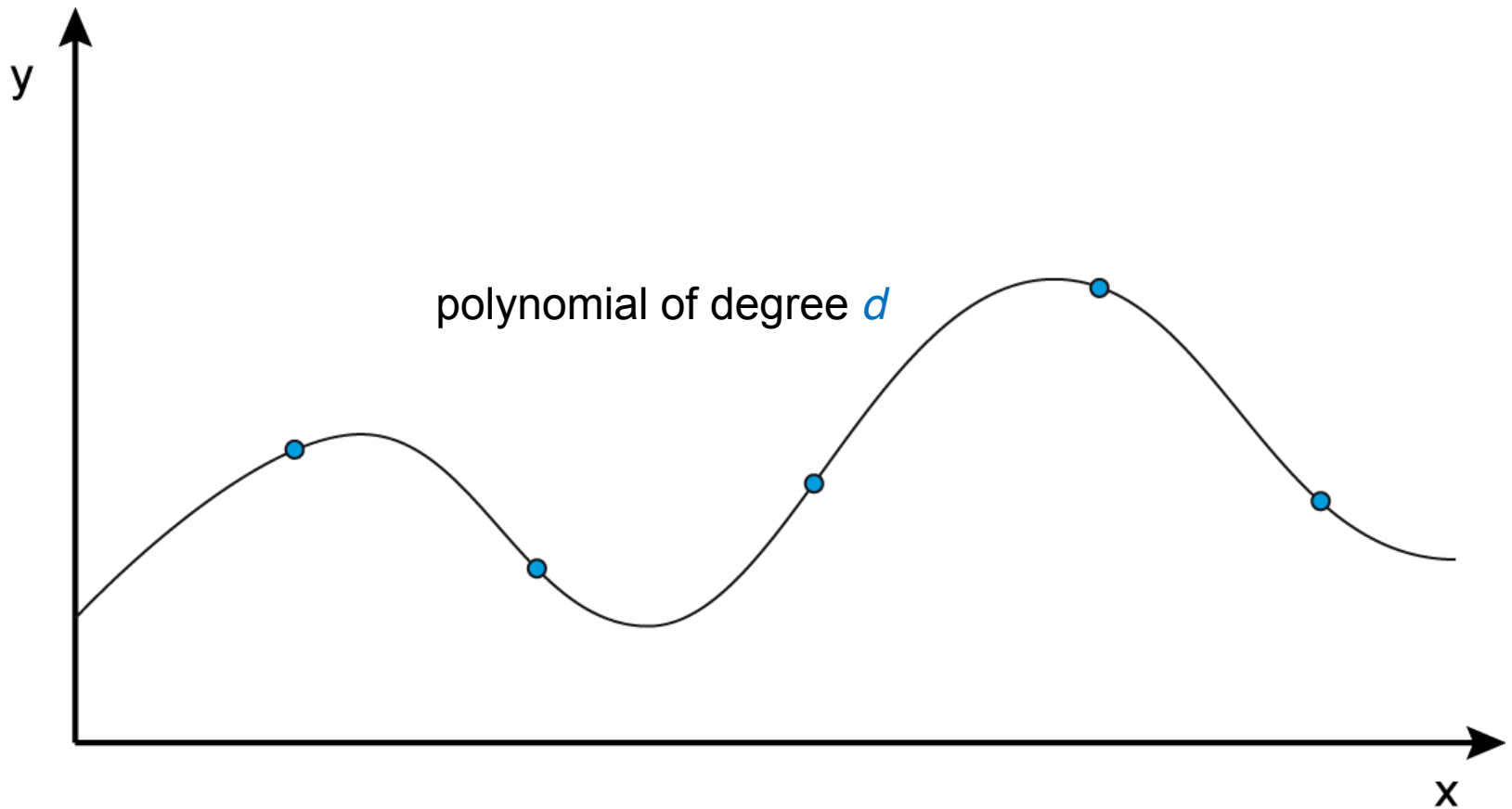
presuming a mean node availability of 50%:

$$r = 10$$

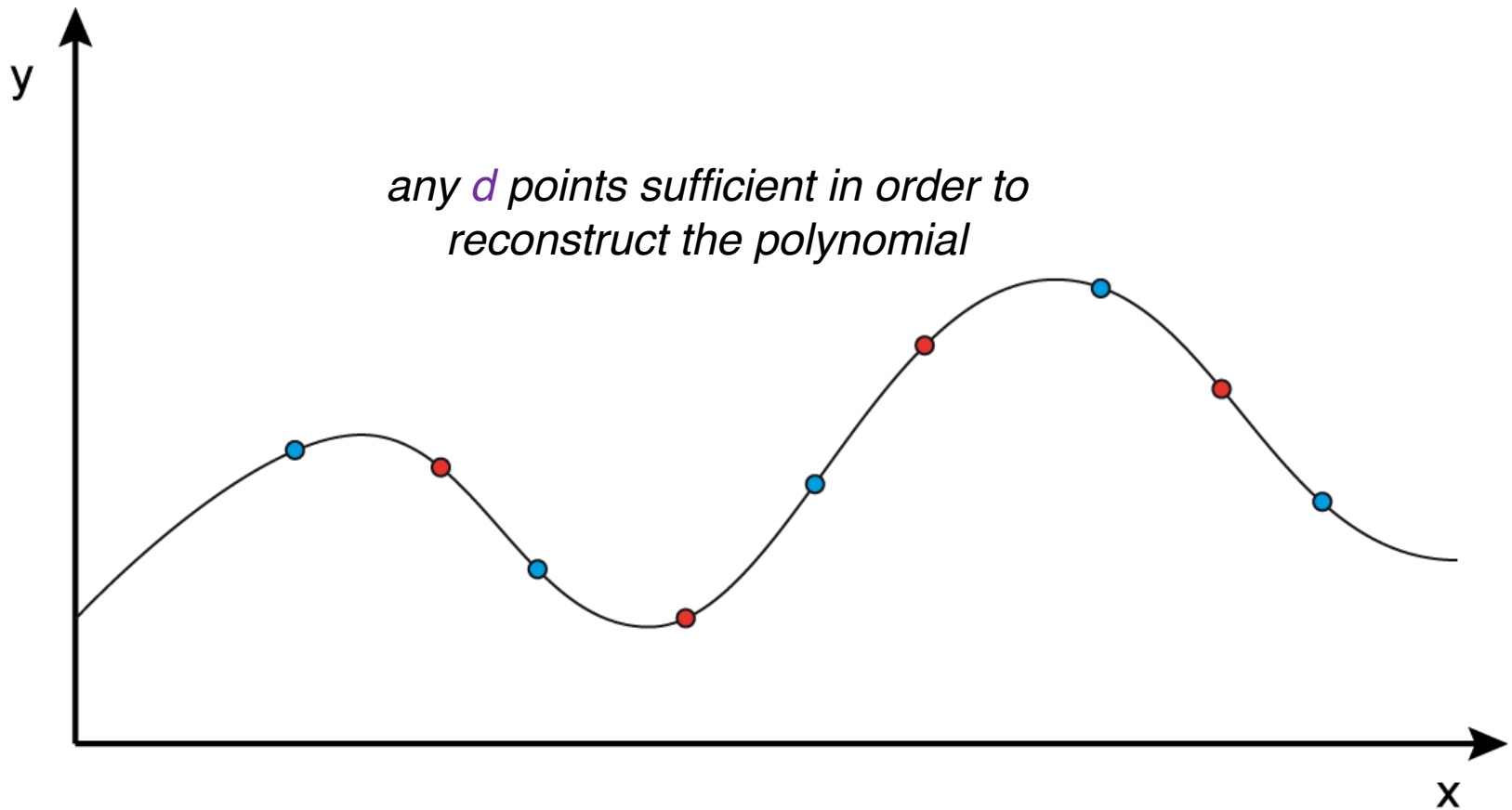
Erasure Coding



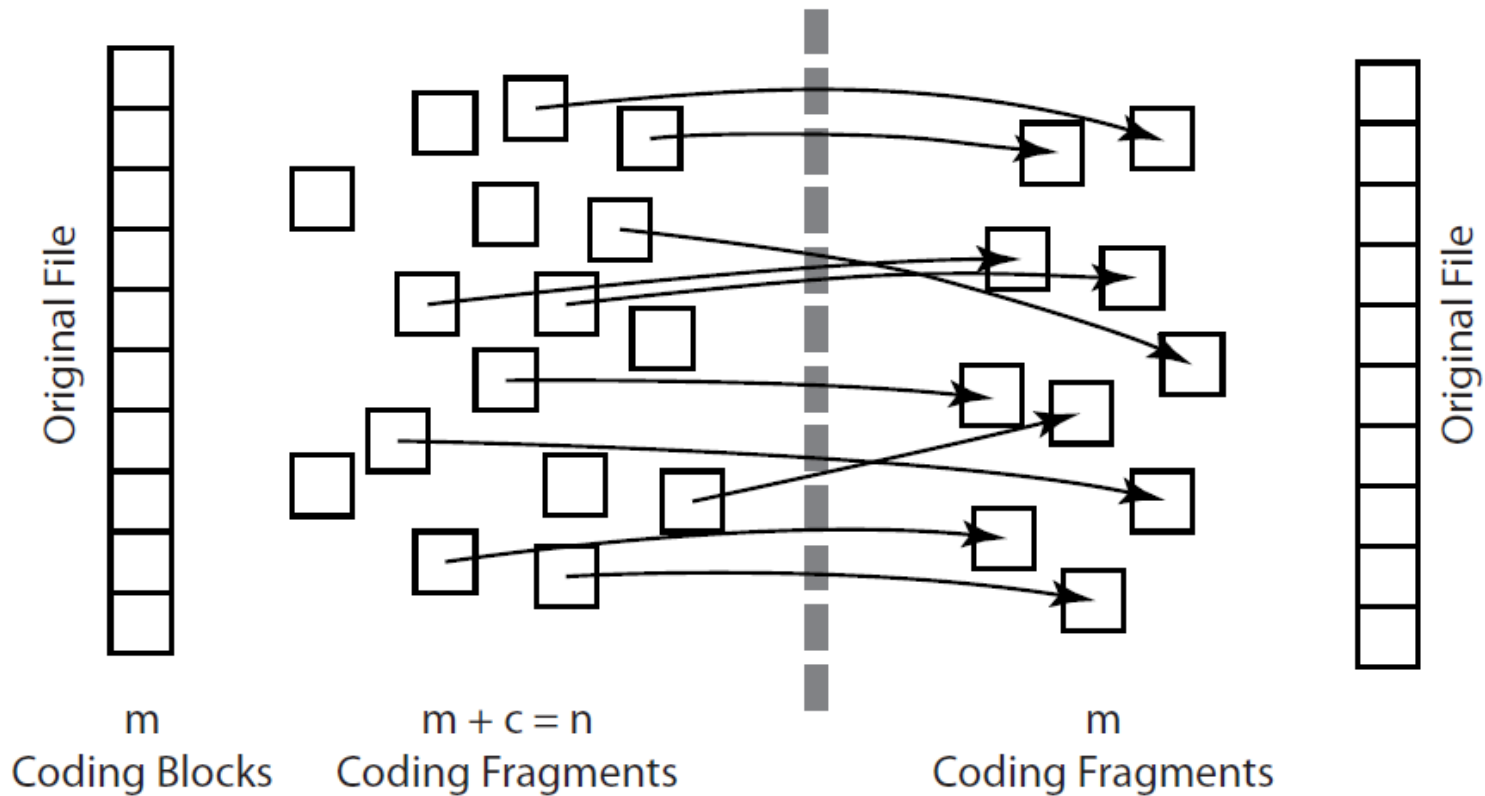
Erasure Coding



Erasure Coding



Erasure Coding



Erasure Coding



$$A_{\text{erasurecoding}} = \sum_{i=m}^n \binom{n}{i} p^i (1-p)^{n-i}$$

p = node availability

$r = n / m$ = redundancy factor



$$A_{\text{erasurecoding}} = \sum_{i=m}^n \binom{n}{i} p^i (1-p)^{n-i}$$

p = node availability

$r = n / m$ = redundancy factor

Availability target of 99.9%

presuming a mean node availability of 50%:

$$r = 2.49$$

How can Wuala be studied?

Wuala is a closed source project written in Java

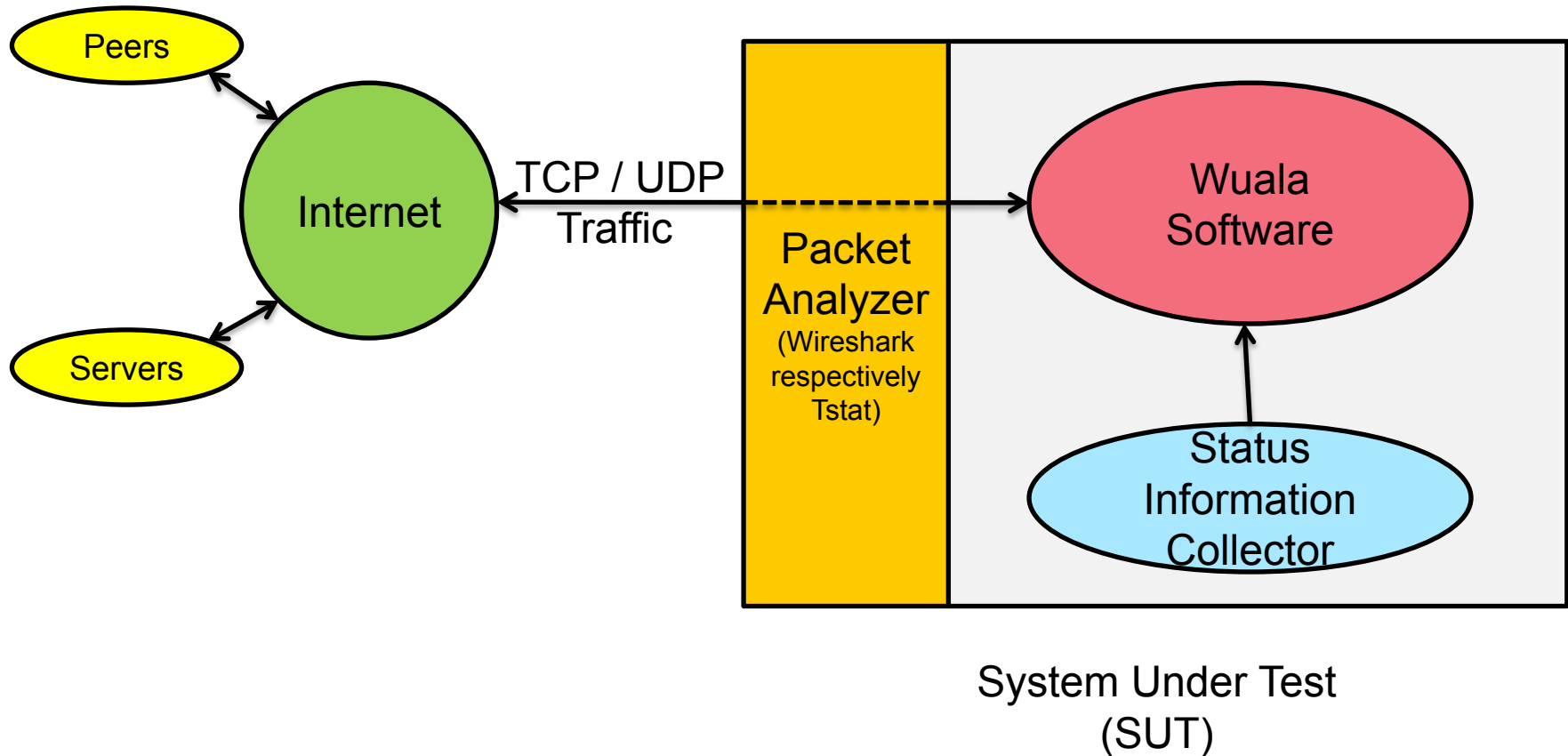
Java bytecode obfuscated:

- No debug information available
- Name mangling was performed

→ Decompiling not helpful

However, observing network traffic should be meaningful!

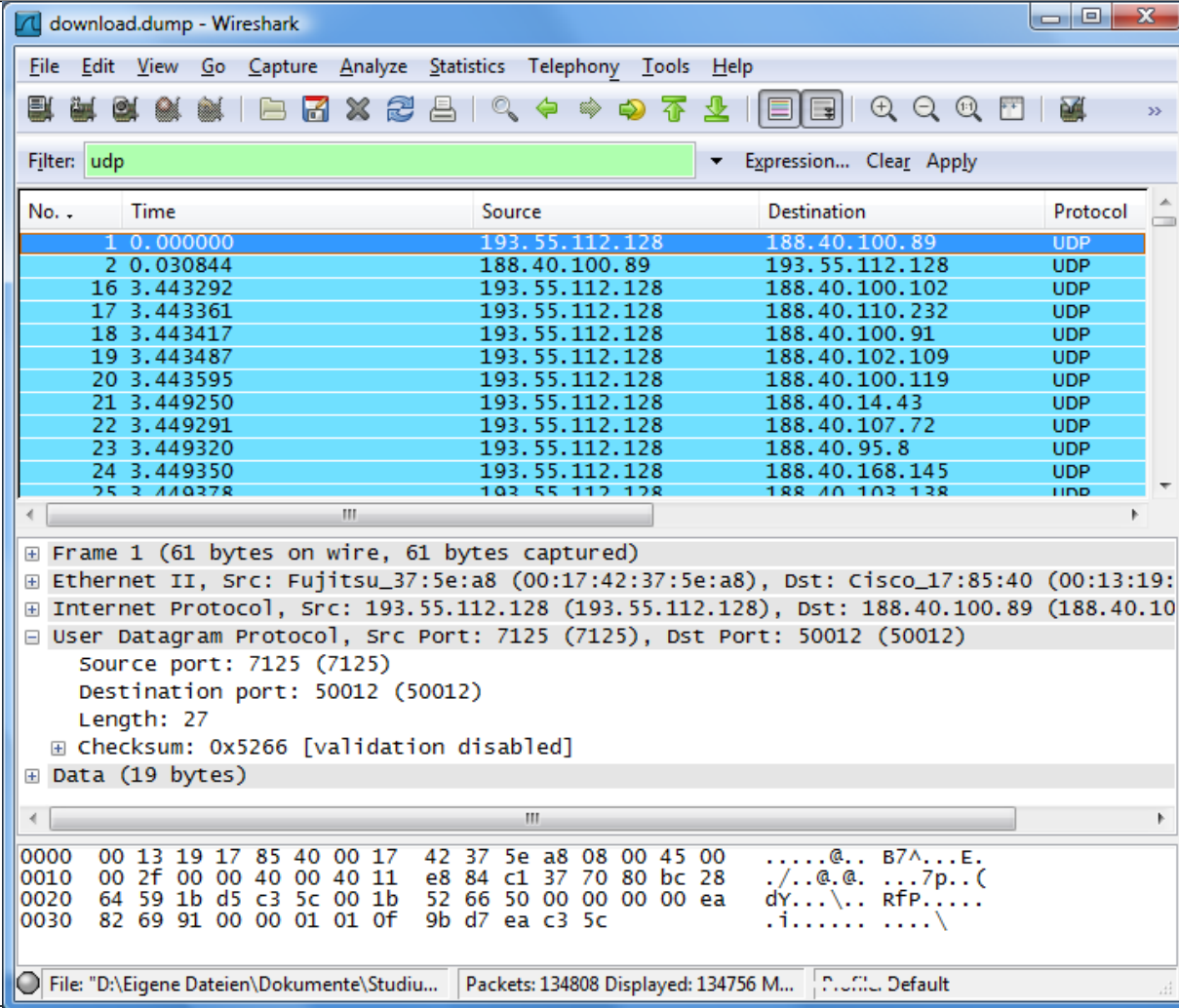
General Experimental Setup



Used Packet Analyzer Depending on Purpose

	Wireshark	Tstat
Level of detail	complete packet dump	only summarized data
Output	binary file	text file
Content of packets	preserved	dropped
Postprocessing	possible	limited
Amount of data	large	small

Wireshark – Example Capture



download.dump - Wireshark

File Edit View Go Capture Analyze Statistics Telephony Tools Help

Filter: **udp** Expression... Clear Apply

No. .	Time	Source	Destination	Protocol
1	0.000000	193.55.112.128	188.40.100.89	UDP
2	0.030844	188.40.100.89	193.55.112.128	UDP
16	3.443292	193.55.112.128	188.40.100.102	UDP
17	3.443361	193.55.112.128	188.40.110.232	UDP
18	3.443417	193.55.112.128	188.40.100.91	UDP
19	3.443487	193.55.112.128	188.40.102.109	UDP
20	3.443595	193.55.112.128	188.40.100.119	UDP
21	3.449250	193.55.112.128	188.40.14.43	UDP
22	3.449291	193.55.112.128	188.40.107.72	UDP
23	3.449320	193.55.112.128	188.40.95.8	UDP
24	3.449350	193.55.112.128	188.40.168.145	UDP
25	3.449378	193.55.112.128	188.40.103.138	UDP

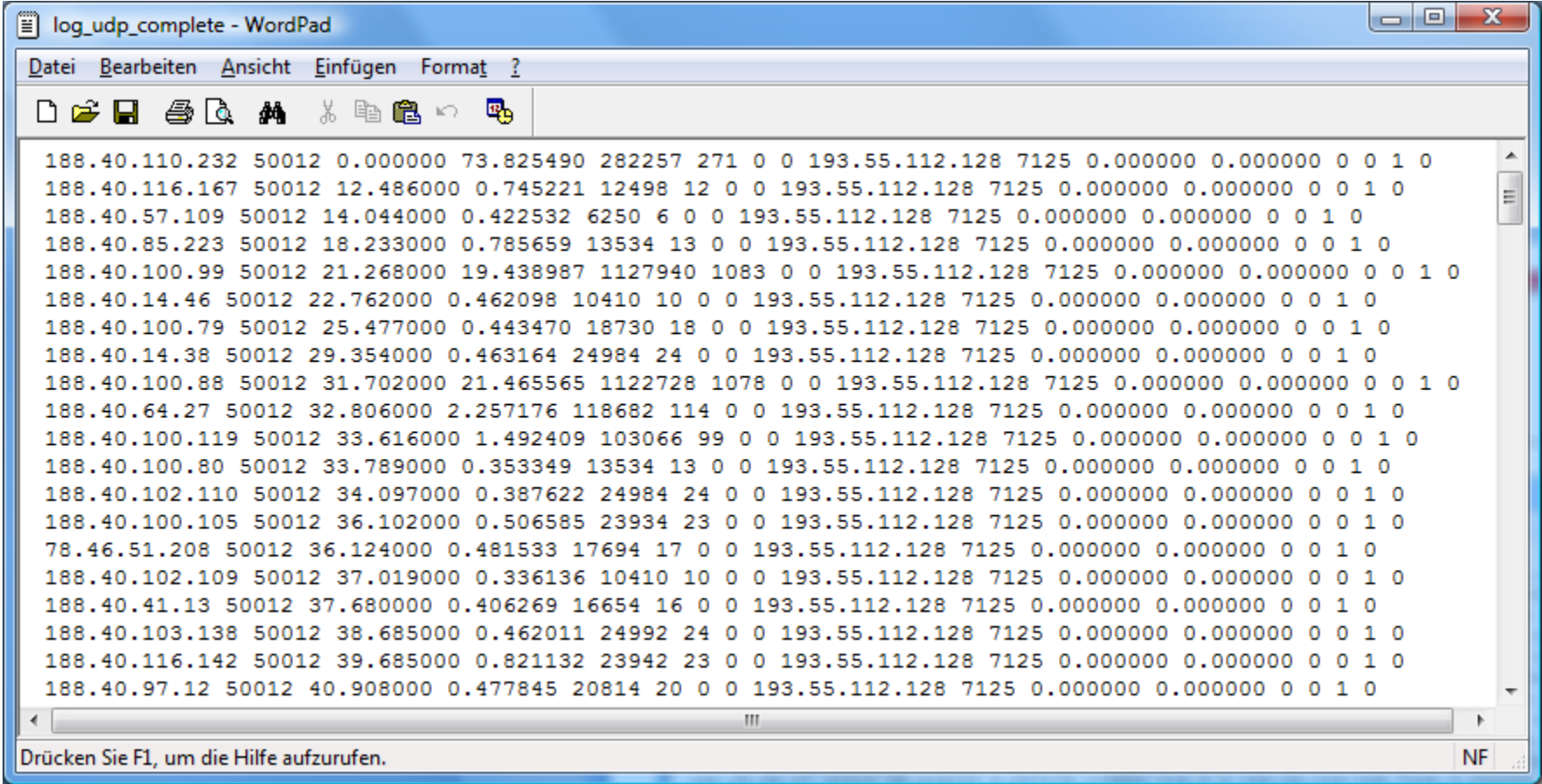
Frame 1 (61 bytes on wire, 61 bytes captured)

- Ethernet II, Src: Fujitsu_37:5e:a8 (00:17:42:37:5e:a8), Dst: Cisco_17:85:40 (00:13:19:00:17:85:40)
- Internet Protocol, Src: 193.55.112.128 (193.55.112.128), Dst: 188.40.100.89 (188.40.100.89)
- User Datagram Protocol, Src Port: 7125 (7125), Dst Port: 50012 (50012)
 - Source port: 7125 (7125)
 - Destination port: 50012 (50012)
 - Length: 27
 - Checksum: 0x5266 [validation disabled]
- Data (19 bytes)

```
0000 00 13 19 17 85 40 00 17 42 37 5e a8 08 00 45 00  ....@.. B7^...E.
0010 00 2f 00 00 40 00 40 11 e8 84 c1 37 70 80 bc 28  ./..@.@. ...7p..(
0020 64 59 1b d5 c3 5c 00 1b 52 66 50 00 00 00 00 ea  dY...\.. RfP.....
0030 82 69 91 00 00 01 01 0f 9b d7 ea c3 5c          .i..... \
```

File: "D:\Eigene Dateien\Dokumente\Studiu... Packets: 134808 Displayed: 134756 M... Profile: Default

Tstat – Example Output



```
log_udp_complete - WordPad
Datei Bearbeiten Ansicht Einfügen Format ?
188.40.110.232 50012 0.000000 73.825490 282257 271 0 0 193.55.112.128 7125 0.000000 0.000000 0 0 1 0
188.40.116.167 50012 12.486000 0.745221 12498 12 0 0 193.55.112.128 7125 0.000000 0.000000 0 0 1 0
188.40.57.109 50012 14.044000 0.422532 6250 6 0 0 193.55.112.128 7125 0.000000 0.000000 0 0 1 0
188.40.85.223 50012 18.233000 0.785659 13534 13 0 0 193.55.112.128 7125 0.000000 0.000000 0 0 1 0
188.40.100.99 50012 21.268000 19.438987 1127940 1083 0 0 193.55.112.128 7125 0.000000 0.000000 0 0 1 0
188.40.14.46 50012 22.762000 0.462098 10410 10 0 0 193.55.112.128 7125 0.000000 0.000000 0 0 1 0
188.40.100.79 50012 25.477000 0.443470 18730 18 0 0 193.55.112.128 7125 0.000000 0.000000 0 0 1 0
188.40.14.38 50012 29.354000 0.463164 24984 24 0 0 193.55.112.128 7125 0.000000 0.000000 0 0 1 0
188.40.100.88 50012 31.702000 21.465565 1122728 1078 0 0 193.55.112.128 7125 0.000000 0.000000 0 0 1 0
188.40.64.27 50012 32.806000 2.257176 118682 114 0 0 193.55.112.128 7125 0.000000 0.000000 0 0 1 0
188.40.100.119 50012 33.616000 1.492409 103066 99 0 0 193.55.112.128 7125 0.000000 0.000000 0 0 1 0
188.40.100.80 50012 33.789000 0.353349 13534 13 0 0 193.55.112.128 7125 0.000000 0.000000 0 0 1 0
188.40.102.110 50012 34.097000 0.387622 24984 24 0 0 193.55.112.128 7125 0.000000 0.000000 0 0 1 0
188.40.100.105 50012 36.102000 0.506585 23934 23 0 0 193.55.112.128 7125 0.000000 0.000000 0 0 1 0
78.46.51.208 50012 36.124000 0.481533 17694 17 0 0 193.55.112.128 7125 0.000000 0.000000 0 0 1 0
188.40.102.109 50012 37.019000 0.336136 10410 10 0 0 193.55.112.128 7125 0.000000 0.000000 0 0 1 0
188.40.41.13 50012 37.680000 0.406269 16654 16 0 0 193.55.112.128 7125 0.000000 0.000000 0 0 1 0
188.40.103.138 50012 38.685000 0.462011 24992 24 0 0 193.55.112.128 7125 0.000000 0.000000 0 0 1 0
188.40.116.142 50012 39.685000 0.821132 23942 23 0 0 193.55.112.128 7125 0.000000 0.000000 0 0 1 0
188.40.97.12 50012 40.908000 0.477845 20814 20 0 0 193.55.112.128 7125 0.000000 0.000000 0 0 1 0
Drücken Sie F1, um die Hilfe aufzurufen. NF
```

Experiment 1: Discover Wuala Servers

How many servers are supporting the network?

- Servers should show a different behaviour
- Connection endpoints (IP addresses and ports) of particular interest
- Presumably servers located in special subnets

Experiment 1: Discover Wuala Servers

Test procedure:

- High level of participation in the network required:
 - 2 machines running as storage node for 2 months
 - 1 machine running mass download of files (20 GiB in total) for 4 days
(location of data collected by crawler)
- All machines connected to the Internet via 100 Mbit/s link
- Tstat used as packet analyzer
- Database for ownership lookups of IP addresses

Experiment 1: Results

Number of IP Adresses	Country	Provider	Subnetwork
293 (!)	Germany	Hetzner Online AG	188.40.0.0/16 178.63.0.0/16 78.46.0.0/15
2	Switzerland	Atrila GmbH	188.92.144.0/21

Easy to differentiate between servers and peers:

Servers:

UDP port 50012

Peers:

random UDP port in range 7100-7600 (until specified in the application options)

Experiment 2: Where Are Files Stored?

Who is responsible to store uploaded data?

Test procedure:

- Upload files with random content and different sizes
- Sharing storage deactivated
- Use Wireshark to acquire information about endpoints, amount and content of transferred data

Experiment 2: Results

	File Size	Protocol	Stored on	Special Characteristic
Tiny File	0 – 4 KiB	TCP	Servers only	Embedded in metadata
Small File	4 KiB – 292 KiB	TCP	Servers only	Transferred to two servers
Medium File	292 KiB – 1 MiB	UDP	Servers only	Erasure coding performed
Big File	1 MiB – 14 GiB	UDP	Servers and peers	Erasure coding performed

Experiment 2: Results

- Wuala does not store data of all files on peers!
- Only data of files bigger than 1 MiB is sent to other peers
- Small files seem not to be worth the effort distributing them to peers

Experiment 3:

How Are Big Files Transferred?

How is fragmentation done?

Test procedure:

- Upload 100 MiB file with random content
- Sharing storage deactivated
- Wireshark used to generate dump file
- Tshark (delivered with Wireshark) to filter out signaling traffic
- Tstat to scan dump file
- Observe temporary folders

Experiment 3: General Result



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Upload performed in following steps:

1. Encrypt file in temporary folder
2. Generation of Transmission Blocks (usage of erasure coding)
3. Upload to servers
4. Upload seems to be finished (!)
5. Subsequent upload to peers in background

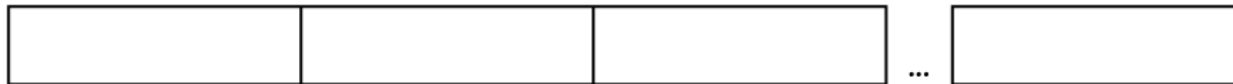


Generation of Transmission Blocks



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Divide file into 100 KiB Coding Blocks



Generation of Transmission Blocks

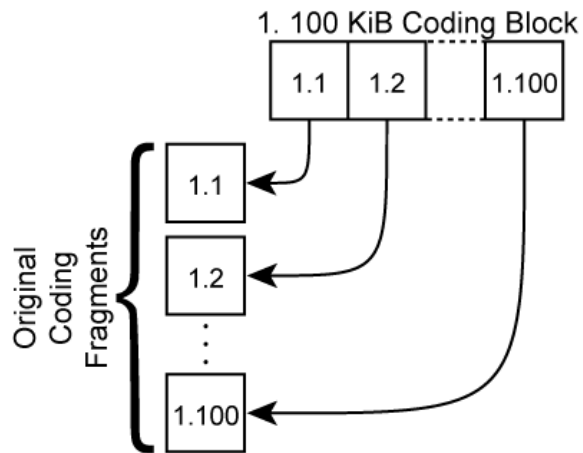


TECHNISCHE
UNIVERSITÄT
DARMSTADT

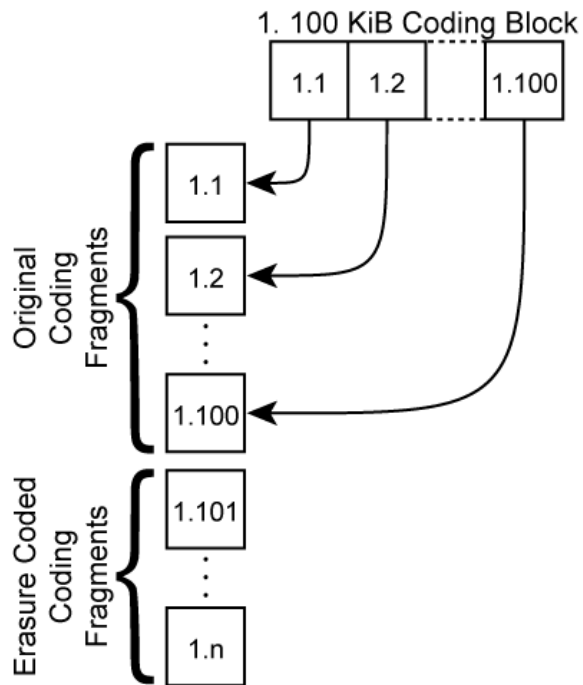
1. 100 KiB Coding Block



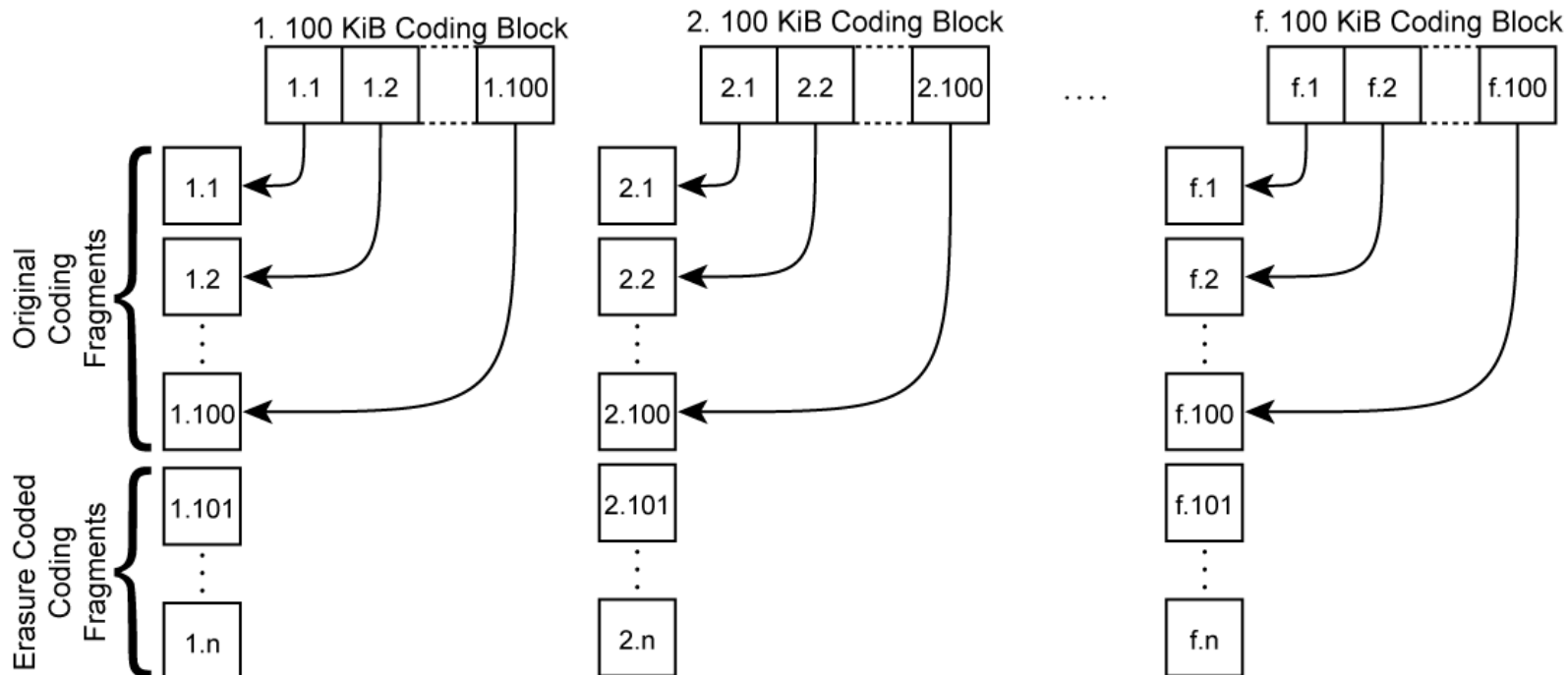
Generation of Transmission Blocks



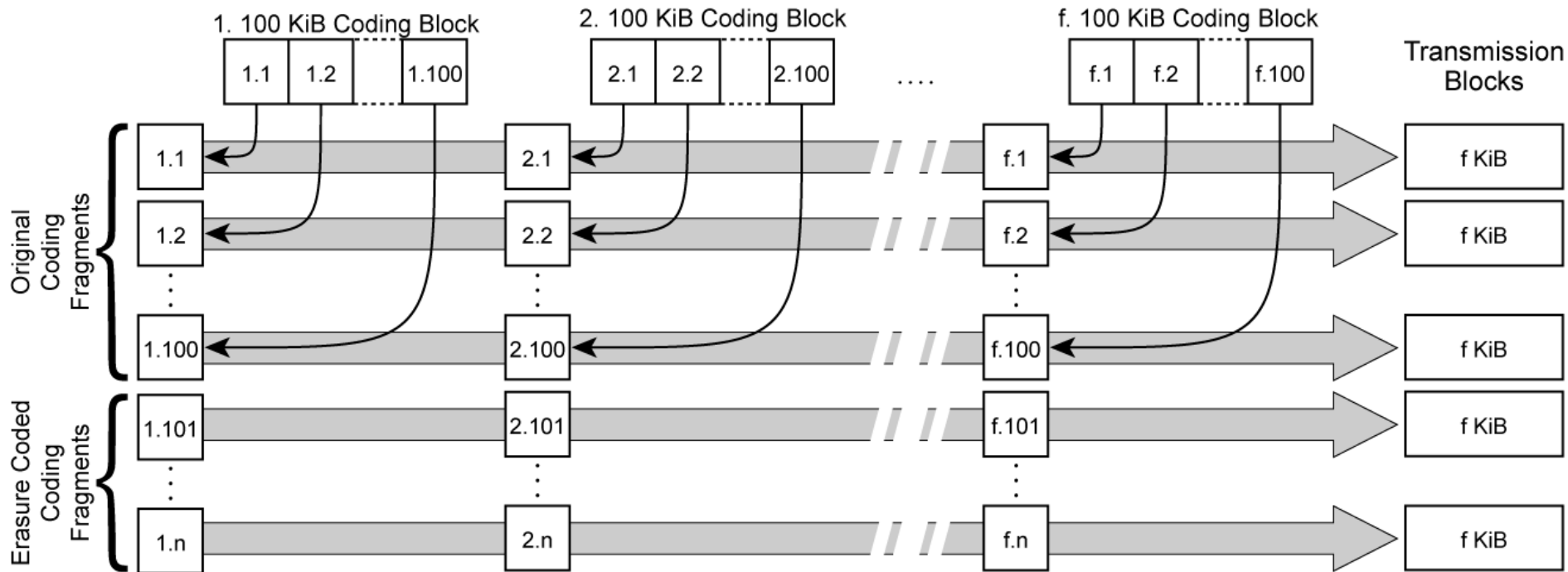
Generation of Transmission Blocks



Generation of Transmission Blocks



Generation of Transmission Blocks



Generation of Transmission Blocks



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Consequences:

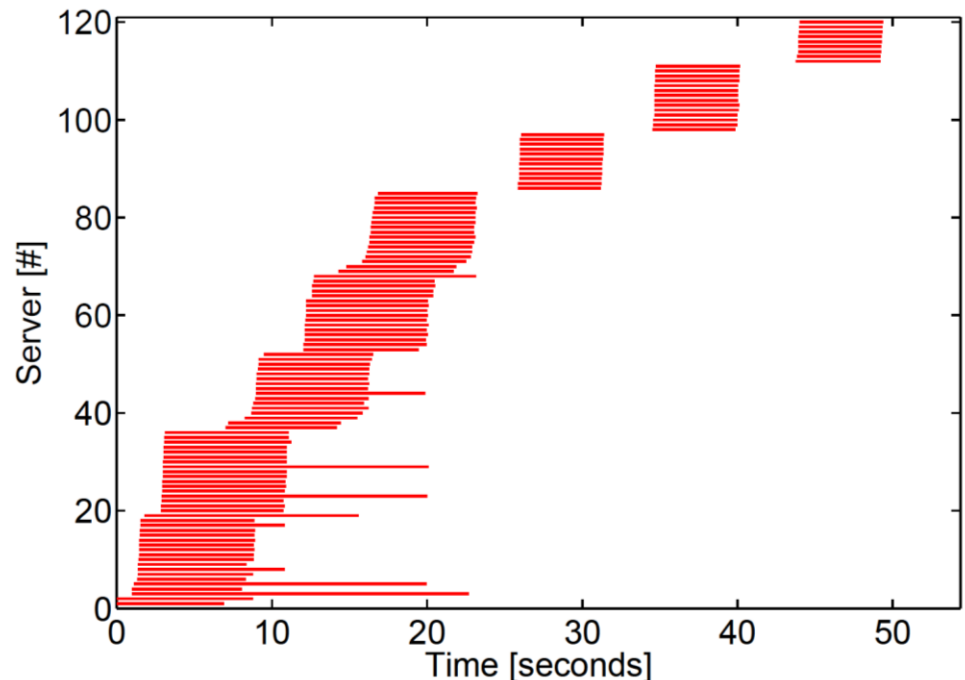
- Interleaving allows streaming of files
- Download from many sources in parallel
- Switching to different sources possible
- Fast downloads



Experiment 3: Results – Server Upload of 100 MiB file

First only upload to servers:

- Upload to 120 distinct servers via UDP
- Most transfer sessions ~1 MiB traffic, some servers receive this amount multiple times
- 144 MiB in total
 - redundancy on servers!
- Upload with full speed
- Concurrent transfers
- Upload completed after 50 seconds

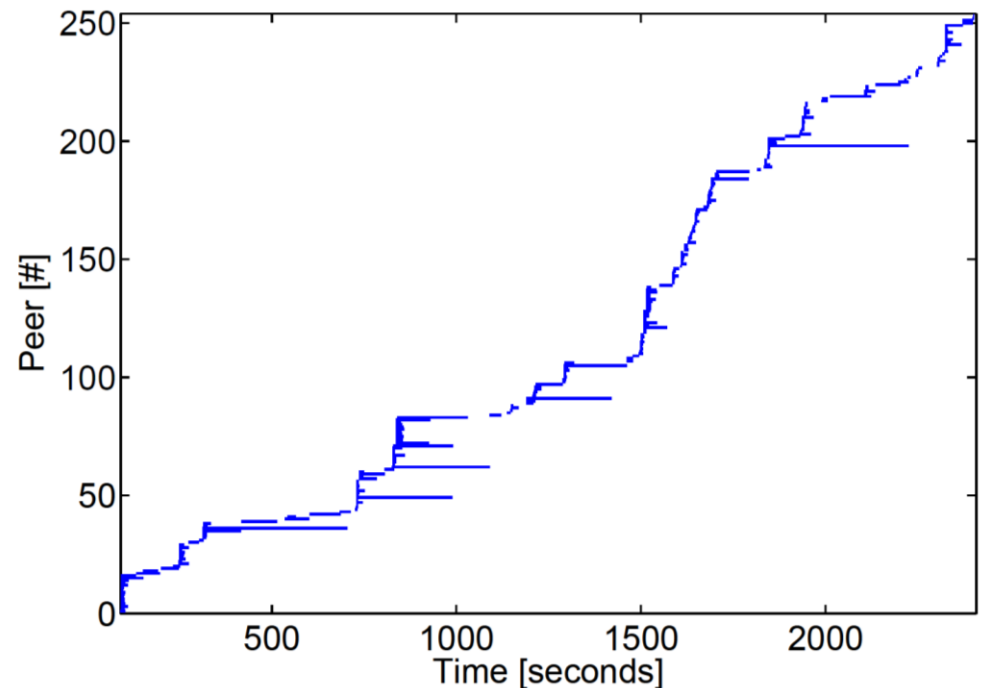


Experiment 3: Results – Peer Upload of 100 MiB file

Subsequent upload to peers:

- Upload to 250 distinct peers via UDP
- Most transfer sessions ~1 MiB traffic
- 273 MiB in total
- Upload with limited speed
- Finished after 40 minutes

Overall ~400 MiB uploaded



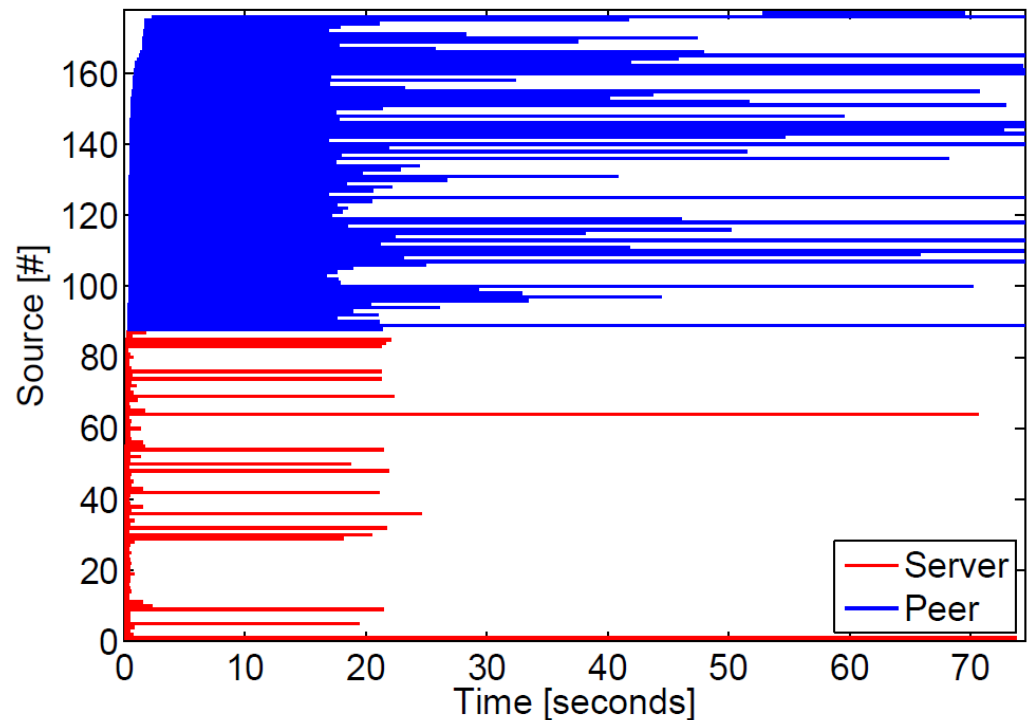
Experiment 4:

Who is serving data during download?

- Download 100 MiB random file again
 - right after upload
 - after 1 week
- Sharing storage deactivated
- Wireshark used to generate dump file
- Tshark (delivered with Wireshark) to filter out signaling traffic
- Tstat to scan dump file

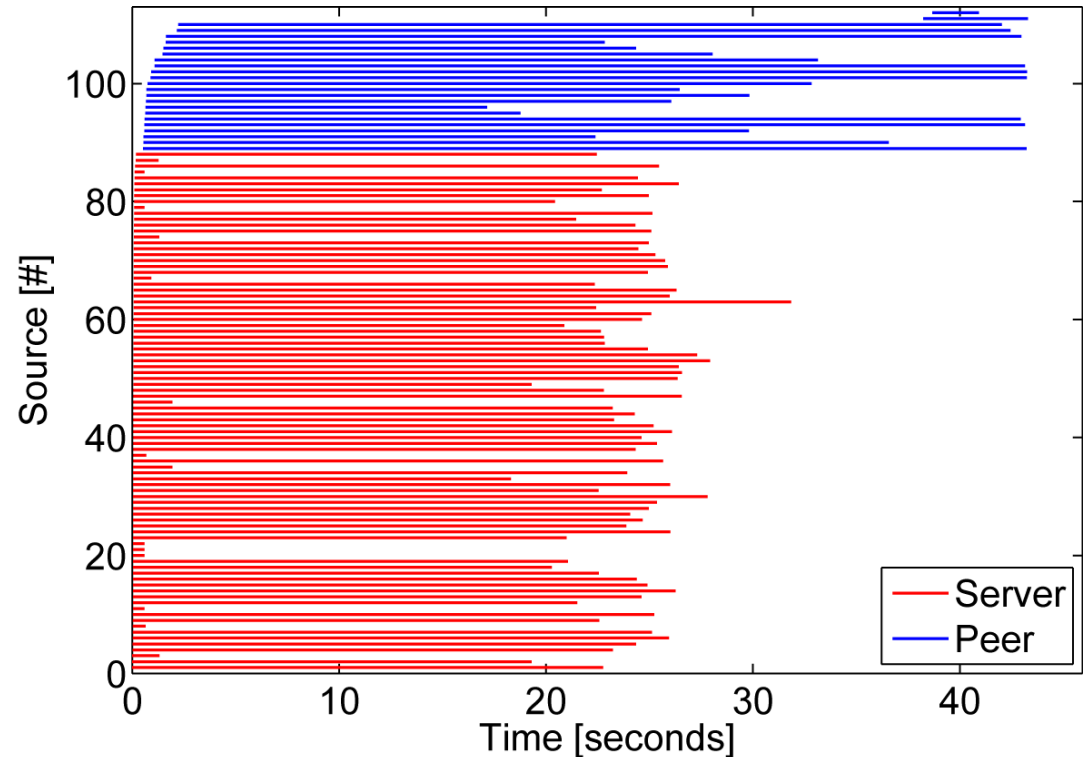
Experiment 4: Results – Right after Upload

- 21 MiB received from servers
- 97 MiB received from peers
- 118 MiB in total
- Download completed after 72 seconds (~1,6 MiB/s)



Experiment 4: Results – 1 Week Later

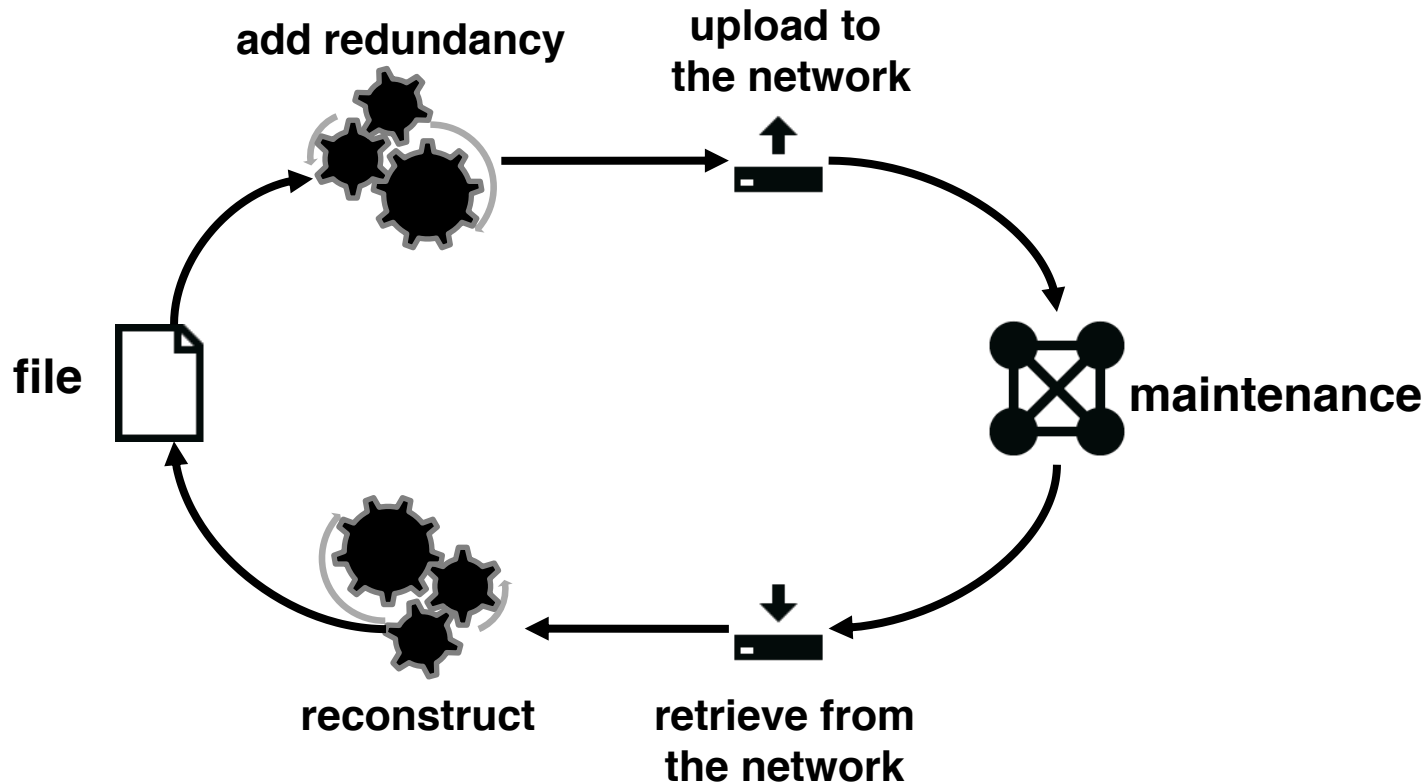
- 100 MiB received from servers
 - 23 MiB received from peers
 - 123 MiB in total
 - Download completed after 43 seconds (~2,8 MiB/s)
- Faster!



Experiment 5: Maintenance

Since peers may leave the network forever:

→ Are files maintained by a peer over time?



Experiment 5: Maintenance



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Test procedure:

- Trading storage deactivated
- Upload 100 MiB file
- File locally deleted
- Use Tstat to observe the peer over 2 months



Experiment 5: Maintenance

Result:

No download within 2 months!

However: Subsequent Upload to Peers **after Download**

- As soon as data is not well distributed on peers
- Similar to initial peer upload
- After one week 132 MiB have been uploaded to peers again

→ Even for big files Wuala is relying on servers only!

Experiment 6:

How Are Sources for a Special File Located?



TECHNISCHE
UNIVERSITÄT
DARMSTADT

- Is there a DHT to locate sources of files?
 - Characteristic of a P2P network
- Or do servers provide this information?
 - Characteristic of a Client / Server Architecture



Experiment 6: How Are Sources for a Special File Located?

- Long term observation revealed no communication to peers except for file transfers
 - no JOIN messages (required by common DHTs!)
- Use Wireshark to generate a dump file before initiating a download
- Put focus on first UDP packets received from servers

Experiment 6: Results

Repetition of special byte sequence obvious in datagrams

→ Presumably IP addresses of servers

First attempt:

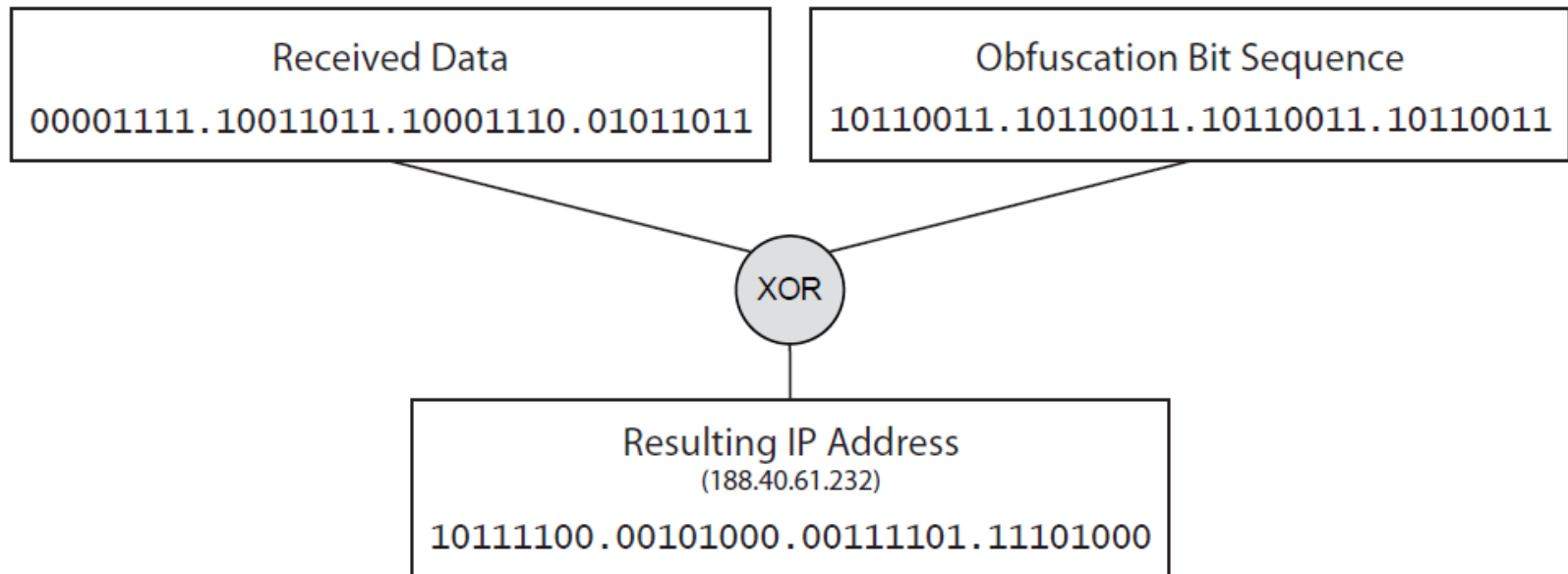
XOR

- would result in constant cipher
- cheap usage

Experiment 6: Results

Repetition of special byte sequence obvious in datagrams

→ Presumably IP addresses of servers



Experiment 7: Earned On-line Storage vs. Occupied Local Storage



TECHNISCHE
UNIVERSITÄT
DARMSTADT

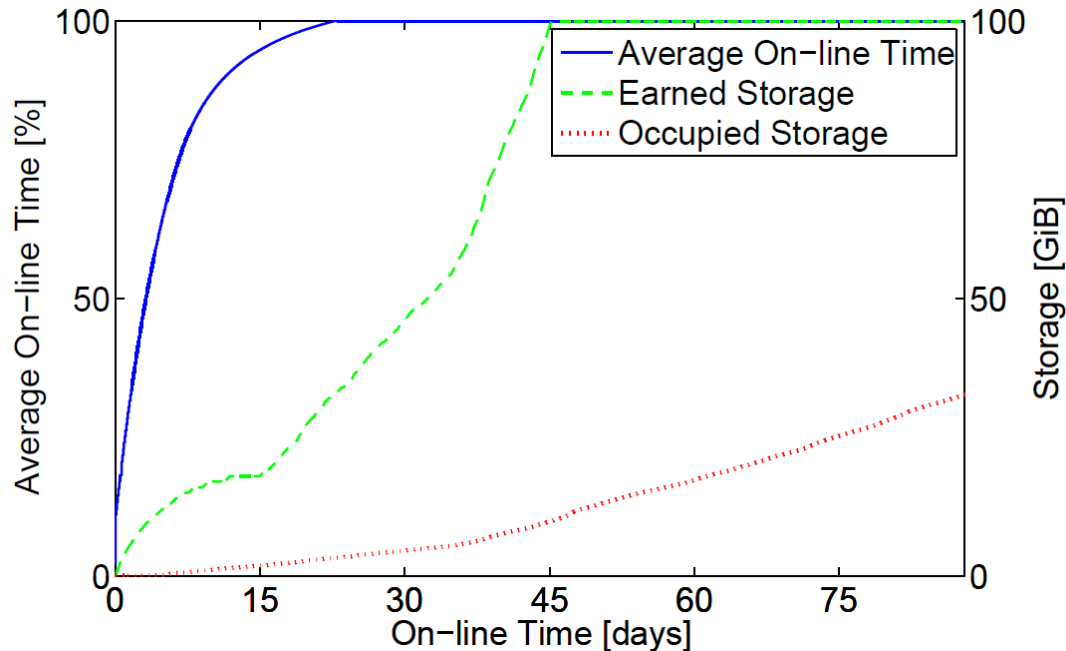
Is there any imbalance, pointing out that users earn more on-line storage than they actually offer?

Procedure:

- Tstat observing a machine operating as storage node
- Duration of 90 days for data acquisition
- Log all status information provided by the Wuala application
- Log amount of used local storage



Experiment 7: Results



First only depending on average on-line time and maximum shared space:

$$\text{earned storage} = \text{on-line time} \times \text{shared space}_{\text{maximum}}$$

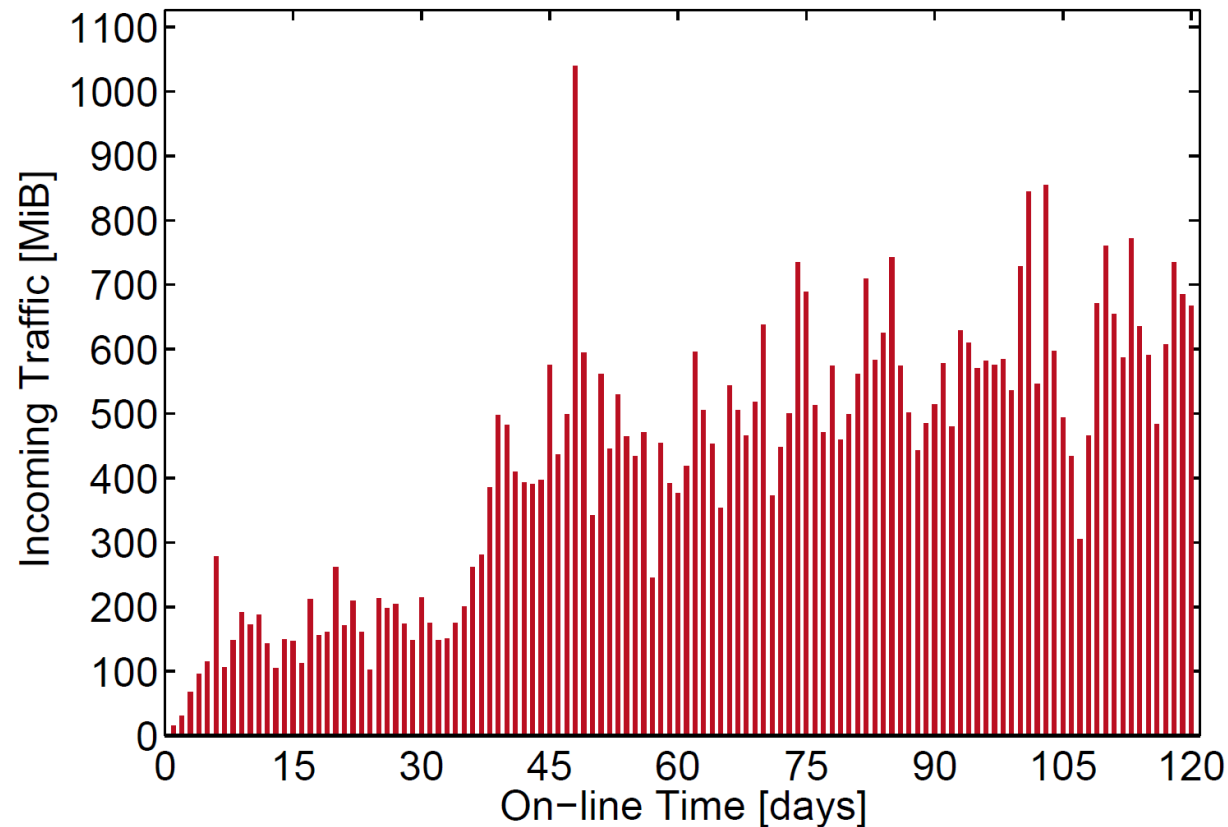
After 2 GiB data has been stored:

$$\text{earned storage} = \min\{\text{on-line time} \times \text{shared space}_{\text{occupied}} \times 10; 100\}$$

Overall a long-lasting gap between earned storage and occupied storage

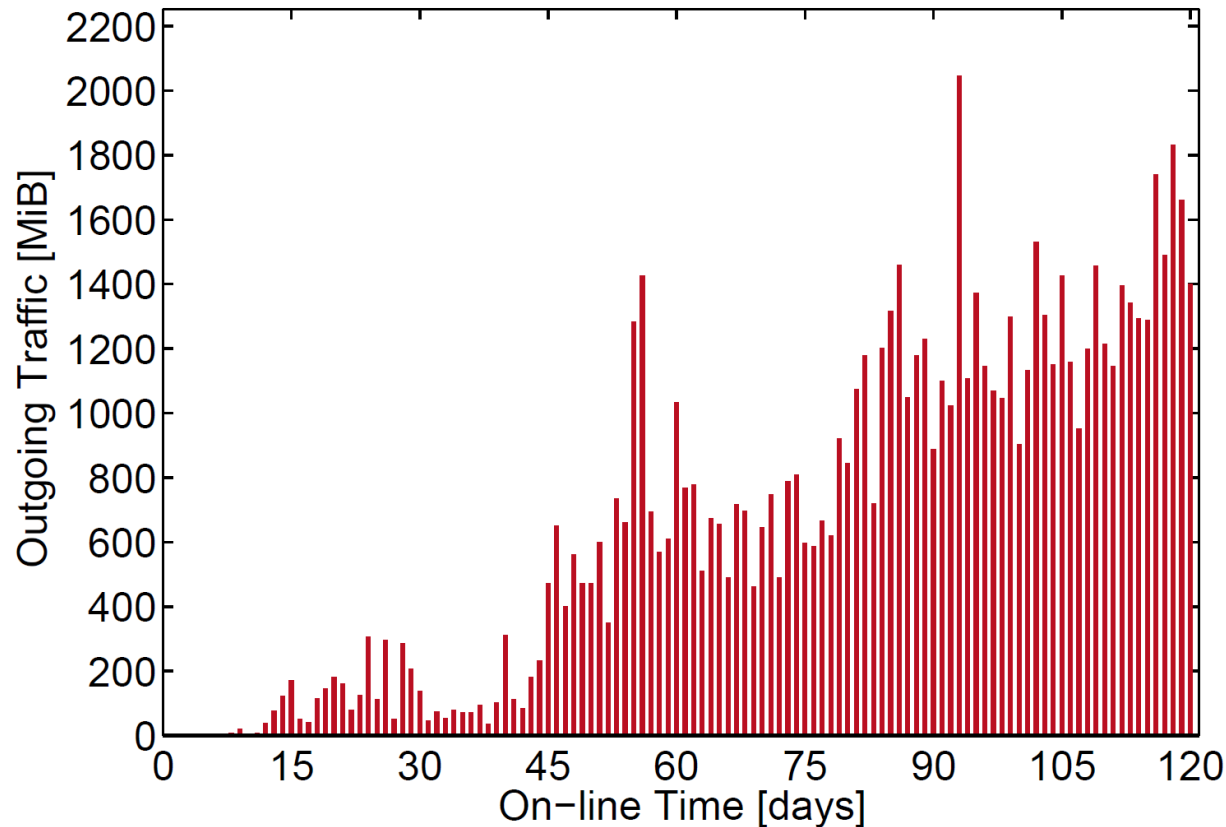
Incoming Traffic over 120 Days

Incoming traffic increases over time - indicates rating of peers



Outgoing Traffic over 120 Days

Correlation: More data stored locally → more requests for stored data
~20 KiB/s after 4 months



Conclusion

- Wuala only **peer assisted**, far away from pure P2P
- **No DHT used on peers**
- Lots of servers responsible to guarantee access on files in long term
- Not all data is uploaded to peers
- Imbalance between offered local storage and on-line storage
- Fast download / streaming possible
- Service dependent on company
- **Peers used to save traffic and load on servers**

Thank you!



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Questions?

