

Lab Class Protocol-Design

P2P-Overlay, Part V

P2P-Protokol, Version 0.2

File Transfer

- Split file into chunks of ≤ 1024 Bytes
- Send Chunks as payload of response messages
- Response codes are
 - 350 for **all but the last** chunk (3xx codes mean 'temporarily OK')
 - 270 for **last** chunk (2xx codes mean 'OK, done') Signals end of transfer!!

P2P-Protokol, Version 0.2

File Transfer

- Extending the P2P Node Software by adding File Transfer:
 - Request file transfer
 - Send file in small chunks
 - Reassemble chunks into file

P2P-Protokol, Version 0.2

File Transfer

- New response message type:

```
P2P/0.2 350 TRANSPORT FOR viper:2000 FROM
boa:2000 MESSAGE-ID 123
KEY file.mp3 TTL 2\r\n
Application: transfer\r\n
Content-Length: 1024\r\n
Offset: 2048\r\n
\r\n
<payload>
```

P2P-Protokol, Version 0.2

File Transfer

- Application-ID:
`Application: transfer`
- Version-0.2 GET Request:

```
GET FROM viper:2000 FOR boa:3000 KEY file.mp3
MESSAGE-ID 12 TTL 2 P2P/0.2\r\n
Application: transfer\r\n
Content-Length: 0\r\n
\r\n
```

P2P-Protokol, Version 0.2

File Transfer

- New Header Field:
 - Offset: from where in the file comes this chunk?
Needed for reassembly of file in the case of reordered packets.

P2P-Protokol, Version 0.2

File Transfer

- File data comes as payload of a sequence of response messages
- Reordering is covered by *Offset Header*
- We assume that no messages are lost!
- **Careful:** 270 response marks last chunk, but there might be 350 response still coming in!

P2P-Protokol, Version 0.2

File Transfer

- Example (file 'file.mp3', size is 2748):
 - SEARCH ... KEY file.mp3 ...
 - ← FOUND ... KEY file.mp3 ...
 - GET ... KEY file.mp3 ...
 - ← 350 TRANSPORT ... KEY file.mp3 ...
Content-Length: 1024
Offset: 0
 - ← 350 TRANSPORT ... KEY file.mp3 ...
Content-Length: 1024
Offset: 1024
 - ← 270 TRANSPORT ... KEY file.mp3 ...
Content-Length: 700
Offset: 2048

P2P-Protokol, Version 0.2

DONE!