TECHNISCHE UNIVERSITÄT BERLIN
Fakultät IV – Elektrotechnik und Informatik
Fachgebiet Intelligente Netze und Management verteilter Systeme
Prof. Anja Feldmann, Ph.D.
Gregor Maier, Jörg Wallerich, Andi Wundsam

10th Work Sheet Praktikum Protokolldesign WS 07/08

Question 1: (100 points) File transfer in the P2P network

Since version 0.2 of our P2P protocol supports messages with payload and can route such messages through the P2P overlay, so now we are going to implement file transfer on this work sheet.

A download is initiated with a `GET`-Message (see sheet 5). Such a `GET`-Request for version 0.2 of the P2P protocol is specified as follows:

```
GET FROM <node id> FOR <node id> KEY <filename> MESSAGE-ID <id> TTL <ttl> P2P/0.2\r\n
Content-Length: 0\r\n
Application: transfer\r\n
\r\n
```

The `FROM`-field specifies the node that wants to download the file. The `FOR`-field specifies a node of which we know that it has a copy of the requested file available (cf. `SEARCH` and `FOUND`).

When the `FOR`-node receives such a `GET`-message, it can immediately start transfering the file. For the transfer the file has to be split in small slices with a maximum size of 1024 bytes per slice. These slices are then sent as the body of several response-message (one slice per message). All these messages are response-messages of type `TRANSPORT`. The reply-code for all messages but the last is 350. The last message of the transfer has a different reply-code that signals the end of the transfer. The reply-code of the last message is 270.

These response messages have the following format:

```
P2P/0.2 350 TRANSPORT FOR <node id> FROM <node id> MESSAGE-ID <id> KEY <filename> TTL <ttl>\r\n
Application: transfer\r\n
Content-Length: <num bytes, max. 1024>\r\n
Offset: <offset in bytes>\r\n
\r\n
<payload>
```

The `FOR` and `FROM`-fields are used as usual and are used for specifying sender and receiver of the message. The `KEY`-field in the first line is always copied from the `GET` message. But every message has a new, unique `MESSAGE-ID`!

The application is specified as `transfer`. The `Offset`-header identifies the offset in the file to which the first byte of this message's payload belongs. This is required if messages get reordered in the network. They are somewhat similiar to TCP sequence numbers. We can assume that messages are never lost, therefore we *do not* need acknowledgements.

The node software should reassemble the incoming messages correctly and store the file on the computer.

Deliverables:

- Your program/script (both source code and binary if written in C, C++ or Java)
- A short description of how to use the software, and what things are not quite working yet.

For details concerning the submission of your solutions see FAQ
(`http://www.net.t-labs.tu-berlin.de/teaching/ws0708/PD_labcourse/`).
Due: Tuesday, January 22., 2008, 11:59h s.t.