ENSEA
Ecole Nationale Supérieure de l'Electronique et de ses Applications

2nd year – Major in Computer Science

Lab Work Synthesis – TFTP Client

C. BARÈS, N. PAPAZOGLOU

Objectives: Develop a TFTP client using RFC specifications and Wireshark captures.

General Advice:

- *Refer to the instructions from the 1st lab; they are identical.*
- *You must call your professor at each checkpoint (indicated by the ⚠ symbol) to validate your progress. If they are not available at that specific moment, continue and call them later.*

Protocol

You will create a client program capable of exchanging files using the Trivial File Transfer Protocol (TFTP).

TFTP is commonly used for network installation of operating systems (PXE protocol), diskless node operations, and firmware updates for network equipment (routers, IoT, etc.).

To achieve this, you will need to rely on:

- RFCs:
  - RFC1350: TFTP v2
  - RFC2347: TFTP Option Extension
  - RFC2348: TFTP Blocksize Option
  - RFC2349: TFTP Timeout & Tsize Options
  - RFC7440: TFTP Windowsize Option
- Wireshark captures:
  - Captures made by you (+++);
  - Captures available on the Wireshark website[1] (+);
  - Captures available on the internet on the CloudShark website (-).

You can test your client using, but not limited to, one of the following options:

- A TFTP server from your distribution (`sudo apt install atftp` on the VM)[2];
- The TFTP server provided on Moodle[3];
- `netcat` and `hexdump` as a "frame analyzer":

```
$ nc -l -u 1069 | hexdump -C
```

- The TFTP server located on the machine `srvtpinfo1.ensea.fr`, port 69.

[1] https://wiki.wireshark.org/SampleCaptures#TFTP

[2] In this case, the server files are located in `/srv/tftp/` and the port is 69.

[3] In this case, the port is 1069, and the address needs to be specified

Tasks to Accomplish

You will create two clients to be used from the command line:

- One for downloading a file from the server:

```
$ gettftp host file
```

- One for uploading a file to the server:

```
$ puttftp host file
```

You will implement the following features in order:

1. Use command-line arguments for the `gettftp` and `puttftp` programs to obtain request information (server and file).
2. Call `getaddrinfo` to obtain the server's address.
3. ⚠Reserve a connection socket to the server.
4. ⚠For `gettftp`:
    a) Build a properly formed Read Request (RRQ) and send it to the server.
    b) Receive a file consisting of a single Data (DAT) packet and its acknowledgment (ACK).
    c) Receive a file consisting of multiple Data (DAT) packets and their respective acknowledgments (ACK).
5. ⚠For `puttftp`:
    a) Build a properly formed Write Request (WRQ) and send it to the server.
    b) Send a file consisting of a single Data (DAT) packet and receive its acknowledgment (ACK).
    c) Send a file consisting of multiple Data (DAT) packets and receive their respective acknowledgments (ACK).
6. Use the `blocksize` option.
7. Find the optimal `blocksize`.
8. Handle packet loss (and thus acknowledgments) and error packets (ERR).

Remarks for different points in the specifications:

1. Retrieve a pointer to an `addrinfo` structure using `getaddrinfo`, using an address provided via the command line (see TDm3).
2. Use this `addrinfo` structure to reserve a socket and open a connection to the server;
3. For `gettftp`:
    a) When constructing the Read Request (RRQ), adhere to the RFC: byte order, presence of the null character `'\0'`, choice of transfer mode, etc.
    b) For your tests, the server has the following files:
        - Files containing only zeros: `zerosXXX`
        - Files containing only ones: `onesXXX`
        - Files that alternate between 8 zeros and 8 ones: `alt256`
        - Small-sized files: `zeros256`, `ones256`, `alt256`;
        - "Special" sized files: `zeros512`, `ones512`;
        - Larger files: `zeros1024`, `ones1024`, `zeros2048`, `ones2048`;

    - A file `ensea.png`: you can verify its correct transfer by checking its PNG signature;
4. For `puttftp`, the same remarks apply, and pay attention to acknowledgments.
5. The `blocksize` option: see RFC2348. What is the maximum file size transferable?
6. Use Wireshark or think…