CS 428/528

Instructor: Yan Wang

Individual Project 2: Build a File Transfer Application based on UDP protocol

Due: May 5th, 2019 23:59:59pm

In this project, you'll need to implement a File Transfer Application based on UDP protocol. Since UDP provides unreliable transport service (i.e., no guarantee of delivery, no guarantee of ordered delivery, no protection from duplication). So, the client-server application must implement a mechanism to provide reliability. Therefore, the goal of this project is to let you know how to form different types of packets for different purposes, and how to ensure reliability control using ARQ.

What programming language to choose?

You must work by yourself on this project using C.

What requirements this project has?

1. The file should be separated into packets of 1500B length (header+data).
2. Design different packets for at least datagram and ACK (more types of packets are allowed).
3. Use UDP protocol to implement the file transfer application.
4. Use an ARQ method (e.g., STOP & Wait) to guarantee the received file.

How should you run your code?

Yourname_server.c *(run on one computer first)*

The **server** program *MAY* take **1 argument**:

```
Yourname_server   port#
```

and will listen to the ***port#*** and **save** the incoming **file** with the same name as the original file.

Example:

```
Yourname_server  8000
```

will receive the file from ***port 8000*** into the current the working directory.

Yourname_client.c *(run on the other computer)*

The **client** program *MAY* take **3 arguments**:

```
Yourname_client   filename  hostname  port#
```

and will **send** the file ***filename*** to the **socket address (*hostname, port#*)**

Example:

```
Yourname_client  myFile  labXX  8000
```

will send the file ***myFile*** in the working directory to ***labXX*** at ***port 8000***.

Some Hints:

1. What should be included in the header of different types of packets?
2. How to transfer the filename?
3. Check " Trivial File Transfer Protocol " for a rough idea (helpful link: https://erg.abdn.ac.uk/users/gorry/eg3561/inet-pages/tftp.html)

What to hand in?

1. Submit your code (e.g., yourname_client.c and yourname_server.c) with in-line documentation file to blackboard (*deduct 10% points per day after the deadline*)
2. Submit a word/PDF with your code to graphically show how you design the packets.
3. Submit a Readme file with your code to describe your code
   - Instructions about how to run your client and server
   - Completion status of the project (e.g., what has been implemented and tested, what has not)
   - Anything else you want the TA to be aware of while grading your code
4. Show the demo of your project to TA (e.g., orally explain how you design the different packets; give some detailed explanation on where and how you write the code to achieve the required methods) *(your code must run correctly using two desktops in lab G7 or Q22. When do the demo, TA and you will open two computers in the lab as server and client respectively. It is your responsibility to make sure that your code compiles and runs correctly on those computers.)*

Grading Criteria

1. In-line documentation - 10 points
2. Readme file – 20 points
3. Correctness - 80 points
   - You will be given a large binary data file (>30MB) to transfer.
   - Same as the Project 1, the client and the server should be on the same network but different hosts.
   - The status of the packet transmission should be shown in the terminal while transferring the file (e.g., packet # sent, packet # received, and ACK etc.)
   - The received file should be exactly the same as the original file (comparison with the original file will be performed).

You can get **extra points (10pt each)** if you implement:

- Go-Back-N or Selective Repeat ARQ instead of Stop & Wait.
- Error detection and correction method.

❖ **NO credit** if your project does not compile

❖ **NO credit** if you copy codes from any resources other than yours *(your project must be your original work. we will use MOSS2 to detect plagiarism in the projects.)*