# American University of Beirut

Department of Electrical and Computer Engineering

EECE 350 Computer Networks
Spring 2017

## PROJECT EVALUATION

The project grade will be based on your report, and on your code correctness and performance.

*READ CAREFULLY*

Your report should be in the format provided in this document; any deviation from this format will result in a grade penalty.

Your report should be in one PDF document. *Word files are not accepted*.

Your code should be in two text files: sender.py and receiver.py. You should also include in your submission a README.txt file to explain how to run your code.

# American University of Beirut

Department of Electrical and Computer Engineering

EECE 350 Computer Networks

PROJECT REPORT

Group Members:


| Name | ID Number | Email |
|---|---|---|
| 1. Yasmine Wehbe | 201601264 | yww00@mail.aub.edu |
| 2. Julien Allam | 201601464 | jaa66@mail.aub.edu |

***Describe Your Approach to Reliable File Transfer Protocol Design***

***The problem given required that we use UDP as a transport protocol, which is unreliable.***

***To make our protocol reliable, we added elements to the header of the packet, used as a check between the sender and the receiver***

***We used Alternating Bit Protocol, which is usually used in Stop-and-Wait. It occupies 1 byte in the header of the packet for convenience, since we considered our packets as strings. This reserved byte that serves as an ACK will either be 0 or 1, alternating. The reason behind our choice is that it is easy to implement, requiring little space. ACK allows us to track ordering. It also prevents duplicates in an easy way since the ACK number can only be 0 or 1.***

***To control retransmissions, we used a timeout value. How soon should the sender conclude that the packet was lost before retransmitting? After multiple tests, we calculated the average RTT time to be around 2sec, hence we chose 4sec for the timeout.***

***We also used the Checksum. Checksum value has 2 bytes reserved in the header. It is the typical checksum function that is used in the UDP pseudo-header. It is calculated both at the sender and the receiver sides. If both checksums are equal then the data has not been corrupted.***

***Challenges Faced – How Did You Solve Them?***

***We faced many challenges while trying to implement the project.***

***The main challenge using Python as a network tool. We particularly had trouble with the syntax and the indentations.***

***Another challenge was using a file as input, particularly how to signal the end of the file. To solve this, we used the function file.read until an empty string is found. We then sent an End of File (EOF) packet to tell the receiver that the file has been fully sent.***

***We had trouble to concatenate the header to the rest of the data. We then decided to convert all the data to string type for an ease of use.***

***Finally, the function getsizeof(data) was leading to a wrong result, until we realized that this function adds 37 bytes to the results***

***List All Features in Your Design***

***-Error detection***

***-Checksum function***

***-ACKs: Alternating Bit Protocol***

***-Convenient timeout values***

***-Timestamp function that outputs the time down to microsecond***

***-Time function that calculates the duration of the connection on both sender and receiver sides***

***-Input arguments from terminal and output corresponding error messages in case of syntax error (such as cannot input a port unless it is between 1024 and 60000)***

***-Corrupted packets are dropped and resent***

***-Our code allows us to input the name of any file (of any extension) and transfer it. The sender sends the receiver the name of the file before starting the data transmission.***

***Provide an Overview of Testing Methods***

***We were required to use Network Emulation (netem) on Linux to test our code.***

***We used the following commands in terminal, each of which tests a certain source of a possible unreliability.***

***1.sudo tc qdisc add dev lo root netem rate 1mbit delay 25ms***

***2.sudo tc qdisc add dev lo root netem rate 1mbit loss 20%***

***3.sudo tc qdisc add dev lo root netem rate 1mbit delay 25ms reorder 10%***

***4.sudo tc qdisc add dev lo root netem rate 1mbit corrupt 20%***

***5.sudo tc qdisc add dev lo root netem rate 1mbit duplicate 20%***

***We cleared after testing each instruction. We as well did other tests such as changing the rate.***

***To clear use the following command:***

***sudo tc qdisc del dev lo root***

***We also tried sending different types of files such as .txt and .pdf, both of which were sent correctly.***

***Show Your Test Results***

***I-)Test with reliable connection:***

****sending file name & extension to receiver***

*<12:58:03.239912> [sending] PKT0 1024 (1024)*
*<12:58:03.240398> [recv data] PKT0 1024 (1024) ACCEPTED*

****receiving file***

*<12:58:03.240872> [recv ack] ACK0 1024*
*<12:58:03.241284> [sending] PKT1 2048 (1024)*
*<12:58:03.241711> [recv data] PKT1 2048 (1024) ACCEPTED*
*<12:58:03.242102> [recv ack] ACK1 2048*
*<12:58:03.242508> [sending] PKT0 3072 (1024)*
*<12:58:03.242925> [recv data] PKT0 3072 (1024) ACCEPTED*
*<12:58:03.243367> [recv ack] ACK0 3072*
*<12:58:03.243779> [sending] PKT1 4096 (1024)*
*<12:58:03.244189> [recv data] PKT1 4096 (1024) ACCEPTED*
*<12:58:03.244579> [recv ack] ACK1 4096*
*<12:58:03.244989> [sending] PKT0 5120 (1024)*
*<12:58:03.245403> [recv data] PKT0 5120 (1024) ACCEPTED*
*<12:58:03.245882> [recv ack] ACK0 5120*
*<12:58:03.246290> [sending] PKT1 6144 (1024)*
*<12:58:03.246712> [recv data] PKT1 6144 (1024) ACCEPTED*
*<12:58:03.247101> [recv ack] ACK1 6144*
*<12:58:03.247517> [sending] PKT0 7168 (1024)*
*<12:58:03.247929> [recv data] PKT0 7168 (1024) ACCEPTED*
*<12:58:03.248322> [recv ack] ACK0 7168*
*<12:58:03.248688> [sending] PKT1 8192 (1024)*
*<12:58:03.249059> [recv data] PKT1 8192 (1024) ACCEPTED*
*<12:58:03.249480> [recv ack] ACK1 8192*

****end of file has been reached, sending EOF packet***

*<12:58:03.249860> [sending] PKT0 9216 (1024)*
*<12:58:03.250377> [recv data] PKT0 9216 (1024) ACCEPTED*

***<12:58:03.259670> [receiver completed]***

****Duration of process from receiver side: 0.0203139781952ms***
***...Receiver exited gracefully***

***<12:58:03.270404> [recv ack] ACK0 9216***
***<12:58:03.270511> [sender completed]***

****Duration of process from sender side: 0.0311231613159ms***
***...Sender exited gracefully***

***II-) Netem tests***

***1) 25ms delay***

****sending file name & extension to receiver***

***<12:58:54.339296> [sending] PKT0 1024 (1024)***

***<12:58:54.375055> [recv data] PKT0 1024 (1024) ACCEPTED***

****receiving file***

***<12:58:54.411225> [recv ack] ACK0 1024***

***<12:58:54.411748> [sending] PKT1 2048 (1024)***

***<12:58:54.445802> [recv data] PKT1 2048 (1024) ACCEPTED***

***<12:58:54.479788> [recv ack] ACK1 2048***

***<12:58:54.480269> [sending] PKT0 3072 (1024)***

***<12:58:54.514302> [recv data] PKT0 3072 (1024) ACCEPTED***

***<12:58:54.548313> [recv ack] ACK0 3072***

***<12:58:54.548795> [sending] PKT1 4096 (1024)***

***<12:58:54.582834> [recv data] PKT1 4096 (1024) ACCEPTED***

***<12:58:54.616858> [recv ack] ACK1 4096***

***<12:58:54.617343> [sending] PKT0 5120 (1024)***

***<12:58:54.651381> [recv data] PKT0 5120 (1024) ACCEPTED***

***<12:58:54.685387> [recv ack] ACK0 5120***

***<12:58:54.685869> [sending] PKT1 6144 (1024)***

***<12:58:54.719904> [recv data] PKT1 6144 (1024) ACCEPTED***

***<12:58:54.753921> [recv ack] ACK1 6144***

***<12:58:54.754402> [sending] PKT0 7168 (1024)***

***<12:58:54.788444> [recv data] PKT0 7168 (1024) ACCEPTED***

***<12:58:54.822481> [recv ack] ACK0 7168***

***<12:58:54.822934> [sending] PKT1 8192 (1024)***

***<12:58:54.856925> [recv data] PKT1 8192 (1024) ACCEPTED***

***<12:58:54.890937> [recv ack] ACK1 8192***

*****end of file has been reached, sending EOF packet***

***<12:58:54.891401> [sending] PKT0 9216 (1024)***

***<12:58:54.925391> [recv data] PKT0 9216 (1024) ACCEPTED***

***<12:58:54.925857> [receiver completed]***

*****Duration of process from receiver side: 0.561451196671ms***

***...Receiver exited gracefully***

***<12:58:54.959414> [recv ack] ACK0 9216***

***<12:58:54.959523> [sender completed]***

*****Duration of process from sender side: 0.620733976364ms***

***...Sender exited gracefully***

***<u>2) 20% packet loss</u>***

*****sending file name & extension to receiver***

***<13:01:54.878890> [sending] PKT0 1024 (1024)***

***<13:01:54.888206> [recv data] PKT0 1024 (1024) ACCEPTED***

***receiving file

<13:01:54.897147> [recv ack] ACK0 1024

<13:01:54.897551> [sending] PKT1 2048 (1024)

<13:01:54.906498> [recv data] PKT1 2048 (1024) ACCEPTED

<13:01:54.915475> [recv ack] ACK1 2048

<13:01:54.915940> [sending] PKT0 3072 (1024)

<13:01:54.924941> [recv data] PKT0 3072 (1024) ACCEPTED

<13:01:54.933876> [recv ack] ACK0 3072

<13:01:54.934280> [sending] PKT1 4096 (1024)

<13:01:54.945420> [recv data] PKT1 4096 (1024) ACCEPTED

<13:01:58.943794> [recv corrupted packet]

<13:01:58.952776> [recv ack] ACK1 4096

<13:01:58.953218> [sending] PKT0 5120 (1024)

<13:01:58.962183> [recv data] PKT0 5120 (1024) ACCEPTED

<13:01:58.971846> [recv ack] ACK0 5120

<13:01:58.972301> [sending] PKT1 6144 (1024)

<13:01:58.981296> [recv data] PKT1 6144 (1024) ACCEPTED

<13:02:06.987798> [recv corrupted packet]

<13:02:06.996763> [recv ack] ACK1 6144

<13:02:06.997245> [sending] PKT0 7168 (1024)

<13:02:07.006204> [recv data] PKT0 7168 (1024) ACCEPTED

<13:02:07.015144> [recv ack] ACK0 7168

<13:02:07.015526> [sending] PKT1 8192 (1024)

<13:02:07.024477> [recv data] PKT1 8192 (1024) ACCEPTED

<13:02:07.033429> [recv ack] ACK1 8192

*******end of file has been reached, sending EOF packet***

***<13:02:07.033841> [sending] PKT0 9216 (1024)***

***<13:02:07.042754> [recv data] PKT0 9216 (1024) ACCEPTED***

***<13:02:07.043160> [receiver completed]***

*******Duration of process from receiver side: 12.164026022ms***

***...Receiver exited gracefully***

***<13:02:07.051715> [recv ack] ACK0 9216***

***<13:02:07.051824> [sender completed]***

*******Duration of process from sender side: 12.1733958721ms***

***...Sender exited gracefully***

***3) <u>25ms delay 10% reorder</u>***

*******sending file name & extension to receiver***

***<13:03:33.827665> [sending] PKT0 1024 (1024)***

***<13:03:33.861828> [recv data] PKT0 1024 (1024) ACCEPTED***

*******receiving file***

***<13:03:33.895870> [recv ack] ACK0 1024***

***<13:03:33.896375> [sending] PKT1 2048 (1024)***

***<13:03:33.930445> [recv data] PKT1 2048 (1024) ACCEPTED***

***<13:03:33.965789> [recv ack] ACK1 2048***

***<13:03:33.966366> [sending] PKT0 3072 (1024)***

***<13:03:34.001067> [recv data] PKT0 3072 (1024) ACCEPTED***

***<13:03:34.035055> [recv ack] ACK0 3072***

***<13:03:34.035553> [sending] PKT1 4096 (1024)***

***<13:03:34.036034> [recv data] PKT1 4096 (1024) ACCEPTED***

*<13:03:34.070008> [recv ack] ACK1 4096*

*<13:03:34.070502> [sending] PKT0 5120 (1024)*

*<13:03:34.104555> [recv data] PKT0 5120 (1024) ACCEPTED*

*<13:03:34.138634> [recv ack] ACK0 5120*

*<13:03:34.139211> [sending] PKT1 6144 (1024)*

*<13:03:34.180501> [recv data] PKT1 6144 (1024) ACCEPTED*

*<13:03:34.214589> [recv ack] ACK1 6144*

*<13:03:34.215066> [sending] PKT0 7168 (1024)*

*<13:03:34.249118> [recv data] PKT0 7168 (1024) ACCEPTED*

*<13:03:34.283572> [recv ack] ACK0 7168*

*<13:03:34.284024> [sending] PKT1 8192 (1024)*

*<13:03:34.318313> [recv data] PKT1 8192 (1024) ACCEPTED*

*<13:03:34.359199> [recv ack] ACK1 8192*

*****end of file has been reached, sending EOF packet*

*<13:03:34.359760> [sending] PKT0 9216 (1024)*

*<13:03:34.393733> [recv data] PKT0 9216 (1024) ACCEPTED*

*<13:03:34.394186> [receiver completed]*

*****Duration of process from receiver side: 0.541340112686ms*

*...Receiver exited gracefully*

*<13:03:34.427737> [recv ack] ACK0 9216*

*<13:03:34.427840> [sender completed]*

*****Duration of process from sender side: 0.600620985031ms*

*...Sender exited gracefully*

***4) <u>20% corrupted packets</u>***

*******sending file name & extension to receiver***

***<13:04:43.193631> [sending] PKT0 1024 (1024)***

***<13:04:43.202823> [recv data] PKT0 1024 (1024) ACCEPTED***

*******receiving file***

***<13:04:43.211809> [recv ack] ACK0 1024***

***<13:04:43.212261> [sending] PKT1 2048 (1024)***

***<13:04:43.221309> [recv data] PKT1 2048 (1024) ACCEPTED***

***<13:04:43.230285> [recv ack] ACK1 2048***

***<13:04:43.230834> [sending] PKT0 3072 (1024)***

***<13:04:43.239831> [recv data] PKT0 3072 (1024) ACCEPTED***

***<13:04:43.248780> [recv ack] ACK0 3072***

***<13:04:43.249179> [sending] PKT1 4096 (1024)***

***<13:04:43.258160> [recv data] PKT1 4096 (1024) ACCEPTED***

***<13:04:43.274532> [recv ack] ACK1 4096***

***<13:04:43.275012> [sending] PKT0 5120 (1024)***

***<13:04:47.284168> [recv corrupted packet]***

***<13:04:47.305244> [recv data] PKT0 5120 (1024) ACCEPTED***

***<13:04:47.319729> [recv ack] ACK0 5120***

***<13:04:47.320272> [sending] PKT1 6144 (1024)***

***<13:04:47.329286> [recv data] PKT1 6144 (1024) ACCEPTED***

***<13:04:47.340355> [recv ack] ACK1 6144***

***<13:04:47.340801> [sending] PKT0 7168 (1024)***

***<13:04:47.349780> [recv data] PKT0 7168 (1024) ACCEPTED***

***<13:04:47.364236> [recv ack] ACK0 7168***

***<13:04:47.364686> [sending] PKT1 8192 (1024)***

***<13:04:51.376131> [recv corrupted packet]***

***<13:04:55.376133> [recv data] PKT1 8192 (1024) ACCEPTED***

***<13:04:55.385084> [recv ack] ACK1 8192***

*****end of file has been reached, sending EOF packet***

***<13:04:55.385478> [sending] PKT0 9216 (1024)***

***<13:04:55.397736> [recv data] PKT0 9216 (1024) ACCEPTED***

***<13:04:55.398179> [receiver completed]***

*****Duration of process from receiver side: 12.2043910027ms***

***...Receiver exited gracefully***

***<13:04:55.415194> [recv ack] ACK0 9216***

***<13:04:55.415304> [sender completed]***

*****Duration of process from sender side: 12.2221119404ms***

***...Sender exited gracefully***

***5) <u>20% duplicate packets</u>***

*****sending file name & extension to receiver***

***<13:05:50.522376> [sending] PKT0 1024 (1024)***

***<13:05:50.531467> [recv data] PKT0 1024 (1024) ACCEPTED***

*****receiving file***

***<13:05:50.539616> [recv corrupted packet]***

***<13:05:50.548161> [recv ack] ACK0 1024***

*<13:05:50.548637> [sending] PKT1 2048 (1024)*

*<13:05:50.574092> [recv data] PKT1 2048 (1024) ACCEPTED*

*<13:05:50.582229> [recv corrupted packet]*

*<13:05:50.590741> [recv corrupted packet]*

*<13:05:50.599310> [recv corrupted packet]*

*<13:05:50.607841> [recv ack] ACK1 2048*

*….*

*….*

*….*

*<13:05:51.989334> [recv corrupted packet]*

*<13:05:51.997862> [recv corrupted packet]*

*<13:05:52.006407> [recv corrupted packet]*

*<13:05:52.015842> [recv corrupted packet]*

*<13:05:52.023490> [recv ack] ACK1 8192*

*****end of file has been reached, sending EOF packet*

*<13:05:52.023931> [sending] PKT0 9216 (1024)*

*<13:05:52.424611> [recv data] PKT0 9216 (1024) ACCEPTED*

*<13:05:52.425057> [receiver completed]*

*****Duration of process from receiver side: 1.90251302719ms*

*...Receiver exited gracefully*