

Project 2: Transmission Control Protocol

Computer Networks (CS-UH 3012) - Summer 2022

1 Code of Conduct

All assignments are graded, meaning we expect you to adhere to the academic integrity standards of NYU Abu Dhabi. To avoid any confusion regarding this, we will briefly state what is and isn't allowed when working on an assignment.

1. Any document and program code that you submit must be fully written by yourself.
2. You can discuss your work with fellow students, as long as these discussions are restricted to general solution techniques. In other words, these discussions should not be about concrete code you are writing, nor about specific results you wish to submit.
3. When discussing an assignment with others, this should never lead to you possessing the complete or partial solution of others, regardless of whether the solution is in paper or digital form, and independent of who made the solution.
4. You are not allowed to possess solutions by someone from a different year or section, by someone from another university, or code from the Internet, etc.
5. There is never a valid reason to share your code with fellow students.
6. There is no valid reason to publish your code online in any form.
7. Every student is responsible for the work they submit. If there is any doubt during the grading about whether a student created the assignment themselves (e.g. if the solution matches that of others), we reserve the option to let the student explain why this is the case. In case doubts remain, or we decide to directly escalate the issue, the suspected violations will be reported to the academic administration according to the policies of NYU Abu Dhabi. More details can be found at:
<https://students.nyuad.nyu.edu/academics/registration/academic-policies/academic-integrity/>

2 Project Objectives

The task of this project is to implement TCP from the ground up and mainly consists of the reliable data transfer part of TCP.

This project is a group project and you must find exactly one partner to work with. Once you have settled on a partner, you need to inform us by sending an email to the TA and the professor, see the due date under submission details and policy. There will be a 10% penalty if the deadline of the group formation is crossed.

Please read the complete project description carefully before you start so that you know exactly what is being provided and what functionality you are expected to add.

3 Task : Simplified TCP Sender/Receiver

The task is to implement a "Reliable Data Transfer" protocol, following the description of section 3.5.4 from the textbook. The idea here is to build a simplified TCP sender and receiver that is capable of handling packet losses and retransmissions.

The following functionalities must be implemented:

- Sending packets to the network based on a fixed sending window size (e.g. WND of 10 packets)
- Sending acknowledgments back from the receiver and handling what to do when receiving ACKs at the sender
- A timeout mechanism to deal with packet loss and retransmission

For this task, there is no need to buffer out-of-order packets at the receiver and they can be simply discarded, i.e. no ACKs are sent for out-of-order packets. The out-of-order buffering will be implemented in task 2. However, lost packets must be retransmitted by the sender. For the timeout mechanism, you can assume a fixed timeout value that is appropriate for the emulated network scenario using MahiMahi (see below).

We have provided you with a simple (stop-and-wait) starter-code that consists of the following:

- rdt receiver: this holds the implementation of a simple reliable data transfer protocol (rdt) receiver
- rdt sender: this holds the implementation of a simple reliable data transfer protocol (rdt) sender
- Setup document for using mahimahi to emulate different network conditions ([link](#))

The simple rdt protocol is implemented on top of the UDP transport protocol. During the lab session, Dena showed you how to use the network emulator MahiMahi to test your sender and receiver functionality in an emulated network environment.

4 Grading

Description	Score (/20)
Extending the sender to send 10 packets	4
Properly sending and handling ACKs	5
Retransmissions of lost packets	5
Properly receiving the exact file on the receiver (no errors)	4
Coding style and usage of meaningful comments	2

5 Submission Details and Policy

Submission Deadlines:

1. Group formation: June 30, 2022 (-10% penalty if not met)
2. Submission due date: July 6, 2022

Submission Format and System: You can directly submit your files as a zip file on Brightspace (<https://brightspace.nyu.edu/>). Due to technical limitations, submissions via email are not accepted.

Late Submissions: Late submissions will be penalized by 10% per 24 hours, with a maximum of 3 days late. In case of a late submission, please upload your zip file to Brightspace and inform the TA and the professor.