CS2505 – Assignment 05
23.03.2020

This assignment is a replacement of the Lab 5 assignment.

1. The grading for the assignment will take into account the **correctness** of your solution, the **approach** taken, and **comments**, which should be clear and concise. We will be checking carefully for plagiarism and penalties will be applied.
2. This assignment will consist of a report including some Python program elements some questions. To maximise your Continuous Assessments marks, please answers all sections.
3. This assignment is graded out of a maximum of 4 marks.

Your solutions for this assignment must be submitted on Canvas within the specified deadline. Please note that no late submission will be accepted by Canvas. You do not need to include your name or UCC ID in the name of the submitted files (Canvas does that automatically). Please upload **one** document in **pdf** format and use filename Assignment5.pdf.

Assignment 5:

This assignment is focused on reliable data transfer.

As discussed in the lectures, there are a number of reliable data transfer protocols, including stop-and-wait, Go-back-N and Selective Repeat. All of them have their advantages and disadvantages, which we discussed in class. For this assignment, we will focus on the selective repeat protocol.

Assignment exercise:

1. draw an Extended Finite State Machine (EFSM) including all events and actions (see lecture notes for examples of what they look like) for both sender and receiver side of the **selective repeat** protocol, assuming unidirectional packet transmissions by the sender and acknowledgements by the receiver. Use events and actions such as `rdt_send(data)`, `udt_send(data)`, `timeout`, `deliver(data)`, `rdt_rcv(data)`, `make_pkt(seq_num, data, checksum)`, etc. as presented in the lectures in conjunction with the different reliable data transfer protocols.
   Note: the stated events and actions may not be enough and you will need to think of what else is needed. For your answer assume that the communication channel can both introduce packet errors as well as packet loss. Also assume that transmission delay can be variable. Finally, the maximum sequence number value possible is `max_seq_num`.

2. translate the two EFSMs into Python code. Assume that the functions you use in events and actions exist, e.g. you will not need to write code for them. You can include the code together with the diagrams for the EFSM in the one document. Note that indentation is part of Python syntax and needs to be considered in the presentation of the code.