

## Instructions for JNW2 Project DLC3 – Reliable DLC

See chapter 5 of *Understanding Internet Protocols*. Project DLC3 is basically the same as given at the end of Chapter 5. However, in JNW2 the working of the ARQ is in `SerialInterface.java` (methods `performGoBackNSendLogic` and `performGoBackNReceiveLogic`) and the code that controls the logic is in `GoBackN.java`.

Your assignment is to complete `GoBackN.java` based on the algorithms in the comments it contains, which are essentially the same as those in *Understanding Internet Protocols*.

Starting with DLC3 you will run JNW2 as a simulator. You do this using the latest JNW2 and replacing `BitStuffing` and `CrcFcs` with your own code. Right-click on the JNW2 project in NetBeans and select “Properties” and then “Run”. Use the Main Class option to select `JNW2.RunSimulation`; also, type under Arguments “ReliableDLC.txt” to load the appropriate network configuration.

Your assignment is to code seven short functions that control the operation of the JNW2 Reliable DLC. To understand what is needed, first read the comments in `GoBackN` to learn how the JNW2 go-back-N algorithm works, then read through carefully the methods `performGoBackNSendLogic` and `performGoBackNReceiveLogic` in `SerialInterface.java`. Then you should be ready to code the seven functions and test their operation as a reliable DLC.

When it is working correctly, the reliable DLC transfers exactly one copy of each sent “email” between host 1.1 (which is router1) and host 2.1 (which is router2), despite errors in data transmission. The message is displayed when received at the application layer and the packet containing it is counted in the statistics at the end of the simulation run.

You can edit the configuration file `ReliableDLC.txt` to make debugging easier. It is a good idea to start with only one email file and zero bit errors. Then work your way up to passing all three ‘email’ files and setting the bit error rate to  $5E-4$ . That is higher than most usable networks and it will give you enough errors to check out the ARQ in the reliable DLC.

If your code is working properly, when you change the `<serialLinkBitErrorRate>` in `ReliableDLC.txt` from 0 to  $5E-4$  you should see that the number of Ticks to simulation completion increases, because the DLC is retransmitting corrupted frames.

Submit your `GoBackN.java` and a copy of the output produced when you use the original `ReliableDLC.txt` configuration. Be sure to include your name in the code comment provided.