This project consists of building an S&W (Stop and Wait) reliable protocol. The S&W is going to be built on top of UDP, and it is supposed to provide a reliable transport service to the SFTP application (developed in proj1, which needs to change to call your new send and receive functions). Messages are sent one at a time, and each message needs to be acknowledged when received, before a new message can be sent. The S&W implements basically the protocol rdt2.2 presented in the text book.

The S&W consists of a client and a server. Communication is unidirectional, i.e., data flows from the client to the server. The server starts first and waits for messages. The client starts the communication. Messages have seq number 0 or 1. Before sending each message, a checksum is calculated and added to the S&W header. After sending each message, the client waits for a corresponding ACK. When it arrives, if it is not the corresponding ACK (or if the checksum does not match), the message is sent again. If it is the corresponding ACK, the client changes state and returns to the application, which can now send one more message. This means that the S&W blocks on writes.

The server, after receiving a message, checks its checksum. If the message is correct and has the right seq number, the server sends an ACK0 or ACK1 message (according to the seq number) to the client, changes state accordingly, and deliver data to the application.

The protocol should deal properly with duplicate data messages and duplicate ACK messages. Follow the FSM in the book!

The S&W message contains the header and the application data. No reordering is necessary, since the S&W is sending the exact message given by the application, one by one.

To verify your protocol, use the result of a random function to decide whether to send the right checksum or just zero. This will fake the error effect.

### 146 Lab Projects - General Rules

- Projects are individual. Discussions are ok, but each student should have his/her own code.
- Each project requires a demo, during which the student should explain how the code works.
  The presentations will be in the lab on the specified date. Demos are part of the grade. You only get full credit if you demo.
- Besides the demo, each student should deliver the code to the TA.
- Keep the solutions simple!