Assignment 2 FAQs

1. **What is the ideal timeout value?**

   You can use 100 milliseconds. But you can try other values as well.
   Please do not use values greater than 200 milliseconds second, otherwise testing your program would take too much time.

2. **How much data should be in each packet?**

   Each packet has to have exactly 500 bytes data unless the number of remaining characters is less than 500 (only occurs for the last packet).
   So each packet (other than last data packet, ACK packets and EOT) has to be exactly 512 bytes (4 bytes for type + 4 bytes for seqnum + 4 bytes for data length + 500 bytes of data).

3. **What is the seqnum of the EOT packet?**

   The seqnum of the EOT packet should be equal to ((1 + <seqnum-of the-last-data-packet>) mod 32).

4. **Do input files have any special character that could cause some encoding/decoding problems?**

   All the input files contain only ASCII characters.

5. **What are the possible values for the emulator max delay?**

   It could get any value **greater** than 0 (Use 1 for simulating no delay).

6. **What are the possible values for the emulator packet discard probability?**

   The discard probability has to be >= 0 and < 1. Probability 0 means that the emulator does not drop any packet at all.

7. **I get the following error. What does it mean?**

   ```
   Can't bind local address in backward receiving.
   : Address already in use
   ```

   One possibility is that you're binding the wrong UDP socket on the wrong port. Make sure you **don't bind a UDP socket to the ports that you're sending to**. Only the ports that you use for receiving UDP packets should be bound in each code.
   Another possibility is that the UDP port number that you use for running the emulator is not open (It is used by some other program).

To solve this problem you can use the following command. It will return 3 random open ports between 1024 and 65535.

```
comm -23 <(seq 1024 65535 | sort) <(ss -tan | awk '{print $4}' | cut -d':' -f2 | grep "[0-9]\{1,5\}" | sort -u) | shuf | head -n 3
```

* You can use `ctrl + c` to terminate the emulator, otherwise, you may not be able to use the same port numbers again.

8. **Do I need to implement the packet class if I want to implement in a language other than Java?**

   Yes, you have to implement the packet class yourself. You must use exactly the same encoding approach (as packet.java) for your packet class since the emulator only works with that encoding. You can check your encoding by running the emulator in the verbose mode. If the seqnum that the emulator prints are similar to your packets seqnums then your encoding is fine, otherwise (the seqnums are some very large or negative values) it means that your encoding is not correct.

   *** array() function from Bytebuffer class in java use big-endian order to convert a Bytebuffer instance to a byte array. So you have to use the same endianness to create a byte array from the UDP data.

9. **Should I use multithreading for implementing the assignment?**

   We strongly suggest you to use multithreading to implement the sender (one thread for sending packets and one for listening for the acks). There are some approaches which use only a single thread. You can use any approach as long as you satisfy all the requirements of A2. Receiver could be implemented using a single thread and there is no need of using multithreading.

Useful information:

1. Sender, Emulator, and Receiver on may run on 3 different machines or a single machine for testing. Make sure your code works in both cases.
2. You don't need to send a Makefile if you are using a scripting language like python, otherwise, you have to have a Makefile. However, you must include shell scripts names "sender.sh" and "receiver.sh" in the root folder of your submission that run your code with given arguments.
3. Don't log EOT packet. Or if you want to log, please log them in both seqnum.log and arrival.log. Never log EOT packet in ack.log.
4. Receiver should create the destination file if it doesn't exist and overwrite it if it exists (Just open the file using 'w' flag).
5. Partial grading is not possible for submissions that don't run. Don't expect to receive marks if your code doesn't work for the simplest configurations. Partial grading is done in grading the log files.
6. If the first received packet by receiver is not #0, Either ACK -1 (i.e. #31) or do not ACK.