Here, FYI, are the most recent Q&A's for this lab assignment:

- *You write a very clean, portable program! I only made a few changes to compile it on Windows NT using Microsoft Developer Studio. I have not started the assignment yet. I thought you might be interested in what changes I made as an initial step.*

```
1. include stdlib.h
2. add function prototypes at the top of the file for init,
   generate_next_arrival, and insertevent.
3. declared functions that don't return anything as void.
4. cast a bunch of constants to float.
5. removed a couple of unreferenced local variables.
6. set mmm to RAND_MAX in function jimsrand.
7. exit takes an argument under windows (i.e. exit(0) instead of exit())
8. removed redefinitions of malloc (it  is already declared in stdlib.h.)
```

  I'm impressed. You must have put a lot of time into this. I included the whole modified file in case you wanted to take a look.

  Well, thanks. Here is the modified code (modification to the code I handed out) that this student turned in. I have not tried that code myself.

- *The gcc compiler complains about the use of exit() in your code.*

  Change exit() to exit(0) if this is a problem.

- *The compiler complains about the use of the time variable in your code.*

  Some compilers will do this. You can change the name of my emulator's *time* variable to *simtime* or something like that.

- *My timer doesn't work. Sometimes it times out immediately after I set it (without waiting), other times, it does not time out at the right time. What's up?*

  My timer code is OK (hundreds of students have used it). The most common timer problem I've seen is that students call my timer routine and pass it an integer time value (wrong), instead of a float (as specified).

- *You say that we can access you time variable for diagnostics, but it seems that accessing it in managaing our timer interrupt list would also be useful. Can we use time for this purpose?*

  Yes.

- *The jimsrand() function is not returning the right values on the edlab machines*

  To make jimsrand() work on the edlab Alphas, the variable mmm should be set to RAND_MAX (it may be necessary to include stdlib.h to get this constant).

- *Why is there a timer on the B side?*

  This is there in case you want to implement bi-directional data transfer, in which case you would want a timer for the B->A data path.

- *How concerned does our code need to be with synchronizing the sequence numbers between A and B sides? Does our B side code assume that Connection Establishment (three-way handshake) has already taken place, establishing the first packet sequence number ? In other words can we just assume that the first packet should always have a certain sequence number? Can we pick that number arbitrarily?*

  You can assume that the three way handshake has already taken place. You can hard-code an initial sequence number into your sender and receiver.

- *You instruct us to use both ACK and NAK messages. How would we signify a NAK event given the pkt struct you defined?*

  A hack would be to use negative numbers for NAKs. I probably should have included an explicit ACK/NAK bit in the packet definition.

- *In a bidirectional implementation, how would the sender indicate that a packet includes only an ACK and not an ACK plus data (when there's no data available on which to piggyback the ACK)?*

  The would be no need to use the sequence number (or you could set it to a default value like -1) if there is no data in the packet.

- *When I submitted my assignment I could not get a proper output because the program core dumped..... I could not figure out why I was getting a segmentation fault so ....*

  Offhand I'm not sure whether this applies to your code, but it seems most of the problems with seg. faults on this lab stemmed from programs that printed out char *'s without ensuring those pointed to null- terminated strings. (For example, the messages -- packet payloads -- supplied by the network emulator were not null-terminated) This is a classic difficulty that trips up many programmers who've

recently moved to C from a more safe langauge.