1.

A deadlock occurs when two processes sharing a singular resource. When the use of this conflicts such that no process can access the resource it becomes a deadlock.  
The deadlock in our networking program occurs when…..

2.

3.  
To solve the problem with bit errors we added a header checksum to the packet type. This checksum is simply an addition of all other header fields. This meant that in the sender and receiver we had to add a check that the packet checksum was indeed the same as the sum of all other header fields for each packet. If not the packet must have had a bit error introduced and would be dropped.

4.  
The select function allows a program to wait on file descriptors or similar (such as sockets in our case) until they become ready. That is, when it becomes possible to perform I/O operations without blocking

5.  
To check that the file was copied exactly we looked at the properties window in the file view for the outfile and made sure it had the exact number of bytes as the infile. To complement this, we also had “START” and “END” in the files respectively and checked to make sure these were copied where they should be.  
Could we make the file more diverse and then have another program to check it??

6.

7.  
Assuming that a packet has probability P in the range (0,1) of being lost. To send N packets there will be N\*P failures needing retransmission, these failures also having a . Along with this there is also a 0.1 chance that any packet not lost has a bit error and must be dropped.