Question 1: The testing that we’re doing in this lab is quite basic and inadequate. For example, when we receive a packet, we merely check that the packet was previously sent and not yet received. What other checks would be useful to assure that the mesh is faithfully transmitting the packets we have given to it? Note that this question is notasking how to give the mesh a more comprehensive set of packets to send, but rather  
how to ensure that the packets we do send are faithfully received.

**For a single packet, we can test whether its value is correct (same as the original value), test whether it is ever taken by the virtual environment and whether it is taken at where it is supposed to be.**

**For a set of packets, we can also test whether the number of packets finally taken by the virtual environment matches the number we sent.**

Question2: Now for the other half – how might you better choose which packets to send, when to send them and when to take them once they’ve arrived at their destination?

**Using code to generate a series of random packet might be a good method, as long as we give the testbench enough time to run, we have a good chance to find some potential errors.**

**To choose the packets, we can try to choose different combinations of the start mesh stops and destination, such as (0,0) 🡪 (0,0), (0,0) 🡪 (0,1) ……, (1,0) 🡪 (0,0)**

**And we can try to send multiple packets at the same time and try to make them travel pass the same mesh stop at the same time.**

**And we can also send several packets that start from different mesh stops but have same destination, such that we can test the priority of the Drv\_FIFO.**