1. What is the need for multiplexing/demultiplexing at the transport layer and how does it help?

- The network layer protocol is concerned with only transfer of data packets from one host to another via the network. It does not know about how many applications are running above and for what purposes. The application layer, on the other hand, deals only with user applications that have specific logic for specific purposes. More than one application is often executed simultaneously, all of which may require the support of the same network layer stack for data transfer. The transport layer in the middle thus acts as the logical multiplexing/demultiplexing agent which multiplexes outgoing packets from several application layer protocols to the single network layer protocol, and demultiplexes incoming packets from the same network layer to the appropriate application layer protocols. The transport layer achieves this with the help of port numbers, where each port number is assigned to a specific application. Since the IP address of a single machine is the same for all packets, it is the tuple (IP address, port number) which is unique for each application that requires the service of the network. The transport layer thus bridges between the network layer below and numerous applications above.

2. Does TCP use the same buffer for all applications or different buffers for different applications?

- For a single stream connection with a socket, TCP maintains separate send and receive buffers. But different applications open up their own TCP connections as per their requirements. Thus at any time, there are multiple TCP streams which may be active throughout the system at the same time. For each such TCP connection, separate send and receive buffers are maintained by TCP.

3. Does fragmentation and reassembly occur at every network layer device or only at the source/destination?

- Fragmentation of IP packets can happen at any device of the network layer, but reassembly only happens at the end destination. This is because it will be costly to perform fragmentation and reassembly even at intermediate routers, and also because fragments of the same flow may actually take different routes to reach the destination as per their routing, thus it is only at the end destination where all the fragments will be received and it will be reassembled.

4. If TCP maintains Maximum Segment Size (MSS), then why is IP fragmentation required?

- First note that the different protocols belonging to the different layers have been built independently of each other. The Maximum Segment Size (MSS) is a feature specific only to TCP and other transport layer protocols may not support the concept of MSS. But IP should be able to handle all transport layer protocols, hence this is one reason as to why IP must support fragmentation (in case, say, some transport layer protocol does not support MSS and sends arbitrarily long segments to the IP layer).

  Next, considering the TCP/IP stack, MSS is often calculated with the underlying physical layer MTU in mind. Now it is not easy to get the least MTU among all the links that the packet will travel from source to destination. Also, the minimum MTU may not remain constant for the entire duration of active TCP connection. Had this been the case, then the transport layer could have always maintained the segment size, such that no IP fragmentation would have been required. But in reality, between the source and destination transport layer stack, getting the exact MTU is difficult, also the MTU can change if the packets start to follow a different route from source to destination based on the links in the new route. Thus the source and destination TCP stack may agree on a default or standard MSS, which may be greater than the MTU of the current path and thus network layer fragmentation would be required

Finally, note that the transport layer stack only exists at the two endpoints of the connection, at the source and destination and not on intermediate routers. Therefore, MSS is just an agreement between the end hosts only regarding the size of the data segment that they wish to exchange in a single segment. They may prefer to keep a bigger segment to reduce segmentation and reassembly overhead at the transport layer instead of keeping multiple smaller segments. Thus in such cases, the segments may become bigger than the MTU, which is when fragmentation by the IP layer would be required.it is not easy to get the complete MTU

5. TCP chooses an arbitrary number between 0 and 2^32 - 1 for the number of the first byte. Does it restrict the size of the file to be transmitted?

- No, the size of the file is not restricted. This is because TCP supports wraparound of the sequence number once all the 2^32 - 1 numbers are exhausted. So after the last possible sequence number has been used, for the next packet in the same stream, the SEQ number will be wrapped around to zero and the stream will continue. However, the relative sequence number will always be maintained in order to avoid confusion when two packets of the same stream get the same sequence number after wraparound. However, the concept of relative sequence number is another topic by itself, which requires separate discussion.