**TEAM1 - Design**

**TCP_CLOSE ==> bind()**

During bind() if the sk->sk_state is TCP_CLOSE, then there are no active sockets and hence continue to bind the socket by calling get_port function on the inet->num.

**TCP_LISTEN ==> connect()**

This should not happen and hence an INVALID condition. If sk->sk_state is TCP_LISTEN then, we must return with -EINVAL or -ECONNREFUSED. If this occurs then we can set the state to TCP_CLOSE and return the error or we can just return the error.

**TCP_ESTABLISHED ==> send/recv()**

If the state is TCP_ESTABLISHED, then we must continue to send/receive and remain in the TCP_ESTABLISHED state.

**TCP_SYN_SENT ==> listen()**

This is an INVALID state. If the state is TCP_SYN_SENT, during listen() we should set the error to -EINVAL and return.

**SYN2 ==> TCP_CLOSE**

If we get a SYN2 during TCP_CLOSE state then remain in the same state. Make sure that the skb is freed. No other action is required.

**DATA ==> TCP_LISTEN**

If we get data during TCP_LISTEN state, then we must neglect/drop the data and remain in the same state as we are waiting for a SYN1 packet. We must make sure that the skb is freed and no other action is required.