PRESENT STATE - _ESTABLISHED
API_CALL - bind()

There is a bind request for a socket on a port which is already in established state. The inet_bind function checks for the state and if it is anything other than TCP_CLOSE it will release socket and return EINVAL which means an invalid argument.

PRESENT STATE - _CLOSE
API_CALL - listen()

This is the case of a passive open and the state will be changed to TCP_LISTEN. The listener will wait for a SYN1 packet to send a SYN2 packet.

PRESENT STATE - _LISTEN
API_CALL - send/receive

When the machine is in the LISTEN state it is waiting for a connection request from the other end (SYN1 packet). But if it receives a data packet we can return an error "ENOTCONN" indicating that the connection is still not set for data transfer.

PRESENT STATE - _SYN_SENT
PACKET_TYPE - SYN1

Two conditions can occur:

a) If one of the end point receives a SYN1 in a _SYN_SENT state it checks if it is the same connection (sport=dport,saddr=daddr) from which it was expecting a SYN2. If it’s the same it means both the end points are initiating a connection. We can send a SYN2 packet and go to the _ESTABLISHED state.

b) If it’s a SYN1 from a new endpoint then we can handle the connection by calling a bind for the new socket on the same port and then send a SYN2.

PRESENT STATE - _ESTABLISHED
PACKET_TYPE - SYN2

If the SYN2 is from the same active connection then we RESET the connection to maintain the synchronization of the endpoints else we can discard the packet.

PRESENT STATE - _CLOSE
PACKET_TYPE - reset

This situation occurs when one device closes or aborts the connection without the other one knowing about it. This means one of the endpoints was in the ESTABLISHED state while the other may be in the CLOSED state (no connection). In this case we can send a RESET function to maintain synchronization of both ends.