1.

Once the application has called listen(), the TCP stack will perform the 3-way handshake for any incoming connections. These connections are queued in the kernel, and accept() then retrieves the next connection from the queue and returns it.  
  
There's a backlog argument to listen, and it specifies how large this queue should be (although I think some implementations ignore this, and use a limit built into the stack). When the queue is full, the stack will no longer perform the handshake for incoming connections; the clients should retry, and their connections will succeed when the queue has room for them.  
  
It's done this way so that the client receives the SYN/ACK as quickly as possible in the normal case (when the backlog queue has room), so it doesn't have to retransmit the SYN.

2. The handshake has already happened. The accept() method just delivers you a socket from a queue of already accepted connections. While the queue is empty, it blocks.