

The *listen* Function

- ◆ The *listen* function performs two actions:
 - It converts an unconnected socket into a *passive socket*. This informs the kernel to accept incoming connection requests to this socket
 - It limits the number of connections that will be queued for this server
- int listen(int sockfd, int backlog);
- Returns: 0 if OK, -1 on error

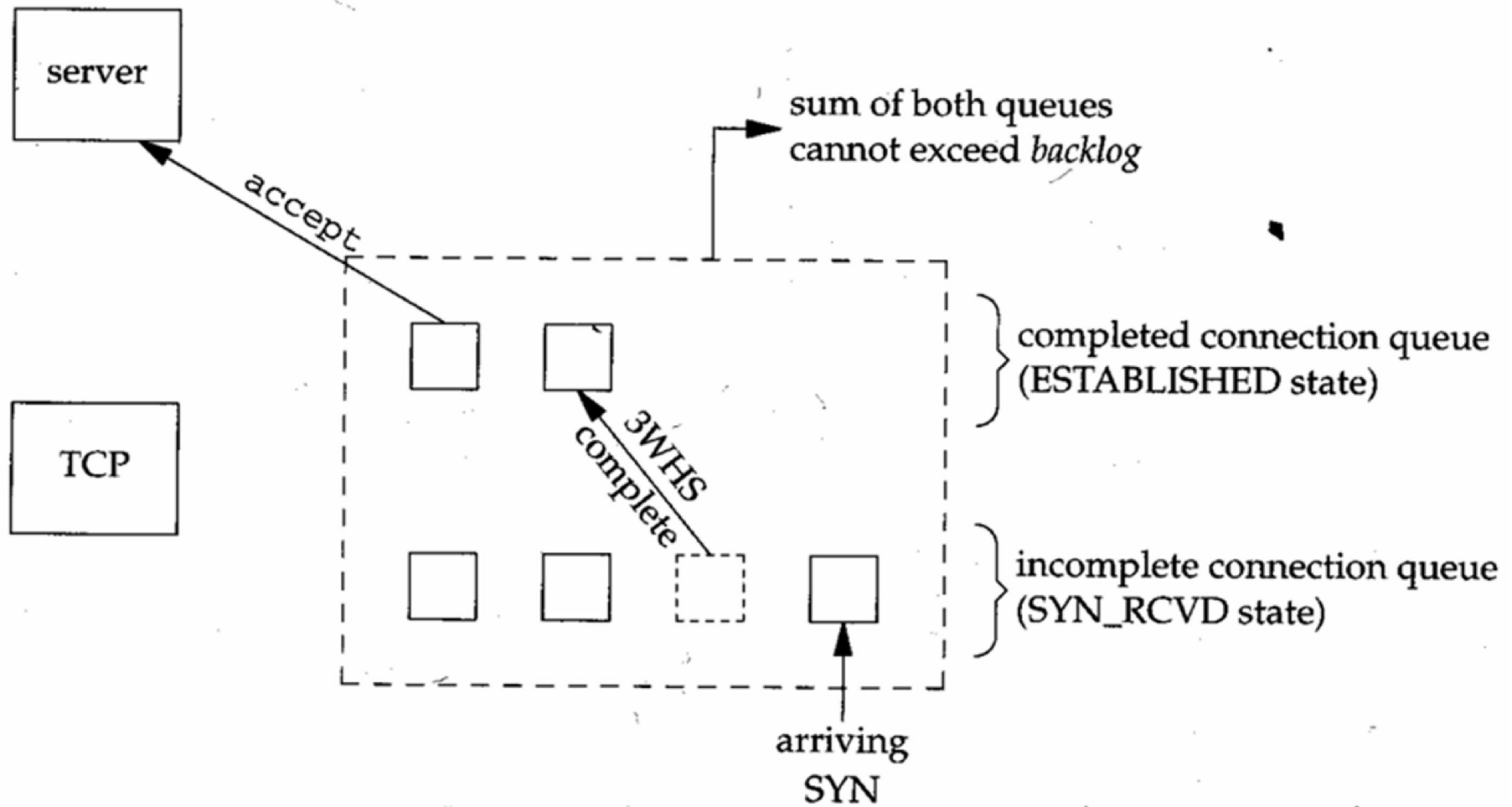
The *listen* Function

- ◆ For a *listening* socket, the kernel maintains two queues:
 - An *incomplete* connection queue contains an entry for each SYN segment that has arrived from a client but is awaiting completion of the TCP 3WHS. These sockets are in the SYN_RCVD state.
 - Entries remain on this queue until the third segment of the 3WHS arrives or until the entry times out.
 - A *completed* connection queue contains an entry for each client for which the TCP 3WHS has been completed. These sockets are in the ESTABLISHED state.
 - Entries on this queue are returned to the process when accept is called

The *listen* Function

- ◆ TCP ignores the arriving SYN when the queues are full
 - It does not send an *RST* (reset) preventing the client's *connect* returning an error
 - Instead the client TCP entity is allowed *retransmit* its SYN segment
- ◆ Data that arrives after the 3WHS completes, but before the server calls *accept*, is queued by the TCP entity up to the size of the *connected* socket's receive buffer

TCP Queues



The *backlog* argument

- ◆ The *backlog* argument specifies the maximum value for the sum of both queues
 - Do not specify a backlog of 0, as different implementations interpret this differently
 - If you do not want any clients connecting to your *listening* socket then close it
- ◆ A backlog of 5 was often used in the 1980s when busy servers handled only a few hundred client connections per day
 - With the growth of the World Wide Web (WWW) HTTP servers can handle millions of connections per day, this small number is inadequate
 - Hence we specify an environment variable LISTENQ which can be changed without a recompile of the code.