```
#include <sys/socket.h>
int listen(int sockfd, int backlog);  /* backlog指定了该套接口排队的最大连接个数 */
```

调用listen导致套接口从CLOSED状态转换到LISTEN状态。

监听窗口维持两个队列(队列的大小与backlog有关)：

1. 未完成队列，每个这样的SYN分节对应一项；已由某个客户发出并到达服务器，而服务器正在等待完成相应的TCP三次握手，此套接口处于SYN_RCVD状态。
2. 完成队列，完成TCP三次握手过程的每一项；该套接口处于ESTABLISHED状态。

队列已满的情况，如何处理?

当一个客户SYN到达时，若这个队列是满的，TCP就忽略该分节，也就是不会发送RST。

这么做的原因在于，队列已满的情况是暂时的，客户TCP如果没收收到RST，就会重发SYN，在队列有空闲的时候处理该请求。如果服务器TCP立即响应一个RST，客户的connect调用就会立即返回一个错误，强制应用进程处理这种情况，而不会再次重发SYN。而且客户端也不无区别该套接口的状态，是“队列已满”还是“该端口没有在监听”。

SYN泛滥攻击

向某一目标服务器发送大量的SYN，用以填满一个或多个TCP端口的未完成队列。每个SYN的源IP地址都置成随机数（IP欺骗），这样防止攻击服务器获悉黑客的真实IP地址。通过伪造的SYN装满未完成连接队列，使得合法的SYN不能排上队，导致针对合法用户的服务被拒绝。

防御方法：

1. 针对服务器主机的方法。增加连接缓冲队列长度和缩短连接请求占用缓冲队列的超时时间。该方式最简单，被很多操作系统采用，但防御性能也最弱。

2. 针对路由器过滤的方法。由于DDoS攻击，包括SYN-Flood，都使用地址伪装技术，所以在路由器上使用规则过滤掉被认为地址伪装的包，会有效的遏制攻击流量。

3. 针对防火墙的方法。在SYN请求连接到真正的服务器之前，使用基于防火墙的网关来测试其合法性。它是一种被普遍采用的专门针对SYN-Flood攻击的防御机制。

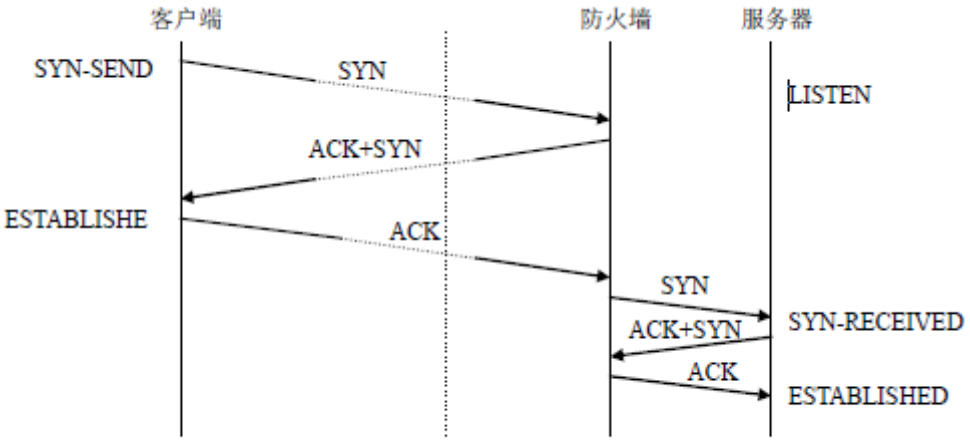

图1　基于防火墙的SYN-Flood防御机制