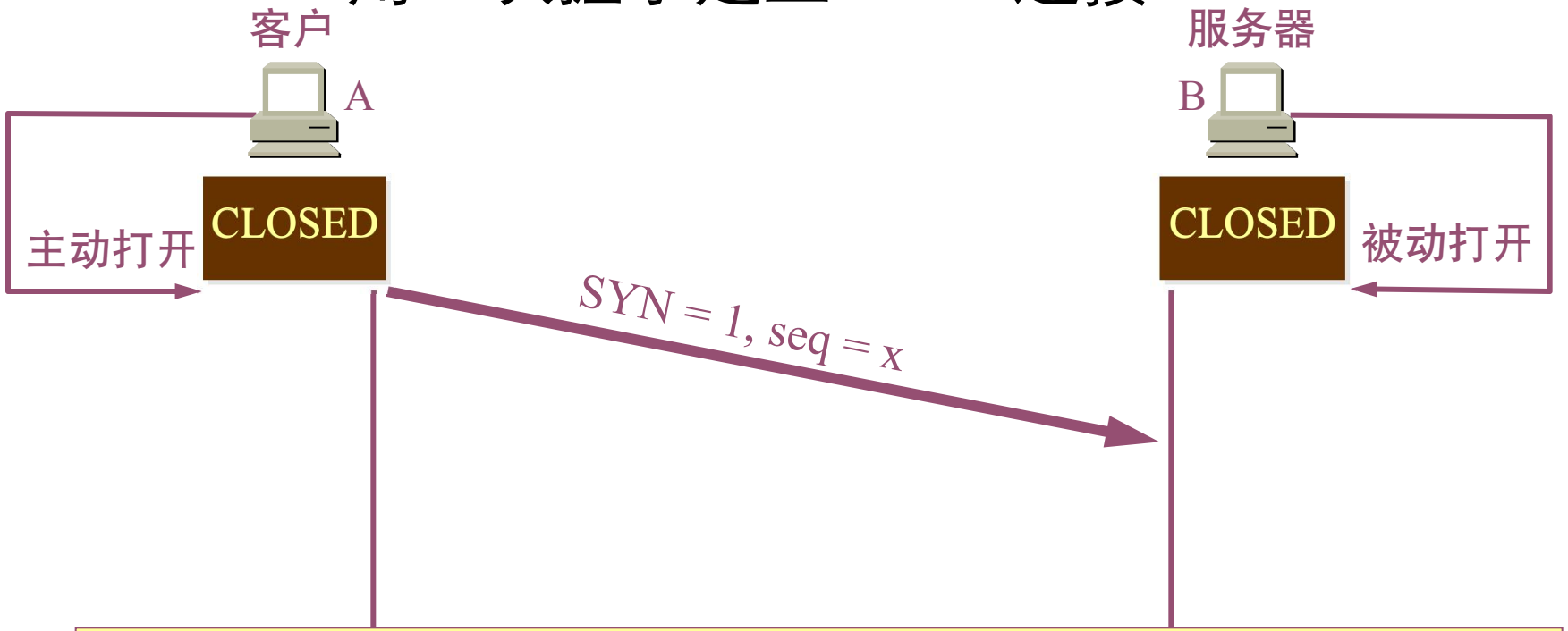
# 5.9.1　TCP 的连接建立

- TCP 建立连接的过程叫做握手。
- 握手需要在客户和服务器之间交换三个 TCP 报文段。称之为三报文握手。
- 采用三报文握手主要是为了防止已失效的连接请求报文段突然又传送到了，因而产生错误。

# 5.9.1 TCP 的连接建立

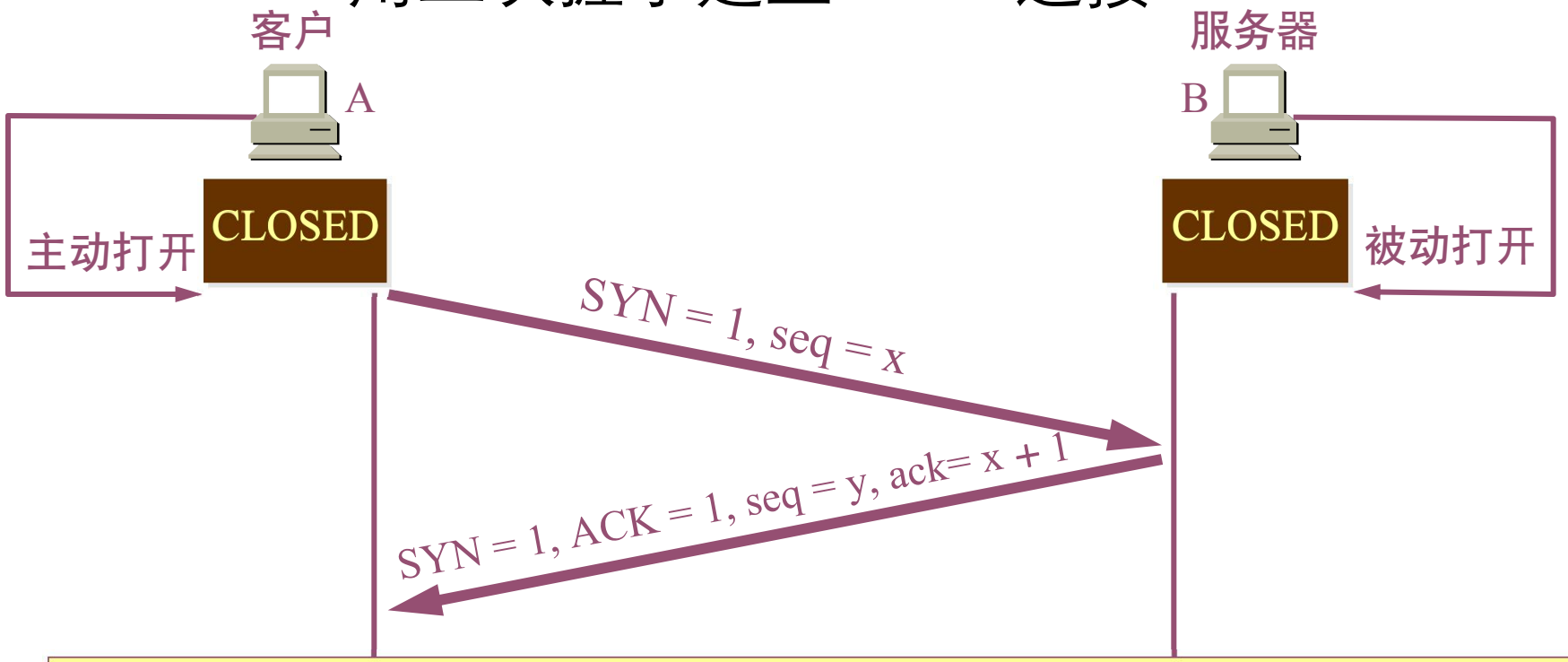
用三次握手建立 TCP 连接




A 的 TCP 向 B 发出连接请求报文段，其首部中的同步位 SYN = 1，并选择序号 seq = x，表明传送数据时的第一个数据字节的序号是 x。

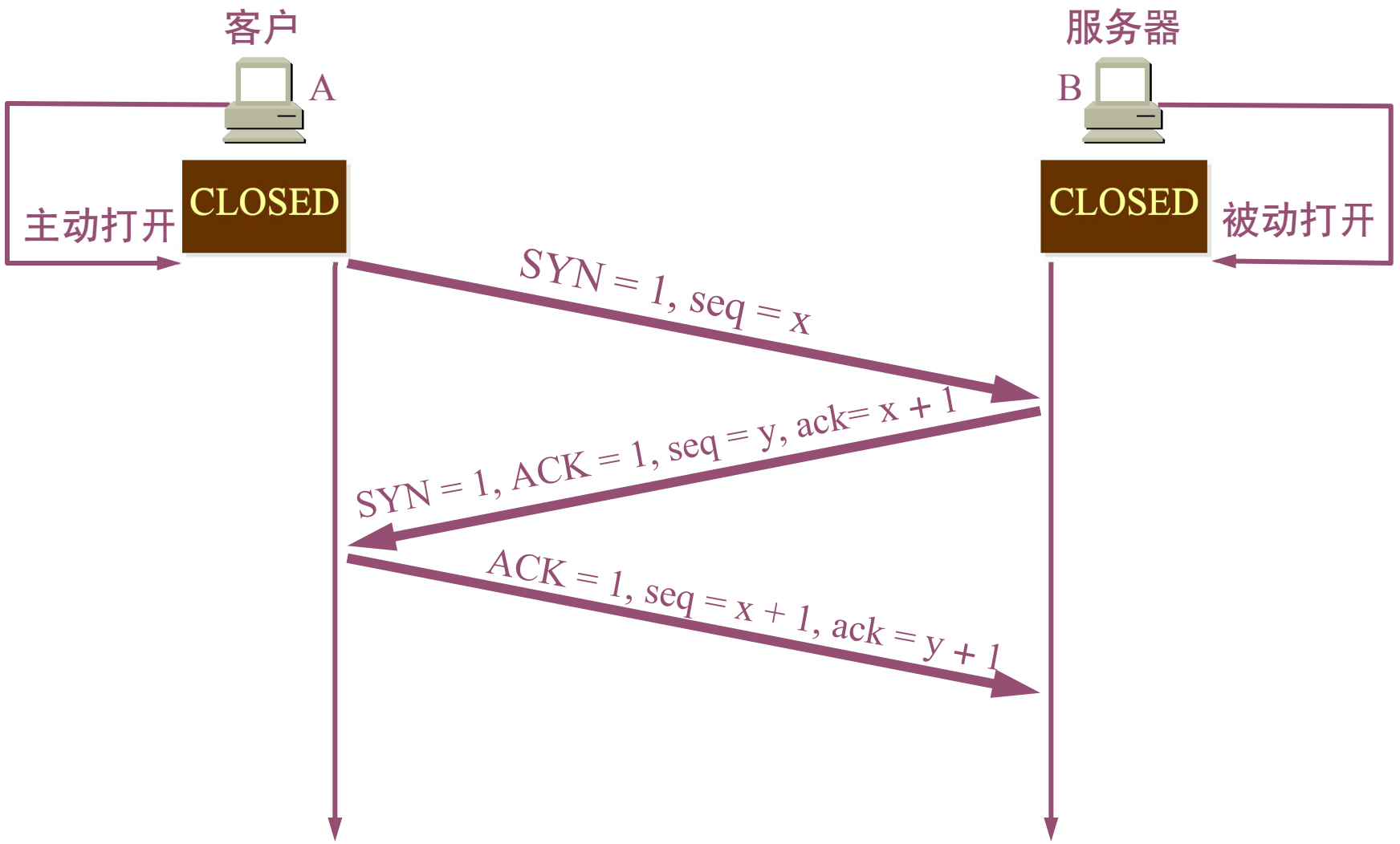
# 5.9.1 TCP 的连接建立

用三次握手建立 TCP 连接

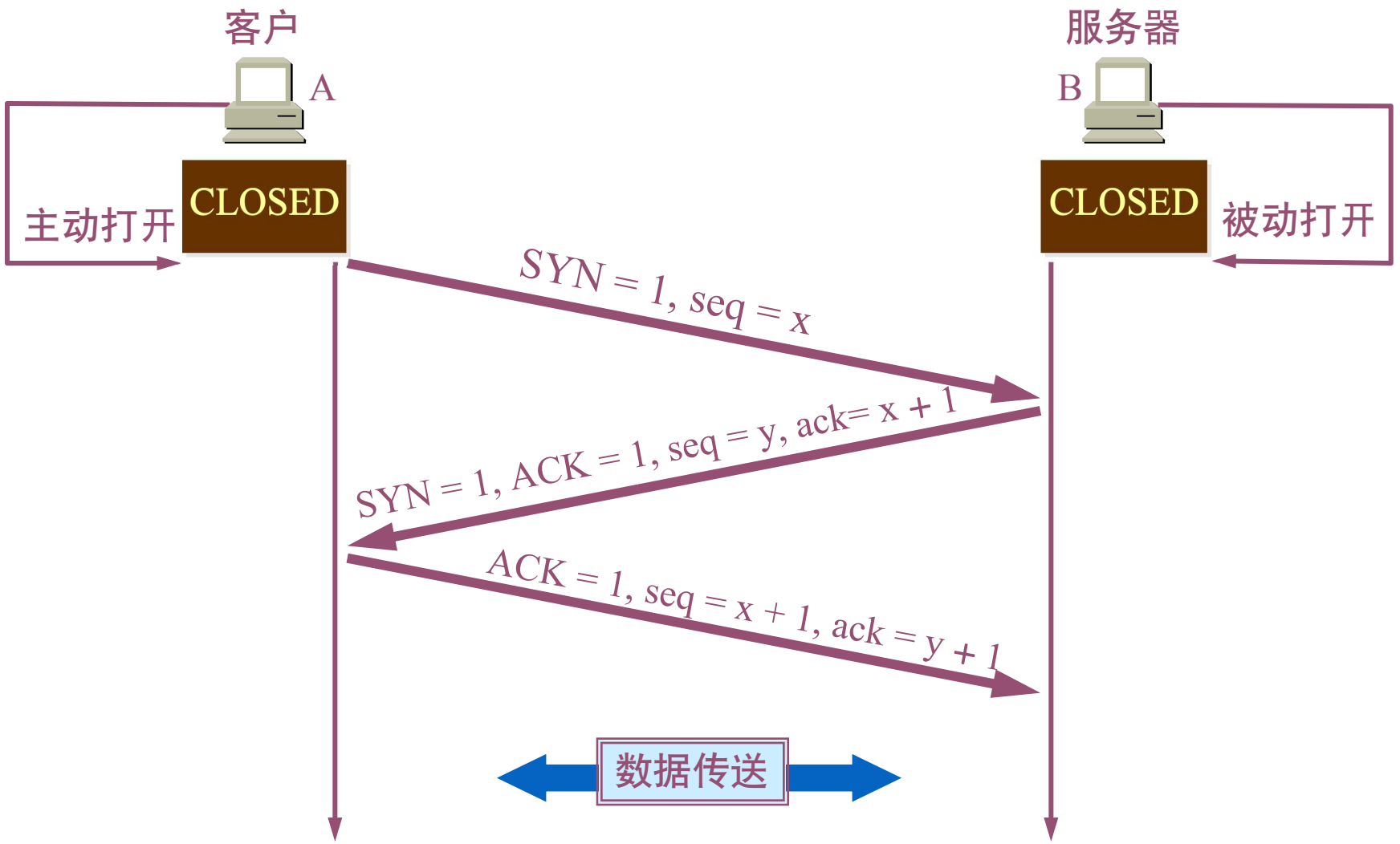



- B 的 TCP 收到连接请求报文段后，如同意，则发回确认。
- B 在确认报文段中应使 SYN = 1，使 ACK = 1，其确认号ack = x + 1，自己选择的序号 seq = y。

- A 收到此报文段后向 B 给出确认，其 ACK = 1，确认号 ack = y + 1。
- A 的 TCP 通知上层应用进程，连接已经建立。

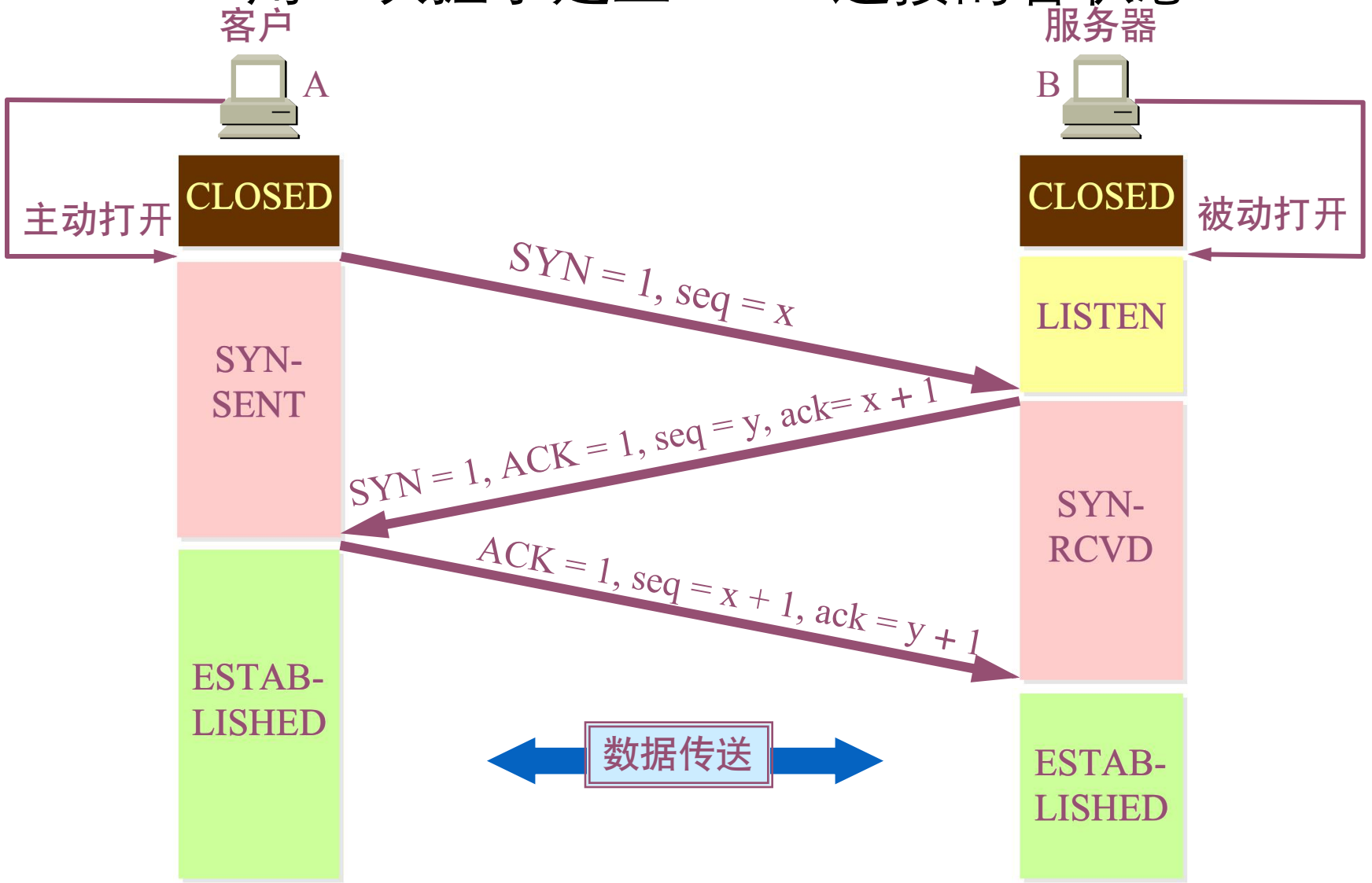


- B 的 TCP 收到主机 A 的确认后，也通知其上层应用进程：TCP 连接已经建立。



# 5.9.1 TCP 的连接建立

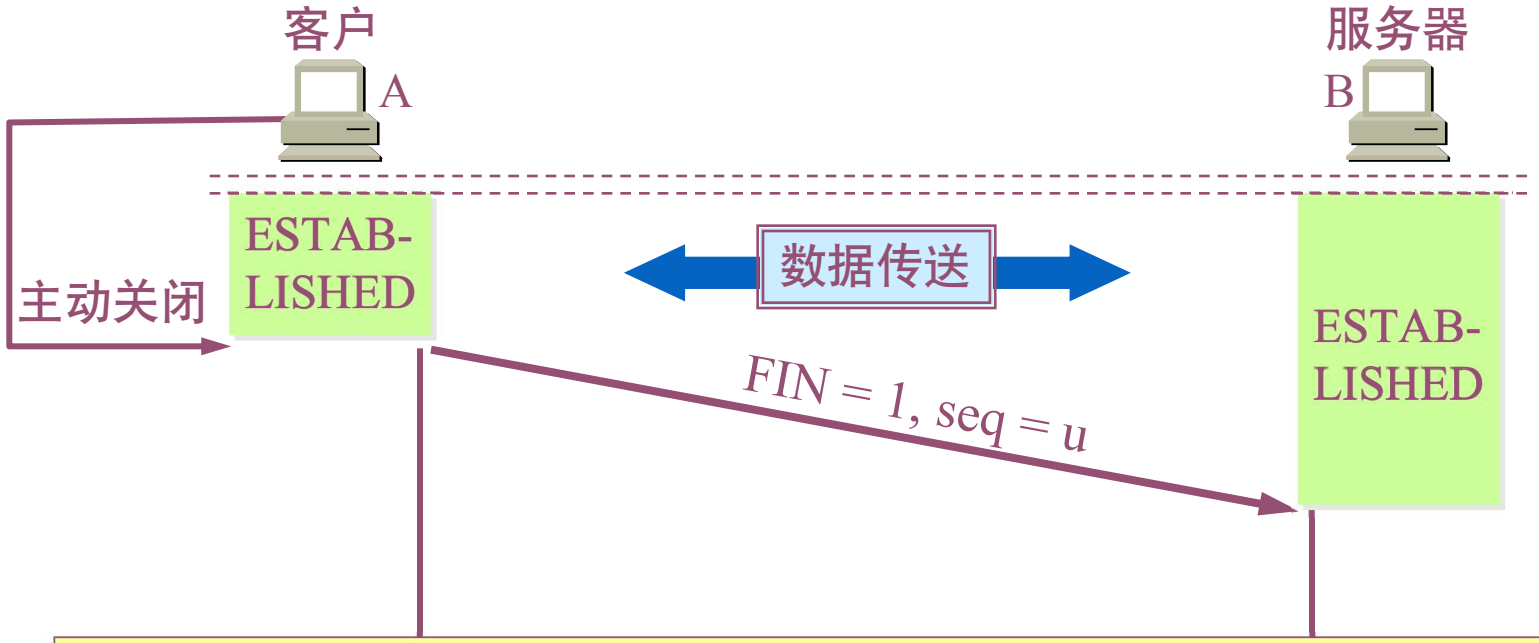
用三次握手建立 TCP 连接的各状态



# 5.9.2　TCP 的连接释放

- TCP 连接释放过程比较复杂。
- 数据传输结束后，通信的双方都可释放连接。
- TCP 连接释放过程是四报文挥手。

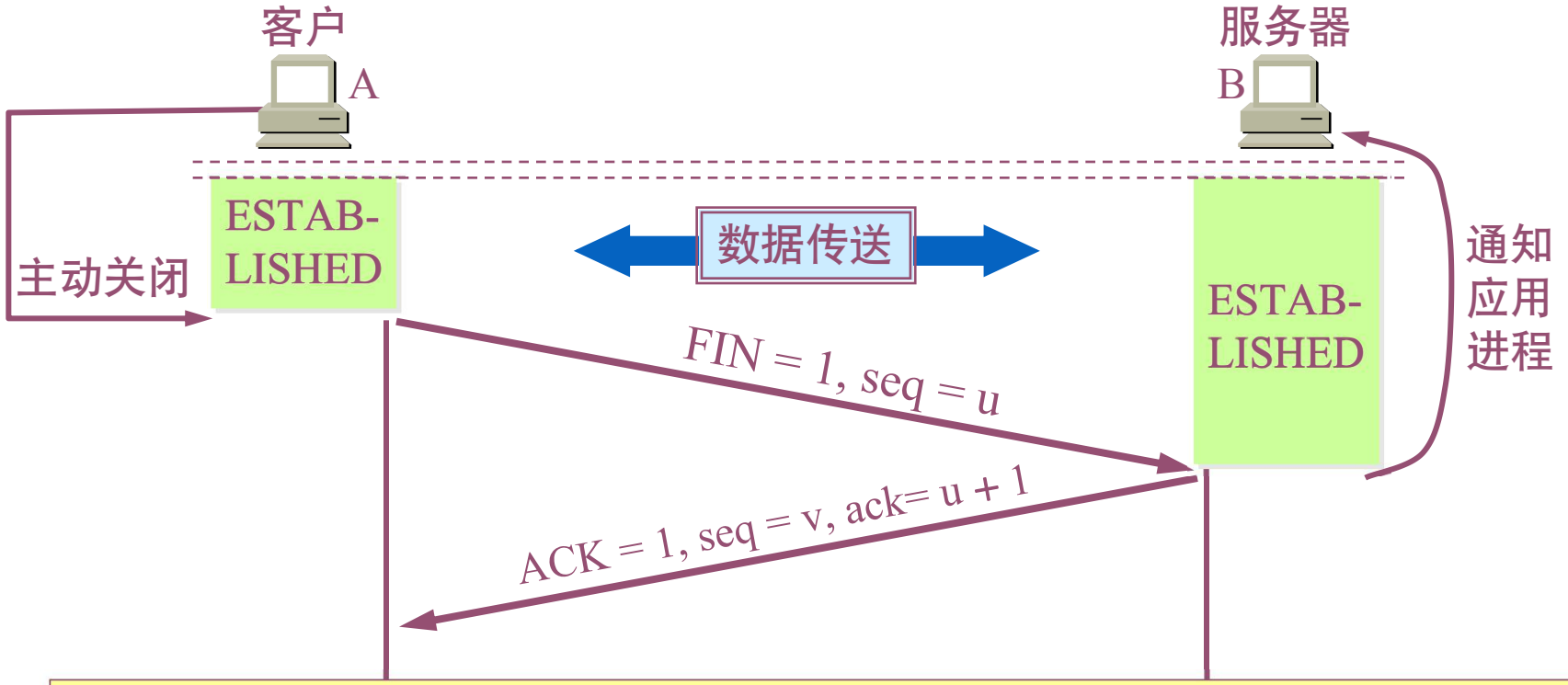
# 5.9.2 TCP 的连接释放




- 数据传输结束后，通信的双方都可释放连接。现在 A 的应用进程先向其 TCP 发出连接释放报文段，并停止再发送数据，主动关闭 TCP 连接。
- A 把连接释放报文段首部的 FIN = 1，其序号 seq = u，等待 B 的确认。

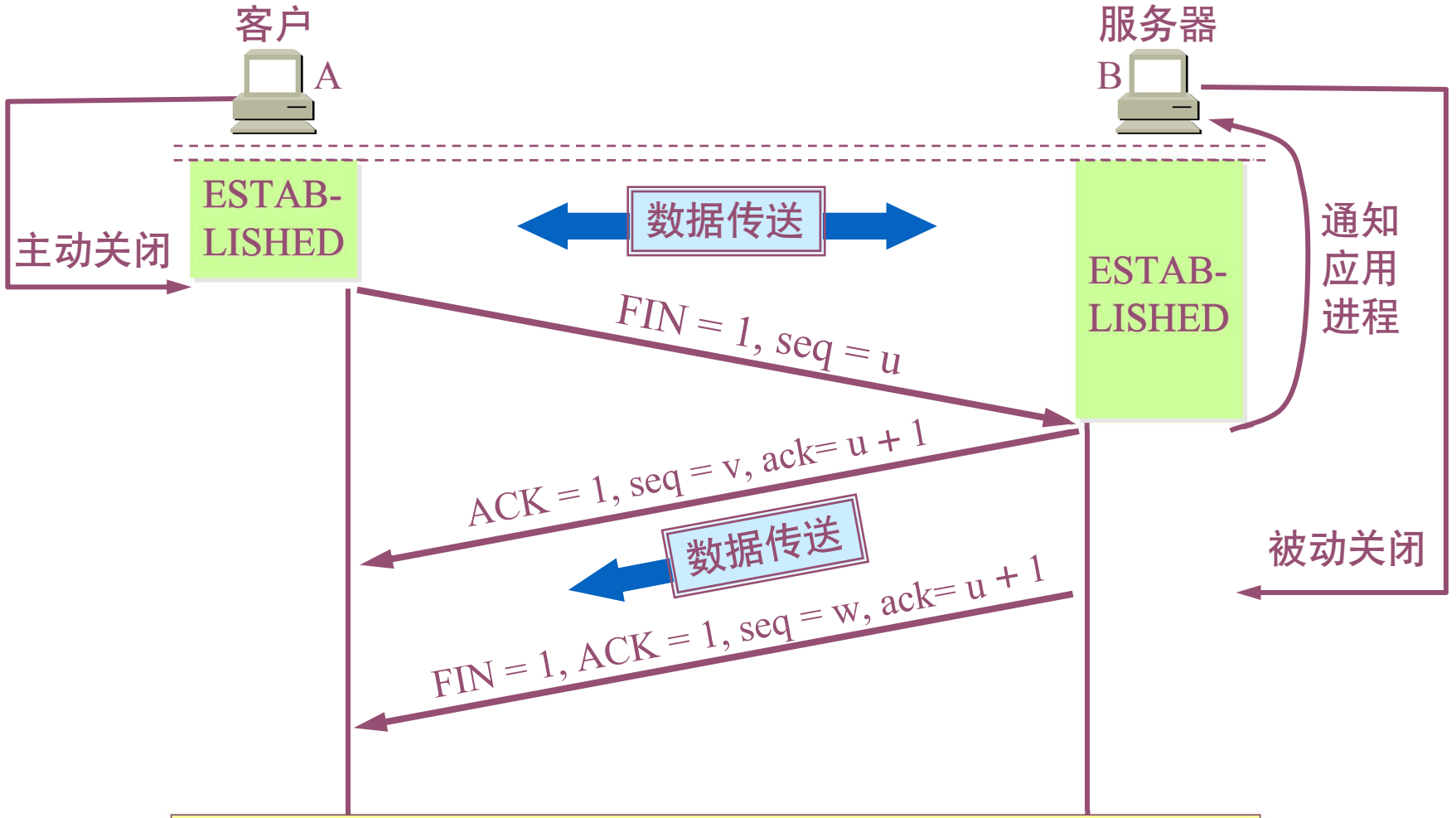
CLOSED

# 5.9.2 TCP 的连接释放



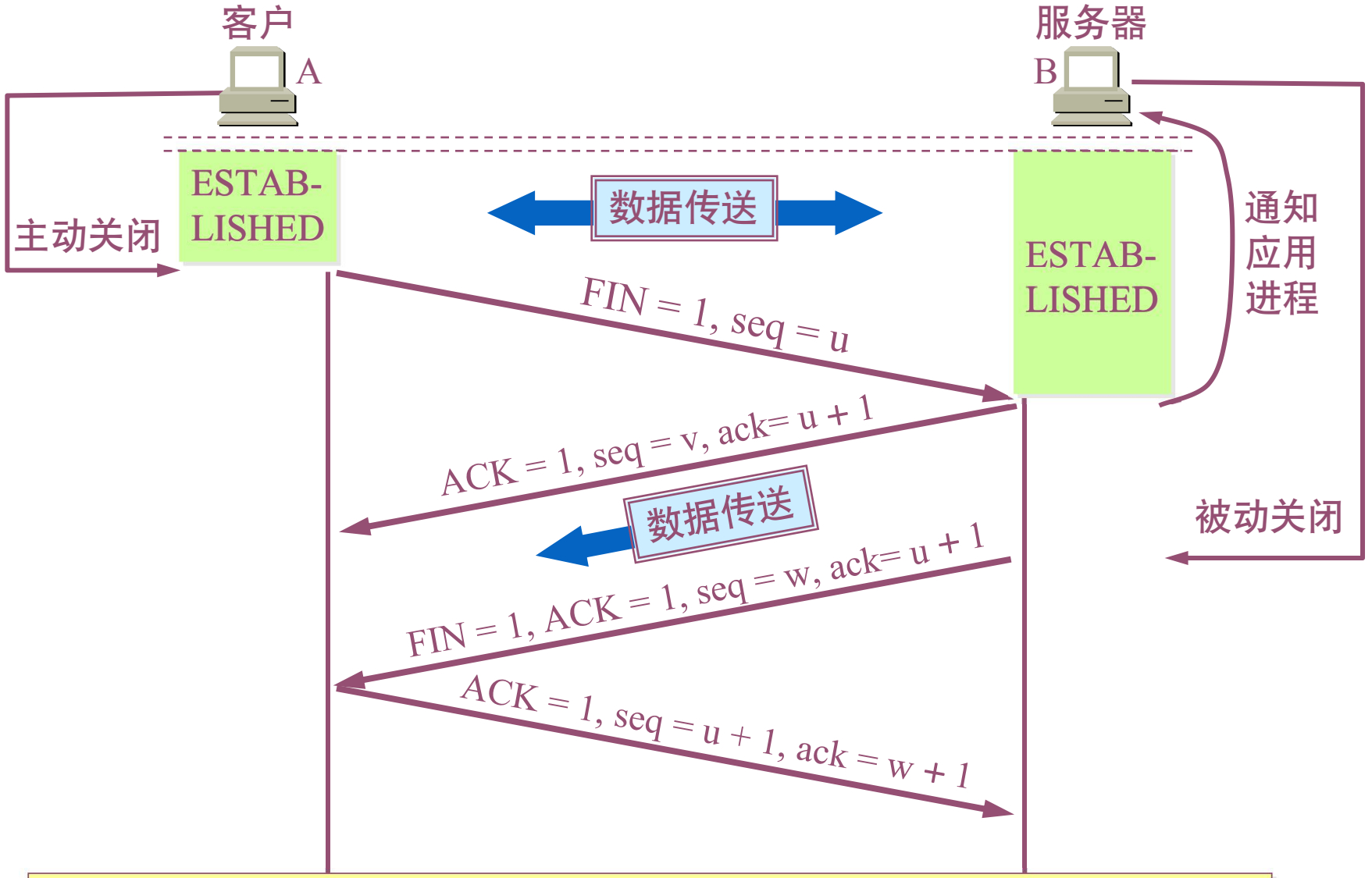

- B 发出确认，确认号 ack = u + 1，而这个报文段自己的序号 seq = v。
- TCP 服务器进程通知高层应用进程。
- 从 A 到 B 这个方向的连接就释放了，TCP 连接处于半关闭状态。B 若发送数据，A 仍要接收。

# 5.9.2 TCP 的连接释放



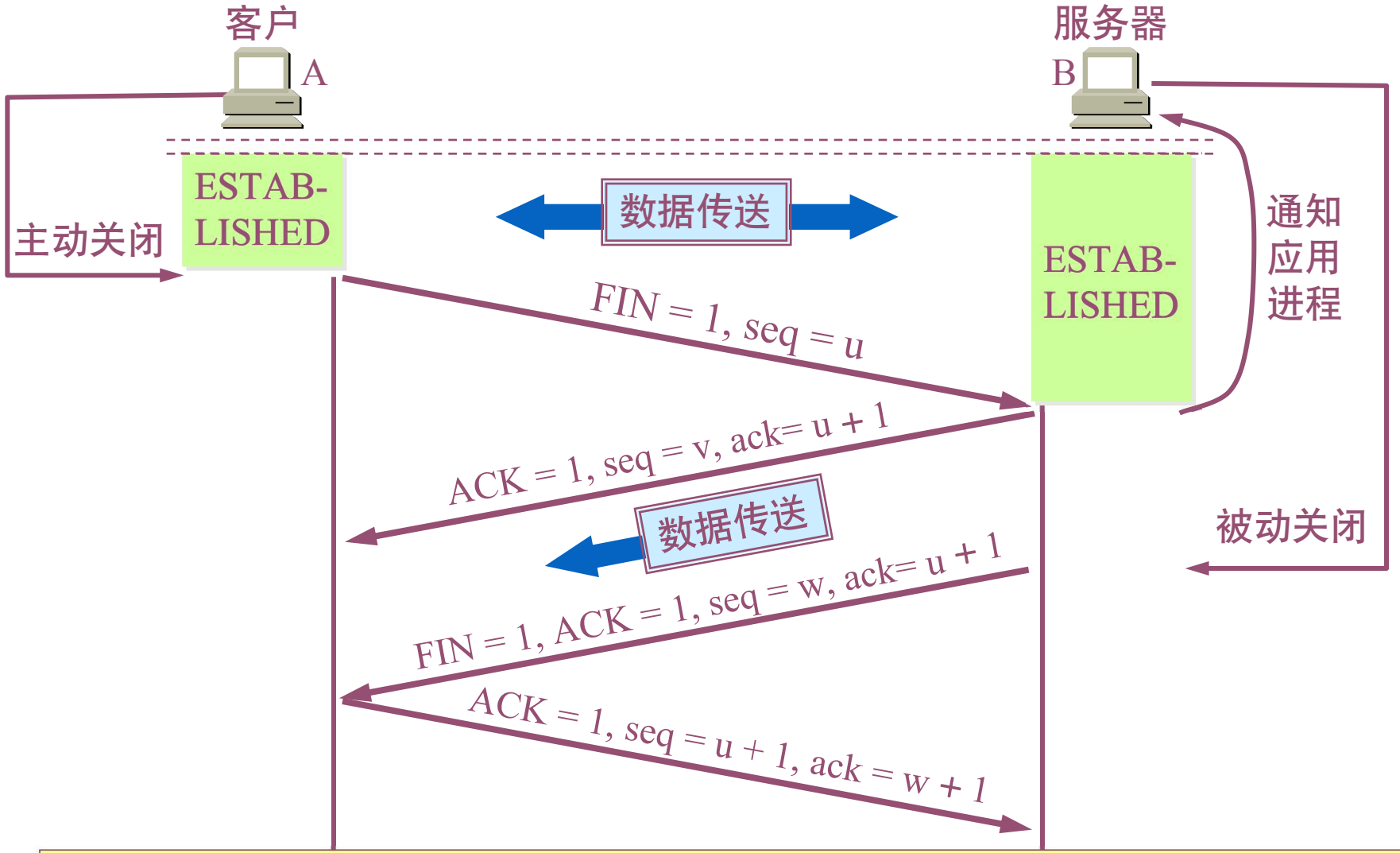

- 若 B 已经没有要向 A 发送的数据，其应用进程就通知 TCP 释放连接。

# 5.9.2 TCP 的连接释放




- A 收到连接释放报文段后，必须发出确认。

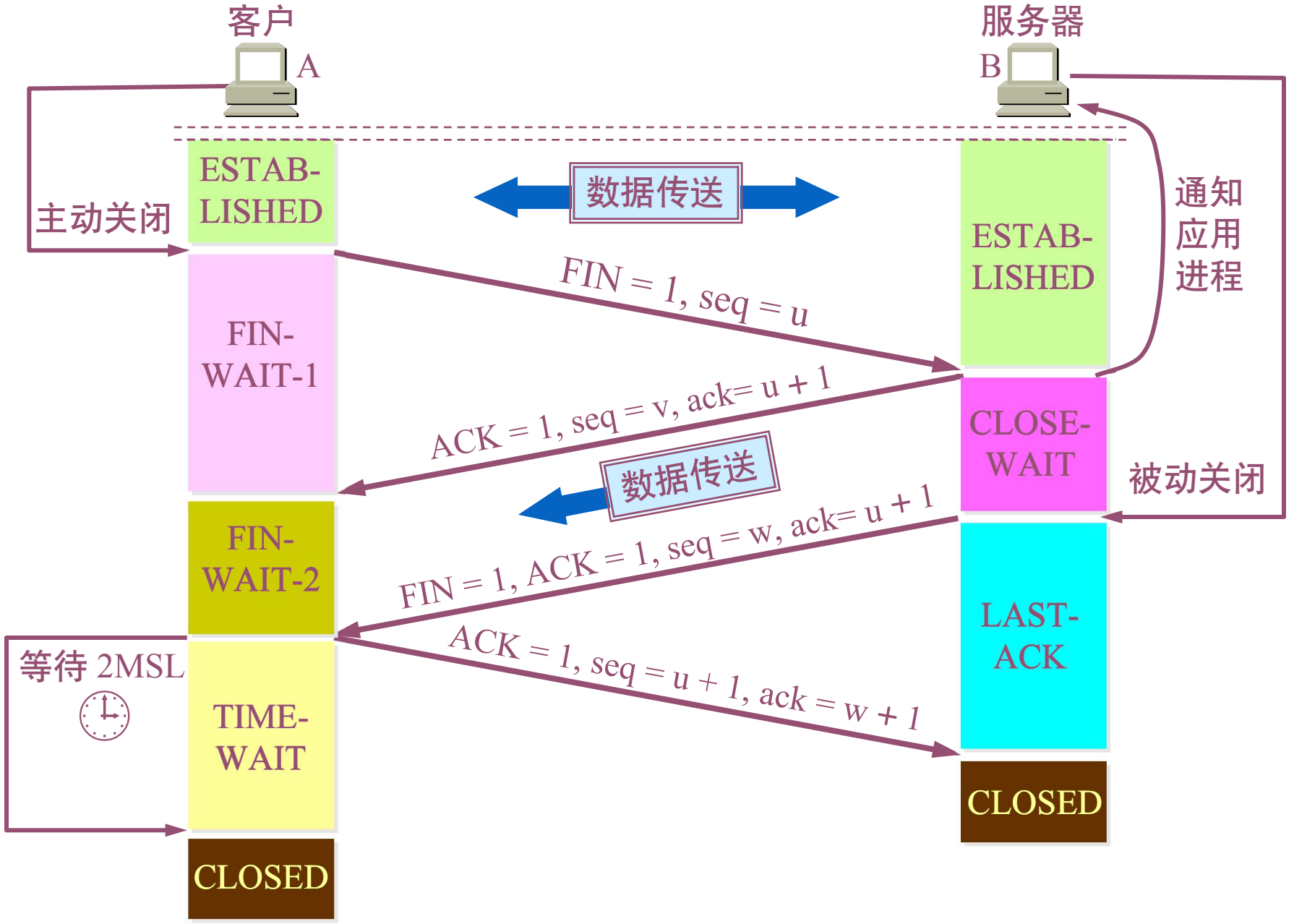
# 5.9.2 TCP 的连接释放




- 在确认报文段中 ACK = 1，确认号 ack = w + 1，自己的序号 seq = u + 1。

TCP 连接必须经过时间 2MSL
（Max Segment Lifetime） 后才真正释放掉。

# TCP 的连接释放



# A 必须等待 2MSL的时间

- 第一，为了保证 A 发送的最后一个 ACK 报文段能够到达 B。
- 第二，防止“已失效的连接请求报文段”出现在本连接中。A 在发送完最后一个 ACK 报文段后，再经过时间 2MSL，就可以使本连接持续的时间内所产生的所有报文段，都从网络中消失。这样就可以使下一个新的连接中不会出现这种旧的连接请求报文段。