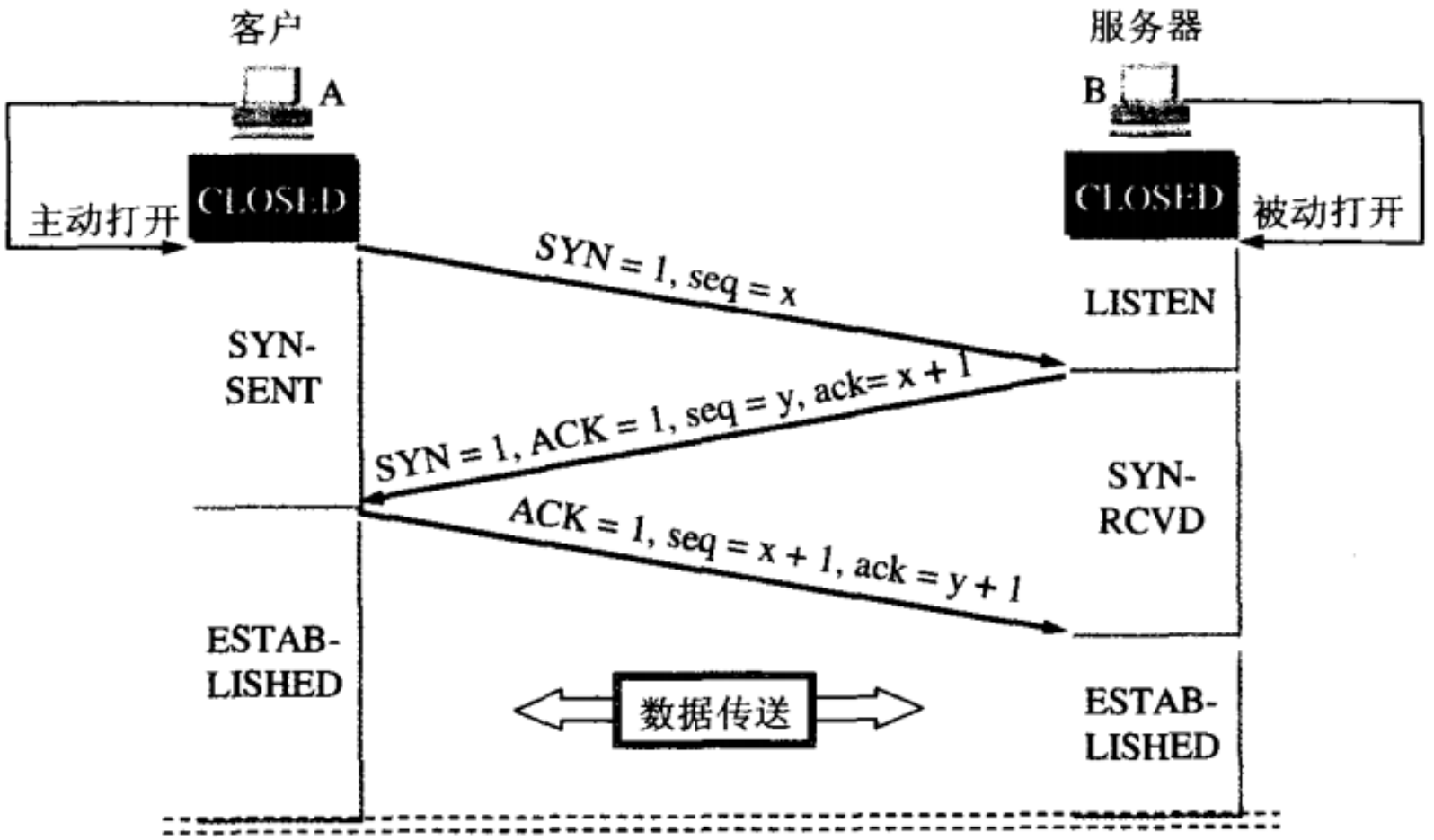
TCP/IP详解学习笔记（13）-- TCP连接的建立与终止

1.TCP连接的建立



设主机B运行一个服务器进程，它先发出一个被动打开命令，告诉它的TCP要准备接收客户进程的连续请求，然后服务进程就处于听的状态。不断检测是否有客户进程发起连续请求，如有，作出响应。设客户进程运行在主机A中，他先向自己的TCP发出主动打开的命令，表明要向某个IP地址的某个端口建立运输连接，过程如下：

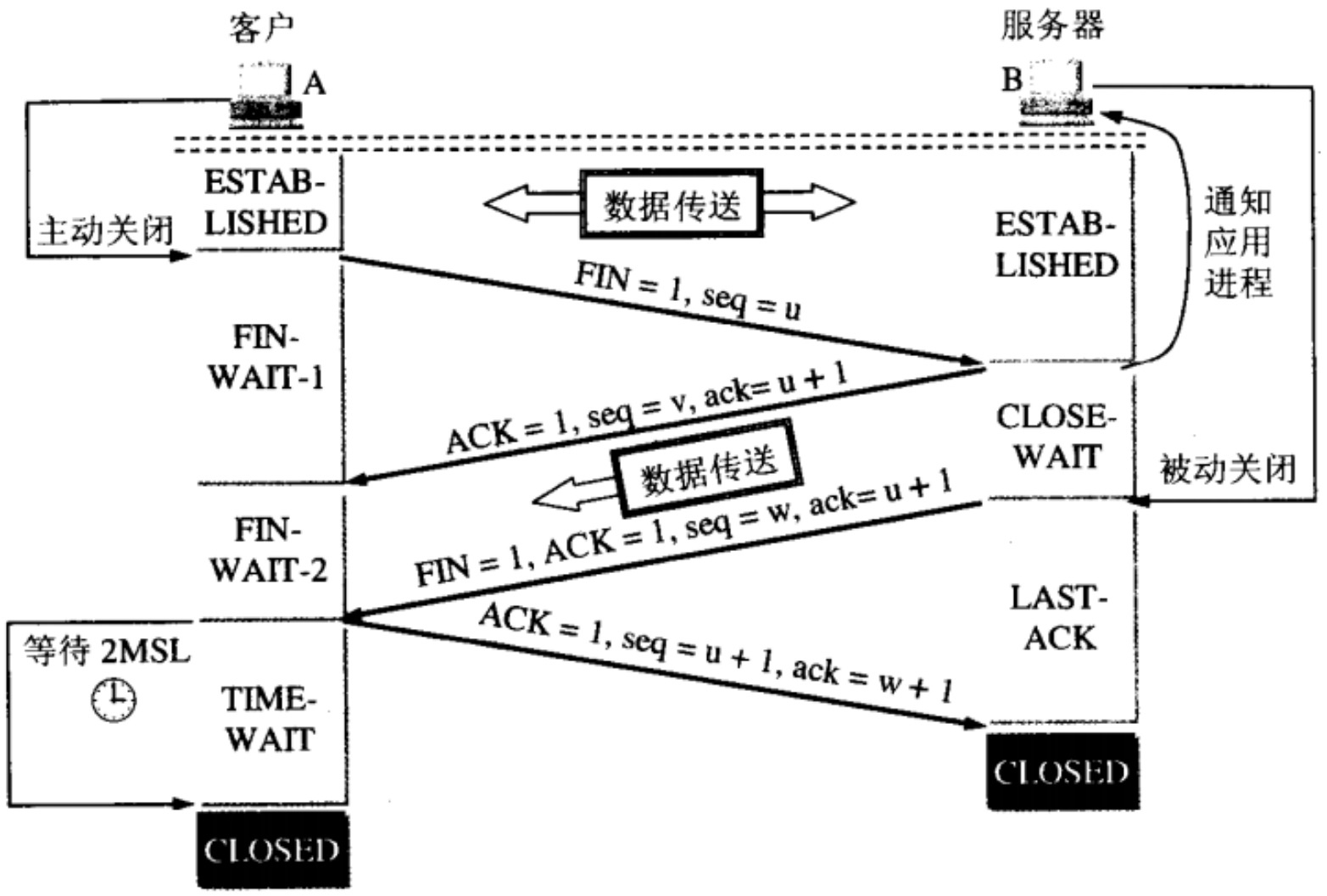
1）主机A的TCP向主机B的TCP发出连接请求报文段，其首部中的同步比特SYN应置1，同时选择一个序号x，表明在后面传送数据时的第一个数据字节的序号是x。

2）主机B的TCP收到连接请求报文段后，如同意，则发挥确认。在确认报文段中应将SYN置为1，确认号应为x+1，同时也为自己选择一个序号y

3）主机A的TCP收到此报文段后，还要向B给出确认，其确认号为y+1

4）主机A的TCP通知上层应用进程，连接已经建立，当主机B的TCP收到主机A的确认后，也通知上层应用进程，连接建立。

2.TCP连接的释放



在数据传输完毕之后，通信双方都可以发出释放连接的请求。释放连接的过程为如上图所示：

1）数据传输结束后，主机A的应用进程先向其TCP发出释放连接请求，不在发送数据。TCP通知对方要释放从A到B的连接，将发往主机B的TCP报文段首部的终止比特FIN置为1，序号u等于已传送数据的最后一个字节的序号加1。

2）主机B的TCP收到释放连接通知后发出确认，其序号为u+1，同时通知应用进程，这样A到B的连接就释放了，连接处于半关闭状态。主机B不在接受主机A发来的数据；但主机B还向A发送数据，主机A若正确接收数据仍需要发送确认。

3）在主机B向主机A的数据发送结束后，其应用进程就通知TCP释放连

接。主机B发出的连接释放报文段必须将终止比特置为1，并使其序号w等于前面已经传送过的数据的最后一个字节的序号加 1，还必须重复上次已发送过的ACK=u+1。

4）主机A对主机B的连接释放报文段发出确认，将ACK置为1，ACK=w+1, seq=u+1。这样才把从B到A的反方向连接释放掉，主机A的TCP再向其应用进程报告，整个连接已经全部释放。

3.注意的问题

- 三次握手建立连接时，发送方再次发送确认的必要性
-
    - 主要是为了防止已失效的连接请求报文段突然又传到了B,因而产生错误。假定出现一种异常情况，即A发出的第一个连接请求报文段并没有丢失，而是在某些网络结点长时间滞留了，一直延迟到连接释放以后的某个时间才到达B，本来这是一个早已失效的报文段。但B收到此失效的连接请求报文段后，就误认为是A又发出一次新的连接请求，于是就向A发出确认报文段，同意建立连接。假定不采用三次握手，那么只要B发出确认，新的连接就建立了，这样一直等待A发来数据，B的许多资源就这样白白浪费了。
- 四次挥手释放连接时，等待2MSL的意义
-
    - 第一，为了保证A发送的最有一个ACK报文段能够到达B。这个ACK报文段有可能丢失，因而使处在LAST-ACK状态的B收不到对已发送的FIN和ACK报文段的确认。B会超时重传这个FIN和ACK报文段，而A就能在2MSL时间内收到这个重传的ACK+FIN报文段。接着A重传一次确认。
    - 第二，就是防止上面提到的已失效的连接请求报文段出现在本连接中，A在发送完最有一个ACK报文段后，再经过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。

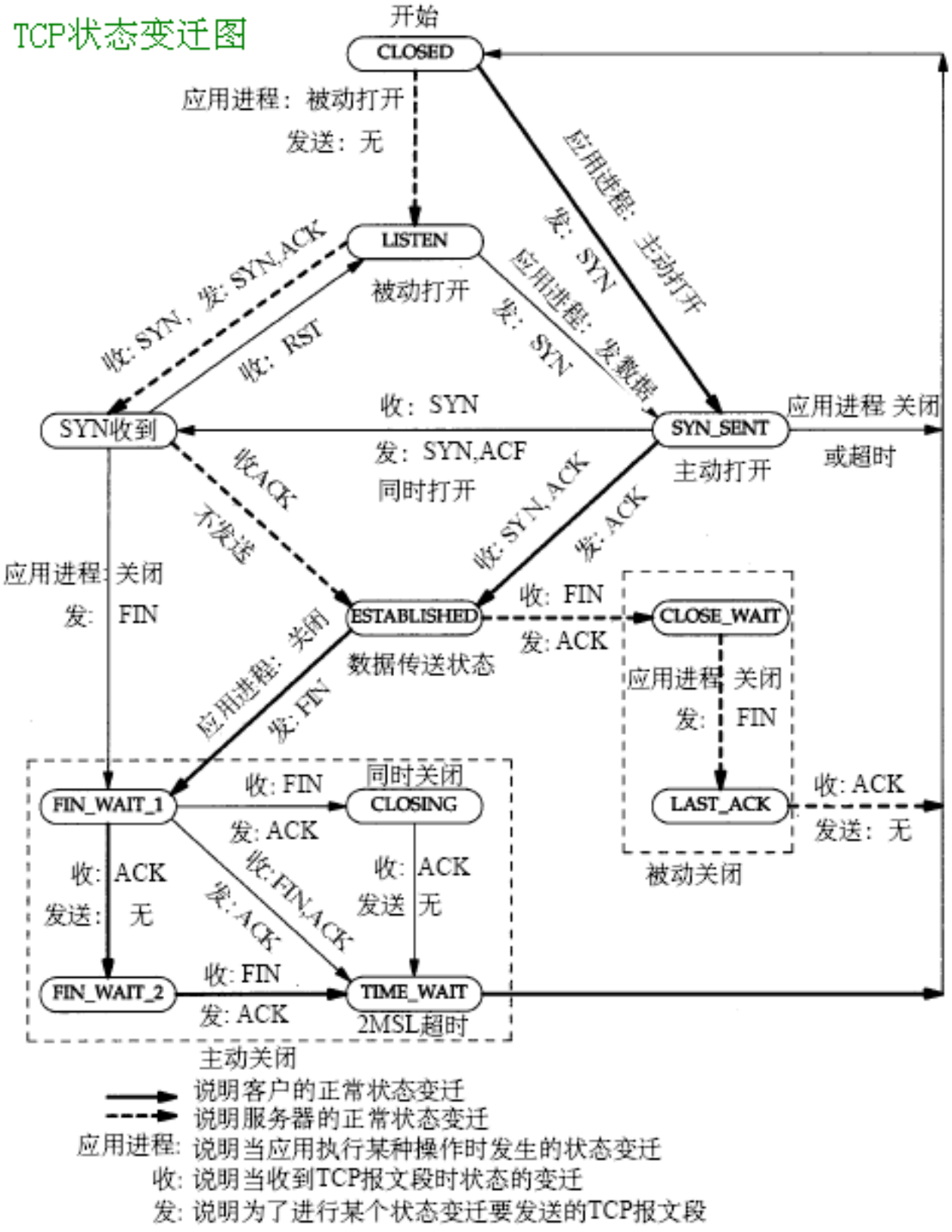
4.TCP的有限状态机

连接的建立和释放所要求的步骤可以用一个有限状态机来表达，该状态机有11种状态。每一种状态中都存在一些合法的事件，当合法事件发生的时候，可能需要采取某个动作。当其他事件发生的时候，则报告一个错误。

状 态	描 述
CLOSED	关闭状态，没有连接活动或正在进行
LISTEN	监听状态，服务器正在等待连接进入
SYN RCVD	收到一个连接请求，尚未确认
SYN SENT	已经发出连接请求，等待确认
ESTABLISHED	连接建立，正常数据传输状态
FIN WAIT 1	（主动关闭）已经发送关闭请求，等待确认
FIN WAIT 2	（主动关闭）收到对方关闭确认，等待对方关闭请求
TIMED WAIT	完成双向关闭，等待所有分组死掉
CLOSING	双方同时尝试关闭，等待对方确认
CLOSE WAIT	（被动关闭）收到对方关闭请求，已经确认
LAST ACK	（被动关闭）等待最后一个关闭确认，并等待所有分组死掉

TCP建立与释放的变迁如图所示：

TCP状态变迁图



- 客户进程变迁的过程（粗实线）
-
    - 连接建立：设一个主机的客户进程发起连接请求（主动打开），这时本地TCP实体就创建传输控制快（TCB），发送一个SYN为1的报文，进入SYN_SENT状态。当收到来自进程的SYN和ACK时，TCP就发送出三次握手中的最后一个ACK，进而进入连接已经建立的状态ESTABLISHED。
    - 连接释放：设运行客户进程主机本地TCP实体发送一个FIN置为1的报文，等待着确认ACK的到达，此时状态变为FIN_WAIT_1。当运行客户进程主机收到确认ACK时，则一个方向的连接已经关闭。状态变成FIN_WAIT_2。当运行客户进程的主机收到运行服务器进程的主机发送的FIN置为1的报文后，应响应确认ACK时，这

是另一个连接关闭。但此时TCP还要等待一段时间后才删除原来建立的连接记录。返回到初始的CLOSED状态，这是为了保证原来连接上的所有分组都从网络中消失了。

- 服务器进程变迁的过程（粗虚线）
-
    - 连接建立：服务器进程发出被动打开，进入监听状态LISTEN。当收到SYN置为1的连接请求报文后，发送确认ACK，并且报文中的SYN也置为1，然后进入SYN_RCVD状态。在收到三次握手最后一个确认ACK时，就转为ESTABLISHED状态。
    - 连接释放：当客户进程的数据已经传送完毕。就发出FIN置为1的报文给服务器进程，进入CLOSE_WAIT状态。服务器进程发送FIN报文段给客户进程，状态变为LAST_ACK状态。当收到客户进程的ACK时，服务器进程就释放连接。删除连接记录。回到原来的CLOSED状态。