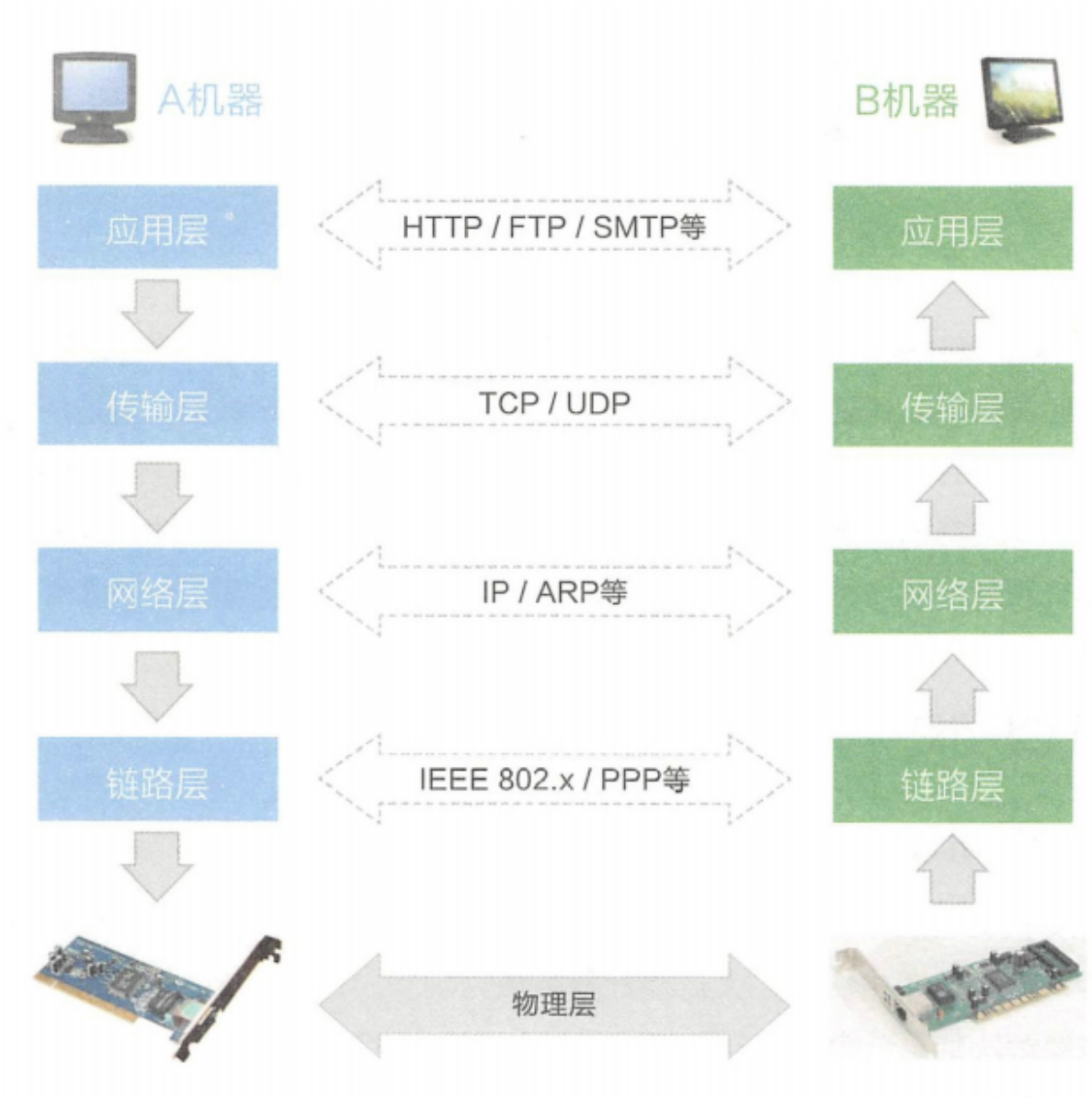
计算机网络

网络协议



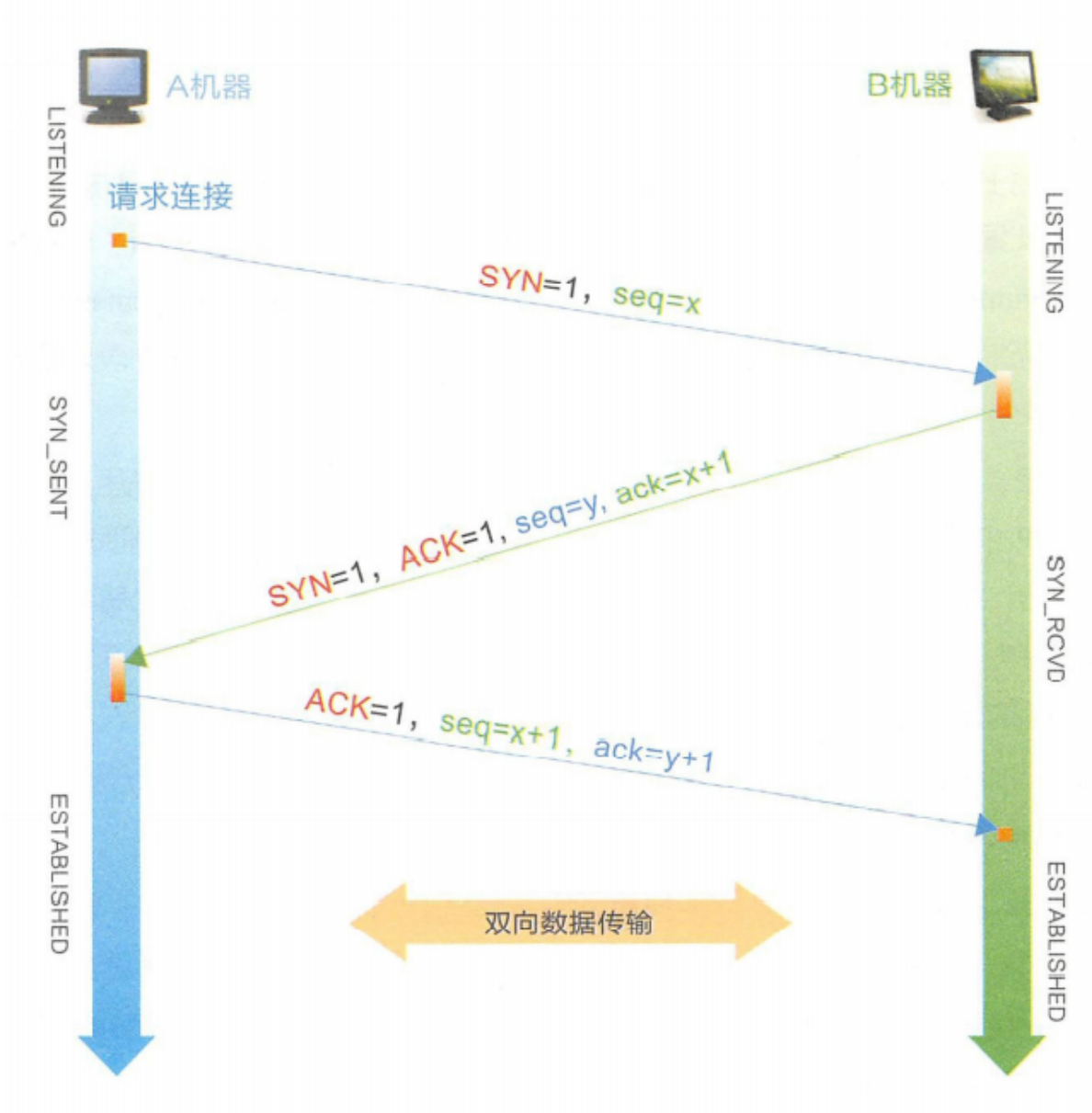
1. 链路层：写入源和目标机器的物理地址、数据、校验位。MAC地址是计算机世界唯一标识。
2. 网络层：根据IP定义网络地址。子网内根据ARP进行MAC寻址，子网外进行路由转发数据包（IP数据包）
3. 传输层：数据包通过网络层发送到目标计算机后，应用程序在逻辑层定义逻辑端口，确认身份后将数据包交给应用程序，实现端口与端口间通信（TCP，UDP）
4. 应用层：传输层数据到达应用程序时，以某种同意规定的协议格式解读数据。

TCP

三次握手：

- 机器发出一个数据包并将 SYN ， 表示希望建立连接。这个包中的序列号假设是x
- B机器收到A机器发过来的数据包后，通过 SYN 得知这是一个建立连接的 请求，于是发送一个响应包并将SYN和ACK 标记都置1 。假设这个包中的 序列号是y ， 而确认序列号必须是 x+1 ， 表示收到了 发过来的 SYN。

- A收到B的响应包后需进行确认，确认包中将ACK值1 ，并将确认序列号设置为 y+1 ，表示收到了来自B的SYN。

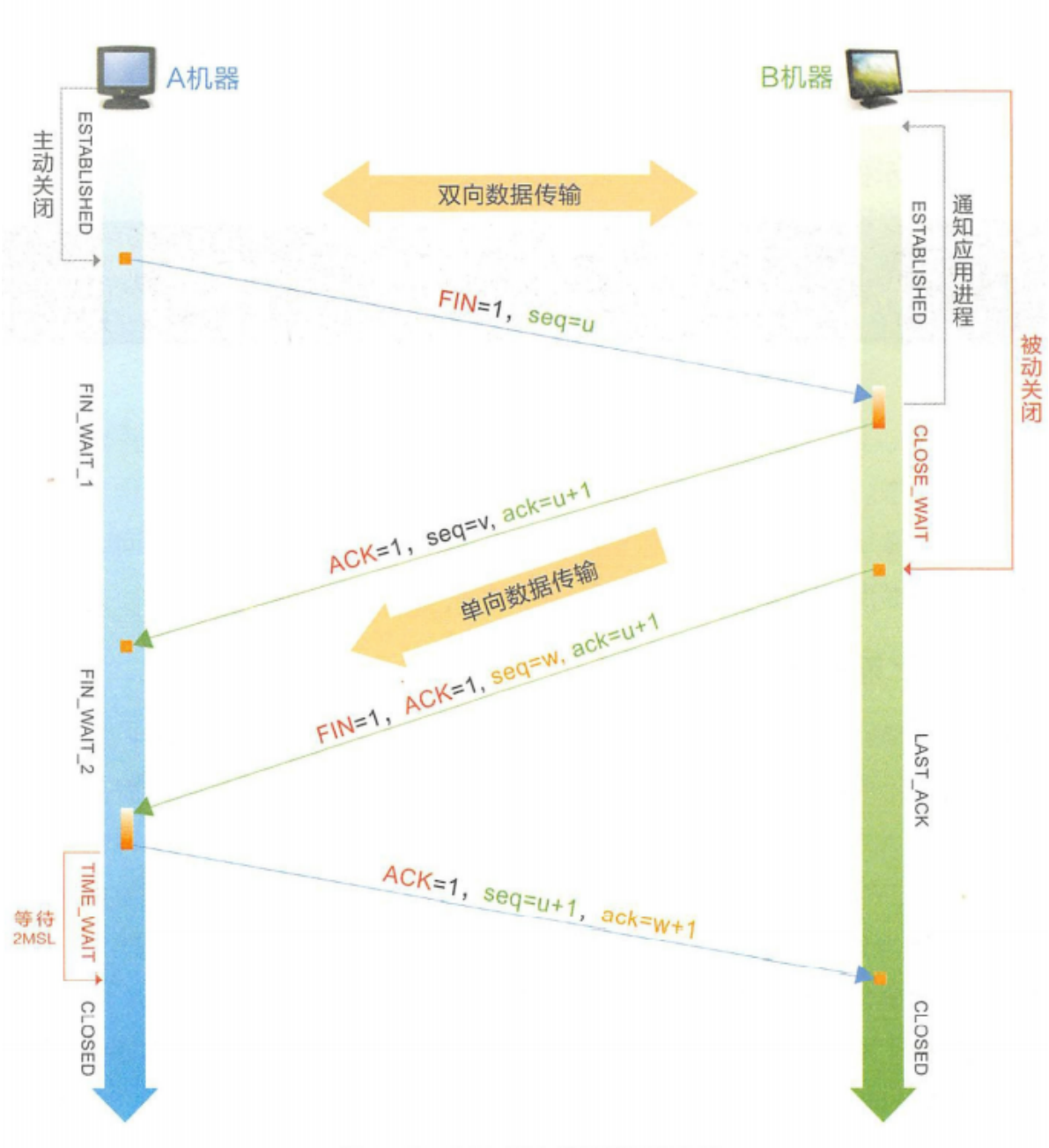


为什么需要三次握手：

1. 信息对等。双方需要确认四类信息才能建立连接：自己发报能力，自己收报能力，对方发报能力，对方收报能力。第三次握手后B才能确认自己的发报能力和对方的收报能力是正常的。
2. 防止请求超时导致脏连接。比如连接超时，报文仍然在网络上传输，然后在连接关闭了之后到达B，最后只有B单方面地创建连接完毕。

四次挥手：

1. A机器想要关闭连接，则待本方数据发送完毕后，传递 FIN 信号给B机器。
2. B机器应答 ACK ，告诉A机器可以断开，但是需要等B机器处理完数据，再主动给A机器发送 FIN 信号。这时，A机器处于半关闭状态（ FIN_WAIT_2 ），无法再发送新的数据。
3. B机器做好连接关闭前的准备工作后，发送 FIN 给A机器，此时B机器也进入半关闭状态（CLOSE_WAIT）
4. A机器发送针对B机器 FIN 的 ACK 后， 进入TIME-WAIT 状态，经过 2MSL (Maximum Segment Lifetime ）后，没有收到B机器传来的报文，则确定B机器已经收到A机器最后发送的 ACK 指令，此时TCP连接正式释放.



为什么要等2MSL：

1. 如果A接收到B发送的FIN和ACK之后直接发送ACK并关闭，可能导致B机器无法确保收到最后的ACK指令，无法进入CLOSED状态（网络问题）
2. 防止失效请求。防止已失效的请求数据包与正常连接的数据包混淆而发生异常。

为什么是2MSL：

最坏情况是：去向ACK消息最大存活时间（MSL) + 来向FIN消息的最大存活时间(MSL)。

为什么要四次：

服务端收到客户端的SYN连接请求报文后，可以直接发送SYN+ACK报文。其中**ACK报文是用来应答的，SYN报文是用来同步的。**但是关闭连接时，当服务端收到FIN报文时，很可能并不会立即关闭SOCKET，所以只能先回复一个ACK报文，告诉客户端，"你发的FIN报文我收到了"。只有等到我服务端所有的报文都发送完了，我才能发送FIN报文，因此不能一起发送。故需要四次挥手。

信息安全

HTTPS:

- 安全套接字层SSL。SSL协议工作于传输层和应用层之间。
- RSA算法：公钥和私钥，非对称加密。私钥用于解密，公钥用于加密（任何人都可以知道）
    - 非对称加密的安全性基于大质数分解的困难性。一方发送消息时，使用另一方的公钥进行加密生成密文，收到密文的一方再通过私钥进行解密。
    - 缺点是加密和解密耗时长，只适合对少量数据进行处理，于是只使用非对称加密来传输密钥本身，最终其实还是用的对称加密。最后使用证书来防止中间人劫持。