TCP/IP是用于实现电子设备之间的互联的一套协议。采用的是分层架构：

网络接口层、网际层、传输层和应用层。

既然是协议，它只是制定了必要的规则，没有规定实现。

1网络接口层主要处理底层物理的数据。

2网际层，可以认为是最主要的一层，IP协议就在该层。

两台设备要互联，不可能世界上每两台电子设备都用两根网线连接，数据都是通过一个个电子设备的传输，到达目的设备。这其中一种叫路由器的设备起了关键的连接作用。IP协议就是保证数据几经辗转能最终到达目的设备的，试想，世界上设备那么多，情况那么复杂，要让数据准确高效到达，是件多么困难的事情。

为了实现上述目标，

首先采用统一地址：IP地址,现在我们还是用Ipv4比较多。

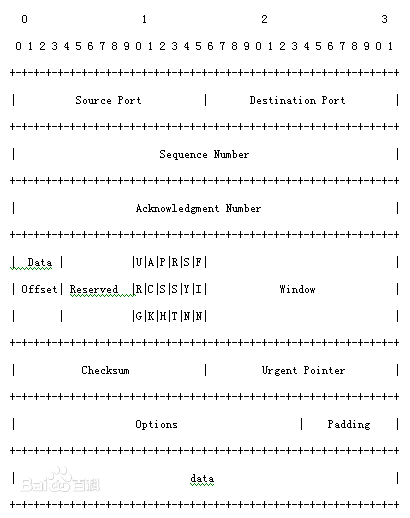
IP地址分了5类：不多说，说了将来也要忘。

与IP地址相关的：子网(有子网就有相应的子网掩码)，DNS(名字与IP的映射)，DHCP(动态主机配置协议)，不多说，用到时才看吧，要不看了也忘。

然后，就是路由算法了，路由算法不止一种，当然都会用到路由表。

所以是网际层保证了设备之间的连接

3传输层，保证的是端到端的连接。



16位端口号（port number）：告知主机该报文段是来自哪里（源端口）以及传给哪个上层协议或应用程序（目的端口）的。进行TCP通信时，客户端通常使用系统自动选择的临时端口号，而服务器则使用知名服务端口号。1.3节中提到过，所有知名服务使用的端口号都定义在/etc/services文件中。

32位序号（sequence number）：一次TCP通信（从TCP连接建立到断开）过程中某一个传输方向上的字节流的每个字节的编号。假设主机A和主机B进行TCP通信，A发送给B的第一个TCP报文段中，序号值被系统初始化为某个随机值ISN（Initial Sequence Number，初始序号值）。那么在该传输方向上（从A到B），后续的TCP报文段中序号值将被系统设置成ISN加上该报文段所携带数据的第一个字节在整个字节流中的偏移。例如，某个TCP报文段传送的数据是字节流中的第1025～2048字节，那么该报文段的序号值就是ISN+1025。另外一个传输方向（从B到A）的TCP报文段的序号值也具有相同的含义。

32位确认号（acknowledgement number）：用作对另一方发送来的TCP报文段的响应。其值是收到的TCP报文段的序号值加1。假设主机A和主机B进行TCP通信，那么A发送出的TCP报文段不仅携带自己的序号，而且包含对B发送来的TCP报文段的确认号。反之，B发送出的TCP报文段也同时携带自己的序号和对A发送来的报文段的确认号。

4位头部长度（header length）：标识该TCP头部有多少个32bit字（4字节）。因为4位最大能表示15，所以TCP头部最长是60字节。

6位标志位包含如下几项：

URG标志，表示紧急指针（urgent pointer）是否有效。

ACK标志，表示确认号是否有效。我们称携带ACK标志的TCP报文段为确认报文段。

PSH标志，提示接收端应用程序应该立即从TCP接收缓冲区中读走数据，为接收后续数据腾出空间（如果应用程序不将接收到的数据读走，它们就会一直停留在TCP接收缓冲区中）。

RST标志，表示要求对方重新建立连接。我们称携带RST标志的TCP报文段为复位报文段。

SYN标志，表示请求建立一个连接。我们称携带SYN标志的TCP报文段为同步报文段。

FIN标志，表示通知对方本端要关闭连接了。我们称携带FIN标志的TCP报文段为结束报文段。

16位窗口大小（window size）：是TCP流量控制的一个手段。这里说的窗口，指的是接收通告窗口（Receiver Window，RWND）。它告诉对方本端的TCP接收缓冲区还能容纳多少字节的数据，这样对方就可以控制发送数据的速度。

16位校验和（TCP checksum）：由发送端填充，接收端对TCP报文段执行CRC算法以检验TCP报文段在传输过程中是否损坏。注意，这个校验不仅包括TCP头部，也包括数据部分。这也是TCP可靠传输的一个重要保障。

16位紧急指针（urgent pointer）：是一个正的偏移量。它和序号字段的值相加表示最后一个紧急数据的下一字节的序号。因此，确切地说，这个字段是紧急指针相对当前序号的偏移，不妨称之为紧急偏移。TCP的紧急指针是发送端向接收端发送紧急数据的方法。我们将在后面讨论TCP紧急数据。



TCP 的三次握手连接：

HTTP协议：应用层使用最广泛的协议。

目的是为了获取网络资源：浏览器、爬虫软件等 向 WEB服务器请求资源，当然还要经过一些中间层。

HTTP协议认为底层提供的是可靠传输，所以传输层使用的是TCP协议。(其实，HTTP没有规定下层一定要用TCP协议，只要能保证可靠传输的协议，都行)

请求：一定要实现GET、HEAD方法。别的方法如POST基本上也都会实现。

响应：

1xx消息——请求已被服务器接收，继续处理

2xx成功——请求已成功被服务器接收、理解、并接受

3xx重定向——需要后续操作才能完成这一请求

4xx请求错误——请求含有词法错误或者无法被执行

5xx服务器错误——服务器在处理某个正确请求时发生错误

HTTP协议特点：

无连接：每次连接处理一个请求，请求完成后，连接断开

无状态：不会记录上次的数据，如果需要上次数据，要重传。(可以在客户端，服务器通过别的方式记住状态)