我们知道OSI模型（用于实现数据通讯）详细划分为七层。分别是：

应用层（application）

表示层（presentation）

会话层（session）

网络层（network）

数据链路层（data link）

物理层（physical）

OSI提供了一个很有用的模式去解释各个不同层面的网络协议，这些协定就像积木一样层层堆叠，因此此架构常被称为堆叠（stack），或是协议堆叠。每一个协议都只和与之对应的协议沟通，然后将结构向相邻的协议解释。彼此之间如何传送资料都有必然的协议，而且层层相扣，共同协力完成任务。各层之间无需知道其他层是如何工作的，只关心相同层级的协议就行。但层与层之间却有一套既定协议**相互交换处理结果**。这样有一个明显的好处是：这留给各层都有自己的设计与发展空间，当某一层要进行协议更新，其它层无需同时被修改。

物理层（physical）：

在这一层里你必须做出一些机械和电子方面的决定，也就是要定义出在终端和网络之间要使用的设备。同时，采用何种布线也要在这里决定出来。这层实际是定义了应用在网络传输中的各种设备规范，以及如何将硬件所携载的信号转换成电脑可以理解的电信号（0和1），这通常都是设备上面韧体的功能。这些规格一般都是由硬件的生产厂商制定。如：数据线的接脚、电压、波长等等。

数据链路层：

在这层指定了要采用的信息单元（message unit，通常在LAN（无线局域网）上面的信息单元被称为frame。即数据包），还有它们的格式、以及如何通过网络。每一个frame都会被赋予一个MAC位址码和侦错监测值（checksum）。

Ethernet的的data link layer frame （数据包）看起来如下图：



Data link layer frame 结构：

数据在实体层是以bit为单位来传输的，数据链路层要制定不同网络形态的数据包格式，确保数据能够在不同的网络实体（光纤、电话数据线等）上进行资料传输。

数据链路层通常会管辖以下的功能：

* 网卡的物理地址（physical address），也被称为MAC Address
* 虚拟电路（virtual circuit）连接和逻辑连接（logical link）的建立和结束
* 控制数据包的传送和错误检测方式
* 数据包的传送和接收顺序及传递方式
* 在得不到回应或重复发送的情形下进行修复的程序
* 处理物理层的转换和管理
* 对数据包进行侦错和确认
* 检查发送数据包的目的地位置以确保资料能正确的被送抵目的地的网络层

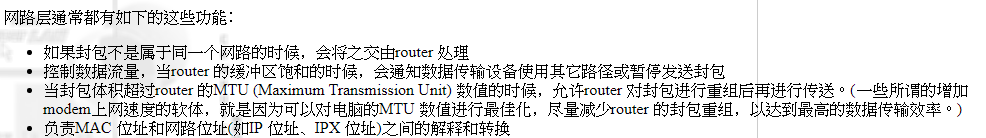
总的来说，这层的工作就是保证一个无错误的物理上的数据传输。

网络层（network layer）

这层就好比是一个中间人存在于网络功能和使用者功能之间。它会定义出封装包在网络中移动的路由和其处理过程。这层还决定了网络是如何进行管理功能的。

网络层的主要功能是让封装包（packet）在不同的网络之间成功的进行传递，它规定了网络的定址方式，及处理资料在不同网络之间的传递方式、处理子网络之间的传递、决定路由路径、网络环境、资料处理顺序等。

发送端电脑在封装包被传送出去之前，都会先为其建立header。作为在网络或子网间进行路由的依据。网络层在辨认和处理资料的时候，会忽略由高层协定制定的定义，只负责为数据在一个或多个网络间建立、维护和终止连接。



传输层（transport layer）：

在这层，将会设定如何控制节点之间的资料传递，还有错误检测和修正的方法。

由于大多数网络，由于物理上面的限制，一次所通过的数据通常只有数千byte而已（IP封包最大体积为65536bytes），然而许多需要在网络中传输的资料都会超过这个数值。传输层的主要功能是确保电脑资料正确的传送到目的地。它的工作就是“打包”，也就是将电脑资料变成封包的形态，再赋予一定的检测手段，将资料正确的传到目的电脑，然后再将封包重组回资料。封包如果残缺则重复，也能够将重复的封包剔除。

传送层可以等资料收集到足够大的数量的时候才发送出去，并非应用程序每次产生一个数据就进行一次传送，也就能减少不必要的传输次数，以保证高效率的传输；反之，当应用程序产生大资料量数据时，则将之分拆成较小的封包在进行传送。

传输层的主要功能有：

* 接管由上层协议传来的资料，并进行“分拆”和“打包”等工作。
* 利用点对点的方式进行资料传送和回应的确认
* 在得到接收端的资料缓冲区饱和信息之后，暂停资料发送
* 能在单一位址上处理不同的程式协议（如ftp，http），并分别进行追踪及转换