## 网络7层

第一层，物理层  
OSI模型最低层的“劳苦大众”。它透明地传输比特流，就是传输的信号。该层上的设备包括集线器、发送器、接收器、电缆、连接器和中继器。  
  
第二层，数据链路层  
这一层是和包结构和字段打交道的和事佬。一方面接收来自网络层（第三层）的数据帧并为物理层封装这些帧；另一方面数据链路层把来自物理层的原始数据比特封装到网络层的帧中。起着重要的中介作用。  
数据链路层由IEEE802规划改进为包含两个子层：介质访问控制（MAC）和逻辑链路控制（LLC）。  
智能集线器、网桥和网络接口卡（NIC）等就驻扎在这一层。但是网络接口卡它同样具有物理层的一些编码功能等。  
  
第三层，网络层  
这一层干的事就比较多了。它工作对象，概括的说就是：电路、数据包和信息交换。   
网络层确定把数据包传送到其目的地的路径。就是把逻辑网络地址转换为物理地址。如果数据包太大不能通过路径中的一条链路送到目的地，那么网络层的任务就是把这些包分成较小的包。  
这些光荣的任务就派给了路由器、网桥路由器和网关。  
以后几层属于较高层，通常驻留在跨网络相互通信的计算机中，而不象以上几层可以独自为阵。设备中只有网关可跨越所有各层。  
  
第四层，传输层。  
确保按顺序无错的发送数据包。传输层把来自会话层的大量消息分成易于管理的包以便向网络发送。  
  
第五层，会话层。  
在分开的计算机上的两种应用程序之间建立一种虚拟链接，这种虚拟链接称为会话（session）。会话层通过在数据流中设置检查点而保持应用程序之间的同步。允许应用程序进行通信的名称识别和安全性的工作就由会话层完成。  
  
第六层，表示层。  
定义由应用程序用来交换数据的格式。在这种意义上，表示层也称为转换器（translator）。该层负责协议转换、数据编码和数据压缩。转发程序在该层进行服务操作。  
  
第七层，应用层，该层是OSI模型的最高层。应用层向应用进程展示所有的网络服务。当一个应用进程访问网络时，通过该层执行所有的动作。  
纵观七层，从低级到高级。作一个形象的比喻就是从汇编到了BASIC，越到高层与硬件的关联就越弱。

所谓的网络七层协议就是OSI模型，具体分为：应用层、表示层、会话层、传输层、网络层、数据链路层、物理层。

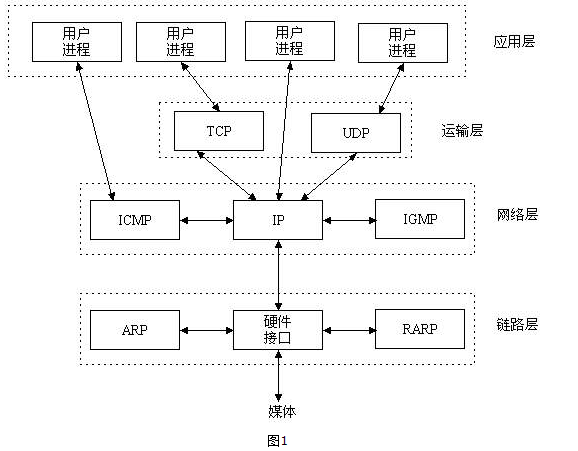
7——应用层   
6——表示层   
5——会话层   
4——传输层   
3——网络层   
2——数据链路层   
1——物理层

物理介质  
七层模型在Windows程序下的体现：  
物理层----就是我们看得见的网卡。网卡的作用就是把线路发送过来的高频电流转化数据包，然后传给网卡驱动程序，同是也把网卡驱动程序传送过来的数据包转化成电信号传送出去。定义通过网络设备发送数据的物理方式：是网络媒介和设备间的接口。  
数据链路层----是网卡驱动程序。定义控制通信连接的程序；封包；监测和改正包传输错误。  
网络层----即NDIS，NDIS提供网络接口。决定网络设备间如何传输数据；根据唯一的网络设备地址选择包；提供流和拥塞控制，以阻止同时网络资源的损耗。  
传输层----即TCP，TCP协议的封包处理是在这一层进行的。管理网络中首尾连接的信息传送；提供通过错误恢复和流控制装置传送可靠且有序的包；提供无连接面向包的传送。  
会话层----即SPI，SPI是服务提供者接口，管理用户间的会话和对话；控制用户间的连接和挂断连接；报告上层错误。  
表示层----API，它为应用程序提供接口。API负责SPI与应用程序之间的通信；定义不同体系间不同数据格式；具体说明独立结构的数据传输格式；编码和解码数据；加密和解密数据；压缩和解压缩数据。  
应用层----EXE，就是大家常见的应用程序。定义用于网络通信和数据传输的用户接口程序；提供标准服务，比如虚拟终端、文档以及任务的传输和操作。  
七层协议与Windows结构的生产力映射如下：  
7 应用层 7 应用程序（exe）  
6 表示层 6 Winsock API（dll）  
5 会话层 5 SPI（dll）  
4 传输层 4 TDI（vxd、sys）  
3 网络层 3 NDIS（vxd、sys）  
2 数据链路层 2 网卡驱动程序（vxd、sys）  
1 物理层1网卡

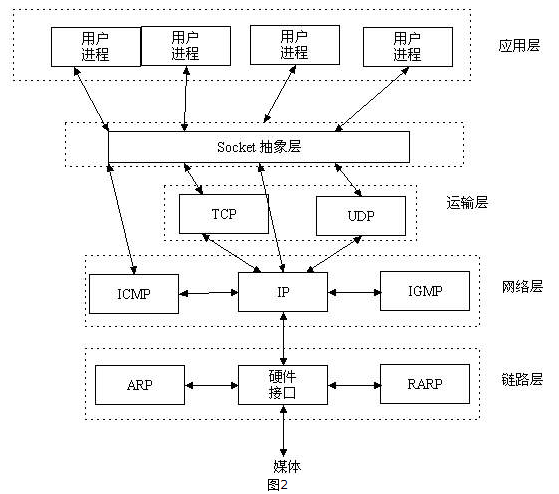
## Socket网络编程的原理

 对TCP/IP、UDP、Socket编程这些词你不会很陌生吧？随着网络技术的发展，这些词充斥着我们的耳朵。那么我想问：  
  
1.什么是TCP/IP、UDP？  
2.Socket在哪里呢？  
3.Socket是什么呢？  
4.你会使用它们吗？  
  
**什么是TCP/IP、UDP？**

TCP/IP（Transmission Control Protocol/Internet Protocol）即传输控制协议/网间协议，是一个工业标准的协议集，它是为广域网（WANs）设计的。  
UDP（User Data Protocol，用户数据报协议）是与TCP相对应的协议。它是属于TCP/IP协议族中的一种。  
这里有一张图，表明了这些协议的关系。



**Socket在哪里呢？**在图1中，我们没有看到Socket的影子，那么它到底在哪里呢？还是用图来说话，一目了然。



**Socket是什么呢？**Socket是应用层与TCP/IP协议族通信的中间软件抽象层，它是一组接口。在设计模式中，Socket其实就是一个门面模式，它把复杂的TCP/IP协议族隐藏在Socket接口后面，对用户来说，一组简单的接口就是全部，让Socket去组织数据，以符合指定的协议。  
**你会使用它们吗？**前人已经给我们做了好多的事了，网络间的通信也就简单了许多，但毕竟还是有挺多工作要做的。以前听到Socket编程，觉得它是比较高深的编程知识，但是只要弄清Socket编程的工作原理，神秘的面纱也就揭开了。  
一个生活中的场景。你要打电话给一个朋友，先拨号，朋友听到电话铃声后提起电话，这时你和你的朋友就建立起了连接，就可以讲话了。等交流结束，挂断电话结束此次交谈。生活中的场景就解释了这工作原理，也许TCP/IP协议族就是诞生于生活中，这也不一定。

