

# 简单理解Socket

## 题外话

前几天和朋友聊天，朋友问我怎么最近不写博客了，一个是因为最近在忙着公司使用的一些控件的开发，浏览器兼容性搞死人；但主要是因为这段时间一直在看html5的东西，看到web socket时觉得很有意思，动手写几个demo，但web socket需要特定的服务器支持，由于标准制定工作还没完成，所以没有多少主流的服务器支持，自己在网上下载了几个实现，包括php的、C#的、甚至Node.js的，但一个是协议变化比较大，很多代码已经过时了，再就是有一些支持最新的标准，但是我想稍微改造一下，看人家源代码的时候云里雾里，看看别人的代码行数也不多，决定自己实现一个。

悲剧由此开始，虽然哥们儿国内非知名工科大学毕业，但好歹也是科班CS出身，但大学得过且过，什么TCP/IP协议，什么socket了都没概念。为了做出一个简单的支持广播的websocket server，在网上找了很多相关代码，左抄一句，右抄一句，弄了一个星期竟然还是漏洞百出，调试不起来，只好从头来过了，先补一些基本知识，然后再一步步根据原理实现，今天终于实现了绝大部分功能，由此真的感受到了，搞计算机必须得有理论指导实践，否则只能像个没头苍蝇到处乱撞。

## TCP/IP

要想理解socket首先得熟悉一下TCP/IP协议族，TCP/IP ( Transmission Control Protocol/Internet Protocol ) 即传输控制协议/网间协议，定义了主机如何连入因特网及数据如何在它们之间传输的标准，

从字面意思来看TCP/IP是TCP和IP协议的合称，但实际上TCP/IP协议是指因特网整个TCP/IP协议族。不同于ISO模型的七个分层，TCP/IP协议参考模型把所有的TCP/IP系列协议归类到四个抽象层中

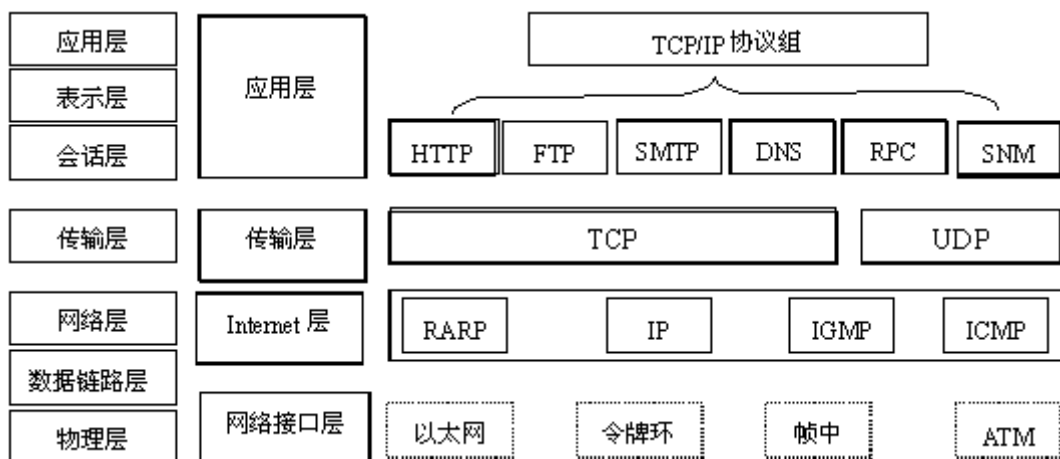
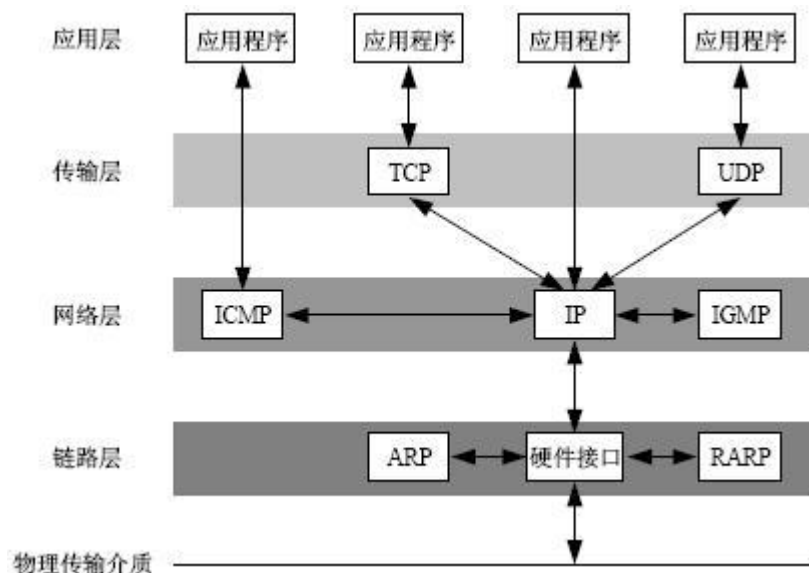
应用层：TFTP，HTTP，SNMP，FTP，SMTP，DNS，Telnet 等等

传输层：TCP，UDP

网络层：IP，ICMP，OSPF，EIGRP，IGMP

数据链路层：SLIP，CSLIP，PPP，MTU

每一抽象层建立在低一层提供的服务上，并且为高一层提供服务，看起来大概是这样的



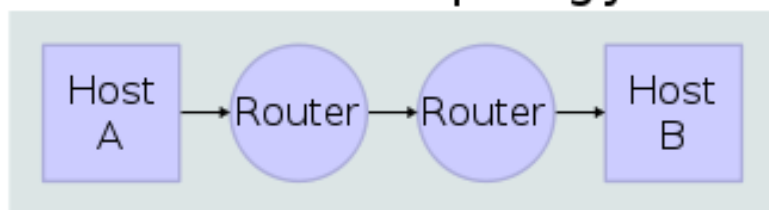
估计有兴趣打开此文的同学都对此有一定了解了，加上我也是一知半解，所以就不详细解释，有兴趣同学可以上网搜一下资料

[维基百科](#)

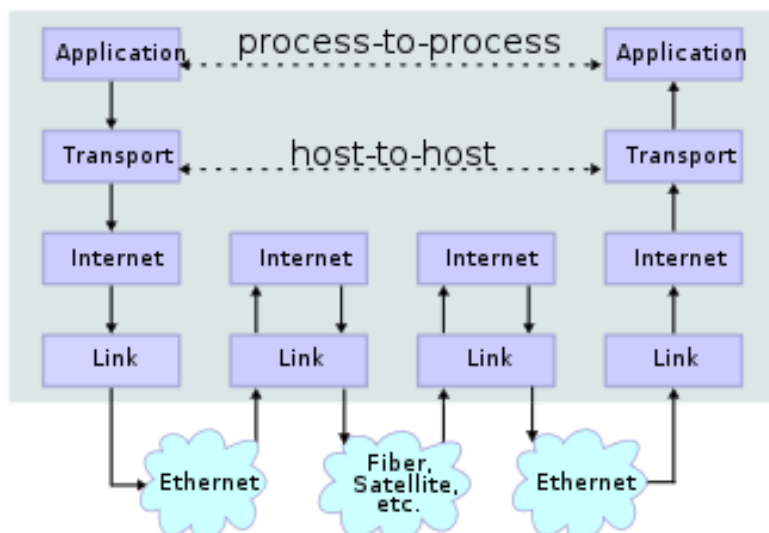
[百度百科](#)

在TCP/IP协议中两个因特网主机通过两个路由器和对应的层连接。各主机上的应用通过一些数据通道相互执行读取操作

# Network Topology



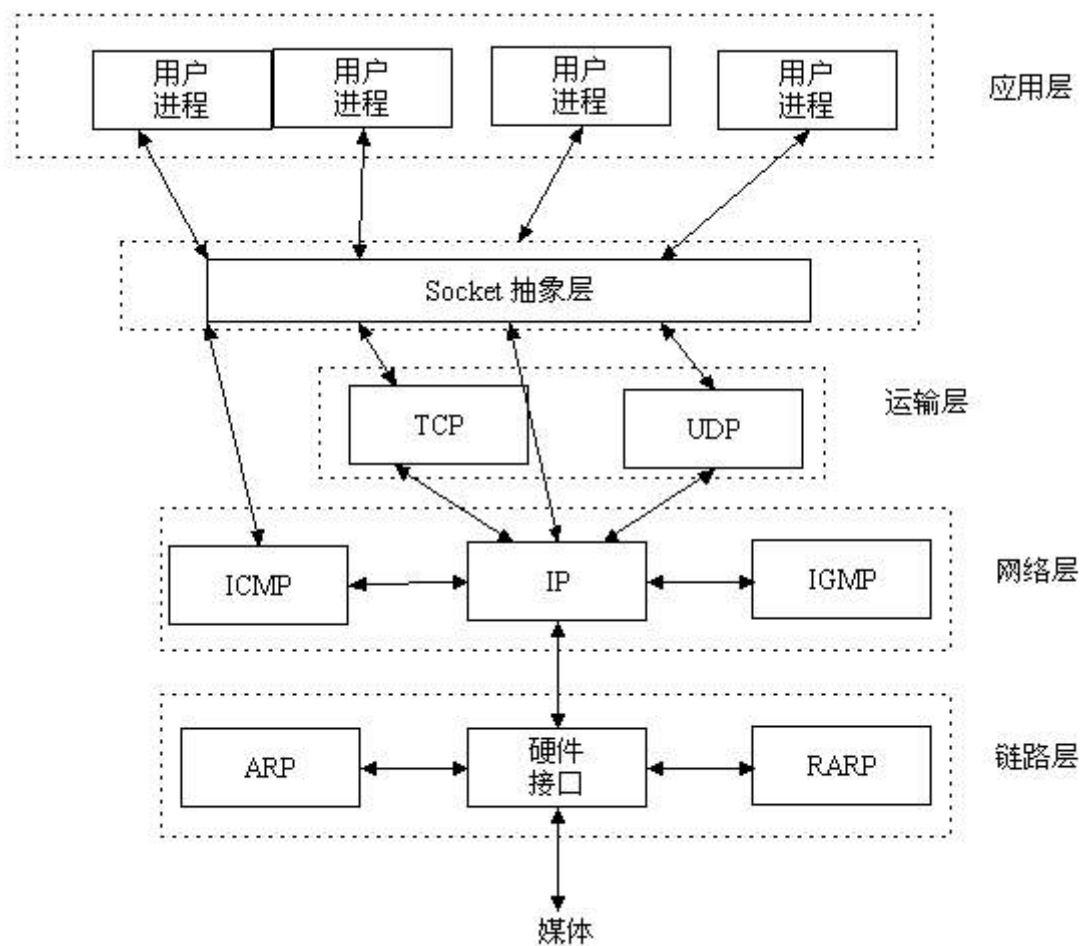
## Data Flow



## socket

我们知道两个进程如果需要进行通讯最基本的一个前提能够唯一的标示一个进程，在本地进程通讯中我们可以使用PID来唯一标示一个进程，但PID只在本地唯一，网络中的两个进程PID冲突几率很大，这时候我们需要另辟它径了，我们知道IP层的ip地址可以唯一标示主机，而TCP层协议和端口号可以唯一标示主机的一个进程，这样我们可以利用ip地址 + 协议 + 端口号唯一标示网络中的一个进程。

能够唯一标示网络中的进程后，它们就可以利用socket进行通信了，什么是socket呢？我们经常把socket翻译为套接字，socket是在应用层和传输层之间的一个抽象层，它把TCP/IP层复杂的操作抽象为几个简单的接口供应用层调用已实现进程在网络中通信。



socket起源于UNIX，在Unix一切皆文件哲学的思想下，socket是一种"打开—读/写—关闭"模式的实现，服务器和客户端各自维护一个"文件"，在建立连接打开后，可以向自己文件写入内容供对方读取或者读取对方内容，通讯结束时关闭文件。