IP地址 端口号 字节序 套接字

一、IPV4

**什么是ip：**在因特网中用于区分主机的唯一标识

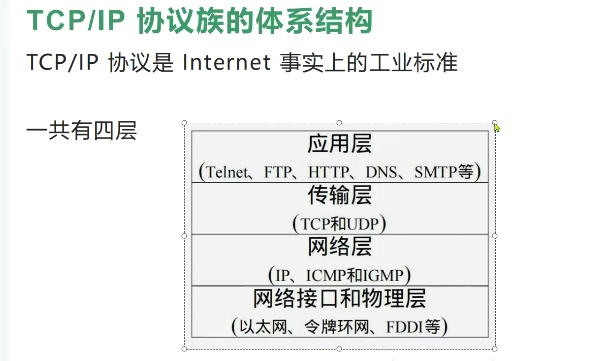
**表现形式：**点分形式的字符串 “a.b.c.d”，即192.168.16.58，每一位由0-255的数字组成

网络通信时，数据发送方、数据接收方需要明确的三大问题：

·协议

·ip

·端口



**ip地址分类：**

IP协议是为计算机网络相互连接进行通信而设计的协议

因特网上的每台计算机和其它设备都规定了一个唯一的地址，叫做“IP地址”。由于有这种唯一的地址，才保证了用户在连网的计算机上操作时，能够高效且方便地从千千万万台计算机中选出自己所需的对象来。

IP地址是一个32位的二进制数，通常分为4个“8位二进制数”（4字节）。IP地址通常用“点分十进制”表示成（a.b.c.d）的形式，其中，a,b,c,d都是0~255之间的十进制整数。

例：点分十进IP地址（100.4.5.6），实际上是32位二进制数（01100100.00000100.00000101.00000110）。

**公有地址**

公有地址（Public address）由Inter NIC（Internet Network Information Center因特网信息中心）负责。这些IP地址分配给注册并向Inter NIC提出申请的组织机构。通过它直接访问因特网。

**私有地址**

私有地址（Private address）属于非注册地址，专门为组织机构内部使用。

以下列出留用的内部私有地址

A类：10.0.0.0--10.255.255.255 网络ip 主机ip

B类：172.16.0.0--172.31.255.255

C类：192.168.0.0--192.168.255.255



**二、端口**

用于区分主机接收到的数据该交由哪一个进程来处理；

逻辑意义上的端口，一般是指TCP/IP协议中的端口，端口号的范围从0到65535；比如用于浏览网页服务的80端口，用于FTP服务的21端口等等。

1.端口号小于256的定义为常用端口，服务器一般都是通过常用端口号来识别的。

2.客户端只需保证该端口号在本机上是惟一的就可以了。客户端口号因存在时间很短暂又被称为临时端口号；

3.大多数TCP/IP实现给临时端口号进行分配1024—49151之间的端口号。大于5000的端口号是为其他服务器预留的。

我们应该在自定义端口时，要避免使用经常用到的端口，如：80、21等。

**三、字节序**

TCP/IP协议规定，网络数据流应采用大端字节序，即低地址高字节。

例如上一节的UDP段格式，地址0~1是16位的源端口号，如果这个端口号是1000（0x03e8），则地址0是0x03，地址1是0xe8，也就是先发0x03，再发0xe8，这16位在发送主机的缓冲区中也应该是低地址存0x03，高地址存0xe8。

但是，如果发送主机是小端字节序，这16位被解释成0xe803，而不是1000。因此，发送主机把1000填到发送缓冲区之前需要做字节序的转换。同样地，接收主机如果是小端字节序，接到16位的源端口号也要做字节序的转换。如果主机是大端字节序的，发送和接收都不需要做转换。同理，32位的IP地址也要考虑网络字节序和主机字节序的问题。

字节序就是为了解决由于cpu差异化导致的整型数据存放的兼容问题。

为使网络程序具有可移植性，使同样的C代码在大端和小端计算机上编译后都能正常运行，可以调用以下库函数做网络字节序和主机字节序的转换。

#include <arpa/inet.h>

uint32\_t htonl(uint32\_t hostlong); host主机to网络net，变为长整型

uint16\_t htons(uint16\_t hostshort);host主机to网络net，变为短整型

uint32\_t ntohl(uint32\_t netlong);net网络to主机host，变为长整型

uint16\_t ntohs(uint16\_t netshort);net网络to主机host，变为短整型

h：host，n：network，l：32位长整数long，s：16位短整数short。

如果主机是小端字节序，这些函数将参数做相应的大小端转换然后返回，如果主机是大端字节序，这些函数不做转换，将参数原封不动地返回。

**四、套接字（socket）**

**套接字（socket）：**是系统内核中的一种数据结构，也是一种进程间通信的方式，适用于不同主机进程通信；

**套接字有两种格式：**

半相关描述的套接字格式为：{协议，本地IP，本地端口}；

完全相关描述的套接字格式为：{协议、本地IP，本地端口，对方IP，对方端口}。

用于同一主机的进程通信：

传统三种：有名、无名管道、信号

IPC 三种：共享内存、消息队列、信号灯

套接字本身具有“插座”的意思，独立于具体协议的网络编程接口；

**在OSI模型中，**主要位于会话层(高层)和传输层(低层)之间

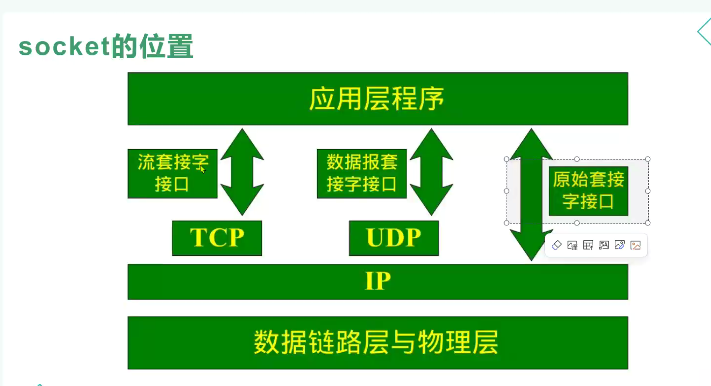
**BSD Socket（伯克利套接字）**是通过标准的UNIX文件描述符和其它程序通讯的一个方法，目前已经被广泛移植到各个平台。

文件描述符：

非负整数（0-1023）：0 1 2

用于内核打开正在访问的文件

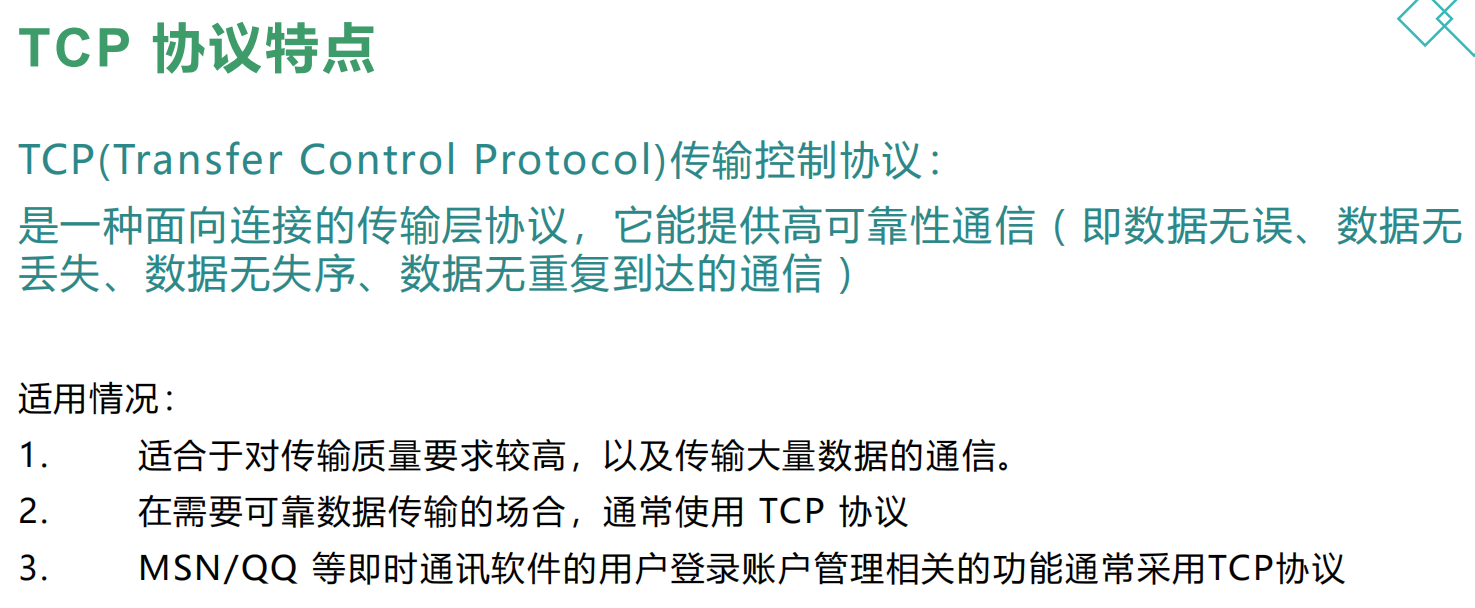
**套接字(socket)类型：**



**1、流式套接字(SOCK\_STREAM)**

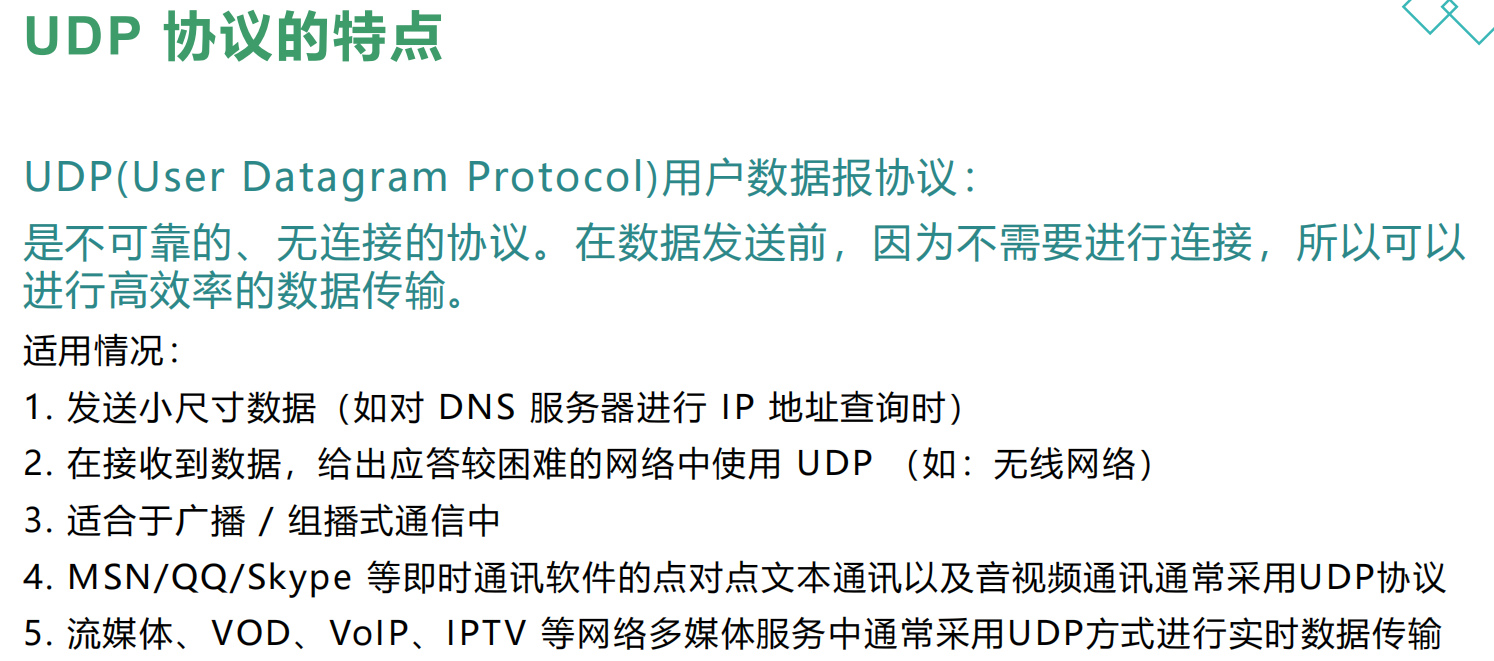
用于TCP通信，提供了一个面向连接、可靠的数据传输服务，数据无差错、无重复的发送且按发送顺序接收。

内设置流量控制，避免数据流淹没慢的接收方。数据被看作是字节流，无长度限制。



**2、数据报套接字(SOCK\_DGRAM)**

用于UDP通信，提供无连接服务。数据包以独立数据包的形式被发送，不提供无差错保证，数据可能丢失或重复，顺序发送，可能乱序接收。



**3、原始套接字(SOCK\_RAW)**

用于新网络协议的测试，可以对较低层次协议如IP、ICMP直接访问。

**版权声明：**本文为CSDN博主「এ᭄星辰」的原创文章，遵循CC 4.0 BY-SA版权协议，转载请附上原文出处链接及本声明。

**原文链接：**<https://blog.csdn.net/qq_63626307/article/details/127519646>