1. 在我们用Socket开发网络应用程序中，还有一个就是端点的概念，在网络中，通过IP地址，协议和端口号可以唯一地确定网络上的一个应用程序，其中把IP地址和端口的组合叫做端点(EndPoint)。每个Socket需要绑定到一个端点上与其他端点进行通信。
2. // 获得本机的Ip地址，即127.0.0.1，这个是巡回地址

IPAddress localaddress =IPAddress.Loopback;

1. // 创建可以访问的断点，49155表示端口号，如果这里设置为0，表示使用一个由系统分配的空闲的端口号

IPEndPoint endpoint = new IPEndPoint(localaddress,49155);

1. Socket不负责应用层协议，仅仅负责传输层的协议。
2. 现在介绍下网络端口号(port)的概念，在同一个网络地址中，为了区分使用相同协议的不同应用程序，为不同的应用程序分配一个数字编号，我们把这个编号就成为网络端口号(就是区分同一个网络地址中不同的进程)。端口号是由一个两个字节的整数，所以取值范围为0~65535，这些端口号又分为三类：

第一类的范围是0~1023，称为众所周知的端口，这些端口号由特定的网络程序使用，例如，TCP协议使用80端口来完成Http协议的传输。

第二类的范围是1024~49151，称为登记端口，一般情况下不应该在程序中使用。

第三类的范围是49152~65535，称为私有端口， 这些端口可以由普通用户程序使用。

1. tcpClient()当使用这种不带任何参数的构造函数时，将使用本机默认的ip地址并将使用默认的通信端口号0。这样情况下，如果本机不止一个ip地址，将无法选择使用。以下语句示例了如何使用默认构造函数来创建新的 TcpClient：
2. 利用TCP传输数据时，数据是以字节流的形式进行传输，客户端与服务器端建立连接后，发送方需要先将发送的数据转换为字节流，然后将其发送给对方，发送数据时，可以通过程序不断地将数据流陆续写入TCP的发送缓冲中，然后TCP自动从发送缓冲中提取一定量的数据，将其组成TCP报文段发送到IP层，再通过IP层(也就是网络层)之下的网络接口发送出去；接受端从IP层接收到TCP报文段后，将其暂时保存在接受缓冲中，然后我们通过程序依次读取接受缓冲中的数据，从而达到相互通信的目的(简单的说就发送方把数据转换为数据流，再把数据流存储在发送缓冲中，然后传输层低层的协议从发送缓冲中读取数据把数据发送出去，然后接收端从底层接受到数据把数据存储在接收端的缓冲中，然后我们写的程序只是从缓冲中依次读取数据，然后显示出来，在客户端我们写代码做的事情是把数据写入Write写入发送端的缓冲中，然后服务器端(接收端)用Read方法在自己的缓冲中读取数据，用一句话概括，TCP的传输就是对数据的写——读操作)括号中的内容只是我个人理解，因为这样我感觉理解起来比较容易，对于刚开始接触TCP的朋友可以这样理解，然后再一句句话去扩展。
3. TCP协议的连接建立通过三次握手来完成的
4. UDP是一种简单、**面向数据报(Sock\_Dgram)**的**无连接**协议，提供的是不一定可靠的传输服务。
5. TCP是一种面向连接、可靠的，**面向字节流(Sock\_Stream)**的传输协议,对于“无连接”是指在正式通信前不必与对方先建立连接

(1)UDP的可靠性不如TCP，因为TCP传输前要首先建立连接，这样就增加了TCP传输的可靠性，所以UDP也被称为不可靠的传输协议，关于TCP的介绍可以看我上一篇博客的介绍。

(2)UDP不能保证有序传输。即UDP不能确保数据的发送和接收顺序。

1. 由于UDP进程在通信之前是不需要建立连接，消息接收方可能并不知道是谁给它发的消息，因此UDP编程分为两种模式：一种“实名发送”，即接收方可以由收到的消息得知发送方进程端口，另外一种则为“匿名发送”，即接收方并不知道发给它信息的远程进程究竟来自哪个端口。
2. 组播组又称为多路广播组，组播地址的范围在224.0.0.0到239.255.255.255的D类IP地址（至于这个概念大家可以百度百科里面就查看）。任何发送到组播地址的消息都会被发送到组内所有成员设备上，组可以使永久的也可以是临时，大多数我们使用的都是临时的，仅在有成员的时候才存在。
3. UDP匿名发送，会随机生成一个没有使用的端口
4. **P2P的基本原理**
   1. P2P应用程序一般分为**发现、连接和通信**3个阶段
   2. 一台计算机要和另外一台计算机通信，必须知道对方的IP地址和监听端口，否则就无法向对方发送消息。
   3. 完成对等节点的发现后，接下来就可以根据需要，选择TCP、UDP或者其他协议完成数据传输。如果选择TCP，则需要先建立连接，再利用该连接传输数据，关于TCP的内容可以查看我之前的专题；如果选择UDP，则无须建立连接，直接在对等节点之间通信就可以了。