

使用JavaJDK中Java.net包控制UDP协议 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/267/2021\\_2022\\_\\_E4\\_BD\\_BF\\_E7\\_94\\_A8Java\\_c104\\_267208.htm](https://www.100test.com/kao_ti2020/267/2021_2022__E4_BD_BF_E7_94_A8Java_c104_267208.htm) 什么是UDP协议 UDP协议的全称是用户数据报，在网络中它与TCP协议一样用于处理数据包。在OSI模型中，在第四层传输层，处于IP协议的上一层。UDP有不提供数据报分组、组装和不能对数据包的排序的缺点，也就是说，当报文发送之后，是无法得知其是否安全完整到达的。为什么要使用UDP 在选择使用协议的时候，选择UDP必须要谨慎。在网络质量令人不十分满意的环境下，UDP协议数据包丢失会比较严重。但是由于UDP的特性：它不属于连接型协议，因而具有资源消耗小，处理速度快的优点，所以通常音频、视频和普通数据在传送时使用UDP较多，因为它们即使偶尔丢失一两个数据包，也不会对接收结果产生太大影响。比如我们聊天用的ICQ和OICQ就是使用的UDP协议。在Java中操纵UDP 使用位于JDK中Java.net包下的DatagramSocket和DatagramPacket类，可以非常方便地控制用户数据报文。在描述它们之前，必须了解位于同一个位置的InetAddress类。InetAddress实现了Java.io.Serializable接口，不允许继承。它用于描述和包装一个Internet IP地址，通过三个方法返回InetAddress实例：`getLocalhost()`：返回封装本地地址的实例。`getAllByName(String host)`：返回封装Host地址的InetAddress实例数组。`getByName(String host)`：返回一个封装Host地址的实例。其中，Host可以是域名或者是一个合法的IP地址。DatagramSocket类用于创建接收和发送UDP的Socket实例。和Socket类依赖SocketImpl类一样

，DatagramSocket类的实现也依靠专门为它设计的DatagramSocketImplFactory类。DatagramSocket类有3个构建器：DatagramSocket()：创建实例。这是个比较特殊的用法，通常用于客户端编程，它并没有特定监听的端口，仅仅使用一个临时的。DatagramSocket(int port)：创建实例，并固定监听Port端口的报文。DatagramSocket(int port, InetAddress localAddr)：这是个非常有用的构建器，当一台机器拥有多于一个IP地址的时候，由它创建的实例仅仅接收来自LocalAddr的报文。值得注意的是，在创建DatagramSocket类实例时，如果端口已经被使用，会产生一个SocketException的异常抛出，并导致程序非法终止，这个异常应该注意捕获。

DatagramSocket类最主要的方法有4个：

Receive(DatagramPacket d)：接收数据报文到d中。receive方法产生一个“阻塞”。Send(DatagramPacket d)：发送报文d到目的地。SetSoTimeout(int timeout)：设置超时时间，单位为毫秒。Close()：关闭DatagramSocket。在应用程序退出的时候，通常会主动释放资源，关闭Socket，但是由于异常地退出可能造成资源无法回收。所以，应该在程序完成时，主动使用此方法关闭Socket，或在捕获到异常抛出后关闭Socket。“阻塞”是一个专业名词，它会产生一个内部循环，使程序暂停在这个地方，直到一个条件触发。

100Test 下载频道开通，各类考试题目直接下载。详细请访问 [www.100test.com](http://www.100test.com)