数据链路层 网络层 传输层 应用层

TCP UDP

TCP与UDP基本区别

1.基于连接与无连接

2.TCP要求系统资源较多，UDP较少；

3.UDP程序结构较简单

4.流模式（TCP）与数据报模式(UDP);

5.TCP保证数据正确性，UDP可能丢包

6.TCP保证数据顺序，UDP不保证

UDP应用场景：

1.面向数据报方式

2.网络数据大多为短消息

3.拥有大量Client

4.对数据安全性无特殊要求

5.网络负担非常重，但对响应速度要求高

具体编程时的区别

1.socket()的参数不同

　　 2.UDP Server不需要调用listen和accept

　　 3.UDP收发数据用sendto/recvfrom函数

　　 4.TCP：地址信息在connect/accept时确定

　　 5.UDP：在sendto/recvfrom函数中每次均 需指定地址信息

　　 6.UDP：shutdown函数无效

编程区别

通常我们在说到网络编程时默认是指TCP编程，即用前面提到的socket函数创建一个socket用于TCP通讯，函数参数我们通常填为SOCK\_STREAM。即socket(PF\_INET, SOCK\_STREAM, 0)，这表示建立一个socket用于流式网络通讯。

　 SOCK\_STREAM这种的特点是面向连接的，即每次收发数据之前必须通过connect建立连接，也是双向的，即任何一方都可以收发数据，协议本身提供了一些保障机制保证它是可靠的、有序的，即每个包按照发送的顺序到达接收方。

　　而SOCK\_DGRAM这种是User Datagram Protocol协议的网络通讯，它是无连接的，不可靠的，因为通讯双方发送数据后不知道对方是否已经收到数据，是否正常收到数据。任何一方建立一个socket以后就可以用sendto发送数据，也可以用recvfrom接收数据。根本不关心对方是否存在，是否发送了数据。它的特点是通讯速度比较快。大家都知道TCP是要经过三次握手的，而UDP没有。

基于上述不同，UDP和TCP编程步骤也有些不同，如下：

TCP:

TCP编程的服务器端一般步骤是：

　　1、创建一个socket，用函数socket()；

　　2、设置socket属性，用函数setsockopt(); \* 可选

　　3、绑定IP地址、端口等信息到socket上，用函数bind();

　　4、开启监听，用函数listen()；

　　5、接收客户端上来的连接，用函数accept()；

　　6、收发数据，用函数send()和recv()，或者read()和write();

　　7、关闭网络连接；

　　8、关闭监听；

TCP编程的客户端一般步骤是：

　　1、创建一个socket，用函数socket()；

　　2、设置socket属性，用函数setsockopt();\* 可选

　　3、绑定IP地址、端口等信息到socket上，用函数bind();\* 可选

　　4、设置要连接的对方的IP地址和端口等属性；

　　5、连接服务器，用函数connect()；

　　6、收发数据，用函数send()和recv()，或者read()和write();

　　7、关闭网络连接；

UDP:

与之对应的UDP编程步骤要简单许多，分别如下：

　　UDP编程的服务器端一般步骤是：

　　1、创建一个socket，用函数socket()；

　　2、设置socket属性，用函数setsockopt();\* 可选

　　3、绑定IP地址、端口等信息到socket上，用函数bind();

　　4、循环接收数据，用函数recvfrom();

　　5、关闭网络连接；

UDP编程的客户端一般步骤是：

　　1、创建一个socket，用函数socket()；

　　2、设置socket属性，用函数setsockopt();\* 可选

　　3、绑定IP地址、端口等信息到socket上，用函数bind();\* 可选

　　4、设置对方的IP地址和端口等属性;

　　5、发送数据，用函数sendto();

　　6、关闭网络连接；

TCP和UDP是OSI模型中的运输层中的协议。TCP提供可靠的通信传输，而UDP则常被用于让广播和细节控制交给应用的通信传输。

UDP补充：

UDP不提供复杂的控制机制，利用IP提供面向无连接的通信服务。并且它是将应用程序发来的数据在收到的那一刻，立刻按照原样发送到网络上的一种机制。即使是出现网络拥堵的情况下，UDP也无法进行流量控制等避免网络拥塞的行为。此外，传输途中如果出现了丢包，UDO也不负责重发。甚至当出现包的到达顺序乱掉时也没有纠正的功能。如果需要这些细节控制，那么不得不交给由采用UDO的应用程序去处理。换句话说，UDP将部分控制转移到应用程序去处理，自己却只提供作为传输层协议的最基本功能。UDP有点类似于用户说什么听什么的机制，但是需要用户充分考虑好上层协议类型并制作相应的应用程序。

TCP补充：

TCP充分实现了数据传输时各种控制功能，可以进行丢包的重发控制，还可以对次序乱掉的分包进行顺序控制。而这些在UDP中都没有。此外，TCP作为一种面向有连接的协议，只有在确认通信对端存在时才会发送数据，从而可以控制通信流量的浪费。TCP通过检验和、序列号、确认应答、重发控制、连接管理以及窗口控制等机制实现可靠性传输。

TCP与UDP区别总结：

1、TCP面向连接（如打电话要先拨号建立连接）;UDP是无连接的，即发送数据之前不需要建立连接

2、TCP提供可靠的服务。也就是说，通过TCP连接传送的数据，无差错，不丢失，不重复，且按序到达;UDP尽最大努力交付，即不保 证可靠交付

3、TCP面向字节流，实际上是TCP把数据看成一连串无结构的字节流;UDP是面向报文的

UDP没有拥塞控制，因此网络出现拥塞不会使源主机的发送速率降低（对实时应用很有用，如IP电话，实时视频会议等）

4、每一条TCP连接只能是点到点的;UDP支持一对一，一对多，多对一和多对多的交互通信

5、TCP首部开销20字节;UDP的首部开销小，只有8个字节

6、TCP的逻辑通信信道是全双工的可靠信道，UDP则是不可靠信道