# 网络编程

1. 什么叫网络编程

答：网络编程从大的方面说就是对信息的发送到接收，中间传输为物理线路的作用。

网络编程即使用套接字（socket）来达到各进程间的通信，现在一般称为TCP/IP编程；

网络编程分为服务端和客户端。服务端就相当于我们平时所说的服务器，有固定的IP地址，随时等待连接并做出响应；而客户端相当于各种端系统，找到服务端进行连接获取服务。

1. 什么时网络通信协议？

计算机网络中实现通信必须有一些约定，即通信协议；包括对速率、传输代码、代码结构、传输控制步骤、出错控制等制定的标准。

1. 网络OSI七层模型偶有那些？TCP属于哪一层？

|  |  |
| --- | --- |
| 应用层 |  |
| 表示层 |  |
| 会话层 |  |
| 传输层 |  |
| 网络层 |  |
| 数据链路层 |  |
| 物理层 |  |

TCP协议属于传输层。

1. 网络编程提供了哪些协议？

Java的网络编程提供了两种通信协议：TCP（传输控制协议）和UDP（数据报协议）

1. TCP和UDP的区别是什么？

回答方式一：

共同点：UDP协议与[TCP](https://baike.baidu.com/item/TCP" \t "https://baike.baidu.com/item/UDP/_blank)协议一样用于处理数据包。TCP和UDP都属于传输层协议。

不同点：TCP协议是可靠的传输协议，传输前会采用“三方握手”的方式后建立连接，以保证数据的可靠性。而UDP协议是无连接、不可靠的传输协议，即发出去的数据不一定接收得到，及时通讯工具一般使用这种协议。

关系：相互补充。

实例：UDP适用于多媒体数据流，如：视频、音频等，正因为UDP协议的控制选项较少，在数据传输过程中延迟小、数据传输效率高，适合对可靠性要求不高的应用程序，或者可以保障可靠性的应用程序，如DNS、TFTP、SNMP等。

回答方式二：

连接方面

tcp面向连接，udp不需要连接

tcp需要三次握手四次挥手请求连接

可靠性

tcp是可靠传输；一旦传输过程中丢包的话会进行重传

udp是不可靠传输，但会最大努力交付

工作效率

udp实时性高，比tcp工作效率高

因为不需要建立连接，更不需要复杂的握手挥手以及复杂的算法，也没有重传机制

是否支持多对多

tcp是点对点的

udp支持一对一，一对多，多对多

首部大小

tcp首部占20字节

udp首部占8字节

1. TCP程序设计

在Java中使用Socket（套接字）完成TCP程序的开发。服务器端使用ServerSocket等待客户端连接（accept后进入阻塞状态），客户端使用Socket对象。

1. UDP程序设计

使用UDP发送数据，对方不一定收到，所以一般情况下，需要客户端始终等待服务器端发送过来的消息。

1. get和post的区别

冪等/不冪等（可缓存/不可缓存）

get请求是冪等的，所以get请求的数据是可以缓存的

而post请求是不冪等的，查询查询对数据是有副作用的，是不可缓存的

传参

get传参，参数是在url中的

准确的说get传参也可以放到body中，只不过不推荐使用

post传参，参数是在请求体中

准确的说post传参也可以放到url中，只不过不推荐使用

安全性

get较不安全

post较为安全

准确的说两者都不安全，都是明文传输的，在路过公网的时候都会被访问到，不管是 url还是header还是body，都会被访问到，要想做到安全，就需要使用https

参数长度

get参数长度有限，是较小的

准确来说，get在url传参的时候是很小的

post传参长度不受限制

发送数据

post传参发送两个请求包，一个是请求头，一个是请求体，请求头发送后服务器进行验 证，要是验证通过的话就会给客户端发送一个100-continue的状态码，然后就会发送请 求体

字符编码

get在url上传输的时候只允许ASCII编码

1. 描述一下3次握手的过程

客户端和服务端之间通过三次握手建立连接，四次挥手释放连接。

三次握手：客户端先向服务端发起一个SYN包，进入SYN\_SENT状态，服务端收到SYN后，给客户端返回一个ACK+SYN包，表示已收到SYN，并进入SYN\_RECEIVE状态，最后客户端再向服务端发送一个ACK包表示确认，双方进入establish状态。

说明：之所以是三次握手而不是两次，是因为如果只有两次，在服务端收到SYN后，向客户端返回一个ACK确认就进入establish状态，万一这个请求中间遇到网络情况而没有传给客户端，客户端一直是等待状态，后面服务端发送的信息客户端也接受不到了。

四次挥手：首先客户端向服务端发送一个FIN包，进入FIN\_WAIT1状态，服务端收到后，向客户端发送ACK确认包，进入CLOSE\_WAIT状态，然后客户端收到ACK包后进入FIN\_WAIT2状态，然后服务端再把自己剩余没传完的数据发送给客户端，发送完毕后在发送一个FIN+ACK包，进入LAST\_ACK（最后确认）状态，客户端收到FIN+ACK包后，再向服务端发送ACK包，在等待两个周期后在关闭连接

说明：之所以等待两个周期是因为最后服务端发送的ACK包可能会丢失，如果不等待2个周期的话，服务端在没收收到ACK包之前，会不停的重复发送FIN包而不关闭，所以得等待两个周期

腾讯面试问题：

1. 腾讯的及时通讯模块用的什么协议或原理？
2. 既然UDP不可靠，为什么还要使用？为什么不用TCP协议？