# 网络相关 面试积累

## TCP

**请简单解释一下TCP三次握手**

客户端发送带有SYN标志的包到服务端（一次握手）

服务端发送带有SYN和ACK标志的包到客户端（二次握手）【服务端确认自己接受和对方发送正常。所以需要第三次挥手】

客户端发送带有ACK标志的包到服务端（三次握手）【服务端确认自己发送对方接受正常】

**为什么要三次握手**

三次握手为了建立可靠的通信信道。双方确认**双方**的**发**与**收**是正常的。

**为什么要传回 SYN**

这是为了告诉客户端，我接收的信息就是你发送的信息。

**传了 SYN，为啥还要传 ACK**

ACK是用于验证自己发送和对方的接收正常。

**请简单解释一下TCP四次挥手**

客户端发送一个FIN，用来关闭客户端到服务器的数据传送。【客户端发出通知：我想关闭通信】

服务器收到发回一个ACK。确认序号为收到的FIN加1。【服务端：我知道啦，等我发完剩下数据就关闭。，此时客户端处于半关闭状态。】

服务器关闭与客户端的连接，发送一个FIN给客户端。【服务端：我可以关闭了】

客户端发回ack报文确认。【客户端：知道了。 此时真正关闭通信。】

**为什么要四次挥手。**

因为任何一方都可以在数据传送结束之后发出连续释放的通知，待对方确认后进入半关闭状态，当另一方也没有数据在发送的时候则发出连续释放通知，通知对方确认之后，就完全关闭了这个TCP的连接。

**TCP是如何保证可靠传输的**

1. 应用数据被分割成TCP认为最适合发送的数据块。
2. TCP给发送的每一个包进行编号，接收方对数据包进行排序，把有序数据传送给应用层。
3. **校验和**：TCP将保持它首部和数据的校验和，这是一个端到端的校验和，目的是检测数据在传输过程中的任何变化，如果收到端的检验和有差错，TCP将丢弃这个报文段和不接收不确认收到此报文段。
4. TCP的接收端会丢弃重复的数据。
5. **流量控制：**TCP连接的每一方都有固定大小的缓冲空间，GDP的接收端只允许发送端发送接收端缓冲区能接纳的数据当接收方来不及处理，发行方的数据能提示发送方降低发送的速率，防止包丢失，TCP使用的流量控制协议是可变大小的滑动窗口协议。 TCP利用滑动窗口实现流量控制。
6. **拥塞控制：**当网络拥塞时减少数据的发送。
7. **ARQ协议**
8. **超时重传：**当TCP发出一个段后，它启动一个定时器，等待目的端确认收到这个报文端，如果不能及时收到一个确认，将重发这个报文段。

**请谈谈ARQ协议**

A RQ协议是自动重传请求，他通过使用**确认**和**超时**这两个机制在不可靠服务的基础上实现可靠的信息传输，如果发送方在发送后一段时间内没有收到确认帧，他通常会重新发送。ARQ包括停止等待ARQ协议和连续ARQ协议。

**停止等待ARQ协议**

原理是每发完一个分组就停止发送，等待对方确认，如果过了一段时间还是没有收到ack确认说明没有发送成功，需要重新发送，直到收到确认后再发下一个分组。在停止等待协议中，如果接收方收到重复分组就丢弃该分组，但同时还要发送确认。

比较简单，但是信道利用率低，等待时间很长。

**连续ARQ协议**

发送方维持一个发送窗口，凡为发送窗口内的分组可以连续发送，而不需要等待对方确认，接收方一般采用累计确认对按需到达的最后一个分组发送确认表明这个分组为止，所有的分组都已经正确收到了。

信道利用率高容易实现，即使确认丢失，也不必重传，不能向发送方反映出接收方已经正确收到的所有分组信息。比如发送方发了5条信息，中间第3条确实这是接收方，只能对前两条发送确认发送方无法知道后三个分组的下落，只好把后三个全部重传，这也叫回退N表示需要退回来重传已经发送过的N个消息。

**请谈谈滑动窗口和流量控制**

TCP利用滑动窗口实现流量控制。流量控制是为了控制发送方发送速率，保证接收方来得及接收。接收方发送确认报文中的窗口的字段，可以用来控制发送方窗口大小，从而影响发送方的发送速率。窗口字段设置为0就不能发数据了。

**请谈谈拥塞控制及具体算法**

在某一时段内网络中某一资源的需求超过了该资源所能提供的能力，这个网络就回拥塞，拥塞控制是一个全局的过程，涉及到所有的主机，所有的路由器，以及降低网络传输线路有关的所有因素。

为了进行拥塞控制，TCP发送方要维持一个拥塞窗口的状态变量，拥塞窗口的大小取决于网络的拥塞程度，并且动态变化，发送方让自己的发送窗口取为拥塞窗口和接收方的接受窗口中较小的一个。

TCP的拥塞控制采用4种算法，慢开始，拥塞避免，快重传和快恢复。

**慢开始**：发送方发送数据的量是从小到大的。一开始发送小数据量。

**拥塞避免**：让拥塞窗口缓慢增大。拥塞窗口初始值为1，每经过一个传播轮次加倍。

**快重传与快恢复**：没有快充传与快恢复的话，如果数据报丢失了，TCP会使用定时器来要求传输暂停。有了它，如果接收端接收到一个不按顺序的数据端，它会立即发送给发送端一个重复确认，如果发送端接收到三个重复确认，它会假定确认指出的数据段丢失了并立即重传这些丢失的数据段，这样的话就不会因为重传时要求的暂停被耽误了，当有单独的数据包丢失时，快重传和恢复能最有效的工作，当有多个数据信息包在某一段很短的时间内丢失时，他都不能很有效的工作。

**TCP和UDP的区别？**

UDP不需要提前建立连接，是一种最有效的工作方式，应用在比如QQ语音，QQ视频直播等等。

TCP是提供面向连接的服务，需要先连接，结束之后要释放连接。一般用于文件传输，发送和接收邮件，远程登录等场景。

## HTTP

http是超文本传输协议。最初的目的是为了提供一种发布和接受HTML页面的方法。是万维网上的文件都要遵循的标准。

**HTTP协议组成有哪几部分？**

请求和响应

**HTTP请求报文与响应报文格式**

请求报文三部分：

请求行：请求方法，URI，HTTP版本信息。

请求首部

请求体

响应报文三部分：

状态行：HTTP版本，状态码，状态码的原因短语。

响应首部

响应体

**常见的HTTP相应状态码**

1xx：请求已接收，继续处理。

2xx：成功，请求已成功处理

3xx：重定向

4xx：客户端错误：请求有语法错误，请求无法实现。

5xx：服务器错误

200 请求被正常处理。

204 请求被受理但是没有资源可以返回。

206 客户端只请求资源的一部分。

301 永久重定向。

302 临时重定向。

304 发送附带条件的请求时，条件不满足时返回。与重定向无关。

307 临时重定向，与302相似，但强制使用post方法。

400 请求报文语法错误，服务端无法识别。

401 请求需要认证。

403 请求的资源被禁止访问。

404 服务器无法找到对应资源。

500 服务器内部错误。

503 服务器正忙。

**GET和POST区别**

1. get重点在从服务器获取资源，post重点在向服务器发送数据。
2. get传输数据通过URL请求，以field=value的形式，置于URL后，并用？连接，多个请求之间用&连接。 过程是用户可见的。

post传输是通过HTTP的post机制，将资源与对应值封存在请求体重发给服务器。是用户不可见的。

1. Get传输数据量小，因为受URL长度限制，但效率高。

Post可以传输大量数据，所以上传文件的时候只能用Post方法。

1. get不安全，URL可见，可能会泄露信息。

post更安全。

1. get方式只支持ASCII字符，向服务器传中文可能会造成乱码。

Post支持标准字符集，可以正确传递中文字符。

**HTTP协议与HTTPS协议的区别**

1. 端口：HTTP的URL由http:// 开始，并默认使用80端口，

HTTPS的URL 由https:// 开始，默认使用443端口。

1. 安全性和资源消耗

HTTP协议运行在TCP之上，明文传输，客户端和服务端都无法验证对方的身份。

HTTPS协议运行在SSL/TLS上，SSL/TLS运行在TCP上。 所有传输的内容都经过加密。加密采用对称加密，但是对称加密的秘钥用服务器方的证书进行非对称加密传输

**对称加密和非对称加密？**

对称加密：密钥只有一个，加密解密用同一个密码。加密解密速度快。

非对称加密：密钥成对出现，分为公钥和私钥，（两种钥匙无法互相推知） 加密解密使用不同的秘钥。（私钥加密公钥解密，公钥加密，私钥解密），相对对称加密速度慢。

**HTTPS协议是如何工作的**

1. **客户端发送HTTPS请求**用户在浏览器输入https网址，连接到Server的443端口。
2. **服务端的配置**： 服务器必须有一套数字证书，可以自己制作也可以向组织申请。自己颁发的证书需要客户端验证通过才可以继续访问。而使用受信任的公司申请的证书不会弹出提示页面。这套证书其实就是一对公钥和私钥。
3. **传送证书**：
4. **客户端解析证书**：客户端的TLS来完成，首先会验证公钥是否有效。如果发现异常，就会弹出警告框，提示证书有问题。如果证书没有问题，就生成一个随机值，然后用证书对该随机值进行加密。除非有私钥，不然看不到这个随机值。
5. **传送加密随机值**： 传送用证书加密的随机值，服务端得到这个随机值，这就是对称加密的秘钥进行加密解密。
6. **服务端解密随机值**。
7. **传输加密信息。**
8. **客户端解密。**

**HTTP是不保存状态的协议,如何保存用户状态?**

Http是无状态协议。可以通过Session，通过服务端记录用户的状态。（购物车，给特定用户创建特定的Session来标识用户并跟踪。）

在服务端保存Session的方法很多，最常用的是内存和数据库（比如：内存数据库Redis）Session跟踪实现方式：在Cookie中附加一个Session ID 来跟踪。

## Cookie 与 Session

**Cookie的作用是什么?和Session有什么区别？**

都是跟踪浏览器用户身份的会话方式。

Cookie一般在浏览器保存用户信息，session主要通过服务端记录用户的状态。

Cookie存放为一个token在cookie中，下次登录就可以根据这个token值来查找用户了，登录一次网站后访问网站的其他页面就不需要重新登录。

Session的典型场景就是购物车，添加商品的购物车的时候系统不知道是哪个用户操作的，因为HTTP协议是无状态的，服务端给特定的用户创建特定的section之后就可以标识这个用户并跟踪了。

**Cookie被禁用了咋整？**

利用URL 重写，把Session ID 直接附加在URL路径后面。

**Session的实现机制是什么样子的，分布式环境下有什么注意事项。**

Session通过将session ID保存在cookie中，cookie中叫（jsessionId）每一次会话时读取cookie来实现session的保持。

当浏览器不支持cookie或者是用户阻止了cookie的时候，可以把在省会话ID附在页面中的所有URL上，这些页面作为响应发送给客户，这样当用户单击URL时，会话ID被自动作为请求行的一部分，而不是作为请求头发送给服务器，这种方法称为URL重写。

实际项目的tomcat一般都是集群的，客户的请求可能会分配到不同的tomcat上，因此如果使用session的话要考虑session共享的问题。最好可以用分布式缓存，比如redis来替换session，这样信息存储放在redis中就不需要考虑section共享的问题了。

**如何注销Session**

当用户关闭浏览器或者长时间不访问超时，或者服务器调用了HttpSession的invalidate（）方法后，Session 失效。

**设置Session的时长如何操作，默认时长是多少呢？**

tomcat中session的默认过期时间是20分钟，可以在web.xml中显示声明过期时间(一般不会修改这个)，如下：

<session-config>

<session-timeout>300</session-timeout> <!-- 单位为分钟 -->

</session-config>

**HTTP 1.0和HTTP 1.1的主要区别是什么?**

1. **长连接**：
   1. 在HTTP1.0的时候，默认使用短连接，每一次请求都要重新建立一次连接。 HTTP是基于TC pip协议的，每一次建立或者断开连接都需要三次握手，4次挥手的开销，如果每次请求都这样的话，开销会比较大。
   2. HTTp1.1起默认使用长连接，默认开启。connection: keep-alive
      1. 持续连接有非流水线方式和流水线方式，流水线方式是客户在收到响应报文之前就能接着发送新的请求报文，
      2. 非流水线方式是客户在收到前一个响应后才能发送下一个请求。
2. **错误状态响应码**:1.1中新增了24个状态码。409请求的资源与当前状态发生冲突。410 服务器上的某个资源被永久删除。
3. **缓存处理**：策略1.1的更多一些。
4. **带宽优化及网络连接的使用**： 1.0不支持断点续传，可能客户端只需要某个对象的一部分，服务器却将整个对象都传过来了。1.1 在请求头引入了range头域，允许只请求资源的某个部分，返回码是206（partial Content） 可以方便开发者充分利用带宽和连接。

**各种协议与HTTP协议之间的关系**

* 客户端 请求一个 URL ， DNS 返回IP地址
* HTTP的职责，申城针对目标Web服务器的HTTP请求报文。
* TCP 为了方便通信将HTTP请求报文分割成报文段，按序号分为报文段，把每个报文段可靠地传给对方。
* IP的职责：搜索对方的地址，一边中转一边传送。
* ===========================================
* TCP协议：从对方那里接收到报文段，重组报文，按序号以原来的顺序重组请求报文。
* HTTP：对Web服务器请求的内容进行处理，找到服务器计算机的具体资源
* 请求的处理结果同样利用TCP IP通信协议向用户回传。

OSI与TCP/IP各层的结构与功能,都有哪些协议?

OSI是七层协议 应用层，表示层，会话层，运输层，网络层，数据链路层，物理层。

TCP/IP是四层协议 应用层，运输层，网际层，网络接口层

五层协议：应用层，运输层，网络层，数据链路层，物理层。

应用层：进程之间的交互来完成特定网络应用。交互的数据单元称为报文。

协议：域名解析系统DNS，HTTP协议，电子邮件的 SMTP协议。

运输层：向两台主机进程之间的通信提供通用的数据传输服务。

协议：传输控制协议TCP，面向连接的可靠服务。

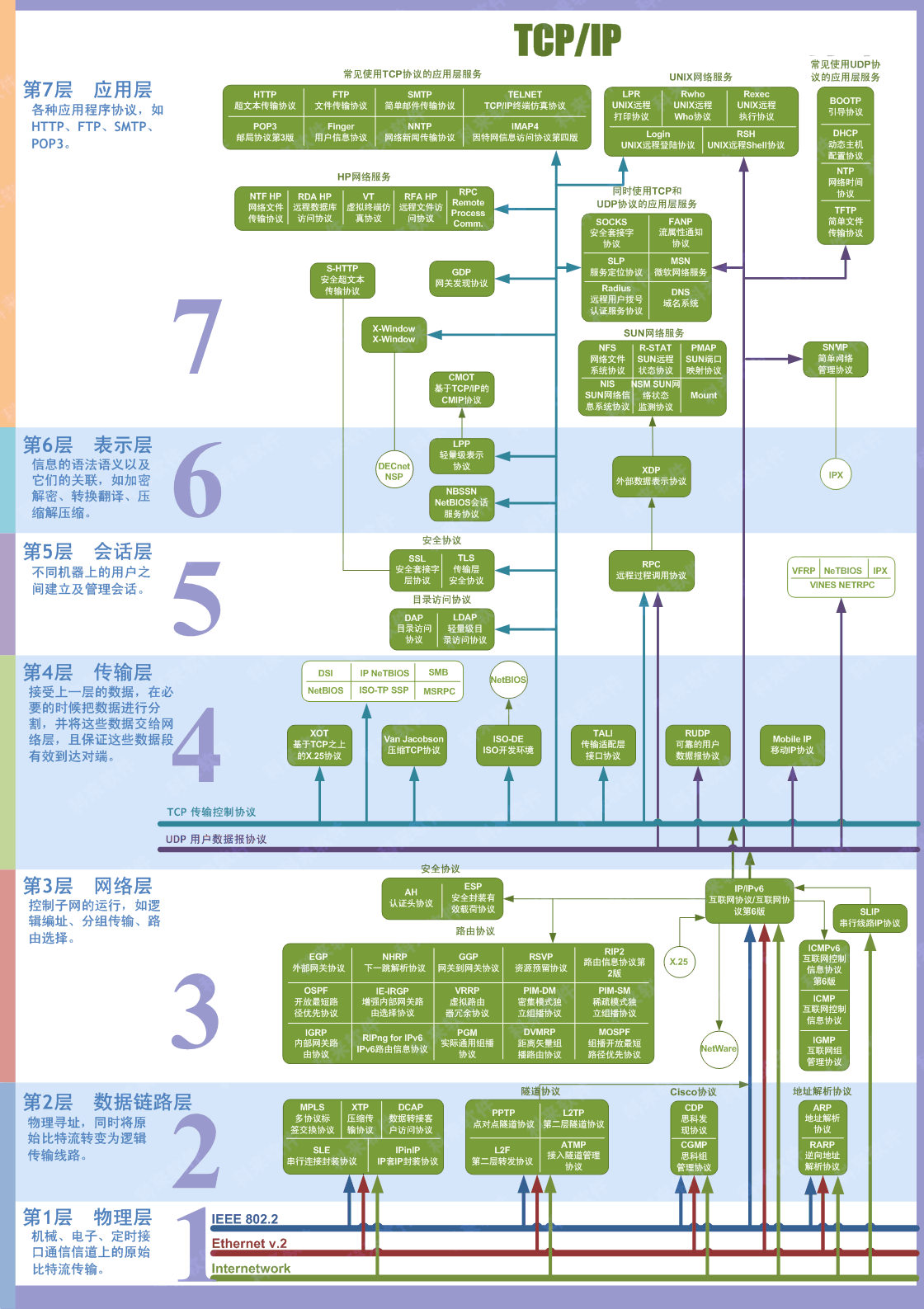
用户数据协议UDP，无连接的，不保证可靠性的服务。

网络层（网际层，IP层）：选择合适的路由和交换节点，确保数据的及时传送。

协议：IP

数据链路层：链路传输的协议，将网络层交下来的IP数据报组装成帧，两个相邻节点的链路上传送帧。

物理层：数据单位是比特。实现计算机之间的透明传输，尽可能屏蔽传输介质的物理设备差异。



## Java Web 基础

Servlet,Filter和Listener分别是什么，用在什么地方，jsp页面如何进行处理呢

请求转发、URL重定向和包含有什么区别，如何实现呢

如何判断远程机器上某个端口是否开启呢？项目中需要查看域名在本地的解析IP，如何操作呢？

Servlet中，调用jsp展示元素和返回String（即api，一般是json数据）有什么区别呢？

nginx+tomcat模式下，服务器段如何获取客户端请求IP呢？

servlet的生命周期是什么？

Servlet是否是线程安全的呢？