1. 网络为什么要分层

网络的分层和我们后端开发的时候分层都是一样的目的，分层之后将代码模块化，每一层只需要专注做一类事情就可以了。更易于代码的维护，扩展还有复用。

1. TCP/IP 4层模型

4层模型就是把7层网络模型中的应用层，表示层，会话层统一成了应用层，传输层，网络层和7层网络模型中的数据链路层和物理层统一成了数据链路层。

1. **OSI7层网络体系结构与TCP/IP协议模型**

**7层网络结构包括应用层，表示层，会话层，传输层，网络层，数据链路层，物理层。**

**应用层**：应用层就是处理用户的业务逻辑的，比如点击发送邮件，打开网页等，都会进入应用层协议的处理。常见的应用层协议就是HTTP协议，负责浏览器和服务器的通信还有FTP协议，负责文件的传输，还有SMTP，邮件的传输协议。

**表示层**：表示层负责数据的编码。使发送端和接收端有统一的编码格式。

会话层：应用程序和网络之间的接口，负责建立，维持和终止通信

传输层：传输层保证数据的可靠传输。常见的就是TCP/UDP。

网络层：网络层负责两个主机之间，传输线路的连接以及选择。

数据链路层和物理层就是具体的数据在网络之间的传输了。

1. **讲一下对称加密算法和非对称加密算法？**

**对称加密算法是指加密和解密使用同一个方式，这种加密算法最大的问题使不能保证传输的可靠，可能在传输过程中被盗取信息，密码就随之被破解了。**

**非对称加密算法是指一对非对称的密钥，分为公钥和私钥，公钥是所有人都知道的，而私钥只有自己知道，我们发送的时候就发送公钥，然后通过私钥解密。这样就解决了对称加密算法的传输不可靠的问题了。**

1. HTTP是哪一层的协议？

应用层。

1. http常见的状态码

首先是1开头的1xx:请求正在处理当中

然后是2开头的2xx:2开头的表示正常的处理完毕了。比如常见的200，还有204，204表示处理成功了，但是不返回任何内容。

然后是3开头的，3开头的表示重定向，301是永久的移动，即当前的URI已经不用的，资源处理都应该交给新的URI，还有对应的302，临时移动。

4开头的就是客户端错误，如403，没有权限，404，not found，网页找不到，400,客户端语法有错误。

5开头的就是服务器的错误了。500即在请求的时候发生了错误，503表示停机或者在维护。

1. http和https的区别

Http是明文的，不安全，而HTTP的安全超文本协议，在HTTP的基础上有更强的安全性，简单了说，HTTPS就是使用了SSL的HTTP。因此他们的区别主要有3点，1是传输协议的不同，HTTP的明文传输，而HTTPS是加密传输的，因此HTTPS是安全的，而HTTP不安全，第二是默认端口号不同，HTTP默认端口号使80，HTTPS默认端口号是443。第三是HTTPS的使用需要证书申请，而HTTP不需要。另外在效率上HTTP是优于HTTPS的，因为HTTPS多了加密和解密的过程。

1. http报文，请求报文， HTTP和TCP的区别

http报文分为了请求报文和响应报文。具体呢又分为通用信息头，请求头，请求空行，请求体，响应头，响应体。其中通用信息头中的信息包括了请求URL地址，请求的方法，HTTP协议版本，状态码等信息。请求头和请求体中是key-value的形式存储了请求的各种信息。比如缓存的信息，cache，请求内容的语言，编码方式，长度等。

1. TCP3次握手以及3次握手的原因

第一次握手：客户端向服务端发送请求报文SYN，初始化序号seq=x

第二次握手：当服务端收到客户端的SYN报文，如果同意建立连接，则向客户端发送确认报文，SYN = 1，ACK=1，确认号为ack = x+1，同时初始化序号y，

第三次握手：客户端收到确认报文后，还要向B发出确认，确认号ack=y+1,序号为x+1。

那么为什么需要3次握手不能两次握手呢，我们可以从3次握手的过程来看；

第一次握手，客户端发送网络包，服务端接收到了，表明客户端的发送能力，服务端的接收能力是正常的。

第二次握手，服务端发送网络包，客户端收到了，表明服务端的发送能力，接收能力和客户端的发送和接收能力都是正常的。不过，这时候服务端并不知道，客户端接收能力是否正常。也有可能客户端没接收到。不论接收到和没有接收到，服务端都是不知道的。因此需要第三次握手。

第三次握手，客户端向服务端发送，这样服务端就能确认客户端的发送和接收都是正常的。

需要三次握手的原因就是客户端和服务端都需要确认双方的发送能力和接收能力都是正常的，如果两次握手的话，服务端在不知道客户端是否能够接收的情况下就建立连接，如果此时客户端不需要建立连接了，那么就会造成资源的浪费。在网络拥堵的情况下是很容易发生的。

1. TCP4次挥手以及4次挥手的原因

4次挥手的话客户端和服务端都能够发起。这里我们假设是客户端发起的。

第一次挥手：客户端发送释放报文FIN = 1，需要seq = u;并停止发送数据

第二次挥手：服务端收到FIN之后，会发送确认报文ack = u+1。服务端进入CLOSE\_WAIT状态。这个状态是为了服务端还有向客户端发送还没发完的数据，因此此时客户端不能发数据，但是服务端能发。

第三次挥手：如果服务端没有要发的数据，也想要断开连接了，和客户端第一次挥手一样，发送一个释放连接报文，FIN=1，

第四次挥手：客户端收到服务器的释放报文后，同样发送一个ACK作为应答，并且将序列号+1作为序列号，此时客户端进入TIME\_WAIT状态。客户端等待2MSL后释放连接，服务端收到确认后也释放连接。

4次挥手的原因。因为当服务端收到客户端的关闭连接请求时，服务端可能还没有发完的数据，因此只能先回复一个ACK报文，告诉客户端，你的请求我收到了。只有当服务端所有的数据发送完毕之后，服务端才能发送FIN报文，因此需要4次挥手。

1. 为什么要等待2MSL？

MSL是一个报文的最大生存时间，2MSL就是一个报文一个来回的最大时间。等待2MSL主要是为了防止客户端向服务端发送的确认报文丢失，如果确认报文丢失了，服务端会从新发起FIN报文，释放连接请求，然后客户端可以重新确认。如果2MSL还没有收到新的请求，就说明服务端已经收到了，那么就可以正常关闭了。

1. TCP的可靠性体现在哪？

TCP通过校验和，序列号，确认应答，重发控制，连接管理，窗口控制以及拥塞控制等保证可靠传输。

校验和：校验和位于首部，目的是校验如果数据在传输过程中发生了变化，那么数据将会被丢弃并且不会确认应答，从而保证了数据的正确性。

重发控制就是：当TCP发出一个报文之后，有一个定时器，如果超过了这个时间还没有确认应答，将重新发送，从而保证数据能够发送给对方。

连接管理:3次握手，4次挥手。保证正确的建立连接和关闭连接，减少资源的浪费。

滑动窗口控制：因为如果每发一个段都需要确认应答再发下一个段，这样效率十分的慢，窗口控制就是发送方发出去一个段之后，不需要等待确认就可以继续发送，但是并不能无限制的发送，接收方的接收能力有限，因此接收方会告诉发送发一个限制值，这个值就是窗口的大小。滑动窗口控制就是为了提高数据发送的效率。

拥塞控制：如果网络出现阻塞，报文丢失了，发送方会重新发送，这样会造成网络更加拥堵，因此需要控制发送方的发送速率。主要就是通过控制窗口大小来实现的。慢开始，拥塞避免，快重传，快恢复。

1. TCP和UDP的区别？适用场景

UDP是面向无连接的，他的结构比较简单，除了端口号就没有什么了。他的机制就是当收到应用程序发来的数据，立刻就不发到网上，失败了也不会重发，顺序乱掉也不会处理，流量控制，阻塞控制也没有，它的特点就是简单高效，但是不安全。因此UDP适用于需要处理速度快，对丢包不敏感的应用，比如直播，视频电话等，所以网络不好的时候经常由卡顿，且卡了后卡掉的内容也没了。

TCP是面向连接的，在传输数据之前需要3次握手，断开时需要4次挥手。充分实现了数据传输时的各种控制功能，保证了可靠传输，可以丢包重发，顺序乱了也能保证顺序的正确性。TCP适用于对准确性较高的场景，比如问文件，邮件的传输等。

1. mac(物理)地址和ip地址的区别？有了Mac,为什么还要ip地址

mac(物理地址)是刻录在网卡上的物理地址，在全球范围内都具有唯一性，而ip地址是网络中主机的逻辑地址，只在同一个网段中具有唯一性。因为我们在传输层传输的时候，并不是直接将数据发送到目标主机的，由于网络拥堵等各种原因，需要经过多个地址的转发，比如A主机到B主机，他可能由A到C到D再到B，因此也有了网络层的寻址以及路径的选择。

1. 打开一个网站，经历哪些过程，分别用到了什么协议

浏览器查找域名的IP地址 （DNS：获取域名对应的IP）

浏览器向web服务器发送HTTP请求（cookies会随着请求发送给服务器）

服务器处理请求 （请求 处理请求 参数、cookies、生成一个HTML响应）

服务器返回HTTP报文，发回一个HTML响应。

浏览器收到响应，解析并渲染页面，浏览器开始显示HTML。

连接结束

使用的协议:

DNS: 获取域名对应的IP TCP: 与服务器建立TCP连接

IP: 建立TCP协议时，需要发送数据，发送数据在网络层上使用IP协议

OSPF：IP数据包在路由器之间，路由选择使用OSPF协议

ARP：路由器在与服务器进行通信的过程中，将IP地址装换成MAC地址

HTTP：客户端浏览器与Web服务器之间的应用层通信协议，在TCP建立完成后，使用HTTP协议访问网页

1. DNS解析过程，DNS劫持了解吗。

DNS解析就是由域名到IP地址的解析过程。

1. 当我们在浏览器中输入地址的时候，会先在浏览器的缓存中查找
2. 如果浏览器的缓存中没有，就回去系统的缓存中查找，系统的解析呢是通过c盘中一个叫hosts的文件来解析的，我们也可以添加自己的域名映射，这个过程也被很多黑客所利用，黑客通过修在电脑中的host文件，使当输入特定的域名的时候就会跳转到他指定的ip地址中去，造成所谓的DNS劫持。
3. 当系统中找不到的时候，就会在本地服务器找，本地服务器通常在所在城市的某个角落，性能很快。
4. 如果本域名地服务器没有就回去根服务器中查找。
5. 然后根域名服务器返回给一个主域名服务器，然后本地服务器再向主服务器发送请求进行查找，并返回一个名称服务器。
6. 然后通过名称服务器进行查找，最后返回Ip地址。
7. 最后本地服务器把这个结果缓存起来。
8. 然后返回给用户，用户把这个结果缓存到本地系统中。
9. GET和POST有什么不一样

get方法和post方法是HTTP协议的两种请求方法，

1. 首先，在带参数的时候，get方法是在URL中，所有人都可见的，而post方法是在请求体中，不过他们的本质都是TCP传输，其实也可以参数带在请求体中用get方法，在URL中用POST方法，不过需要服务器的支持。
2. 由于get方法的参数是在地址栏上的，浏览器和服务器处于性能的考虑是对URL的长度有限制的，即get方法的参数的长度有现在，而post方法在请求体中，长度就没有限制了。不过http协议并没有这一限制，如果使用postman发送get请求，几千的长度也是能够发送出去的，长度的限制主要来源于服务器。
3. 数据类型get方法只能使用ascii码的字符，post方法没有限制。
4. get方法的参数能够被缓存且参数会保存在浏览器的历史当中，post方法的参数不会被缓存也不会保存在历史中。
5. 因此在安全性上post方法相对get方法要更加安全一点，不过http都是明文传输，使用post也不是绝对安全的，要真正安全还是得使用https。
6. 后退或者刷新得时候，get方法是无害的，而post方法会被重新提交，可能照成错误。
7. HTTP是不保持状态的协议，如何保存用户状态？

HTTP是无状态的协议，每次请求完毕关闭连接之后，之前的状态就没有了。要想要吃用户状态，可以通过seeion或者cookie来存储用户的状态。

1. session和cookie

cookie是以key,value存储在浏览器的一小段文本，当向服务器请求时，cookie会随着请求一起发过去。它有一个特点，不可跨域性。即不同域之间的cookie是不能互相访问和修改的，因此我们设置cookie的时候通常还要设置他的域domin。另外保存的时间基本单位为秒，可以通过setAge指定，如果没有设置的话，默认在浏览器关闭就没有了。删除cookie的话只需要将当前的cookie指定为0秒即可删除了。就是时间到了自动删除。最后是他的安全性，http是不安全的，cookie在其中传输就不安全，可以通过设置cookie的secure属性为true，此时浏览器只会在HTTPS等安全协议中进行传输，不过cookie内容仍然不是加密的，如果需要再提高安全性，需要对cookie的内容也加密。

session是以key,value存储在服务器的数据信息。由于他在服务器端，因此也更安全。生命周期来说，session有默认的时间，一般为几十分钟，超过就没有了，另外把客户端关闭或服务器关闭，session也会消失，但是cookie不一定会。

1. ARP协议？

地址解析协议，通过解析IP地址得到mac(物理)地址

1. DDos攻击了解吗

DDos攻击是分布式Dos攻击，即普通的Dos攻击是单个主机对你发起攻击，向你发起虚假的请求，耗费你服务器的资源，搞崩你的服务器，DDos攻击是多台服务器同时想你发起攻击，相对来说威胁更大。