一、七层模型和五层模型：

1、物理层（建立、维护、断开物理连接）：网线、电缆、集线器 。这一层数据的单位称为比特（bit）

2、数据链路层（建立逻辑连接、进行硬件地址寻址、差错校验等功能）： arp，数据的单位称为帧（frame）

3、网络层（进行逻辑地址寻址，实现不同网络之间的路径选择）：ipv4、icmp，单位为数据包（packet）

4、传输层（定义传输数据的协议端口号，以及流控和差错校验）：tcp、udp，单位为数据段（segment）

5、会话层（建立、管理、终止会话）：rpc

6、表示层（数据的表示、安全、压缩）：jpeg、ascii，

7、应用层（网络服务与最终用户的一个接口）：http、ftp、smtp

8、五层模型：物理层、数据链路层、网络层、传输层、应用层（会话层、表示层、应用层三合一）

一、http返回码：

2XX（成功） 200：成功

3XX（重定向） 300：多种选择 301：永久移动 302：临时移动

4XX（客户端） 400：错误请求 401：未授权，用户需要登录 403：禁止访问 404 : 访问的资源不存在

5XX（服务端） 500：服务器内部错误 501：功能尚未实施 502：网关错误 503：服务不可用 504：网关超时

502错误定义：是网关错误， 通俗点说就是和web服务器通信失败

错误的原因：

1、网络不通，不能访问web服务器， 有可能断网， 开启了防火墙等， 可以通过ping命令来定位

2、web服务器没有启动， 可以通过查看日志来定位这个问题，或者查看端口是否启动

3、web服务器请求太多，响应不了请求，这个表现是有时好有时不好，可以通过查看web服务器日志来定位

二、浏览器http全流程：

第一步：在浏览器输入地址，回车。

第二步：浏览器查找域名的 IP 地址：浏览器分解url、生成http请求消息、解析这个域名，获取ip地址（浏览器会首先查看本地硬盘的 hosts 文件查找 ip 地址，没有则发出一个 DNS请求到本地 DNS 服务器，如果还是没有，则会一层一层往上请求到根dns服务器）

第三步：浏览器向 web 服务器发送一个 HTTP 请求：拿到域名对应的 IP 地址之后，浏览器会以一个随机端口（1024<端口<65535）向服务器的 WEB 程序（常用的有 httpd,nginx 等）80 端口发起 TCP 的连接请求（tcp模块1.创建套接字、2.连接服务器、3.发送数据； IP模块发送请求：生成IP头部、mac头部； MAC模块发送请求：生成生成报头、校验序列、电信号；4、PHY模块发送请求：网卡中的PHY模块会将通用电信号转换成网络传输所需的格式，通过网线发送出去。经过网络转发后，最终到达服务器。）

客户端向服务器发起 http 请求的时候，会有一些请求信息，请求信息包含三个部分：

| 请求方法 URI 协议/版本

| 请求头(Request Header)

| 请求正文：

第四步：这个连接请求到达服务器端后，进入到网卡、进入到内核的 TCP/IP 协议栈（用于识别该连接请求，解封包），最终到达 WEB 程序，建立了 TCP/IP的连接。

第五步：服务器根据内部逻辑处理请求，返回一个 HTTP 响应

第六步：本地接收服务器响应，浏览器显示 HTML。(vue 实例有一个完整的生命周期，也就是从开始创建、初始化数据、编译模版、挂载 Dom -> 渲染、更新 -> 渲染、卸载等一系列过程，我们称这是 Vue 的生命周期。)

http优化

资源内联：既然每个资源的首次访问都会存在握手等rtt损耗，那么越少数量的资源请求就越好，例如在一个html中src访问css，不如直接将其这个css集成到html中

图片懒加载：用到的时候再加载

服务器渲染：让服务端先将页面渲染好，再发送给客户端，也可以减少rtt的次数

三、URL组成格式

图中中括号是可选项

protocol 协议，常用的协议是http

hostname 主机地址，可以是域名，也可以是IP地址

port 端口 http协议默认端口是：80端口，如果不写默认就是:80端口

path 路径 网络资源在服务器中的指定路径

parameter 参数 如果要向服务器传入参数，在这部分输入

query 查询字符串 如果需要从服务器那里查询内容，在这里编辑

fragment 片段 网页中可能会分为不同的片段，如果想访问网页后直接到达指定位置，可以在这部分设置

九、交换机与路由器的区别：

①工作所处的OSI层次不一样，交换机工作在OSI第二层数据链路层，路由器工作在OSI第三层网络层

②寻址方式不同：交换机根据MAC地址寻址，路由器根据IP地址寻址

③转发速不同：交换机的转发速度快，路由器转发速度相对较慢。

十. Http与Https的基本概念和他们的区别

HTTP的中文叫超文本传输协议，它负责完成客户端到服务端的一系列操作，是专门用来传输注入HTML的超媒体文档等web内容的协议，它是基于传输层的TCP协议的应用层协议

HTTPS可以理解为HTTP+ssl/tls（数字证书）的组合

1、http和https的区别

http的URL以http://开头，而https的URL以https://开头

http是不安全的，而https是安全的

http的标准端口是80，而https的标准端口是443

在OSI网络模型中，https的加密是在传输层完成的，因为SSl是位于传输层的，TSL的前身是SSL，所以同理

http无需认证证书，而https需要认证证书

2、https

HTTPS在HTTP的基础上加入了SSL协议，SSL依靠证书来验证服务器的身份，并为浏览器和服务器之间的通信加密。

HTTP协议以明文方式发送内容，不提供任何方式的数据加密，如果攻击者截取了Web浏览器和网站服务器之间的传输报文，就可以直接读懂其中的信息。

证书内容：

证书信息：过期时间和序列号

所有者信息：姓名等

所有者公钥

3、https的工作原理

首先服务端给客户传输证书，这个证书就是公钥，包含了很多信息，比如说证书的颁发机构，证书的过期时间

客户端进行证书的解析，若果发现没问题，就生成一个随机值（私钥），然后用证书（也就是服务端发送的公钥）进行加密，并发给服务端

服务端使用私钥将这个信息进行解密，得到客户端的私钥，然后客户端和服务端就可以通过这个私钥进行通信了

服务端将信息进行对称加密，私钥正好直有客户端和服务端知道，所以信息就比较安全了

服务端将进行对称加密后的消息进行传递

客户端使用私钥进行信息的解密

4、get和post的区别：

1、区别就是GET把参数包含在URL中，POST通过request body传递参数。

2、GET产生一个TCP数据包；POST产生两个TCP数据包。对于GET方式的请求，浏览器会把http header和data一并发送出去，服务器响应200（返回数据）；

而对于POST，浏览器先发送header，服务器响应100 continue，浏览器再发送data，服务器响应200 ok（返回数据）。

六、tcp报文：

七、tcp和udp的区别：

TCP提供面向连接的传输，通信前需要先建立连接（三次握手机制）；UDP提供无连接的传输，通信前不需要建立连接。

TCP提供可靠的传输（有序、无差错、不丢失、不重复）；UDP提供不可靠的传输。

TCP面向字节流的传输，它能将信息分割成组，并在接收端将其重组；UDP是面向数据报的传输，没有分组开销。

TCP提供拥塞控制和流量控制；UDP不提供拥塞控制和流量控制机制。

八、tcp协议的连接过程（三次握手、四次挥手）

1.说一说TCP三次握手，四次断开？

置位概念：根据TCP的包头字段，存在三个重要的标识ACK、SYN、FIN

ACK：表示验证字段

SYN：位数置为1，表示建立TCP连接

FIN：位数置为1，表示断开TCP连接

TCP的三次握手：

（1）由客户端先向服务端发送建立TCP连接的请求报文，其中报文包含seq序列号，是由发送端随机生成的，并且还将报文中SYN字段置为1，表示需要建立TCP连接。

（2）服务端会回复客户端发送的TCP连接请求报文，其中包含seq序列号，是由服务端随机生成的，并且将回复报文的SYN字段置为1，而且会产生ACK字段，ACK字段数值是在客户端发过来的seq序列号基础上加1进行回复，以便客户端收到信息时，知晓自己的TCP建立请求已经得到了验证。

（3）客户端收到服务端发送的TCP建立验证请求后，会使自己的序列号加1表示，并且再次回复ACK验证请求，在服务端发送过来的seq基础上加1，进行回复。

TCP建立过程状态机转变：

（1）一开始建立连接之前，服务端和客户端的状态都为CLOSED。

（2）服务端创建socket后开始监听，变为LISTEN状态。

（3）客户端建立连接请求，向服务端发送SYN报文，客户端的状态变为SYN\_SENT。

（4）服务端收到客户端的报文后会向客户端发送ACK和SYN报文，此时服务端的状态变为SYN\_RCVD。

（5）然后，客户端收到ACK和SYN，会向服务端发送ACK，客户端状态变为ESTABLISHED，服务端收到客户端的ACK后也变为ESTABLISHED。

（6）至此，TCP3次握手完成，建立连接！

TCP断开连接四次挥手过程：

（1）客户端发送TCP连接请求的报文，其中报文中包含了seq序列号，是由发送端随机生成的，并且还将报文中的FIN字段置为1，表示需要断开TCP连接。

（2）服务端会回复服务端发送的TCP断开请求报文，其中包含seq序列号，是由回复端随机生成的，而且会产生ACK字段，ACK字段数值，是在服务端发过来的seq序列号基础上加1进行回复的，一边服务端收到信息时，知晓自己的TCP断开请求已经得到了验证。

（3）服务端在回复完服务端的TCP断开请求后，不会马上就进行TCP连接的断开，客户端会先确保断开前，所有传输到服务端的数据是否已经传输完毕，一但确认传输数据完毕就会将回复报文的FIN字段置为1，并产生随机seq序列号。

（4）客户端收到客户端的TCP断开请求后，会回复客户端的断开请求，包含随机生成的seq字段和ACK字段，ACK字段会在客户端的TCP断开请求的seq基础上加1，从而完成客户端请求的验证回复。

（5）至此TCP断开的四次挥手过程完毕。

2.为什么连接的时候是三次握手，关闭的时候确是四次握手？

因为当服务端收到客户端的SYN连接请求报文后，可以直接发送SYN+ACK报文。其中ACK报文是用来应答的，SYN报文是用来同步的。但是关闭连接时，当服务端收到FIN报文时，很可能并不会立即关闭SOCKET，所以只能先回复一个ACK报文，告诉客户端，你发的FIN报文我收到了。只有等到我服务端所有的报文都发送完了，我才能发送FIN报文，因此不能一起发送，故需要四步握手。

3.为什么TIME\_WAIT状态需要经过2MSL（最大报文段生存时间）才能返回到CLOSE状态？

虽然按道理，四个报文都发送完毕，我们可以直接进入CLOSE状态了，但是我们必须假象网络是不可靠的，有可能最后一个ACK丢失。所以TIME\_WAIT状态就是用来重发可能丢失的ACK报文。在客户端发送出最后的ACK回复，但该ACK可能丢失。服务端如果没有收到ACK，将不断重复发送FIN片段。所以客户端不能立即关闭，他必须确认服务端接收到了ACK。客户端会在发送出ACK之后进入到TIME\_WAIT状态。客户端会设置一个计时器，等待2MSL是两倍的最大报文段生存时间。如果直到2MSL，客户端都没有再次收到FIN，那么客户端推断ACK已经被成功接收，则结束TCP连接。

4.为什么不能用两次握手进行连接？

三次握手完成两个重要的功能，既要双方做好发送数据的准备工作（双方都知道彼此已经准备好），也要允许双方就初始序列号进行协商，这个序列号在握手过程中被发送确认。

如果改成两次握手，可能会发生死锁现象。客户端给服务端发送了一个连接请求，服务端收到了这个分组，并发送了确认应答分组。按照两次握手的协议，服务端认为已经成功地建立了，可以开始发送数据分组。可是客户端的应答分组在传输中被丢失的情况下，将不知道服务端是否已经准备好，不知道服务端建立什么样的序列号，客户端甚至怀疑服务端手否收到自己的连接请求分组。在这种情况下，客户端认为连接还未建立成功，将忽略服务端发来的任何数据分组，只等待连接确认应答分组。而服务端在发出的分组超时后，重复发送同样的分组。这样就形成了死锁。

5.如果已经建立了连接，但客户端突然出现故障了怎么办？

TCP还设有一个保活计时器，显然，客户端如果出现故障，服务器不能一直等下去，白白浪费资源。服务器没收到一次客户端的请求后都会重新复位这个计时器，若两小时还没有收到客户端的任何数据，服务端就会发送一个探测报文段，以后每隔75分钟发送一次，若一连发送10个探测报文任然没反应，服务端就认为客户端出了故障，接着就关闭连接。

6：TCP，UDP区别，为什么可靠和不可靠

TCP的优点： 可靠，稳定 TCP的可靠体现在TCP在传递数据之前，会有三次握手来建立连接，而且在数据传递时，有确认、窗口、重传、拥塞控制机制，在数据传完后，还会断开连接来节约系统资源。

TCP的缺点： 慢，效率低，占用系统资源高，易被攻击 TCP在传递数据之前，要先建连接，这会消耗时间，而且在数据传递时，确认机制、重传机制、拥塞控制机制等都会消耗大量的时间，而且要在每台设备上维护所有的传输连接，事实上，每个连接都会占用系统的CPU、内存等硬件资源。 而且，因为TCP有确认机制、三次握手机制，这些也导致TCP容易被人利用，实现DOS、DDOS、CC等攻击。

UDP的优点： 快，比TCP稍安全 UDP没有TCP的握手、确认、窗口、重传、拥塞控制等机制，UDP是一个无状态的传输协议，所以它在传递数据时非常快。没有TCP的这些机制，UDP较TCP被攻击者利用的漏洞就要少一些。但UDP也是无法避免攻击的，比如：UDP Flood攻击……

UDP的缺点： 不可靠，不稳定 因为UDP没有TCP那些可靠的机制，在数据传递时，如果网络质量不好，就会很容易丢包。

问：一次HTTP请求的全过程，包括域名解析、定位主机等

域名解析 --> 发起TCP的3次握手 --> 建立TCP连接后发起http请求 --> 服务器响应http请求，浏览器得到html代码 --> 浏览器解析html代码，并请求html代码中的资源（如js、css、图片等） --> 浏览器对页面进行渲染呈现给用户

问：TCP三次握手

第一次握手：客户端发送syn包(seq=x)到服务器，并进入SYN\_SEND状态，等待服务器确认;第二次握手：服务器收到syn包，必须确认客户的SYN(ack=x+1)，同时自己也发送一个SYN包(seq=y)，即SYN+ACK包，此时服务器进入SYN\_RECV状态;第三次握手：客户端收到服务器的SYN+ACK包，向服务器发送确认包ACK(ack=y+1)，此包发送完毕，客户端和服务器进入ESTABLISHED状态，完成三次握手。