计算机网络面试题汇总

1. 讲讲计算机网络七层模型和五层模型,都有哪些协议

OSI 是七层协议，从上到下是应用层、表示层、会话层、传输层、网络层、数据链路层、物理层。

七层协议从上到下分别是：

应用层：为应用程序提供网络服务，包含的协议有 HTTP、DNS、SMTP 等

表示层：数据格式转换、数据加密等

会话层：建立、断开和维护网络会话链接

传输层：为上层协议提供可靠的数据传输，包含的协议有 TCP、UDP

网络层：寻址和路由，包含的协议有 IP、ICMP 等

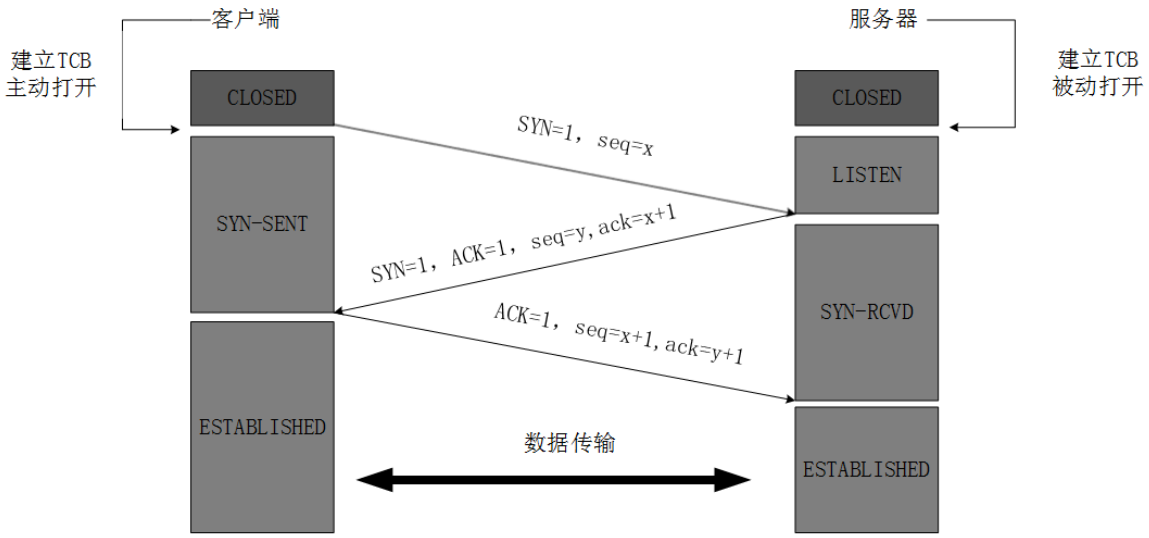
数据链路层：每个主机都有一个唯一的 MAC 地址，最终的寻址是由数据链路层完成的，包含的协议有 ARP 等。

物理层：完成最终的物理信号的传输。

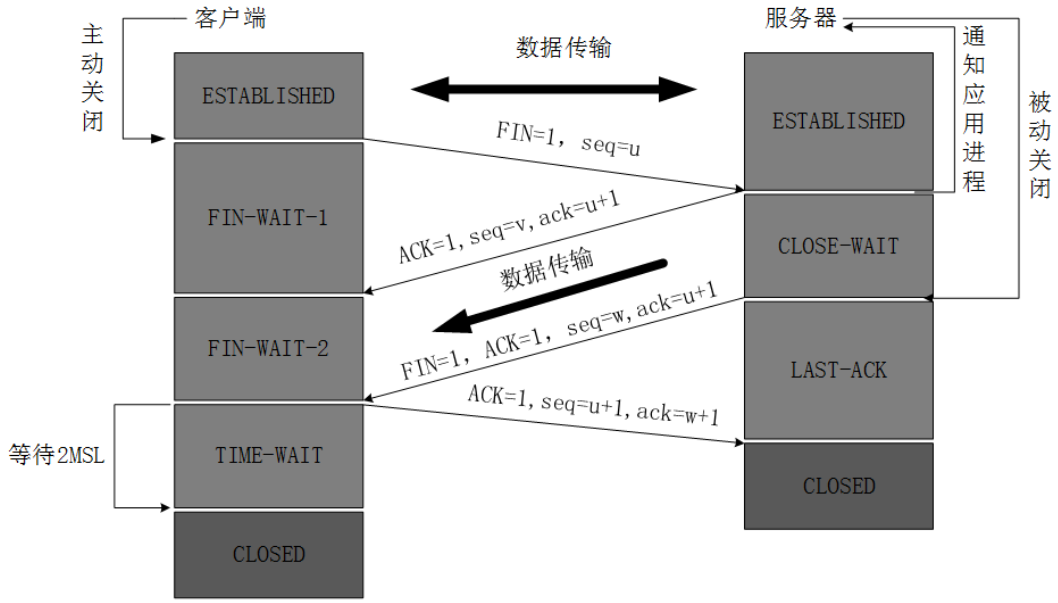
五层协议与七层协议相比，合并了应用层、表示层和会话层。TCP/IP 四层协议进一步合并了数据链路层和物理层。

2. 讲一讲 TCP 三次握手和四次挥手

三次握手：



四次挥手：



3. 为什么要进行三次握手而不是两次

防止客户端没有收到服务端的确认报文，导致浪费服务端的资源。

防止已经失效的连接报文又达到服务端，导致服务端建立连接但是客户端没有连接，使得资源白白的浪费了。

4. 为什么四次挥手最后要等待 2MSL

因为 MSL 是报文的最大生存时间，如果服务端没收到客户端的消息，然后重传的话，那么客户端一定会在 2MSL 内收到该消息，等待 2MSL 还没收到消息说明服务端没有进一步的信息，因此可以释放连接。

5. TCP、UDP 协议的区别

TCP 和 UDP 都是工作于 OSI 模型中传输层的协议。

区别在于：

1. TCP 是面向连接的，UDP 不是面向连接的。(意味着 TCP 要进行三次握手与四次挥手，拥塞控制等，而 UDP 不需要)。因此 TCP 提供可靠的数据传输(数据准确性和数据顺序性)，而 UDP 无法保证。
2. TCP 是点到点的通信，而 UDP 支持一对一、一对多、多对多的通信。
3. TCP 是面向数据流的(通过滑动窗口对数据的发送进行控制)，而 UDP 是面向报文的(直接交给网络层，不进行分拆)。(因为 UDP 不需要使用滑动窗口的机制进行拥塞控制，所以无需面向数据流)
4. TCP 报文首部开销大、UDP 报文首部开销小。
5. 基于以上区别，TCP 适合数据准确性高的场景，比如文件传输；而 UDP 适合实时性高但是准确性要求不高的场景，比如视频对话等。

4. TCP 和 UDP 报文首部的区别

TCP 报文首部是 20 字节，UDP 报文首部是 8 字节。

UDP 报文首部包括：源端口、目标端口、数据长度、校验和(检验数据传输前后的一致性)。

TCP 报文首部包括：源端口、目标端口、序号(数据开始的序号)、确认序号(期望收到对方应答报文的开始序号)、报文首部长度、标志位(例如 ACK=1，SYN=1，FIN=1 等)、校验和、接受窗口大小(流量控制)、紧急指针、保留、可选项。

6. TCP 协议如何保证可靠传输

本质上是这么几点：一致性、有序性、不重复、不丢失。

1. 通过报文首部的校验和字段来保证数据前后的一致性。
2. TCP 会对数据包进行编号，从而保证有序性。
3. 接收方会舍弃掉重复的数据。
4. ARQ 协议，避免数据丢失。
5. 流量控制与拥塞控制。(本质上是为了保证前面几点)。

5. ARQ 协议是什么

又叫自动重传协议，是保证数据传输可靠性的重要协议，分为停止等待 ARQ 协议和连续 ARQ 协议。

停止等待 ARQ 协议：每发完一个数据包就停下来等待应答，在超时时间内收到确认信息则继续传输，否则进行重传。

连续 ARQ 协议：维持一个滑动窗口，位于窗口内的数据可以连续发出，只需要对最后一个数据进行应答即可，如果收到错误信息或者超时未收到信息，则全部进行重传。

6. TCP 协议是怎么进行流量控制的

流量控制是作用于接收方的，是为了保证接收方能够来得及接受发送方的数据。

实现方式是连续 ARQ 协议实现的，接收方的报文中维持一个窗口大小的字段，用来控制发送方的发送速度，确认机制与重传机制是由 ARQ 协议保证的。

7. TCP 协议是怎么进行拥塞控制的

拥塞控制是作用于网络的，是为了避免过多的数据注入网络导致拥塞。

发送方维持一个拥塞窗口，实际窗口取拥塞窗口和滑动窗口的较小值，首先是**慢开始算法，**窗口由 1 开始指数变大，当超过慢开始门限的适合，使用**拥塞避免算法**，此时让拥塞窗口按线性增大，当出现拥塞时(连续未收到确认应答)，将慢开始门限变为出现拥塞时窗口大小的一半，然后将拥塞窗口从 1 开始重新来过。

快重传：要求接收方当发现有数据没有达到时(即发现报文失序)，要立刻通知发送方，从而提高网络吞吐。当发送方连续三次收到接收方对某个数据的确认请求时，就立刻重新发送该数据。

快恢复：快恢复是配合快重传使用的，但发送方收到快重传指令时，将慢开始门限设为此时拥塞窗口大小的一半，然后从该慢开始门限直接进入拥塞避免算法。原因是如果发送方能连续收到 3 个确认信息，代表网络并不拥塞，此时从 1 开始恢复是不必要的，因此我们采用这种快恢复。

7. 说说 TCP 粘包

所谓的粘包是一种正常的现象，TCP 协议是面向字节流的，本身不会区分不同的包，实际操作中需要应用层的协议来进行分割，比如说带上数据的长度，比如说在每个包的结尾添加特殊字符等。

8. 基于 TCP 和 UDP 的应用层协议都有哪些

TCP ：http、smtp、ftp

UDP ：dns

7. 在浏览器中输入 url 地址 ->> 显示主页的过程

DNS 服务进行 ip 查找(一层层的查询，本地 host 文件->本地 DNS 服务器->根服务器) -> 三次握手建立 TCP 连接 -> 向该 ip 发送 http 请求 -> server 端处理请求进行响应 -> 四次挥手断开连接 -> 浏览器对返回内容进行解析，进行网页渲染

8. HTTP 协议包括哪些请求？

get、post、head(类似于 get 请求，但是返回的响应中没有具体内容，用于快速获取报头)、put(新建一个文档)、delete(删除一个文档)

9. 简述 HTTP 中 GET 和 POST 的区别

1. get 请求是将 header 和 data 一起发出去，而 post 请求是先发 header，服务器返回 100，然后再发送 data。因此 get 请求产生一个 tcp 数据包，post 请求返回两个 tcp 数据包。
2. get 请求参数在 url 里，长度有限制，post 请求参数在 body 体中，长度没有限制。因此 post 请求比 get 请求更安全。

10.什么是 http 重定向

重定向是指服务器进行 url 转发的过程，一般返回码是 30x。重定向分为永久重定向和临时重定向，重定向一般用于要访问资源已经迁移的情况。

11.HTTP 和 HTTPS 的区别

1. http 是明文传输，https 是使用 ssl 加密的密文传输，因此 https 更加安全
2. http 的端口是 80，https 的端口是 443
3. https 需要申请 ca 证书，需要一定的费用

ssl 算法详解：私钥：钥匙，公钥：锁，只有私钥能打开公钥加密的内容。首先服务器申请 ca 证书，拿到私钥和公钥，并将公钥发送给客户端，客户端验证 ca 证书合法性后，生成一个随机值(客户端的私钥)，和数据一起用公钥加密后发送给服务器，服务器通过私钥解开加密的内容拿到随机值，通过对称加密将返回的信息和该随机值混合在一起，返回给客户端，客户端通过该随机值进行解密，得到服务器返回的信息。实际上是一个非对称加密算法(公钥私钥)和对称加密算法混合而来的加密算法。

12.HTTP 长连接、短连接

此处都是针对 tcp 连接而言的，http1.0 中默认是短连接，http1.1 中默认是长连接，长连接默认在请求头中加入参数 Connection:keep-alive，短连接每次会话都需要重新建立一次连接，而长连接一旦建立后就不会中断。

长连接对请求频繁的情况速度更快，而短连接在请求不频繁的情况下更加节省服务器的资源。

13.cookie 和 session 的区别

http 协议本身是无状态的，cookie 和 session 都是在 http 服务中保持状态的方法。

区别是：

1. cookie 是保存在客户端的，每次 http 请求都要带上 cookie 来表明会话状态；而 session 是保存在服务端的。
2. cookie 可以长时间保持，比如一些默认登陆等；而 session 一般存活时间较短，当客户端关闭或者会话超时的情况下，session 就会关闭。
3. cookie 存储在客户端，容易被获取，因此安全性较差，而 session 存储在服务端，因此安全性较好。
4. 因为要在 http 请求中传输，所以 cookie 的格式和长度都有限制，而 session 则自由很多。

需要注意的是，因为服务端使用 sessionID 来唯一标识 session，因此客户端每次请求需要带上 sessionID，因此 session 机制可能要依赖 cookie 机制来传递 sessionID(也可以携带在参数中)。

14. 如果浏览器禁止了 cookie，那怎么办

cookie 一般用来标识用户登陆状态，如果 cookie 被禁止了，有两种方式：

1. 在每次请求时都带上 sessionId 参数来标识用户属于哪个已登陆的 session。
2. Token 机制，服务端生成一串字符串，用作客户端的标识，客户端每次请求时带上该字符串即可。

15. 如何解决分布式 session 问题

分布式情况下，用户在某一个服务器登陆了，怎么保证下次访问的机器上还有该 session 信息。

1. 通过代理的方式保证同一用户的每个请求都会发送到固定的服务器上。
2. 通过广播的方式将 session 信息进行广播。

3. 通过一个缓存中间件将所有的 session 信息同一储存。

14.各种协议与 HTTP 协议之间的关系

应用层的 http 请求先要经过传输层的 tcp 协议处理，tcp 会对数据进行划分并且进行标号，然后通过网络层的 ip 协议进行寻址并传输，服务端的网络层接收到数据后向上交给 tcp 处理，通过一些数据可靠性方面的处理，交给应用层的 http 协议进行实际的请求。

15.http 请求中常见的状态码有哪些

100：继续，客户端继续进行操作，比如 post 请求中会返回。

200 : success，请求成功

301：永久重定向，302：临时重定向

404：客户端错误，无法根据 url 找到资源

500：服务端内部错误，无法完成请求

16.http 中请求报文和响应报文的组成

请求报文由请求行(请求方法、请求地址、http 版本)、请求头、请求体组成。

响应报文由状态码、响应头和响应体组成。

HTTP 的请求头有哪些信息：

connection：是否保持连接

cache-control：缓存控制

content-encoding/length/language：内容标识

Accept / language / charset：客户端能接受的内容

user-agent：客户端信息

17.HTTP 1.0、HTTP 1.1 及 HTTP 2.0 的主要区别是什么

http 1.1 主要新增的特性是支持长连接、支持多种缓存策略、带宽及连接优化等。

http 2.0 主要新增的特性是支持多路复用(**多个请求可以在同一个连接上并行执行**)、支持 header 压缩，支持二进制格式等。

18.请简单说一下你了解的端口及对应的服务？

ftp : 21

ssh : 22

smtp : 25

http : 80

https : 443

mysql : 3306

19.IP 地址分为哪几类？简单说一下各个分类

A 类：以 0 开头，8 位网络号，24 位主机号

B 类：以 10 开头，16 位网络号，16 位主机号

C 类：以 110 开头，24 位网络号，8 位主机号

D 类：以 1110 开头，称为多播地址

E 类：以 11110 开头，留作后续实验用途

20.什么是子网掩码

ip 地址分为网络号和主机号，同一网络中的主机可以相互通信，否则需要借助默认网关进行转发(默认网关可以理解为不同网段之间通信的关口)。

子网掩码是由一串前缀 1 组成的 32 位二进制数，与 ip 地址进行与操作后可以得到网络号，同一网络中主机可以直接通信(也就是 ARP 协议中直接广播给同一网段下所有主机)，如果网络号不同的话，则需要先发送给默认网关 IP 对应的主机，由该主机代为转发。

21.简单解释一些 ARP 协议的工作过程

ARP 协议工作在数据链路层，主要是用来做 IP 地址到 MAC 地址的转换。首先，每个主机都会维持一个 ARP 列表，记录了 IP 地址对应的 MAC 地址，当处理一个请求时，主机首先查看自己的 ARP 列表中是否有目标 ip 的记录，有的话即向对应的 MAC 地址发送请求，否则就向同一网络中的其他主机发送请求，其他主机收到请求后检查目标 ip 是否是自己，是的话则进行回复告知 MAC 地址。如果源主机收到回复则进行数据发送，否则等待，超时则失败。

注意，实际都是需要使用 MAC 进行寻址，ip 本身是无法寻址的，因此无法直接根据 ip 地址找到对应的主机，如果两个 ip 地址都在同一网段中，则直接广播即可；如果不在同一网段中，则需要借助默认网关进行转发，最终找到 MAC 地址。

22.NAT 协议是什么

NAT 协议是实现**私网访问公网**的一种协议，因为 ip 地址的数量限制，我们只使用一个公网 ip，其他主机都使用私网 ip，NAT 协议可以实现私网 ip 到公网 ip 的转换，这样我们就可以不局限于 ip 地址的数量，可扩展的主机大大增加了。

注意：在 ip 地址中有一部分私有地址是保留地址，专门用于这种私有网络的情况。

在 NAT 协议连接下的所有主机构成了一个局域网。

根据 ip 地址的访问只能访问到有公网 ip 的那台机器，无法直接访问到背后某台地址是私有 ip 的机器。

23.ICMP 协议是什么

ICMP 协议工作在网络层，主要功能是确定 ip 数据包是否到达目标 ip，并且返回出现错误的原因，ping 命令就是在 ICMP 协议基础上实现的。

24.RPC 调用和 HTTP 调用之间的区别

rpc 和 http 并不是同级的概念，http 协议是工作在应用层的协议，而 rpc 不是协议，只是对已有协议的封装，并且加上了服务发现、服务治理等功能，rpc 可以直接基于 tcp

实现，也可以基于 http 2.0 实现。

rpc 调用相比于 http 调用的优势在于：

1. 效率更高
2. 功能更加全面，各种服务管理的功能都可以继承到 rpc 框架中
3. 使用的体验是面向调用过程的，更加简便

25.说说计算机网络中常见的三大攻击

1. DDos 攻击：短时间内发起大量的连接请求，在三次握手的第三步一直不发送确认回复，导致服务器资源耗尽无法响应正常的请求。
2. XSS 攻击：将恶意前端代码包装在数据中，导致恶意代码被执行的问题。
3. SQL 注入攻击：在填写 SQL 参数时添加恶意代码，例如 ‘a’ or '1'='1'，导致不需要密码验证的情况下就可以直接拿到结果。

25. 负载均衡算法有哪些

负载均衡的算法主要有三种：随机(完全随机、加权随机)、轮询(完全轮询、加权轮询)、哈希(例如根据用户进行哈希，保证该用户每次请求落在同一台服务器上，解决了分布式 session 的问题，进了提高容错性，进阶版是**一致性哈希算法**)。

26. 说说 ngnix

ngnix 是一个轻量级的反向代理服务器，正向代理指的是直接打到具体的机器上，反向代理指的是统一被 ngnix 服务器接受，然后派发给后端服务器，此时需要一些负载均衡策略。

ngnix 性能优异，原因是它采用非阻塞的事件驱动机制，运用了 epoll 模型。

27. 路由器和交换机有什么区别

路由器是根据 ip 地址进行寻址的，交换机是根据 MAC 地址进行寻址的。

28. 什么是 socket 的，和 tcp 协议有什么关系

socket 其实是对 tcp/ip 的协议的封装的应用，让程序员可以更加方便的使用 tcp/ip 协议栈，基本的 socket 编程步骤是：

服务端：

1. 创建一个 socket，使用 socket()
2. 绑定 ip、端口号到 socket 上，使用 bind()
3. 等待接受连接，使用 listen()
4. 接受一个连接，使用 accept()
5. 收发数据，使用 send() / receive()
6. 关闭连接

客户端：

1. 创建一个 socket，使用 socket()
2. 设置目标 ip 和 port，使用 connect 函数进行连接
3. 收发数据，使用 send() / receive()
4. 关闭网络连接

29：如果大量出现 time-wait 状态怎么办

如果系统中存在大量短连接，则有可能出现大量的 time-wait 状态，有可能导致端口耗尽，无法新建连接的情况。解决方法是设置长连接(connection=keep-alive)，连接的复用，缩短 time-wait 时长等。

参考文章：

1. https://juejin.cn/post/6844903662838349838
2. https://blog.csdn.net/qzcsu/article/details/72861891
3. https://zhuanlan.zhihu.com/p/24001696
4. https://zhuanlan.zhihu.com/p/75536009