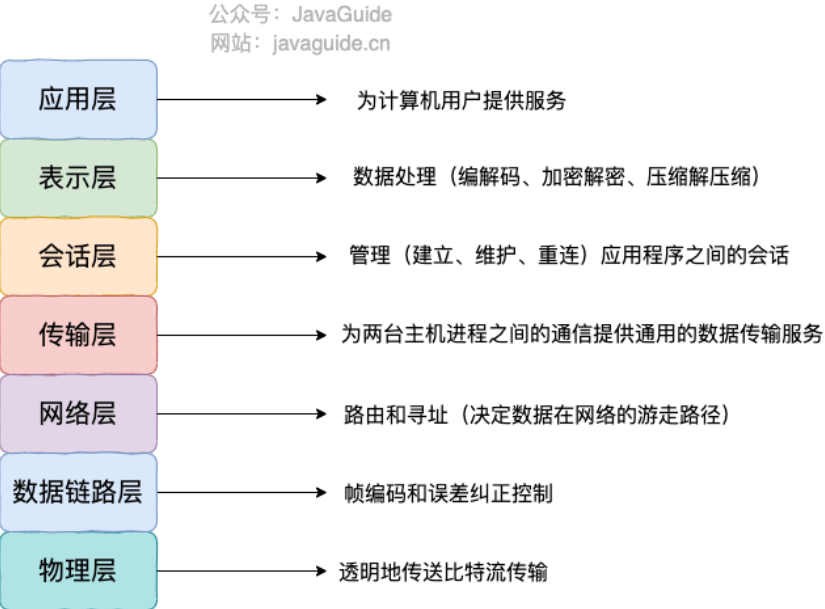
### 计算机网络基础

1. OSI七层模型是什么？每一层的作用？



1. TCP/IP四层模型？每一层作用？

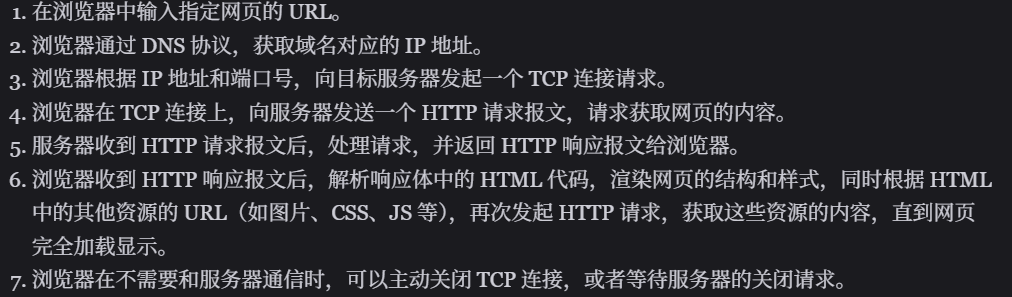


1. 为什么网络要分层?

1各层之间相互独立，每层只需要专注与一类事情，并提供服务给上层。2提高了灵活性和可替换性，每层都可以根据实际需要进行替换。3大问题化小问题解决

### HTTP

1. 从输入URL到显示页面发生了什么过程？



1. HTTP状态码有哪些？



1. HTTP与HTTPS有什么区别？

1端口号80和443；2URL前缀；3安全性和资源消耗，HTTP建立在TCP之上的，所有传输内容都是明文，客户端和服务端无法验证对方身份。HTTPS建立在SSL/TLS之上的HTTP协议，也就是说在HTTP协议之上添加了一个对称加密规则，但对称加密的密钥用服务端的证书进行了非对称加密。

1. HTTP1.0与HTTP1.1的区别？（长连接、状态码、控制对象范围、host头字段允许一个ip多个域名）

**连接方式** : HTTP/1.0 为短连接，HTTP/1.1 支持长连接。

**状态响应码** : HTTP/1.1 中新加入了大量的状态码

**带宽**：HTTP/1.0 中，存在一些浪费带宽的现象，例如客户端只是需要某个对象的一部分，而服务器却将整个对象送过来了，并且不支持断点续传功能，HTTP/1.1 则在请求头引入了 range 头域，它允许只请求资源的某个部分

**Host 头（Host Header）处理** :HTTP/1.1 引入了 Host 头字段，允许在同一 IP 地址上托管多个域名

1. HTTP2.0 （引入了多路复用提高效率同时缓解了队头阻塞问题）

**多路复用（Multiplexing）**：HTTP/2.0 在同一连接上可以同时传输多个请求和响应

**二进制帧（Binary Frames）**：HTTP/2.0 使用二进制帧进行数据传输，而 HTTP/1.1 则使用文本格式的报文。

**队头阻塞**：HTTP/2 引入了多路复用技术，允许多个请求和响应在单个 TCP 连接上并行交错传输，解决了 HTTP/1.1 应用层的队头阻塞问题，但 HTTP/2 依然受到 TCP 层队头阻塞 的影响。

**头部压缩（Header Compression）HPACK**

**服务器推送（Server Push）**：HTTP/2.0 支持服务器推送，可以在客户端请求一个资源时，将其他相关资源一并推送给客户端，从而减少了客户端的请求次数和延迟。

1. HTTP3.0 （UDP、安全性）

**传输协议：**HTTP/2.0 是基于 TCP 协议实现的，HTTP/3.0 新增了 (UDP)QUIC（Quick UDP Internet Connections） 协议来实现可靠的传输，提供与 TLS/SSL 相当的安全性

**连接建立：基于UDP不需要建立连接**

**头部压缩：QPACK**

**队头阻塞**：HTTP/3.0 在一定程度上解决了队头阻塞，这些数据流之间独立互不影响，多路复用+轮询

1. HTTP是无状态的协议那么如何保存用户状态？

Session 机制的存在就是为了解决这个问题，服务端给特定的用户创建特定的 Session 之后就可以标识这个用户并且跟踪这个用户了。并返回session ID给客户端，客户端在下次发送请求时会将ID保存在cookie中。

1. HTTPS中SSL/TSL协议如何实现？

首先客户端发送明文请求；服务端会生成一对公钥和密钥，并把证书返回给客户端；客户端会验证证书合法性，若合法生成一个随机数并用证书加密给服务端；服务端用密钥解密后对该随机数使用对称加密算法生成一个密钥，并加密随机数后反回给客户端；客户端使用随机数和加密内容分析得到对称加密密钥。至此对外仅暴露了公钥。

### Ping ICMP

作用：用来检测网络中主机之间的连通性和延迟

原理：通过发送ICMP查询报文向目标主机发送请求响应，若成功则正确显示结果，否则向源主机发送ICMP差错报文显示错误类型

### TCP

1. 三次握手

三次握手中第一次是客户端发送一个想要建立连接的请求，第二次是服务端表示确认客户端的请求但还不知道客户端的态度，服务端进入**SYN\_RECV** ；第三次才能真正确认双方都以及建立，客户端和服务端都进入**ESTABLISHED**。

Syn标志位代表着同步，用于建立连接阶段；ACK表示确认号代表下一条待接受序号，确保不丢失任何数据；seq序列号，标识tcp报文段顺序。

第一次握手只携带syn是因为没有建立连接前没有序号。第二次握手syn表示服务端也想建立连接，ack用来确认这个连接建立报文段回应的是是一个syn请求。第三次握手可以携带数据。

1. 四次挥手

第一次挥手客户端发送一个FIN标志报文段，客户端进入FIN-WAIT-1状态；

第二次挥手服务端发送一个对客户端FIN标志报文段的ACK确认报文，服务端进入CLOS-WAIT，客户端进入FIN-WAIT-2。

第三次挥手服务端发送FIN标志报文段，服务端进入LAST-ACK.

第四次客户端发送ACK，客户端进入TIME-WAIT，并等待2MSL时间后没有收到回复就代表服务端已经进入close状态，客户端也可以关闭连接了。

MSL表示最大报文生存时间，表示一个tcp报文在网络中能存在的最长时间。

**为什么不是一个MSL？这是因为检测到ACK丢失就花了一个MSL,选择2个msl时间是统计规律得到的，比如在一个丢包率为1/100的网络中连续两次丢包的概率只有万分之一。**

1. **可靠传输 （保证按序到达的同时不会出错，同时考虑到传输速率）**

原理：**1.**TCP协议将应用层数据划分为一个个报文段，对每个段赋予一个序列号，那么接收端就可以根据序列号判断下一个待接收报文段和去重操作。**2.校验和**用来确保数据在传输过程中没有受到修改，校验和是通过首部和数据计算得到的。**3.重传机制**：在数据包丢失或延迟的情况下，重新发送数据。**4.流量控制：**接收方和发送方都有一个固定大小的缓存区，用于匹配不同数据传输速率的主机。接收端只允许发送端发送接受端缓存区能接纳的数据。**5.拥塞控制：**目的是为了解决网络拥塞时，减少数据的发送。TCP在发送数据时需要同时考虑接收端的缓存区待接受数据的能力和网络拥塞程度。发送端发送数据的大小时滑动窗口和拥塞窗口的最小值

1. 如何实现流量控制？

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。发送端的发送速率由接收端的缓存区大小和数据处理效率控制，发送端发送的数据来不及处理就会被暂存在接受缓存区中，直到接受缓存区最左边数据接收到后才会向右滑动。**接收窗口的大小是根据接收端处理数据的速度动态调整的。**

1. 拥塞控制如何实现？

拥塞控制是为了应对不同的网络拥塞状况而动态调整发送端拥塞窗口的大小。每次进行数据传输时，都有一个初始化阈值，先采用慢开始算法，即每次都拥塞窗口x2大小，直到达到阈值之后开启拥塞避免算法每个RTT时间增长1个大小。若在传输过程中出现超时情况（出现数据报丢失情况），那说明网络状况很糟糕需要慢开始算法并设置阈值为当前的一半。若连续三次收到重复ACK说明接收端想要的数据报丢失了因此将拥塞窗口设置为当前一半并采用拥塞避免算法。

1. ARQ自动重传请求协议

ARQ是数据链路层和传输层采用的错误纠正协议之一，通过确认和重传机制实现在不可靠服务基础上的可靠传输，包括停止等待ARQ和连续ARQ。

停止等待ARQ就是每次发送一个分组，并在规定时间内等待分组的确认，超时则自动重传。

连续ARQ协议类似于TCP的滑动窗口，采用累计确认的方式，提高信道利用率。

### ARP （MAC、ARP表、广播问询、单播响应）

在解释ARP协议之前，必须了解什么是MAC地址，MAC地址就是物理地址，网络中的每一个设备都只有一个物理地址类似于身份证，而ip是虚拟地址用于定位主机的局域网，那么一个IP报在物理层面上传输过程总是需要知道下一跳的位置，因此需要用到ARP协议。ARP协议是地址解析协议，负责网络层地址和数据链路层地址之间的转换问题。ARP只能分析在从源地址到目标地址路径中下一跳的位置，具体来说，源主机发送一个IP报时首先会查本地ARP表中是否有对应IP地址的下一跳MAC地址，有就直接发，没有就发送一个广播地址，局域网内的所有设备都会收到这个分组并检查，无效则弃之，有效则会分析IP报中的源mac地址，并采用单播响应，这样源主机就知道下一跳的位置。

### NAT网络地址转换协议（解决IPV4地址不够问题）

NAT协议又叫网络地址转换协议，用来转换私有网络与公有网络之间转换ip地址，实现多个主机共享一个公网ip来解决ipv4地址不够问题。

实现方式有三种：

1. 静态NAT

一个内部主机占用一个公网ip，这样做对于节约公网ip没有意义，主要是为了防止内网ip直接暴露给外网。

1. 动态NAT

路由器内部有一个公网ip地址池，一旦内部主机有需求时就会分配，使得内网ip映射到公网ip

1. NAPT

实现了一个公网ip对应多个内部ip，其通过公网ip＋端口号映射内网ip