1 各层协议

- OSI七层协议：应用层、表示层、会话层、运输层、网络层、数据链路层和物理层。多用于理论。
- TCP/IP四层协议：应用层、运输层、网络层、网络接口层 （数据链路层和物理层合并）。实现中使用的体系。 -表示层和会话层在七层协议中是交给程序开发者去实现的。

1.1 TCP/IP各层具体协议

（1）网络接口层： 主要是物理层的一些接口，比如电缆等； （2）网络层： 提供独立于硬件的逻辑寻址，实现物理地址与逻辑地址的转换； 在TCP/IP协议族中，网络层协议包括：IP协议（网际协议），ICMP协议（Internet互联网控制报文协议），以及 IGMP协议（Internet组管理协议）。 （3）传输层： 为网络提供流量控制，错误控制和确认服务； 包括TCP（传输控制协议）和UPD（用户数据包协议）。 （4）应用层：

1.2 TCP/IP的特点

2 以太网的特点

2.1 帧结构

3 集线器、交换机、路由器的作用及所属层

4 IP数据报常见字段

5 ARP协议的作用

6 浏览器输入URL后打开网页的过程

6.1 DNS域名解析

浏览器DNS查询过程

- 本地解析：包括本地浏览器缓存查找对应IP；本地host文件查询对应IP；本地路由器DNS查询IP；
- DNS服务器解析：ISP DNS检查缓存；ISP DNS进行递归查找，从根域名服务器开始。 然后返回给浏览器，浏览器再将 IP 地址打在协议上，同时请求参数也会在协议搭载，然后一并发送给对应的服务器。

6.2 与目的主机进行TCP连接（三次握手）

- 第一次握手：建立连接时，客户端A发送SYN=1（同步序列编号），Seq=X序号给服务器，客户端进入SYN_SENT状态，等待客户端确认；
- 第二次握手：服务器端发送SYN=1，Ack=X+1（确认序号编号）,Seq=Y发送给客户端，服务器进入SYN_RCVD状态；（此时称为半连接）
- 第三次握手：客户端收到确认，检查Ack是否等于X+1，设置Ack = Y+1发送给服务器，服务器检查Ack是否为Y+1，然后建立连接开始传送数据。
- “三次握手”的目的是“为了防止已失效的连接请求报文段突然又传送到了服务端，因而产生错误”：例如A发送了一个请求报文，由于在某个环节延迟了，已经失效的报文在很久之后传送到了服务端，服务端进行确认，如果没有三次握手，那么服务端就开始建立连接，等待客户端发送数据，但是客户端没有发送有效的请求所以不会发送数据，就造成了服务端等待浪费资源。

6.3 浏览器通过HTTP/HTTPS协议发送请求

6.4 服务器处理请求

发送与收取数据（浏览器与目的主机开始HTTP访问过程）

6.5 浏览器获得页面数据

6.6 与目的主机断开TCP连接（四次挥手）

- 第一次挥手：浏览器所在主机向服务器发出连接释放报文，然后停止发送数据；
- 第二次挥手：服务器接收到释放报文后发出确认报文，然后将服务器上未传送完的数据发送完；
- 第三次挥手：服务器数据传输完毕后，向客户机发送连接释放报文；
- 第四次挥手：客户机接收到报文后，发出确认，然后等待一段时间后，释放TCP连接。

6.7 为什么连接的时候是三次握手，关闭的时候却是四次握手

因为在建立连接时，服务端在收到SYN请求后，可以直接发送SYN+ACk，其中SYN是用来同步，ACK是用来应答的。但在关闭连接时，服务端在收FIN报文时，很可能并不会立刻停止传送数据，所以先发送ACK应答，等到服务端数据传送完毕后再发送FIN报文。

6.8 TIME_WAIT状态

按理四次挥手的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，有可以最后一个ACK丢失。所以TIME_WAIT状态就是用来重发可能丢失的ACK报文。在Client发送出最后的ACK回复，但该ACK可能丢失。Server如果没有收到ACK，将不断重复发送FIN片段。Client会设置一个计时器，

等待2MSL的时间。如果在该时间内再次收到FIN，那么Client会重发ACK并再次等待2MSL。所谓的2MSL是两倍的MSL(Maximum Segment Lifetime)。MSL指一个片段在网络中最大的存活时间，2MSL就是一个发送和一个回复所需的最大时间。如果直到2MSL，Client都没有再次收到FIN，那么Client推断ACK已经被成功接收，则结束TCP连接。

7 HTTPS和HTTP的区别

- http是超文本传输协议，信息是明文传输，https 则是具有安全性的ssl加密传输协议；
- http和https使用的是完全不同的连接方式用的端口也不一样，前者是80，后者是443；
- http的连接很简单，是无状态的。
- HTTPS协议是由SSL+HTTP协议构建的可进行加密传输、身份认证的网络协议，要比http协议安全。
- Https采用对称加密和非对称加密结合方式 对称加密：加密和解密都是同一个密匙。 非对称加密：密钥成对出现，分为公钥和私钥，公钥加密需要私钥解密，私钥加密需要公钥解密。