## TCP



三次握手前，服务端的状态从CLOSED变为LISTEN, 同时在内部创建了两个队列：半连接队列和全连接队列，即SYN队列和ACCEPT队列。

**半连接队列：**当客户端发送SYN到服务端，服务端收到以后回复ACK和SYN，状态由LISTEN变为SYN\_RCVD，此时这个连接就被推入了SYN队列，也就是半连接队列。

**全连接队列**：当客户端返回ACK, 服务端接收后，三次握手完成。这个时候连接等待被具体的应用取走，在被取走之前，它会被推入另外一个 TCP 维护的队列，也就是全连接队列(Accept Queue)。

**SYN FLOOD：**用客户端在短时间内伪造大量不存在的 IP 地址，并向服务端疯狂发送SYN。服务端处理大量的SYN包并返回对应ACK, 势必有大量连接处于SYN\_RCVD状态，从而占满整个**半连接队列**，无法处理正常的请求。由于是不存在的 IP，服务端长时间收不到客户端的ACK，会导致服务端不断重发数据，直到耗尽服务端的资源。

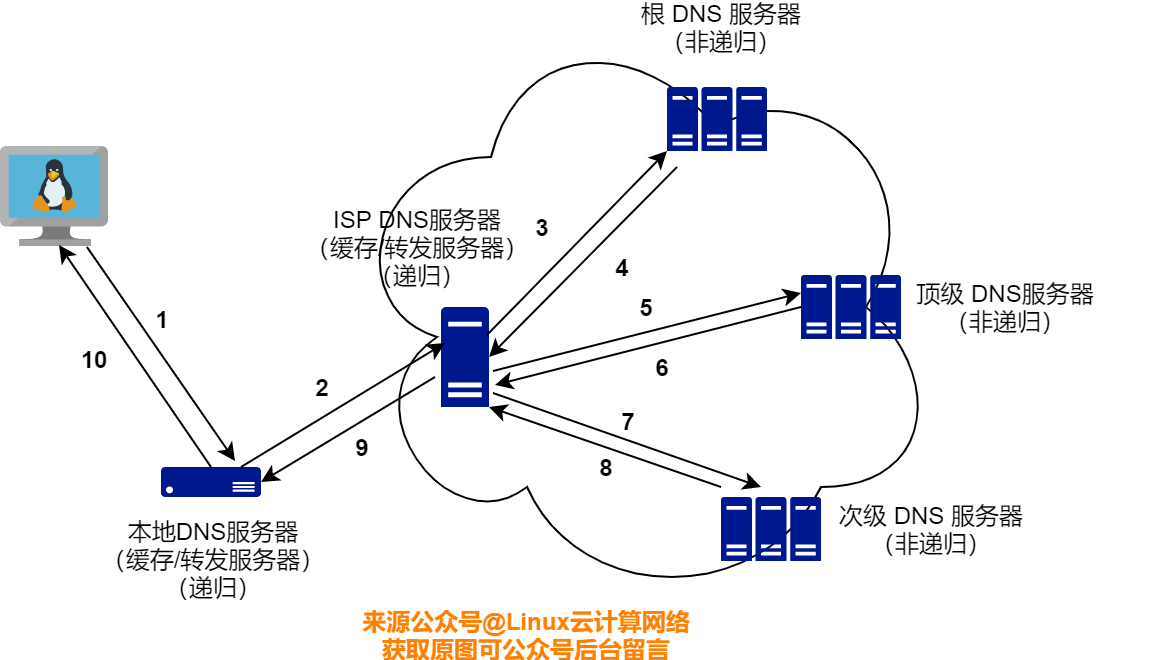
可选项：TimeStamp、MSS、SACK、window scale

**Keep-alive:** sudo sysctl -a | grep keepalive

## DNS

C:\Windows\System32\drivers\etc\hosts

/etc/hosts



**本地 DNS 服务器** ，特指内网的 DNS 服务器，提供内网主机之间的域名查询服务，一般作为缓存/转发之用。/etc/hosts 和 /etc/resolv.conf ，前者记录了内网主机 hostname 和 IP 之间的映射关系，后者记录了外网的 DNS 服务器地址。

**ISP DNS 服务器**

**公共 DNS 服务器（**114.114.114.114**）**

**递归查询**：一层接一层地去查询，直到查询到最终的结果之后才返回

**迭代查询**：每一层查询完之后都会返回，返回的结果是下一层 DNS 服务器的地址，根据返回的结果，接着去查询下一层，直到查询到最终的结果为止

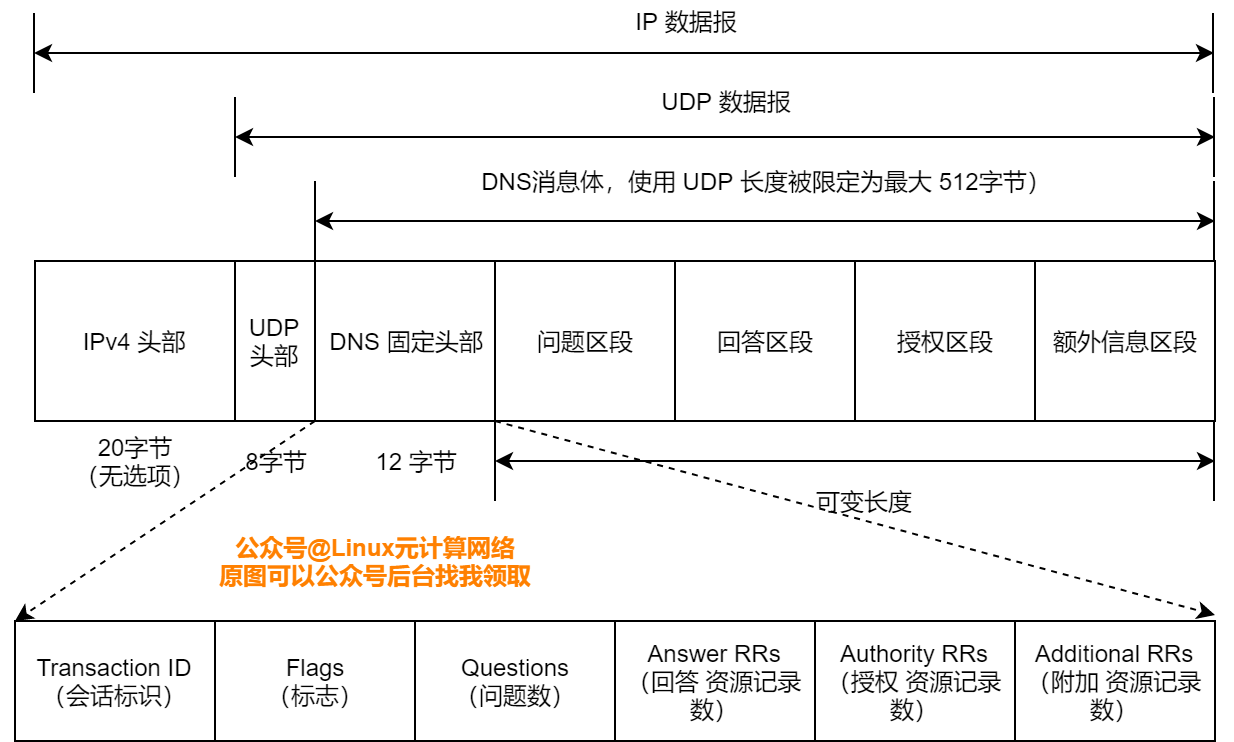
**DNS缓存**：浏览器缓存、系统缓存、本地DNS缓存（硬件）、路由器缓存、ISP缓存

**清除DNS缓存**：

Windows: ipconfig /flushdns

Linux：systemctl restart dnsmasq.service

DNS 是用户层协议，传输层主要使用 UDP 协议，只有大包才可能使用 TCP 协议。 DNS 报文由 12 字节固定长度的首部和 4 个长度可变的字段组成。使用 UDP 时，整个 DNS 报文长度限定为 512 字节，如果使用 TCP 或者扩展域，DNS 消息体的长度可以进行扩展，并在传输时借助 TCP 来分段。



**DNS Cache 污染**：中间 DNS 服务器将上级服务器返回结果进行修改，给客户端一个改变了（污染）的信息

DNS 劫持：和 DNS Cache 污染比较像，不过这种可以被外部第三者劫持进行恶意修改。针对 DNS 劫持，我们可以简单地更换域名服务器，比较靠谱的一个是Google 提供的 8.8.8.8。

**DNS 欺骗**：用一个假的 DNS 应答来欺骗用户计算机，让其相信这个假的地址，并且忽略真正的 DNS 应答

**DNS 放大攻击**：这是一种 DDoS 攻击， 利用 DNS 回复包比请求包大的特点，放大流量，伪造请求包的源 IP 地址为受害者 IP，将应答包的流量引入受害的服务器

**DNS 负载均衡器**：dig baidu.com

DNS工具：yum install bind-utils

dig selfboot.cn +trace @8.8.8.8

nslookup

host baidu.com

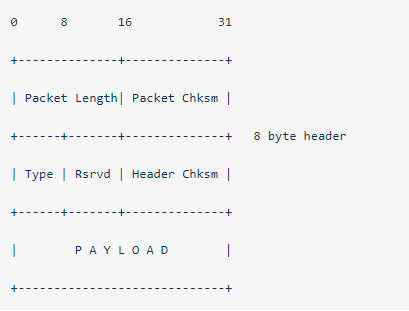
host 39.156.69.79 或 dig -x

whois coolshell.cn

## TNS

TNS(Transparent Network Substrate) 协议用于客户端连接Oracle数据库，它可以使用其他一些协议进行通信，如：TCP/IP, IPX/SPX, IPC, Named Pipes等。

TNS 包由一个header和payload 组成



Type:

|  |  |
| --- | --- |
| 1 | Connect |
| 2 | Accept |
| 3 | ACK |
| 4 | Refuse |
| 5 | Redirect |
| 6 | Data |
| 7 | NULL |
| 8 | ---- |
| 9 | ABORT |
| 10 | ---- |
| 11 | Resend |
| 12 | Marker |
| 13 | Attention |
| 14 | Control |

## SSL

第一步，客户端给出协议版本号、一个客户端生成的随机数（Client random），以及客户端支持的加密方法。

第二步，服务器确认双方使用的加密方法，并给出数字证书、以及一个服务器生成的随机数（Server random）。

第三步，客户端确认数字证书有效，然后生成一个新的随机数（Premaster secret），并使用数字证书中的公钥，加密这个随机数，发给服务器。

第四步，服务器使用自己的私钥，获取客户端发来的随机数（即Premaster secret）。

第五步，两端根据约定的加密方法，使用前面的三个随机数，生成"对话密钥"（session key），用来加密接下来的整个对话过程。

生成私钥：openssl genrsa -out [www.key](http://www.key) 4096

生成csr(证书签名请求)：openssl req -new -key [www.key](http://www.key) -config [www.cnf](http://www.cnf) -out [www.csr](http://www.csr)

证书与私钥组合：cat [www.crt](http://www.crt) [www.key](http://www.key) > www.pem

CA为证书颁发签名，公钥进行验签证实证书合法

## telnet

1、本地计算机上必须装有包含Telnet协议的客户程序；必须知道远程[主机](https://baike.baidu.com/item/%E4%B8%BB%E6%9C%BA" \t "_blank)的Ip地址或[域名](https://baike.baidu.com/item/%E5%9F%9F%E5%90%8D" \t "_blank)；必须知道登录标识与口令。

Telnet 协议的主体由三个部分组成：网络虚拟终端NVT 、操作协商定义、协商有限自动机

2、 telnet IP 端口 或 telnet 域名 端口

## 巨帧



## Vlan



## Oracle

yum install -y oracle-database-preinstall-19c-1.0-1.el7.x86\_64.rpm

yum install -y oracle-database-preinstall-19c-1.0-1.el7.x86\_64.rpm

/etc/init.d/oracledb\_ORCLCDB-19c configure

vi /etc/profile.d/oracle.sh

export ORACLE\_HOME=/opt/oracle/product/18c/dbhome\_1

export PATH=$PATH:/opt/oracle/product/18c/dbhome\_1/bin

export ORACLE\_SID=ORCLCDB

su - oracle

sqlplus / as sysdba

alter user system identified by system

show pdbs

sqlplus.exe [system/system@10.0.0.181:1521/ORCLPDB1](mailto:system/system@10.0.0.181:1521/ORCLPDB1)

database = cx\_Oracle.connect('system/system@192.168.4.181:1521/ORCLPDB1')

cursor = database.cursor()

cursor.execute("create table METADATA (CZRKRYID varchar2(20),CZRKQFRQ date,CZRKSDZP blob,CZRKGMSFHM varchar2(18),SSGAJGJGDM varchar2(12),CCC\_DX\_ETL\_TIME number)")

cursor.execute("select table\_name from user\_tables")

cursor.fetchall()

cursor.execute("select \* from METADATA")

cursor.close ()    
conn.close ()

## DHCP

常用的2个端口：67(DHCP server),68(DHCP client)

DHCP DISCOVER ：客户端开始DHCP过程发送的包，是DHCP协议的开始

DHCP OFFER ：服务器接收到DHCP DISCOVER之后做出的响应，它包括了给予客户端的IP（yiaddr）、客户端的MAC地址、租约过期时间、服务器的识别符以及其他信息

DHCP REQUEST ：客户端对于服务器发出的DHCP OFFER所做出的响应。在续约租期的时候同样会使用。

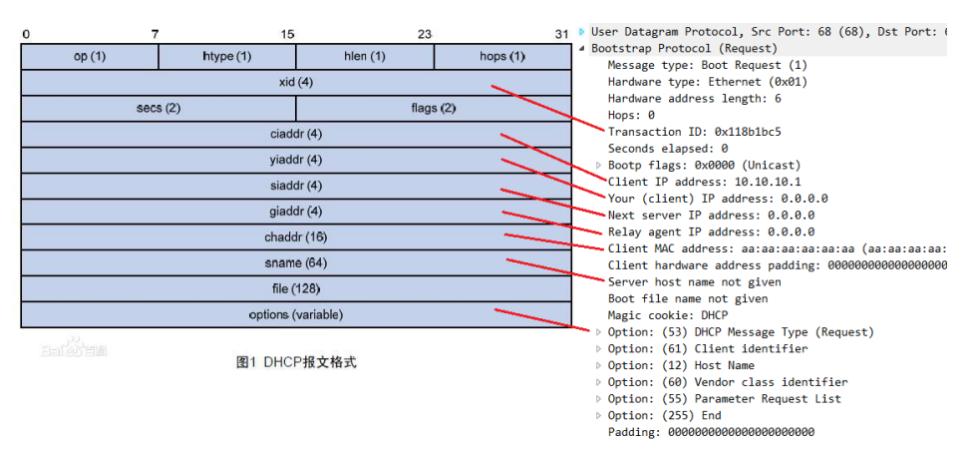
DHCP ACK ：服务器在接收到客户端发来的DHCP REQUEST之后发出的成功确认的报文。在建立连接的时候，客户端在接收到这个报文之后才会确认分配给它的IP和其他信息可以被允许使用。

DHCP NAK ：DHCP ACK的相反的报文，表示服务器拒绝了客户端的请求。

DHCP RELEASE ：一般出现在客户端关机、下线等状况。这个报文将会使DHCP服务器释放发出此报文的客户端的IP地址

DHCP INFORM ：客户端发出的向服务器请求一些信息的报文

DHCP DECLINE :当客户端发现服务器分配的IP地址无法使用（如IP地址冲突时），将发出此报文，通知服务器禁止  
使用该IP地址。

  
Op：Request（1） 和 Reply（2）   
HW Type： 硬件类型，一般是以太网：1   
HW Len： 硬件地址长度，单位字节。对应以太网：6（mac地址长度为6字节48bit）   
Transaction ID：事务ID，随机数，有客户端生成，服务器Reply时，会把Request中的Transaction拷贝到Reply报文中。   
Secs： 距离第一次发射IP请求或Renew请求过去的秒数   
Flags：标志位，目前仅第一个bit有使用，置1 标明广播   
Client IP Address：当前客户端的IP地址，如果当前客户端没有IP地址，则置0   
Your IP Address： 服务器想客户端提供IP地址时，会把IP地址填入本字段   
Next Server IP Address：客户端引导时需要的另一个服务器的IP地址   
Gateway （Relay） IP Address： 网关（中继）IP地址，有DHCP 中继器在转发DHCP报文的时候填入   
Server Name： Server名字，有64bytes，一般不使用，填充为0   
Boot File name： boot file的路径，128bytes， 一般不使用，填充为0   
Option： 选项，不定长度。 DHCP报文中比较重要的字段，后面会有比较详细的介绍。

**静态租约表 动态租约表**

## DHCPv6

消息类型：

Solicit：DHCPv6客户端使用Solicit报文来确定DHCPv6服务器的位置。

Advertise：DHCPv6服务器发送Advertise报文来对Solicit报文进行回应，宣告自己能够提供DHCPv6服务。

Request：DHCPv6客户端发送Request报文来向DHCPv6服务器请求IPv6地址和其它配置信息。

Confirm：DHCPv6客户端向任意可达的DHCPv6服务器发送Confirm报文检查自己目前获得的IPv6地址是否适用与它所连接的链路

Renew: DHCPv6客户端向给其提供地址和配置信息的DHCPv6服务器发送Renew报文来延长地址的生存期并更新配置信息。

Rebind: 如果Renew报文没有得到应答，DHCPv6客户端向任意可达的DHCPv6服务器发送Rebind报文来延长地址的生存期并更新配置信息。

Reply: DHCPv6服务器发送携带了地址和配置信息的Reply消息来回应从DHCPv6客户端收到的Solicit、Request、Renew、Rebind报文。DHCPv6服务器发送携带配置信息的Reply消息来回应收到的Information-Request报文。用来回应DHCPv6客户端发来的Confirm、Release、Decline报文。

Release: DHCPv6客户端向为其分配地址的DHCPv6服务器发送Release报文，表明自己不再使用一个或多个获取的地址。

Decline: DHCPv6客户端向DHCPv6服务器发送Decline报文，声明DHCPv6服务器分配的一个或多个地址在DHCPv6客户端所在链路上已经被使用了。

Reconfig: DHCPv6服务器向DHCPv6客户端发送Reconfigure报文，用于提示DHCPv6客户端，在DHCPv6服务器上存在新的网络配置信息。

Information-request: DHCPv6客户端向DHCPv6服务器发送Information-Request报文来请求除IPv6地址以外的网络配置信息。

Relay-forw: 中继代理通过Relay-Forward报文来向DHCPv6服务器转发DHCPv6客户端请求报文。

Relay-repl: DHCPv6服务器向中继代理发送Relay-Reply报文，其中携带了转发给DHCPv6客户端的报文。

