运维网络基础

1. 什么是网络?

```

所谓网络，就是通过一定的形式连接起来的物体，物体与物体之间可以实现通信。
用什么连接:
    网线
        网线分类: 五类线 超五类 六类线(千兆) 按箱子305米
```

```
        水晶头 橙白 橙 绿白 蓝 蓝白 绿 棕白 棕
568B线序(568A线序)
        8芯线: 4芯 1236芯 用来传输数据 其他4芯没
用(额外供电 反向供电 POE供电)
        传输距离100米(超过100米需要加信号放大器 交
换机 HUB)
    光纤 千兆万兆
        多模光纤 传输距离近  2公里
        单模光纤 传输距离远  120公里
    wifi

物体于物体:
    计算机-计算机
    计算机-服务器
    服务器-服务器
    计算机-交换机
    计算机-路由器
    交换机-交换机
    路由器-路由器

    物联网: 所有的物体都是由网络进行互通互联
            ip不够用
#网络的重要性

所有的系统都有网络!
我们的生活已经离不开网络。
运维生涯50%的生产故障都是网络故障!

```

1. 如何通过网络实现多台主机之间的通讯

```

1)在两台主机之间需要有传输介质(网线、光纤、无线等)

```

2)在两台主机上面需要有网卡设备
　网卡：全球唯一的地址(MAC)　类似家的位置

　　在发送信息时：将二进制（数字信号）信息转换为高低电压（电信号）　　#调至过程
　　在接收信息时：将高低电压（电信号）信息转换为二进制数（数字信号）　　#解调过程

3)在进行数据传输之前，需要协商网络传输速率。

　　网卡速度参数　100Mbps==100M bit 每秒==每秒钟传输多少个bit（0或1）
　　1M=1000k 100M=100000k
　　1k=1000b 100000k=100000000b

　　网络传输数据单位：比特　bit　　1bit=1/8byte
　　磁盘存储数据单位：字节　byte　　1byte=8bit
网卡的100Mbps同样是以bit/s来定义的 所以100Mb/S＝100000KByte/s=(100000/8)KByte/s=12500KByte/s
在理论上1秒钟可以传输12.5MB的速据 考虑到 干扰的因素 每秒传输只要超过10MB就是正常了 现在出现了1000Mbps的网卡那么速度就是100MB/S
K是千 M是兆 G是吉咖 T是太拉

1Byte(字节)=8bit(位)

1KB=1024Byte(字节)

1MB=1024KB

1024MB=1GB

```
1024GB=1TB

真实的运营商下载带宽: ADSL拨号 用户名和密码 非对称式网络 上传和下载带宽不一样  家用
重启路由器重新获取新的公网IP地址
2M  4M
6M  8M
企业专线: 对称式网络 配置公网IP地址 上传和下载相同 价格翻好几倍 10MB  4-8万 运营商不同
```

2. 网络设备说明介绍

*1. 什么是交换机*3

```
实现一个网络内多台主机之间的通讯

#如何利用交换机实现通讯

    1. 在数据前面设置目标地址和源地址，目标地址和源地址用mac地址进行标识
        mac称为物理地址，每块网卡上都有的一个标识身份信息的
        mac地址全球唯一，不能进行修改，mac地址用16进制标识

    2. 在网络通讯初期，会利用广播方式进行发送数据包，在通讯的过程，数据包的发送一定是有去有回的。
        在一个交换网路中，如果产生了大量广播数据包时会产生广播风暴，影响主机性能，这样的问题称为广播风暴问题

```

```
#解决广播风暴问题思路：

    减少广播产生数量，将一个大的交换网络切割为几个小的交换网络（局域网，广播域）

#交换机的种类

    傻瓜交换机（TP-link/Dlink/水星...）
    4口 8口 12口 16口 24口 48口
    程控交换机（存储程序控制交换机，配置管理，思科、华为、华三、锐捷、中兴、瑞斯康达）
    二层交换  支持数据转发 vlan隔离端口
    三层交换  功能比较多 DHCP ACL访问控制列表 支持路由器功能
OSI七层模型
```

思科交换机硬件设备

2. 什么是路由器

```
实现不同局域网之间主机通讯，可以隔离广播风暴（路由不同的接口连接不同广播域）
```

路由类似于现实生活中从A地去往B地可能需要先步行，在坐车，在做飞机才能到达B地，这样的整个过程在网络中对应数据的传递过程就称为路由。因此一个数据信息跨越不同的网段传递到目的地址，就可以把传递数据的过程称为路由，也可以看做每条传递数据的路径。

1）需要有身份标识信息：ip地址

逻辑地址（可以改变的地址/？？？） 利用10进制方式进行显示

IP地址由两部分组成: 交换网络标识信息+主机地址标识信息===网段地址+主机地址

192.168.13 网段 1-254 主机地址

10.0.0 网段 1-254 主机地址

昌平区 网段 xx地址 192.168.11.x

海淀区 网段 xx地址 192.168.10.x

内网卡---交换机 192.168.11.0/24(192.168.11.1~192.168.11.254)

外网卡---运营商 IP : 221.218.210.53

查公网ip的方法：

windows，打开浏览器，访问百度，搜IP即可

linux：curl ifconfig.me ip.sb

高级路由器还有上网行为管理器和防火墙功能

论坛：鸿鹄论坛(网络工程师)

2）路由实现数据传输通讯时，会根据路由表信息进行数据包路由

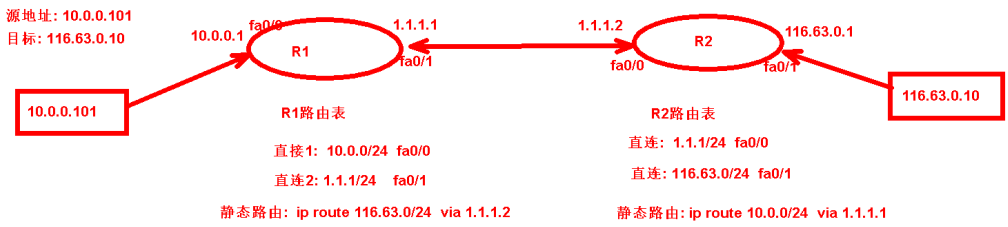
```
    实现不同网段之间通讯需要经过一条必经之路，这条路称为网关

经过了多少跳 经过多少个路由器
下一跳  下一个路由器的接口 下一个路由器IP
我们到百度中间经过了多少跳？ 经过了多少个路由器。

-----------------------------------------------------------
配置通过一个路由器实现让不同网段通信
Router>
Router>?
Exec commands:
  <1-99>      Session number to resume
  connect     Open a terminal connection
  disable     Turn off privileged commands
  disconnect  Disconnect an existing network connection
  enable      Turn on privileged commands
  exit        Exit from the EXEC
  logout      Exit from the EXEC
  ping        Send echo messages
  resume      Resume an active network connection
  show        Show running system information
  ssh         Open a secure shell client connection
  telnet      Open a telnet connection
  terminal    Set terminal line parameters
  traceroute  Trace route to destination
Router>enable       # 进入特权模式 可以简写en
Router#
```

```
51
52 Router#show running-config  # 查看当前路由器所有的配置
53 Router# config t   # 进入到配置模式
54 Router(config)#interface fa0/0  # 可以简写 int fa0/0 进入到这个接口
55 Router(config-if)#
56 Router(config-if)#ip address 10.0.0.1 255.255.255.0  # 配置IP地址为10.0.0.1
57 Router(config-if)#
58 Router(config-if)#no shutdown   # 开启此端口
59 Router(config-if)#int fa0/1     # 切换到fa0/1接口
60 Router(config-if)#
61
62 Router(config-if)#ip add 116.63.0.1 255.255.255.0
63 Router(config-if)#no shut
64
65 Router#
66 Router#show ip route
67      10.0.0.0/24 is subnetted, 1 subnets
68 C       10.0.0.0 is directly connected, FastEthernet0/0
69      116.0.0.0/24 is subnetted, 1 subnets
70 C       116.63.0.0 is directly connected, FastEthernet0/1
71
72 面试题: 如何让不同的网段进行通信?
73 中间加个路由器。
74
75 ------------------------------------------------------------
```

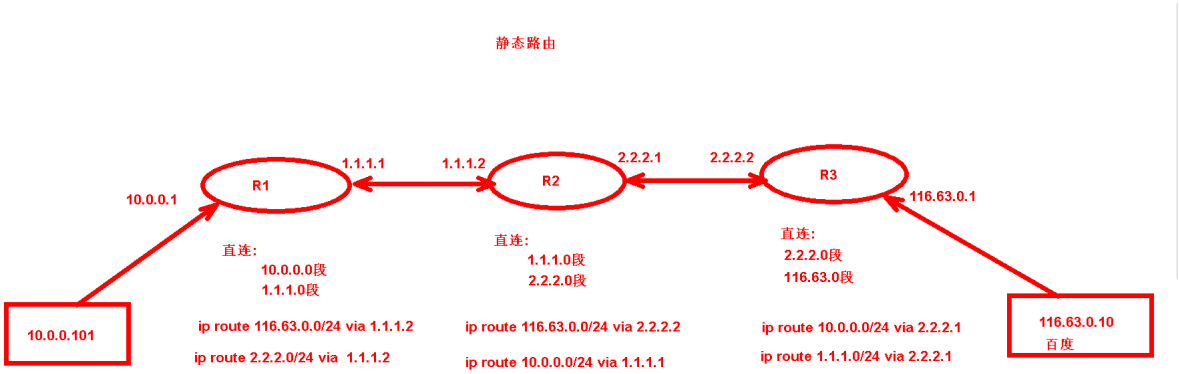
```
两个路由器配置:
R1路由器
Router>enable
Router#
Router#config t
Enter configuration commands, one per line.
End with CNTL/Z.
Router(config)#int fa0/0
Router(config-if)#ip add 10.0.0.1
255.255.255.0
Router(config-if)#no shut
Router(config-if)#no shutdown

Router(config-if)#int fa0/1
Router(config-if)#ip add 1.1.1.1
255.255.255.0
Router(config-if)#no shut


查看R1路由器路由表
Router#show ip route
     1.0.0.0/24 is subnetted, 1 subnets
C       1.1.1.0 is directly connected,
FastEthernet0/1
     10.0.0.0/24 is subnetted, 1 subnets
C       10.0.0.0 is directly connected,
FastEthernet0/0
Router#
Router(config)#ip route 116.63.0.0
255.255.255.0  1.1.1.2


```

```
R2路由器配置
Router(config)#int fa0/0
Router(config-if)#ip add 1.1.1.2
255.255.255.0
Router(config-if)#no shut
Router(config-if)#no shutdown

Router(config-if)#int fa0/1
Router(config-if)#ip add 116.63.0.1
255.255.255.0
Router(config-if)#no shut
Router(config)#ip route 10.0.0.0
255.255.255.0 1.1.1.1


--------------------------------------------
--------------------
```

静态路由 手动指定路由信息到目标地址



三个路由器配置:




```
修改R2路由器配置:
Router(config)#int fa0/1
Router(config-if)#no ip address   # 删除接口的IP地址
Router(config-if)#ip add 2.2.2.1 255.255.255.0
Router(config-if)#no shut
Router(config)#ip route 116.63.0.0 255.255.255.0  2.2.2.2



R3路由器配置
Router>enable
Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#int fa0/0
Router(config-if)#ip add 2.2.2.2 255.255.255.0
Router(config-if)#no shut
Router(config-if)#int fa0/1
Router(config-if)#ip add 116.63.0.1 255.255.255.0
Router(config-if)#no shut
```

```
Router(config)#ip route 10.0.0.0 255.255.255.0 2.2.2.1

```

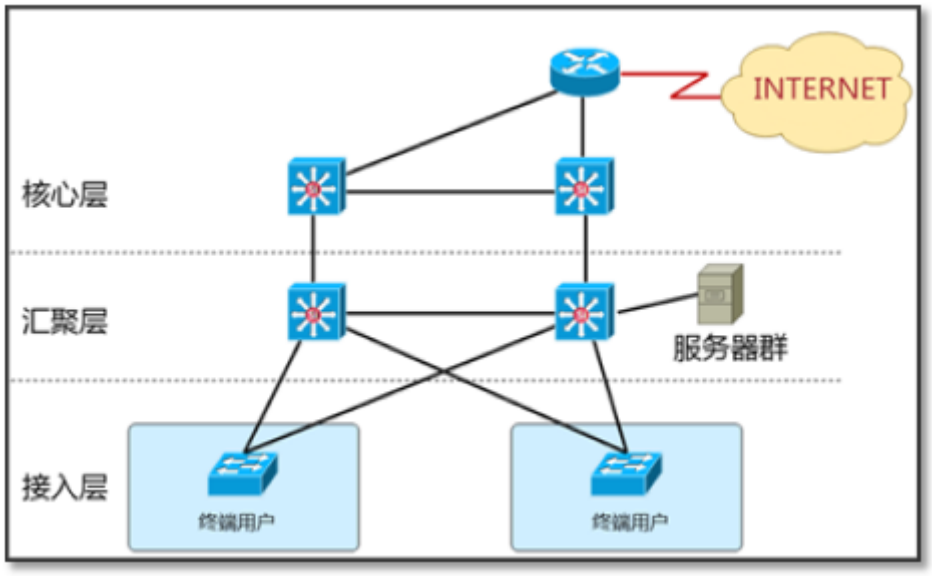
```
动态路由(路由自动学习过程)
RIP OSPF EIGRP ISIS BGP
来老男孩学习(协议 RIP)
R1 张前龙:  金融 厨师 将我会的技能告诉相邻的同桌(路由器)我会什么  经理 PS  艺术 直播
R2 王亚楠:  经理 PS  自动从R1路由器学习了金融和厨师技能，告诉相邻的路由我会什么 经理 PS  金融 厨师 艺术直播
R3 黄雅萍   艺术 直播 自动从R2路由器学习到了 经理 PS 金融 厨师  告诉相邻的路由器我会什么 艺术 直播



R1路由器配置:
第一步: 启动协议
Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#route rip          # 进入到rip协议

第二步: 宣告自己的网段

```

```
Router(config-router)#network 10.0.0.0
Router(config-router)#network 1.1.1.0

R2路由器配置

Router#config t
Enter configuration commands, one per line.
End with CNTL/Z.
Router(config)#route rip
Router(config-router)#network 1.1.1.0
Router(config-router)#network 2.2.2.0

R3路由器配置
Router#config t
Enter configuration commands, one per line.
End with CNTL/Z.
Router(config)#route rip
Router(config-router)#network 2.2.2.0
Router(config-router)#network 116.63.0.0
```

3. 网络层次结构

1. 网络拓扑

```
网络层次结构
        核心层：主要部署路由器设备，用于连接外网线路，还要具备冗余能力
        汇聚层：主要部署三层交换设备，用于相应安全访问控制  进行链路汇聚
        接入层：主要部署二层交换设备，用于终端设备接入

一层交换机：只支持物理层协议。
二层交换机：支持物理层和数据链路层协议。
三层交换机：支持物理层，数据链路层及网络层协议。

```



基本网络层次划分标准示意图

2. 网络类型

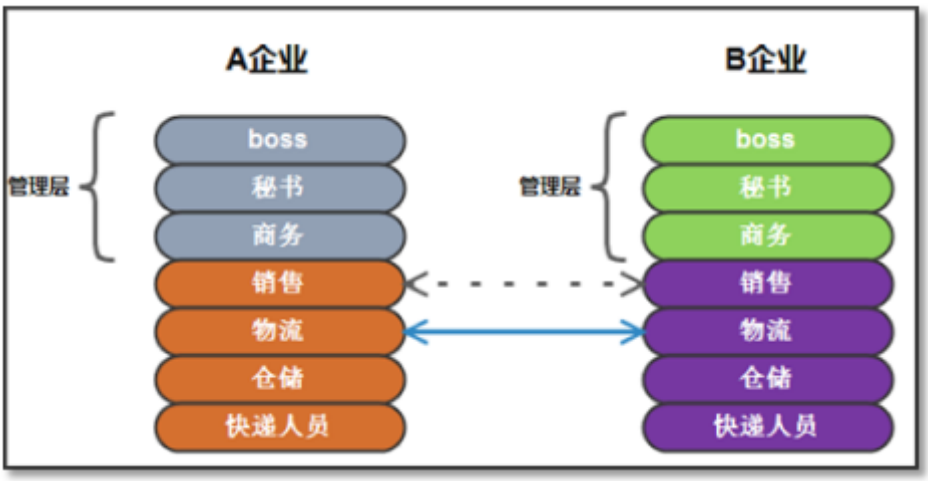
```
局域网：本地私有的一个网络范围。规模较大的局域网，也会称为园区网。
    教室  家庭    公司内  校园
    公网： 全球任意一个可以上网的地方都可以直接访问到

    城域网：网络的覆盖面积达到了一个城市，就可以称为城域网。

    广域网：覆盖面积  达到了全国或全球，就称为广域网，全球最大的广域网就是Internet互联网。

```

3. 网络层次模型（OSI7层模型）

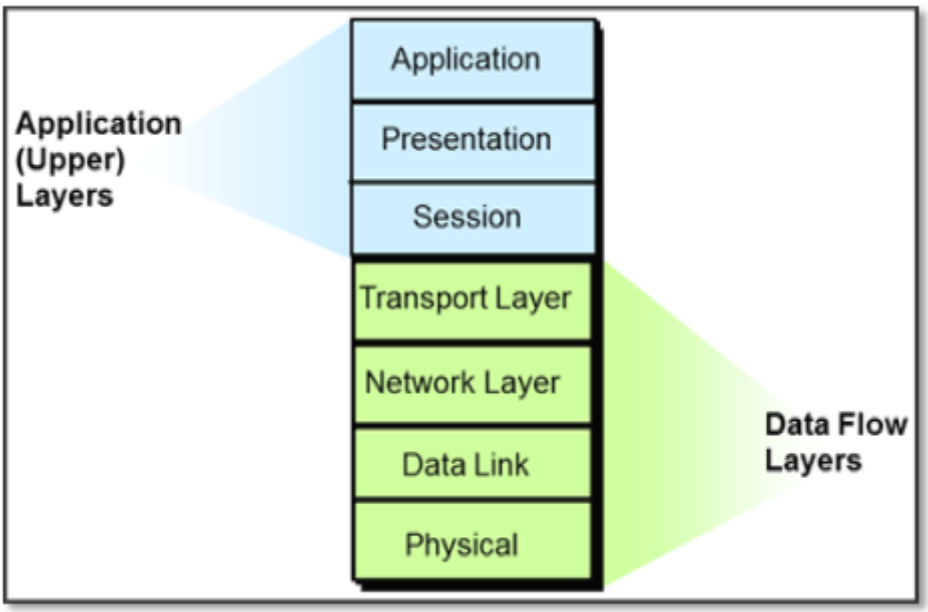
```
1
2
```

OSI是Open System Interconnection的缩写，意为开放式系统互联。国际标准化组织（ISO）制定了OSI模型，该模型定义了不同计算机互联的标准，是设计和描述计算机网络通信的基本框架。OSI模型把网络通信的工作分为7层，分别是物理层、数据链路层、网络层、传输层、会话层、表示层和应用层。

OSI七层模型功能介绍



OSI 7层模型形象比喻示意图

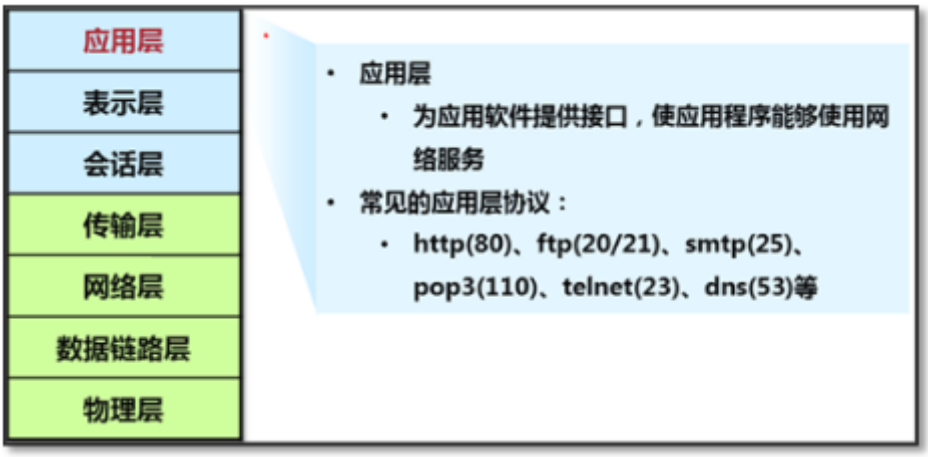


模型结构示意图

4. OSI七层模型详解

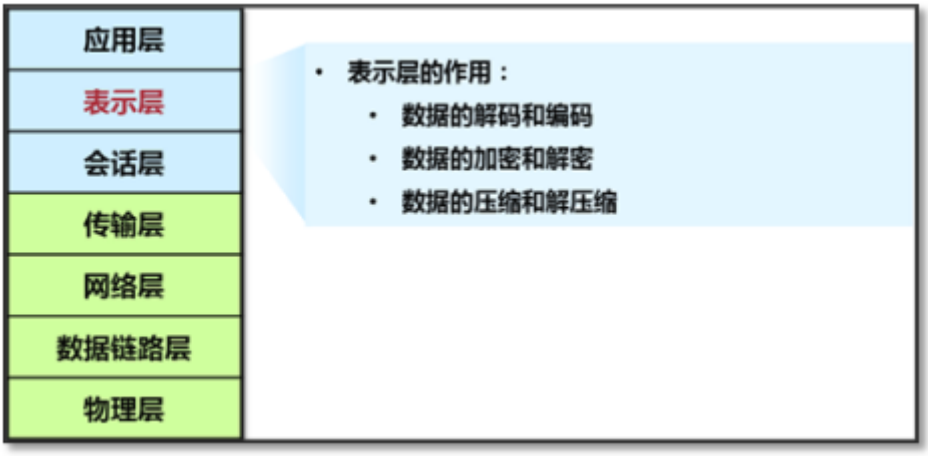
应用层

面试题说一下七层模型都有哪七层?

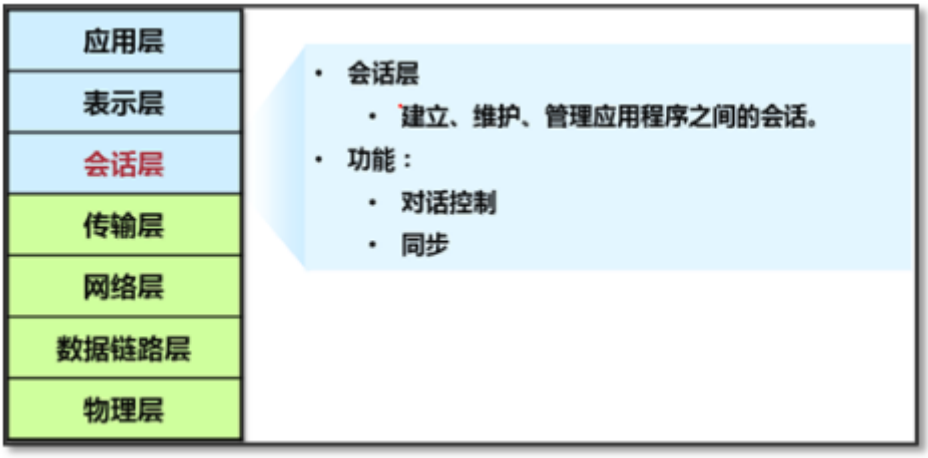



```
1
2 主要就是提供应用程序可以接入网络接口，并根据程序的不同对应不同的接口协议。
3
```

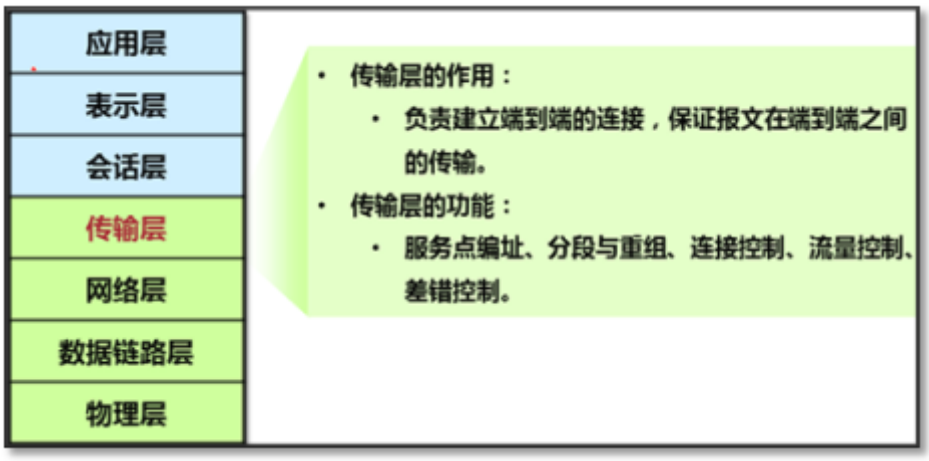
表示层



会话层

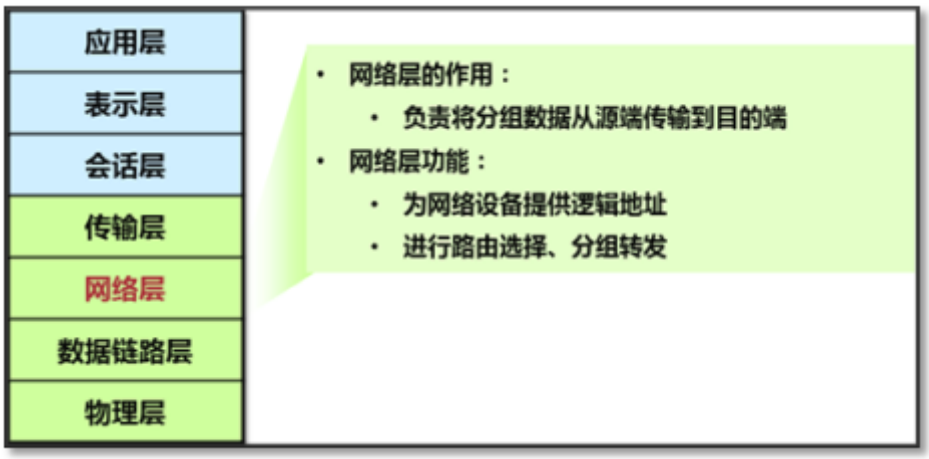


传输层



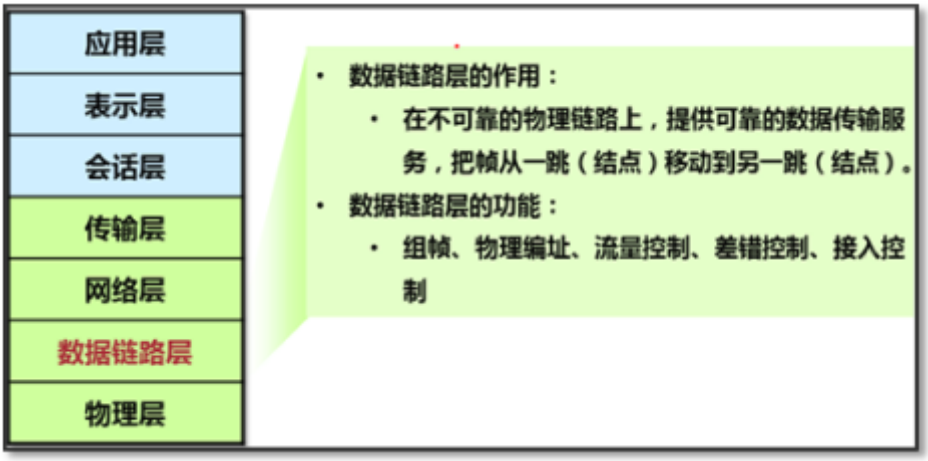

```
1
2 负责网络中端到端的连接（TCP、UDP）。
3
```

网络层

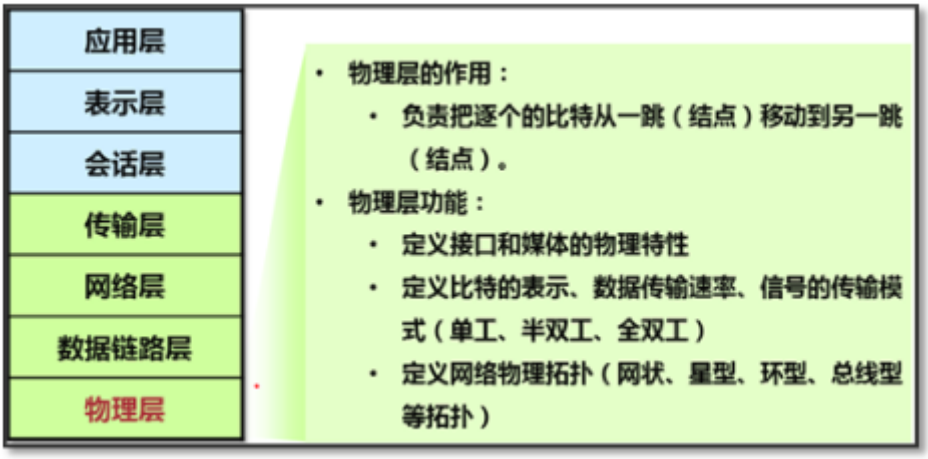


```
1
2  网络层的主要作用就是路由和寻址，主要接触到的是IP协议，即IP地址。
3
```

数据链路层



物理层

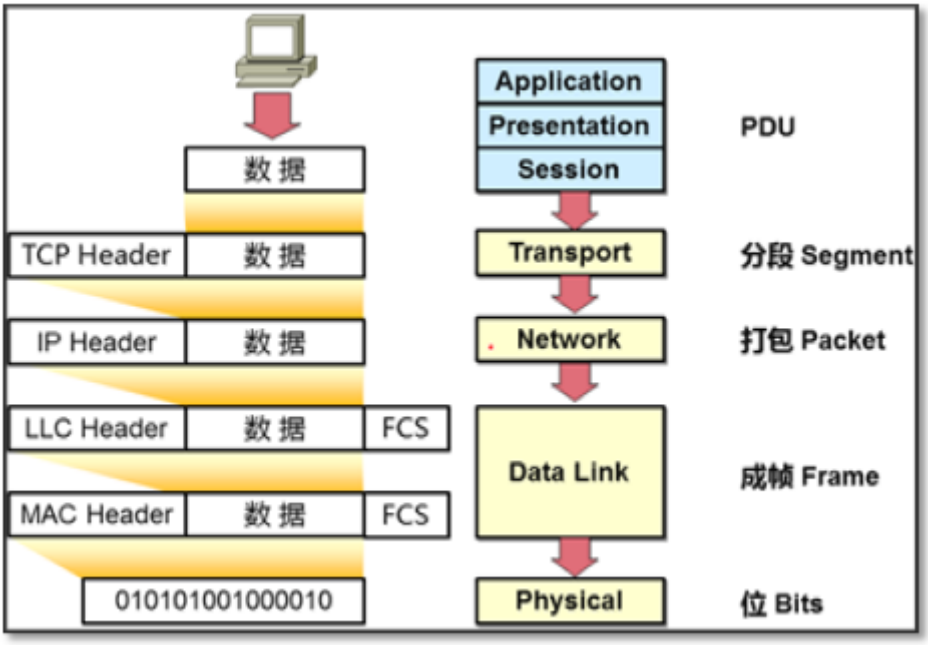


```
Layer1物理层：物理层标准规定了信号、连接器和电缆要求、接口类型、线缆类型、设备（集线器hub）。

单工：  （广播电台）通讯时候。只有一方作为发送方，另一方作为接受方

半双工:（对讲机） 通讯的某一时刻，只有一方作为发送方，另一方作为接受方,通讯时刻发生转变，发送方可以变为接收方，接收方可以变为发送方

全双工:（电话）      同一时刻，双方皆可以是发送方，又可以是接收方

```

4. 数据包封装与解封装

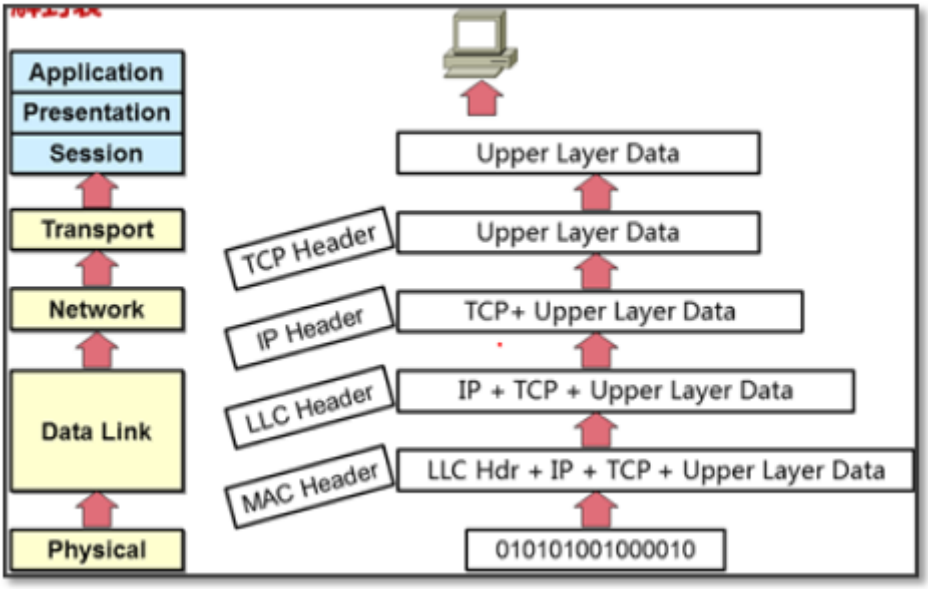
封装

```
封装过程：由上至下进行封装
          应用层、表示层、会话层   PDU  数据
          传输层：分段  TCP协议
          网络层：打包  TCP协议+IP地址
          数据链路层:成帧   TCP协议+IP地址+MAC地址
          物理层：位          数据成为比特

```



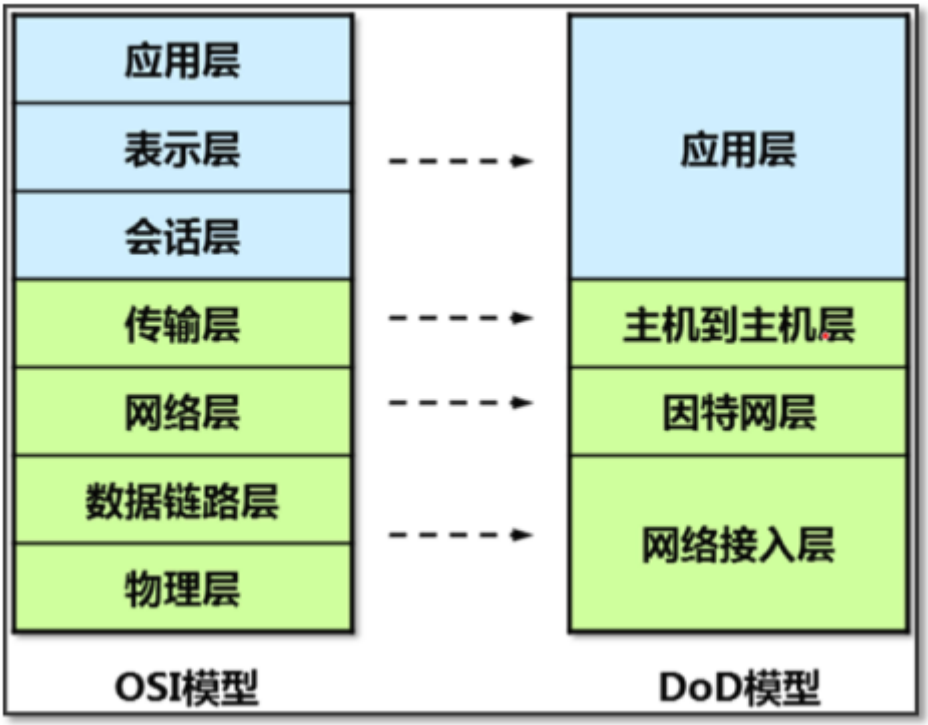
解封装

```
1
2  拆包过程：由下至上进行拆包
3            物理层：位    比特
4            数据链路层：查看MAC地址
5            网络层：查看IP地址
6            传输层：查看TCP协议
7            前三层：数据内容
8
9  不知名端口：1024以上的端口称为不知名端口
10 cat /proc/sys/net/ipv4/ip_local_port_range
11
```

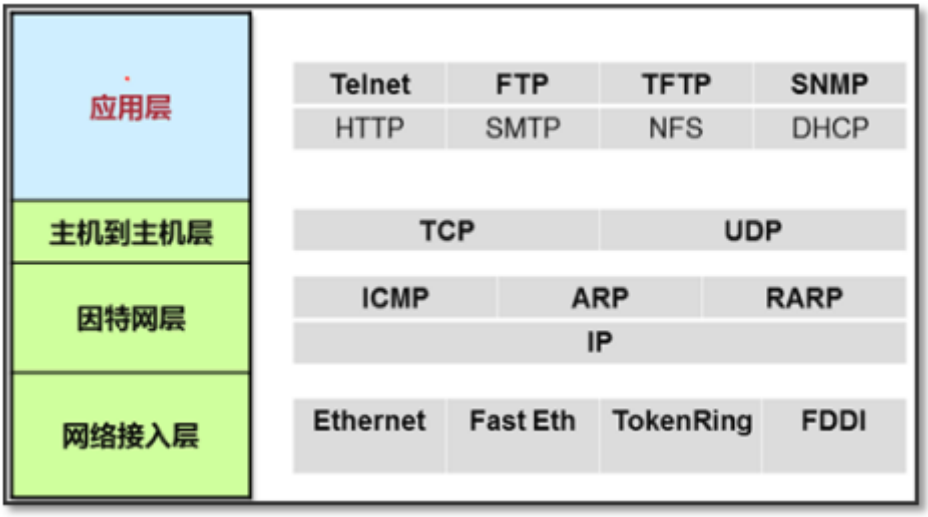


5. TCP/IP模型（4层模型）

1. OSI7层模型与TCP/IP模型（DOD）对应关系



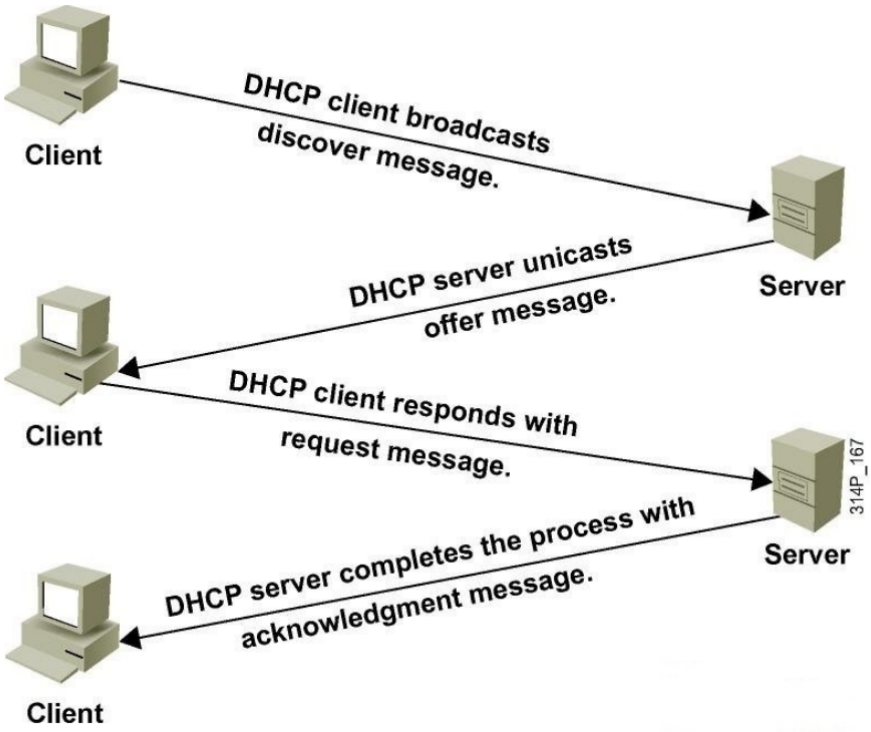
2. TCP/IP协议簇相关协议汇总



3. 应用层协议介绍

```
面试题: 面试官问 ssh默认端口 http默认端口?
FTP             21              明文协议,文件传输协议,基于TCP
TFTP            69              简单文件传输协议，基于UDP
SSH             22              安全外壳协议，远程连接,加密
Telnet          23              明文协议,远程连接
SNMP            161/162         简单网络管理协议，基于UDP
SMTP            25              简单邮件传输协议，基于TCP
HTTP            80              超文本传输协议
HTTPS           443             超文本传输安全协议
DHCP            67/68/546       动态主机设置协议,C(67),S(68),546(V6)

```

DHCP原理图



家用路由器

DHCP设置

```
1
2 DNS称为域名系统，在网站运行中起到了至关重要的作用，主要作用是负责把网站域名解析为对应的IP地址。
3 一般域名提供商，提供的dns服务器，都是走udp53端口的。
4
```

DNS解析过程

域名是什么 ?

```
1
2  举个例子，https://www.baidu.com，这个其实并不是域名，其中https是指协议，去掉https后，www.baidu.com.（注意最后面有一个点号）才是真正的域名。
3  www对应了一个业务 www.baidu.com--->百度的首页
4  news对应了另一个业务 news.baidu.com-->百度的新闻业务
5  域名的第二部分: baidu sina weibo  oldboyedu jd taobao 权威域名。全球唯一 不能冲突
6  域名的第三部分: .com 顶级域名服务器 .cn .org .net .中国 .我爱你
7
8
9
10 每个域名的最后面都有一个点号 "." 表示根域名，为了方便在实际使用的时候被省略了。
11
12 根域名的下一级就是顶级域名了，.com 也就是顶级域名，常见的顶级域名后缀有.com、.cn、.net、.org 等，这些都是固定的，用户不能自己修改，只能选择。
13
14 顶级域名的下一级又是权威域名，如baidu.com中的.baidu，这个权威域名就是我们自己可注册的域名。
15 www.baidu.com www.weibo.com www.linuxnc.com
16 顶级域名下就是主机名了，www是指主机名，这个是我们可以自己定义的，通常在http服务器如nginx中可以修改。
17
18 http https 协议
19 .com .cn .org .net. ...  顶级域名
20 baidu sina weibo linuxnc 权威域名
```

```
www.baidu.com --->显示百度的主页
news.baidu.com--->显示的百度的新闻页面

www.baidu.com的www  news.baidu.com 的news 表示主机名 表示不同的业务 不同的页面

.com.
.cn.
.org.

.是根域名
```

域名解析过程分析

```
下面就还以www.baidu,com这个网址来分析一下dns的解析过程。

当浏览器拿到输入的www.baidu.com后，首先会去浏览器的dns缓存中去查询是否有对应记录，如果查询到记录就可以直接返回ip地址，完成解析。

如果浏览器没有缓存，那就再去查询操作系统的缓存，同样的，如果查询到记录就可以直接返回ip地址，完成解析。

如果操作系统也没有缓存，那就再去查看本地hosts文件，Windows下host文件一般位于"C:\Windows\System32\drivers\etc"。

近几年网上流传的通过修改本地host文件来避免双11女友剁手的段子，其实就是将淘宝的支付接口解析到错误的ip地址，从而导致支付不成功。

```

如果本地host文件也没有相应记录，那就需要求助于本地dns服务器了，所以应该要知道本地dns的ip地址。

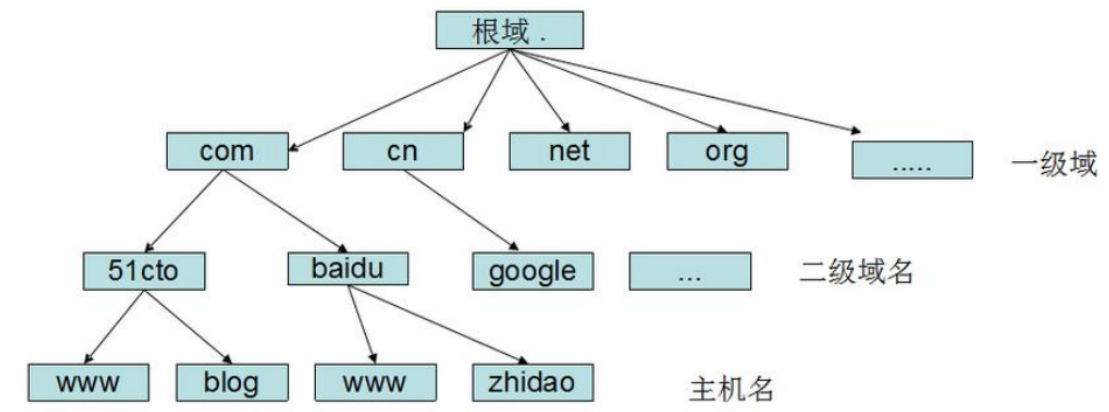
Linux本地DNS: 223.5.5.5 LocalDNS LDNS

本地dns服务器ip地址一般是由本地网络服务商如移动、电信提供，一般是通过DHCP自动分配，当然你也可以自己手动配置。目前用的比较多的是谷歌提供的公用dns 8.8.8.8和国内的公用dns 114.114.114.114及阿里的223.5.5.5。

你之前可能有遇到过电脑可以正常上QQ但是就是不能打开网页的怪现象，这种情况大多数可能就是dns域名解析出问题了，你可以尝试手动把dns设置为公用dns。

找到本地dns后，它也会先去查询一遍它自己的缓存，如果有记录就返回，如果没有记录，它将开始要去我们前面提到的根域名服务器查询了。注意由于根域名服务器ip地址一般都是固定的，所以本地dns服务器一般都内置了根域名服务器ip地址。https://www.uedbox.com/post/50977/

目前全球一共有13个根域名服务器（这里并不是指13台服务器，是指13个ip地址，按字母a-m编号），为了能更高效完成全球所有域名的解析请求，根域名服务器本身并不会直接去解析域名，而是会把不同的解析请求分配给下面的其他服务器去完成，下面是dns域名系统的树状结构图。



注意dns域名服务器一般分三种，分别是根域名服务器(.)、顶级域名服务器(.com)、权威域名服务(.baidu.com) 。

当根域名接收到本地dns的解析请求后，发现是后缀是.com，于是就把负责.com的顶级域名服务器ip地址返给本地dns。

本地dns拿着返回的ip地址再去找到对应的顶级域名服务器，顶级域名又把负责该域名的权威服务器ip返回去。

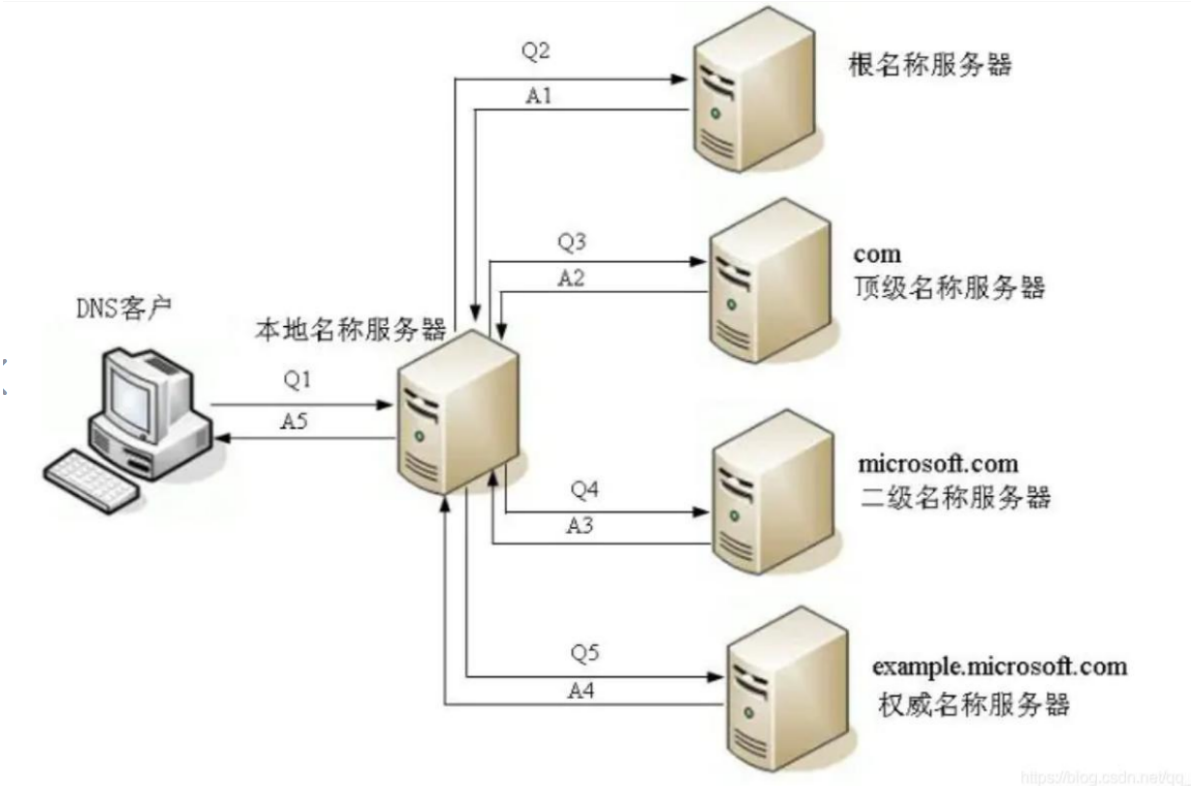
本地dns又拿着ip去找对应的权威服务器，权威服务器最终把对应的主机ip的解析记录（俗称A记录）返回给本地dns。

本地dns会将解析后的ip地址信息进行缓存，缓存好将A记录信息返回给客户端。

客户端收到本地dns响应的A记录信息，会将A记录缓存到本地，然后使用解析后的ip地址访问www.baidu.com。

至此就完成了域名解析的全过程。

下面用一张图来展示上面迭代查询的过程。




```
解析期间涉及到两个特殊查询：
客户端---本地dns服务器：递归查询
本地dns服务器---根域名服务器  顶级域名服务器  权威域名服务器：迭代查询
所谓递归查询过程就是“查询的递交者”更替，而迭代查询过程则是“查询的递交者”不变。

A记录：
从域名到IP的解析过程，被称为A记录；www.baidu.com---1.1.1.1

获取A记录命令方法：yum -y install bind-utils

1） dig www.baidu.com
    dig @223.5.5.5 www.baidu.com +trace   ---显示完整DNS解析过程
2） nslookup www.baidu.com
3） host www.baidu.com
4） ping www.baidu.com
```

```
面试题:
1.浏览器输入www.baidu.com 查询浏览器缓存 有返回IP 没有则查询本地的HOSTS
2.如果HOSTS有返回IP 如果没有继续查询本地的DNS
3.本地DNS一般是我们自己配置的比如223.5.5.5 8.8.8 114.114.114.114，查询本地DNS是否有对应的IP 如果有返回给浏览器 如果没有则查询.根服务器
----------1-3过程称为递归查询---------------
4.根服务器不存储域名解析，会给LDNS返回顶级域.com的服务器IP地址
5.LDNS重新请求.com域名服务器 .com不存在域名解析 .com会返回权威域名服务器的IP地址给LDNS
6.LDNS重新请求baidu.com权威域名服务器,权威域名服务器就是我们自己配置的A记录解析，将A记录对应的IP地址返回给LDNS
7.LDNS拿到后自己缓存一份 返回给浏览器一份
8.浏览器和拿到的百度服务器IP地址建立连接
---------迭代查询 有去有回

1.浏览器--->本地HOSTS-->LDNS--->
2.LDNS--->根 根返回 顶级域.com
3.LDNS--->顶级域 顶级域返回权威域
4.LDNS--->权威域服务器 权威域名返回 A记录解析对应的服务器IP地址
5.浏览器-->百度IP建立连接

```

4. 主机到主机层协议介绍

```
TCP: 传输控制协议,是一种面向连接的、可靠的、基于字节流的传输层通信协议。

    特点：面向连接，可靠，传输效率低

```

```
4
5       应用场景：web浏览器，电子邮件，文件传输程序
6
7       外卖员-->快递必须送到我们的手中
8
9   UDP: 用户数据报协议，属于无连接的传输协议
10
11      特点：无连接、不可靠、快速传输
12
13      应用场景：域名系统（DNS），视频流，IP语音（VOIP）
14      外卖员-->快递放在门口
15
16
17  TCP UDP协议端口号范围 1 - 65535 （可以的），真正端口号总数为2的16次方=65536
18
19  面向连接：是指通信双方在通信时，要事先建立一条通信线路，其有三个过程：建立连接、使用连接和释放连接。
20
21  面向无连接：是指通信双方不需要事先建立一条通信线路，而是把每个带有目的地址的包（报文分组）送到线路上，由系统自主选定路线进行传输。
```

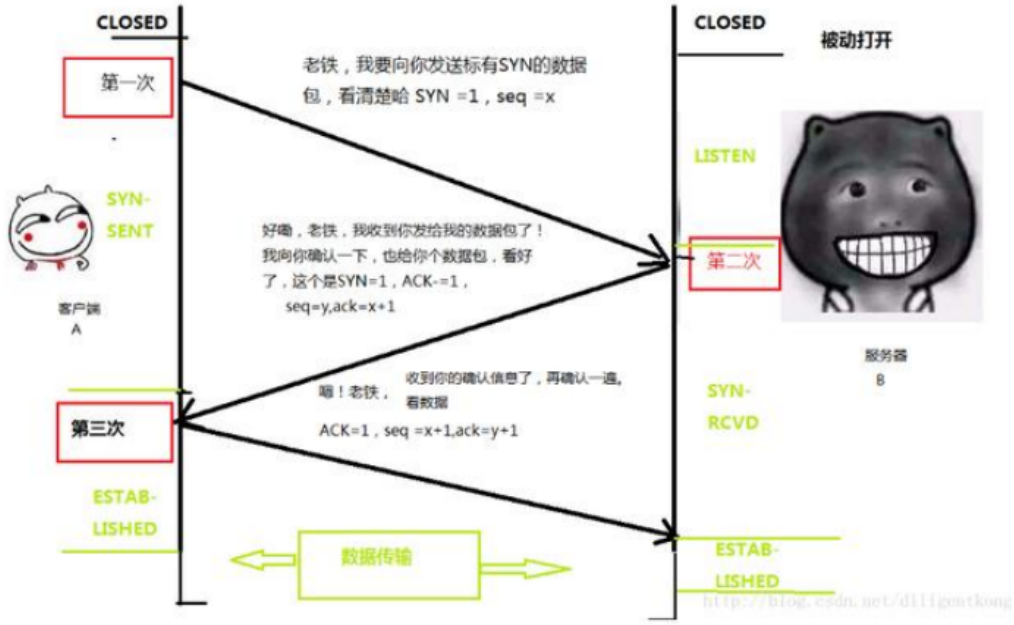
5. 控制字段介绍

```
ACK：表示确认控制字段，确认数据是否接收到
SYN：表示请求建立连接字段，和主机建立连接时使用
FIN：表示请求断开连接字段，和主机断开连接时使用
PSH： 表示有DATA数据传输，PSH为1表示的是有真正的TCP数据包内容被传递
RST： 表示连接重置。一般是在FIN之后才会出现为1的情况，表示的是连接重置。
seq序列号：将大的数据进行拆分后标记序列信息，便于接收方将拆分后的数据信息进行组装，在原有序列号基础上+1进行回复，告知发送方下次再给我发送的数据是什么

```

> 注意：传输数据包，没有真正数据信息（三次握手过程），传输数据过程，在三次握手之后。

6. TCP三次握手

TCP三次握手过程如下图所示：



服务端:
CLOSED--->启动服务LISTEN监听
客户端:
CLOSED--->发送第一个请求后 SYN_SENT

服务端:
LISTEN--->第一次响应客户端后 SYN_RCVD 同步已发送状态

客户端:
SYN_SENT-->发送第三次数据包后 ESTABLISHED建立连接状态

服务端:
SYN_RCVD-->收到客户端第三次数据后ESTABLISHED 建立连接

TCP是面向连接的，无论哪一方向另一方发送数据之前，都必须先在双方之间建立一条连接。

刚开始，客户端和服务器都处于CLOSED状态.

此时，客户端向服务器主动发出连接请求，服务器被动接受连接请求.

1. TCP服务器进程先创建传输控制块TCB，时刻准备接受客户端进程的连接请求，此时服务器就进入了LISTEN（监听）状态.

2. TCP客户端进程也是先创建传输控制块TCB，然后向服务器发出连接请求报文，此时报文首部中的同步标志位SYN=1，同时选择一个初始序列号seq = x，此时TCP客户端进程进入了SYN-SENT（同步已发送状态）状态。TCP规定，SYN报文段（SYN=1的报文段）不能携带数据，但需要消耗掉一个序号。

3. TCP服务器收到请求报文后，如果同意连接，则发出确认报文。确认报文中的ACK=1，SYN=1，确认序号是x + 1，同时也要为自己初始化一个序列号seq = y，此时TCP服务器进程进入了SYN-RCVD（同步收到）状态。这个报文也不能携带数据，但是同样要消耗一个序号。

4. TCP客户端进程收到确认后还要向服务器给出确认。确认报文的ACK=1，确认序号是y + 1，自己的序列号是x + 1.

5. 此时TCP连接建立，客户端进入ESTABLISHED（已建立连接）状态。当服务器收到客户端的确认后也进入ESTABLISHED状态，此后双方就可以开始通信了。

举个栗子：

TCP三次握手好比在一个夜高风黑的夜晚，你一个人在小区里散步，不远处看见小区里的一位漂亮妹子迎面而来，但是因为路灯有点暗等原因不能100%确认，所以要通过招手的方式来确定对方是否认识自己。

你首先向妹子招手(syn)，妹子看到你向自己招手后，向你点了点头挤出了一个微笑(ack)。同时妹子也向你招了招手(syn)，你看到妹子向自己招手后知道对方是在寻求自己的确认，于是也点了点头挤出了微笑(ack)。

于是两人加快步伐，走到了一起。

我们来回顾一下，这个过程中总共有四个动作，

你招手
妹子点头微笑
妹子招手
你点头微笑

其中妹子连续进行了两个动作，先是点头微笑(回复对方)，然后再次招手(寻求确认)，实际上我们可以将这两个动作合成一个动作，招手的同时点头和微笑(syn+ack)。于是这四个动作就简化成了三个动作。

你招手
妹子点头微笑并招手
你点头微笑

这就是三次握手的本质，中间的一次动作是两个动作的合并。通过这个案例，不知你对TCP三次握手，有没有进一步的理解。

三次握手面试题:
1.伟苹想和仓姐姐搞对象,我想和你搞对象SYN=1，并且送给仓姐姐第一个礼物seq=x
2.仓姐姐说收到了你的渴望ACK=1。我也想和你搞对象SYN=1，并且也给伟苹送了第一个礼物seq=y。并且和你说你下次应该给我第二个礼物了Ack=x+1
3.伟苹确认收到了苍姐姐的信息ACK=1。并且给苍姐姐第二个礼物seq=x+1,并且说苍姐姐你下次应该给我第二个礼物了Ack=y+1。
4.伟苹和仓姐姐连接成功。

```
62
63 1.A向B请求建立连接 SYN=1 seq=x
64 2.B向A回复 ACK=1 SYN=1 seq=y,Ack=x+1
65 3.A向B回复 ACK=1,seq=x+1,Ack=y+1
66
67
68 重点:
69 OSI七层模型
70 TCP/IP四层模型
71 DNS解析
72 TCP三次握手
73 TCP和UDP区别
74
```

为什么要三次握手

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。

举个栗子：

> “已失效的连接请求报文段”的产生在这样一种情况下：客户端发出的第一个连接请求报文段并没有丢失，而是在某个网络结点长时间的滞留了，以致延误到连接释放以后的某个时间才到达服务端。本来这是一个早已失效的报文段。但服务端收到此失效的连接请求报文段后，就误认为是客户端再次发出的一个新的连接请求。于是就向客户端发出确认报文段，同意建立连接。假设不采用“三次握手”，那么只要服务端发出确认，新的连接就建立了。由于现在客户端并没有发出建立连接的请求，因此不会理睬服务端的确认，也不会向服务端发送数据。但服务端却以为新的运输连接已经建立，并一直等待客户端发来数据。这样，服务端的很多资源就白白浪费掉了。采用“三次握手”的办法可以防止上述现象发生。例如刚才那种情况，客户端不会向服务端的确认发出确认。服务端由于收不到确认，就知道客户端并没有要求建立连接。”

这就很明白了，防止了服务器端的一直等待而浪费资源。

7. TCP四次挥手

数据传输完毕后，双方都可以释放连接.

此时客户端和服务器都是处于ESTABLISHED状态，然后客户端主动断开连接，服务器被动断开连接.

1. 客户端进程发出连接释放报文，并且停止发送数据。

释放数据报文首部，FIN=1，其序列号为seq = u（等于前面已经传送过来的数据的最后一个字节的序号加1），此时客户端进入FIN-WAIT-1（终止等待1）状态。 TCP规定，FIN报文段即使不携带数据，也要消耗一个序号。

2. 服务器收到连接释放报文，发出确认报文，ACK=1，确认序号为 u + 1，并且带上自己的序列号seq = v，此时服务端就进入了CLOSE-WAIT（关闭等待）状态。
TCP服务器通知高层的应用进程，客户端向服务器请求了断开连接，这时候处于半关闭状态，即客户端已经没有数据要发送了，但是服务器若发送数据，客户端依然要接受。这个状态还要持续一段时间，也就是整个CLOSE-WAIT状态持续的时间。

3. 客户端收到服务器的确认请求后，此时客户端就进入FIN-WAIT-2（终止等待2）状态，等待服务器发送连接释放报文（在这之前还需要接受服务器发送的最终数据）

4. 服务器将最后的数据发送完毕后，就向客户端发送连接释放报文，FIN=1，确认序号为v + 1，由于在半关闭状态，服务器很可能又发送了一些数据，假定此时的序列号为seq = w，此时服务器就进入了LAST-ACK（最后确认）状态，等待客户端的确认。

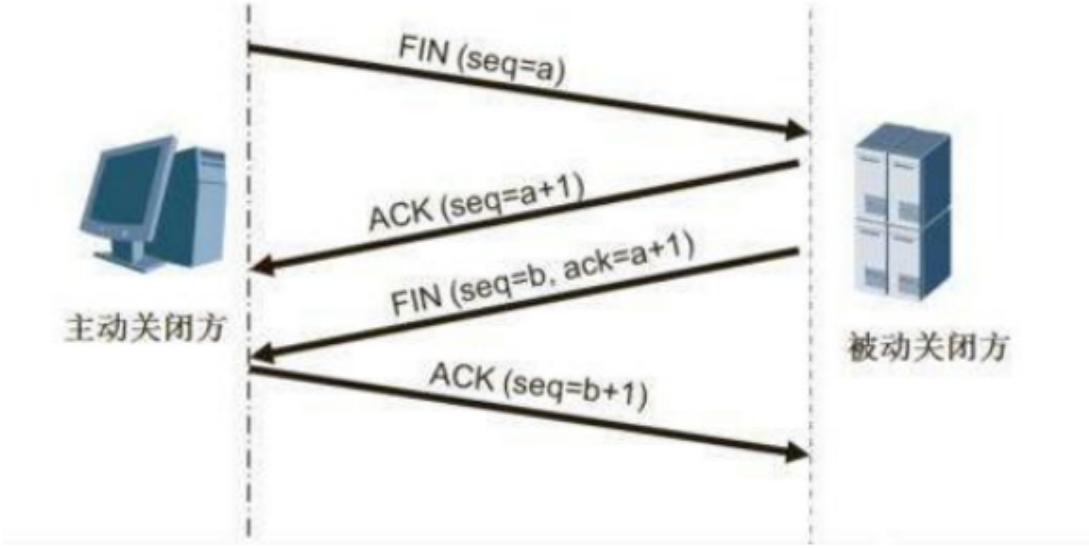
5. 客户端收到服务器的连接释放报文后，必须发出确认，ACK=1，确认序号为w + 1，而自己的序列号是u + 1，此时客户端就进入了TIME-WAIT（时间等待）状态。注意此时TCP连接还没有释放，必须经过2*MSL（最长报文段寿命）的时间后，当客户端撤销相应的TCB后，才进入CLOSED状态。

6. 服务器只要收到了客户端发出的确认，立即进入CLOSED状态。同样，撤销TCB后，就结束了这次的TCP连接。可以看到，服务器结束TCP连接的时间要比客户端早一些。

小知识点：在网络传输层，tcp模块中有一个tcb（传输控制模块，transmitcontrolblock），它用于记录tcp协议运行过程中的变量。对于有多个连接的tcp，每个连接都有一个tcb。tcb结构的定义包括这个连接使用的源端口、目的端口、目的ip、序号、应答序号、对方窗口大小、己方窗口大小、tcp状态、top输入/输出队列、应用层输出队列、tcp的重传有关变量。

四次挥手:
1.伟苹向仓姐姐发送分手的请求。我要和你分手 FIN
2.苍姐姐收到回复 好的收到了 ACK
3.苍姐姐发送给伟苹 我们分手吧 FIN
4.伟苹回复 好的分吧 ACK

SYN 请求建立连接
ACK 确认收到
seq 序列号
Ack 确认下次的序列号
FIN 分开 断开连接



为什么要四次分手?

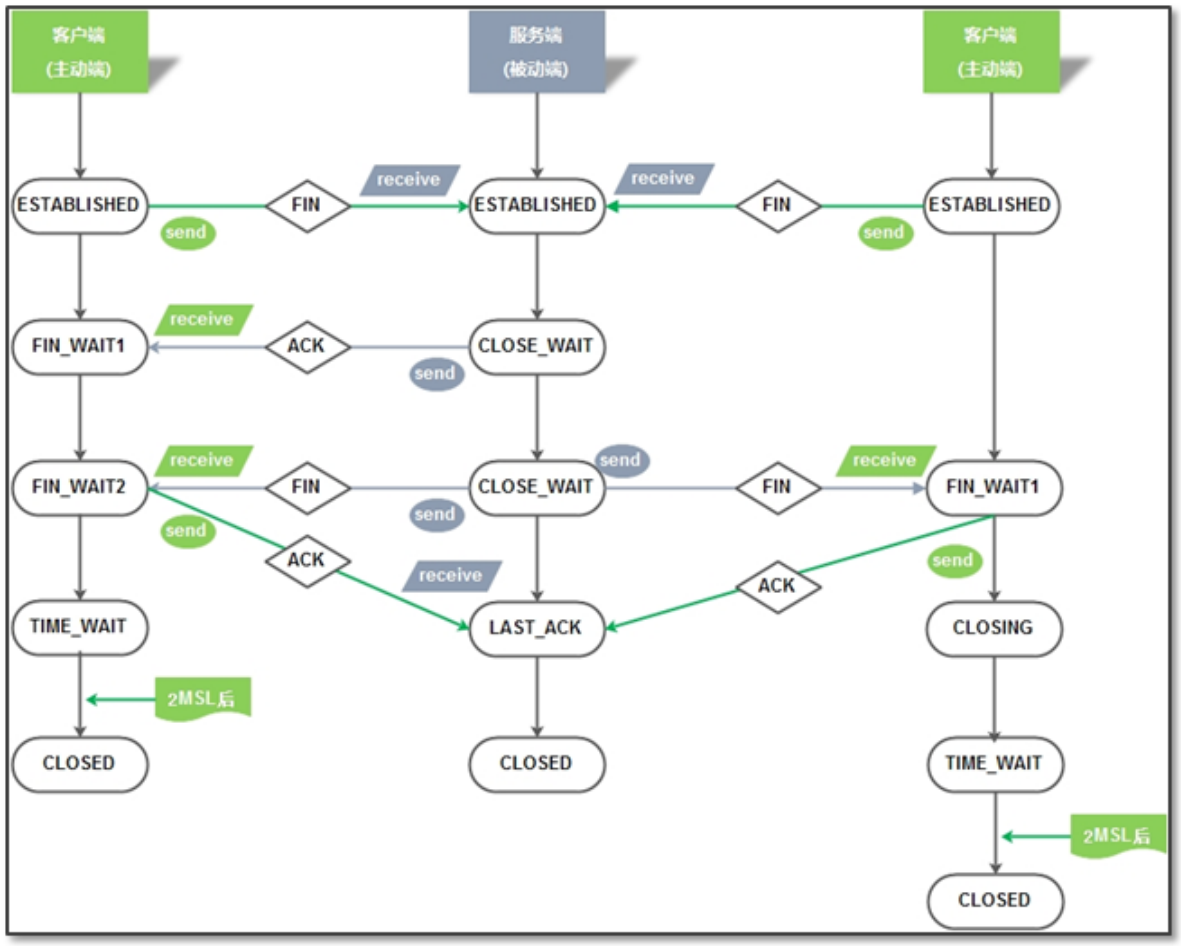
```
1
```

那四次分手又是为何呢？TCP协议是一种面向连接的、可靠的、基于字节流的传输层通信协议。TCP是全双工模式，这就意味着，当主机1发出FIN报文段时，只是表示主机1已经没有数据要发送了，主机1告诉主机2，它的数据已经全部发送完毕了；但是，这个时候主机1还是可以接受来自主机2的数据；当主机2返回ACK报文段时，表示它已经知道主机1没有数据发送了，但是主机2还是可以发送数据到主机1的；当主机2也发送了FIN报文段时，这个时候就表示主机2也没有数据要发送了，就会告诉主机1，我也没有数据要发送了，主机1收到主机2的FIN报文段时，回复ACK，表示知道主机2也没有数据传输了，之后彼此就会愉快的中断这次TCP连接。

为什么TIME_WAIT 状态还需要等2*MSL秒之后才能返回到CLOSED 状态呢?

```
1    因为虽然双方都同意关闭连接了，而且握手的4个报文也都发送完毕，按理可以直接回到CLOSED状态（就好比从SYN_SENT状态到ESTABLISH状态那样），但是我们必须假想网络是不可靠的，你无法保证你最后发送的ACK报文一定会被对方收到，就是说对方处于LAST_ACK状态下的SOCKET可能会因为超时未收到ACK报文，而重发FIN报文，所以这个TIME_WAIT状态的作用就是用来重发可能丢失的ACK报文。
2
```

8. TCP协议的十一种状态集转换



TCP11种状态集表示含义

```
1    客户端发送FIN给服务端-->ESTABLISHED-->FIN_WAIT1第一次等待
2    服务端收到回复ACK： ESTABLISHED--->CLOSED_WAIT关闭等待
3    收到后客户端：FIN_WAIT1-->FIN_WAIT2第二次等待
```

服务端继续回复客户端FIN: CLISED_WAIT--->LAST_ACK
客户端回复确认ACK：FIN_WAIT2-->TIME_WAIT时间等待-->CLOSED状态
最后服务端收到确认ACK： LAST_ACK---> CLOSED

各个状态的意义如下：

CLOSED：初始状态，表示TCP连接是“关闭着的”或“未打开的”。

LISTEN ：表示服务器端的某个SOCKET处于监听状态，可以接受客户端的连接。

SYN_RCVD ：表示服务器接收到了来自客户端请求连接的SYN报文。在正常情况下，这个状态是服务器端的SOCKET在建立TCP连接时的三次握手会话过程中的一个中间状态，很短暂，基本上用netstat很难看到这种状态，除非故意写一个监测程序，将三次TCP握手过程中最后一个ACK报文不予发送。当TCP连接处于此状态时，再收到客户端的ACK报文，它就会进入到ESTABLISHED状态。

SYN_SENT ：这个状态与SYN_RCVD状态相呼应，当客户端SOCKET执行connect()进行连接时，它首先发送SYN报文，然后随即进入到SYN_SENT状态，并等待服务端的发送三次握手中的第2个报文。SYN_SENT状态表示客户端已发送SYN报文。

ESTABLISHED ：表示TCP连接已经成功建立。

FIN_WAIT_1 ：这个状态得好好解释一下，其实FIN_WAIT_1和FIN_WAIT_2两种状态的真正含义都是表示等待对方的FIN报文。而这两种状态的区别是：FIN_WAIT_1状态实际上是当SOCKET在ESTABLISHED状态时，它想主动关闭连接，向对方发送了FIN报文，此时该SOCKET进入到FIN_WAIT_1状态。而当对方回应ACK报文后，则进入到FIN_WAIT_2状态。当然在实际的正常情况下，无论对方处于任何种情况下，都应该马上回应ACK报文，所以FIN_WAIT_1状态一般是比较难见到的，而FIN_WAIT_2状态有时仍可以用netstat看到。

FIN_WAIT_2 ：上面已经解释了这种状态的由来，实际上FIN_WAIT_2状态下的SOCKET表示半连接，即有一方调用close()主动要求关闭连接。注意：FIN_WAIT_2是没有超时的（不像TIME_WAIT状态），这种状态下如果对方不关闭（不配合完成4次挥手过程），那这个FIN_WAIT_2状态将一直保持到系统重启，越来越多的FIN_WAIT_2状态会导致内核崩溃。

TIME_WAIT ：表示收到了对方的FIN报文，并发送出了ACK报文。TIME_WAIT状态下的TCP连接会等待2*MSL（Max Segment Lifetime，最大分段生存期，指一个TCP报文在Internet上的最长生存时间。每个具体的TCP协议实现都必须选择一个确定的MSL值，RFC 1122建议是2分钟，但BSD传统实现采用了30秒，Linux可以cat /proc/sys/net/ipv4/tcp_fin_timeout看到本机的这个值），然后即可回到CLOSED可用状态了。如果FIN_WAIT_1状态下，收到了对方同时带FIN标志和ACK标志的报文时，可以直接进入到TIME_WAIT状态，而无须经过FIN_WAIT_2状态。（这种情况应该就是四次挥手变成三次挥手的那种情况）

CLOSING ：这种状态在实际情况中应该很少见，属于一种比较罕见的例外状态。正常情况下，当一方发送FIN报文后，按理来说是应该先收到（或同时收到）对方的ACK报文，再收到对方的FIN报文。但是CLOSING状态表示一方发送FIN报文后，并没有收到对方的ACK报文，反而却也收到了对方的FIN报文。什么情况下会出现此种情况呢？那就是当双方几乎在同时close()一个SOCKET的话，就出现了双方同时发送FIN报文的情况，这是就会出现CLOSING状态，表示双方都正在关闭SOCKET连接。

CLOSE_WAIT ：表示正在等待关闭。怎么理解呢？当对方close()一个SOCKET后发送FIN报文给自己，你的系统毫无疑问地将会回应一个ACK报文给对方，此时TCP连接则进入到CLOSE_WAIT状态。接下来呢，你需要检查自己是否还有数据要发送给对方，如果没有的话，那你也就可以close()这个SOCKET并发送FIN报文给对方，即关闭自己到对方这个方向的连接。有数据的话则看程序的策略，继续发送或丢弃。简单地说，当你处于CLOSE_WAIT状态下，需要完成的事情是等待你去关闭连接。

LAST_ACK ：当被动关闭的一方在发送FIN报文后，等待对方的ACK报文的时候，就处于LAST_ACK状态。当收到对方的ACK报文后，也就可以进入到CLOSED可用状态了。

重点小结:

1.OSI七层模型

应用层 表示层 会话层 传输层 网络层 数据链路层 物理层

2.DNS解析流程

3.TCP三次握手

4.TCP四次挥手

9. 因特网层协议介绍

```
ICMP      Internet控制报文协议。它是TCP/IP协议簇的一个子协议，用于在IP主机、路由器之间传递控制消息。控制消息是指网络通不通、主机是否可达、路由是否可用等网络本身的消息。这些控制消息虽然并不传输用户数据，但是对于用户数据的传递起着重要的作用。

RARP      反向地址转换协议

ARP       地址解析协议,是根据IP地址获取物理地址的一个TCP/IP协议。作用：有效的避免广播风暴的产生

          动态ARP：自动完善ARP表信息,会定时更新ARP条目,自动更新ARP表时会消耗服务器性能,适用于主机更换频繁网络。
          静态ARP：手工配置ARP表信息,不会实时更新ARP条目，节省服务器性能，适用于主机更换不频繁网络。
```

10. 网络接入层介绍

```
Ethernet      以太网（Ethernet）是一种计算机局域网技术。

FastEth       快速以太网（Fast Ethernet）是一类新型的局域网，其名称中的“快速”是指数据速率可以达到100Mbps，是标准以太网的数据速率的十倍。

Token Ring    令牌环网（Token Ring）是一种LAN协议，其中所有的工作站都连接到一个环上，每个工作站只能同直接相邻的工作站传输数据。

FDDI          光纤分布式数据接口，在光缆网络上发送数字和音频信号的一组协议。
```

6. IP地址

1. IP地址基本概念

```
    IP地址（Internet Protocol Address）是指互联网协议地址，又译为网际协议地址。

    IP地址在网络层将不同的物理网络地址统一到了全球唯一的IP地址上（屏蔽物理网络差异），是唯一标识互联网上计算机的逻辑地址（相当于手机号码，可以通过唯一的手机号码找到手机），所以IP地址也被称为互联网地址（可见其重要性）。
```

2. IP地址格式

```

```

我们目前常用的IPv4中规定，IP地址长度为32位二进制，在表示时，一般将32位地址拆分为4个8位二进制，再转为4个十进制数表示，每个数字之间用点隔开，如127.0.0.1（localhost），这种描述方式被称为“点-数表示法”点分十进制。

11111111 11111111 11111111 11111111
  255.255.255.255

0 0 0 0 0 0 0 1 二进制
                  1 十进制
0 0 0 0 0 0 1 0

                  2 十进制
0 0 0 0 0 0 1 1
                  3 十进制

1 1 1 1 1 1 1 1
128 64 32 16 8 4 2 1

二进制转十进制算法：将每个为1的位置相加得到十进制
0 0 0 0 0 1 0 1 转换成十进制 等于 5
0 1 0 0 0 0 0 0 转换成十进制 等于 64
1 1 0 0 0 0 0 0 转换成十进制 等于 192
1 1 1 1 1 1 1 1 转出成十进制 等于 255

十进制转二进制：对每位为1的相减 做减法运算

```
1     1   1    1     1     1    1    1
128  64  32   16     8     4    2    1
68十进制转换成二进制
68-64=4
4-4=0
0    1   0 0  0  1   0  0

172十进制转换成二进制
172-128=44
44-32=12
12-8=4
4-4=0
1   0  1  0  1  1 0 0


192.168.13.253
192  =  1 1 0 0 0 0 0 0
168  =  1 0 1  0 1  0 0 0
13   =  0 0 0 0 1 1 0 1
253  =  1 1 1 1 1 1 0 1



IP地址层次：分为网络号和主机号两个层次。网络号表示主机所属网络，主机号表示主机本身。网络号与主机号的位数与IP地址分类有关。
```

192.	168.	10.	1
11000000	10101000	00001010	00000001

IP地址十进制与二进制对应关系

十进制数	二进制数	十进制数	二进制数
1	1	11	1011
2	10	12	1100
3	11	13	1101
4	100	14	1110
5	101	15	1111
6	110	16	10000
7	111	17	10001
8	1000	18	10010
9	1001	19	10011
10	1010	20	10100

十进制数	二进制数	二进制	十进制数	二进制数	二进制
1	1	00001	11	1011	01011
2	10	00010	12	1100	01100
3	11	00011	13	1101	01101
4	100	00100	14	1110	01110
5	101	00101	15	1111	01111
6	110	00110	16	10000	10000
7	111	00111	17	10001	10001
8	1000	01000	18	10010	10010
9	1001	01001	19	10011	10011
10	1010	01010	20	10100	10100

二级制与十进制对应数值表

根据第两张表的信息进行数据的逻辑总结可以得知如下结论：

二进制	00001	00010	00100	01000	10000
逻辑运算	2的0次方	2的1次方	2的2次方	2的3次方	2的4次方
十进制	1	2	4	8	16

二级制转换十进制公式表

因此可以得知对于点分十进制而言，对应的每个数值即为下图所示：

二进制	10000000	01000000	00100000	00010000	00001000	00000100	00000010	00000001
逻辑运算	2的7次方	2的6次方	2的5次方	2的4次方	2的3次方	2的2次方	2的1次方	2的0次方
十进制	128	64	32	16	8	4	2	1

二级制转换十进制公式表

根据上面说到的将 32 位数字分为 4 端，即每段 8 位数字；通过上图也可以得知主机地址的初步理解算法

192.	168.	10.	1
11000000	10101000	00001010	00000001

IP地址十进制与二进制对应关系

3. IP地址分配

> IP地址分配的基本原则是：要为同一网络（子网、网段）内不同主机分配相同的网络号，不同的主机号。

4. IP地址类型

#公有地址

公有地址（Public address）由Inter NIC（Internet Network Information Center因特网信息中心）负责。这些IP地址分配给注册并向Inter NIC提出申请的组织机构。通过它直接访问因特网。全球唯一，不能出现重复.

#私有地址

私有地址（Private address）属于非注册地址，专门为组织机构内部使用。缓解了地址枯竭 是可以重复使用的（不同局域网内）

#以下列出留用的内部私有地址

A类 10.0.0.0--10.255.255.255

10.0.0.1-10.0.0.254

10.0.1.1=10.0.1.254

10.0.2.1=10.0.2.254

10.1.0.1=10.1.0.1=10.1.0.254

B类 172.16.0.0--172.31.255.255

172.16.0.1-172.16.0.254

172.16.1.1-172.16.1.254

11111111=255

C类 192.168.0.0--192.168.255.255

192.168.0.1-192.168.0.254

192.168.1.1-192.168.1.254

192.168.2.1-192.168.2.254

需要实现配置私网地址的服务器可以访问外网（互联网）？？？

```
26 NAT --- 网络地址转换技术（化妆），将私网地址转换为公网地址
```

5. IP地址常见分类

```
1  #A类IP地址
2
3      一个A类IP地址是指， 在IP地址的四段号码中，第一段号码为网络号码，剩下的三段号码为本地计算机的号码。如果用二进制表示IP地址的话，A类IP地址就由1字节的网络地址和3字节主机地址组成，网络地址的最高位必须是“0”。A类IP地址中网络的标识长度为8位，主机标识的长度为24位，A类网络地址数量较少，有126个网络，每个网络可以容纳主机数达1600多万台。
4
5  A类IP地址 地址范围1.0.0.1到127.255.255.254 （二进制表示为：00000001 00000000 00000000 00000001 - 01111111 11111111 11111111 11111110）。最后一个是广播地址。
6
7  A类IP地址的子网掩码为255.0.0.0，每个网络支持的最大主机数为256的3次方-2=16777212台。
8
9  #B类IP地址
10
11     一个B类IP地址是指，在IP地址的四段号码中，前两段号码为网络号码。如果用二进制表示IP地址的话，B类IP地址就由2字节的网络地址和2字节主机地址组成，网络地址的最高位必须是“10”。B类IP地址中网络的标识长度为16位，主机标识的长度为16位，B类网络地址适用于中等规模的网络，有16384个网络，每个网络所能容纳的计算机数为6万多台。
12
```

B类IP地址地址范围128.0.0.1-191.255.255.254 （二进制表示为：10000000 00000000 00000000 00000001----10111111 11111111 11111111 11111110）。 最后一个是广播地址。

B类IP地址的子网掩码为255.255.0.0，每个网络支持的最大主机数为256的2次方-2=65534台。

#C类IP地址

一个C类IP地址是指，在IP地址的四段号码中，前三段号码为网络号码，剩下的一段号码为本地计算机的号码。如果用二进制表示IP地址的话，C类IP地址就由3字节的网络地址和1字节主机地址组成，网络地址的最高位必须是“110”。C类IP地址中网络的标识长度为24位，主机标识的长度为8位，C类网络地址数量较多，有209万余个网络。适用于小规模的局域网络，每个网络最多只能包含254台计算机。

C类IP地址范围192.0.0.1-223.255.255.254 （二进制表示为: 11000000 00000000 00000000 00000001 - 11011111 11111111 11111111 11111110）。

C类IP地址的子网掩码为255.255.255.0，每个网络支持的最大主机数为256-2=254台

#D类地址用于多点广播（Multicast）。

D类IP地址在历史上被叫做多播地址(multicast address)，即组播地址。在以太网中，多播地址命名了一组应该在这个网络中应用接收到一个分组的站点。多播地址的最高位必须是“1110”，范围从224.0.0.0到239.255.255.255。

```
#E类IP地址

以“11110”开始，为将来使用保留。
```

6. 特殊的IP地址

```
1.每一个字节都为0的地址（“0.0.0.0”）对应于当前主机；

2.IP地址中的每一个字节都为1的IP地址（“255. 255. 255. 255”）是当前子网的广播地址；

3.IP地址中凡是以“11110”开头的E类IP地址都保留用于将来和实验使用。

4.IP地址中不能以十进制“127”作为开头，该类地址中数字127. 0. 0. 1到127. 255. 255. 255用于回路测试，
如：127.0.0.1可以代表本机IP地址，用
“http://127.0.0.1”就可以测试本机中配置的Web服务器。

5.169.254.0.0~169.254.255.255,是开启了dhcp服务的设备但又无法获取到dhcp的会随机使用这个网段的ip

总结: 3类
A类   前1位网络位置  后三位主机位
      127.0.0.1  127.255.255.254
B类: 前2位网络位置  后2位主机位
      130.0.0.1  130.0.255.254
C类: 前3位网络位置  后1位为主机位
    192.168.13.1 192.168.13.254

```

7. 子网掩码

子网掩码又叫网络掩码、地址掩码

上面我们说到IP地址分为网络号与主机号，但是路由如何区分网络号与主机号呢？就需要通过子网掩码。子网掩码必须与IP地址结合使用，A、B、C类的子网掩码分别为255.0.0.0，255.255.0.0与255.255.255.0（网络号字节为255，主机号字节为0）。

也就是说给你一个IP地址，那么怎么知道它的网络号和主机号各是多少位呢？

如果不指定，就不知道哪些位是网络号、哪些是主机号，这就需要通过子网掩码来实现

子网掩码的重要作用：就是将某个IP地址划分成网络地址和主机地址两部分。

子网掩码的位数就是网络的位数。A类网络的网络位数是8位，子网掩码就是255.0.0.0，B类网络的网络位数是16位，子网掩码是255.255.0.0，C类是24位，255.255.255.0。

```
/8   255.0.0.0
/16  255.255.0.0
/24  255.255.255.0
```

1. 例1：不同子网下的主机能否直接通信（是否在同一网络/段下）

```
假设两个IP地址分别是172.20.0.18和172.20.1.16，子网掩码都是255.255.255.0。

我们可以知道两者的网络标识分别是172.20.0和172.20.1，无法直接通信，也就无法PING通。要想能相互通信，需要将子网掩码改成255.255.0.0
```

2. 如何理解172.20.1.0/18

11111111 11111111 11000000 00000000

255.255.192.0

为什么要子网划分

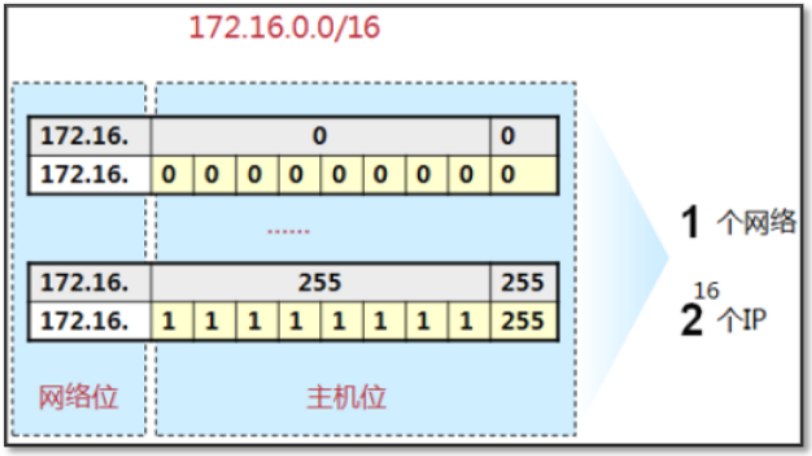
```
一个大的地址范围区域，你不进行划分的时候，会造成地址浪费
一个大的地址范围区域，可能会产生大量广播风暴，影响主机性能
一个大的地址访问区域，可能会造成网关路由器负载过高

将一个大的网段切割成一个一个小的局域网段，就称为子网划分

一个网段中可以有多少个地址=2的n次方-2  n表示的就是这个网段中有多少个主机位
        -2  表示网络地址不能用    表示广播地址不能用
        一个局域网中的地址在使用时要预留一个作为网关地址

```

掩码地址的表示方式

	网络位			主机位	
IP地址	192	168	1	10	
对应掩码	255	255	255	0	
十进制	11111111	11111111	11111111	00000000	/24

掩码地址表示

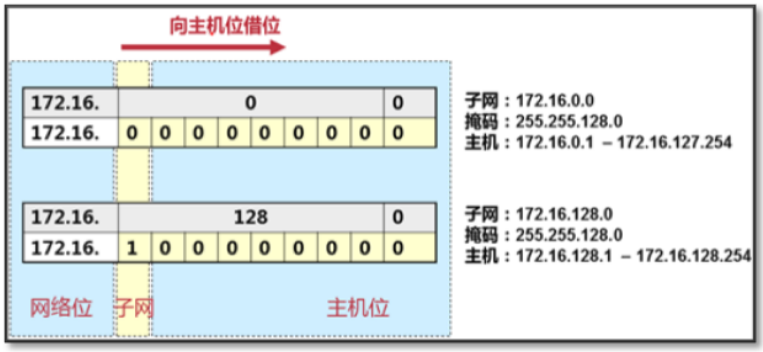
根据掩码如何进行子网划分



B 类地址主机位地址范围表示

网络位向右移动，占用主机位，即向主机位借位，生成新的网络位

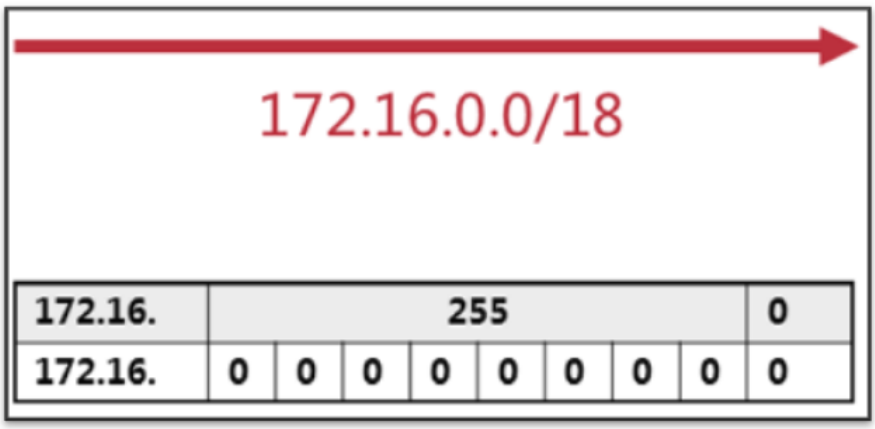
其中/16就表示了子网掩码所指定的网络位个数，A类一般是/8 B类一般是/16 C类/24



B 类地址子网化主机位地址范围表示

实质上就是移动网络位和主机位中间分隔的线，向主机位进行移动，减少主机数量，扩大拥有的子网数量；新的子网产生，掩码表示的信息也要进行变化，从而可以更好的表示网络地址的规划情况

用示例题进行掩码的说明：



划分子网实例示意图 01

划分子网实例示意图02

	网络位	主机位			地址划分信息
默认IP类型	172.16	255	0		子网：172.16.0.0 掩码：255.255.0.0 主机：172.16.0.1-172.16.255.254
二进制说明	172.16	00000000	0		
掩码	255.255	0	0	/16	

	网络位	主机位			地址划分信息
默认IP类型	172.16	子网位 255	0		
二进制说明	172.16	00000000	0		子网：172.16.0.0 掩码：255.255.192.0 主机：172.16.0.1-172.16.63.254
掩码	255.255	192	0	/18	
二进制说明	172.16	01000000	0		子网：172.16.192.0 掩码：255.255.192.0 主机：172.16.64.1-172.16.127.254
掩码	255.255	192	0	/18	
二进制说明	172.16	10000000	0		子网：172.16.192.0 掩码：255.255.192.0 主机：172.16.128.1-172.16.191.254
掩码	255.255	192	0	/18	
二进制说明	172.16	11000000	0		子网：172.16.192.0 掩码：255.255.192.0 主机：172.16.192.1-172.16.255.254
掩码	255.255	192	0	/18	

划分子网实例示意图03

运营商或者领导给我们一个IP地址加子网掩码我们必须要知道可配置的IP地址范围

116.63.0.10/29 网关: 116.63.0.9

https://www.36130.com/subnetmask/

网关: 116.63.0.9

116.63.0.10 255.255.255.248 网关 116.63.0.9 配置到哪个服务器上.

8. 网关

```
1 网关（Gateway）又称网间连接器，就是一个网络连接到另一个网络的“关口”。
2
```

网关实质上是一个网络通向其他网络的IP地址。比如有网络A和网络B，网络A的IP地址范围为“192.168.1.1~192. 168.1.254”，子网掩码为255.255.255.0；网络B的IP地址范围为“192.168.2.1~192.168.2.254”，子网掩码为255.255.255.0。在没有路由器的情况下，两个网络之间是不能进行TCP/IP通信的，即使是两个网络连接在同一台交换机(或集线器)上，TCP/IP协议也会根据子网掩码(255.255.255.0)判定两个网络中的主机处在不同的网络里。而要实现这两个网络之间的通信，则必须通过网关。如果网络A中的主机发现数据包的目的主机不在本地网络中，就把数据包转发给它自己的网关，再由网关转发给网络B的网关，网络B的网关再转发给网络B的某个主机(如附图所示)。网络B向网络A转发数据包的过程。

所以说，只有设置好网关的IP地址，TCP/IP协议才能实现不同网络之间的相互通信。那么这个IP地址是哪台机器的IP地址呢？网关的IP地址是具有路由功能的设备的IP地址，具有路由功能的设备有路由器、启用了路由协议的服务器(实质上相当于一台路由器)、代理服务器(也相当于一台路由器)。

ifconfig route netstat 命令属于net-tools网络工具包

```
ip add    linux
ifconfig  linux
ping      linux windows
nslookup  linux
route -n

tracert -d www.baidu.com windows
```

```
[root@oldboy:~]# traceroute -n www.baidu.com

[root@oldboy ~]# route -n   # 查看网关
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         10.0.0.2        0.0.0.0         UG    0      0        0 eth0
10.0.0.0        0.0.0.0         255.255.255.0   U     0      0        0 eth0
169.254.0.0     0.0.0.0         255.255.0.0     U     1002   0        0 eth0

给网卡配置多个IP地址: 临时重启失效  笔记
[root@oldboy ~]# ip add add 10.0.0.201/24 dev eth0
删除临时IP:
[root@oldboy ~]# ip address del 10.0.0.201/24 dev eth0

临时删除网关:
[root@oldboy ~]# ip route del 0/0 via 10.0.0.2
配置临时的网关:
[root@oldboy ~]# ip route add 0/0 via 10.0.0.2

#笔试题 给linux系统配置一个默认网关
vim /etc/sysconfig/network-scripts/ifcfg-ens33
GATEWAY
```

```
[root@oldboy ~]#route add default gw 10.0.0.3
[root@oldboy ~]#route del default gw 10.0.0.3  # 删除默认网关

知识重点:
1.OSI七层模型
应用层
表示层
会话层
传输层
网络层
数据链路层
物理层
2.TCP/TP四层
应用层
主机到主机层
inter网
接入层

3.DNS解析流程

4.TCP三次握手
  为什么要三次握手
5.TCP四次挥手

6.TCP11状态 笔记

7.IP地址/子网掩码   子网掩码决定了IP可用的数量



```

9. 抓包方式

抓包方式：wireshark抓包软件在Windows中使用

Linux抓包命令tcpdump是一个抓包工具，用于抓取互联网上传输的数据包

```
    tcpdump是一个用于截取网络分组，并输出分组内容的工具。凭借强大的功能和灵活的截取策略，使其成为类UNIX系统下用于网络分析和问题排查的首选工具
    tcpdump 支持针对网络层、协议、主机、网络或端口的过滤，并提供and、or、not等逻辑语句来帮助你去掉无用的信息

#常用选项
[root@oldboyedu ~]#yum -y install tcpdump

-i          #监听哪一个网卡
-n          #不把ip解析成主机名
-nn         #不把端口解析成应用层协议
-c          #指定抓包的数量
-S          #不把随机序列和确认序列解析成绝对值
-w          #将流量保存到文件中,文件中的信息是无法直接查看的
-r          #读取文件中的内容
-v          #输出一个稍微详细的信息，例如在ip包中可以包括ttl和服务类型的信息。
-vv         #输出详细的报文信息。
-nnvvi ens33

#实例
企业中遇到无法远程连接拍错流程: IDC机房服务器 云服务器
```

1.ping通
2.通过页面或者机房插显示器连接服务器抓包
tcpdump -nni eth0 port 12345
3.在公司windows电脑使用telnet连接测试
cmd窗口-->telnet 服务器IP地址 12345

结果:
1.如果服务端可以看到来源公网IP地址 说明服务端做的限制
2.如果服务端来源IP地址不是公司公网IP地址 则需要放行抓取到的公网(打电话解决)
3.如果服务端收不到任何信息，说明是公司网络问题

保存到文件中
[root@oldboy:~]# tcpdump -w 1.txt -nnvvi ens33 dst www.baidu.com
查看文件中的内容
[root@oldboy:~]# tcpdump -r 1.txt

1、默认启动

tcpdump -vv   #普通情况下，直接启动tcpdump将监视第一个网络接口上所有流过的数据包。

2、过滤主机

tcpdump -i eth1 host 192.168.1.1   #抓取所有经过eth1，目的或源地址是192.168.1.1的网络数据

tcpdump -i eth1 src host 192.168.1.1   #指定源地址，192.168.1.1

```
tcpdump -i eth1 dst host 192.168.1.1    #指定目的地址，192.168.1.1

3、过滤端口

tcpdump -i eth1 port 80          #抓取所有经过eth1，目的或源端口是80的网络数据

tcpdump -i eth1 src port 80      #指定源端口

tcpdump -i eth1 dst port 80      #指定目的端口

4、协议过滤

tcpdump -i eth1 arp

tcpdump -i eth1 ip

tcpdump -i eth1 tcp

tcpdump -i eth1 udp

tcpdump -i eth1 icmp

#抓tcp某端口的数据包

tcpdump -i eth0 tcp port 21 -nn

5、常用表达式

非 : ! or "not" (去掉双引号)

且 : && or "and"
```

```
78
79 或 : || or "or"
80
81 #抓取所有经过eth1，目的地址是192.168.1.254或
   192.168.1.200端口是80的TCP数
82
83 tcpdump -i eth1 '((tcp) and (port 80) and
   ((dst host 192.168.1.254) or (dst host
   192.168.1.200)))'
84
85 #抓取所有经过eth1，目标MAC地址是
   00:01:02:03:04:05的ICMP数据
86
87 tcpdump -i eth1 '((icmp) and ((ether dst
   host 00:01:02:03:04:05)))'
88
89 #抓取所有经过eth1，目的网络是192.168，但目的主机不
   是192.168.1.200的TCP数据
90
91 tcpdump -i eth1 '((tcp) and ((dst net
   192.168) and (not dst host 192.168.1.200)))'
```

10. Linux常用网络命令

1. 网卡命令规则

```
1 CentOS-6之前基于传统的命名方式如：eth1，eth0....
2
3 Centos-7提供了不同的命名规则，默认是基于固件、拓
  扑、位置信息来分配。这样做的优点是命名是全自动的、可
  预知的，缺点是比eth0、wlan0更难读。比如enp5s0
4
5 biosdevname和net.ifnames两种命名规范
```

```
#net.ifnames的命名规范为:

    设备类型+设备位置+数字

#设备类型:

    en 表示Ethernet

    wl 表示WLAN

    ww 表示无线广域网WWAN

#实际的例子:

    eno1    #板载网卡

    enp0s2  #pci网卡

    ens33   #pci网卡

    wlp3s0  #PCI无线网卡

    wwp0s29f7u2i2   #4G modem

    wlp0s2f1u4u1   #连接在USB Hub上的无线网卡

#biosdevname的命名规范为:

    根据系统BIOS提供的信息对网络接口进行重命名。

    em[1-N] #表示主板（嵌入式）NIC （对应机箱标签）
```

```
    pci      #表示PCI插槽中的卡，端口1至N

实际的例子:

    em1      #板载网卡

    p3p4     #pci网卡

    p3p4_1   #虚拟网卡

CentOS-7

默认内核参数(biosdevname=0（dell服务器默认是1），net.ifnames=1):  网卡名 "enp5s2"

biosdevname=1，net.ifnames=0：网卡名 "em1"

biosdevname=0，net.ifnames=0：网卡名 "eth0"(最传统的方式,eth0 eth1)

#定义网卡命令规则

在安装系统时，选择安装选项，按tab键，在跳出的一行内容后面添加net.ifnames=0 biosdevname=0

#命令行设置网卡名称规则

[root@qls ~]# cd /etc/sysconfig/network-scripts/  #修改网卡配置文件
[root@qls network-scripts]# mv ifcfg-ens33 ifcfg-eth0
```

```
[root@qls network-scripts]# sed -i "s#ens33#eth0#g" ifcfg-eth0
[root@qls ~]# vim /etc/sysconfig/grub  #GRUB添加kernel参数
GRUB_CMDLINE_LINUX="...net.ifnames=0 biosdevname=0 quiet"
[root@qls ~]# grub2-mkconfig -o /boot/grub2/grub.cfg
[root@qls ~]# reboot   #重启系统生效
```

2. 网卡配置文件详解

```
#动态ip
[root@qls ~]# cat /etc/sysconfig/network-scripts/ifcfg-eth0
TYPE="Ethernet"
PROXY_METHOD="none"
BROWSER_ONLY="no"
BOOTPROTO="dhcp"
DEFROUTE="yes"
IPV4_FAILURE_FATAL="no"
IPV6INIT="yes"
IPV6_AUTOCONF="yes"
IPV6_DEFROUTE="yes"
IPV6_FAILURE_FATAL="no"
IPV6_ADDR_GEN_MODE="stable-privacy"
NAME="eth0"
UUID="fb32c09d-5a9f-40b9-852b-0f44ff2202ed"
DEVICE="eth0"
ONBOOT="yes"

#静态ip
```

```
[root@qls ~]# cat /etc/sysconfig/network-scripts/ifcfg-eth0
TYPE="Ethernet"
BOOTPROTO="static"
NAME="eth0"
DEVICE="eth0"
ONBOOT="yes"
IPADDR="10.0.0.88"
NETMASK="255.255.255.0"
GATEWAY="10.0.0.254"
DNS1="223.5.5.5"
DNS2="223.6.6.6"


#详解：

TYPE=Ethernet        #网卡类型，一般是Ethernet，还有其他的如bond，bridge

BOOTPROTO=dhcp       #获取IP地址的方式，启动的协议，获取配置的方式。
                             dhcp表示动态获取
                             static或none表示静态手工配置，若想使用本地配置好的IP则应该设置成这个

DEFROUTE=yes         #是否设置默认路由，若为yes则可以在该文件通过PREFIX这个参数来设置子网掩码

PEERDNS=yes          #yes表示由DHCP来获取DNS，no表示/etc/resolv.conf来控制，默认为yes。
                             yes：如果DNS设置，修改/etc/resolv.conf中的DNS
```

```
                                    no：不修改/etc/resolv.conf中的DNS

NAME=eth0                   #这个参数对应的值是网卡名，是给用户看的

UUID=...                    #通用唯一识别码，若vmware克隆的虚拟机无法启动网卡可以去除此项

DEVICE=eth0                 #系统逻辑设备名

ONBOOT=yes                  #开机启动时是否激活网卡设备，centos7装完网卡后默认设置成no

HWADDR=...                  #以太网硬件地址，mac地址）。若是vmware克隆的虚拟机无法启动网卡，也要改这个。

NM_CONTROLLED=yes     #是否通过NetworkManager管理网卡设备

IPADDR=...                  #设置网卡对应的IP地址，网络服务启动，网卡激活后会自动将该地址配置到网卡上
                                    前提（BOOTPROTO=static）而不是dhcp

PREFIX=24                   #子网掩码长度，不要这么写PREFIX=255.255.255.0

NETMASK=255.255.255.0     #生产环境中一般用这种方式指定子网掩码

GATEWAY=10.0.0.254    #该网卡配置的IP对应的网关（默认路由）
```

```
                        若主机是多网卡设备，该参数只能在一个网卡配置文件里面出现，一台主机只有一个默认路由

DNS1=...                #主DNS,若这里设置了值，则会优先于/etc/resolv.conf中设置的DNS服务器的地址
                        需要和“PEERDNS=no”配合使用

DNS2=...                #次dns

USERCTL=no              #USERCTL=yes/no是否允许非root用户控制该设备

IPV6INIT=no             #是否启用IPV6

BROADCAST=...           #广播地址

PROXY_METHOD=none       #代理方式，一般不用这个参数

BROWSER_ONLY=no         #没有什么用。

#修改网卡配置文件的方法

vim  /etc/sysconfig/network-scripts/ifcfg-eth0

nmtui    #需要开启NetworkManger
```

3. 网络管理命令

ping

```
ping命令主要的功能是用来检测网络的连通情况和分析网络速度。
```

```
#常用选项

    -t      #持续ping，不中断。不加该选项只ping4个包。
    -c      #ping的包数，默认是4个。
    -W      #多长时间ping一次。
    -f      #极速ping。

windows: ping不通不能表示服务器不能访问 可能服务器禁止了ICMP协议(禁ping)
C:\Users\oldboy-lidao996>ping -n 2 www.baidu.com

正在 Ping www.a.shifen.com [110.242.68.3] 具有 32 字节的数据:
来自 110.242.68.3 的回复: 字节=32 时间=11ms TTL=52
来自 110.242.68.3 的回复: 字节=32 时间=12ms TTL=52

110.242.68.3 的 Ping 统计信息:
    数据包: 已发送 = 2，已接收 = 2，丢失 = 0 (0% 丢失)，
往返行程的估计时间(以毫秒为单位):
    最短 = 11ms，最长 = 12ms，平均 = 11ms

centos:
[root@oldboyedu ~]#ping -c2 -W1 www.baidu.com
PING www.a.shifen.com (110.242.68.4) 56(84) bytes of data.
```

```
64 bytes from 110.242.68.4 (110.242.68.4): icmp_seq=1 ttl=128 time=16.3 ms
64 bytes from 110.242.68.4 (110.242.68.4): icmp_seq=2 ttl=128 time=11.1 ms

-c   ping的个数
-w1 延时1秒返回请求

```

nc

```
nc是netcat的简写，有着网络界的瑞士军刀美誉。因为它短小精悍、功能实用，被设计为一个简单、可靠的网络工具

#常用选项

     -l      #用于指定nc将处于侦听模式。
     -u      #指定nc使用UDP协议，默认为TCP
     -v      #输出交互或出错信息，新手调试时尤为有用
     -w      #超时秒数，后面跟数字
     -z      #表示zero，表示扫描时不发送任何数据
```

nmap

```
Nmap即网络映射器对Linux系统/网络管理员来说是一个开源且非常通用的工具。Nmap用于在远程机器上探测网络，执行安全扫描，网络审计和搜寻开放端口。

#常用选项

    -p              #指定端口号
    -p22            #单个端口
    -p22,80         #多个端口
    -p1-1000        #1到1000之间的端口


企业案例:
写一个脚本 探测自己所有的服务器上开放的服务以及端口并且计算一些每个服务占用所有服务的百分比
100台服务器
```

telnet

```
一种远程登录的工具。同样可以检查某个主机是否开启某个端口
默认端口23
#用法  主要功能链接路由器、交换机  telnet 192.168.13.1

[C:\~]$ telnet 10.0.0.99 22


Connecting to 10.0.0.99:22...
Connection established.     # 连接成功  说明22端口开启
To escape to local shell, press 'Ctrl+Alt+]'.
SSH-2.0-OpenSSH_7.4
```

netstat

```
打印网络连接、路由表、tcp11种状态。
查看当前系统中运行了哪些服务端口
#常用选项
    -l      #只显示监听套接字。
    -n      #不做名字解析
    -t      #显示tcp端口
    -u      #显示udp端口
    -p      #显示pid和程序名字
    -r      #显示路由表
    -a      #显示所有的套接字

netstat -tnulp
```

ss

```
跟netstat命令差不多。

#常用选项
    -l      #只显示监听套接字。
    -n      #不做名字解析
    -t      #显示tcp端口
    -u      #显示udp端口
    -p      #显示pid和程序名字
    -r      #解析主机名
    -a      #显示所有的套接字

```

tracert

```
(windows) 路由跟踪（检查你与目标之间每个路口是否畅通）

#常用选项

    -d    #禁止把IP解析为对应的域名（主机名）
C:\Users\oldboy-lidao996>tracert -d www.baidu.com

通过最多 30 个跃点跟踪
到 www.a.shifen.com [110.242.68.3] 的路由:

  1     1 ms     1 ms     2 ms  192.168.11.1
  2     2 ms    <1 毫秒     5 ms  192.168.1.1
  3     4 ms     3 ms     3 ms  221.218.208.1
  4     2 ms     8 ms     6 ms  61.148.162.57
  5     3 ms     7 ms     4 ms  202.106.34.1

```

```
  6     4 ms     3 ms     3 ms  202.96.12.1

 DNS流程:
 浏览器缓存-->HOSTS-->windows缓存-->LDNS-->根

 查看windows缓存:
 ipconfig/displaydns
C:\Users\oldboy-lidao996>ipconfig/flushdns

Windows IP 配置

已成功刷新 DNS 解析缓存。


cmd---->mstsc 调出远程桌面
```

traceroute

```
路由跟踪（检查你与目标之间每个路口是否畅通）

#常用选项

      -n    禁止把IP解析为对应的域名（主机名）
在Linux系统中使用:
[root@oldboyedu ~]#traceroute -n -I www.baidu.com
traceroute to www.baidu.com (110.242.68.4), 30 hops max, 60 byte packets
 1  10.0.0.2  0.216 ms  0.176 ms  0.100 ms
 2  192.168.11.1  25.034 ms  24.795 ms  24.615 ms
 3  192.168.1.1  22.107 ms  21.941 ms  21.748 ms
```

iftop

```
iftop界面说明：

界面上面显示的是类似刻度尺的刻度范围，为显示流量图形的长条作标尺用的。

中间的<= =>这两个左右箭头，表示的是流量的方向。

TX：发送流量
RX：接收流量
TOTAL：总流量
Cumm：运行iftop到目前时间的总流量
peak：流量峰值
rates：分别表示过去 2s 10s 40s 的平均流量

#常用选项

-i       #设定监测的网卡

-B       #以bytes为单位显示流量(默认是bits)

-n       #使host信息默认直接都显示IP

-P       #使host信息及端口信息默认就都显示

-m       #设置界面最上边的刻度的最大值，刻度分五个大段显示

按q退出监控。

yum -y install dstat
dstat -nf
```

网络重点:
1.网络常用命令
ping
ip add
route -n
ifconfig
nslookup
tcpdump
iftop
iotop
netstat -tnulp
ss -an|grep tcp
telnet

tracert
nmap

windows
ping
nslookup
mstsc
cmd
ipconfig/all
ipconfig/flushdns
ipconfig/displaydns

2.OSI七层模型

3.TCP三次握手

4.TCP四次挥手

5.DNS解析流程

6.子网掩码决定IP可用数量

7.静态路由动态路由理解

#下次内容 开始二阶段架构
#周一 综合考试