# 一、网络中常用的命令

1、ipconfig：用于显示当前的TCP/IP配置的设置值。这些信息一般用来检验人工配置的TCP/IP设置是否正确。

2、ping：这是ICMP的应用。它利用ICMP协议包来侦测另一个主机是否可达、即网络连通性。ping程序来计算间隔时间，并计算有多少个包被送达。用户就可以判断网络大致的情况。我们可以看到， ping给出来了传送的时间和TTL的数据。

3、traceroute: 也是ICMP的应用，是用来侦测主机到目的主机之间所经路由情况的重要工具。尽管ping工具也可以进行侦测，但是，因为ip头的限制，ping不能完全的记录下所经过的路由器。所以Traceroute正好就填补了这个缺憾。

4、用**host**命令，**host ip**就可以得到服务器的域名，**host domainName** 就得到IP。

5、ARP：是TCP/IP协议族中的一个重要协议，用于确定对应IP地址的网卡物理地址。

使用arp命令，能够查看本地计算机或另一台计算机的ARP高速缓存中的当前内容。此外，使用arp命令可以人工方式设置静态的网卡物理地址/IP地址对，使用这种方式可以为缺省网关和本地服务器等常用主机进行本地静态配置，这有助于减少网络上的信息量

6、nslookup：是查询任何一台机器的IP地址和其对应的域名。它通常需要一台域名服务器来提供域名。如果用户已经设置好域名服务器，就可以用这个命令查看不同主机的IP地址对应的域名。

7、nbtstat：可以查看计算机上网络配置的一些信息。使用这条命令还可以查找出别人计算机上一些私人信息。如果想查看自己计算机上的网络信息，可以运行nbtstat -n，可以得到你所在的工作组，计算机名以及网卡地址等等；

8、netstat：用来了解网络当前的状态。netstat命令能够显示活动的TCP连接、计算机侦听的端口、以太网统计信息、IP路由表、IPv4统计信息（对于IP、ICMP、TCP和UDP协议）以及IPv6统计信息（对于IPv6、ICMPv6、通过IPv6的TCP以及UDP协议）。使用时如果不带参数，netstat显示活动的TCP连接。



# 二、计算机网络之面试常考

## **1、网络分层**

**OSI分层 （7层）**：物理层、数据链路层、网络层、传输层、会话层、表示层、应用层。

**TCP/IP分层（4层）**：网络接口层、 网际层、运输层、 应用层。

**五层协议   （5层）**：物理层、数据链路层、网络层、运输层、 应用层。

**每一层的协议如下：**

|  |  |  |
| --- | --- | --- |
| 层次 | 协议 | 物理设备 |
| 物理层 | RJ45、CLOCK、IEEE802.3 | 中继器，集线器，网关 |
| 数据链路层 | PPP、FR、HDLC、VLAN、MAC | 网桥，交换机 |
| 网络层 | IP、ICMP、ARP、RARP、OSPF、IPX、RIP、IGRP | 路由器 |
| 运输层 | TCP、UDP、SPX |  |
| 应用层 | FTP、DNS、Telnet、SMTP、HTTP、WWW、NFS |  |

**每一层的作用如下：**

**物理层：**物理层的任务就是透明地传送比特流。物理层还要确定连接电缆插头的定义及连接法。（bit）

**数据链路层：**数据链路层的任务是在两个相邻结点间的线路上无差错地传送以帧为单位的数据。每一帧包括数据和必要的控制信息。

是点到点的传递（帧Frame）

**网络层：**网络层的任务就是要选择合适的路由，使发送站的运输层所传下来的分组能够正确无误地按照地址找到目的站，

并交付给目的站的运输层。（IP数据包PackeT）

**传输层：**运输层的任务是向上一层的进行通信的两个进程之间提供一个可靠的端到端服务，使它们看不见运输层以下的数据通信的

细节。【提供端到端的可靠报文传递和错误恢复（报文段Segment、用户数据报）】

**应用层：**应用层直接为用户的应用进程提供服务（应用协议数据单元APDU）

## **2、IP地址的分类**



A类地址：以0开头， 第一个字节范围：0~127（1.0.0.0 - 126.255.255.255）；

B类地址：以10开头， 第一个字节范围：128~191（128.0.0.0 - 191.255.255.255）；

C类地址：以110开头，第一个字节范围：192~223（192.0.0.0 - 223.255.255.255）；

10.0.0.0—10.255.255.255，172.16.0.0—172.31.255.255，192.168.0.0—192.168.255.255。（Internet上保留地址用于内部）

**IP地址与子网掩码相与得到主机号**

## **3、ARP是地址解析协议，简单语言解释一下工作原理。**

1）：首先，每个主机都会在自己的ARP缓冲区中建立一个ARP列表，以表示IP地址和MAC地址之间的对应关系。

2）：当源主机要发送数据时，首先检查ARP列表中是否有对应IP地址的目的主机的MAC地址，如果有，则直接发送数据，如果没有，就向本网段的所有主机发送ARP数据包，该数据包包括的内容有：**源主机** **IP地址，源主机MAC地址，目的主机的IP 地址**。

3）：当本网络的所有主机收到该ARP数据包时，首先检查数据包中的IP地址是否是自己的IP地址，如果不是，则忽略该数据包，如果是，则首先从数据包中取出源主机的IP和MAC地址写入到ARP列表中，如果已经存在，则覆盖，然后将自己的MAC地址写入ARP响应包中，告诉源主机自己是它想要找的MAC地址。

4）：源主机收到ARP响应包后。将目的主机的IP和MAC地址写入ARP列表，并利用此信息发送数据。如果源主机一直没有收到ARP响应数据包，表示ARP查询失败。

注意：广播发送ARP请求，单播发送ARP响应。 （广播发出去，单播收回来）

## **4、描述：RARP**

RARP是逆地址解析协议，作用是完成硬件地址到IP地址的映射；主要用于无盘工作站，因为给无盘工作站配置的IP地址不能保存。

工作流程：在网络中配置一台RARP服务器，里面保存着IP地址和MAC地址的映射关系，当无盘工作站启动后，就封装一个RARP数据包，里面有其MAC地址，然后广播到网络上去，当服务器收到请求包后，就查找对应的MAC地址的IP地址装入响应报文中发回给请求者。 注意：因为需要广播请求报文，因此RARP只能用于具有广播能力的网络。

## **5、TCP三次握手和四次挥手的全过程**

**1）、三次握手：**建立连接需要“三次握手”

第一次握手：客户端发送syn包(syn=1 ,seq=x)到服务器，并进入SYN\_SEND状态，等待服务器确认；

第二次握手：服务器收到syn包，必须确认客户的SYN（ack=x+1），同时自己也发送一个SYN包（syn=1,ACK=1,seq=y），即 SYN+ACK包，此时服务器进入SYN\_RECV状态；

第三次握手：客户端收到服务器的SYN＋ACK包，向服务器发送确认包ACK(ACK=1,seq=x+1,ack=y+1)，此包发送完毕，客户端和服务 器进入ESTABLISHED状态，完成三次握手。

握手过程中传送的包里不包含数据，三次握手完毕后，客户端与服务器才正式开始传送数据。理想状态下，TCP连接一旦建立，在通信双方中的任何一方主动关闭连接之前，TCP连接都将被一直保持下去。

**2）、四次握手** ：断开一个TCP连接则需要“四次握手”。



第一次挥手：主动关闭方发送一个FIN=1,seq=x，用来关闭主动方到被动关闭方的数据传送，也就是主动关闭方告诉被动关闭方：我已经不会再给你发数据了(当然，在fin包之前发送出去的数据，如果没有收到对应的ack确认报文，主动关闭方依然会重发这些数据)。但是，此时主动关闭方还可以接受数据。

第二次挥手：被动关闭方收到FIN包后，发送一个ACK给对方，确认序号为收到序号+1（与SYN相同，一个FIN占用一个序号）。  
第三次挥手：被动关闭方发送一个FIN，用来关闭被动关闭方到主动关闭方的数据传送，也就是告诉主动关闭方，我的数据也发送完了，不会再给你发数据了。  
第四次挥手：主动关闭方收到FIN后，发送一个ACK给被动关闭方，确认序号为收到序号+1，至此，完成四次挥手。

* **两次握手可以么？**

已失效的连接请求报文段”的产生在这样一种情况下：client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server。本来这是一个早已失效的报文段。但server收到此失效的连接请求报文段后，就误认为是client再次发出的一个新的连接请求。于是就向client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server发出确认，新的连接就建立了。由于现在client并没有发出建立连接的请求，因此不会理睬server的确认，也不会向server发送ACK包。这样就会白白浪费资源。

而经过三次握手，客户端和服务器都有应有答，这样可以确保TCP正确连接。

* **为什么TCP连接是三次，挥手确是四次？**

在TCP连接中，服务器端的SYN和ACK向客户端发送是一次性发送的。

在断开连接的过程中，B端向A端发送的ACK和FIN是分两次发送的。因为在B端接收到A端的FIN后，B端可能还有数据要传输，所以先发送ACK，等B端处理完自己的事情后就可以发送FIN断开连接了。

* **为什么在第四次挥手后会有2个MSL的延时？**

MSL（Maximum Segment Lifetime）：最大报文段生存时间，2个MSL是报文段发送和接收的最长时间。

假定网络不可靠，那么第四次发送的ACK可能丢失，即B端无法收到这个ACK，如果B端收不到这个确认ACK，B端会定时向A端重复发送FIN，直到B端收到A的确认ACK。所以这个2MSL就是用来处理这个可能丢失的ACK的。

## **6、在浏览器中输入一个网址（**[www.baidu.com](http://www.baidu.com)**）后执行的全部过程**

1、**客户端浏览器通过DNS解析到网址（**[**www.baidu.com**](http://www.baidu.com)**）的IP地址（**220.181.27.48）。

* 客户端先检查本地是否有对应的IP地址，若找到则返回响应的IP地址。
* 若没找到则请求上级DNS服务器，直至找到或到根节点。
* DNS查找IP地址的顺序: 浏览器缓存、系统缓存、互联网服务提供商（ISP）的DNS缓存、递归搜索（从浏览器缓存开始，如果没找到就继续往下一个找），最后浏览器会获得一个IP地址，通过这个IP地址找到客户端到服务器的路径。

**2、客户端浏览器发起一个HTTP请求到这个IP(220.161.27.48)**

**3、在传输层TCP传输报文**。TCP协议通过“三次握手”等方法保证传输的安全可靠。

在客户端的传输层，把HTTP会话请求分成报文段，添加源和目的端口，如服务器使用80端口监听客户端的请求，客户端由系统随机选择一个端口如5000，与服务器进行交换，服务器把相应的请求返回给客户端的5000端口。

4、**网络层IP协议查询MAC地址。**

IP协议的作用是把TCP分割好的各种数据包传送给接收方。而要保证确实能传到接收方还需要接收方的MAC地址，也就是物理地址。IP地址和MAC地址是一一对应的关系，一个网络设备的IP地址可以更换，但是MAC地址一般是固定不变的。ARP协议可以将IP地址解析成对应的MAC地址。当通信的双方不在同一个局域网时，需要多次中转才能到达最终的目标，在中转的过程中需要通过下一个中转站的MAC地址来搜索下一个中转目标。

**5、数据到达数据链路层**

在找到对方的MAC地址后，就将数据发送到数据链路层传输。这时，客户端发送请求的阶段结束

**6、服务器接收数据**

　　　接收端的服务器在链路层接收到数据包，再层层向上直到应用层。这过程中包括在运输层通过TCP协议将分段的数据包重新组成原来的HTTP请求报文。

7、**服务器响应请求**

　　 服务接收到客户端发送的HTTP请求后，查找客户端请求的资源，并返回响应报文。

**8、服务器返回响应文件**

　　 请求成功后，服务器会返回相应的HTML文件。接下来就到了页面的渲染阶段了。

9**、 页面渲染：**  解析HTML以构建DOM树 –> 构建渲染树 –> 布局渲染树 –> 绘制渲染树。

## **7、TCP和UDP的区别**

UDP协议在IP协议上增加了复用、分用和差错检测功能

TCP是在不可靠的IP层之上实现的可靠的数据传输协议，主要解决传输的可靠（由于使用了序号和确认）、有序、无丢失和不重复的问题。

|  |  |
| --- | --- |
| TCP | UDP |
| 提供面向连接的、可靠的数据流传输 | 提供的无连接的、不可靠的数据流传输 |
| 连接状态（点对点） | 无连接状态 |
| 传输单位：TCP报文段 | 传输单位：用户数据报 |
| 面向字节流 | 面向报文 |
| 首部开销：20字节 | 8字节 |
| 可靠交付 | 尽最大努力，不保证可靠交付 |
| 工作通信方式：双全工通信 | 工作通信方式：1对1、1对多、多对1、多对多 |
| 没有拥塞控制 | 具有拥塞控制 |
|  |  |

**流量控制：采用大小可变的滑动窗口进行流量控制；**

**拥塞控制：（1）、慢开始和拥塞避免； （2）、快重传和快恢复**

## **8、面向连接和非面向连接的服务的特点**

|  |  |  |
| --- | --- | --- |
|  | **面向连接服务** | **无连接服务** |
|  | 面向连接服务具有建立连接、数据传输和断开连接三个阶段 |  |
| 特点 | 1. 在数据交换之前必须建立连接 2. 数据交换之后，必须终止这个连接 3. 数据传送是按序传送的 4. 数据传输可靠性高 | 1. 两个实体间通信不需要先建立一个连接 2. 下层的有关资源不需要事先进行预留，这些资源在数据传输时动态地进行分配 3. 不需要通信的两个实体同时是活跃的。 4. 因为不需要接收端做任何响应，因而是一种不可靠的服务 5. 灵活方便和比较迅速 |
| 缺点 | 1. 协议复杂 2. 通信效率不高 | 1. 不能防止报文的丢失、重复或者丢失（后发先到）。 |
| 使用场景 | 1. 在一定期间内要向同一个目的地发送许多报文的情况 | 1、发送很短的零星报文 |
| 不适用场景 | 发送很短的零星报文 |  |

注：不需要通信的两个实体同时是活跃的：当发送端的实体正在进行发送时，它才必须是活跃的。这时接收端的实体并不一定必须是活跃的，只有当接收端的实体正在进行接收时，它才必须是活跃的。

## 9、DNS域名系统，简单描述其工作原理。

本地域名服务器：课本P259

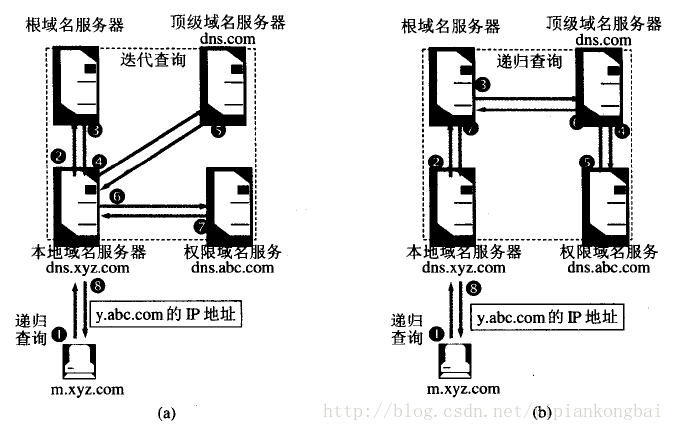
1. 主机向本地域名服务器的查询一般都是采用递归查询。

递归查询就是：如果主机所询问的本地域名服务器不知道被查询的域名的IP地址，那么本地域名服务器就以DNS客户的身份，向其它根域名服务器继续发出查询请求报文(即替主机继续查询)，而不是让主机自己进行下一步查询。因此，递归查询返回的查询结果或者是所要查询的IP地址，或者是报错，表示无法查询到所需的IP地址。

1. 本地域名服务器向根域名服务器的查询的迭代查询。

迭代查询的特点：当根域名服务器收到本地域名服务器发出的迭代查询请求报文时，要么给出所要查询的IP地址，要么告诉本地服务器：“你下一步应当向哪一个域名服务器进行查询”。然后让本地服务器进行后续的查询。

根域名服务器通常是把自己知道的顶级域名服务器的IP地址告诉本地域名服务器，让本地域名服务器再向顶级域名服务器查询。顶级域名服务器在收到本地域名服务器的查询请求后，要么给出所要查询的IP地址，要么告诉本地服务器下一步应当向哪一个权限域名服务器进行查询。最后，知道了所要解析的IP地址或报错，然后把这个结果返回给发起查询的主机。 下图给出了这两种查询的差别：



**下面举一个例子演示整个查询过程：**

**（1）、查询过程**

假定域名为m.xyz.com的主机想知道另一个主机y.abc.com的IP地址(也就是主机想访问y.abc.com)。

例如，主机m.xyz.com打算发送邮件给y.abc.com。这时就必须知道主机y.abc.com的IP地址。

下面是上图a的几个查询步骤：

1、主机m.abc.com先向本地服务器dns.xyz.com进行递归查询。( 如果查到，直接返回；如果没查到，则进行步骤2 )

2、本地服务器采用迭代查询。它先向一个根域名服务器查询。

3、根域名服务器告诉本地服务器，下一次应查询的顶级域名服务器dns.com的IP地址。

4、本地域名服务器向顶级域名服务器dns.com进行查询。

5、顶级域名服务器dns.com告诉本地域名服务器，下一步应查询的权限服务器dns.abc.com的IP地址。

6、本地域名服务器向权限域名服务器dns.abc.com进行查询。

7、权限域名服务器dns.abc.com告诉本地域名服务器，所查询的主机的IP地址。

8、本地域名服务器最后把查询结果告诉m.xyz.com。

整个查询过程共用到了8个UDP报文。

**（2）、优化过程**

为了提高DNS查询效率，并减轻服务器的负荷和减少因特网上的DNS查询报文数量，在域名服务器中广泛使用了高速缓存，用来存放最近查询过的域名以及从何处获得域名映射信息的记录。

例如，在上面的查询过程中，如果在m.xyz.com的主机上不久前已经有用户查询过y.abc.com的IP地址，那么本地域名服务器就不必向根域名服务器重新查询y.abc.com的IP地址，而是直接把告诉缓存中存放的上次查询结果(即y.abc.com的IP地址)告诉用户。

由于名字到IP地址的绑定并不经常改变，为保持高速缓存中的内容正确，域名服务器应为每项内容设置计时器并处理超过合理时间的项(例如每个项目两天)。当域名服务器已从缓存中删去某项信息后又被请求查询该项信息，就必须重新到授权管理该项的域名服务器绑定信息。当权限服务器回答一个查询请求时，在响应中都指明绑定有效存在的时间值。增加此时间值可减少网络开销，而减少此时间值可提高域名解析的正确性。

不仅在本地域名服务器中需要高速缓存，在主机中也需要。许多主机在启动时从本地服务器下载名字和地址的全部[数据库](http://lib.csdn.net/base/mysql)，维护存放自己最近使用的域名的高速缓存，并且只在从缓存中找不到名字时才使用域名服务器。维护本地域名服务器数据库的主机应当定期地检查域名服务器以获取新的映射信息，而且主机必须从缓存中删除无效的项。由于域名改动并不频繁，大多数网点不需花精力就能维护数据库的一致性。

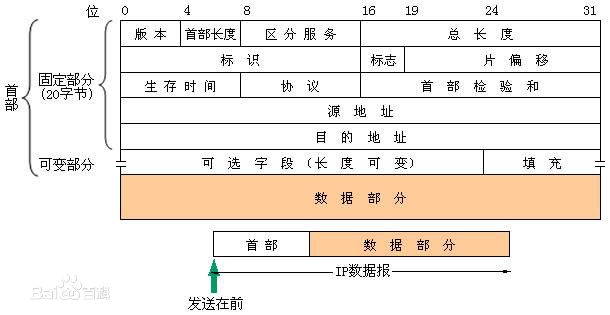
## **10、TCP对应的协议和UDP对应的协议对比**

|  |  |
| --- | --- |
| **TCP对应的协议** | **UDP对应的协议** |
| **FTP**：定义了文件传输协议，使用**21**端口 | **DNS**： 用于域名解析服务，将域名地址转换为IP地址。DNS用的是**53**号端口。 |
| **SMTP**：邮件传送协议，用于发送邮件。服务器开放的是**25**号端口 | **SNMP**：简单网络管理协议，使用**161**号端口，是用来管理网络设备的。由于网络设备很多，无连接的服务就体现出其优势。 |
| **POP3**：它是和SMTP对应，POP3用于接收邮件。POP3协议所用的是**110**端口。 | **TFTP**：简单文件传输协议，该协议在熟知端口**69**上使用UDP服务。 |
| **HTTP**：是从Web服务器传输超文本到本地浏览器的传送协议。 |  |
| **Telnet**：一种用于远程登陆的端口，使用**23**端口，用户可以以自己的身份远程连接到计算机上，可提供基于DOS模式下的通信服务。 |  |

## **11、端口及对应的服务**

|  |  |  |  |
| --- | --- | --- | --- |
| 服务 | 端口 | 服务 | 端口 |
| FTP | 21 | SSH | 22 |
| telnet | 23 | SMTP | 25 |
| Domain | 53 | HTTP | 80 |
| POP3 | 110 | NTP(网络时间协议) | 123 |
| MySql | 3306 | Shell或者cmd | 514 |
| POP-2 | 109 | SQL Server | 1433 |

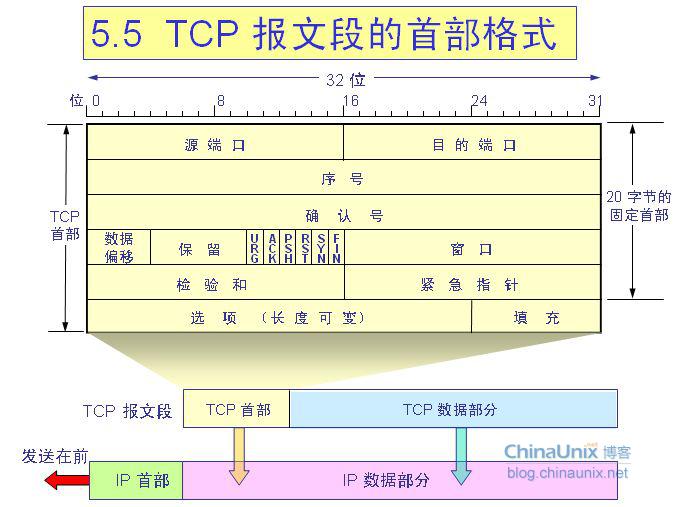
## **12、IP数据包的格式**



IP数据报由**首部**和**数据**两部分组成。首部由**固定部分和可选部分**组成。**首部的固定部分有** **20 字节**。可选部分的长度变化范围为1——40字节。固定部分的字段：

|  |  |  |
| --- | --- | --- |
| 字段名 | 位数（bit） | 解释 |
| 版本 | 4 | 指IP协议的版本。目前广泛使用的IP协议版本号为4（IPV4） |
| 首部长度 | 4 | 可表示的最大数为15个单位（一个单位表示4字节），因此IP首部的长度最大为60字节  注：当IP分组首部长度不是4字节的整数倍时，必须利用最后的填充字段加以填充 |
| 服务类型 | 8 | 用来获取更好的服务；之前很少用，目前多用于将实时多媒体信息在因特网上传送 |
| 总长度 | 16 | 总长度是指：首部和数据部分之和，因此数据报的最大长度为65535字节，即64KB，但是由于链路层的MAC都有一定的最大传输单元（MTU），因此IP数据报的长度一般都不会有理论上的那么大，如果超出了MAC的最大单元就会进行分片。 |
| 标识 | 16 | 相同的标识使得分片后的数据报片能正确的重装成原来的数据报 |
| 标志 | 3 | 目前只有前两个bit有意义。  最低位MF=1表示后面还有分片，MF=0表示这是若干个数据报片的最后一个  中间位DF=0才允许分片；DF=1的意思是不能分片 |
| 片偏移 | 13 | 片偏移指出较长的分组在分片后，某片在原分组中的相对位置（是相对于用户数据字段的起点）  注意：都是8字节为单位。 |
| 生存时间 | 4 | 数据报在网络中的生存时间：最多经过路由器的跳数 |
| 协议 | 4 | 指出该数据报携带的数据是何种协议，以使得目的主机的IP层知道应将数据部分上交给哪个处理程序）  ICMP=1 IGMP=2 TCP=6 EGP=8 IGP=9 UDP=17 Ipv6=41 OSPF=89 |
| 首部校验和 | 16 | 只校验首部，不包括数据部分。  原因：IP数据报每经过一个结点，结点处理机都要重新计算一下首部校验和（因为有些字段的值发上改变），如果将数据一起计算，计算的工作量太大了。  计算方法：将首部划分为多个16位的部分，然后每个16位取反，计算和，再将和取反放到首部校验和。接收方收到后按同样的方法划分，取反，求和，再取反。  如果结果为零，则接收，否则就丢弃 |
| 源地址 | 32 |  |
| 目的地址 | 32 |  |
| 可选字段 | 1-40个字节 |  |

## **13、TCP报文段的格式**



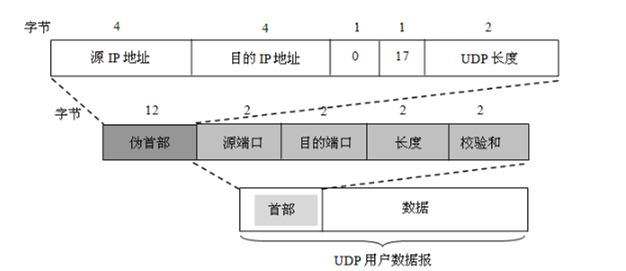
一个TCP报文段分为**首部**和**数据**两部分。首部由**固定部分**和**选项部分**组成，固定部分是20字节。TCP首部的最大长度为60。首部固定部分字段：

|  |  |  |
| --- | --- | --- |
| 字段名 | 位数（bit） | 解释 |
| 源端口 | 16 | 端口是运输层与应用层的服务接口。运输层的复用和分用功能都要通过端口才能实现 |
| 目的端口 | 16 |
| 序号 | 32 | TCP把在一个TCP连接中传送的数据流中的每一个字节都编上一个序号。  序号字段的值是指本报文段所发送的数据的第一个字节的序号。 |
| 确认号 | 32 | 是期望收到对方的下一个报文段的数据的第一个字节的序号，也就是期望收到的下一个报文段首部的序号字段的值。 |
| 数据偏移 | 4 | 指出TCP报文段的数据起始处距离TCP报文段的起始有多远。其实就是TCP首部的长度  必要性：因为有长度不确定的选项字段，所以数据偏移字段是必要的。  注：是以4字节为单位 |
| 保留 | 6 | 目前位置都为0 |
| 用于说明本报文段性质的控制比特 | 1 | 紧急比特URG：当URG= 1时，表明紧急指针字段有效。需要和紧急指针结合使用。 |
| 1 | 确认比特ACK: 只有当ACK=1时，确认号字段才有效 |
| 1 | 推送比特PSH：尽快交付给接收应用进程，而不需要等待整个缓存都填满后再向上交付。 |
| 1 | 复位比特RST: 当RST=1时，表明TCP连接中出现严重差错，必须释放连接。 |
| 1 | 同步比特SYN: 在连接时用来同步序号。  SYN=1 ACK=0 表明是连接请求报文  SYN=1 ACK=1 表明同意建立连接。 |
| 1 | 终止比特FIN:用来释放一个连接。 |
| 窗口 | 16 | 用来控制对方发送的数据量（当前我能都接受窗口的剩余部分），单位为字节。用WIN表示  用接收端的接收能力的大小来控制发送端的数据发送量。 |
| 校验和 | 16 | 包括首部和数据两部分，同时还要加12字节的伪首部进行校验和计算. |
| 紧急指针 | 16 | 紧急指针指出在本报文段中的紧急数据的最后一个字节的序号。紧急指针使接收方知道紧急数据共有多少个字节。注意：即使窗口为0，也可以发送紧急数据。 |
| 可选项 | 1--40 | TCP只有一种选项：最大报文段长度（MSS）。MSS告诉TCP：我的缓存所能接收的报文段的数据字段的最大长度时MSS个字节。  MSS的默认值是536个字节。那么报文段的长度就是556个字节。 |
| 填充 |  |  |

**TCP的12字节伪首部：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 源IP地址（4字节） | 目的IP地址（4字节） | 0 (1字节) | 6(1字节) 代表这是TCP | TCP长度（2） |

## **14、TCP数据报的格式**



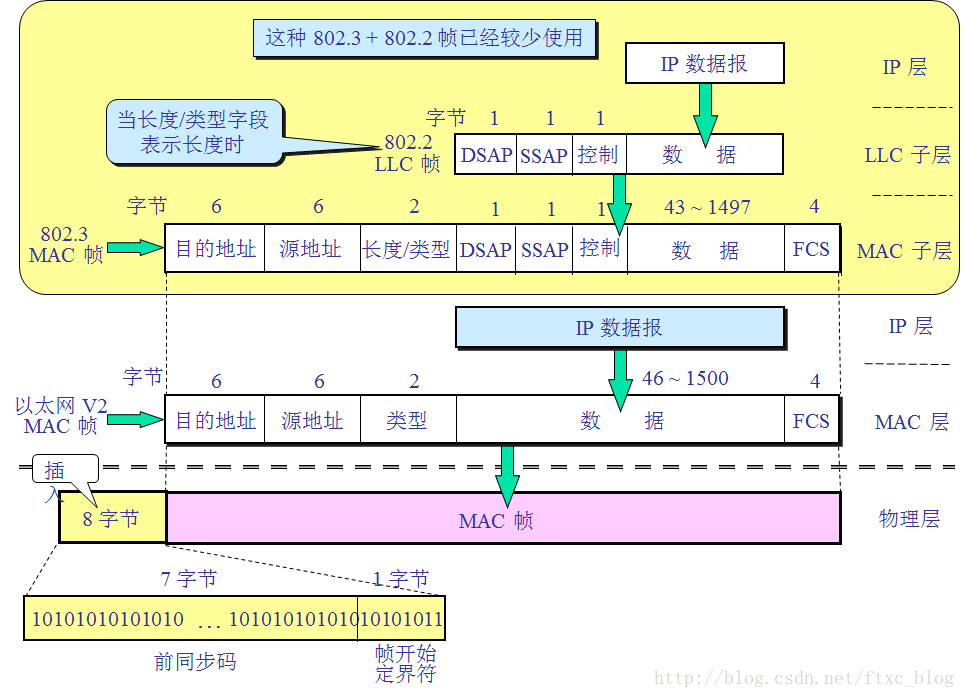
用户数据报UDP由首部和数据部分组成。首部只有8个字节，由4个字段组成，每个字段都是两个字节。

|  |  |  |
| --- | --- | --- |
| 字段名 | 位（bit） | 解释 |
| 源端口 | 16 |  |
| 目的端口 | 16 |  |
| 长度 | 16 | UDP数据报的长度 |
| 校验和 | 16 | 将首部和数据一起校验 |

**UDP的12字节伪首部： 伪首部仅仅是用作计算校验和，既不向下传送也不向上递交。**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 源IP地址（4） | 目的IP地址（4） | 0 (1) | 17(1) 代表这是UDP | UDP长度（2） |

## **15、以太网MAC帧格式**



**以太网MAC帧长度不小于64字节。**

|  |  |  |
| --- | --- | --- |
| 字段名 | 字节（Byte） | 解释 |
| 目的地址 | 6 |  |
| 源目的地址 | 6 |  |
| 字段类型 | 2 | 用来标识上一层使用什么协议，以便把收到的MAC帧的数据上交给上一层的这个协议  Ox0800:表示上一层使用的是IP数据报； |
| 数据字段 | 46--1500 | 46 = 最小长度64 - （ 6 + 6 + 2 + 4） |
| 帧校验序列FCS | 4 |  |

注意：有效的MAC帧长度为64—1518之间。

## 16、了解网关、交换机、路由器的概念，并知道各自的用途

物理层： 中继器，集线器，网关

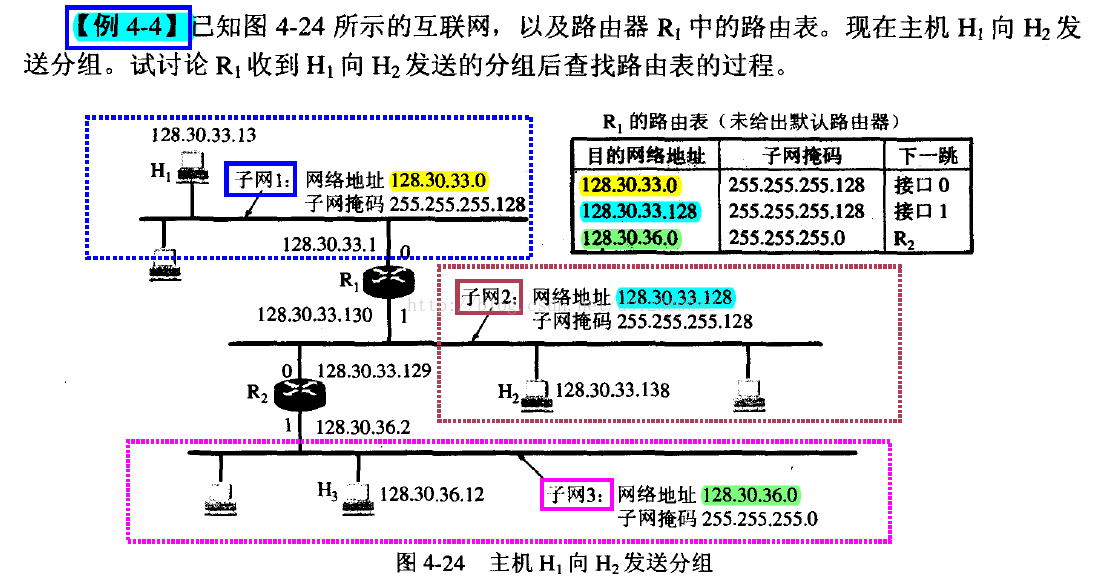
数据链路： 网桥，交换机

网络层： 路由器

运输层：

应用层：

## 17、**基于子网掩码的分组转发规则的一个实例**



**主机H1向H2发送分组的过程：**

首先，判断是直接交付还是简介交付：

同本子网内的子网掩码：255.255.255.128与目标主机H2的IP地址：128.30.33.128 相与得到网络号：128.30.33.128

如果结果等于H1的网络地址，说明目的主机与主机H1是连接在同一个子网上，可以直接交付，不需要找下一跳路由器来转发。

如果结果不等于H1的网络地址，则表明应采取间接交付，必须将分组交给本子网上的一个路由器进行转发。

显然这与主机H1所在的子网1的网络号：128.30.33.0不匹配。

间接交付：

此时，IP数据报就必须交给子网1的默认路由R1，由其转发。

（1）、R1先查找自己的路由表中的第一行。使用子网掩码（255.255.255.128）和目的地址（128.30.33.138）相与，得出128.30.33.128，但是结果与第一行的目的网络地址（12。30.33.0）不匹配。同样的方法去尝试第二行。

（2）、使用子网掩码（255.255.255.128）和目的地址（128.30.33.138）相与，得出128.30.33.128，结果和第二行的目的网络地址相匹配，说明这个网络（子网2）就是收到的分组所要寻找的目的网络。

（3）、找到以后就不需要再继续找了。直接使用下一跳的（接口1）直接交付给主机H2(他们在一个子网上)。