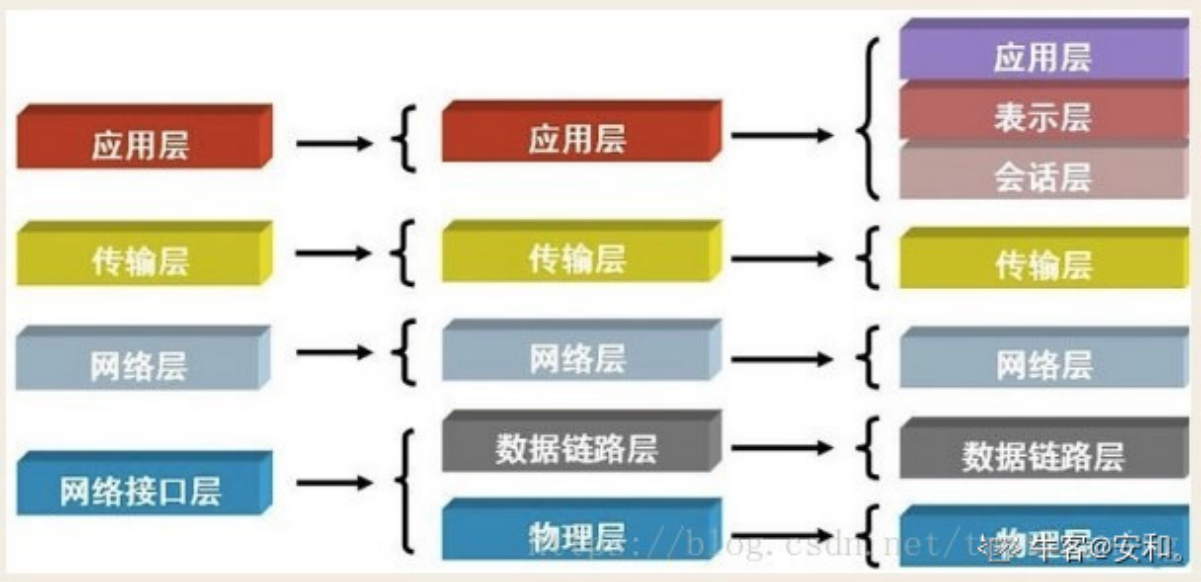
2.网络编程（29道）

2.1列举一下OSI协议的各种分层。说说你最熟悉的一层协议的功能。



（1）七层划分为：应用层、表示层、会话层、传输层、网络层、数据链路层、物理层。
（2）五层划分为：应用层、传输层、网络层、数据链路层、物理层。
（3）四层划分为：应用层、传输层、网络层、网络接口层。（TCP/IP协议对应模型）
（4）各层功能：

应用层	在实现多个应用进程相互通信的同时，完成一系列业务处理所需的服务，比如电子邮件、文件传输、远程登录等。
传输层	为通信双方的主机提供端到端的服务，有两个不同的传输协议TCP和UDP，TCP提供可靠交付，而UDP并不能保证可靠交付。
网络层	处理分组在网络中的活动，例如分组的选路。
网络接口层	处理与电缆（或其他任何传输媒介）的物理接口细节。

2.2 TCP/IP协议包括?

应用层	Telnet（远程登录服务）、FTP（文件传输，使用TCP）、SMTP（建立于

	FTP上的邮件服务）、DNS（域名与IP地址相互转换）等
传输层	UDP（无连接、不可靠）、TCP（面向连接、可靠传输）
网络层	IP（为主机提供一种无连接、不可靠、尽力而为的数据服务）、ICMP（主机与路由器之间传递控制信息）、IGMP（主机与路由器之间进行组播成员信息交互）
网络接口层	ARP（IP 地址-> MAC地址）、RARP（MAC地址 -> IP地址）等

2.3 TCP通信建立和释放的过程？端口的作用？

（1）连接是三次握手，释放是四次挥手。

（2）端口是一个软件结构，被客户进程或服务进程用来发送和接收信息。一个端口对应一个16比特的数。服务进程通常使用一个固定的端口。

2.4 IP地址转换成物理地址的协议？反之？

答案：

（1）将IP地址转换成物理地址的协议是ARP（地址解析协议）。

（2）反之则是RARP（反地址解析协议）。

解读：

（1）ARP协议工作流程：

①首先，每台主机都会在自己的ARP缓冲区(ARP Cache)中建立一个ARP列表，以表示IP地址和MAC地址的对应关系。

②当源主机需要将一个数据包要发送到目的主机时，会首先检查自己ARP列表中是否存在该IP地址对应的MAC地址，如果有，就直接将数据包发送到这个MAC地址；如果没有，就向本地网段发起一个ARP请求的广播包，查询此目的主机对应的MAC地址。此ARP请求数据包里包括源主机的IP地址、硬件地址、以及目的主机的IP地址。

③网络中所有的主机收到这个ARP请求后，会检查数据包中的目的IP是否和自己的IP地址一致。如果不相同就忽略此数据包；如果相同，该主机首先将发送端的MAC地址

和IP地址添加到自己的ARP列表中，如果ARP表中已经存在该IP的信息，则将其覆盖，然后给源主机发送一个ARP响应数据包，告诉对方自己是它需要查找的MAC地址；

④源主机收到这个ARP响应数据包后，将得到的目的主机的IP地址和MAC地址添加到自己的ARP列表中，并利用此信息开始数据的传输。如果源主机一直没有收到ARP响应数据包，表示ARP查询失败。

（2）RARP协议工作流程：

①网络上的每台设备都会有一个独一无二的硬件地址，通常是由设备厂商分配的MAC地址。PC1从网卡上读取MAC地址，然后在网络上发送一个RARP请求的广播数据包，请求RARP服务器回复该PC的IP地址。

②RARP服务器收到了RARP请求数据包，为其分配IP地址，并将RARP回应发送给PC1。

③PC1收到RARP回应后，就使用得到的IP地址进行通讯。

2.5 IP地址的编码分为哪两部分?

IP地址由两部分组成，网络号和主机号。不过是要和“子网掩码”按位与上之后才能区分哪些是网络位哪些是主机位。

2.6应用程序ping发出的是什么报文?

答案：应用程序ping发出的是ICMP请求报文。

解读：ping的原理是利用网络上机器IP地址的唯一性，给目标IP地址发送一个数据包，通过对方回复的数据包来确定两台网络机器是否连接相通，时延是多少。

2.7 socket编程的流程?

（1）服务器端流程：

函数	作用
socket()	创建套接字

bind()	绑定本地IP地址和端口号
listen()	设置监听队列长度
accept()	等待连接
read()	接收信息
close()	关闭套接字

（2）客户端流程：

函数	作用
socket()	创建套接字
connect()	发送连接请求
write()	发送信息
close()	关闭套接字

2.8 epoll是什么?

（1）epoll是Linux网络编程中用于处理大批量文件描述符的机制，是对select/poll的改进。

（2）select监听的fd是有上限的，32位处理器一般为1024；且select/poll每次调用会遍历所有fd，时间复杂度为O(n)，效率太低。而epoll监听的fd数量没有限制，且能在O(1)的时间复杂度内完成操作。

（3）epoll相关的系统调用有：epoll_creat、epoll_ctl、epoll_wait/epoll_pwait（可屏蔽特定信号），分别用来创建一个epoll文件描述符、添加/删除/修改需要侦听的文件

描述符及事件、接收被侦听描述符的IO事件。epoll文件描述符用完之后直接close关闭即可。

2.9 TCP、UDP的区别?

（1）TCP是面向连接的，UDP是面向无连接的。

（2）TCP是面向字节流的，UDP是基于数据报的。

（3）TCP提供可靠服务（正确性、顺序性），UDP提供不可靠服务。

（4）TCP程序结构复杂，占用资源多；UDP程序结构简单，占用资源少。

（5）TCP有拥塞控制；UDP没有拥塞控制，因此网络出现拥塞不会使源主机的发送速率降低，适合于实时应用，如IP电话、实时视频会议。

（6）TCP只支持一对一；UDP支持一对一、一对多、多对一、多对多。

2.10 TCP、UDP的优缺点?

（1）TCP

优点：稳定可靠。在传输数据之前，会有三次握手来建立连接；在数据传输时，有校验和、确认、重传、序列号、窗口、拥塞控制机制；在数据传输结束后，还会有四次挥手来断开连接，节约系统资源。

缺点：效率低、占用系统资源多、易受攻击。建立连接和确认、重传、窗口、拥塞控制机制都会消耗大量时间；每台设备维护连接都会占用CPU、内存等资源；三次握手机制、确认机制导致TCP易受利用和攻击。

（2）UDP

优点：效率高、占用系统资源较少、比TCP稍安全。没有各种复杂机制，传输速度快，占用系统资源较少，被利用的漏洞也少一些。

缺点：不稳定、不可靠。没有各种复杂机制，网络质量不好时很容易丢包。

2.11 TCP、UDP使用场景?

（1）TCP：对网络质量有要求时，比如HTTPS、HTTP、FTP等文件传输协议，POP、SMTP等邮件传输协议。

（2）UDP：对网络质量要求不高时，要求网络通讯速度要快的场景，比如在线视频、网络语音电话、广播通信。

2.12 TCP为什么是可靠连接?

因为TCP传输的数据满足四大条件：不出错、不丢失、不重复、不乱序，而且拥有窗口机制、拥塞控制机制来提高传输效率。

2.13 TCP如何保证可靠传输?

（1）校验和：发送数据报的二进制相加然后取反，检测数据在传输过程中的变化，有差错则丢弃。

（2）确认应答：接收方收到正确的报文就会确认。

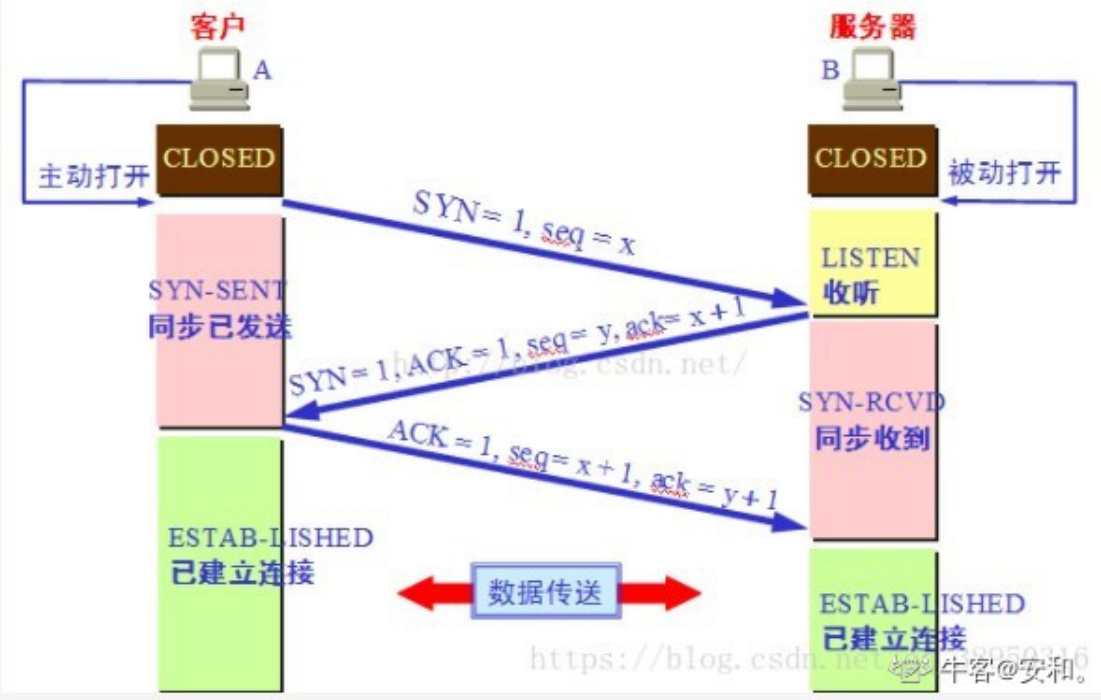
（3）超时重传：发送方等待一定时间后没有收到确认报文则重传。

（4）序列号：发送方对每一个数据包编号，接收方对数据包排序，保证不乱序、不重复。

（4）窗口机制（流量控制）：双方会协调发送的数据包大小，保证接收方能及时接收。

（5）拥塞控制机制：如果网络拥塞，发送方会降低发送速率，降低整个网络的拥塞程度。

2.14简述三次握手的过程。



（1）最初两端的TCP进程都处于CLOSED状态，A主动打开链接而B被动打开链接。B的TCP服务器先创建传输控制块TCB，然后服务器进程就处于LISTEN状态，等待客户端的连接请求。若有则作出响应。

（2）第一次握手：客户端创建传输控制块，然后向服务器发出连接请求报文（将标志位SYN置1，随机产生一个序列号seq=x），接着进入SYN-SENT状态。

（3）第二次握手：服务器收到请求报文后由SYN=1得到客户端请求建立连接，回复一个确认报文（将标志位SYN和ACK都置1，ack=x+1，随机产生一个序列号seq=y），接着进入SYN-RCVD状态。此时操作系统为该TCP连接分配TCP缓存和变量。

（4）第三次握手：客户端收到确认报文后，检查ack是否为x+1，ACK是否为1，是则发送确认报文（将标志位ACK置1，ack=y+1，序列号seq=x+1），此时操作系统为该TCP连接分配TCP缓存和变量。服务器收到确认报文并检查无误后则连接建立成功，两者都进入ESTABLISHED状态，完成三次握手。

2.15为什么需要三次握手，第三次握手去掉行不行?

（1）不行。

（2）首先，第三次握手是为了防止已经失效的连接请求报文突然传输到了服务器，从而建立错误的连接，浪费资源。

（3）其次，第三次握手还能防止发生死锁，因为若为两次握手且服务器发出第二次握手而客户端没有收到，服务器开始传输数据报后客户端便不会理会，导致服务器以

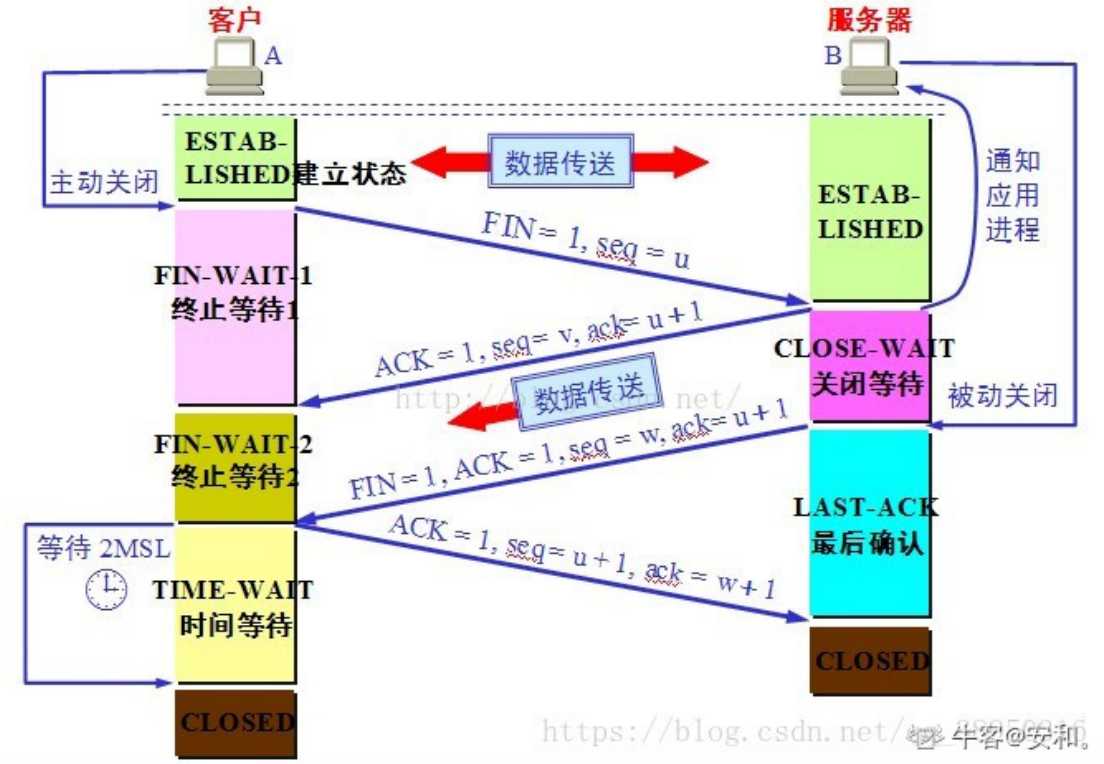
为丢包而源源不断地发送数据报，造成死锁。

2.16为什么服务器端易受SYN攻击？有什么防范措施吗？

（1）因为服务器端的资源是在第二次握手时分配的，而客户端的资源是在第三次握手时才分配的，若服务器收到大量伪造的IP地址发来的SYN包，则需要不断回复并等待确认，由于等待不到确认这些连接就会占满未连接队列，导致后来正常的连接请求报文被丢弃，从而引起网络拥塞甚至系统瘫痪。

（2）防范措施：降低主机的等待时间使主机尽快释放半连接的占用、短时间内收到某IP的重复连接请求报文则丢弃后续请求。

2.17简述四次挥手的过程。



（1）第一次挥手：客户端发出连接释放报文（FIN=1，seq=u），进入FIN-WAIT-1状态。

（2）第二次挥手：服务器收到连接释放报文后发出确认报文（ACK=1，ack=u+1，seq=v），进入CLOSE-WAIT状态。这时客户端向服务器方向的连接就释放了，这个连接处于半关闭状态，服务器还可继续发送数据。

（3）中间状态：客户端收到服务器的确认报文后，进入FIN-WAIT-2状态，等待服务器发送连接释放报文，此时仍要接收数据。

（4）第三次挥手：服务器最后的数据发送完，向客户端发送连接释放报文（FIN=1，ACK=1，ack=u+1，seq=w），进入LAST-ACK状态。

（5）第四次挥手：客户端收到服务器的连接释放报文后，必须发出确认报文（ACK=1，ack=w+1，seq=u+1），进入TIME-WAIT状态。注意此时连接还未释放，必须进过2*MSL（最长报文寿命）的时间，客户端撤销相应的TCB后，才进入CLOSED状态。服务器一旦收到确认报文，立即进入CLOSED状态。

2.18为什么客户端最后还要等待2*MSL（Maximun Segment Lifetime）？

（1）第一，保证客户端发送的最后一个ACK报文能够到达服务器，因为这个ACK报文可能丢失，服务器等待2*MSL后，会重新发送连接释放报文。等待2*MSL保证如果确认报文丢失客户端可以收到重传的连接释放报文，然后再次发送确认报文。

（2）第二，使本连接持续的时间内所产生的所有报文段都从网络中消失，不会出现类似“两次握手”的意外情况。

2.19为什么建立连接是三次握手，而关闭连接是四次挥手?

（1）建立连接的时候，服务器在LISTEN状态下，不需要等待，可以立即建立连接，因此只需要三次握手。

（2）而关闭连接时，服务器收到对方的FIN报文时，仅仅表示对方不再发送数据了但是还能接受数据，而自己未必将所有数据都发送完了，所以服务器可以立即关闭，也可以发送一些数据后再关闭，所以服务器的确认报文（ACK=1）和连接释放报文（FIN=1，ACK=1）一般分开发送，因此形成四次握手。

2.20 DNS服务器和客户机设置完毕后，有哪三个命令可以测试其设置是否正确?

（1）PING：是工作在TCP/IP网络体系结构中应用层的一个服务命令，主要是向特定的目的主机发送ICMP Echo请求报文，测试目的站是否可达及了解其有关状态。

（2）IPCONFIG：可用于显示当前TCP/IP配置的设置值。

（3）NSLOOKUP：用于查询DNS记录（nslookup domain [dns-server]）、查看域名解析是否正常、在网络故障的时候用来诊断网络问题。

2.21网络编程的作用?

网络编程的本质使用套接字来进行进程间通信，但这些进程是分布在不同主机上的进程，因此网络编程在物联网方面有比较大的作用。

2.22 socket网络编程中，哪五个元素可以明确标识一条连接?

五元组：客户端端口、客户端IP、服务端端口、服务端IP、传输协议。

2.23什么是TCP粘包?

TCP粘包就是指发送方发送的若干数据包到达接收方时粘成了一个包。从接收缓冲区来看，后一包数据的头紧接着前一包数据的尾。

2.24造成TCP粘包的原因?

（1）发送方原因：TCP默认使用Nagle算法（主要作用：减少网络中报文段的数量），Nagle算法主要做两件事情：

①只有上一个分组得到确认，才会发送下一个分组。

②收集多个小分组，在一个确认到来时一起封装并发送。

（2）接收方原因：TCP接收到数据包，应用层并不会立即处理。实际上，TCP将收到的数据包保存在接收缓存里，然后应用程序主动从缓存读取收到的分组。这样一来，如果TCP接收数据包到缓存的速度大于应用程序从缓存中读取数据包的速度，多个包就会被缓存，应用程序就有可能读取到多个首尾相接的包。

2.25什么时候需要处理粘包现象?

（1）如果发送方发送的多组数据本来就是同一块数据的不同部分，比如说一个文件被分成多个部分发送，这时当然不需要处理粘包现象。

（2）如果多个分组毫不相关，甚至是并列关系，那么就一定要处理粘包现象了。

2.26如何解决粘包问题?

（1）发送方：可以使用TCP_NODELAY选项来关闭Nagle算法。

（2）接收方：接收方无法处理，只能将问题交给应用层来处理。

（3）应用层：应用层的解决办法简单可行，不仅能解决接收方的粘包问题，还能解决发送方的粘包问题。

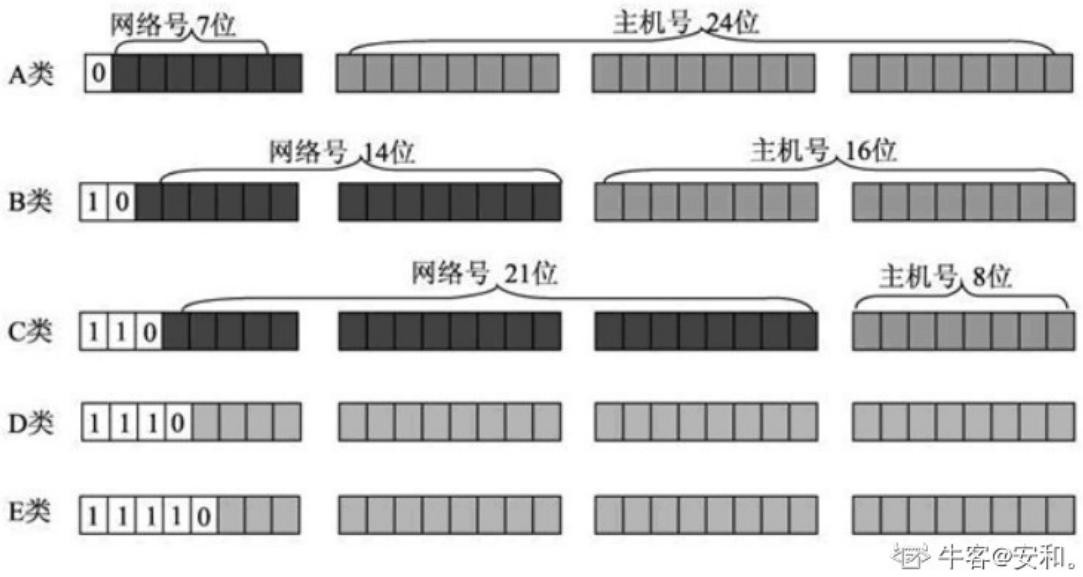
①格式化数据：每条数据都有固定的格式（开始符、结束符），这种方法简单易行，但是选择开始符和结束符时一定要确保每条数据的内部不包含这两种符号。

②发送数据包长度：发送每条数据时，将数据长度一并发送，例如规定数据的前四位是数据的长度，这样接收方应用层在处理时就可以根据长度来判断每个分组的开始和结束位置。

2.27 IP地址的分类?

答案：A类分配给政府机关单位使用，B类分配给中等规模企业使用，C类分配给任何需要的人使用，D类用于多播，E类用于实验。

解读：



（1）由于每类地址的开头是固定的，因此每类地址都有自己的范围：

①A类：0.0.0.0 ~ 127.255.255.255。

②B类：128.0.0.0 ~ 191.255.255.255。

③C类：192.0.0.0 ~ 223.255.255.255。

④D类：224.0.0.0 ~ 239.255.255.255。

⑤E类：240.0.0.0 ~ 255.255.255.254。

（2）在进行IP地址的分配时，有一些IP地址具有特殊含义，不会分配给互联网的主机。保留了一些IP地址用于私有网络，称为私有地址；保留一些IP地址用于测试，称为保留地址。

①A类私有地址范围：10.0.0.0 ~ 10.255.255.255；

A类保留地址范围：127.0.0.0 ~ 127.255.255.255。

②B类私有地址范围：172.16.0.0 ~ 172.31.255.255；

B类保留地址范围：169.254.0.0 ~ 169.254.255.255。

③C类地址私有范围：192.168.0.0 ~ 192.168.255.255。

2.28 IP地址与MAC地址的相同点和不同点?

（1）相同点：都是唯一的。

（2）不同点：

①分配依据不同：IP地址的分配是基于网络拓扑设计的，可以改动；MAC地址是基于制造商的，不能改动。

②长度不同：IP地址为32位，MAC地址为48位。

③寻址协议不同：IP地址应用于OSI第三层，即网络层；MAC地址应用于OSI第二层，即数据链路层。网络层协议可以使数据从一个网络传递到另一个网络（ARP根据目的IP地址，找到中间节点的MAC地址，通过中间节点传送，从而到达目的网络）；数据链路层协议可以使数据从一个节点传递到相同链路的另一个节点上（通过MAC地址）。

2.29路由器、交换机、集线器的区别?

（1）路由器工作在网络层，根据网络层提供的信息（IP地址）来选择路由。

（2）交换机工作在数据链路层，通过数据链路层提供的信息（MAC地址）来选择端口。

（3）集线器工作在物理层，只对信号进行整形、放大后再重发（广播给其他所有端口）。

讨论 评论

帖子还没人回复，快来抢沙发！