



多路复用器 (Multiplexer) 是一种电子设备，它的主要功能是将多个输入信号合并为一个输出信号，以实现在单个通信通道上同时传输多个数据流的能力。它通常被用于计算机网络、数字电视、电话网络等领域。

具体来说，多路复用器可以实现以下功能：

1. 多个输入信号转换为一个输出信号：多路复用器可以将多个输入信号合并为一个输出信号，这样就可以将多个数据流通过一个通信通道传输，从而减少通信通道的数量。
2. 在单一通道上同时传输多个数据流：多路复用器可以在单个通信通道上同时传输多个数据流，从而提高通信通道的利用率。
3. 管理带宽：多路复用器可以根据不同的输入信号，对输出信号进行合理的分配，从而实现带宽管理。
4. 减少信号干扰：多路复用器可以将多个输入信号合并为一个输出信号，从而减少信号之间的干扰。

总的来说，多路复用器的主要功能是将多个信号合并为一个信号，从而提高通信效率和降低通信成本。



在一个IP地址中，通常将地址分为网络地址和主机地址两部分，以便在网络中进行路由和转发。网络地址用于标识一个网络，而主机地址用于标识该网络中的一个主机。

在常见的IP地址中，有些地址是可以被分配给主机的。具体来说，IPv4地址中通常使用子网掩码来划分网络地址和主机地址，其中，子网掩码的二进制表示中，前面一段连续的1表示网络地址，后面一段连续的0表示主机地址。因此，在给定一个IP地址和子网掩码的情况下，可以根据子网掩码的长度来确定网络地址和主机地址的范围。

例如，对于IP地址 192.168.1.100 和子网掩码 255.255.255.0，可以确定前面24位为网络地址，后面8位为主机地址，因此这个IP地址可以被分配给该网络中的一个主机。





在计算机网络中，网络层是OSI参考模型（OSI Model）和TCP/IP协议栈中的第三层。网络层负责将数据包从源主机传输到目标主机，并处理与路由、拥塞控制和流量控制等相关的任务。

在网络层中，有多个协议被用来实现数据传输和路由功能。其中，最常见的协议是Internet协议（IP，Internet Protocol），它是TCP/IP协议栈中的核心协议，也是网络层协议的代表。

因此，Internet协议（IP）是网络层协议中最重要的一个，也是最常用的一个。除了IP协议之外，在网络层还有一些其他的协议，如Internet控制报文协议（ICMP）、地址解析协议（ARP）等，它们也是实现网络层功能的重要协议。



17

## 哪个属于应用层的设备



应用层是OSI参考模型（OSI Model）和TCP/IP协议栈中的最高层，它负责为用户提供应用程序接口（API）和应用服务，以满足用户需求。因此，在应用层中，存在很多不同种类的设备 and 应用程序，用来为用户提供不同类型的应用服务。

下面是一些常见的应用层设备：

1. Web服务器：Web服务器是一种常见的应用层设备，它用于为Web浏览器提供Web页面、文件和其他内容。
2. 邮件服务器：邮件服务器是一种应用层设备，它用于管理电子邮件的发送和接收，包括邮件的存储、转发、过滤和分类等。
3. FTP服务器：FTP服务器是一种应用层设备，它用于支持文件传输协议（FTP），允许用户在网络中上传和下载文件。
4. DNS服务器：DNS服务器是一种应用层设备，它用于管理域名系统（DNS），将域名转换为对应的IP地址，以使用户能够访问特定的网站和服务。
5. 聊天服务器：聊天服务器是一种应用层设备，它用于支持实时通信和聊天应用程序，如即时通讯、语音聊天和视频聊天等。

这些设备和应用程序都是在应用层中工作的，用于向用户提供各种不同类型的服务和功能。



17

## UDP的概念





UDP (User Datagram Protocol) 是一种无连接的传输层协议，它在TCP/IP协议栈中位于IP协议之上，提供了一种简单的数据传输服务。UDP协议与TCP协议相比，具有传输速度快、开销小等优点，但不保证数据的可靠传输和数据包的顺序性。

UDP协议是一种面向无连接的协议，这意味着在数据传输之前，不需要进行握手或建立连接，而是直接将数据包发送到目标地址。UDP协议的数据包（也称为数据报）包含了源地址、目标地址、长度和校验和等信息，但没有TCP协议中的序列号、确认号和窗口等机制。

UDP协议适用于对数据传输速度和实时性要求较高、但对数据完整性要求不高的应用场景，例如音频、视频流媒体等。常见的使用UDP协议的应用程序包括DNS解析、NTP时间同步、实时通信、游戏等。

需要注意的是，由于UDP协议不保证数据的可靠传输和数据包的顺序性，因此在应用程序中需要自行处理数据丢失和乱序等问题。



## 17 IP协议数据包最大载荷，经典以太网帧的最大长度、最小长度



IP协议的数据包最大载荷是由链路层的最大传输单元（MTU，Maximum Transmission Unit）和IP协议的头部长度共同决定的。在以太网中，最常用的MTU大小为1500字节。因此，IP协议数据包的最大载荷为1500字节减去IP头部长度（通常为20字节）和可能存在的选项长度。

经典以太网（也称为Ethernet v2）的帧格式的最大长度为1518字节，其中包括14字节的以太网头部、4字节的CRC尾部以及最大的Payload载荷长度为1500字节。最小长度为64字节，其中包括14字节的以太网头部、4字节的CRC尾部以及46字节的填充。如果数据包的长度小于64字节，则需要通过填充来达到最小长度。

需要注意的是，由于不同类型的网络和协议都有不同的最大传输单元，因此数据包的最大载荷和帧的最大长度也会因此而不同。例如，在IPv6中，最大传输单元为1280字节。



## 17 IPv6地址长度



IPv6 (Internet Protocol version 6) 是IPv4的下一代互联网协议，它采用了128位的地址长度，相比IPv4的32位地址长度有了很大的提升。

IPv6地址的长度为128位，通常使用32个十六进制数表示，每个十六进制数用冒号分隔。例如，一个IPv6地址的表示形式为：

```
makefile
```

[Copy code](#)

```
2001:0db8:85a3:0000:0000:8a2e:0370:7334
```

IPv6地址中的每个十六进制数都表示16位二进制数，因此IPv6地址的总长度为128位，相比IPv4地址的32位长度大大增加了地址空间，可以支持更多的设备和连接。IPv6的地址空间约为2的128次方，相当于340,282,366,920,938,463,374,607,431,768,211,456个地址，因此IPv6协议可以为未来互联网的增长提供更充足的地址资源。



17

## 距离矢量路由算法，链路状态路由协议



距离矢量路由算法（Distance Vector Routing Protocol，简称DVR）和链路状态路由协议（Link State Routing Protocol，简称LSRP）都是常见的路由算法或协议。

距离矢量路由算法是一种基于距离的路由算法，每个路由器都维护一张路由表，用于记录到达目标网络的最短距离和下一跳路由器的信息。路由器之间定期交换路由信息，通过路由信息的更新来更新路由表，以达到动态调整路由的目的。常见的距离矢量路由协议有RIP（Routing Information Protocol）和IGRP（Interior Gateway Routing Protocol）。

链路状态路由协议是一种基于状态的路由协议，它与距离矢量路由算法不同的是，链路状态路由协议要求每个路由器都向其他路由器发送链路状态信息，包括连接的链路状态、链路带宽、拓扑结构、负载等信息，每个路由器通过收集其他路由器的链路状态信息来计算最短路径，以达到动态调整路由的目的。常见的链路状态路由协议有OSPF（Open Shortest Path First）和IS-IS（Intermediate System to Intermediate System）。

总的来说，距离矢量路由算法的计算简单、实现容易，但是可能存在路由环路和计数器溢出等问题；链路状态路由协议的计算复杂、实现相对困难，但是能够快速收敛，对网络的可靠性和稳定性更有保障。



17

## 网络子网划分及其地址数量





在TCP/IP网络中，子网划分是指将一个大的网络划分为多个更小的子网，每个子网可以独立地进行管理和控制，提高了网络的安全性和可扩展性。子网划分使用子网掩码（Subnet Mask）来确定网络中的哪些位用于网络地址，哪些位用于主机地址。

子网掩码通常使用32位的二进制数表示，其中网络位为1，主机位为0。例如，一个子网掩码为255.255.255.0的网络，其中前24位为网络位，后8位为主机位，可以表示256个主机。因此，该网络的地址数量为2的8次方减2，即254个。

更一般地说，对于子网掩码为N位的网络，可以表示2的（32-N）次方个子网，每个子网可以表示2的N次方减2个主机地址。例如，子网掩码为25位的网络可以表示2个子网，每个子网可以表示2的7次方减2个主机地址。

需要注意的是，子网划分需要合理规划，过多的子网划分会导致IP地址浪费，而过少的子网划分则会降低网络的灵活性和可扩展性。因此，在进行子网划分时需要综合考虑网络的规模、安全性、灵活性等因素，选择合适的子网掩码和划分方案。



17

## 静态多路复用技术



静态多路复用（Static Multiplexing）是一种基础的多路复用技术，它通过时间分割或频率分割的方式将多个信号复用到一个通信信道中进行传输，从而提高了通信信道的利用效率。

时间分割静态多路复用（Time Division Multiplexing, TDM）是一种常见的静态多路复用技术，它将信道划分为固定长度的时隙（Slot），每个时隙可以分配给不同的信号进行传输。TDM的实现需要使用一个时隙分配器，用于控制时隙的分配和传输。

频率分割静态多路复用（Frequency Division Multiplexing, FDM）是另一种常见的静态多路复用技术，它将信道划分为不同的频率带，每个频率带可以分配给不同的信号进行传输。FDM的实现需要使用一个频带分配器，用于控制频率带的分配和传输。

静态多路复用技术的优点是实现简单、可靠性高，不需要复杂的调度算法和协议，适用于数据量较小、带宽要求不高的通信场景。但是，由于静态多路复用技术无法动态地适应网络的变化，因此在动态、复杂的通信环境下可能存在性能瓶颈和资源浪费等问题。



17

“熟悉”的端口号，即预留给ftp，http等使用的范围





在TCP/IP协议中，端口号用于标识不同的应用程序或服务，范围从0到65535，其中0到1023的端口号被预留给系统或常用的应用程序使用，通常称为“熟悉的端口号”（Well-known Ports）。

以下是一些常见的“熟悉的端口号”及其对应的应用程序或服务：

- 20/21: FTP (File Transfer Protocol) 数据传输和控制端口
- 22: SSH (Secure Shell) 远程登录协议端口
- 23: Telnet (Teletype Network) 远程终端协议端口
- 25: SMTP (Simple Mail Transfer Protocol) 邮件传输协议端口
- 53: DNS (Domain Name System) 域名解析服务端口
- 80: HTTP (Hypertext Transfer Protocol) 网页浏览服务端口
- 110: POP3 (Post Office Protocol version 3) 邮局协议端口
- 119: NNTP (Network News Transfer Protocol) 新闻组传输协议端口
- 143: IMAP (Internet Message Access Protocol) 电子邮件访问协议端口
- 161/162: SNMP (Simple Network Management Protocol) 网络管理协议端口
- 443: HTTPS (Hypertext Transfer Protocol Secure) 安全网页浏览服务端口

这些端口号通常被操作系统或应用程序预先分配和使用，用户可以使用它们来访问不同的应用程序或服务。同时，用户也可以在需要的时候使用未被占用的端口号来为自己的应用程序或服务分配端口。



17

## CRC计算



CRC (Cyclic Redundancy Check) 是一种常用的数据校验方法，它通过对数据进行多项式计算，生成一个校验值并将其附加到数据末尾，以便在数据传输过程中检测数据是否损坏或遭到篡改。CRC可以检测到多种不同类型的错误，例如单比特错误、双比特错误和其他一些常见的传输错误。

以下是CRC计算的基本步骤：

1. 选择一个CRC生成多项式，通常用二进制表示，并将其转换为一个位串。
2. 将数据看作一个二进制数，将其与一个值为0的CRC寄存器相连。
3. 对于每一个数据位，执行以下操作：

将CRC寄存器左移一位，将下一位数据位移到CRC寄存器的最低位。

如果CRC寄存器的最高位为1，将CRC寄存器与生成多项式进行异或运算。

继续处理下一位数据位，重复上述步骤。

4. 处理完所有的数据位后，将最终CRC寄存器中的值附加到原数据末尾，形成一个新的数据帧。



5. 将数据帧发送给接收方，接收方执行与发送方相同的CRC计算过程，将生成的CRC校验值与接收到的校验值进行比较。如果两者一致，则数据传输成功，否则数据可能被损坏或遭到篡改。

需要注意的是，CRC计算方法可以有多种不同的实现方式和多项式选择，因此在实际应用中需要根据具体情况进行选择和调整。



17 给出一串比特，从高电平开始，自己画出曼彻斯特和差分曼彻斯特编码



假设给出的比特序列为：10101010

1. 曼彻斯特编码：

曼彻斯特编码将每个比特拆分成两个时间段，高电平表示0，低电平表示1。因此，对于给定的比特序列，曼彻斯特编码如下所示：

高低高低高低高低  
0101010101

2. 差分曼彻斯特编码：

差分曼彻斯特编码也将每个比特拆分成两个时间段，但是通过在每个比特的中间点进行翻转来表示比特值。如果比特值为0，则翻转电平；如果比特值为1，则不进行翻转。因此，对于给定的比特序列，差分曼彻斯特编码如下所示：

高低高低高低高低  
0101010101

以上是对比特序列的编码方式，实际上，在通信中，需要将编码后的比特序列转换为对应的电信号进行传输。



17 RED协议的工作原理是什么？



RED（Random Early Detection）是一种用于拥塞控制的网络协议，它可以通过在路由器或网络设备中实现来避免网络拥塞，并提高网络性能。

RED协议的工作原理如下：

1. RED监测队列长度：RED协议在网络设备中实现了一个队列，该队列用于存储要发送的数据包。当队列中的数据包数量超过一个特定的阈值时，RED开始监测队列长度。
2. RED随机丢包：当队列长度超过阈值时，RED开始随机丢弃一定数量的数据包。丢弃的数据包数量与队列长度的超出程度成正比，即队列超出的越多，丢包的数量也越多。
3. RED控制拥塞：当丢包发生时，发送方会认为数据包已经丢失并重新发送。这样，RED协议就可以通过控制丢包率来影响网络中的数据流量，避免网络拥塞并提高网络性能。

总体来说，RED协议通过在网络设备中实现一个简单的队列监测和随机丢包机制，从而控制网络拥塞和提高网络性能。与其他拥塞控制协议相比，RED的优势在于它可以避免网络突发性拥塞，同时对网络中的数据流量进行动态调整，适应网络负载的变化。



17

## 令牌桶和漏桶的工作原理？



令牌桶（Token Bucket）和漏桶（Leaky Bucket）都是流量控制算法，可以用于限制数据包或请求的发送速率。它们的工作原理略有不同：

### 1. 令牌桶(Token Bucket)算法

令牌桶算法会对发送数据包的速率进行限制，可以控制发送速率和发送突发性。它的工作原理是：在令牌桶中定期产生令牌，发送数据包时需要取走一个令牌才能发送。如果令牌桶中没有令牌，则数据包不能发送，需要等待令牌桶中产生令牌后再进行发送。因此，令牌桶算法可以用于控制数据包的发送速率。

### 2. 漏桶(Leaky Bucket)算法

漏桶算法则是对数据流的速率进行限制，可以控制接收速率和处理突发性。它的工作原理是：将所有发送的数据包放入一个漏桶中，漏桶以固定的速率向下游处理器传递数据包，当漏桶中有数据包时，以固定速率取出数据包进行处理。如果漏桶已满，新的数据包则会被丢弃，因此漏桶算法可以用于控制数据流的接收速率。

需要注意的是，令牌桶和漏桶算法一般都是在网络设备中实现的，用于控制流量和保护网络资源。这些算法也可以应用在各种不同的场景中，比如在应用程序中控制数据访问的速率。



17

## 什么是DNS？本地域名服务器、根域名服务器、顶级域名服务器？







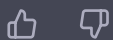
DNS是Domain Name System（域名系统）的缩写，是一个用于将域名转换为IP地址的分布式数据库系统。在互联网上，每台计算机都有一个唯一的IP地址，但是IP地址不便于人们记忆和输入，所以域名被引入到互联网中，用于替代IP地址进行访问。

本地域名服务器（Local DNS Server）是指在本地网络中的一个DNS服务器，用于缓存之前查询过的域名和对应的IP地址，加快本地访问速度，并减轻上级DNS服务器的负担。

根域名服务器（Root Name Server）是DNS层次结构中的最高级别的DNS服务器，它负责解析所有的顶级域名服务器（Top-Level Domain Name Server）的地址。互联网上总共有13台根域名服务器，它们分布在全球各地，负责管理互联网中的所有顶级域名。

顶级域名服务器（Top-Level Domain Name Server）是直接位于根域名服务器下一层的服务器，用于管理顶级域名，例如.com、.org、.net、.cn等。顶级域名服务器会将域名查询请求转发给对应的二级域名服务器（如.baidu.com的服务器就是.com顶级域名服务器），以便查找目标主机的IP地址。

当一个计算机需要访问一个域名时，它会先向本地域名服务器发出DNS查询请求，如果本地域名服务器没有缓存该域名的IP地址，它就会向根域名服务器发送查询请求，根域名服务器会返回对应的顶级域名服务器的IP地址，然后本地域名服务器再向顶级域名服务器发出查询请求，直到找到对应的IP地址，最终将查询结果返回给请求的计算机。



17

## 1、TCP协议

- 1) 在建立连接时，如何解决重复发来数据的问题
- 2) 在传送数据时，如何解决重复发来数据的问题



1) TCP协议在建立连接时采用三次握手（Three-way Handshake）的方式，可以解决重复发来数据的问题。具体来说，三次握手的过程如下：

客户端向服务器发送一个SYN（同步）报文，其中包含一个随机的序列号（sequence number） $x$ 。

服务器接收到SYN报文后，向客户端回复一个SYN-ACK（同步-确认）报文，其中包含一个随机的序列号 $y$ ，以及确认序列号 $ack=x+1$ 。此时服务器进入SYN\_RECEIVED状态。

客户端接收到SYN-ACK报文后，向服务器回复一个ACK（确认）报文，其中确认序列号 $ack=y+1$ 。此时客户端和服务器建立起了连接，并开始传输数据。客户端进入ESTABLISHED状态。

由于每个报文都包含一个随机的序列号，因此如果有重复的报文到达，接收方可以根据序列号判断该报文是否是重复的，并进行丢弃或忽略，从而避免重复发送数据的问题。

2) 在传送数据时，TCP协议采用序列号（Sequence Number）和确认应答

（Acknowledgement）来解决重复发来数据的问题。具体来说，每个TCP报文都包含一个序列号和一个确认序号，用于标识发送和接收的数据。发送方每发送一个报文，就会等待接收方回复一个确认应答，以确认该报文已被接收。如果发送方没有收到确认应答，就会重传该报文，直到接收方正确接收到该报文为止。

此外，TCP还采用了滑动窗口（Sliding Window）机制，用于控制发送方的发送速率。发送方在发送数据时，会按照接收方返回的窗口大小进行发送，如果接收方缓冲区已满，就会告诉发送方窗口大小为0，从而使发送方暂停发送，等待接收方处理数据后再继续发送。这样可以避免发送方发送过多数据导致接收方缓冲区溢出，从而保证数据传输的可靠性和稳定性。



17

## 2、介质访问控制

- 1) 以太网中采用的介质访问控制协议及其工作原理
- 2) 无线局域网中采用的介质访问控制协议及其工作原理
- 3) 二者在侦听信道是否空闲时的区别



1) 以太网中采用的介质访问控制协议是CSMA/CD（Carrier Sense Multiple Access with Collision Detection），其工作原理是在发送数据之前先检测信道是否被占用，如果空闲则发送数据，如果被占用则等待一段随机时间后再次检测。如果在发送过程中发现冲突（即多台设备同时发送数据导致的数据碰撞），则立即停止发送数据并发送一个JAM信号通知其他设备发生了冲突，然后等待一段随机时间后重新发送数据。

2) 无线局域网中采用的介质访问控制协议是CSMA/CA（Carrier Sense Multiple Access with Collision Avoidance），其工作原理是发送数据之前先发送一个RTS（Request to Send）信号请求发送数据，接收方收到RTS信号后回复一个CTS（Clear to Send）信号确认可以发送数据，发送方收到CTS信号后开始发送数据。如果在发送过程中发生冲突，则等待一段时间后重新发送RTS信号。

3) 以太网的CSMA/CD在侦听信道是否空闲时是通过检测信号的电平来判断的，即在发送数据之前先检测信道的电平是否为低电平。而无线局域网的CSMA/CA则是通过侦听一段时间来判断信道是否空闲的，因为无线信号可能会被干扰或衰减，所以需要先侦听一段时间来确认信道是否空闲。此外，无线局域网还存在信号覆盖范围和信道访问优先级等因素，需要更复杂的协议来解决。



17

3、主机A与主机B通过中间路由器相连，问：在主机A给主机B发送消息的过程中，使用了哪些协议和过程？





在主机A给主机B发送消息的过程中，涉及以下协议和过程：

1. 应用层协议：主机A使用特定的应用层协议（如HTTP、FTP、SMTP等）来封装消息。
2. 传输层协议：主机A使用TCP或UDP协议，将应用层协议封装的消息分段，并加上源端口号和目的端口号。
3. 网络层协议：主机A使用IP协议，将传输层协议封装的消息加上源IP地址和目的IP地址，并通过本地网络发送给路由器。
4. 路由选择协议：路由器会根据路由表选择一个合适的路径将数据包转发到主机B所在的网络。
5. 网络层协议：路由器使用IP协议，将数据包从主机A发送到主机B所在的网络，同时修改TTL（Time To Live）值以避免数据包在网络中无限循环。
6. 传输层协议：当数据包到达主机B所在的网络时，路由器会将数据包解封装并交给传输层协议。
7. 应用层协议：主机B的应用程序从传输层协议中接收数据，然后根据应用层协议进行处理，最终将数据呈现给用户。

需要注意的是，实际通信中可能会涉及更多的协议和过程，这里只是简要介绍了主要的几个协议和过程。



17

两个自治系统（把它当成一个AS就行），地址聚合，画出路由表  
给出一个IP地址，问应该到哪个端口，即最长匹配前缀  
AS1和AS2之前用到什么路由协议，什么协议分组保证其过程中可靠传输。



假设有两个自治系统（AS1和AS2），它们之间通过边界路由器相连。为了使路由表尽量精简，可以进行地址聚合。假设AS1的IP地址范围是：192.168.0.0/16和192.169.0.0/16，AS2的IP地址范围是：192.170.0.0/16和192.171.0.0/16。那么可以进行地址聚合，将AS1的地址范围聚合为192.168.0.0/14，AS2的地址范围聚合为192.170.0.0/14。

假设现在有一个IP地址为192.168.1.1，那么可以通过最长匹配前缀匹配到192.168.0.0/14，即这个地址应该到达AS1。

在主机A给主机B发送消息的过程中，涉及到的协议和过程如下：

1. 应用层：主机A的应用程序将数据封装成应用层协议的数据包，如HTTP、FTP、SMTP等。
2. 传输层：主机A的传输层使用TCP或UDP协议将数据包封装成传输层协议的数据段。
3. 网络层：主机A的网络层使用IP协议将数据段封装成网络层协议的数据包，并添加源IP地址和目的IP地址。
4. 数据链路层：主机A的数据链路层使用MAC地址将数据包封装成数据帧。
5. 物理层：主机A的物理层将数据帧转换为比特流，通过物理介质发送到中间路由器。

中间路由器接收到数据帧后，将数据帧从物理层一直向上逐层解封装，然后根据目的IP地址查找路由表，确定数据包应该转发到AS1还是AS2。如果数据包要经过多个路由器才能到达目的主机B，那么每个路由器都会重复上述过程，将数据包逐层解封装，并根据目的IP地址查找路由表，转发到下一个路由器，直到到达目的主机B。

AS1和AS2之间使用的路由协议可以是BGP（Border Gateway Protocol）。BGP是一种自治系统之间的路由协议，用于交换路由信息，使得自治系统之间能够相互通信。在BGP协议中，AS1和AS2都要向对方发送路由信息，包括自己的IP地址范围和可以到达其他AS的路由信息。然后，两个自治系统之间协商，确定最佳的路由路径。BGP协议具有很好的灵活性和可扩展性，适合于大型的互联