# 计网课堂笔记（二）

## 第四章（网络层）

网络层提供的服务

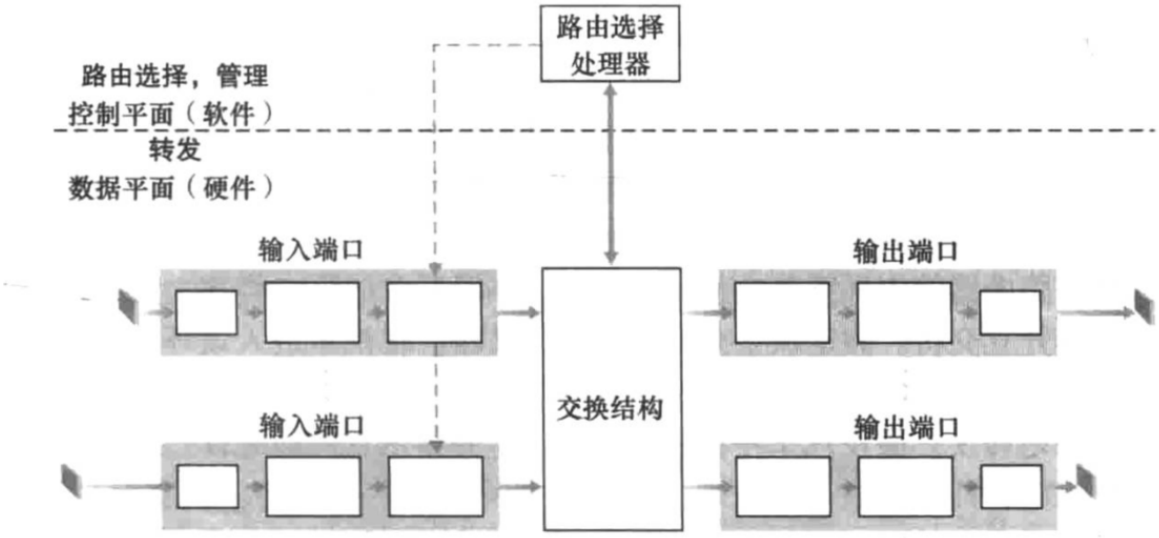
数据报网络 互联网，无连接，不可靠的服务，下面说的都是数据报网络

虚电路网络 ATM，面向连接的服务

网络层分解成数据平面（如何转发），控制平面（如何路由）

网络层提供的是两台主机之间的服务，运输层是两个进程之间的

### 路由器




路由：运行路由算法/协议，控制平面（软件）

- 路由处理器：维护转发表

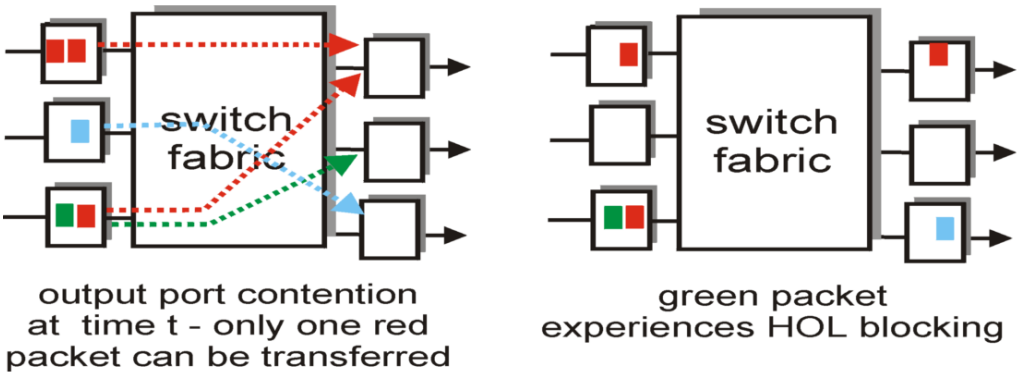
转发：将分组从输入链路转发到适当的输出链路，数据平面（硬件）

- 交换结构
  - 经内存：交换分组被拷贝到系统内存，速度受内存带宽限制 (每个分组需要两次经过总线)，早期的计算机多是在交换结构中通知选择处理器完成选路，后来多在输入端口中依靠输入线路卡完成
  - 经总线交换：经由总线传输到输出端口，被与标签匹配的端口保存，交换速率受到总线带宽的限制

    总线一次只能执行一次读写，所以前两种方法即使分组的目标端口不同也不能并行交换，
  - 经一个互联网络交换：输入和输出端口之前通过网络相连，只要目的输出端口不一样就可以**并行发送**，目的输入端口相同会导致线头阻塞，输入排队
- 输入端口：连接物理链路，接收分组（检查版本号，校验和），**查找路由表，选择输出端口**，转发

  排队：转发速率小于分组到达的速率（n个输入端口的总和），排队溢出缓冲区就会丢包

  线头（线路前部）阻塞：即使该分组要去的端口空闲，因为前面的分组还没完成交换也被迫等待，排在队列前面的分组阻止队列中其他的分组向前移动，交换网络中两个分组要发往同一个端口时不能同时发送就会出现HOL




- 输出端口：存储输入端口交换而来的分组，传输到链路上（就是输入端口反过来）

  排队：分组到达输出端口的速度大于它的处理速度

排队的分组调度

- 先进先出：只有这种分组是按到达的顺序离开的
- 优先权排队：有多个优先级不同的队列，队列内还是符合先进先出
- 循环：也有多个队列（类），在类之间不存在严格的服务优先权，循环调度器在这些类之间轮流提供服务
- 加权公平排队：比循环多了一个权重，按权重分配服务的量，如果权重都相等，那就是循环

# IP协议

## IPv4数据报结构

MTU：最大传输单元，链路层的帧中的数据部分的最大字节数，以太网中的一般为1500字节。
MTU=MSS+TCP首部长度+IP首部长度

MSS：TCP的报文段中的**数据部分**的最大字节数，MTU减去IP的Header和TCP的Header。IPv4的Header和TCP的Header一般都是20字节，则MSS=1500-20-20 = 1460字节（网络层的分片或分组的数据部分是包含了TCP头部的，所以len=MTU-20）

- 固定长度20bits，可以有选项，4bits表示首部长度，以4个字节为单位，所以是20到60字节
- 数据报长度字段显示是IP数据报的总长度（首部加上数据），以字节计，16bits，理论上最大65535个字节，但是以太网帧的载荷MTU有限制，很少超过1500字节
- TCP的校验是针对整个报文的，IP 的校验和是首部校验和
- 分片和重组（IPv4）

  链路层能承载的最大长度 MTU，在路由器分片，在主机重组

  ID/identifier 标识号，原分组和分片具有相同的标识号

  `fragflag=1` 说明后面还有其他分片

  offset 当前分片首字节的字节在分组中的序号/8

  原分组之前都有IP首部，分片后每个分片都有IP首部，内容长度最大=MTU-IP首部长度

  T分片与重组：

| | Len | ID=x | Flag=0 | Offset | ... | 数据Data |
|---|---|---|---|---|---|---|

- 数据长度DataLen=分组长度Len-20（IP首部无选项）
- 分片长不超过MTU，其中分片包含的数据长度为len=MTU-20
- 则分片数量n=DataLen/(MTU-20)，向上取整，则分片首部信息为：

| | Length=MTU | ID=x | Flag=1 | Offset=0 | ... | 数据Data |
|---|---|---|---|---|---|---|

| | Length=MTU | ID=x | Flag=1 | Offset=(MTU-20)/8 | ... | 数据Data |
|---|---|---|---|---|---|---|

| | Length=Len-(n-1)(MTU-20) | ID=x | Flag=0 | Offset=(n-1)*(MTU-20)/8 | ... | 数据Data |
|---|---|---|---|---|---|---|

- time to live，确保路由转发不超过n跳
- 上层协议字段

## IP编址

接口: 主机/路由器与物理链路之间的边界

每个路由器和主机都可以有多个接口，每个接口都会被分配一个32bits的IP地址

IP地址=子网部分+主机部分

书写方式：每个字节用它的十进制形式书写，各字节间以句点隔开

子网：没有路由器的介入，物理上能相互到达

子网部分使得子网内的接口的IP地址有了相同点

T 确定子网个数：分开主机和路由器的每个接口，得到的孤立的网络组的个数（注意有可能没画主机，只画了网线，网线也要算上）

子网掩码，子网部分高位bit是1，主机号对应0，就是遮掩一部分得到想要的部分，IP地址与子网掩码**按位与**得到子网IP地址

**网络划分**

A、B、C类地址

A0开头，B10开头，C110开头，子网部分位数8,16,24，越短，网越大

为了避免混淆，子网部分也要避免出现全0和全1，所以A类地址的最大子网数是$2^k$-2，其他类都是-1（但是避免全0和全1会造成较大的浪费，现在也有其他避免混淆的机制，所以根据题目来定，CIDR一般是不用考虑避免的）

最大主机数：$2^{(32-x)}$-2，全0表示子网地址，全1表示网广播

这样编址可能会造成浪费，C类不够，B类又只能使用一小部分

CIDR（无类别域间路由选择）地址，灵活分配子网部分的位数x，/x

获取单个地址：DHCP 动态主机配置协议（**使用UDP**），每个自网内都有DHCP服务器或者中继代理，主机可向DHCP服务器请求IP地址，即插即用，零配置。由于PC主机可能会经常更换子网，所以IP地址一般不能固定，临时请求的机制是很合理的

DHCP offer中提供推荐IP和租赁期限

# DHCP 概述:

- 主机广播 “DHCP discover” 消息
- DHCP 服务器用 “DHCP offer” 消息响应
- 主机请求IP地址: “DHCP request” 消息
- DHCP 服务器发送地址: “DHCP ack” 消息

获取一块地址：可以直接获取IP地址中的网络号部分，从ISP的地址空间中划分一块给申请者

ISP会被分配给一大块地址，这块地址还会对应相应的域名

T CIDR网络划分：

1. /m，划分成k个大小相同的子网，增加网络地址的位数，$2^{(32-m)}\div k=2^p$，增加p位网络，**主机的编号**全1是广播地址
2. 划分为大小不同的，**从大到小**分配（这样能保证分配的连续，否则最后可能最大的网络的地址分配会断开），需要k台主机，$2^{p-1}-2<k<=2^p-2$，主机编号为p位，网络编号32-p位，注意**编号要互斥**，每次记下来借用的几位编号是什么，之后不能重复，如果最后不够了就从有空的子网里面拆（划分之前可以先算一下够不够，需不需要插空）
3. 主机编号全0表示子网地址，写子网地址的时候把该字节的0都补上，然后换算成10进制，全0的字节不用写

**转发表与路由聚合**

路由器用转发表来存储所有可达前缀和指示下一步怎么转发，转发表项(x,I)，其中x是目的网络前缀, I是**该路由器**的接口之一的接口号，从I接口出去能到达x。

选择接口的原则是最长前缀匹配（最精确），这样如果一大块前缀网络中间空出了一块也没有关系，根据最长匹配原则能匹配到在别的地方的更小块的网络。

根据CIDR的编址方式，前缀能表示IP的范围，能起到路由聚合的作用

路由聚合：单个网络前缀通告多个网络的能力，由这个前缀的多个子网都会被聚集起来

CIDR的聚合性质又被转发表利用，使用前缀表示表示范围减少了转发表项，最长前缀匹配原则又能保证进行正确的寻址

T 制定转发表

找出固定的位，可0可1的位，去掉可0可1的位，看是否符合（更小的部分有没有被更长的前缀匹配）

## NAT网络地址转换

专用地址，只用于某个地域的内部通讯，比如家庭网络，不同的地域内可以重复使用同一个IP，互不干扰。但是和外部通信时由于IP地址不能重复，所以给这个局域网统一分配一个对外的IP地址，从专用地址到对外地址就需要通过网络地址转换NAT，在NAT路由器完成

内部地址和外部地址的转换，单向转换，只能内网转换到外网地址，外网不能直接发送到内网

NAT转换表，完成转换：LAN(源IP 地址, 端口号) <-> WAN(NAT IP 地址, 新端口号) ，转换成的NAT IP是统一的，端口号是唯一的，通过端口号来区分不同的主机，转换成原来的信息，16bit的端口号，一台NAT可以带$2^{16}$台主机

优点：1.解决地址短缺的问题，对外部网络来讲，本地网络只用分配一个IP地址

2.本地网络内部设备不能被外部世界明确寻址，更安全

### IPv6

报文首部

没有选项，定长40bits

地址长度 128bits，16Bytes

流量类型字段可体现优先级

不允许分片，没有首部校验和，减少每一跳的处理时间

可以有选项，但不是标准首部的一部分，而是用下一个首部字段指出

IPv4与IPv6的过渡：

双栈：直接转换头部，但是转换中不兼容的部分会丢失

隧道：保留原IPv6报文，再加一层头部IPv4来封装

# 第五章（控制平面）

控制方式：

- 每路由器控制，每台路由器独立运行路由算法，建立自己的转发表，控制平面和数据平面都在一个路由器内，是一个整体
- 逻辑集中式控制，集中式路由控制器运行路由算法计算并分发路由转发表，每台路由器中有一个控制代理，控制平面在路由器外，和数据平面是分开的

## 路由选择算法

发现开销最小的路径，集中式/分散式，静态/动态，负载敏感/负载迟钝

- 链路状态算法（LS全局）

通过结点状态广播知道整个网络的拓扑结构

Dijkstra算法和最短路径树

如果只用链路承担的流量来计算链路代价，可能会产生振荡，可以不用流量来计算代价（如RIP）或者确保并非所有路由器都同时运行LS算法

- 距离向量算法（DV分布式，迭代的）

Bellman-Fold方程：$d_x(y) = min_v[c(x,v) + d_v(y)]$

每个节点维护自己到所有其他节点的开销和邻居发过来的它们到其他是所有节点的开销

实际操作：每轮循环中，每个节点y向自己的邻居节点发送它的距离向量副本（只有自己这一行），就是$d_v(y)$，然后接收到该副本的节点x把这个邻居当成中转站，自己经过它看到达其他点的距离能不能减小（只比较自己这一行，别的行直接用邻居发过来的）

画表的时候，画自己的一行和**邻居**发过来的行，邻居的行会比自己的更新晚一轮，自己的行第一次更新只经过一跳，第n次交换后可以知道经过n跳的路径和开销，所以达到收敛次数等于不成环能经过的最大跳数（最远距离d-1），最后收敛的时候一定是对称矩阵，但是如果更新了开销的话最大收敛次数就不能这么算了

注意：更新的时候不能和表中原来的值比大小，是循环一遍通过所有的邻居节点到y和直接到y的距离，取最小值更新，和原来比的话坏消息无法传播

好消息收敛得快，坏消息收敛得慢（无穷计数问题），可能会有路由循环，即为到达x, y通过z路由，z又通过y路由，每次都在利用错误信息更新，直接的路径一直不被选中，收敛的很慢

毒性逆转，如果z通过y路由选择到目的地x，则z将通告y，它 （即z）到x的距离是无穷大，这样就可以避免y再利用本来就是从自己这里得出来的信息

毒性逆转中，如果经过了多个中间节点，但是根据算法节点只知道第一跳，所以即时有多个中间节点都是自己的邻居，也只会通告给第一跳的节点距离是正无穷，所以如果有两个以上直接相连的邻居节点，出现多个环路的时候，毒性逆转的方式不可行

算法健壮性：都可能会通告错误的链路开销，但是LS只会用通告来计算自己的转发表，因此有一定的健壮性，而DV算好自己的转发表之后又会把自己的表发给别人，不正确的值会在迭代中扩散

物理长度不能三角形两边之和不大于第三边，但是链路开销可以，DJ和距离向量算法都可以处理这种情况

## 域内选路

自治系统AS：一个区域内的路由器组成的集合，ASN标识

域内路由协议/内部网关协议：域内使用相同的路由协议，不同的AS内运行的路由协议可以不同

- RIP 基于距离向量，用跳数来衡量距离，设置最大跳数

  RIP的表是由应用级进程route-d管理的，表项（目的子网前缀，下一跳到达的路由器，到目的地的跳数），就是维护了自己到其他节点的距离

  RIP通告：以UDP（端口号520）报文发送，IP首部|UDP首部|通告信息（反向封装），每隔30s在邻居之间交换一次，如果180s内没收到通告就认为邻居死机或链路中断

  使用了毒性逆转，无限距离设置为16跳
- 开放最短路优先OSPF 基于DJ算法，各条链路开销是由网络管理员配置的
- OSPF通告：每个路由器把自己到邻居节点的链路情况通过洪泛法散步到整个自治系统，通告直接在网络层传播，没有传输层的报文首部，直接IP首部|通告信息

  当到达某目的地的多条路径具有相同的开销时，OSPF允许使用多条路径

  对单播与多播路由选择的综合支持

  支持单个AS中的层次结构，可以分为主干和区域边界，区域内执行OSPF算法，从而支持更大规模，具有可扩展性

## 域间选路

自治系统通过网关路由器相连

网关路由器：位于AS边缘，与其他AS中的路由器相连

每个AS都有唯一的标识号ASN，桩AS没有ASN，仅承载源或目的为本AS的流量

如何综合内部和域间的路由信息确定最好的路由

BGP（边界网关协议）

- 路由信息交换

  BGP会话：通过半永久TCP连接（端口179）来交换选路信息，内部iBGP/外部eBGP，iBGP连接并不总是与物理链路对应

  BGP路由包含的组件： 目的前缀；AS-PATH（到达目的前缀要经过的AS）； NEXT- HOP（AS-PATH起始的路由器（到达下一个AS要经过的网关路由器）接口的IP地址）； 其他属性

  BGP通告也可以有选择性的发送，对于ISP，任何穿越该ISP主干网的流量（通告的路由路径）必须是其源或目的位于该ISP的客户网络中

  对于接入多个ISP的提供商的客户，为了不把自己当做中继，可以不对任何外部路由器通告（除自身以外）任何其他目的地的路径，那么它将起到一个接入ISP的作用

  一级ISP之间一定是直接通信的，一级ISP不会把自己当做中继

  画出桩网络视角的网络拓扑，看它到别的桩网络要走过哪些路（根据它获得的信息选最短路），他就只知道哪些路

- 选路策略：

  热土豆策略：尽可能快地将分组送出其AS （更明确地说，用可能的最低开销，至于哪条路才是最低开销，就要用域内选路算法），而不担心其AS外部到目的地的余下部分的开销

  BGP路由选择：

  分优先级，本地偏好->最短AS-PATH的路由 ->最靠近 NEXT-HOP路由器的路由 : 热土豆路由->其他标准

注意区分路由相关的信息（BGP组件，可以就叫路由）和转发表项，转发表是给数据平面看的，转发就是从一个端口运送到下一个端口，转发表项(x,I)，其中x是目的网络前缀, I是**该路由器**的接口之一的接口号，从I接口出去能到达x。通过AS之间和之内的综合考虑确定路线，确定从哪个网关出去，现在走路由器的哪个接口，这个选路的结果会生成转发表项。

实验中静态配置的路由，目的前缀，NEXT- HOP，像是通告了一个BGP信息

iBGP, eBGP, OSPF, RIP通告的区别，iBGP和eBGP通告的都是AS外的路由器的前缀，iBGP是在AS之内传递的，eBGP在AS之间，OSPF, RIP通告的都是AS内的路由器的前缀，看域内使用的是那种算法，就进行发送通告。

区分iBGP和域内路由通告，都是在域内的路由器之间传递，但是iBGP通告的是域外的节点信息，域内路由通告通告的是域内节点的信息

# ICMP协议

用于主机路由器之间彼此交流网络层信息，差错报告和请求/应答（ping,traceroute)

它位于IP之上，ICMP报文是承载在IP分组中的，IP首部|ICMP报文

IP over everything

everything over IP 所有的协议和信息都可以封装成IP

traceroute的原理：源主机中的Traceroute向目的地主机发送一系列普通的IP数据报。这些数据报的每个携带了一个具有不可达UDP端口号的UDP报文段。第一个数据报的TTL为1,第二个的TTL为2,第三个的TTL为3,依次类推。TTL耗尽会得到返回的ICMP，包含源IP，到达目的主机也会返回ICMP，源主机就知道该停止发送了

ICMP所有报文的前4个字节都是一样的，包含类型、校验和等固定字段，但是剩下的其他字节则互不相同，总长度根据类型也不同

# 第六章（链路层）

功能：完成帧在两个相邻节点的传输

服务：

- 成帧
- 链路访问，MAC协议
- 在相邻节点之间提供**可靠**的服务（可靠传输机制与传输层类似）
- 差错检测和纠错
- 半双工：一个节点不能同时发送和接收/全双工：能同时
- 流量控制

链路层的主体部分是在网络适配器中实现的，也称为网卡

网络适配器分为链路层控制器和物理传输部分，控制器通常是一个实现了许多链路层服务（成帧、链路接入、差错检测等）的专用芯片。

网卡和总线相连接，是半自治的

## 差错检测

差错检测和纠正比特 (Error- Detection and- Correction EDC）来增强数据D

奇偶校验：二维，可纠正单个错误（包含校验字段），检测出两位的出错，同行或同列则不能纠错，否则可以

校验和：IP校验和只针对IP首部，TCP和UDP包括首部和数据

循环冗余校验：

所有CRC计算采用**模2算术**来做，在加法中不进位，在减法中不借位。这意味着加法和减法是相同的，而且这两种操作等价于操作数的**按位异或（XOR）**

注意模2运算的除法补位的时候把除数和被除数补成一样长度就可以了，被除数不一定要比除数大，如果余数不足r位，在左侧补0

接收方用G去除接收到的<D,R>的d+r位二进制数，如果余数非零则有错误发生

能检测少于r+1位的猝发错误和任意的奇数个比特错误，不能纠正

## 多址访问协议MAC

使用共享信道，要减少同时传输的冲突

分布式算法决定各节点如何共享信道，即决定节点什么时候可以传数据
共享信道既要负责进行数据传输，又要负责分布式算法的控制信息的传输

理想情况： 单个节点R；M个节点，M/R；完全分散（不需要主节点、时钟、时隙同步）；简单

- 信道划分协议

  时分多址（电话 ，传的是离散的信号但是无法被人耳分辨）、频分多址(调频广播)、码分多址

  缺点是信道利用率低，而且TDM一次发送结束后需要等待一轮才能再次发送，优点是完全无冲突

- 随机接入协议

  不划分信道，允许冲突能从冲突中“恢复”

  单个节点能占据整个信道

在不同的方法中效率的定义不同，可以理解成当前（有时隙的话就是当前时隙）传输成功（任意一个节点）的概率

- Aloha

  **纯Aloha：**

  每个节点在一次和下一次决定是否以p的概率传输之间至少间隔一帧的时间

  效率：$p(1-p)^{2(N-1)}$，它在发，它前后都没人发

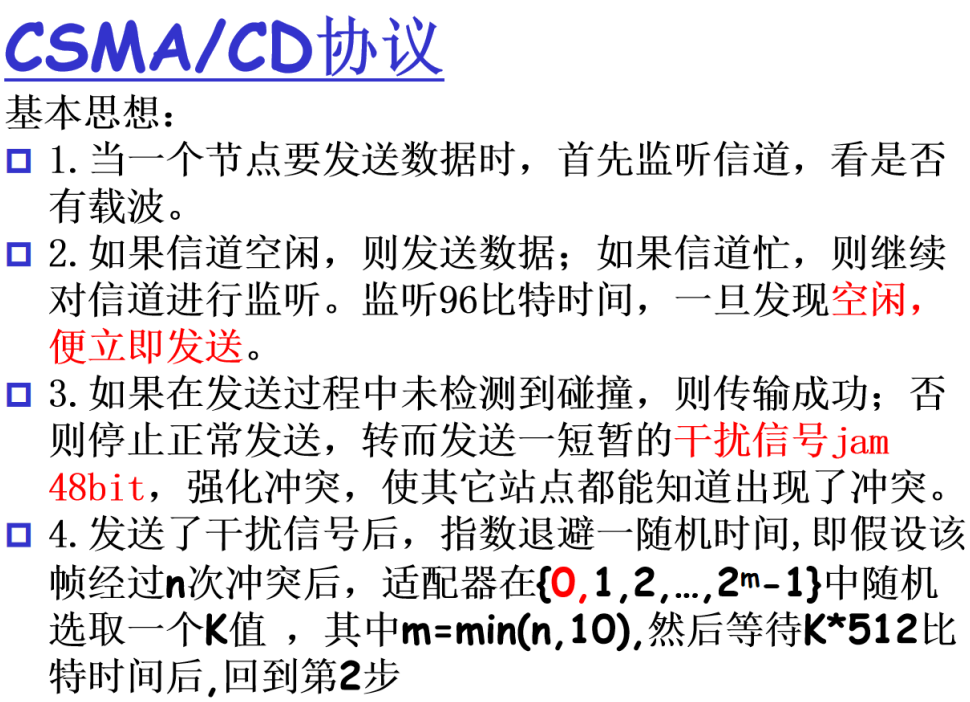
  **分时隙**：

  帧的大小相同，时隙等于一帧传送的时间，节点只能在一个时隙的开始才能传送

  如果有冲突，节点在随后的时隙以概率p重传该帧，直到成功为止

  效率（也叫平均吞吐量）：当有很多节点，每个节点有很多帧要发送时，成功时隙所占的百分比，N个节点：$Np(1-p)^{N-1}$

- 载波侦听多路访问 CSMA

  侦听可能不准，尽管节点B在t1时刻正在传输，但B传输的比特还没有到达D，因此D在t1侦听到信道空闲

  传播时延越大，冲突的概率越大，所以一般会设定一个传播时延的阈值

  有线网络中检测冲突容易，使用CSMA/CD（冲突侦听），无线网络一般是半双工，无法在发送的同时探测冲突，使用CSMA/CA（冲突避免）

  CSMA/CD 效率：当有大量的活跃节点，且每个节点有大量的帧要发送时，帧在信道中无碰撞地**传输**的那部分时间在长期运行时间中所占的份额

  T 冲突检测：

  p18、19

  求信道什么时候空闲，传输+传播都结束，最后一个bit到达

  注意48bit的jam信号

  注意结束退避后，不是立刻发送，而是还要侦听96比特时间，空闲才发送

  单位就用bit时间

  为了能够确保冲突被检测到，传播时间要小于最小帧的传输时间的一半

网络最远两台主机AB之间传播延迟为t，节点A在t0时刻发送数据后只需监听2t时间，若2t时间内没有检测到冲突，则无需再监听

- 最坏情况下节点A在t0+2t时间后检测到冲突（即A信号即将到达B之前B发送数据）
- 最短情况是节点A在t0时刻检测到冲突（即B在t0-t时刻之前已发出数据）
- A，B同时在t0时刻发出数据，则在t0+t时刻都将检测到冲突

- 轮转协议
  - 轮询，有主节点去依次询问每个节点是否有数据要发送，如果问完之后发现有数据要发送就会延迟很大
  - 令牌传递，和轮询类似，还是依次轮转，不过把主节点询问改成了传递令牌

时分多址和轮流的区别：TDMA的时间片是固定的，时长用完了即使还没传输完也只能等到下一次，没有要传的也要空耗时间，轮流的时间更自由，按需选择传输时长

# 交换局域网

48bits的MAC地址，固化在适配器（网卡）的ROM

每一层都有一种标识，运输层和网络层，套接字（IP+端口号），进程寻址，网络层，IP地址，链路层，MAC地址（不只用于互联网中）

广播MAC地址：全1

## ARP协议

判断属于哪个子网也要用最长匹配原则，网络地址也是看最长的，位数不一样，前几位相同也不一定就是在一个子网内，可能网络地址的位数不同

ARP表记录了在局域网内IP地址到MAC地址的映射，ARP只为在**同一个子网上**的主机和路由器接口解析IP地址

局域网内通信：直接用ARP查询MAC地址，广播IP查询，单播回复MAC（查询ARP报文是在广播帧中发送的，而响应ARP报文在一个标准帧中发送）

跨子网通信：需要先发给路由器与该子网相连的接口，这时需要使用一次ARP得到该接口的MAC，发送目的MAC为该接口MAC的帧，路由器收到帧后，可能还会再中转几个路由器（路由器根据IP前缀就可以完成转发，但是在链路上传输还是要有正确的源和目的MAC，所以这中间可能也有ARP），最后与目的子网相连的路由器再使用一次ARP得到目的IP的MAC地址，把源MAC换成自己发送端口的MAC和目的MAC就是终极目的MAC

两个主机发送IP数据报，中间经过n个路由器，假设所有的MAC地址都要用ARP查询，需要查n+1次

MAC只管一段链路（可经过交换机，不经过路由器），IP管全路，源IP和目的IP始终不变，每过一个路由器封装的源MAC和目的MAC会变（经过交换机不变）

## 以太网Ethernet

主流的 LAN技术，服务是无连接，不可靠的

帧结构：前同步码 8个字节，源和目的MAC，type：指明可以支持的高层协议，2字节，数据46~1500字节，CRC 4字节（位于帧的最后），所以最小长度的帧是26+46，最长26+1500

MAC协议：CSMA/CD，必须限制节点间的最大距离,以确保效率

**集线器**

物理层的中继器，信号的复制和放大，没有目的端口

没有物理隔离，会使冲突放大

不能连接不同编码方式的网络

**交换机**

功能：过滤和转发，全双工

转发根据交换机表，表项（MAC地址，通向该地址的接口，表项放置在表中的时间）

可见交换机是链路层设备，根据MAC转发，路由器包含网络层，根据IP转发

自学习得到交换表，哪个端口收到发送者发出的，源MAC可以作为别人的目的MAC

如果交换表中暂时没有目的MAC地址，那么就从所有的端口广播

可以隔离冲突域，交换机缓存帧并且决不会在网段上同时传输多于一个帧

集线器和交换机都可以不用配置，即插即用

# 总结

转换协议

| | | |
|---|---|---|
| CSDN | | |
| NAT | | |
| ARP | | |

标识

| | | |
|---|---|---|
| 应用层、运输层 | 端口号 | 1-1023预留，标识进程 |
| 网络层 | IP | IPv4：32bit，IPv6：128bit（16B） |
| 链路层 | MAC | 48bit |
| 物理层 | | |

服务

| | 连接 | 可靠 | 双工 | 流量控制 | 拥塞控制 | 差错检测 |
|---|---|---|---|---|---|---|
| TCP | 有 | 是 | 全双工 | 有 | 有 | 有，校验和 |
| UDP | 无 | 不 | 全双工 | 无 | 无 | 有，校验和 |
| IP数据报服务 | 无 | 不（最大跳数，很多条路） | 全双工 | 无 | 无 | 有，首部校验和 |
| 链路层 | 有 | 是 | 半或全 | 有 | 无 | 有CRC |
| Ethernet | 无 | 不 | | 无 | 无 | 有 |

报文结构

| | TCP | UDP | IP | MAC |
|---|---|---|---|---|
| 校验和 | | | | |
| 目的和源地址 | | | | |
| 长度字段 | 无 | 无 | 首部和数据的总长度 | 无 |
| 首部长度 | | | | |