# 第2章 物理层

1. 码元：一个固定时长的信号波形，表示一位k进制数字，代表不同离散数值的基本波形。比如0V电平代表000，1V电平代表001，2V电平代表010…7V电平代表111。这里一码元可以携带log28 = 3 bit的信息量。
2. 码元传输率：单位时间内传输的码元个数(脉冲个数或信号变化次数)，单位波特Baud
3. 信息传输率：单位时间内传输的二进制码元个数。如果一个码元携带n bit信息，码元传输率为c时，信息传输率为nc
4. 奈奎斯特定理(奈氏准则Nyquist)：理想低通的信道中(没噪声带且宽有限)，极限数据传输率为2Wlog2V，W是理想低通信道带宽，V是每个码元离散电平数目。
5. 香农定理：带宽受限且噪声干扰的信道的极限传输速率C = Wlog2(1+S/N)，W为信道带宽，信号传输平均功率与高斯噪声功率比。信噪比为10log10(S/N)
6. 非归零码NRZ：用两个电压表示二进制数，比如低电平为0，高电平为1。简单但无法检错，也无法判断一个码元的开始和结束，收发双方难以同步。
7. 非归零反相编码NRZ-I：电平翻转代表0，电平不变代表1。
8. 曼切斯特编码：将一个码元分为两个相等的间隔，前一个高电平后一个低电平表示1，相反则表示0。位中间的跳变即作时钟信号，又作数据信号。以太网就使用曼切斯特码
9. 差分曼切斯特码：若表示1，则前半个码元的电平和上一个码元的后半个电平相同。若表示0，则相反。
10. 数字数据调制为模拟信号，ASK, FSK, PSK, QAM。模拟信号编码为数字信号，比如音频信号的脉码调制(PCM)，三步骤：抽样，量化，编码。采样定理(也是奈奎斯特定理)指出采样频率必须大于等于音频的最大频率的两倍才能保证无损。
11. 电路交换，例如电话网络，两个结点先建立连接，资源独占。解决资源闲置的办法就是多路复用。报文交换，报文携带目标地址、源地址等控制信息，在结点上采用存储转发方式。分组交换是将完整的报文划分为标准大小的数据块，再加上控制信息进行传输。
12. 在网络层，分组交换还可以进一步分为面向连接的虚电路方式和无连接的数据报方式。虚电路就是在分组方式的基础上模拟电路连接。发送方和接受方建立一条逻辑上相连的虚电路，传输时路径固定。发送的每个数据分组还要带上虚电路号这个额外控制信息。

# 第3章 链路层

1. 交换机可以隔离冲突域，交换机的端口没有MAC地址。路由器可以隔离广播域。
2. 信道吞吐率 = 信道利用率 × 发送方发送速率，相当于一个周期内的平均发送速率
3. PPP协议面向字节。标志字段为7E(写成二进制就是01111110)。转义字节为7D，相当于在五个连续的1后填入一个0。
4. HDLC协议面向比特。用比特填充的首尾标记法实现透明传输。标记字段01111110，首尾0，中间六个1，如果比特流数据中有五个连续的1，就在其后填入一个0，接收时把这个0去掉。

# 第4章 网络层

## IPv4

1. 分组格式中，首部长度、总长度、片偏移单位分别为4B、1B、8B

标识，同一个数据报的每个分片标识相同。

标志，占3位，中间位DF(Don’t)=0允许分片。最低(右)位MF(More)=1后面还有分片。

首部检验和，只校验分组的首部。

片偏移的单位是8B，所以IP分片的长度必须是**8B的整数倍**。

协议，6表示TCP，17表示UDP

1. 总长度占16位，因此数据报最大长度为216-1=65535B。但是若大于下层链路层的最大传送单元MTU(以太网是1500B)，必须分片。IP分组在途径的路由器中都可以分片，但只有到达目的主机后才进行重组。
2. 传统的IP地址是分类地址，A, B, C类分别代表前8, 16, 24位是网络号，剩余的是主机号。另外还有D, E类。0.0.0.0在本网络上代表本主机，255.255.255.255代表全网广播，127.0.0.0代表环路自检。A类的网络号中0和127不可用，B类中128.0不可用，C类中192.0.0不可用。主机号全0表示本网络本身，全1表示本网络广播地址。那么排除全0全1，每个网络中主机数量为2k-2，k为主机号长度。
3. 子网划分：从主机号借用前几位作为子网号。结构变为{<网络号>, <子网号>, <主机号>}

做题时，子网号可以全0或全1，毕竟[RFC950]过时了。

1. 子网掩码 -> CIDR，相同前缀的CIDR地址块聚合称为路由聚合，查找路由表时遵循最长匹配原则，匹配的网络前缀越长，地址块就越小，越具体。
2. ARP协议，即地址解析协议，**工作在网络层**。由目标IP地址找到下一跳的MAC地址。其高速缓存中存有IP到MAC的映射表。若没有找到，则用FF-FF-FF-FF-FF-FF在链路层广播ARP请求分组。目标接收方单播回复ARP响应分组。
3. NAT协议是为了解决ipv4地址不够，辟出了几段IP地址作为私有地址，可在不同内网中重复使用。网关路由器为它们分配不同端口号，使它们能被外网区分。NAT路由器能看到端口，所以**工作在传输层**。NAT的表项由管理员添加。
4. 一般路由器转发分组不会修改源IP地址和目的IP地址，但使用NAT协议后，内网连接外网的路由器会修改源IP地址。
5. DHCP协议用于给主机动态分配IP地址。**是应用层协议**，采用C/S模式，基于UDP。四个阶段：新接入的客户机以源IP地址为0.0.0.0机向255.255.255.255广播“DHCP发现”消息。服务器收到后广播“DHCP提供”消息，其中包括提供给客户机的IP地址和配置信息。客户机如果接受该服务器提供的配置信息，则广播回复“DHCP请求”消息。最后服务器广播“DHCP确认”消息。
6. 注意，四次都是广播。但是，客户机还没确定下IP地址，所以是255.255.255.255的广播，不是主机号全1的广播。DHCP协议允许网络上配置多台DHCP服务器。客户机通常挑选最先回复的。也正是因为有多台DHCP服务器，所以才要两次握手才能让服务器们确认客户机选用的是哪一台服务器提供的IP和配置。
7. ICMP协议，网际控制报文协议， ICMP报文是打包成IP数据报发送出去的，所以**ICMP是IP层协议**。ICMP报文分为ICMP差错报告报文和ICMP询问报文。差错报告分为五类，不可达(RIP跳数大于15)，源点抑制(拥塞)，超时，参数问题(差错检验)，改变路由。询问报文分为四类，回送请求，时间戳请求，掩码地址请求，路由器询问和通告。Ping就使用了回送请求报文。
8. 第一个分片的后续分片，组播，特殊地址，ICMP差错报文再出错都不发ICMP差错报文。

## 路由协议

1. 自治系统是一个有权自主决定在本系统中采用何种路由协议的小型单位。通常一个自治系统属于一个行政单位，比如公司、政府部门。自治系统内的路由协议称为内部网关协议(RIP, OSPF)。自治系统间的路由协议称为外部网关协议(BGP)。
2. RIP协议，路由信息协议，基于分布式距离向量路由算法。每个路由器每隔30秒向相邻路由器广播自己的路由表(知道的全部信息)。RIP允许一条路径最多15跳。并用16跳表示不可达。超过180秒收不到相邻某个路由器的广播，则判定其不可达(距离为16)。
3. RIP协议实现简单，开销小。但是，最多15跳使RIP适用的网络规模较小。遇到网络故障时会发生慢收敛现象，俗称“坏消息传得慢”。
4. 慢收敛根本原因：其它邻居B,C和B,C的邻居D到A的路径都是通过自己O的。某时刻自己到某A的费用突然增大。万恶之源：在自己向B,C更新路由表前，B,C先向自己发了更新的路由表，或者自己刚向B,C更新完，但B,C还没向它们的邻居D更新，D又向B,C更新，更新的这条路径是已经过时的那条，由于选取最短的路径，正确的被错误的顶掉了。更新后的路由表中自己到A的路径是通过B,C中的一个，但其实这条路径是不存在的。然后自己向B,C发送新的不准确的路由表。在不断的交换中，不存在的费用小的路径的费用不断+1，直至和真实情况相符(收敛)。解决漏洞办法：毒性逆转技术，在上述例子中，B,C到A都是过O的，则B,C发给O的路由表中到A的距离改为16，即可化解。
5. **RIP协议是应用层协议**(争议)，使用UDP传送数据，端口520。哈工大慕课认为虽然RIP使用到应用层，但其本质是为了实现网络层功能。划分的依据应该是功能实现。
6. OSPF协议，开放最短路径协议，基于分布式链路状态路由算法。每个路由器仅当其链路状态发生变化时，才用泛洪法向全自治系统发送自己与相邻路由器的链路状态。使得每个路由器都可以知道全网的拓扑结构，用Dijkstra算法得出最优路径更新路由表。**OSPF是网络层协议**。报文直接走IP数据报发送，IP数据报首部协议字段为89，可以看作地位和TCP/UDP并列。
7. OSPF协议的路由器每隔10秒会和相邻路由器交换一次**问候分组**，判断邻站的可达性。新接入的路由器也会发送数据库描述分组和相邻路由器交换自己数据库中的摘要信息。OSPF路由器的链路状态只涉及相邻路由器的链接状态，所以不受网络规模扩大影响。也不存在“坏消息传得慢”。
8. BGP协议，边界网关协议，采用的是路径向量路由选择协议。每个自治系统选出至少1个路由器作为该自治系统的BGP发言人。各发言人相互建立TCP连接，交换报文建立会话，利用会话交换路由信息。所以**BGP是应用层协议**。BGP所交换的网络可达性信息有网络前缀、下一跳路由器、经过的自治系统序列。首次交换整个路由表，之后交换变化部分。

# 第5章 传输层

1. 传输层提供进程间的逻辑通信，即端到端。而网络层只提供主机之间的逻辑通信，点到点。这也是为什么UDP存在的意义，直接用IP无法做到端到端。
2. 应用进程通过端口号进行标识。端口号作为应用进程和传输实体进行交互的一种地址。端口号只具有本地意义。不同计算机上使用相同的端口号是没有联系的。而且同一台计算机中TCP和UDP的端口号也互不干扰，可以相同共存。
3. 端口号0-1023为熟知端口号分配给了最重要的应用。1024-49151为登记端口号。49152-65535为客户端使用的端口号。
4. 常用熟知端口号：FTP:21，TELNET:23，SMTP:25，DNS:53，TFTP:69，HTTP:80，SNMP:161

## 滑动窗口机制

1. 停等协议，Stop and Wait，发送方每发一个报文，就停下等，直到收到确认，再发一个，以此往复。
2. 后退N帧协议，Go Back to N，发送方可以连续发送多个窗口，但接收方只按序接收。一旦一个报文超时，则发送方重发这个报文之后的所有报文。GBN采用累计确认。此外，如果用n位给报文编号，那么发送窗口大小应满足1 <= WT <= 2n - 1
3. 选择重传协议，Selective Repeat，相比GBN进一步提高信道利用率，接收方不一定按序接收，失序的报文也缓存，同时对丢失的报文发一个NAK给发送方进行指定重传。SR不支持累计确认，大概是无法分辨是累计确认还是之前的报文都没收到。

WR + WT <= 2n 或者WT <= 2n-1

1. TCP是GBN和SR的混合体，取长补短。TCP和GBN一样采用累计确认，但是收到失序的报文段也会缓存起来。并发送冗余ACK来指明期望收到的报文段。发送方也只重传这一个报文，并不像GBN里面从丢失的开始重传之后所有的。冗余ACK可能就有SR中的NAK的作用。

## UDP

1. UDP只做传输层能做的最少工作。仅在IP之上加了两个服务，多路复用和差错检测。无需建立连接，无连接状态，分组首部开销小，上层可以更好地控制发送时间(无拥塞控制)。UDP是面向报文的，即一次发送完整的数据报，如果超长了，交给IP层去分片。适合一次传输数据量较小的应用。
2. 首部结构：16位源端口号，16位目的端口号，16位总长度，16位校验和。计算校验和时将伪首部(IP信息12字节，计算完丢弃不发送)、首部(校验和部分暂时为全零)、数据、填充部分(非偶数个字节填充0)以2字节为一组相加，最后再取反码。校验和字段不是必须的。或者说通常都不校验。当源主机不想用校验和时，把该字段设为全0。

## TCP

1. TCP在不可靠的IP层上实现可靠的数据传输协议，保证无差错、不丢失不重复、有序。TCP是面向连接的。一条TCP只能连接两个端点(一对一)。TCP提供全双工通信。TCP是面向字节流的，把应用层交下来的数据看成一连串无结构字节流。
2. TCP首部最短20字节，增加可选的选项后也需要满足是4字节的整数倍。

序号seq：占4字节，由于TCP是面向字节流的，他会给数据中每个字节都编上号。序号的表示本次发送的报文段的首个字节的编号。

确认号ack：期望收到的下个报文段首个字节的编号。而且之前序号的字节都正确接收。

数据偏移：即首部长度，以4字节为单位。

确认位ACK：ACK=0时确认号无效。在TCP连接建立后ACK始终为1

同步位SYN：SYN=1,ACK=0表示连接请求，接收方用SYN=1,ACK=1的报文回应

窗口：接收方用来指出当前允许发送方发送的数据量，用于流量控制。

检验和：计算方式和UDP完全一样，只是伪首部内容略有不同。

可选字段：最初只有MSS字段用来协商TCP报文数据部分最大长度。

1. 16进制数字，两个一组为1字节，4组相当于首部一行，5个4组相当于一个最小首部。判断以太网帧是否需要填充，看IP首部中总长度。判断某端收到了多少应用层数据，看两个TCP首部中序号之差。
2. TCP连接的建立采用C/S方式，主动发起连接的进程为客户机，被动等待连接的进程为服务器。连接的建立分3个步骤，称为3次握手：

C: SYN = 1, ACK = 0, seq = x, ack = 0 连接请求报文不带应用层数据

S: SYN = 1, ACK = 1, seq = y, ack = x+1 服务器同时为该连接分配缓存和变量

C: SYN = 0, ACK = 1, seq = x+1, ack = y+1 客户机同时为该连接分配缓存和变量

1. TCP连接的释放可以由两进程的任何一方发起，分为4次挥手：

A: FIN = 1, ACK = 1, seq = u 我要睡了，你还有啥话要嗦

B: FIN = 0, ACK = 1, seq = v, ack = u+1 把没说完的说完

B: FIN = 1, ACK = 1, seq = w, ack = u+1 说完了，我也去睡了

A: FIN = 0, ACK = 1, seq = u+1, ack = w+1 确认你要睡了，然后自己等2MSL后真的去睡

1. TCP连接的建立为什么不采用两次握手? 因为网络层及一下的传输是不可靠的。可能会发生客户端发送的连接请求被阻塞丢失，于是再发送了一次。连接成功完成传输并关闭连接后。先前被阻塞的连接请求送到了服务器。如果只有两次握手机制，服务器确认连接并分配了资源，但迟迟等不到回应，空费时间和资源。但如果是三次握手机制，服务器收不到第三个握手就自动取消为连接分配的资源。
2. 两种事件会导致TCP进行重传，超时和冗余ACK。TCP每发送一个报文段都设置一个计时器。这个计时器到期还没收到该报文段的确认就重传。
3. TCP规定每收到一个比期望的序号大的失序报文段时，就发送一个冗余ACK指明自己期待的字节序号。收到3个对同一序号的冗余ACK时，快速重传对方期待的报文段。
4. 发送窗口的上限为min{ rwnd, cwnd }。rwnd是接收方当前允许的数据量大小，放在TCP报文的首部窗口字段来告知发送方。cwnd是发送方根据当前网络拥塞状况设置的窗口值。
5. TCP的拥塞控制，第一步采用慢开始算法。开始时令cwnd = 1，每收到一个确认将cwnd + 1，cwnd接着是2, 4, 8…指数增长。当增长到一个设定的门限ssthresh时，改用拥塞避免算法。每经过一个往返时延RTT，cwnd + 1，呈线性增长。而当出现一次超时，则令门限ssthresh变为当前cwnd的一半(但不小于2)。cwnd重新设置为1执行慢开始算法。另外，靠近门限的最后一次指数增长不可大于门限。
6. 快重传即为用冗余ACK令发送方更早地重传。快恢复为收到3个冗余ACK时，门限减为当前cwnd一半(这条不变)，再把cwnd设为改变后的门限大小。跳过指数增长的慢开始，直接执行拥塞避免线性增长。
7. TCP采用快重传和快恢复的选项时，如果遇到超时现象，依然和传统方式一样cwnd回到1，从慢开始重传。

# 第6章 应用层