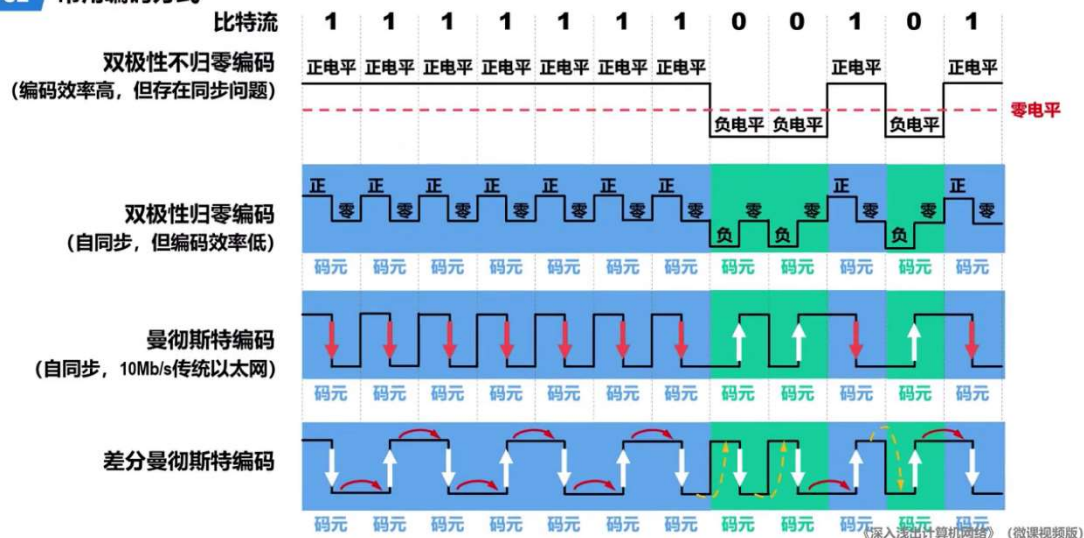
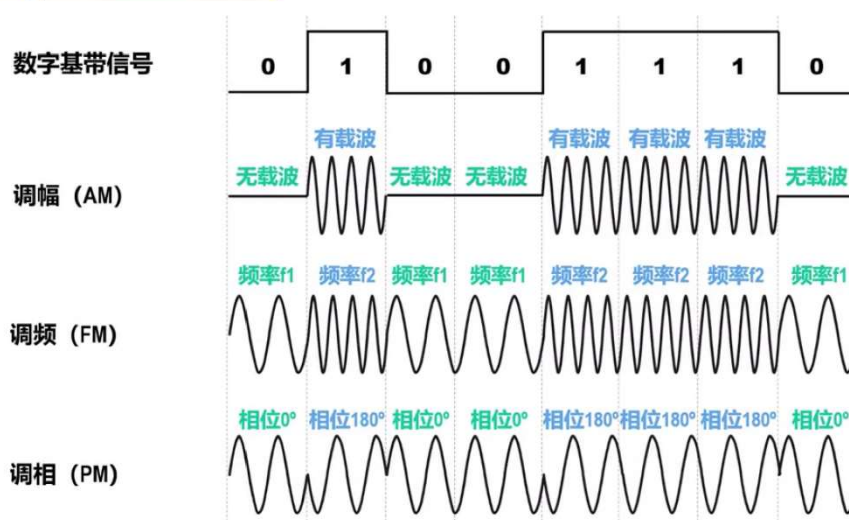


## 02 常用编码方式



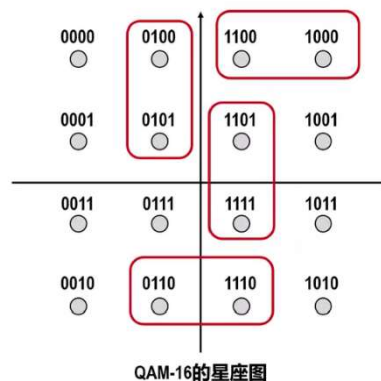
## 03 基本的带通调制方法和混合调制方法



## 03 基本的带通调制方法和混合调制方法

### 混合调制方法举例——正交振幅调制QAM-16

- 12种相位
- 每种相位有1或2种振幅可选
- 可以调制出16种码元(波形), 每种码元可以对应表示4个比特 ( $\log_2 16=4$ )
- 每个码元与4个比特的对应关系采用**格雷码**, 即任意两个相邻码元只有1个比特不同



奈氏准则	香农公式
<p><b>理想低通信道的最高码元传输速率 = <math>2W</math> Baud = <math>2W</math> 码元/秒</b></p> <p><b>W:</b> 信道的频率带宽 (单位为Hz)    <b>Baud:</b> 波特, 即码元/秒</p> <ul style="list-style-type: none"> <li>■ 使用奈氏准则给出的公式, 就可以根据信道的频率带宽, 计算出信道的最高码元传输速率。</li> <li>■ 只要码元传输速率不超过根据奈氏准则计算出的上限, 就可以避免码间串扰。</li> <li>■ 奈氏准则给出的是理想低通信道的最高码元传输速率, 它和实际信道有较大的差别。因此, 一个实际的信道所能传输的最高码元传输速率, 要明显低于奈氏准则给出的上限值。</li> <li>■ 码元传输速率又称为波特率、调制速率、波形速率或符号速率。</li> <li>■ 波特率与比特率有一定的关系: <ul style="list-style-type: none"> <li>□ 当1个码元只携带1比特的信息量时, 波特率 (码元/秒) 与比特率 (比特/秒) 在数值上是相等的。</li> <li>□ 当1个码元携带n比特的信息量时, 波特率 (码元/秒) 转换成比特率 (比特/秒) 时, 数值要乘以n。</li> </ul> </li> </ul>	<p><b>带宽受限且有高斯白噪声干扰的信道的极限信息传输速率</b></p> $C = W \log_2 \left( 1 + \frac{S}{N} \right) \quad (b/s)$ <p><b>C:</b> 信道的极限信息传输速率 (单位为b/s)</p> <p><b>W:</b> 信道的频率带宽 (单位为Hz)</p> <p><b>S:</b> 信道内所传信号的平均功率    <b>N:</b> 信道内的高斯噪声功率</p> <p><b>S/N:</b> 信噪比, 常用分贝 (dB) 表示</p> $\text{信噪比 (dB)} = 10 \log_{10} \left( \frac{S}{N} \right) \quad (dB)$ <ul style="list-style-type: none"> <li>■ 信道的频率带宽W或信道中的信噪比S/N越大, 信道的极限信息传输速率C就越高。</li> <li>■ 实际信道不可能无限制地提高频率带宽W或信道中的信噪比S/N。</li> <li>■ 实际信道中能够达到的信息传输速率, 要比香农公式给出的极限传输速率低不少。这是因为在实际信道中, 信号还要受到其他一些损伤, 例如各种脉冲干扰和信号衰减等, 这些因素在香农公式中并未考虑。</li> </ul>
<p>在信道的频率带宽W一定的情况下, 根据奈氏准则和香农公式, 要想提高信息的传输速率, 就必须采用多元制 (更复杂的调制技术), 并努力提高信道中的信噪比。</p> <p>自从香农公式发表后, 各种新的信号处理和调制方法就不断出现, 其目的都是为了使码元可以携带更多个比特, 进而可以尽可能地接近香农公式给出的传输速率极限。</p>	

【2017年 题34】若信道在无噪声情况下的极限数据传输速率不小于信噪比为30dB条件下的极限数据传输速率, 则信号的状态数至少是 ( D )。

A. 4                      B. 8                      C. 16                      D. 32

解析

设信号状态数 (可调制出的不同基本波形或码元数量) 为X

则每种码元可携带的比特数量为 $\log_2 X$

信道在无噪声情况下的极限数据传输速率 (用奈氏准则计算) =  $2W$  (码元/秒) =  $2W \log_2 X$  (比特/秒)

30dB信噪比条件下的极限数据传输速率 (用香农公式计算) =  $W \log_2 (1+1000)$  (比特/秒)

根据题意列出不等式  $2W \log_2 X \geq W \log_2 (1+1000)$  解得  $X \geq 32$

### 03 循环冗余校验

【CRC举例】接收到的信息为101101001, 生成多项式为  $G(X) = X^3 + X^2 + 1$ , 判断传输是否误码?

1	构造“被除数” 接收到的信息作为被除数
2	构造“除数” 生成多项式各项系数构成的比特串作为除数
3	做“二进制模2除法” 相当于对应位进行逻辑异或运算
4	检查“余数” 余数为0, 可认为传输过程无误码; 余数不为0, 可认为传输过程产生了误码。

$$\begin{array}{r}
 110010 \\
 1101 \overline{) 101101001} \\
 \oplus 1101 \phantom{00} \\
 \hline
 1100 \phantom{00} \\
 \oplus 1101 \phantom{00} \\
 \hline
 1100 \phantom{00} \\
 \oplus 1101 \phantom{00} \\
 \hline
 11
 \end{array}$$

余数不为0  
可认为传输过程产生了误码!

### 03 PPP帧的透明传输

面向比特的同步链路使用零比特填充来实现透明传输



发送方的处理:

对帧的数据载荷进行扫描（一般由硬件完成），每出现5个连续的比特1，则在其后填充一个比特0。

接收方的处理:

对帧的数据载荷进行扫描，每出现5个连续的比特1时，就把其后的一个比特0删除。

### 03 PPP帧的透明传输

面向字节的异步链路使用字节填充来实现透明传输[RFC1662]



发送方的处理:

- (1) 将数据载荷中出现的每一个0x7E减去0x20（相当于异或0x20），然后在其前面插入转义字符0x7D。
- (2) 若数据载荷中原来就含有0x7D，则把每一个0x7D减去0x20，然后在其前面插入转义字符0x7D。
- (3) 将数据载荷中出现的每一个ASCII码控制字符（即数值小于0x20的字符），加上0x20（相当于异或0x20，将其转换成非控制字符），然后在其前面插入转义字符0x7D。

接收方的处理:

进行与发送方相反的变换，就可以正确地恢复出未经过字节填充的原始数据载荷。

### 01 共享式以太网的争用期

- 使用CSMA/CD协议的共享总线以太网上的任意站点在发送帧的过程中都可能会遭遇碰撞。
- 站点从发送帧开始，最多经过时长 $2\tau$ （即 $\delta \rightarrow 0$ ）就可检测出所发送的帧是否遭遇了碰撞。
- 因此，共享总线以太网的端到端往返时间 $2\tau$ 被称为争用期（Contention Period）或碰撞窗口（Collision Window），它是一个非常重要的参数。
  - ☐ 站点从发送帧开始，经过争用期 $2\tau$ 这段时间还没有检测到碰撞，就可以肯定这次发送不会产生碰撞。
- 从争用期的概念可以看出，共享总线以太网上的每一个站点从发送帧开始，到之后的一小段时间内，都有可能遭遇碰撞，而这一小段时间的长短是不确定的，它取决于另一个发送帧的站点与本站点的距离，但不会超过总线的端到端往返传播时延，即一个争用期 $2\tau$ 。
  - ☐ 很显然，总线的长度越长（单程端到端传播时延越大），网络中站点数量越多，发生碰撞的概率就越大。
  - ☐ 因此，共享以太网的总线长度不能太长，接入的站点数量也不能太多。



## 01 共享式以太网的争用期

■ 10Mb/s共享总线以太网（传统以太网）规定：争用期 $2\tau$ 的值为512比特的发送时间，即 $51.2\mu s$ 。

$$\text{争用期 } 2\tau = \frac{512 \text{ b}}{10 \text{ Mb/s}} = \frac{512 \text{ b}}{10 \times 10^6 \text{ b/s}} = 51.2\mu s$$

除考虑了信号传播时延外，还考虑到网络中可能存在转发器所带来的时延以及产生碰撞时继续发送32比特或48比特人为干扰信号所持续的时间等。

$$\text{单程端到端传播时延 } \tau = \frac{51.2\mu s}{2} = 25.6\mu s$$

假设信号的传播速率为 $2 \times 10^8 \text{ m/s}$

$$\text{则总线长度为 } 2 \times 10^8 \text{ m/s} \times 25.6\mu s = 5120 \text{ m}$$

共享总线以太网规定：总线长度不能超过2500m。

以太网V2的MAC帧（最大长度1518B）				
目的地址	源地址	类型	数据载荷	FCS
6B	6B	2B	46B ~ 1500B	4B

满足最小帧长为64B的要求  
(6B + 6B + 2B + 46B + 4B = 64B)

## 01 IPv4地址的分类编址方法

32比特的IPv4地址	
网络号	主机号
<ul style="list-style-type: none"> <li>● 标志主机（或路由器）的接口所连接到的网络</li> <li>● 同一个网络中，不同主机（或路由器）的接口的IPv4地址的网络号必须相同，表示它们属于同一个网络。</li> </ul>	<ul style="list-style-type: none"> <li>● 标志主机（或路由器）的接口</li> <li>● 同一个网络中，不同主机（或路由器）的接口的IPv4地址的主机号必须各不相同，以便区分各主机（或路由器）的接口。</li> </ul>

## 01 IPv4地址的分类编址方法

■ A类、B类和C类地址都是单播地址，只有单播地址可以分配给网络中的主机（或路由器）的各接口。

■ 主机号为“全0”的地址是网络地址，不能分配给主机（或路由器）的各接口。

■ 主机号为“全1”的地址是广播地址，不能分配给主机（或路由器）的各接口。



网络类别	最小可指派网络号	最大可指派网络号	可指派网络数量	每个网络中最大可分配地址数量	不能指派的网络号	占总地址空间
A	1	126	126 ( $2^8 - 1 - 2$ )	16777214 ( $2^{24} - 2$ )	0和127	50% ( $2^{32-1} / 2^{32}$ )
B	128.0	191.255	16384 ( $2^{16-2}$ )	65534 ( $2^{16} - 2$ )	无	25% ( $2^{32-2} / 2^{32}$ )
C	192.0.0	223.255.255	2097152 ( $2^{24-3}$ )	254 ( $2^8 - 2$ )	无	12.5% ( $2^{32-3} / 2^{32}$ )

一般不使用的特殊IPv4地址

网络号	主机号	IP地址	作为源地址	作为目的地址	表示的意思
0	0	0.0.0.0	可以	不可以	在本网络上的本主机（例如，DHCP协议）
0	host-id	0.host-id	可以	不可以	在本网络上的某台主机host-id
全1	全1	255.255.255.255	不可以	可以	只在本网络上进行广播（各路由器均不转发）
net-id	全1	A类：net-id.255.255.255 B类：net-id.255.255 C类：net-id.255	不可以	可以	对网络net-id上的所有主机进行广播
127	非全0或全1的任何数	127.0.0.1~127.255.255.254	可以	可以	用于本地软件环回测试

## 01 IPv4地址的划分子网编址方法

随着更多的中小网络加入因特网，IPv4分类编址方法不够灵活、容易造成大量IPv4地址资源浪费的缺点就暴露出来了。

为新增网络申请新的网络号存在以下弊端：

- ☐ 需要等待时间和花费更多的费用   ☐ 会增加其他路由器中路由条目的数量   ☐ 浪费原有网络号中剩余的大量地址

子网掩码可以表明分类IPv4地址的主机号部分被借用了几个比特作为子网号。

与IPv4地址类似，子网掩码也是由32比特构成的。

- ☐ 用左起多个连续的比特1对应IPv4地址中的网络号和子网号；   ☐ 之后的多个连续的比特0对应IPv4地址中的主机号。

将划分子网的IPv4地址与相应的子网掩码进行逐比特的逻辑与运算，就可得到该IPv4地址所在子网的网络地址。

给定一个分类的IPv4地址和其相应的子网掩码，就可得出子网划分的细节：

- ☐ 划分出的子网数量   ☐ 每个子网可分配的地址数量  
☐ 每个子网的网络地址和广播地址   ☐ 每个子网可分配的最小地址和最大地址

默认子网掩码是指在未划分子网的情况下使用的子网掩码。

- ☐ A类：255.0.0.0   ☐ B类：255.255.0.0   ☐ C类：255.255.255.0

## 01 IPv4地址的无分类编址方法

为了简便起见，可以不明确给出配套的地址掩码的点十进制形式，而是在无分类编址的IPv4地址后面加上斜线“/”，在斜线之后写上网络前缀所占的比特数量（也就是地址掩码中左起连续比特1的数量），这种记法称为斜线记法。

【举例】

128.14.35.7 / 20

网络前缀：20比特  
主机号：12比特 (32-20)

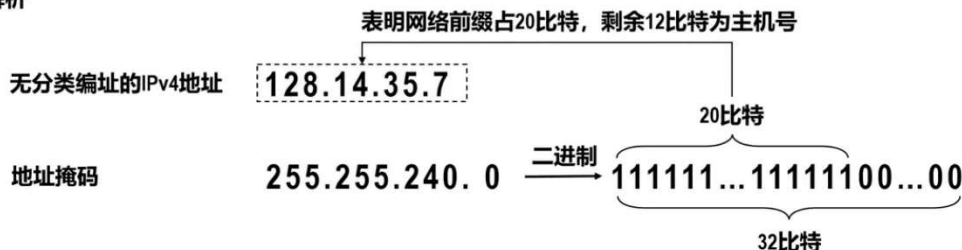
## 01 IPv4地址的无分类编址方法

■ 无分类编址方法使用的**地址掩码**与划分子网使用的**子网掩码**类似，由32比特构成。

- ☐ 用左起多个**连续的比特1**对应IPv4地址中的**网络前缀**；
- ☐ 之后的多个**连续的比特0**对应IPv4地址中的**主机号**。

【举例】假设给定的无分类编址的IPv4地址为128.14.35.7，配套给定的地址掩码为255.255.240.0，请指出该IPv4地址的网络前缀和主机号各自的长度。

解析



## 01 IPv4地址的无分类编址方法

■ 实际上，无分类域间路由选择CIDR是将网络前缀都相同的、连续的多个无分类IPv4地址，组成一个**CIDR地址块**，只要知道CIDR地址块中的任何一个地址，就可以知道该地址块的以下全部细节：

- ☐ 地址块中的**最小地址**
- ☐ 地址块中的**最大地址**
- ☐ 地址块中的**地址数量**
- ☐ 地址块中**聚合某类网络（A类、B类、C类）的数量**
- ☐ **地址掩码**

■ 使用CIDR的一个好处是，可以根据客户的需要**分配适当大小的CIDR地址块**，因此可以更加有效地分配IPv4的地址空间。

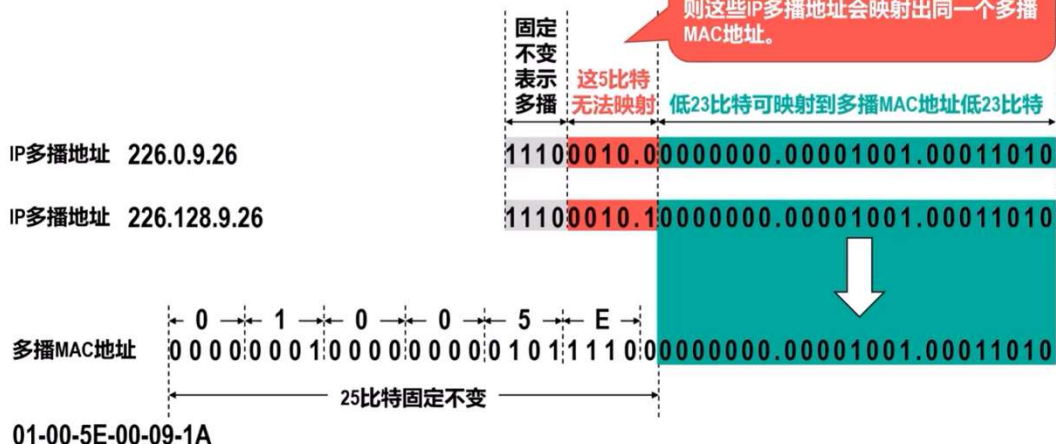
■ 使用CIDR的另一个好处是**路由聚合（也称为构造超网）**。

**网络前缀越长，地址块越小，路由越具体；**

■ 若路由器查表转发分组时发现有多条路由条目匹配，则选择网络前缀最长的那条路由条目，这称为**最长前缀匹配**，因为这样的路由更具体。

## 01 在局域网上进行硬件多播

【举例】IP多播地址与多播MAC地址的映射关系并不是唯一的。





## 网际组管理协议IGMP

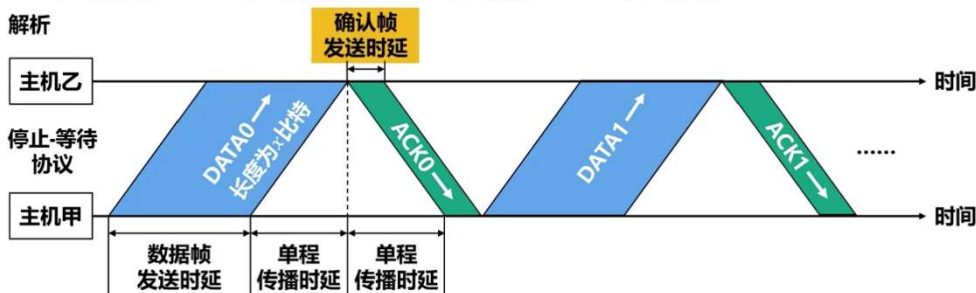
■ 网际组管理协议 (Internet Group Management Protocol, IGMP) 是TCP/IP体系结构网际层中的协议, 其作用是让连接在本地局域网上的多播路由器知道本局域网上是否有主机 (实际上是主机中的某个进程) 加入或退出了某个多播组。

■ IGMP仅在本网络有效, 使用IGMP并不能知道多播组所包含的成员数量, 也不能知道多播组的成员都分布在哪些网络中。

### 02 停止-等待协议的信道利用率

【2018年 题36】主机甲采用停-等协议向主机乙发送数据, 数据传输速率是3kbps, 单向传播延时是200ms, 忽略确认帧的传输延时。当信道利用率等于40%时, 数据帧的长度为 (D)。

A. 240比特 B. 400比特 C. 480比特 D. 800比特



$$\text{信道利用率} = \frac{\text{数据帧的发送时延}}{\text{数据帧的发送时延} + \text{单程传播时延} \times 2}$$

$$40\% = \frac{\frac{x \cdot b}{3k \cdot b/s}}{\frac{x \cdot b}{3k \cdot b/s} + 200ms \times 2}$$

解得  $x = 800b$

### 01 回退N帧协议

#### 发送方

- 发送窗口  $W_T$  的取值范围是  $1 < W_T \leq (2^n - 1)$ , 其中,  $n$  是构成分组序号的比特数量。
  - ☐ 如果  $W_T = 1$  变成了停止-等待协议
  - ☐ 如果  $W_T > (2^n - 1)$  接收方无法分辨新旧数据分组
- 可在未收到接收方确认分组的情况下, 将序号落入发送窗口内的多个数据分组全部发送出去。
- 只有收到对已发送数据分组的确认分组时, 发送窗口才能向前滑动到相应位置。
- 收到多个重复确认时, 可在重传计时器超时前尽早开始重传, 由具体实现决定。
- 发送窗口内某个已发送的数据分组产生超时重传时, 发送窗口内该数据分组的后续已发送的数据分组也必须全部重传, 这就是回退N帧 (Go-back-N, GBN) 协议名称的由来。

#### 接收方

- 接收窗口  $W_R = 1$  的, 因此只能按序接收数据分组。
- 只接收序号落入接收窗口内且无误码的数据分组, 并且将接收窗口向前滑动一个位置, 与此同时给发送方发送相应的确认分组。
- 为了减少开销, 接收方不必每收到一个按序到达且无误码的数据分组就给发送方发送一个相应的确认分组。
  - ☐ 可以在连续收到多个按序到达且无误码的数据分组后 (数量由具体实现决定), 才针对最后一个数据分组发送确认分组, 这称为累积确认。
  - ☐ 或者可以在自己已有数据分组要发送时才对之前按序接收且无误码的数据分组进行捎带确认。
- 接收方收到未按序到达的数据分组后, 除丢弃外, 还可对之前最后一个按序到达的数据分组进行重复确认, 以便发送方尽快重传。

回退N帧协议在流水线传输的基础上, 利用发送窗口来限制发送方连续发送数据分组的数量, 是一种连续ARQ协议。在回退N帧协议的工作过程中, 发送窗口和接收窗口不断向前滑动, 因此这类协议又称为滑动窗口协议。在信道质量较差 (容易出现误码) 的情况下, 回退N帧协议的信道利用率并不比停止-等待协议的信道利用率高。

## 01 选择重传协议

从滑动窗口的角度对比停止-等待协议、回退N帧协议和选择重传协议

停止-等待协议	回退N帧协议	选择重传协议
仅用1个比特给分组编号	用 $n(n > 1)$ 个比特给分组编号	用 $n(n > 1)$ 个比特给分组编号
$W_T = 1$	$1 < W_T \leq (2^n - 1)$ <small><math>W_T</math>超过上限 无法分辨 新旧数据分组</small>	$W_R$ 超过 $W_T$ 没有意义 $1 < W_R \leq W_T$ $W_T + W_R \leq 2^n$ $1 < W_R \leq 2^{(n-1)}$
$W_R = 1$	$W_R = 1$	确保接收窗口向前滑动后, 落入接收窗口内的新序号与之前的旧序号没有重叠, 避免无法分辨新旧数据分组。  当 $W_R$ 取最大值 $2^{(n-1)}$ 时, $W_T$ 能取到的最大值也为 $2^{(n-1)}$ 。

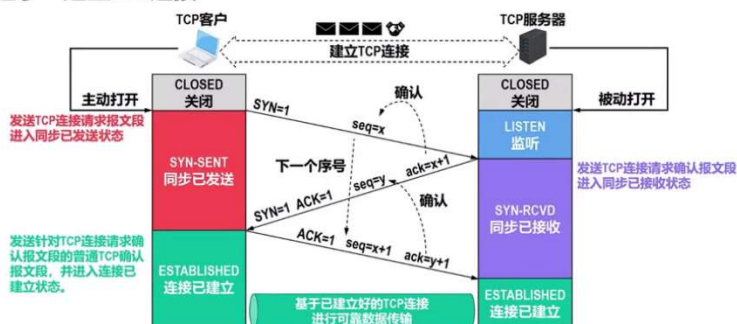
## 01 选择重传协议

用 $n(n > 1)$ 个比特给分组编号, 发送窗口 $W_T$ 与接收窗口 $W_R$ 的关系如下:

$W_R$ 超过 $W_T$ 没有意义	$1 < W_R \leq W_T$	$1 < W_R \leq 2^{(n-1)}$
确保接收窗口向前滑动后, 落入接收窗口内的新序号与之前的旧序号没有重叠, 避免无法分辨新旧数据分组。	$W_T + W_R \leq 2^n$	当 $W_R$ 取最大值 $2^{(n-1)}$ 时, $W_T$ 能取到的最大值也为 $2^{(n-1)}$ 。

发送方	接收方
<ul style="list-style-type: none"> <li>可在未收到接收方确认分组的情况下, 将序号落入发送窗口内的多个数据分组全部发送出去。</li> <li>只有按序收到对已发送数据分组的确认分组时, 发送窗口才能向前滑动到相应位置。</li> <li>如果收到未按序到达的确认分组, 应对其进行记录, 以防止其相应数据分组的超时重发, 但发送窗口不能向前滑动。</li> </ul>	<ul style="list-style-type: none"> <li>可接收未按序到达但没有误码并且序号落入接收窗口内的数据分组。</li> <li>为了使发送方仅重传出现差错的分组, 接收方不再采用累积确认, 而需要对每一个正确接收到的数据分组进行逐一确认。</li> <li>只有在按序接收数据分组后, 接收窗口才能向前滑动到相应位置。</li> </ul>

## 01 “三报文握手”建立TCP连接



请同学们注意:

- TCP规定SYN被设置为1的报文段 (例如TCP连接请求报文段和TCP连接请求确认报文段) 不能携带数据, 但要消耗掉一个序号。
  - 按上述规定, TCP连接请求报文段不能携带数据 (即没有数据载荷), 但是会消耗掉序号x。
  - 因此, TCP客户进程下一次发送的TCP报文段的数据载荷的第一个字节的序号为x+1。
- TCP规定普通的TCP确认报文段可以携带数据, 但如果不携带数据, 则不消耗序号。
  - 如果该报文段不携带数据, 则TCP客户进程要发送的下一个数据报文段的序号仍为x+1。



## 01 “四报文挥手” 释放TCP连接

