

# 计算机网络



计算机与信息学院  
人工智能学院

## 第三章 链路层

- ✓ 链路层功能
- ✓ 差错控制
- ✓ 流量控制与可靠传输
- ✓ 链路层协议

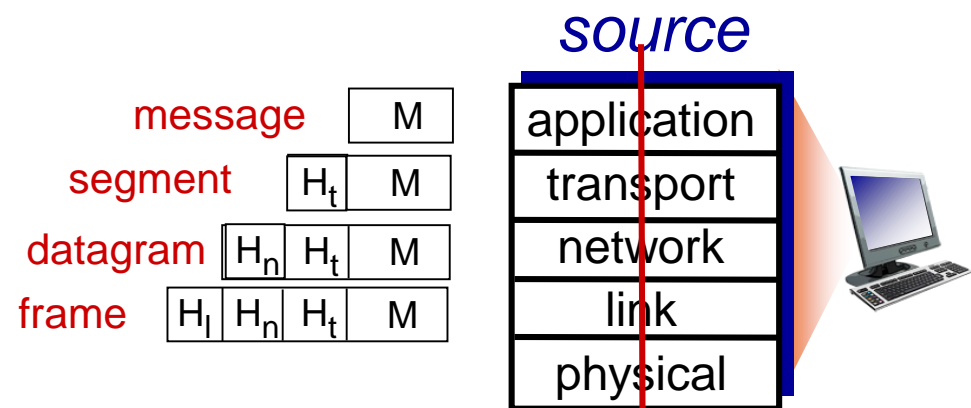
# 链路层的功能

- 术语

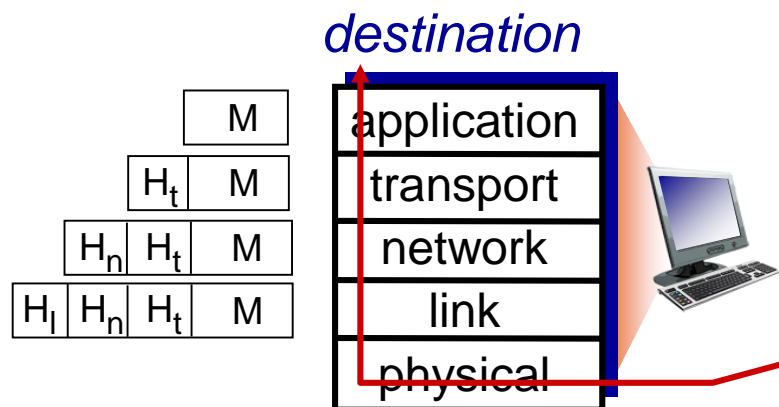
- 主机、路由器等：节点 (nodes)
- 连接两个相邻节点的物理通路：链路 (Link)

点到点链路：两个节点独占链路

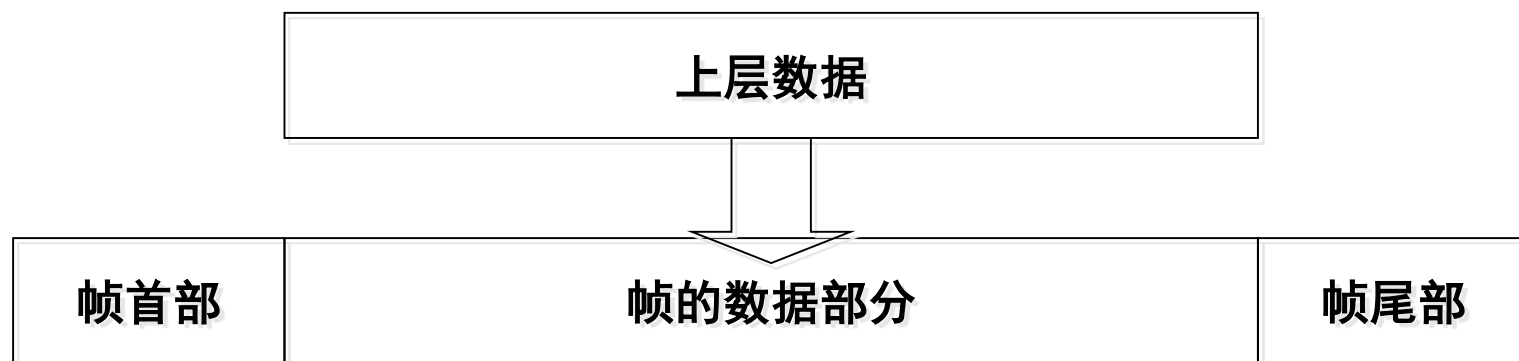
广播链路：多个节点共享链路



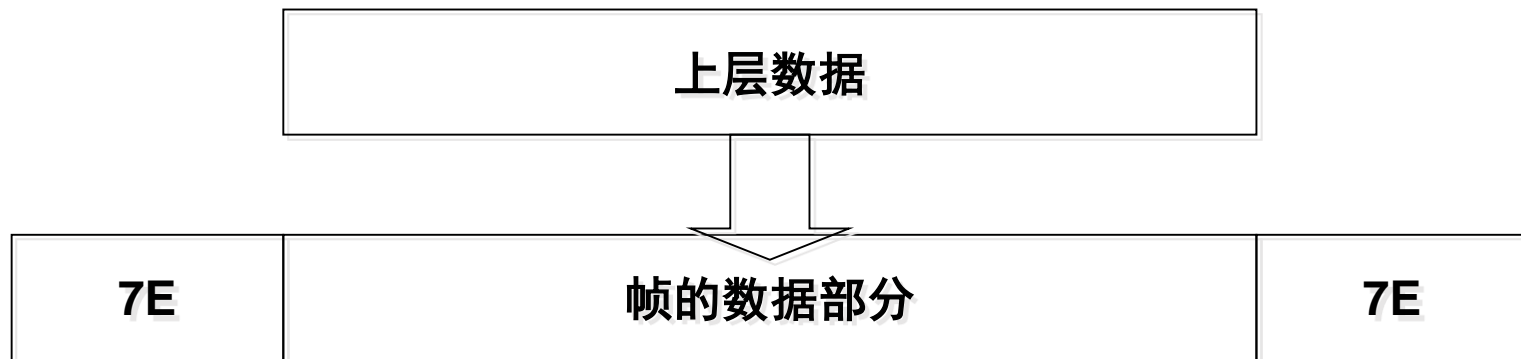
链路层PDU: 帧  
物理层PDU: bit



## 1. 组帧：封装成帧，标识帧的开始和结束

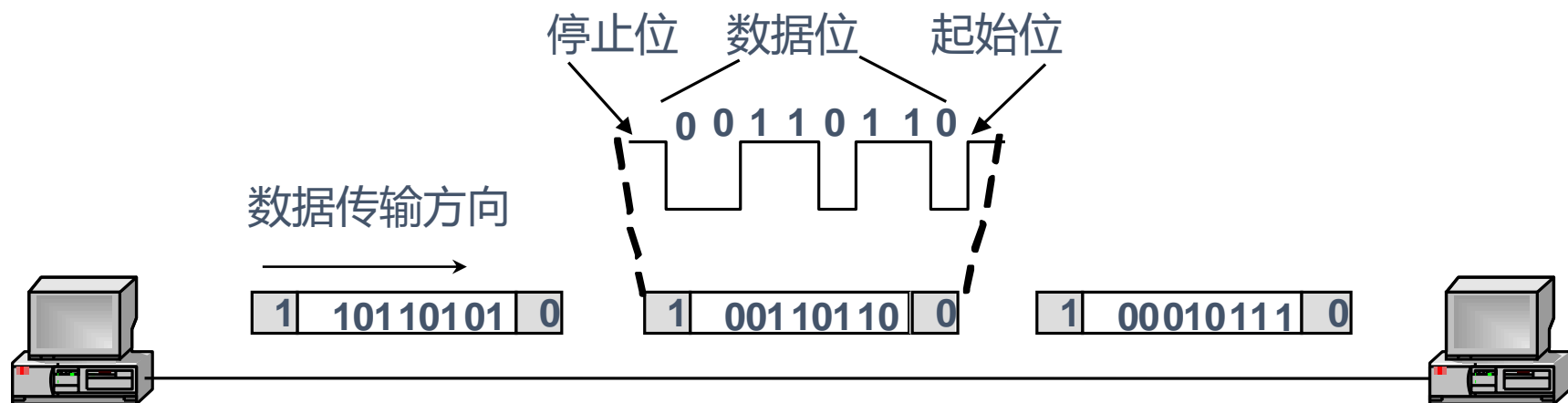


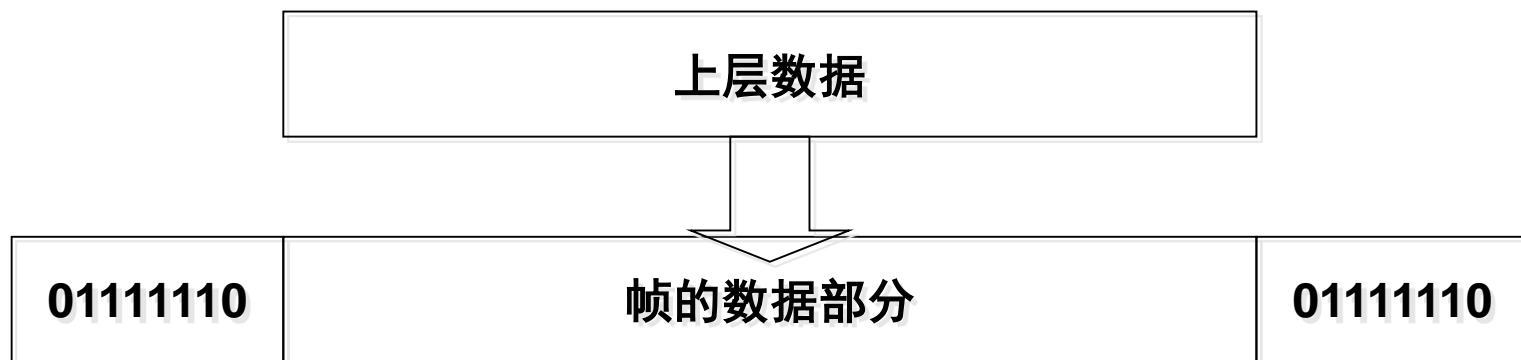
- ✓ 字符填充：添加特殊的控制字符
- ✓ 比特填充：添加特殊的bit



✓ 字符填充：添加特殊的控制字符

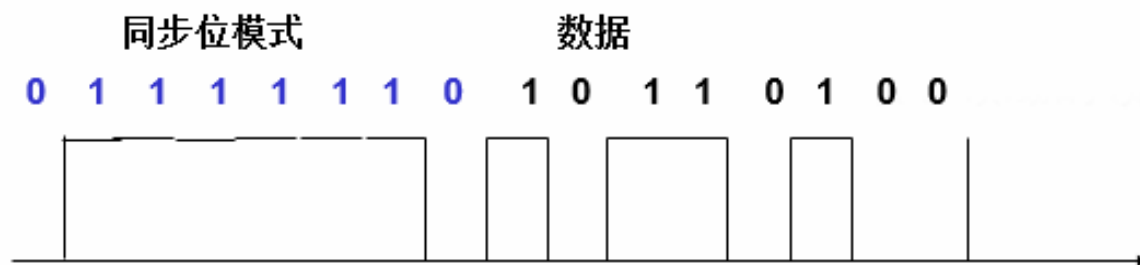
异步传输：以字符为单位传输，字符同步，传输效率低





✓ 比特填充：添加特殊bit流

同步传输：以数据块为传输单位，针对数据块内的每位进行同步——位同步



## 异步传输：7E

数据部分：7E  $\rightarrow$  (0x7D, 0x5E)

7D  $\rightarrow$  (0x7D, 0x5D)

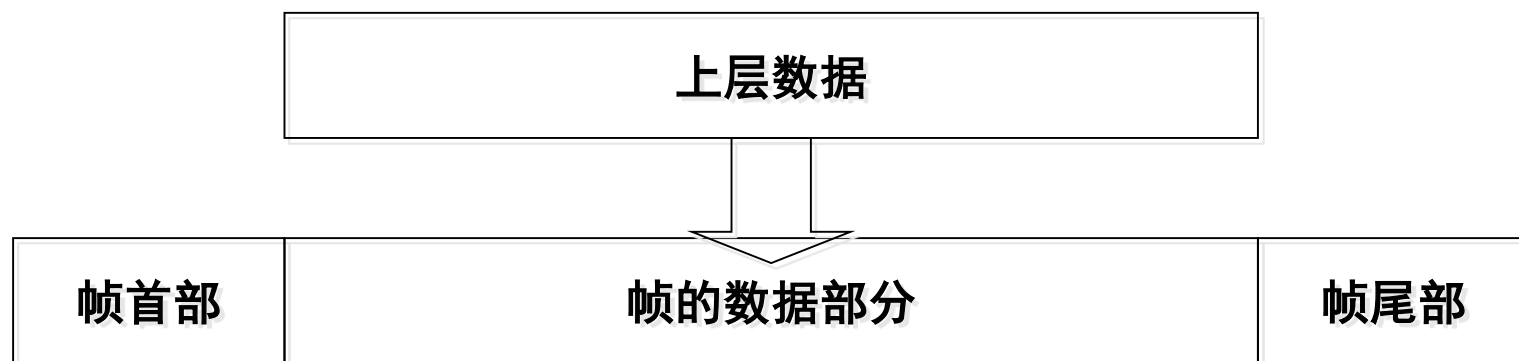
12 7E 7D 34



**同步传输：** 01111110

数据部分出现连续的5个比特 1 时，在其后插入一个比特 0

## 2. 差错检测：处理传输中出现的差错



- ✓ 随机错：单个或少量比特的独立错误
- ✓ 突发错：连续多个比特发生错误

✓ 差错控制编码：在传输的数据（信息位）后增加冗余位

- 奇偶校验码
- **循环冗余校验码（CRC: Cyclic Redundancy Check）**
- 海明码
- **Internet校验和**

# CRC

多项式码，将二进制数据与多项式对应起来，通过多项式的“模2运算”进行差错检测

1011011:  $X^6+X^4+X^3+X+1$

# CRC

- ① 发送方：发送数据 $K$  ( $k$ bit)
- ② 收发双方协商：生成多项式 $G(x)$ ， $r$ 次多项式 ( $r+1$  bit)
- ③ 发送方：
  - 在数据 $K$ 后面添加  $r$  个0 ( $x^r.K(x)$ )
  - 除以 $G(x)$ ，得到的余数  $R(x)$  ( $r$  bit)

**K** 要发送的数据 (**k**位)

**R** 冗余码 (**r**位)

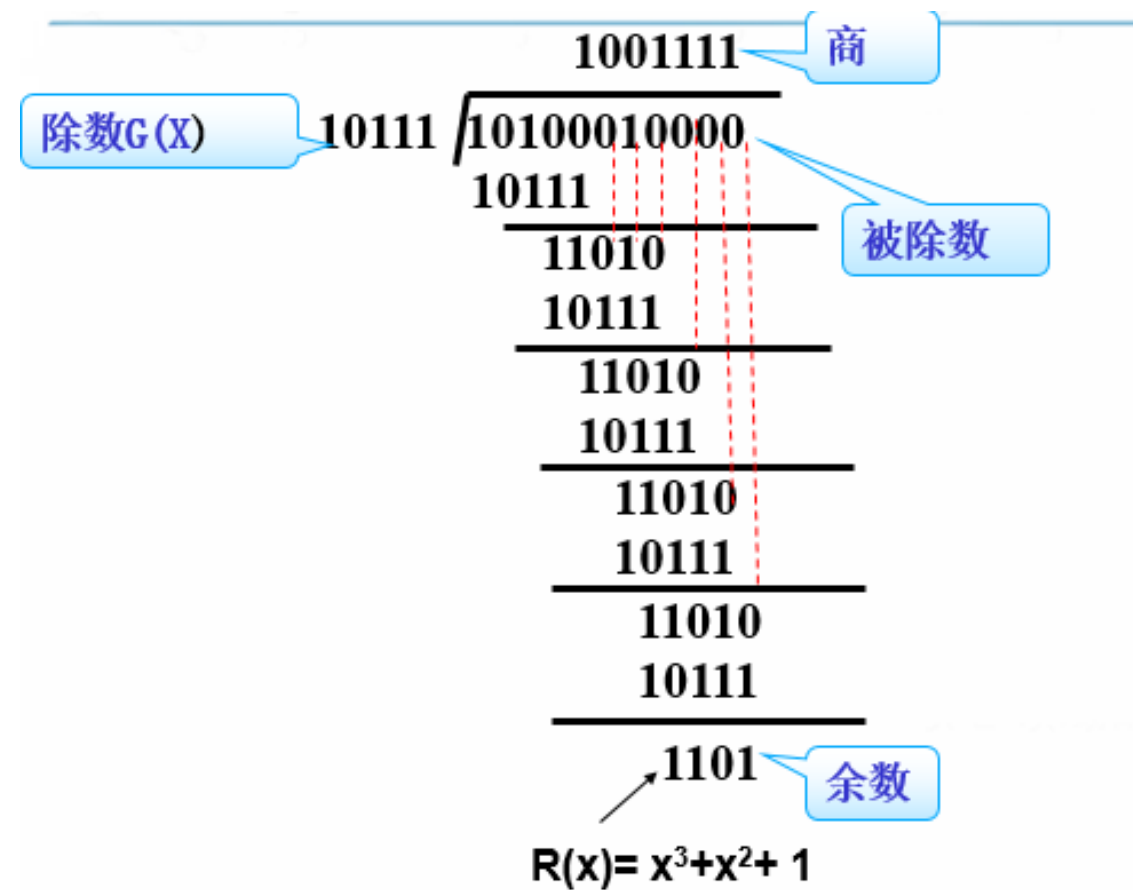
**T**: **k+r**位

## 例

①  $K$  : 1010001,  $K(X) = x^6 + x^4 + 1$

② 取  $G(x) = x^4 + x^2 + x + 1$  (10111)

$\langle K, R \rangle = 10100011101$



✓发送方:

- 在数据K后面添加  $r$  个0 ( $x^r.K(x)$ )
- 除以  $G(x)$  , 得到的余数  $R(x)$  ( $r$ 位)
- 发送  $T(x) = x^r.K(x) + R(x)$

• 接收方:  $T(x)/G(x)$

如没有差错, 则余数必为0

如余数不为0, 产生差错

## 例

①  $K : 1010001, K(X) = x^6 + x^4 + 1$

② 取  $G(x) = x^4 + x^2 + x + 1$  (10111)

$\langle K, R \rangle = 10100011101$



## CRC检错能力：存在漏检的可能

- 余数 $R=0$ ，数据可能出错
- 检错能力取决于选取的生成多项式 $G(X)$

## CRC检错能力：存在漏检的可能

$$\text{CRC-12} = x^{12} + x^{11} + x^3 + x^2 + x + 1$$

$$\text{CRC-16} = x^{16} + x^{15} + x^2 + 1$$

$$\text{CRC-CCITT} = x^{16} + x^{12} + x^5 + 1$$

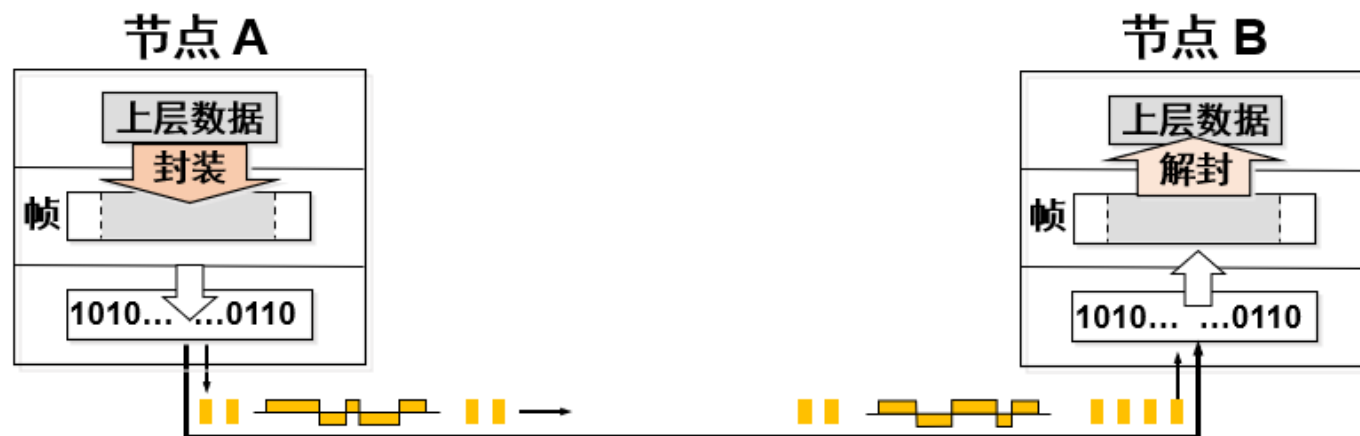
$G(x)$ 含有常数项1，多项式阶数为 $r$ ，能检测出所有少于 $r+1$  bit的突发错误

✓ 差错控制编码：在传输的数据（信息位）后增加冗余位

- 奇偶校验码
- 循环冗余校验码（CRC: Cyclic Redundancy Check）
- 海明码
- Internet校验和

CRC的实现：

- 硬件实现：通过专用电路（移位寄存器、半加器），速度快
- 软件实现



## 组帧, CRC差错检测

1) 帧出错, 如何处理?

2) 帧丢失, 如何处理

——帧失序

# 帧出错

1) 采用纠错码（例如：海明码）

2) 发送方重传出错的帧

接收方收到帧，返回：

- **确认(ACK)帧**：接收方显示告诉发送方，帧正确接收
- **否定确认(NACK)**：接收方显示告诉发送方，帧出错

—————自动重传请求 (**ARQ: Automatic Repeat reQuest**)

1) ACK/NACK出错了，怎么办？

重传：帧重复

不重传：出错

2) 帧丢失，帧失序？

——在不可靠信道上设计可靠传输协议：复杂

# 可靠传输协议

- 1) 停-等协议
- 2) 回退N
- 3) 选择重传

可靠传输：

接收方**正确无误**的接收到发送方发送的所有帧

——**帧正确、不丢失、不重复、不失序**

# 停-等协议

- 1) 发送方每发送一帧，停止发送，启动计时器
- 2) 接收方进行差错检测
  - ✓ 正确：返回ACK
  - ✓ 出错：不返回ACK
- 3) 发送方

在计时器到时之前，收到ACK，发送下一个帧

计时器超时，没有收到ACK，重传——超时重传



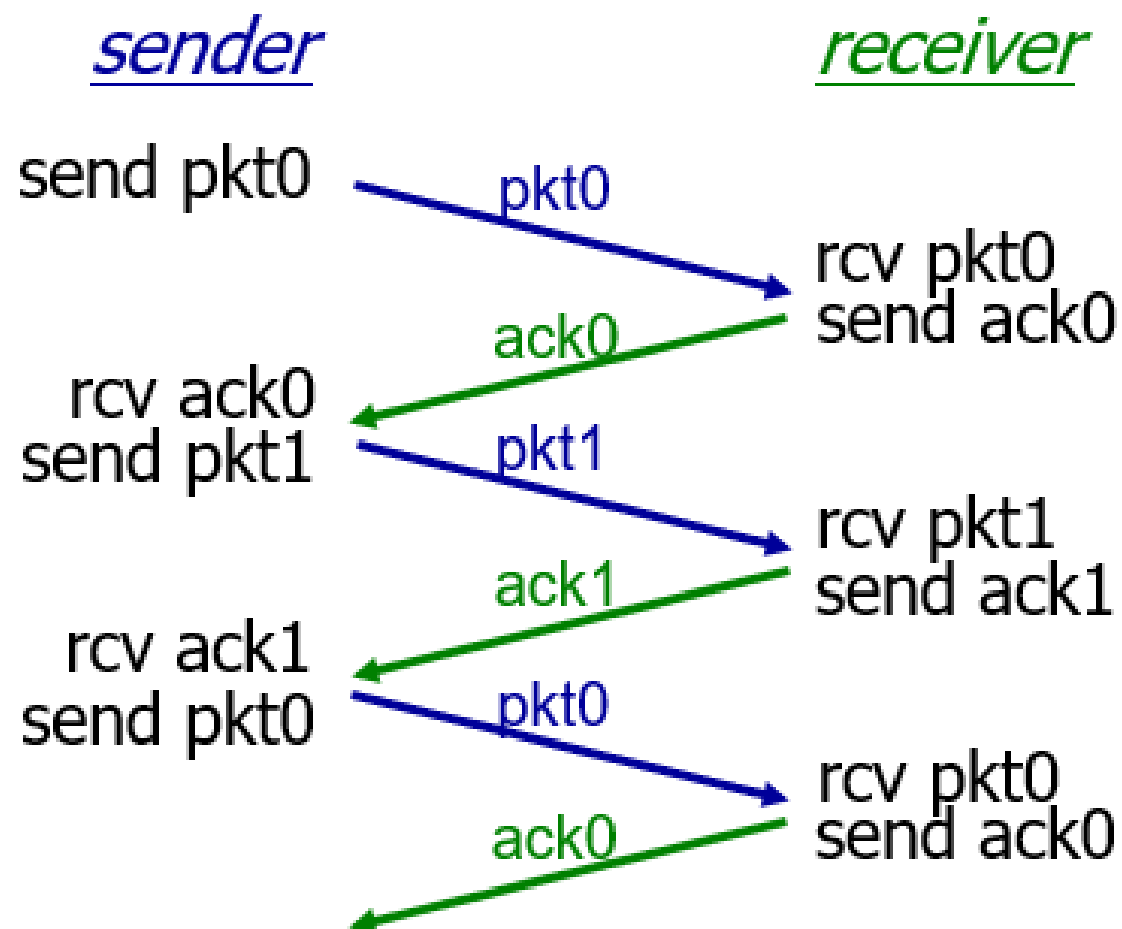
# 停-等协议

4) 对帧进行编号。

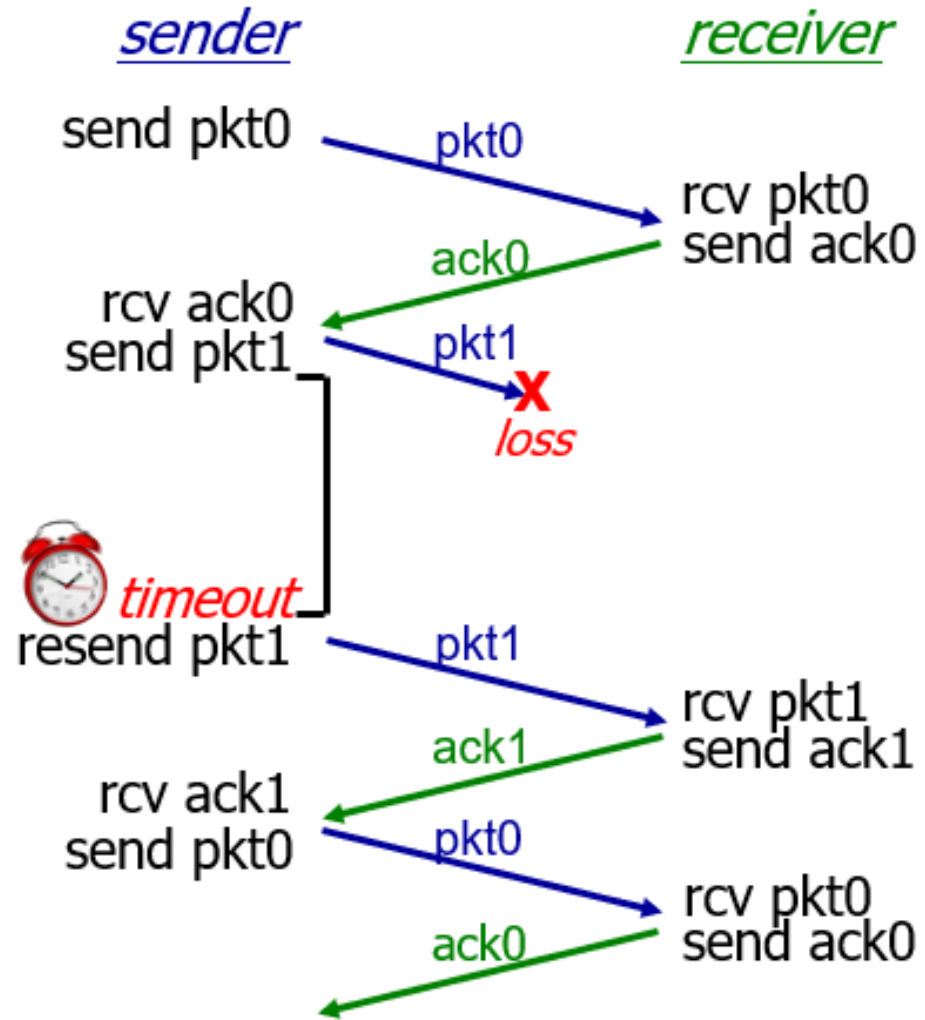
接收方区分当前帧，还是下一个帧

1bit编码：0号帧，1号帧

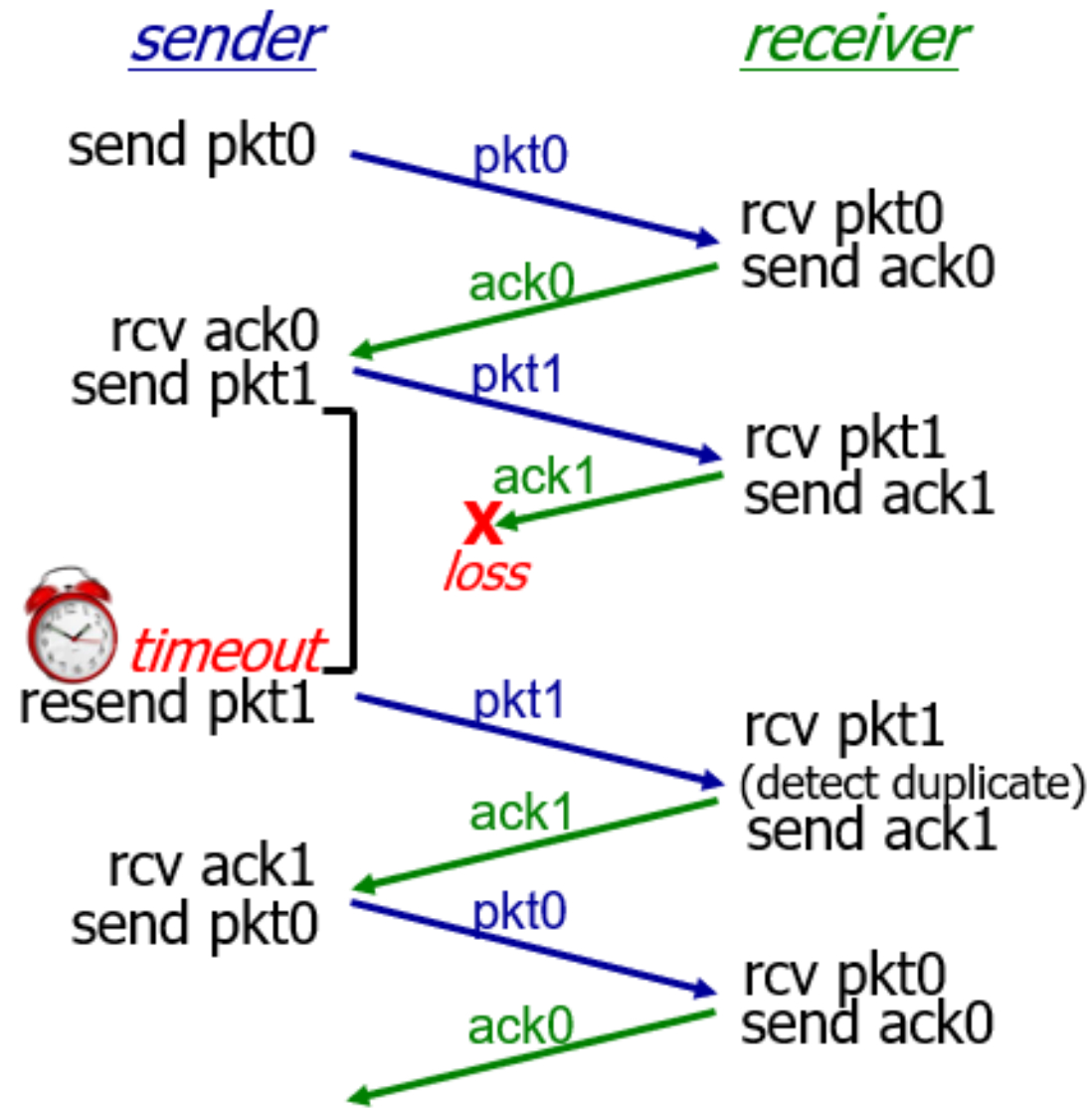
## 停-等协议



(a) no loss

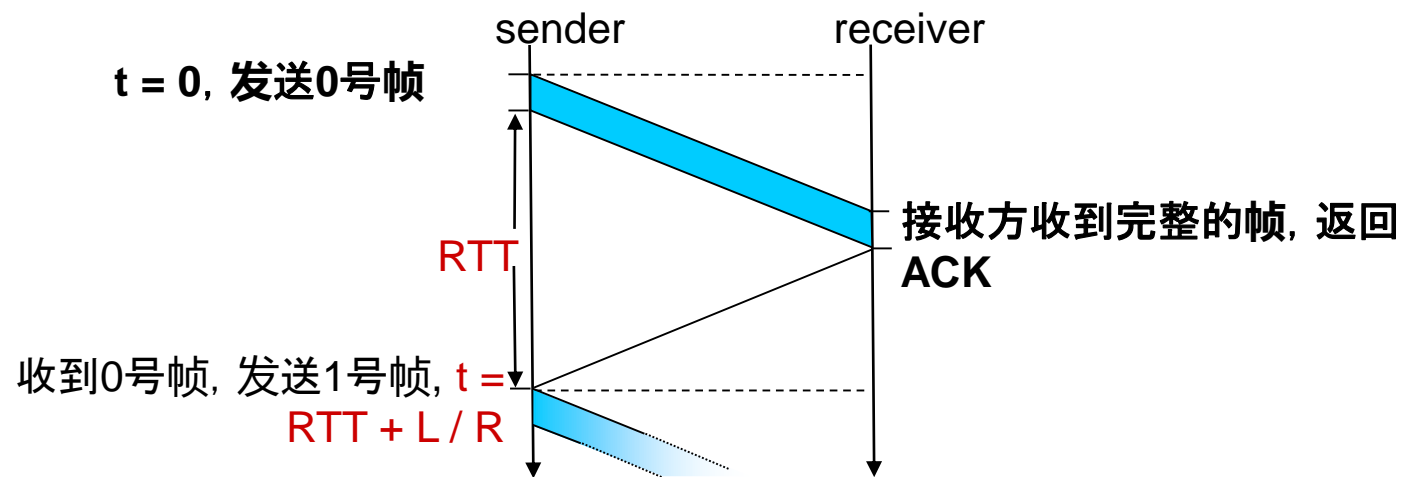


(b) packet loss



(c) ACK loss

- 停等协议：信道利用率太低



$$U_{\text{sender}} = \frac{L/R}{RTT + L/R}$$

**信道利用率：**信道在单位时间内被有效占用的比例

例如：： 1 Gbps的链路， 15 ms端-端传播延时， 帧大小为1kB

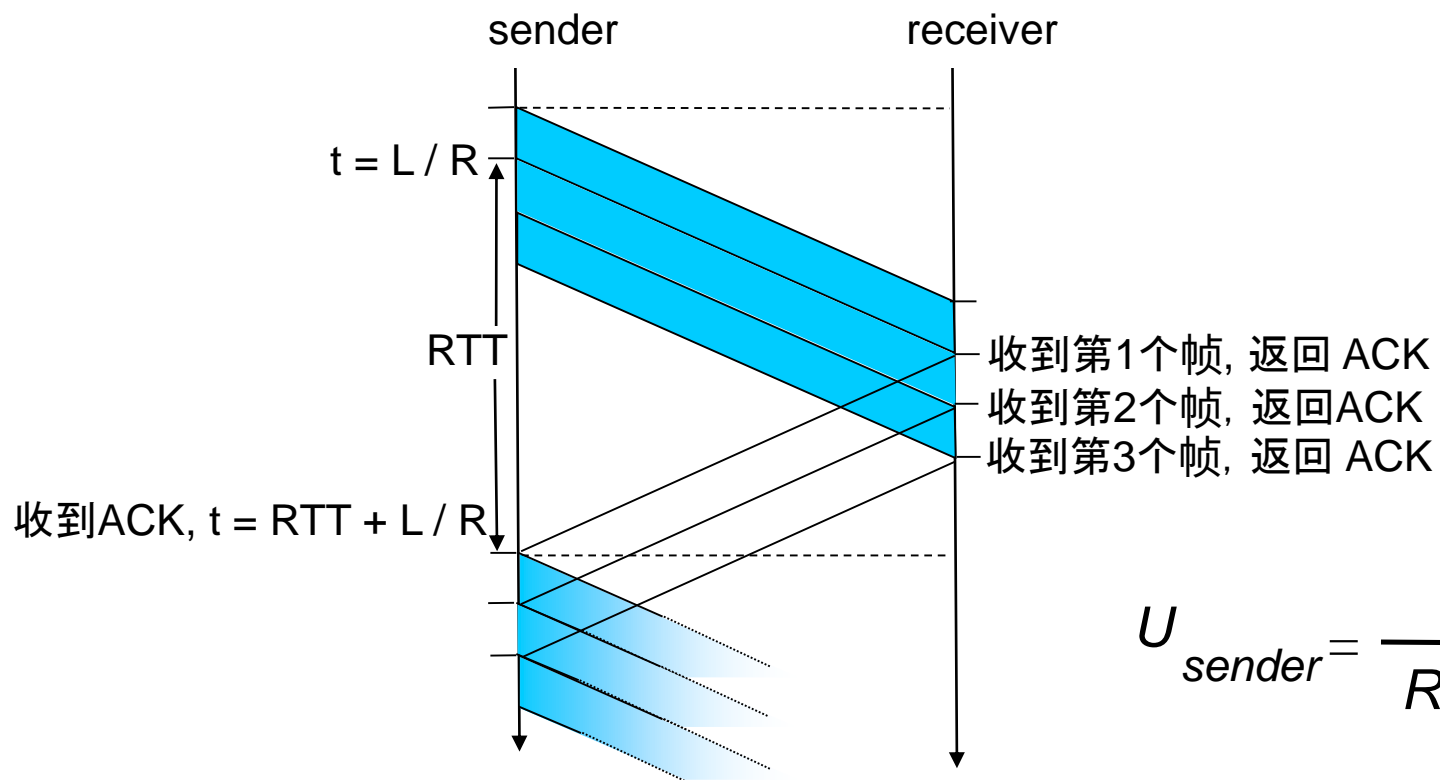
$$T_{\text{transmit}} = \frac{L \text{ (分组长度, 比特)}}{R \text{ (传输速率, bps)}} = \frac{8\text{kb/pkt}}{10^9 \text{ b/sec}} = 8\mu\text{s}$$

$$U_{\text{sender}} = \frac{L / R}{RTT + L / R} = \frac{.008}{30.008} = 0.00027$$

**传播时延比较长的信道，信道利用率非常低**

停等协议：一次发一个帧，不能充分利用信道容量

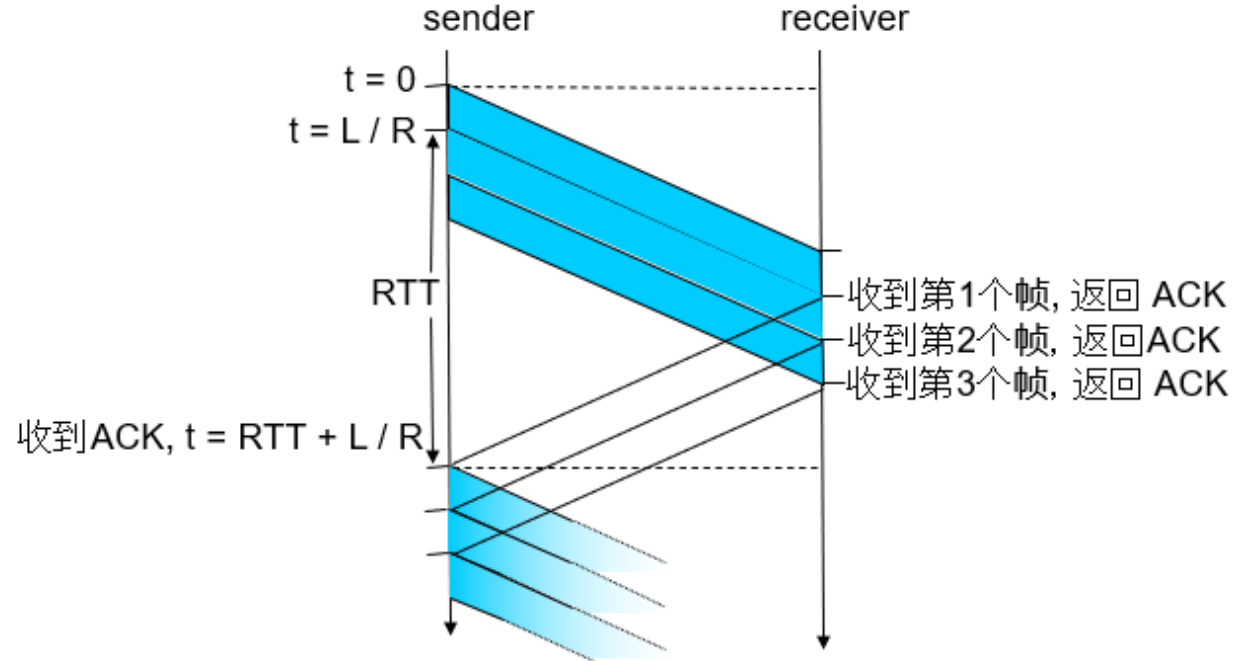
**流水线协议：**允许发送方在**未得到对方确认**的情况下一次发送**多个帧**



$$U_{\text{sender}} = \frac{3L / R}{RTT + L / R} = \frac{.0024}{30.008} = 0.00081$$



# 流水线协议



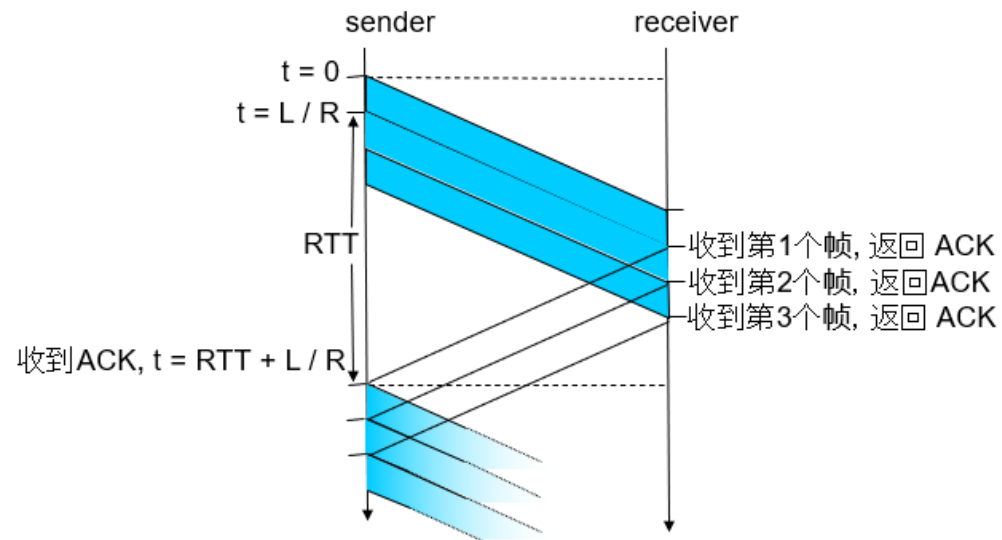
- ✓ 必须增加帧编号的范围：用多个bit表示帧号
- ✓ 确认帧也必需编号：表示对某个编号帧的确认

# 流水线协议

✓ 在发送方/接收方要有缓冲区

➤ 发送方：保留已发送，未得到确认的帧，可能需要重传；

➤ 接收方：保留收到的、还未被上层协议取走的帧；



✓ 尽可能提高信道利用率，发送方连续发帧，将等待时间填满；

- 能够连续发送的帧数： $(L/R + RTT) / (L/R)$

- 采用  $m$  比特对帧进行编号

✓ 停等协议

✓ **流水线协议**

——需要考虑接收方的接收能力，避免接收方来不及接收，缓冲区溢出

——限制发送方的发送速度（**即流量控制**）

## 流量控制

- ✓ 限制发送方连续发送帧的个数;
- ✓ 如何限制?
  - 规定一个窗口，位于窗口内的帧可以连续发送
  - 收到确认后，窗口滑动，

✓ 停等协议

✓ **流水线协议**

在流水线基础上，采用滑动窗口对发送方进行流量限制，即滑动窗口协议

从接收方的角度，接收操作：

# 滑动窗口协议

- 1) 回退N步 (GBN: Go Back N)
- 2) 选择重传 (SR: Select Repeat)

## 回退N (Go Back N)



发送方：窗口大小为N

接收方：

- 1) 按序接收
- 2) 接收到序号为n的帧，对其进行确认ACK
- 3) 所有其他情况下，丢弃收到的帧

——如果n号帧丢失，则n号帧及后续所有发出的帧，发送方都需要超时重发（回退N）

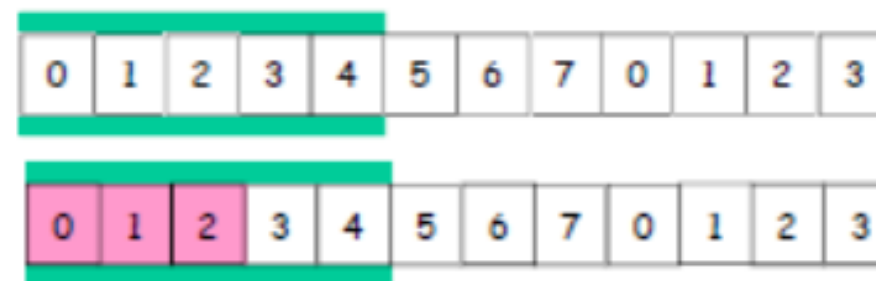


## 回退N (Go Back N)



接收方按序对最近接收的帧进行ACK，表示之前的帧已正确收到（**累积确认**）

## 选择重传 (SR: Select Repeat)



### 接收方

- 1) 乱序接收
- 2) 如n号帧丢失，后续到达的帧可以放到接收窗口中，不向上交付
- 3) 对到达的帧返回ACK

# 发送窗口和接收窗口的大小

GBN

发送窗口:  $W_t = N$

接收窗口:  $W_r = 1$

SR

发送窗口:  $W_t = N$

接收窗口:  $W_r > 1$

停-等

发送窗口:  $W_t = 1$

接收窗口:  $W_r = 1$

# 帧编号

停-等协议：1bit编号，0/1

GBN/SR：窗口大小与帧编号位数之间的关系？

例 **GBN**: 发送窗口 $W_t$ , 接收窗口 $W_r=1$ ,  $m$  bit编号

$W_t$



$W_r$



GBN: 发送窗口 $W_t$ , 接收窗口 $W_r=1$ ,  $m$  bit编号

$$W_t + W_r \leq 2^m$$

$$W_t \leq 2^m - 1$$

SR:

$$W_{\dagger} \leq W_r, \quad W_{\dagger} + W_r \leq 2^m$$

$$W_{\dagger} \leq 2^{m-1}$$

见课后习题3.23

# 第三章 链路层

## ✓ 链路层功能

- 1、组帧
- 2、差错控制
- 3、可靠传输
- 4、流量控制

## ✓ 链路层协议



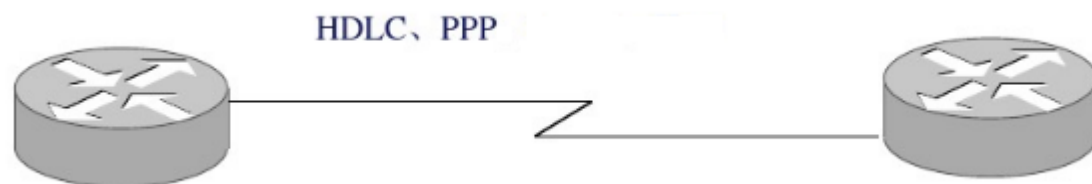
## 4 链路层协议

### 1) 点到点协议 (PPP: Point to Point Protocol)

- ✓ 组帧、差错检测, Internet使用最为广泛的链路层协议

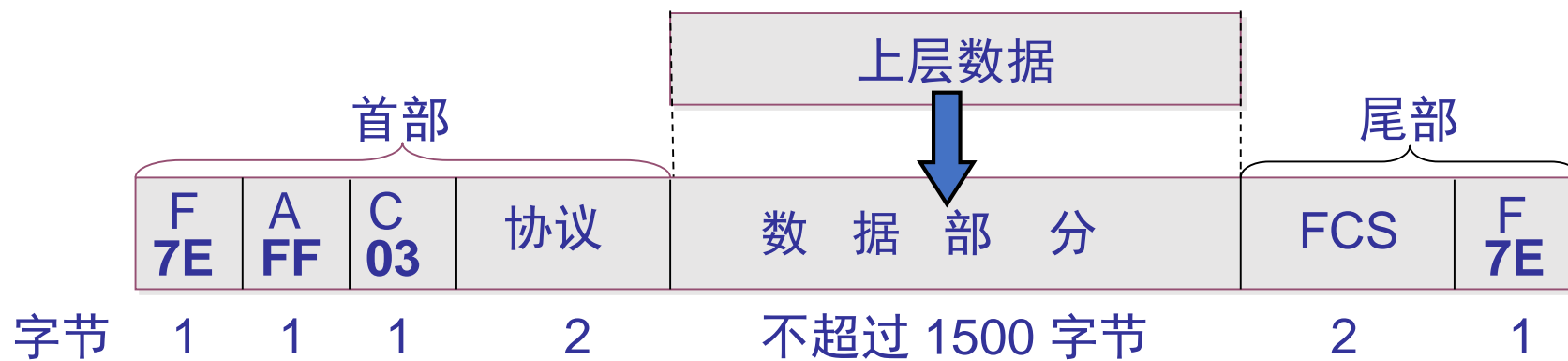
### 2) 高级数据链路控制协议 (HDLC: High-Level Data Link Control)

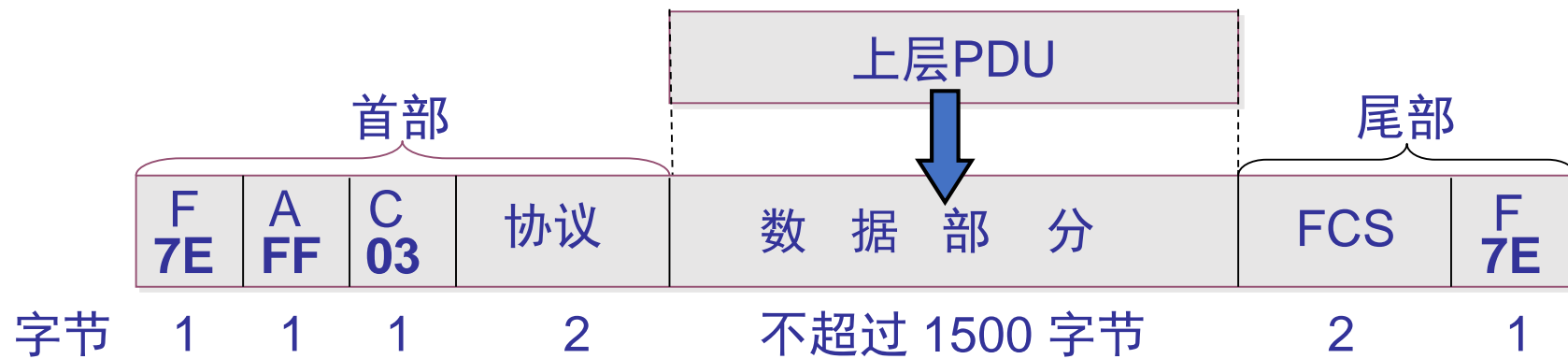
- ✓ +可靠传输, 通常用于专用网络中, 对数据传输的可靠性要求较高的场景



# PPP

- PPP帧格式



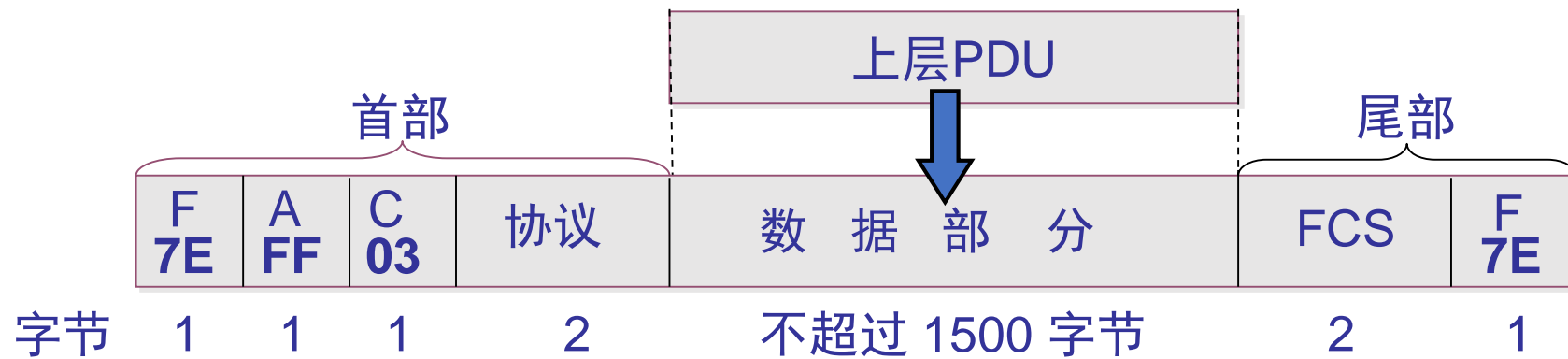


帧定界

异步传输：7E

同步传输：01111110

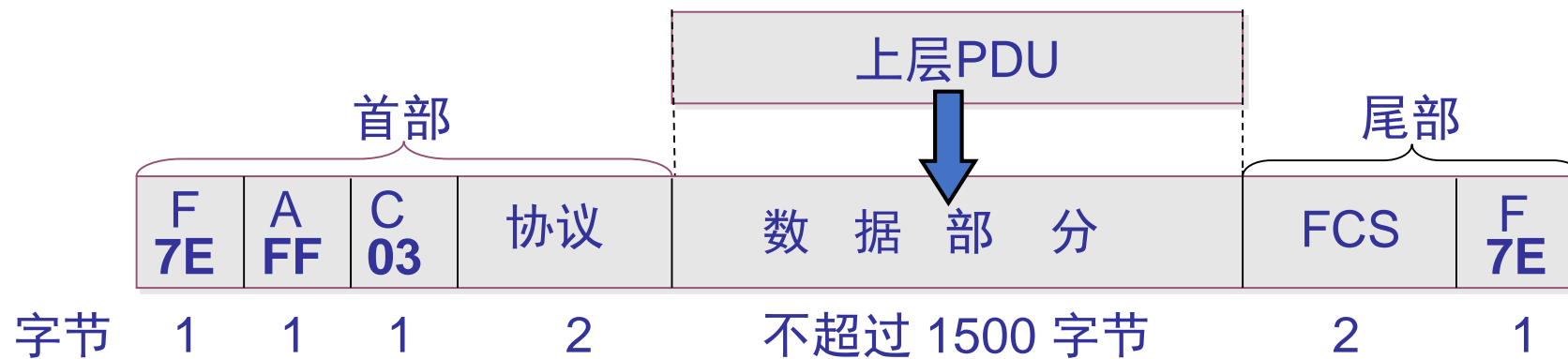
——数据部分出现相同的字符或(bit)，需要进行特殊的处理



地址字段: 0xFF (固定)

控制字段: 0x03 (无编号帧)

FCS (Frame Check Sequence): 16bit的CRC码



协议：标识上层协议的类型

0x0021: IP协议 (IPv4)

0x0035: IP协议 (IPv6)

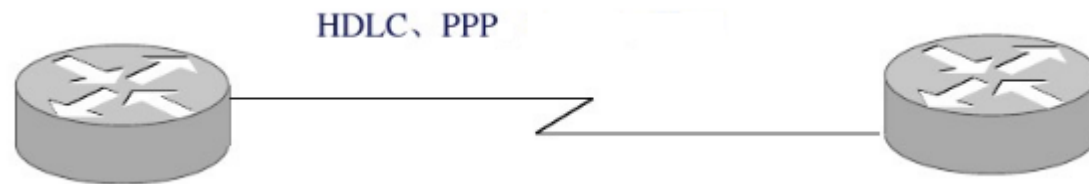
0xC021: LCP协议 (PPP链路控制协议)

.....

# PPP

- ✓ 使用CRC进行差错检测
- ✓ 没有帧编号和确认帧，不提供可靠传输
- 用于信道质量较好的场合，可靠传输交到上层完成

# PPP



- 协议族
  - 链路控制协议：LCP（Link Control Protocol）：建立、配置、测试和维护链路连接
  - 一组网络控制协议：NCP（Network Control Protocol）：用于建立和配置不同的网络层协议
  - 认证协议（PAP：口令认证协议/CHAP：挑战握手协议）
- ✓ 封装上层多种协议，构成PPP数据帧，点到点传输

# HDLC(高级数据链路控制协议)

✓ 提供可靠传输，用于信道质量较差的链路

01111110	地址	控制	数据	帧检验	01111110
字节数 1	1	1	任意	2	1

1	2	3	4	5	6	7	8
0	N(S)			P/F	N(R)		

信息帧:

N(S): 发送帧的编号

N(R): 接收帧的编号 (ACK: 捎带确认)



# HDLC(高级数据链路控制协议)

✓ 提供可靠传输，用于信道质量较差的链路

01111110	地址	控制	数据	帧检验	01111110
字节数 1	1	1	任意	2	1

## 监控帧

01: 要求发送方从序号 N(R) 开始重发后续所有 数据帧 (GBN)

11: 要求发送方仅重发序号为 N(R) 的单个数据帧

1	2	3	4	5	6	7	8
1	0	S	P/F	N(R)			

01111110	地址	控制	数据	帧检验	01111110
字节数 1	1	1	任意	2	1

1	2	3	4	5	6	7	8
0	N(S)			P/F	N(R)		
1	0	S		P/F	N(R)		
1	1	M		P/F	M		

- **信息帧 (I帧)：传输数据**

N(S)：发送帧的帧编号

N(R)：捎带确认

- **监控帧 (S帧)**

——支持GBN和SR

- **无编号帧 (U帧)**

01111110	地址	控制	数据	帧检验	01111110
字节数 1	1	1	任意	2	1

1	2	3	4	5	6	7	8
0	N(S)			P/F	N(R)		
1	0	S		P/F	N(R)		
1	1	M		P/F	M		

- **信息帧 (I帧)：传输数据**

N(S)：发送帧的帧编号

N(R)：捎带确认

- **监控帧 (S帧)**

——支持GBN和SR

- **无编号帧 (U帧)**

## 第三章 链路层

### ✓ 链路层功能

- 1、组帧
- 2、差错控制
- 3、可靠传输
- 4、流量控制

### ✓ 链路层协议