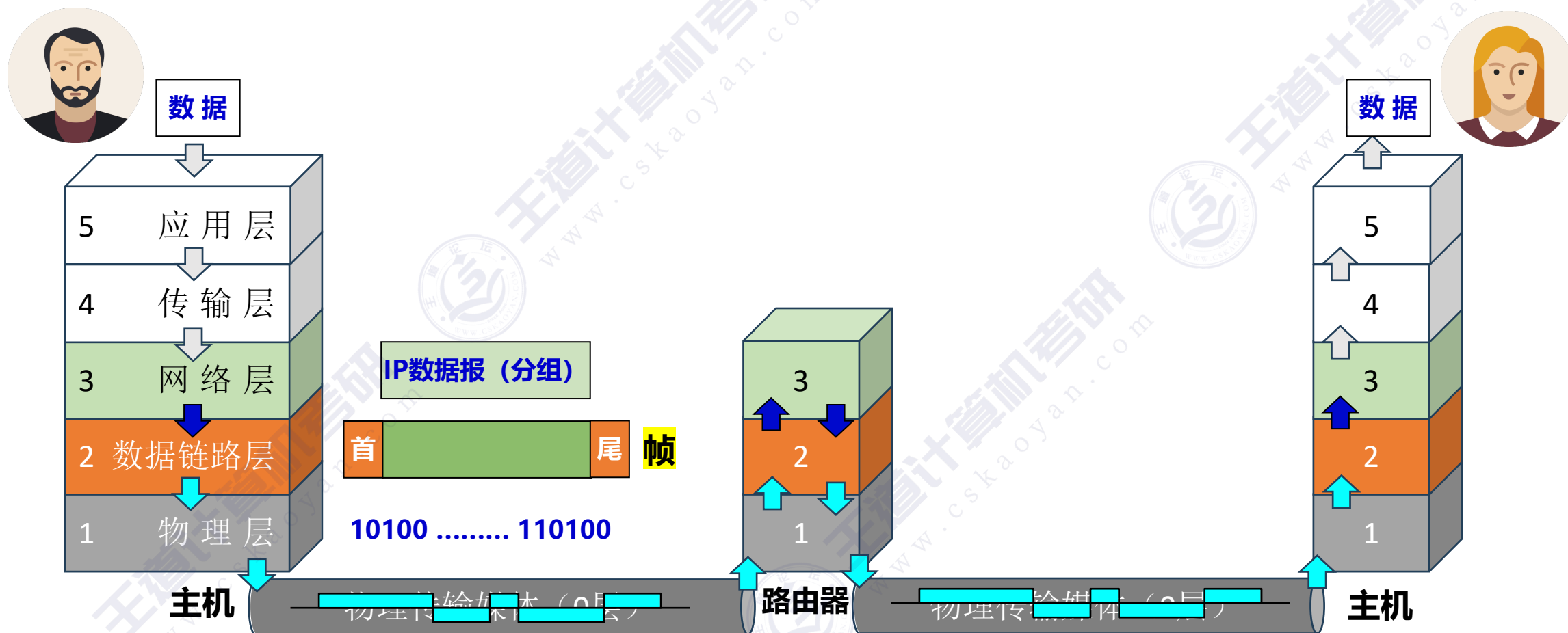


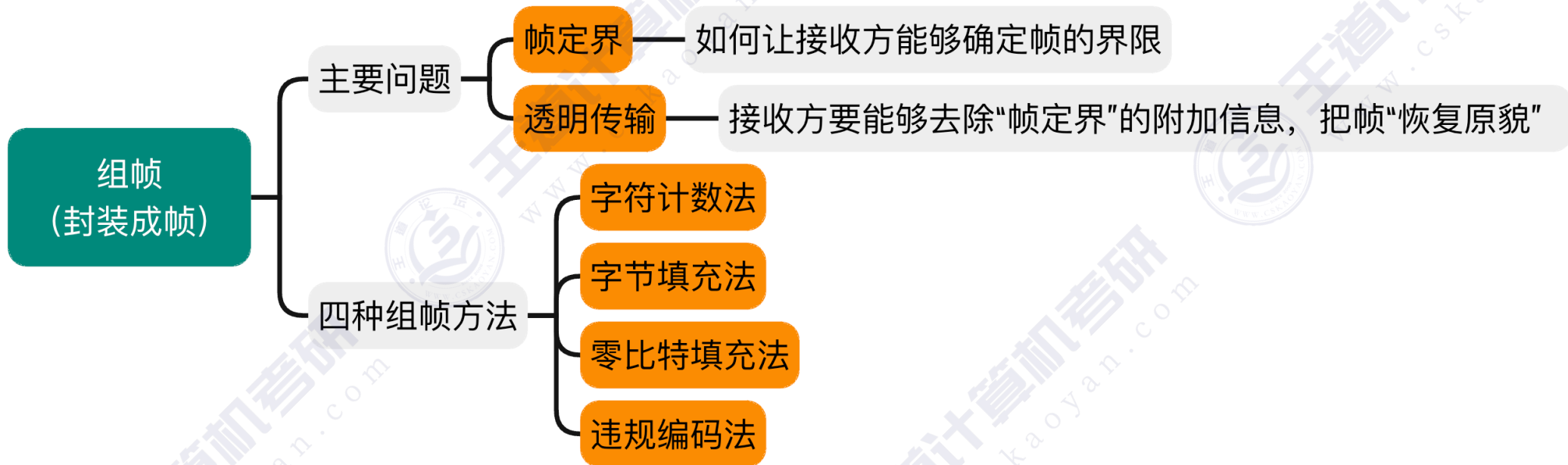
本节内容

# 组帧

# 数据链路层示意图



## 本节总览



# 字符计数法

6字节

帧的数据部分

010000010000010000011011001111110000000101011111

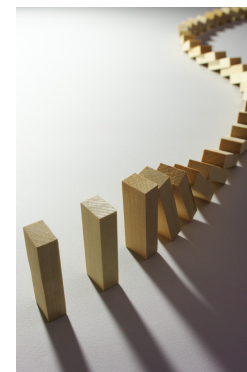
帧  
定  
界

00000111010000010000010000011011001111110000000101011111

帧大小=7字节

注意：帧长 = 计数字段长度 + 帧的数据部分长度

- 原理：在每个帧开头，用一个定长计数字段表示帧长
- 最大缺点：任何一个计数字段出错，都会导致后续所有帧无法定界



# 字节填充法

帧的数据部分

010000010000010000011011001111110000000101011111

帧定界

0000000101000001000001000001101100111111000000010101111100000100

控制字符SOH

控制字符EOT

新的问题

0000000101000001000001000001101100111111000000010101111100000100

控制字符SOH

控制字符EOT

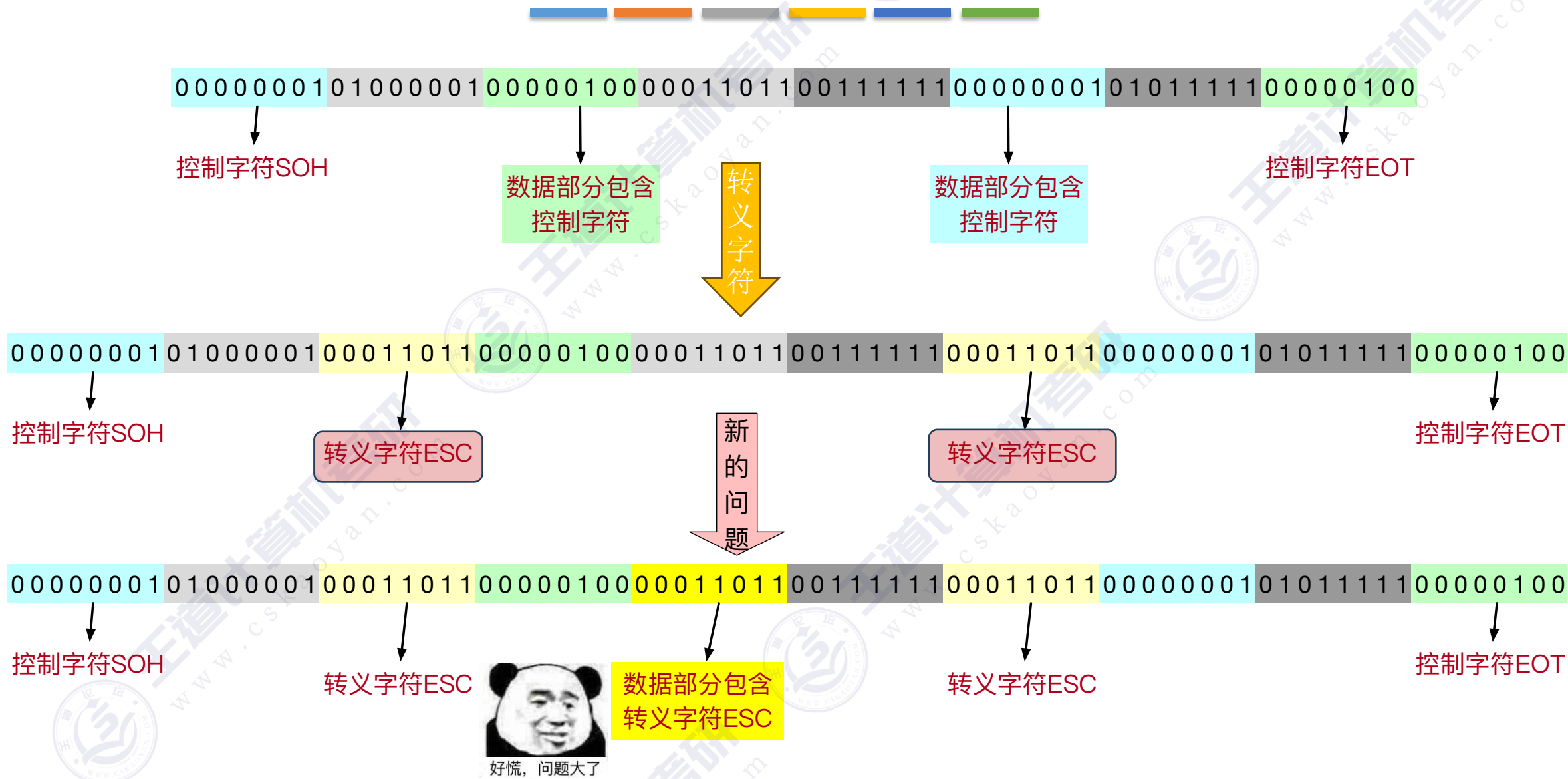
数据部分包含  
控制字符

数据部分包含  
控制字符

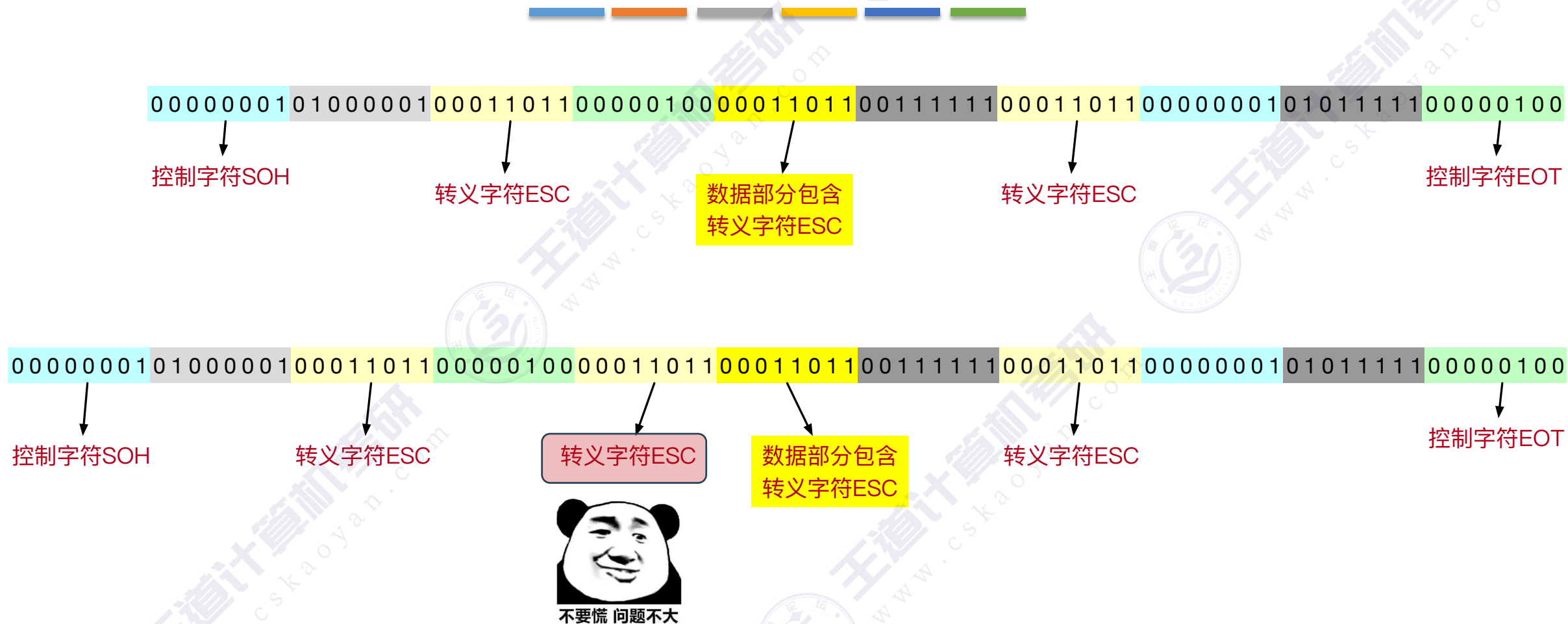


不要慌，问题不大

# 字节填充法



# 字节填充法



如果帧的数据部分包含“特殊字符”，则发送方需要在这些“特殊字符”前填充“转义字符 ESC”（接收方要做逆处理）

# 零比特填充法



帧的数据部分

010000010000010000011011001111110000000101011111

帧定界

0111111001000001000001000001101100111111000000010101111101111110

特殊比特串表示  
帧开始/结束

特殊比特串表示  
帧开始/结束

新的  
问题

011111100100000100000100000110110011111110000000101011111011111110

特殊比特串表示  
帧开始/结束

数据部分可能出现特殊比特串

特殊比特串表示  
帧开始/结束



# 零比特填充法

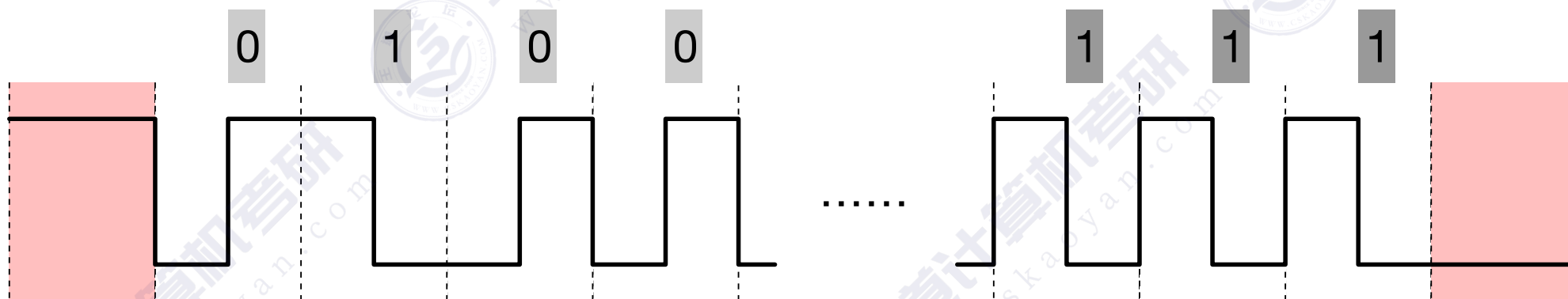


- 发送方需要对帧的数据部分进行处理，每当遇到连续5个1，就填充一个0
- 接受方需要对帧的数据部分进行逆处理，每当遇到连续5个1，就删掉后面的0

## 违规编码法

帧的数据部分

010000010000010000011011001111110000000101011111



违规编码  
(基于曼彻斯特编码)

曼彻斯特编码 (IEEE标准):  
上跳0下跳1看中间, 中必变

如果周期中间不跳变, 则“违规”

违规编码  
(基于曼彻斯特编码)

# 知识回顾与重要考点

## 组帧 (封装成帧)

