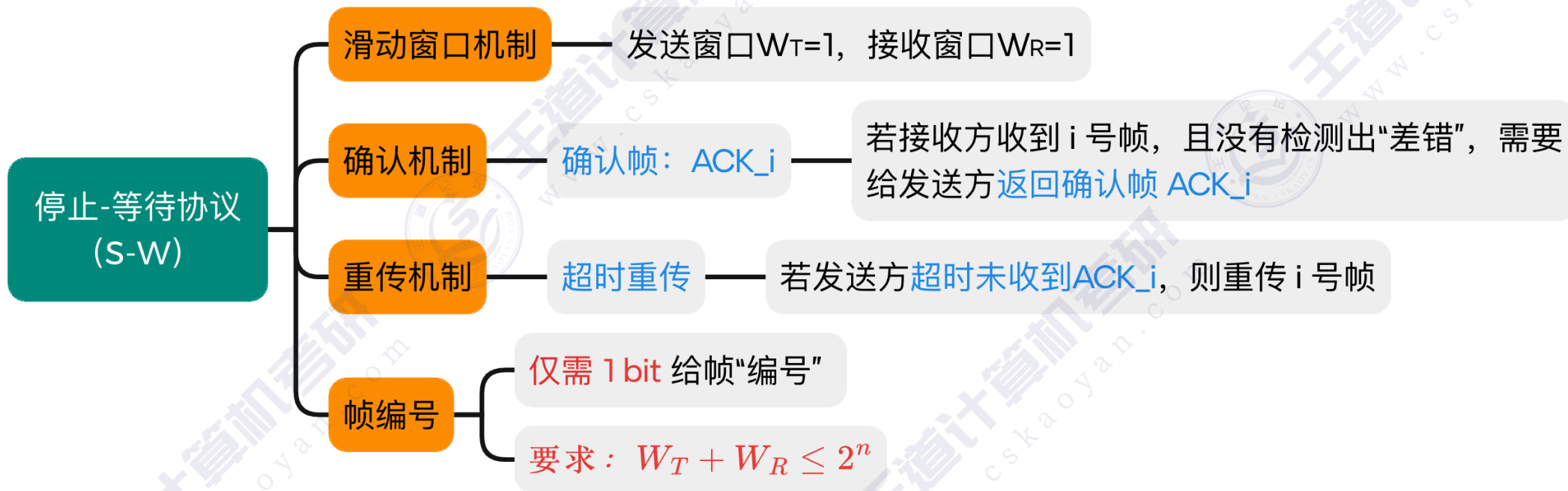


本节内容

# 停止-等待协议 (S-W)

# 要点总览



# 停止-等待协议



发送窗口  
大小  $W_T=1$

发送方

A	B	C	D	E	F	G	H	I	J	K	L
---	---	---	---	---	---	---	---	---	---	---	---

用 1bit 给帧编号

0 1 0 1 0 1 0 1 0 1 0 1

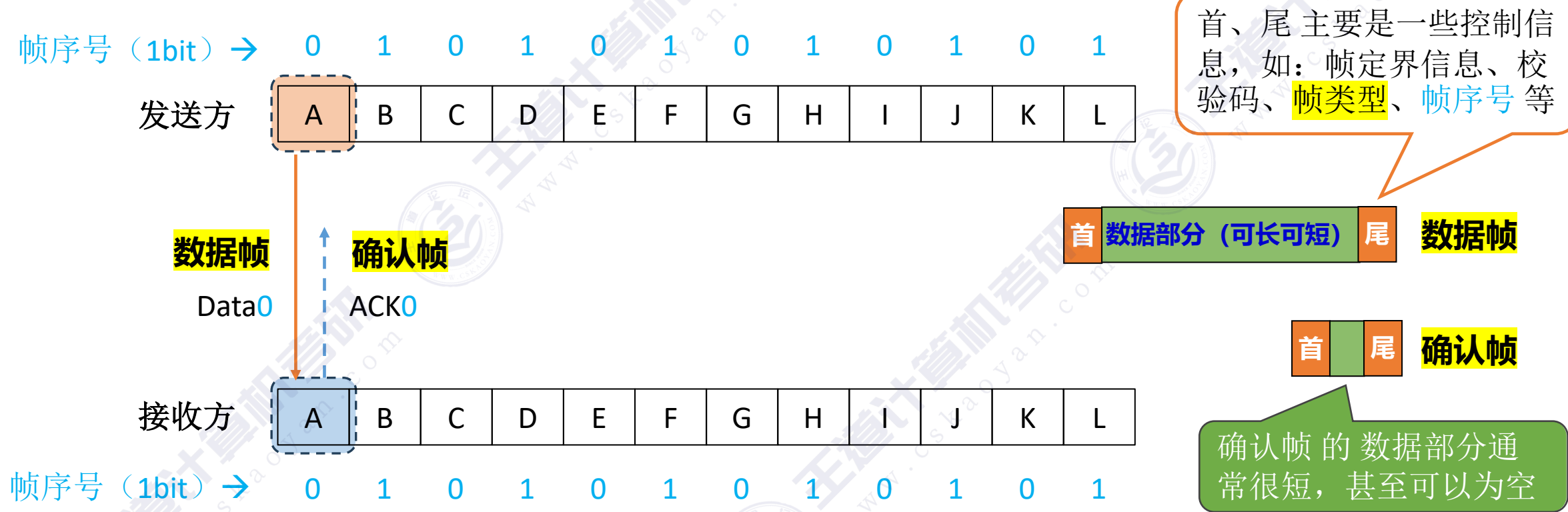
0 1 0 1 0 1 0 1 0 1 0 1

接收方

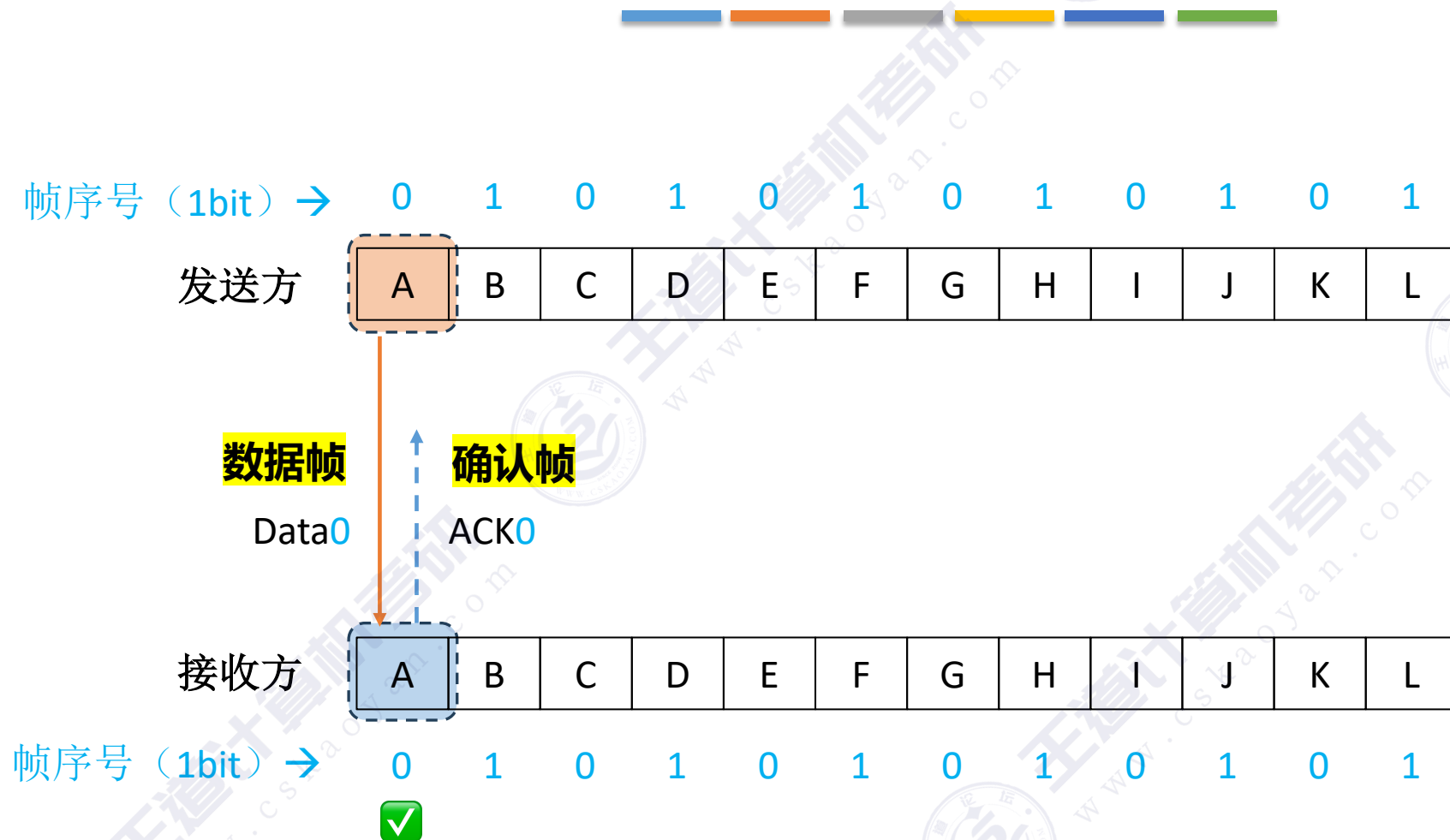
A	B	C	D	E	F	G	H	I	J	K	L
---	---	---	---	---	---	---	---	---	---	---	---

接收窗口  
大小  $W_R=1$

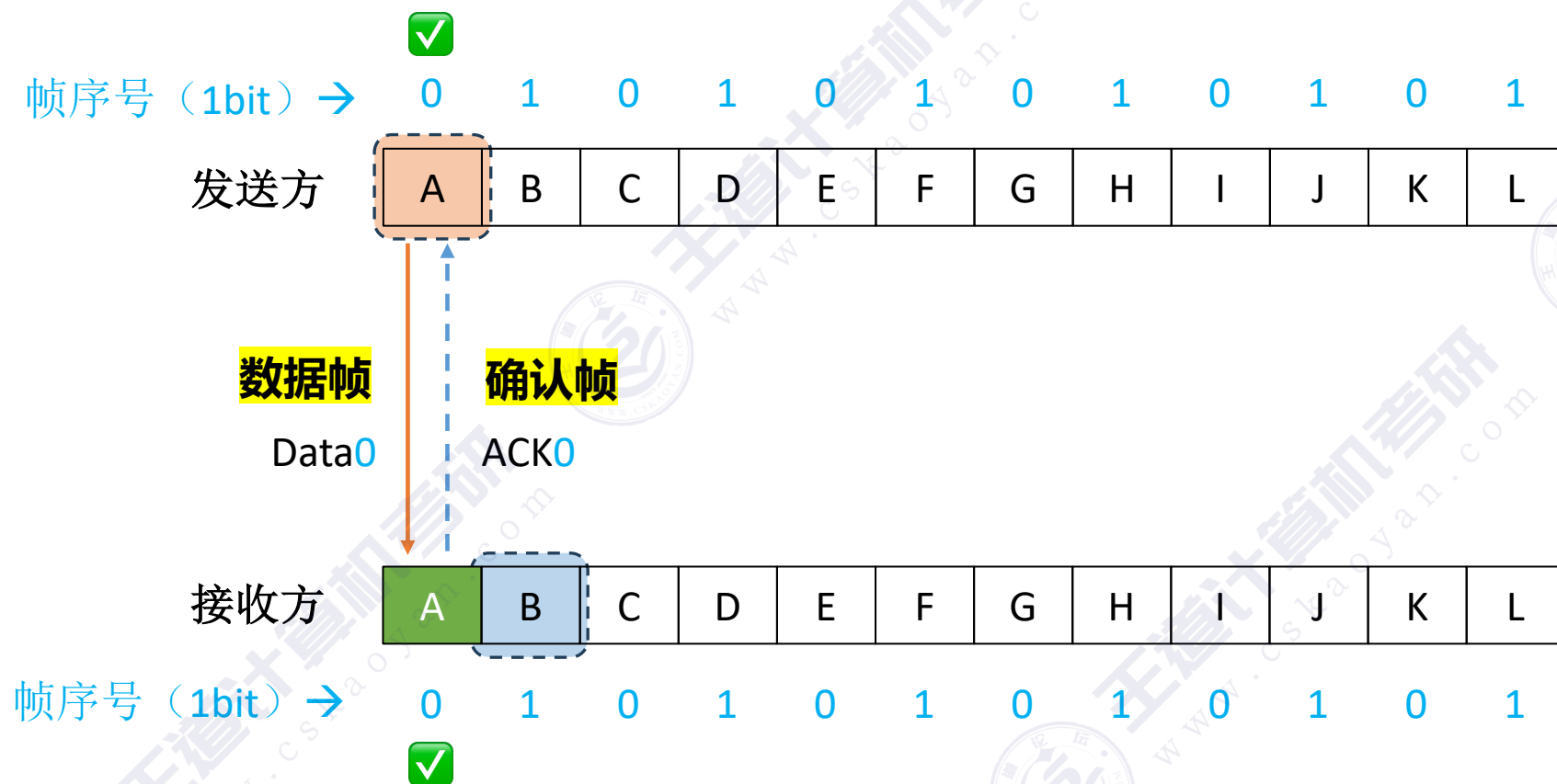
# 数据帧、确认帧、帧序号 的概念



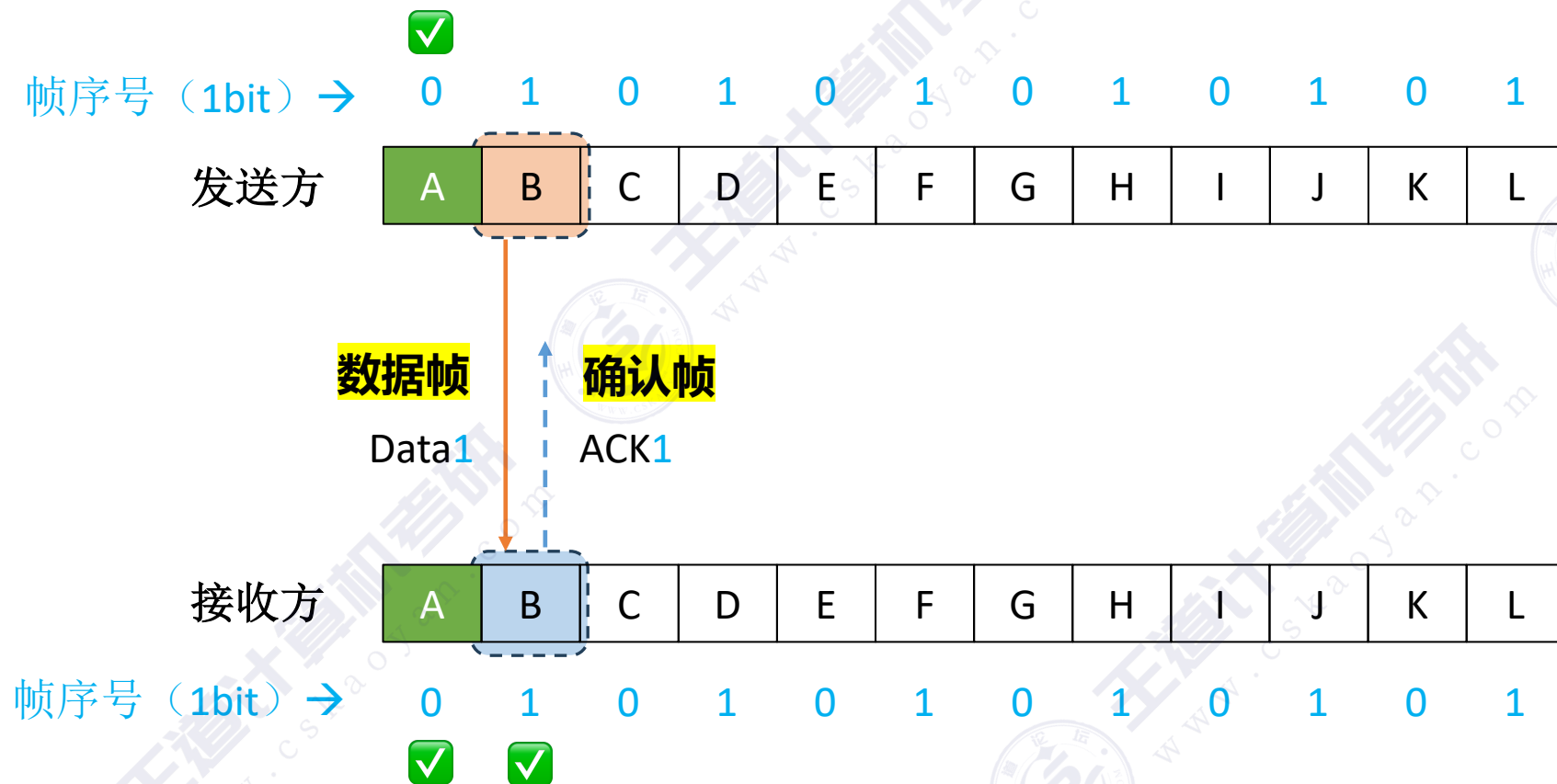
## “正常”情况示例



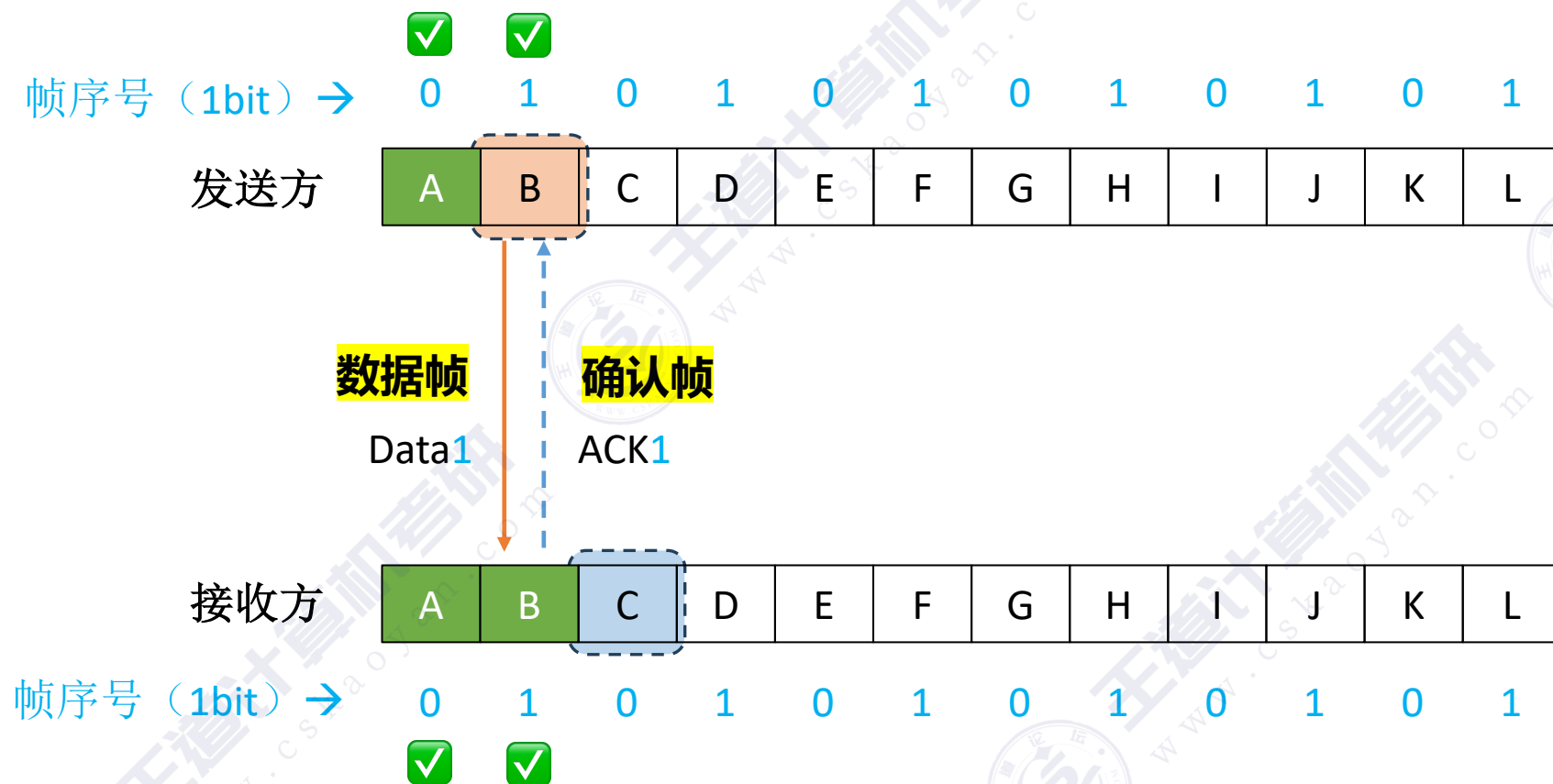
## “正常”情况示例



## “正常”情况示例

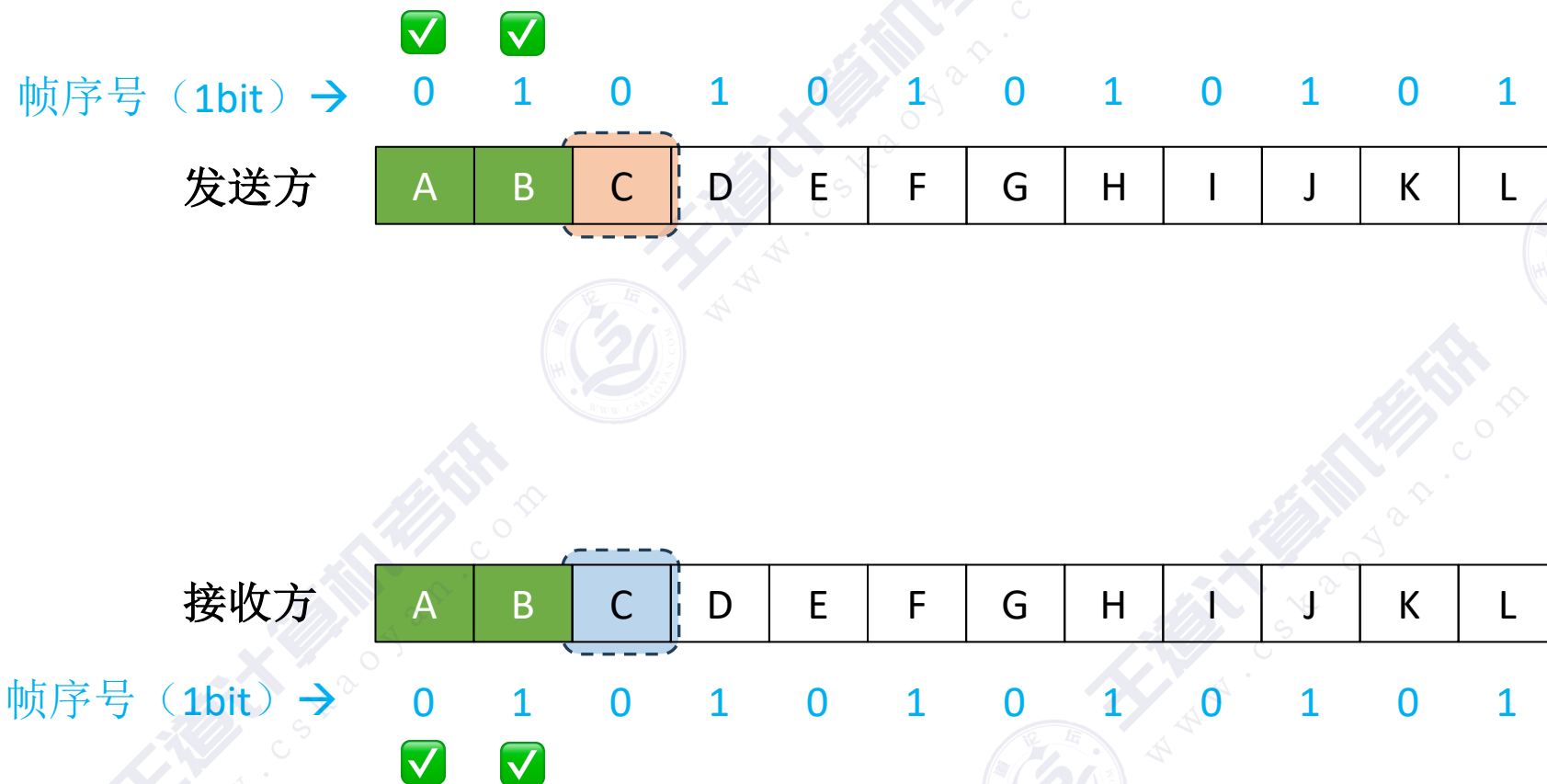


## “正常”情况示例





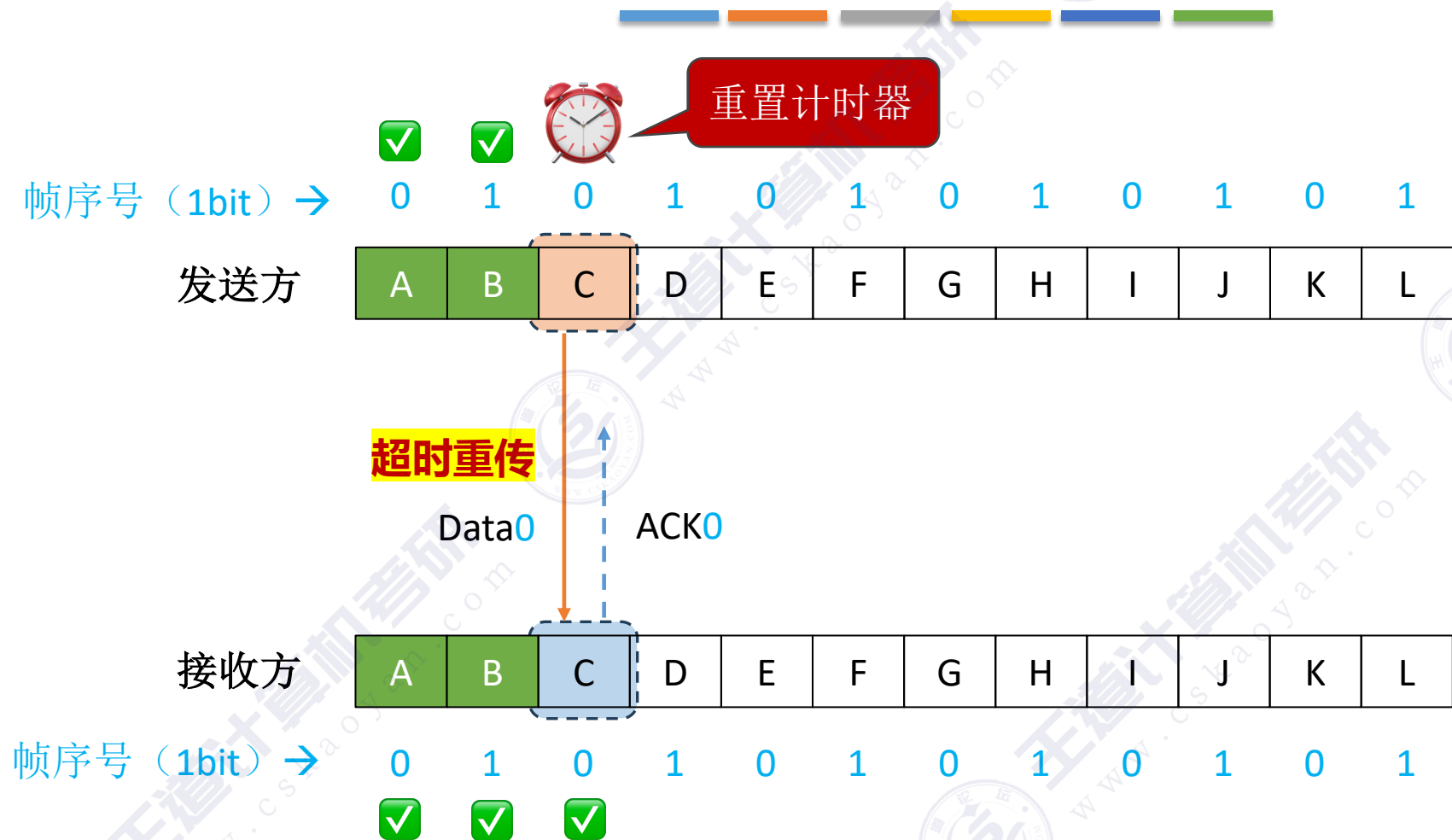
## “正常”情况示例



## 异常情况示例：数据帧丢失



## 异常情况示例：数据帧丢失



# 异常情况示例：数据帧丢失



# 异常情况示例：数据帧丢失



帧序号 (1bit) →



0 1 0 1 0 1 0 1 0 1 0 1

发送方

A	B	C	D	E	F	G	H	I	J	K	L
---	---	---	---	---	---	---	---	---	---	---	---

接收方

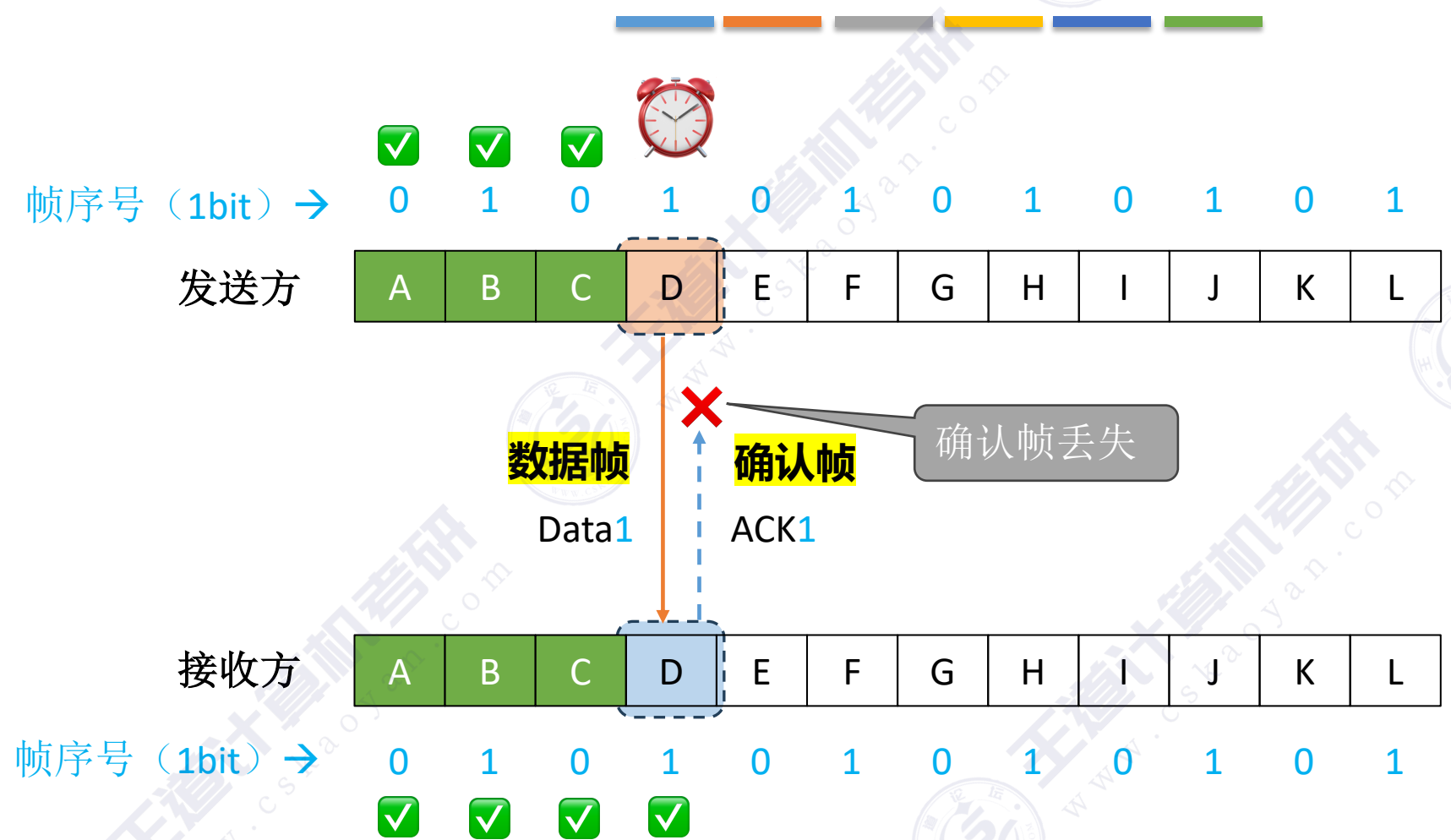
A	B	C	D	E	F	G	H	I	J	K	L
---	---	---	---	---	---	---	---	---	---	---	---

帧序号 (1bit) →

0 1 0 1 0 1 0 1 0 1 0 1



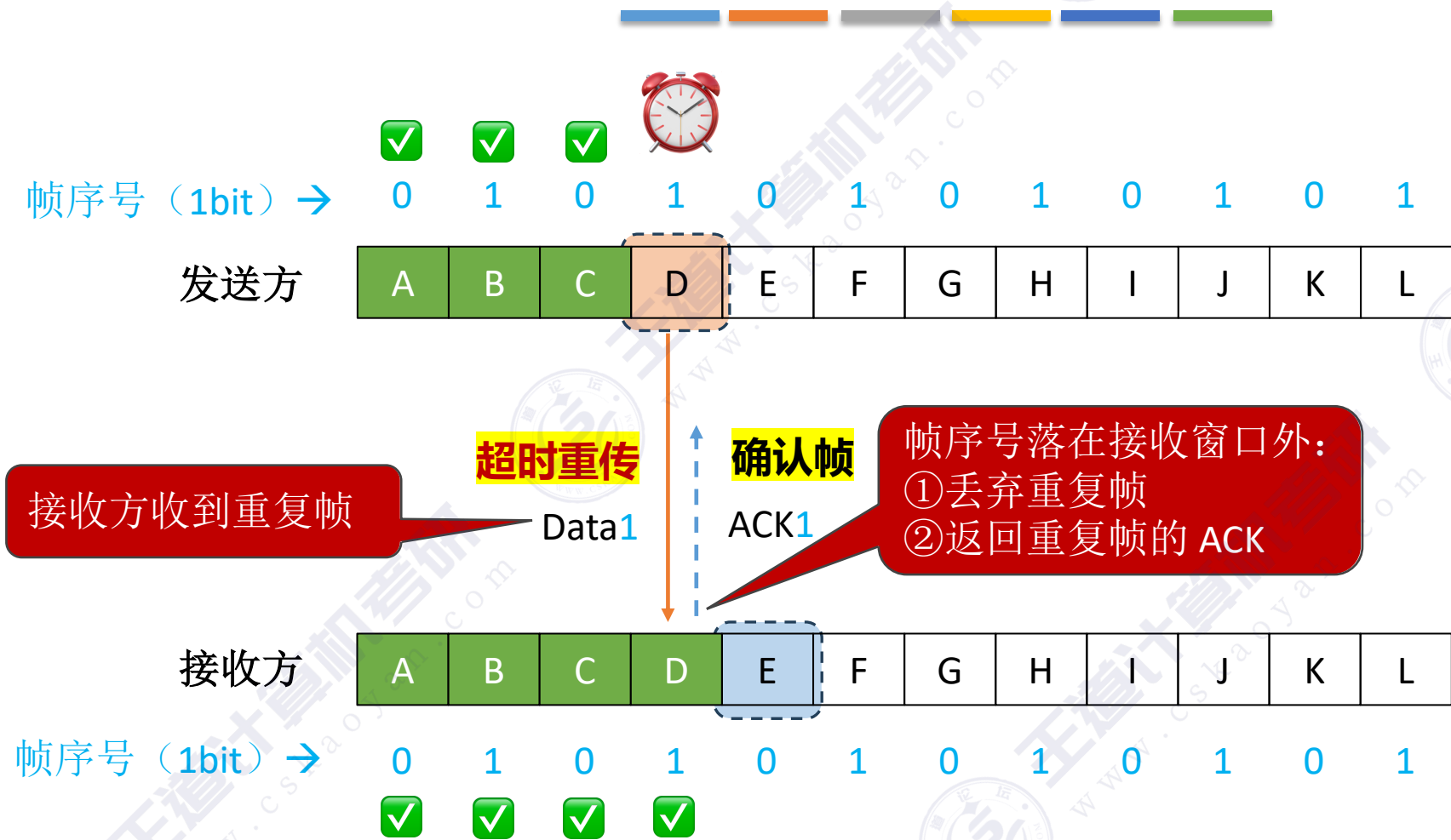
# 异常情况示例：确认帧丢失



# 异常情况示例：确认帧丢失



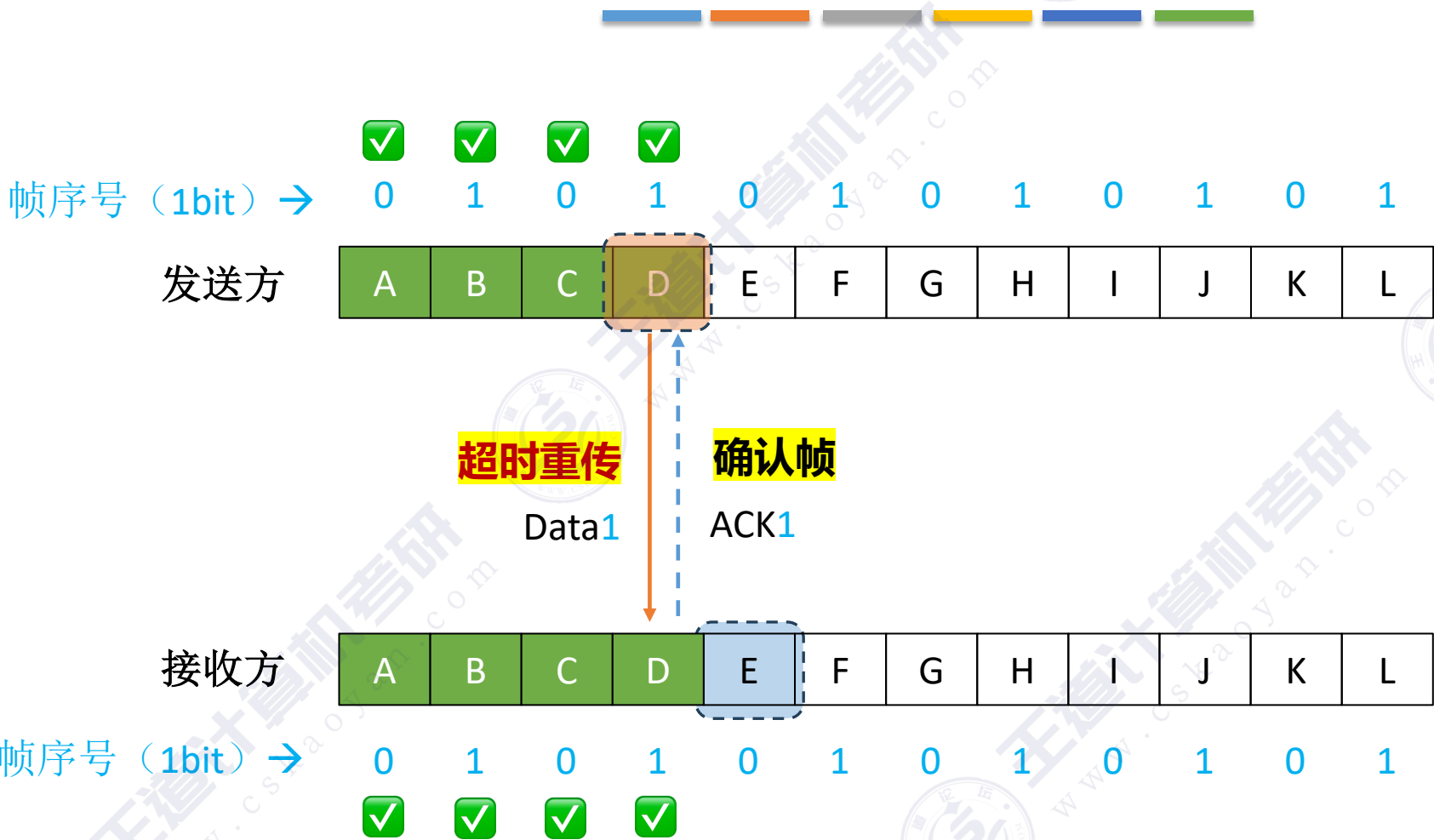
# 异常情况示例：确认帧丢失



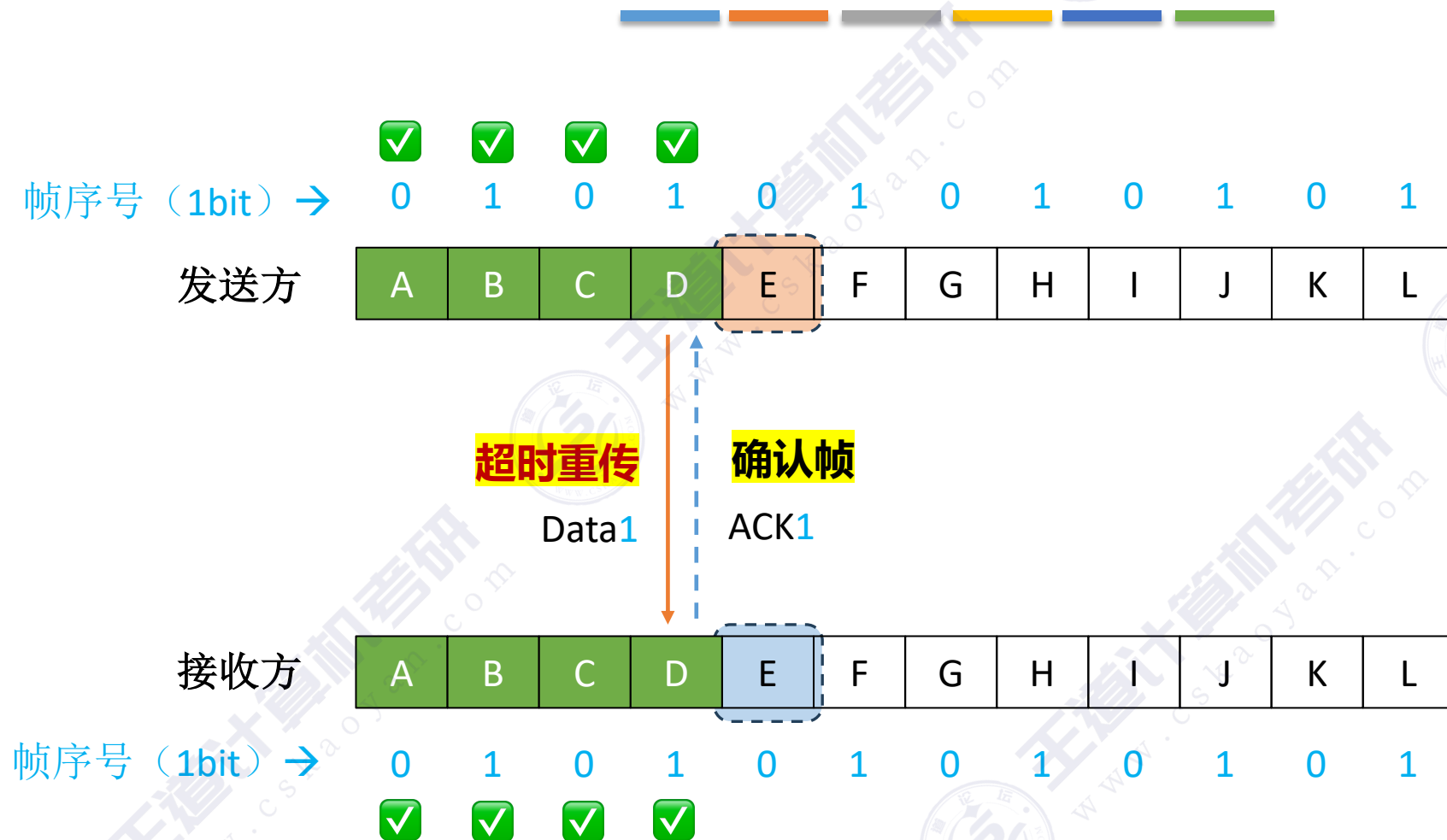
帧序号落在接收窗口外:  
①丢弃重复帧  
②返回重复帧的 ACK



# 异常情况示例：确认帧丢失



## 探讨：为什么一定要给帧“编号”



思考：在刚才这个场景中，如果没有“帧序号”会发生什么？

答：接收方无法判别“重复帧”！

由于接受窗口和发送窗口的距离不超过1，因此用 1 bit 表示帧序号足矣

## 异常情况示例：数据帧有“差错”



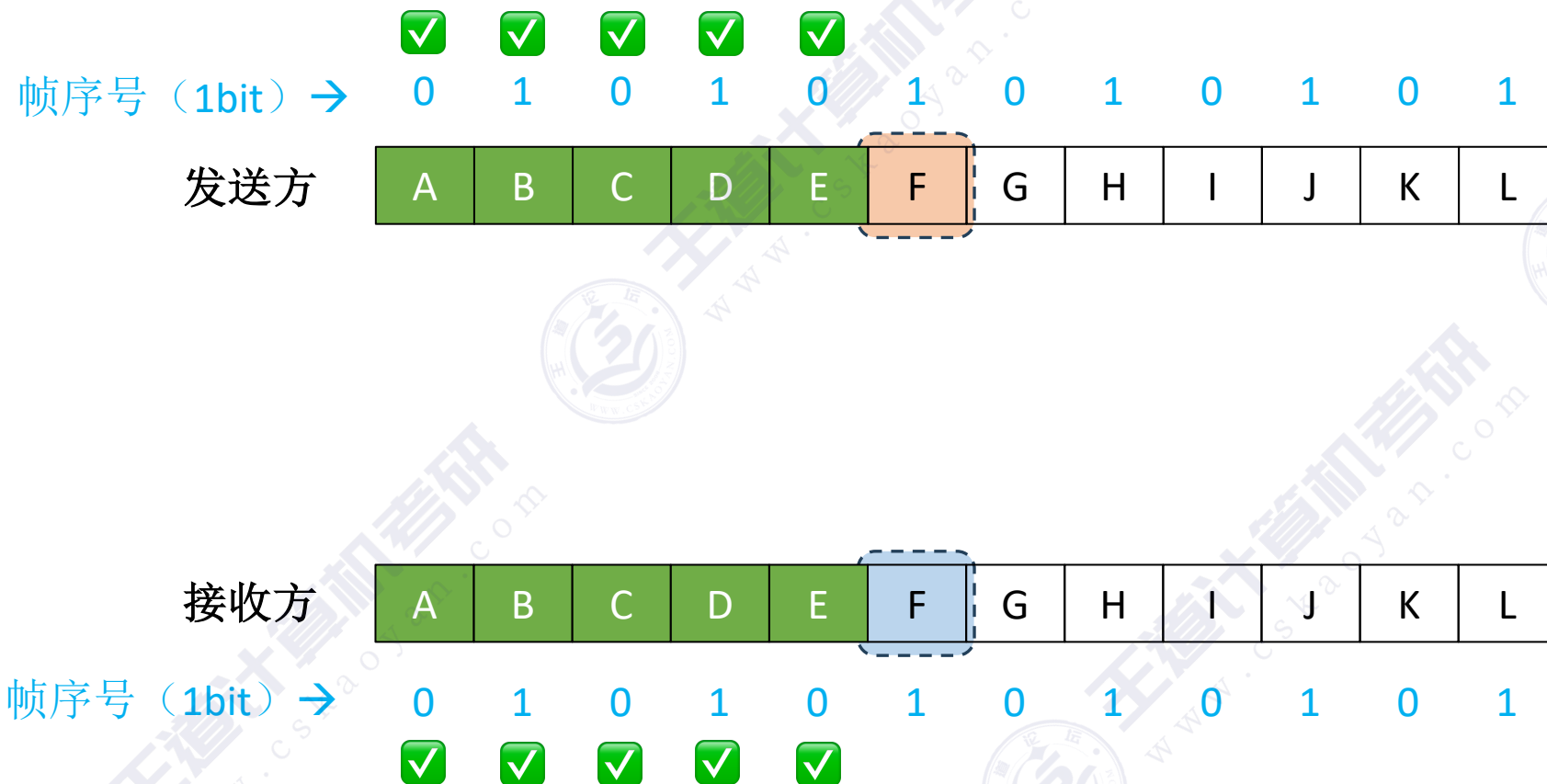
## 异常情况示例：数据帧有“差错”



## 异常情况示例：数据帧有“差错”



## 异常情况示例：数据帧有“差错”



# 知识回顾与重要考点

## 停止-等待协议 (S-W)

### 滑动窗口机制

发送窗口  $W_T=1$ , 接收窗口  $W_R=1$

### 确认机制

确认帧:  $ACK_i$

若接收方收到  $i$  号帧, 且没有检测出“差错”, 需要给发送方返回确认帧  $ACK_i$

### 重传机制

超时重传

若发送方超时未收到  $ACK_i$ , 则重传  $i$  号帧

### 帧编号

仅需 1 bit 给帧“编号”

要求:  $W_T + W_R \leq 2^n$

### 回顾

#### S-W 实现流量控制

在什么情况下发送窗口、接收窗口会“滑动”?

#### S-W 实现可靠传输

如果数据帧丢失, 收/发双方会发生什么?

如果接收方检测出数据帧有“差错”并将其丢弃, 会发生什么?

如果收到重复数据帧, 收/发双方会怎么做?

为什么 S-W 不存在“数据帧失序”问题