

KCP协议分享

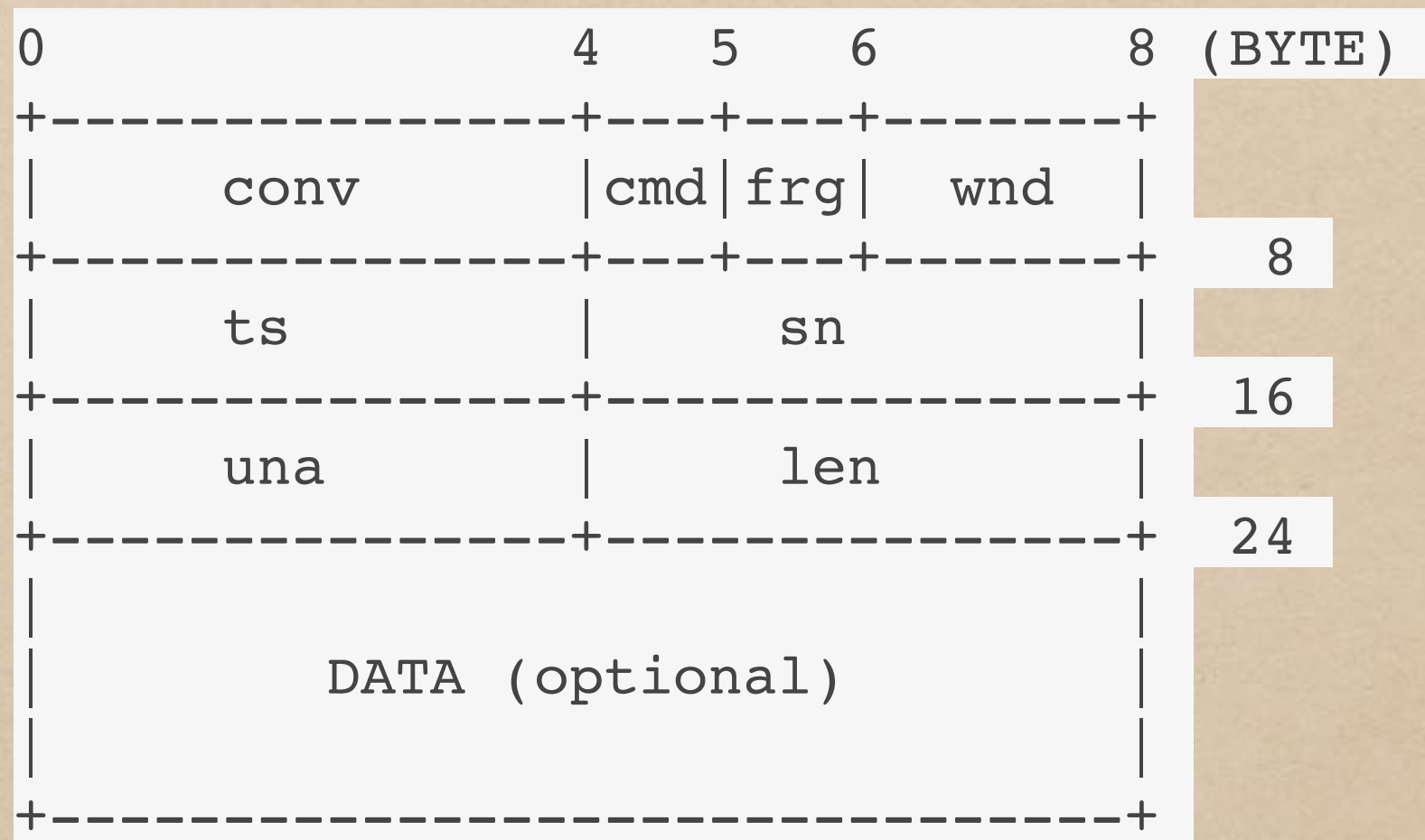
概述

- ◆ KCP协议是一种快速可靠传输ARQ(Automatic Repeat-reQuest)协议
- ◆ 能以比 TCP浪费10%-20%的带宽的代价，换取平均延迟降低 30%-40%
- ◆ 应用场景：游戏、frp、v2ray

网络协议分层模型

+	-----	+
	SESSION	
+	-----	+
	KCP (ARQ)	
+	-----	+
	FEC (OPTIONAL)	
+	-----	+
	CRYPTO (OPTIONAL)	
+	-----	+
	UDP (PACKET)	
+	-----	+
	IP	
+	-----	+
	LINK	
+	-----	+
	PHY	
+	-----	+

协议Header



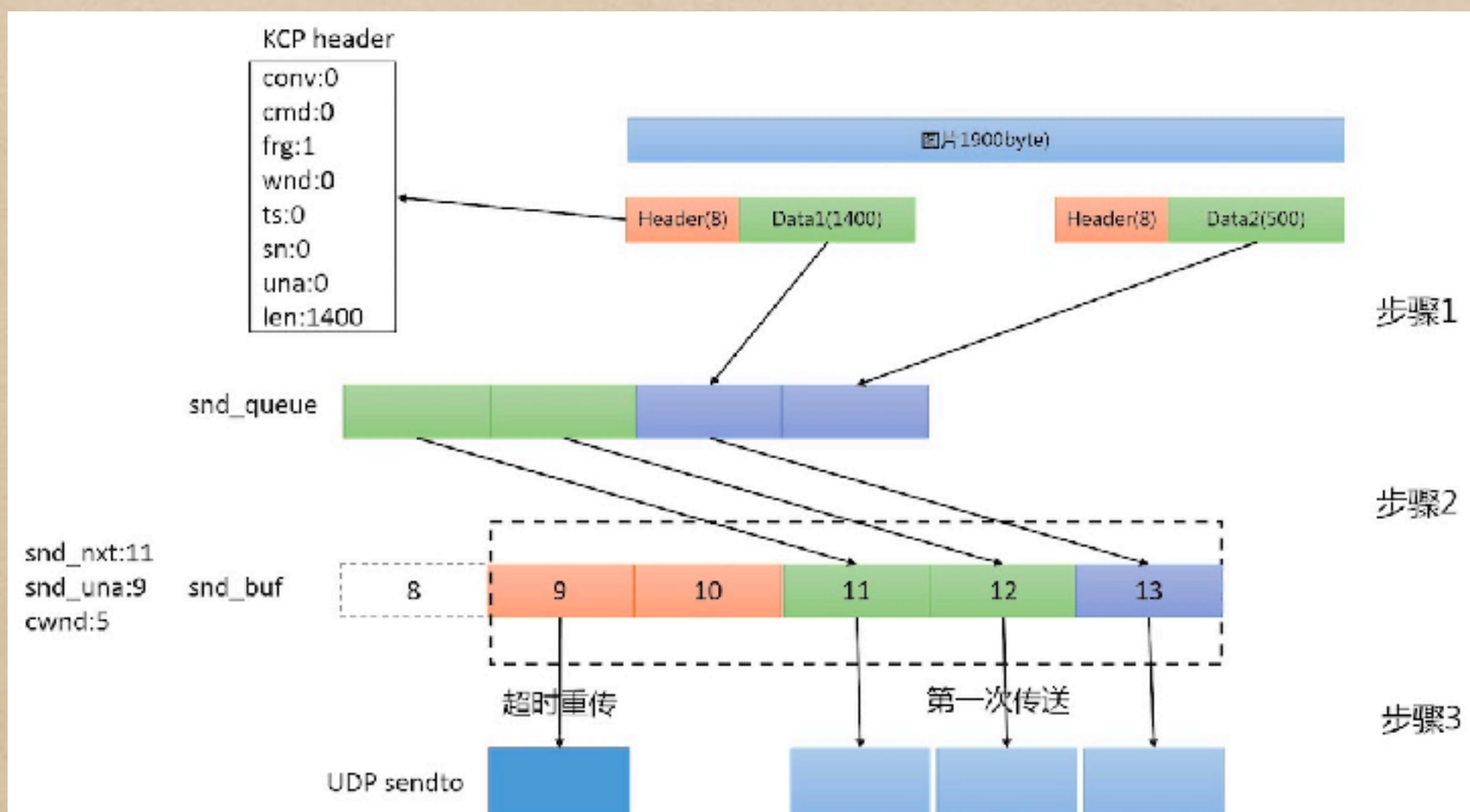
与TCP协议特性对比

协议/功能	TCP	KCP	备注
连接开销	有	无	建立连接-3次握手 关闭连接-4次握手
设置是否延迟Ack	有	有	
设置是否延迟发送数据包	无	有	
拥塞控制	有	有 可关闭	慢启动/拥塞避免/ 拥塞发生/快恢复
快速重传	有	有	
超时重传	有	有 有优化	调整了超时发生后，下次超时的 RTO的翻倍因子
FEC	无	有 可选	前向纠错
加密传输	无	有 可选	对称加密
多路复用	无	有	每个session有独立的滑动接收窗口，消除了TCP队首阻塞问题（和QUIC情况类似）

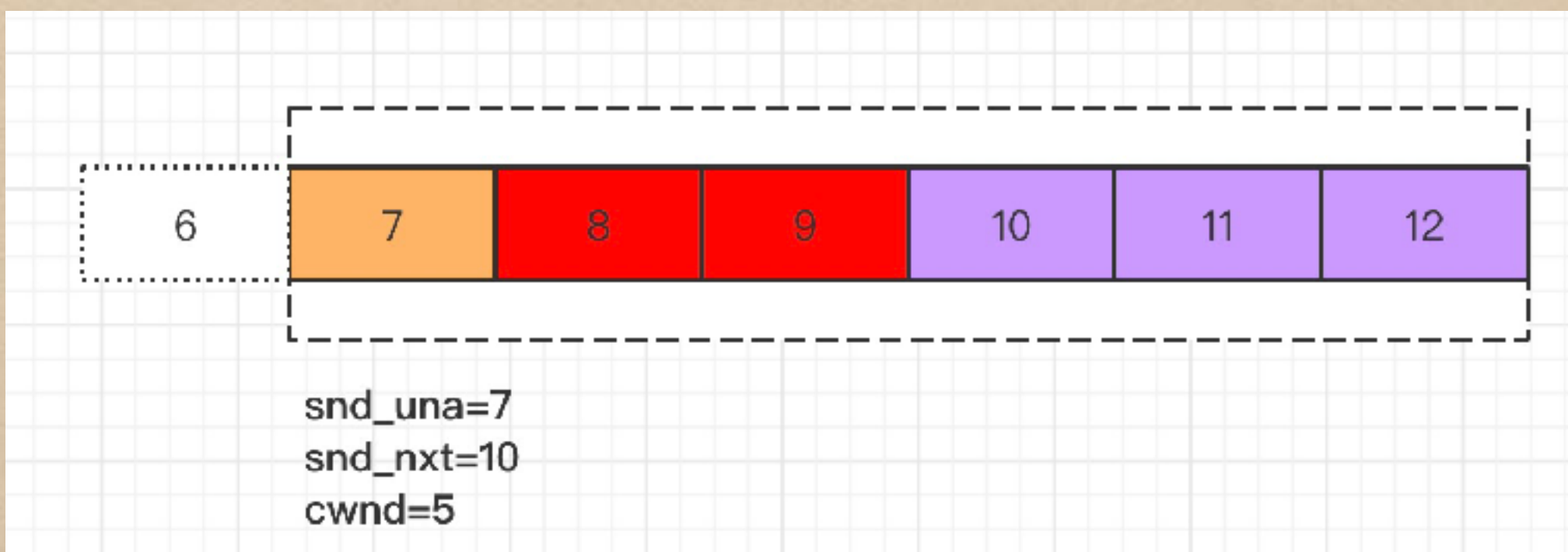
延迟发送

- ◆ 延迟发送Ack
- ◆ 延迟发送数据

延迟发送



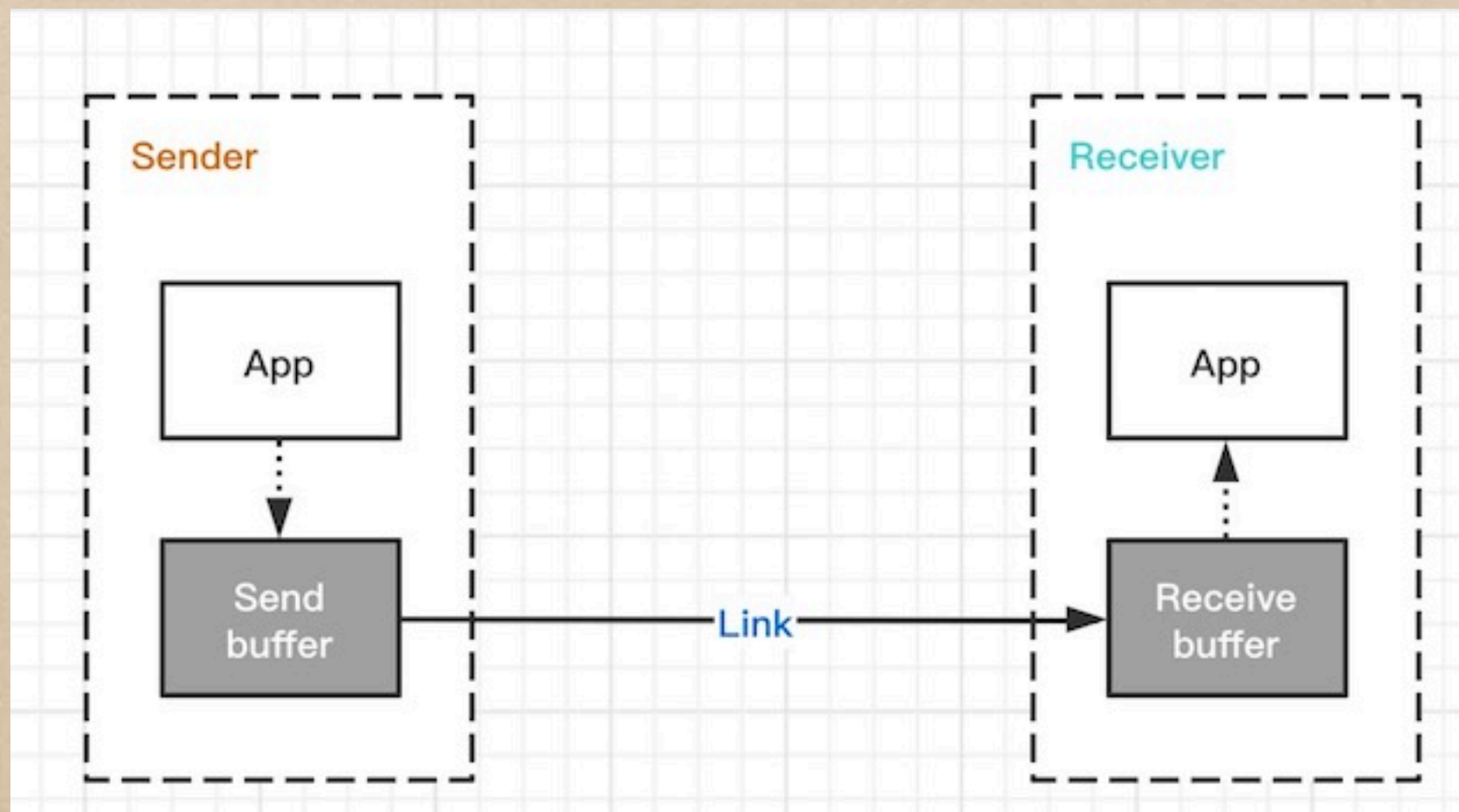
快速重传



流量控制

- ◆ 保护谁?
- ◆ 怎么控?

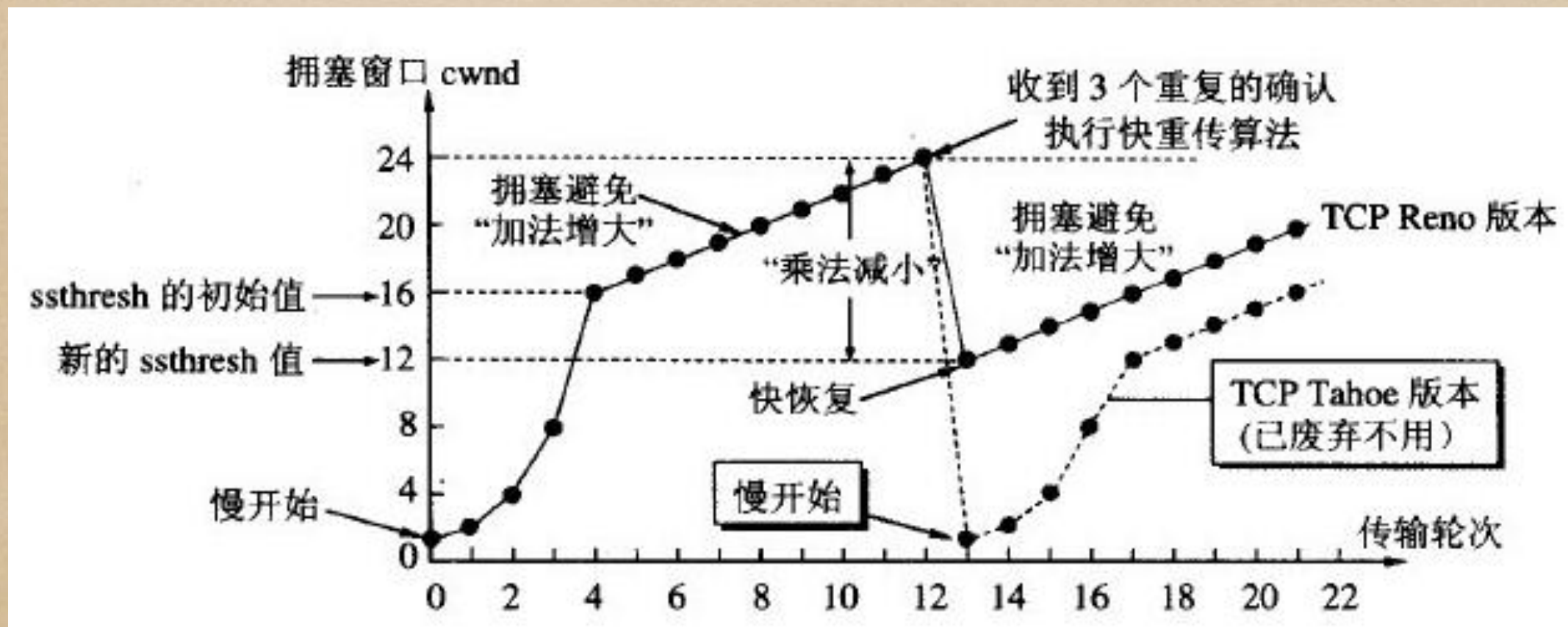
流量控制



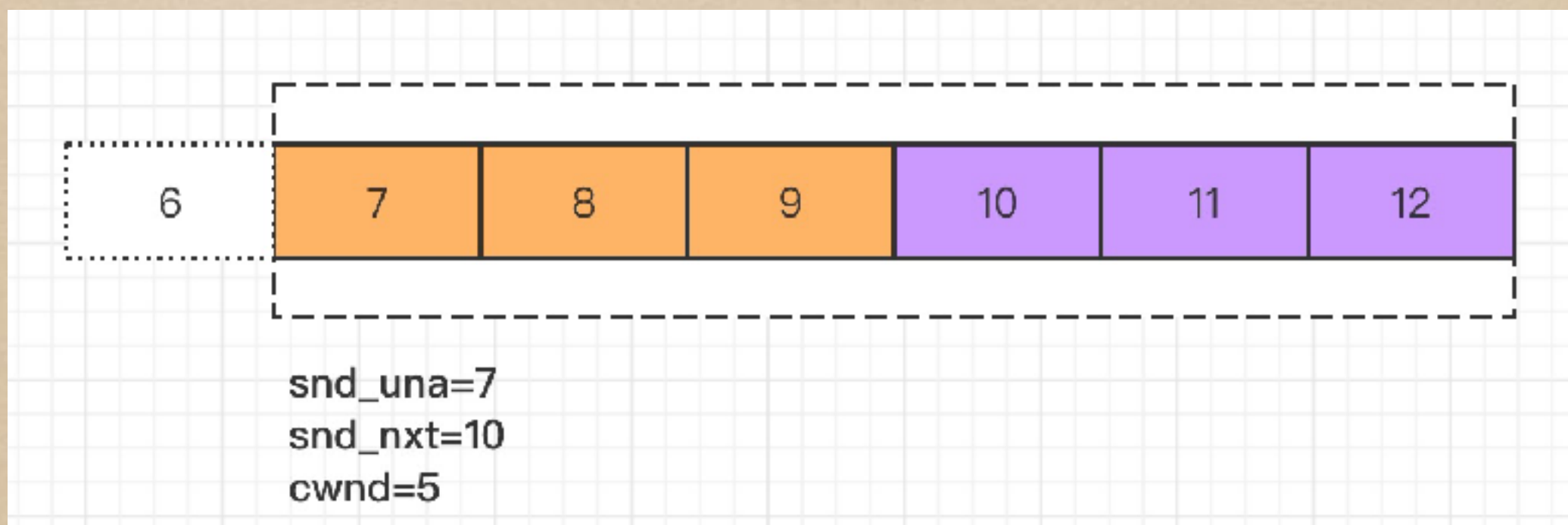
流量控制

- ◆ 发送窗口(snd_wnd)保护本机的发送缓冲区
- ◆ 拥塞窗口(cwnd)来保护发送端与接收端之间的链路
- ◆ 接收窗口(rmt_wnd, 表示接收窗口的空闲大小) 保护接收端的接收缓冲区

流量控制~拥塞窗口



流量控制



FCE(前向纠错)

Reed-Solomon算法

01	00	00	00
00	01	00	00
00	00	01	00
00	00	00	01
1b	1c	12	14
1c	1b	14	12

×

A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	P

=

A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	P
51	52	53	49
55	56	57	25

FCE(前向纠错)

01	00	00	00
00	01	00	00
00	00	01	00
00	00	00	01
1b	1c	12	14
1c	1b	14	12

×

A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	P

=

A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	P
51	52	53	49
55	56	57	25

FCE(前向纠错)

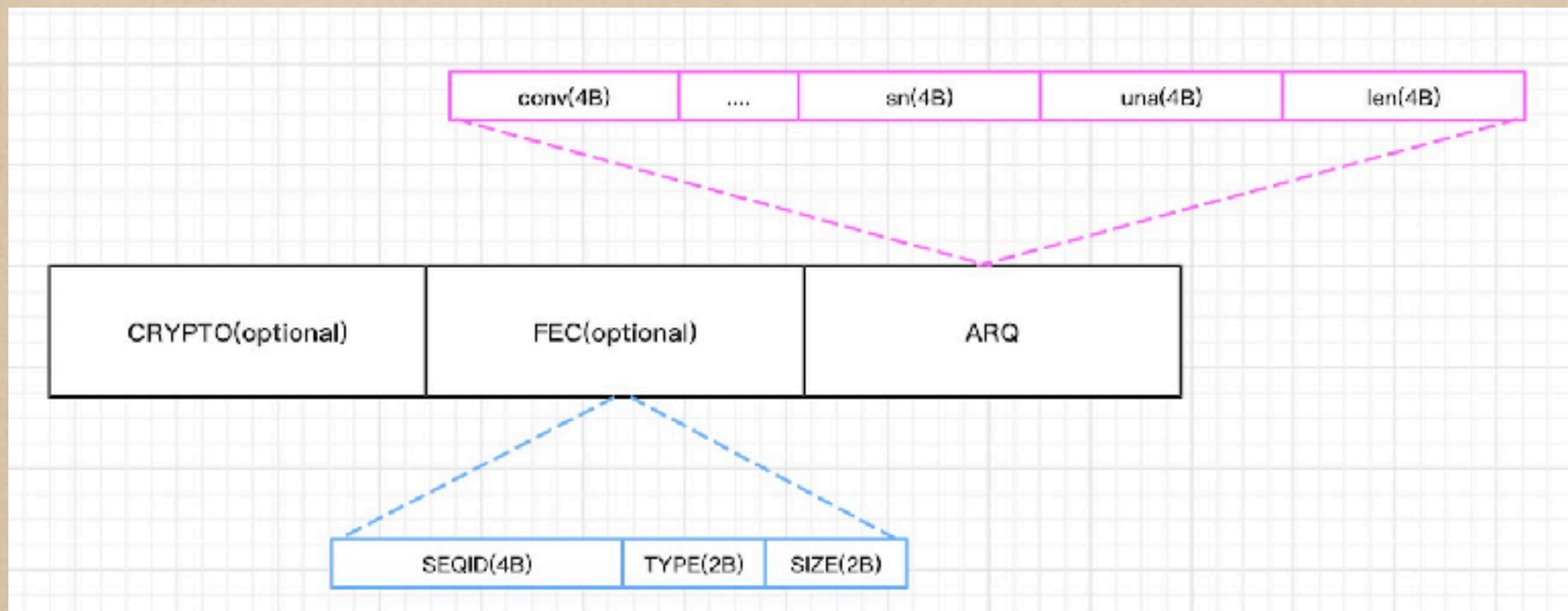
$$\begin{array}{|c|c|c|c|} \hline 01 & 00 & 00 & 00 \\ \hline 00 & 01 & 00 & 00 \\ \hline 8d & f6 & 7b & 01 \\ \hline f6 & 8d & 01 & 7b \\ \hline \end{array} \times \begin{array}{|c|c|c|c|} \hline 01 & 00 & 00 & 00 \\ \hline 00 & 01 & 00 & 00 \\ \hline 1b & 1c & 12 & 14 \\ \hline 1c & 1b & 14 & 12 \\ \hline \end{array} \times \begin{array}{|c|c|c|c|} \hline A & B & C & D \\ \hline E & F & G & H \\ \hline I & J & K & L \\ \hline M & N & O & P \\ \hline \end{array} = \begin{array}{|c|c|c|c|} \hline 01 & 00 & 00 & 00 \\ \hline 00 & 01 & 00 & 00 \\ \hline 8d & f6 & 7b & 01 \\ \hline f6 & 8d & 01 & 7b \\ \hline \end{array} \times \begin{array}{|c|c|c|c|} \hline A & B & C & D \\ \hline E & F & G & H \\ \hline 51 & 52 & 53 & 49 \\ \hline 55 & 56 & 57 & 25 \\ \hline \end{array}$$

~~$$\begin{array}{|c|c|c|c|} \hline 01 & 00 & 00 & 00 \\ \hline 00 & 01 & 00 & 00 \\ \hline 8d & f6 & 7b & 01 \\ \hline f6 & 8d & 01 & 7b \\ \hline \end{array} \times \begin{array}{|c|c|c|c|} \hline 01 & 00 & 00 & 00 \\ \hline 00 & 01 & 00 & 00 \\ \hline 1b & 1c & 12 & 14 \\ \hline 1c & 1b & 14 & 12 \\ \hline \end{array} \times \begin{array}{|c|c|c|c|} \hline A & B & C & D \\ \hline E & F & G & H \\ \hline I & J & K & L \\ \hline M & N & O & P \\ \hline \end{array} = \begin{array}{|c|c|c|c|} \hline 01 & 00 & 00 & 00 \\ \hline 00 & 01 & 00 & 00 \\ \hline 8d & f6 & 7b & 01 \\ \hline f6 & 8d & 01 & 7b \\ \hline \end{array} \times \begin{array}{|c|c|c|c|} \hline A & B & C & D \\ \hline E & F & G & H \\ \hline 51 & 52 & 53 & 49 \\ \hline 55 & 56 & 57 & 25 \\ \hline \end{array}$$~~

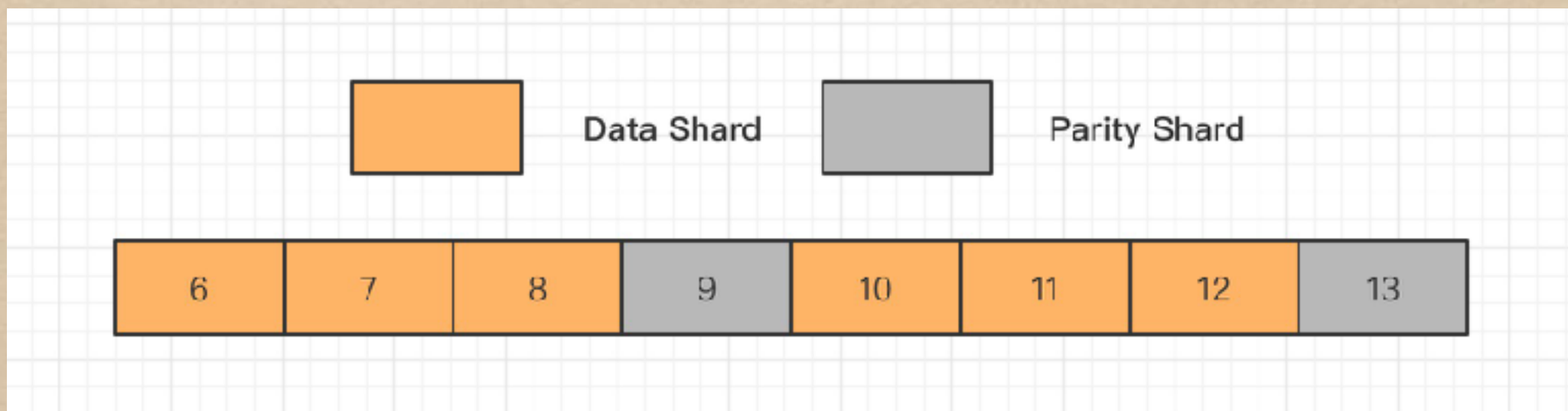
FCE(前向纠错)

- ◆ 包含有 n 个symbol的消息可以通过reed-solomon算法, 得到包含 $n + k$ 个symbol的新消息。任意丢失至多 k 个symbol, 原始消息仍然能够被恢复出来。
- ◆ DVD、CD、二维码
- ◆ 远距离空间传输
- ◆ 存储系统, 比如 minio

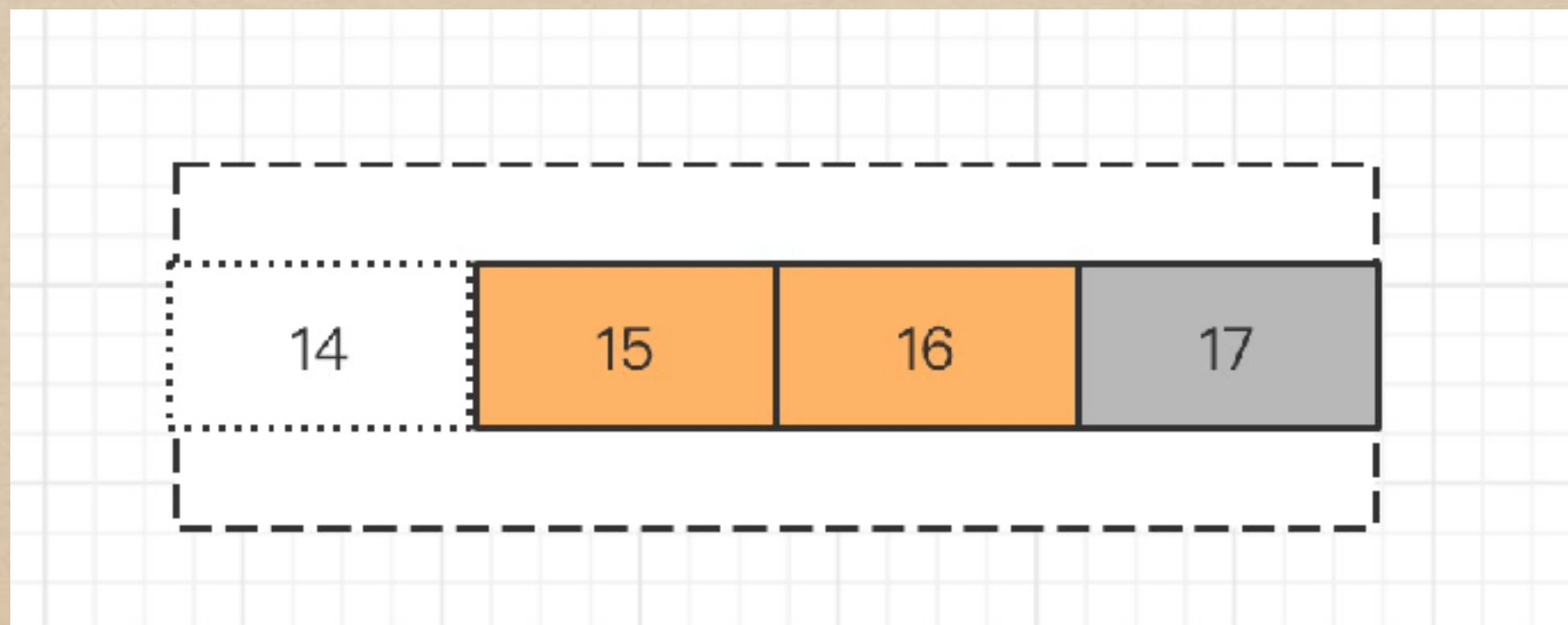
FCE(前向纠错)



FCE(前向纠错)



FCE(前向纠错)



End

◆ Q&A