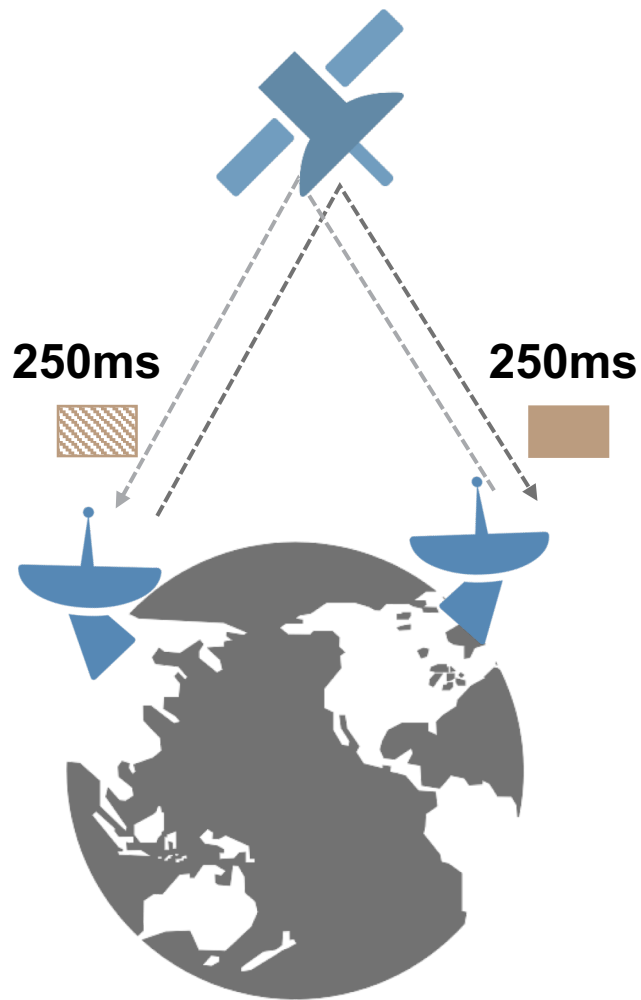


滑动窗口控制机制



长距离 vs. 长报文



假设：在如图所示的卫星通信系统中，采用了停等式流量控制。

- 数据帧长1000b
- 发送速率为50kps
- 传播时延为250ms

- $t = 0$ 开始发送；
- $t = 1000/50000 = 20\text{ms}$ 后发送方发完第一帧
- $t = 20 + 250 = 270\text{ms}$ 后接收方接收完第一帧
- $t = 270 + 250 = 520\text{ms}$ 发送方收到第一帧的确认

$$\text{信道利用率} = 20/520 = 3.8\%$$

 确认帧 (ACK)

 数据帧

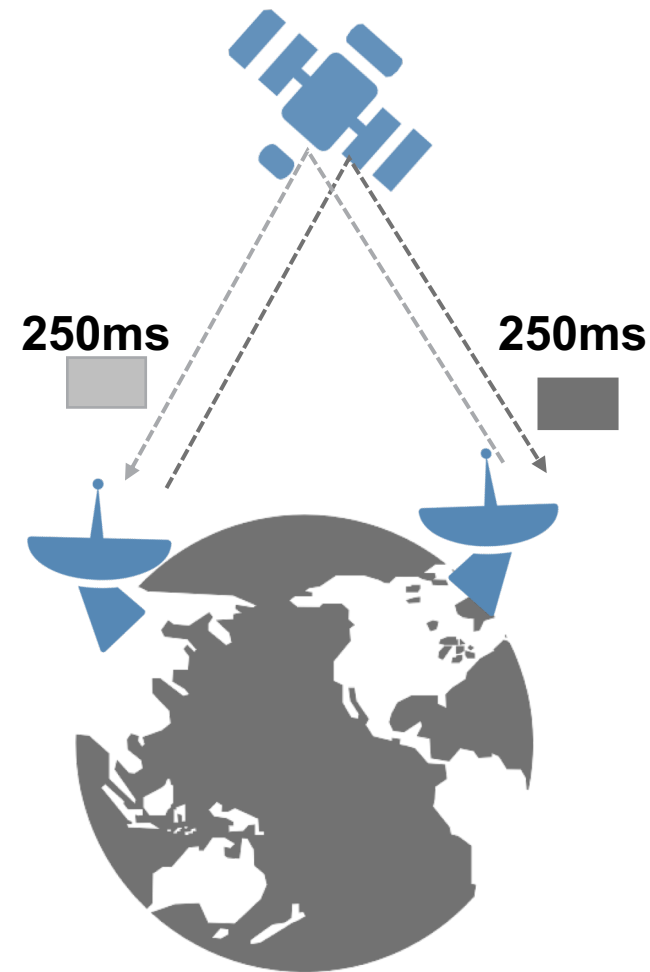


如何提高信道利用率

结论：使用停等式流量控制，高延迟的卫星信道信道利用率仅3.8%

原因	解决方法
信道时延使得发送方等待时间过长	利用传播延迟连续发送n帧(连续式)

$$n = 1 + \frac{\text{传播延迟}(500)}{\text{发送一帧所需时间}(20)} = 26$$



 确认帧 (ACK)

 数据帧



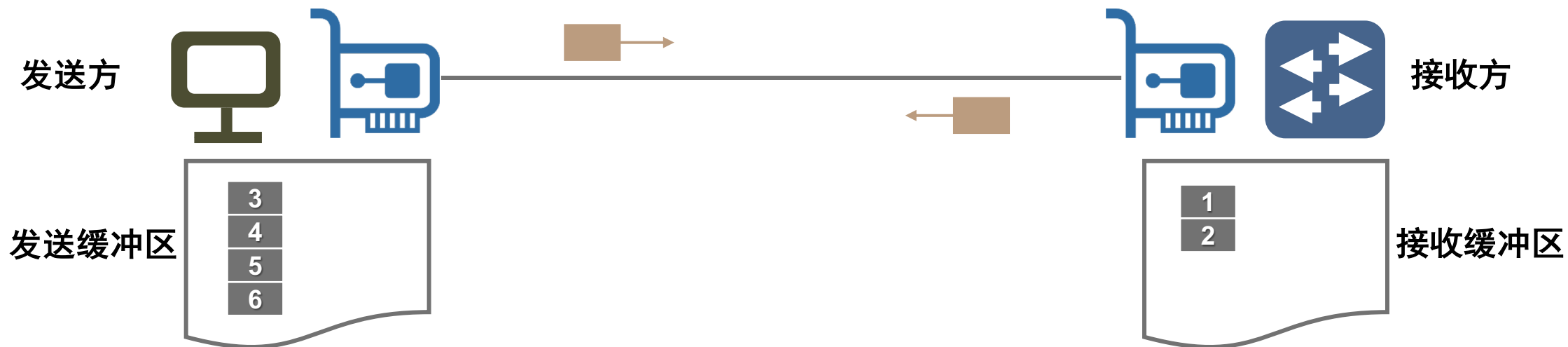
滑动窗口流量控制

滑动窗口控制：利用窗口控制连续发送的数据量。

接收方收到一个帧后发回一个确认，该确认帧包括了接收方想要的下一个帧号。

滑动窗口控制基本条件

- 两个节点通过全双工链路连接
- 收发双方为n个帧分配缓冲区
- 每个数据帧拥有一个序号



 确认帧 (ACK)

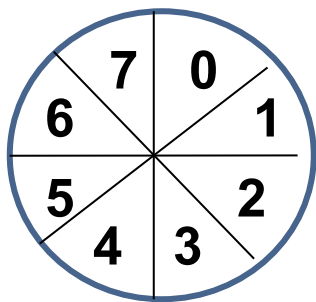
 数据帧



序号以及序号空间特性

序号空间：帧序号的取值范围。通常协议规定了帧格式中表示序号的二进制比特数

例如：序号用二进制 n 位表示，则取值范围是0、1、2、 \dots 、 2^n-1 。



序号空间循环特性

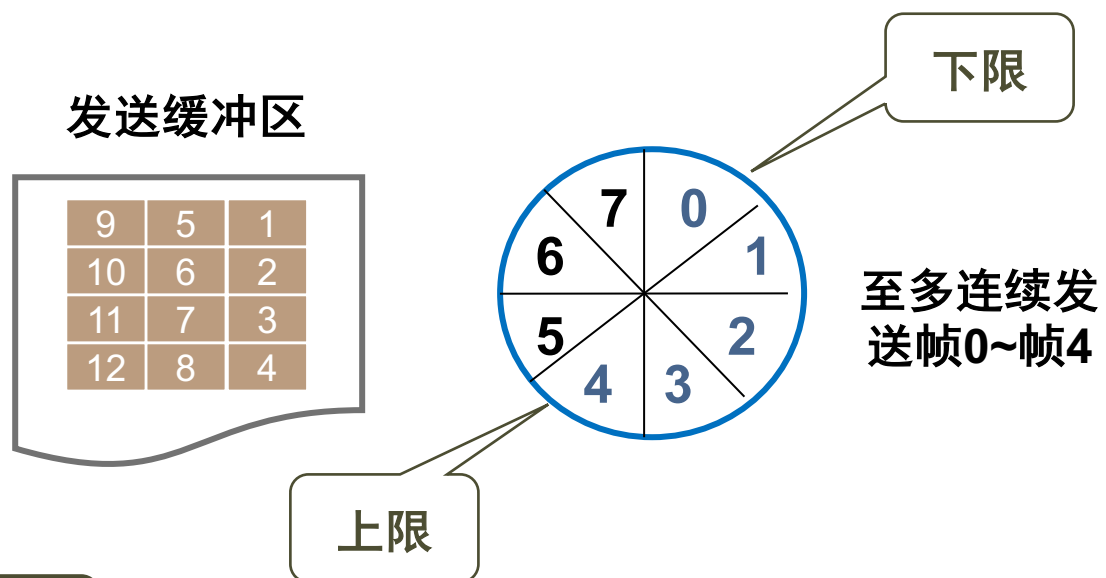
- 被发送的帧按顺序编号
- 当用尽最大序号后下一个帧的序号回滚到0

- 如果序号用二进制3个比特表示，则序号取值范围是0, 1, 2, 3, 4, 5, 6, 7,
- 当最近发出帧的编号为7时，下一个待发送帧的编号为0.

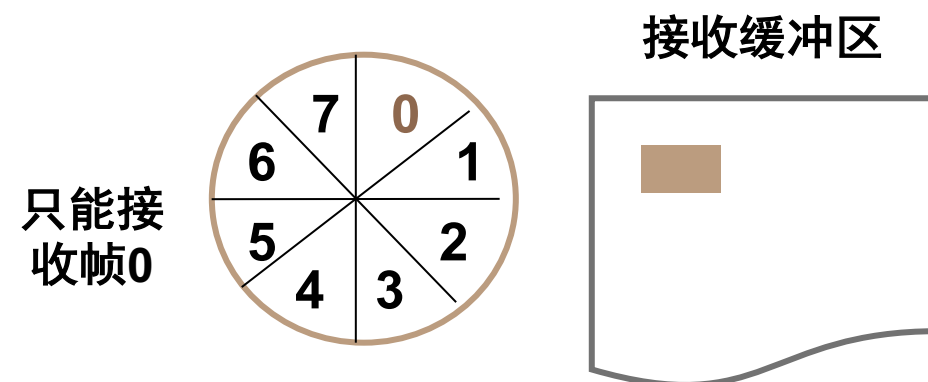


发送窗口 vs. 接收窗口

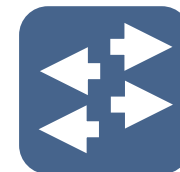
发送窗口(W_T)：给出允许发送方连续发送的帧序号。



接收窗口(W_R)：给出允许接收方接收的帧序号。



- 发送窗口和接收窗口可以不一样大
- 发送窗口和接收窗口沿顺时针方向滑动
- 窗口不是输出/输入缓冲区，只存放待发送/接收帧的序号



滑动窗口的控制原理

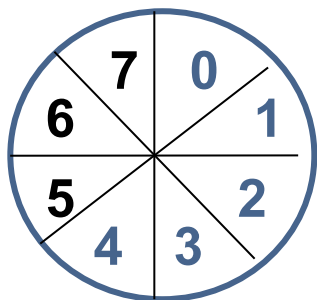
假设：发送窗口 $W_T = 5$ ；接收窗口 $W_R = 1$



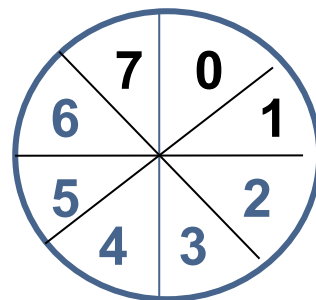
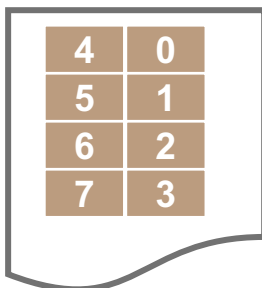
- 发送方：可以连续发送5个帧
- 接收方：只能一个帧一个帧地接收

发送窗口

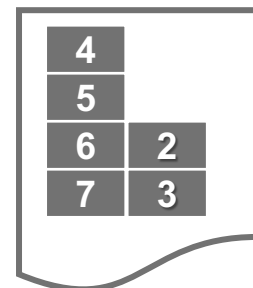
t_0 : 连续发帧0~4



发送缓冲区



发送缓冲区

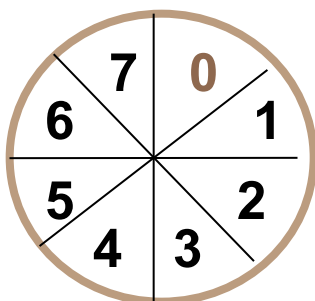


t_i : 收到帧1确认

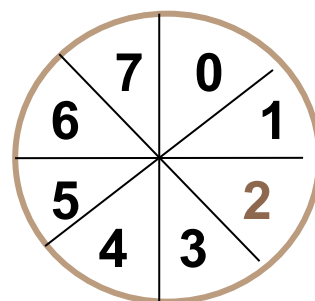
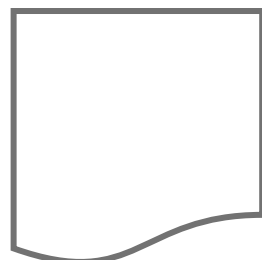
时间

接收窗口

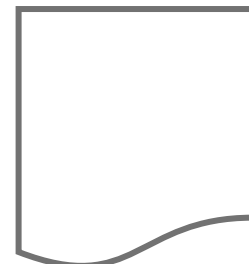
t_0 : 准备接收帧0



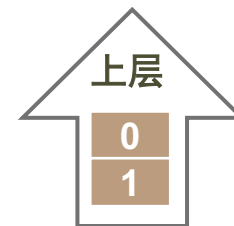
接收缓冲区



接收缓冲区



t_i : 准备接收帧2



数据帧

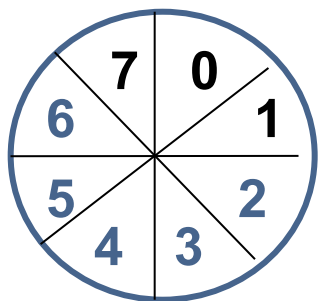


北京大学

滑动窗口的控制原理

窗口操作：在连续发送的情况下，利用模数运算，让序号循环地被使用，但要在收发两端进行适当的控制。

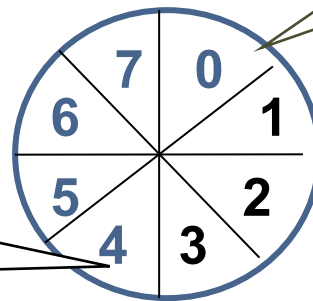
发送窗口



t_i : 收到帧1确认

t_{i+1} : 收到帧3确认

发出两个帧窗口下限前移2个序号

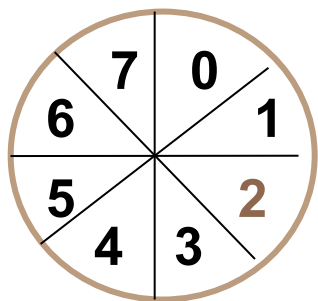
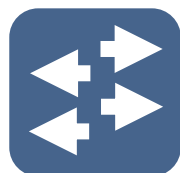


收到两个确认窗口上限前移2个序号

维持发送窗口大小为5

→时间

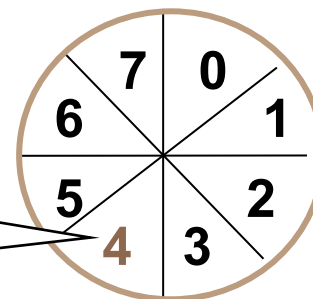
接收窗口



t_i : 准备接收帧2

t_{i+1} : 准备接收帧4

发出一个确认窗口前移1个序号



将收到的帧交给上层后再接收下一个帧



北京大学

滑动窗口基本原理

基本原理

- 发送方和接收方的滑动窗口控制发送和接收
- 发送窗口和接收窗口在序号上滑动

发送方

发送方只能发送帧号落在发送窗口内的帧，收到确认后，将发送窗口向前推进一格

接收方

接收方每收到一个帧，校验正确且序号落在接收窗口内，就向前推进一格，并发出确认帧

