数据通信

数据链路控制协议

殷亚凤
yafeng@nju.edu.cn
http://cs.nju.edu.cn/yafeng/
Room 901, Building of CS

数据链路控制协议

- 发送点和接收点为了实现有效数据通信，存在以下一些要求和目标：
  - **帧同步**：数据以数据块的形式发送，这些数据块称为帧。每个帧的开始和结束必须可以辨别。
  - **流量控制**：发送帧的速度不得超过接收站点接纳的速度。
  - **差错控制**：由传输系统引起的比特差错必须纠正。
  - **寻址**：传输涉及的两个站点的身份必须指明。
  - **控制信息和数据在同一链路上**：接收器必须能够从传输的数据中辨认出控制信息。
  - **链路管理**：数据交换的初始化、维持以及终止等工作，需要站点间协同与合作，因而需要具有管理这些交换的过程。

- **数据链路控制**

数据链路控制协议

1. 流量控制

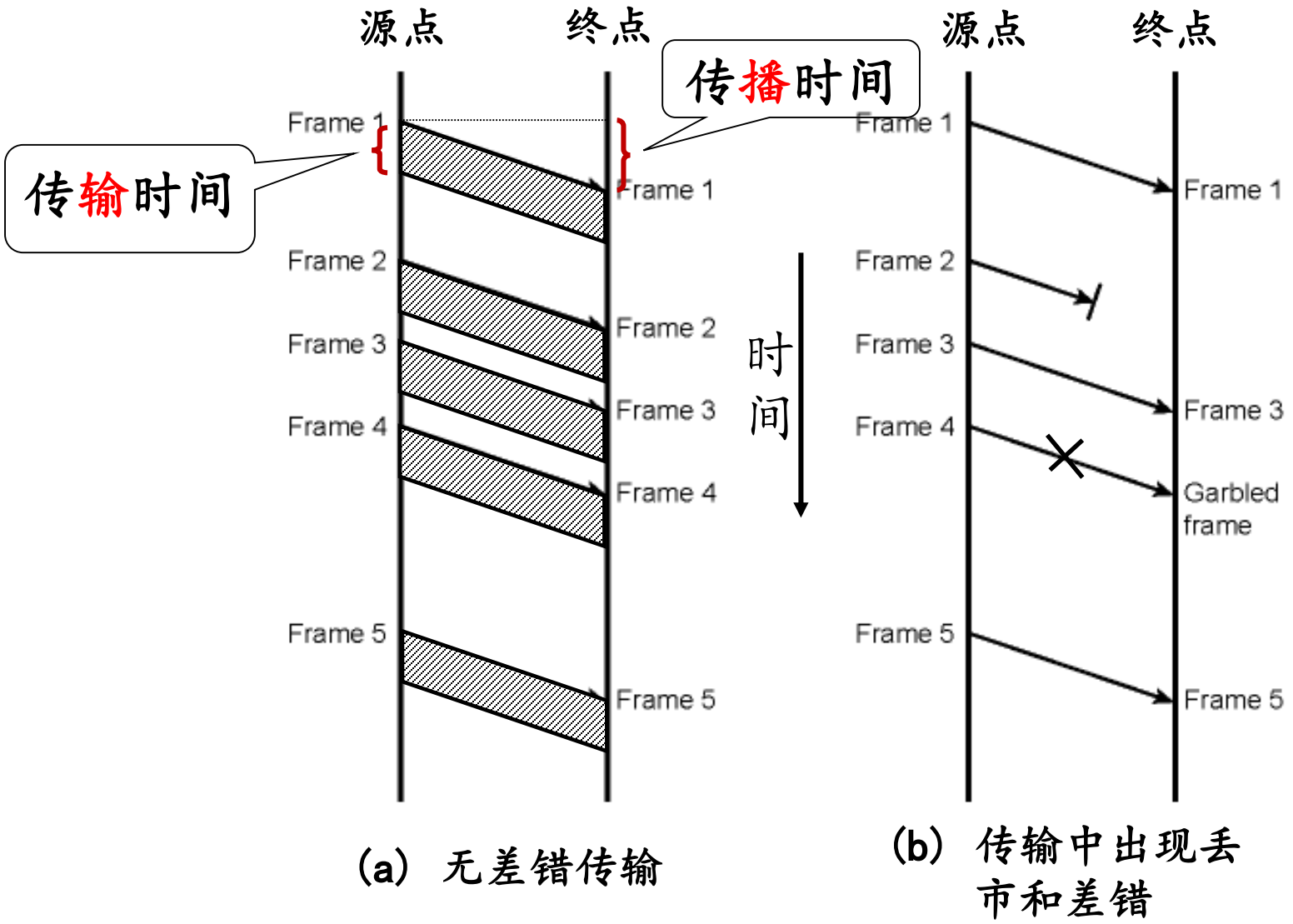
2. 差错控制

3. 高级数据链路控制

流量控制

- **流量控制**是用于确保发送的数据不会超出接收实体接收数据能力的技术
  - 接收端需要一段时间来处理收到的数据
  - 如果发送端发送快于接收端接收速度，会出现缓存溢出

帧传输模型



(a) 无差错传输

(b) 传输中出现丢市和差错

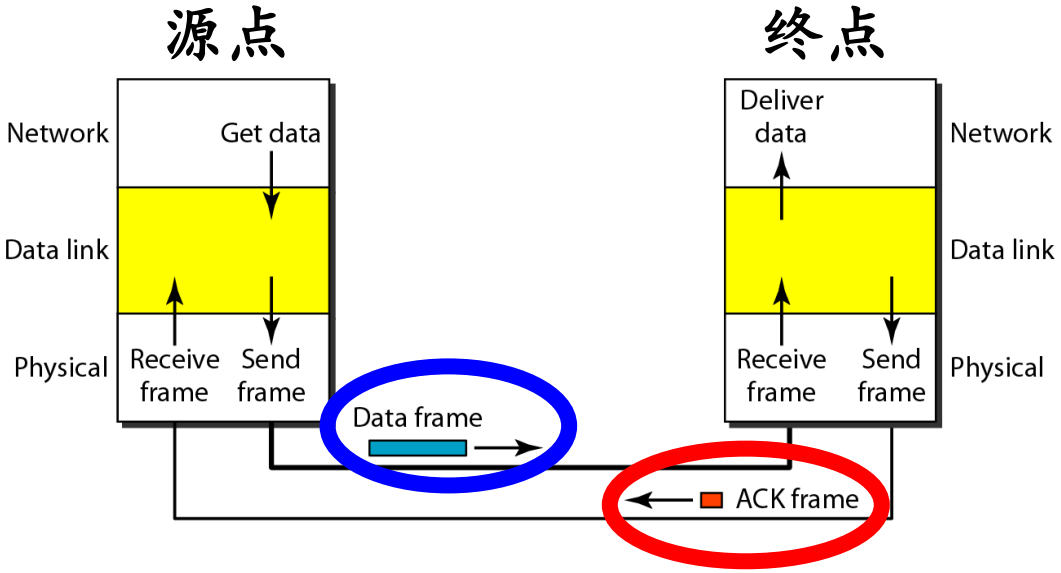
停止等待流量控制

源实体传输一个帧

目的实体接收到帧之后，返回一个确认

源点在发送下一帧之前必须等待确认

终点可以不发送确认，从而终止数据流



停止等待流量控制

- 把大块的**数据切分成小数据块**传输：
  - 接收方缓存空间有限
  - 大块数据容易发生错误，出现错误时重传小块数据更容易
  - 避免一个站点长时间占用传输媒体

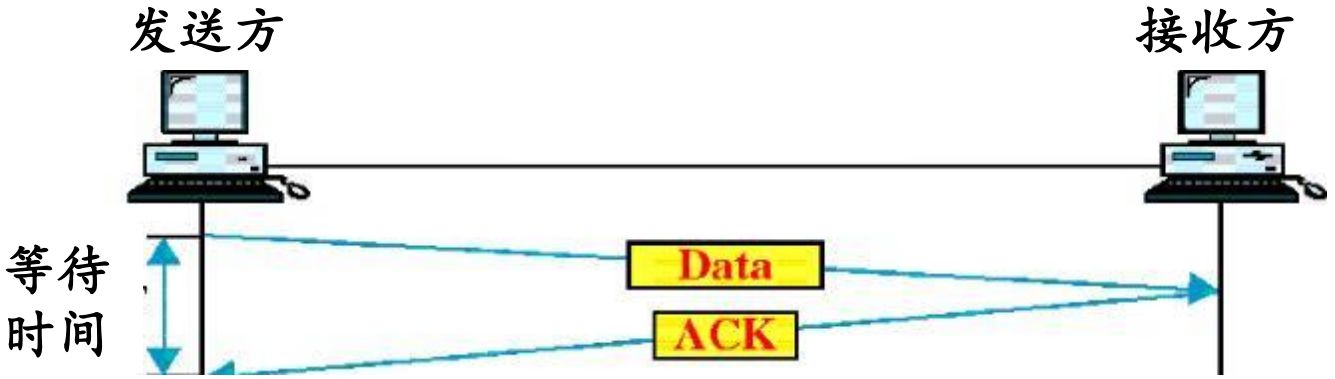
- **链路的比特长度** $B = R \times \frac{d}{v}$

  R是链路的数据率bps，d是链路长度m，V是传播速度m/s

- 传输时间取1时，**传播时延/时间** $a = \frac{B}{L}$

  - L是一个帧中的比特数(以比特为单位的帧长度)
  - a是传播时间与传输时间的比值

停止等待流量控制



$$a = \frac{t_{prop}}{t_{frame}} = \frac{\text{传播时间}}{\text{传输时间}} = \frac{d/V}{L/R} = \frac{\mathrm{Rd}}{VL} = \frac{B}{L}$$

R = 数据率（bps）

d = 链路的长度/距离（m）

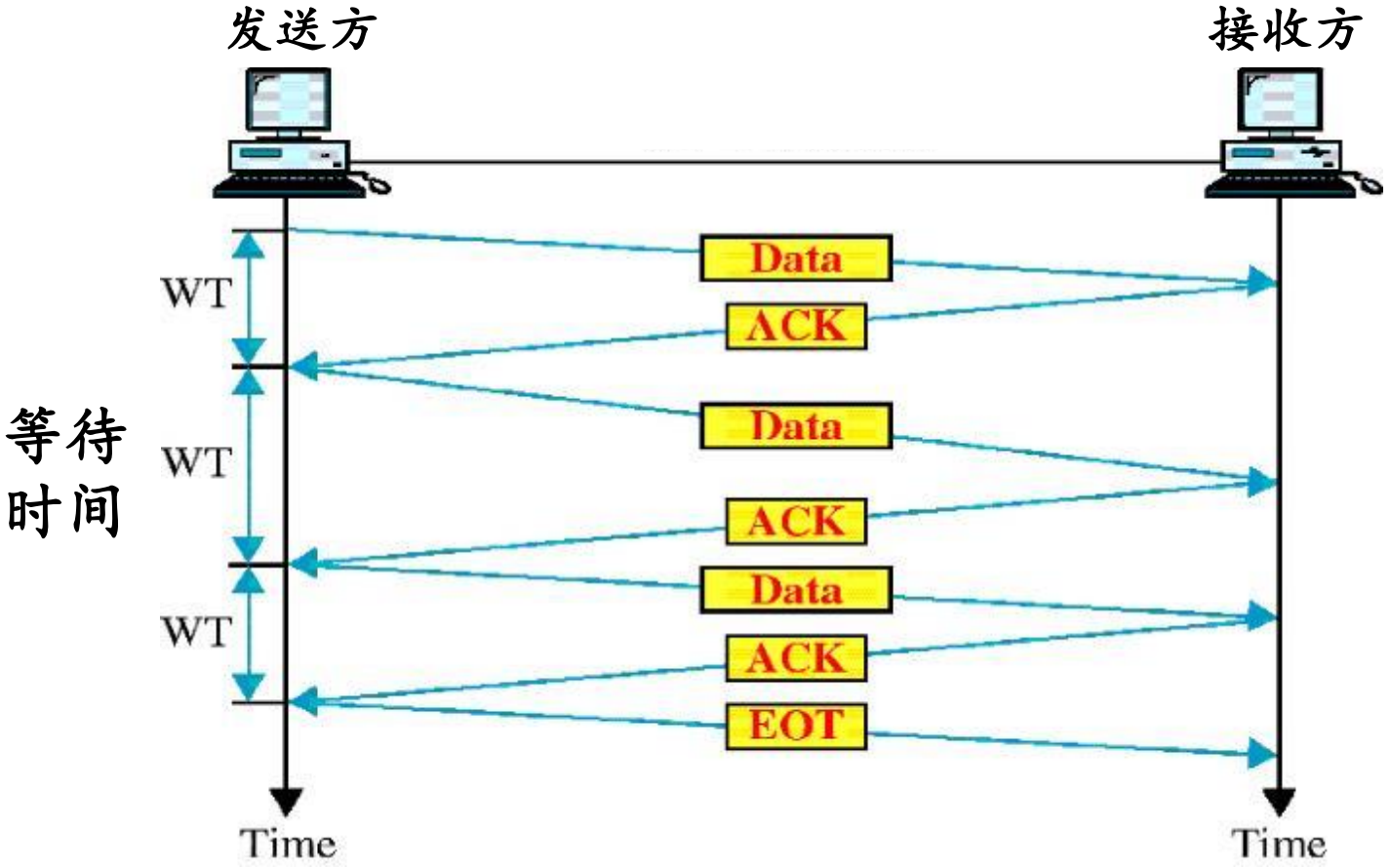
V = 传播速度（m/s）

B = 以比特为单位的链路长度（bits）

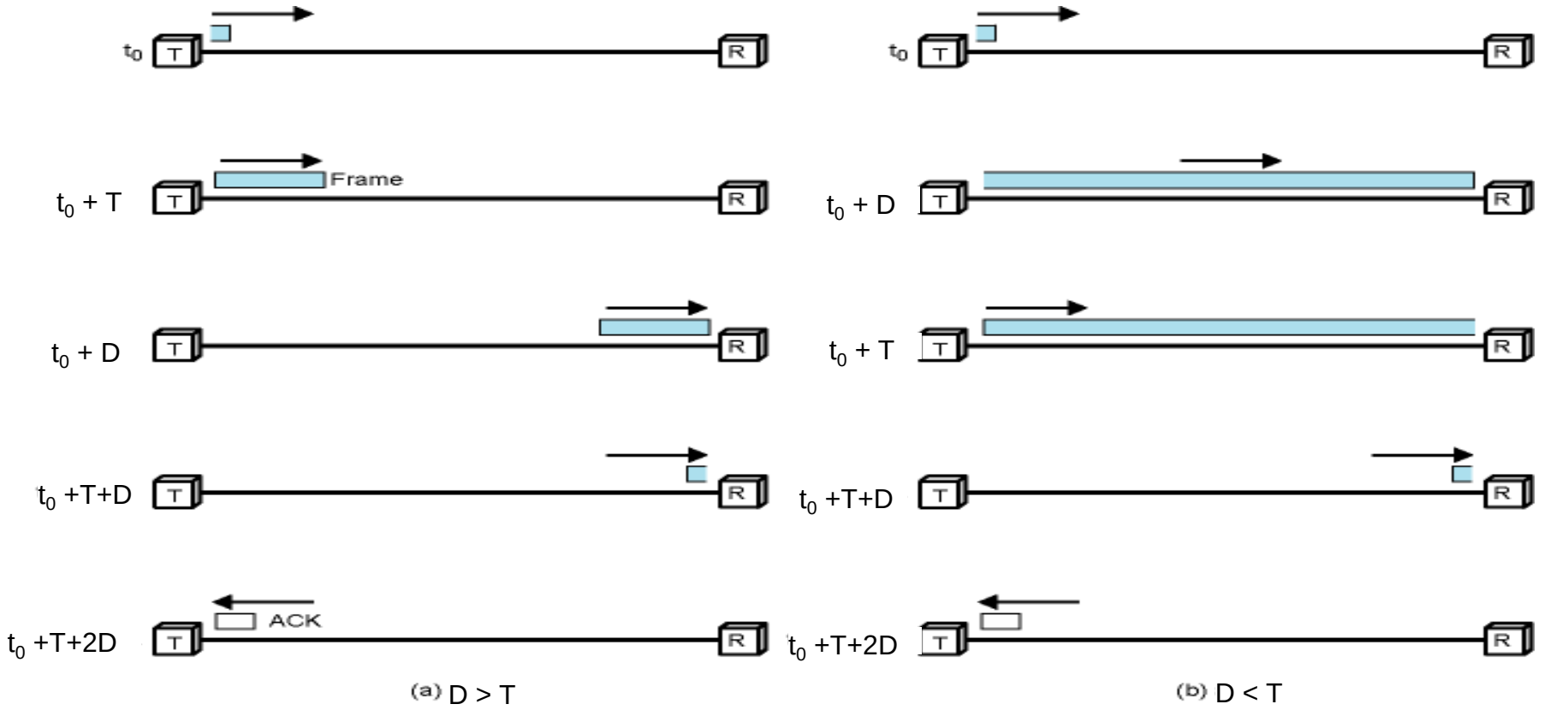
L = 以比特为单位的帧长度（bits）

停止等待流量控制

- 停等协议链路利用率在哪些情况下较高？
  - **传输**时间大于**传播**时间
  - **帧长**大于**链路的比特长度**



链路利用率



(a>1)　传播时间= D, 传输时间 = T　(a<1)

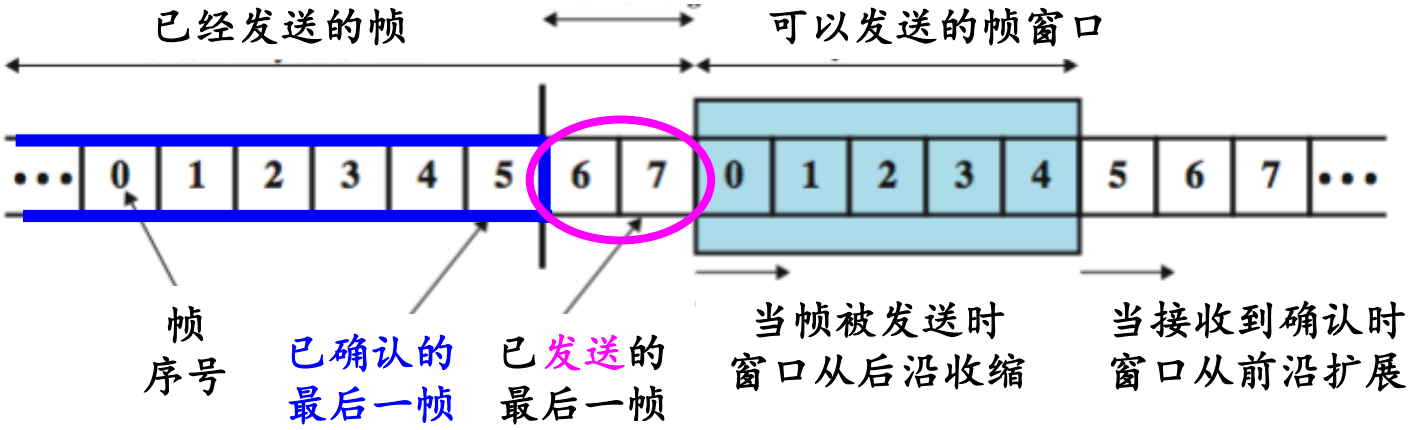
$$U = \frac{t_{frame}}{T_F} = \frac{t_{frame}}{2t_{prop} + t_{frame}} = \frac{1}{1+2a}$$

滑动窗口流量控制

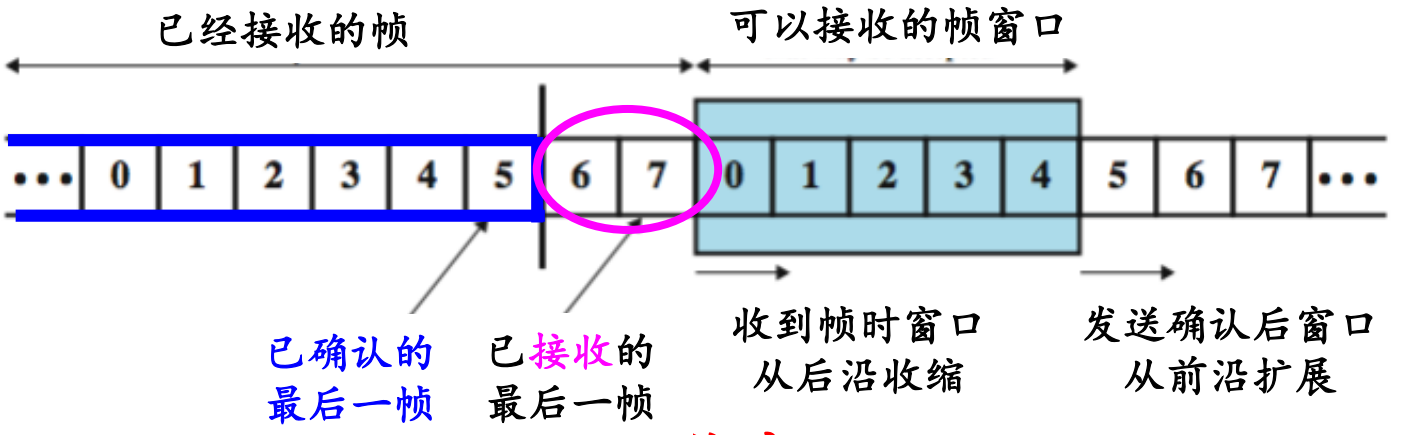
- 停止等待流量控制协议不能够同时发送多个帧
- **滑窗协议允许一次发送多个帧**
  - 接收端缓存大小W
  - 发送端在没有收到ACK前可以发送 W 个帧
  - 每个帧通过序号来标识
    - 序号大小受字段长度限制 (k bits)
    - 帧以 2^k 为模编号 (0 … 2^k-1)
  - ACK 包含下一个期望收到的帧编号

滑动窗口描述

帧被缓存直至
收到确认信息

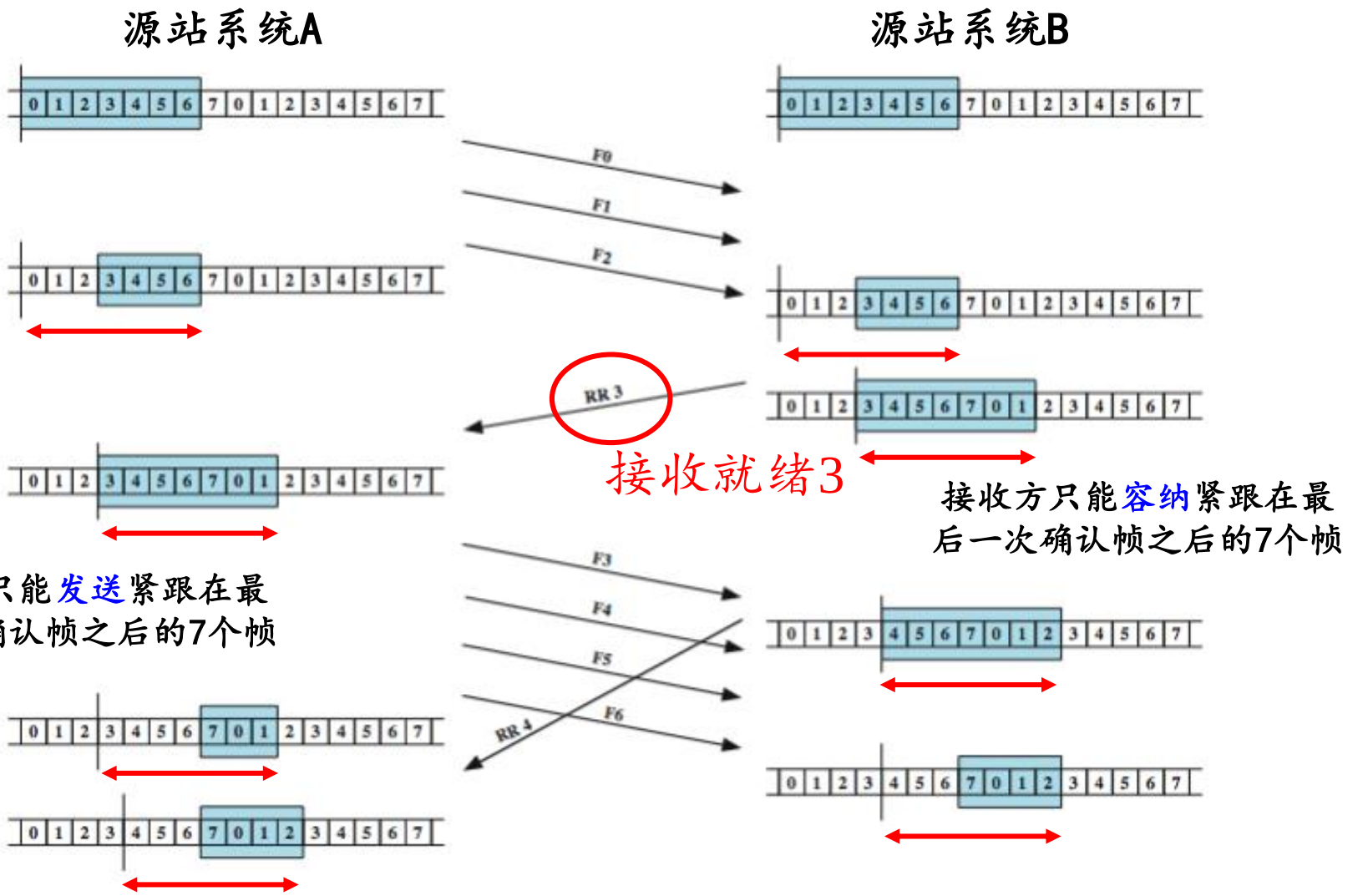


发送方



接收方

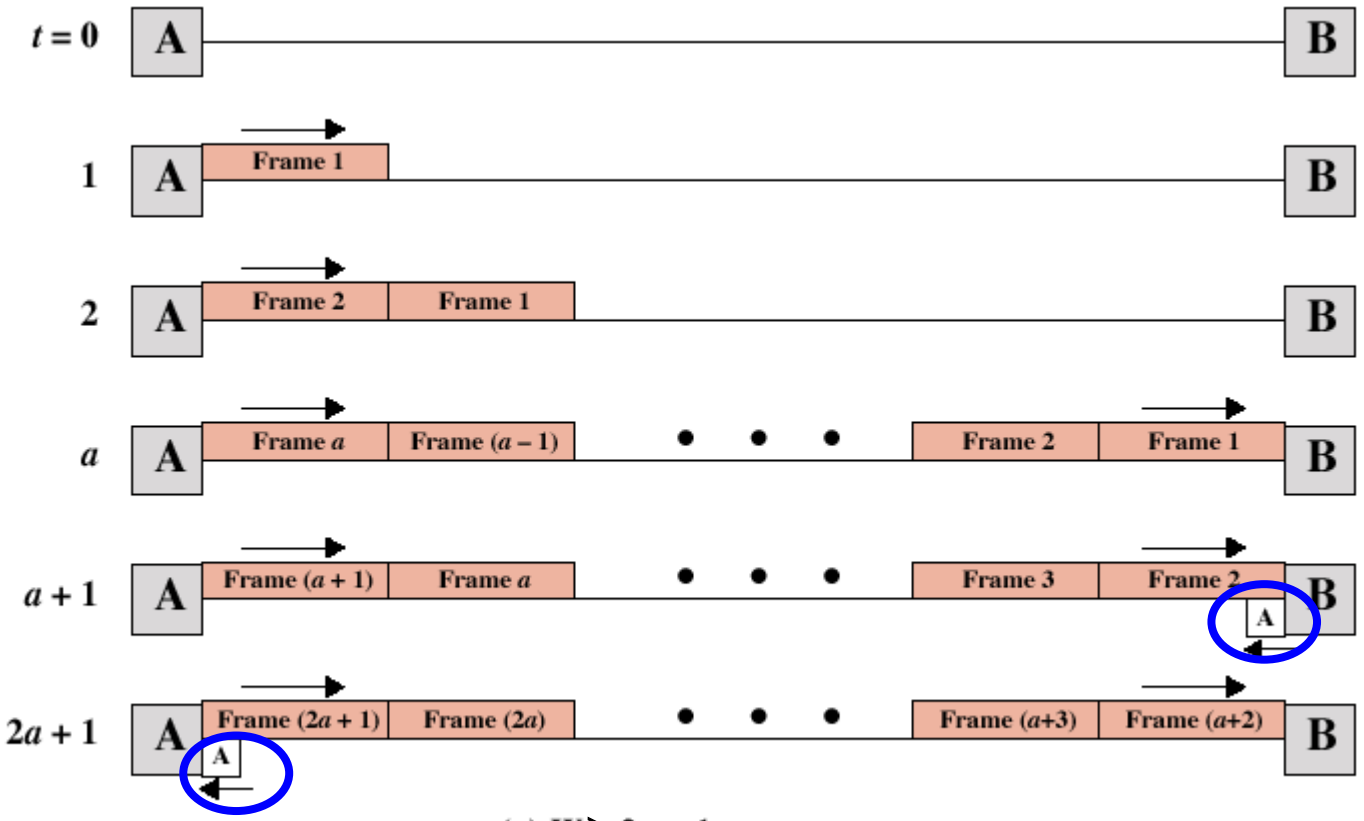
滑动窗口协议示例


源站系统A
源站系统B
F0
F1
F2
RR 3
接收就绪3
接收方只能容纳紧跟在最后一次确认帧之后的7个帧
发送方只能发送紧跟在最后一次确认帧之后的7个帧
F3
F4
F5
F6
RR 4
窗口大小为7(2³−1)

滑动窗口协议优化

- 接收端允许发送“*Receive Not Ready（**RNR**）*”报文来**切断对方的帧流**
- 之后，接收端必须通过一个**正常的确认帧来重启滑动窗口**
- 如果是**双向链路**，可以使用**捎带*“piggybacking”***
  - 一个帧包含**发送数据**和**ACK**
  - 如果没有数据发送，发送独立的确认帧
  - 如果需要发送数据，但是没有新的确认，则重新发送上一次已经发送过的确认

滑动窗口的性能

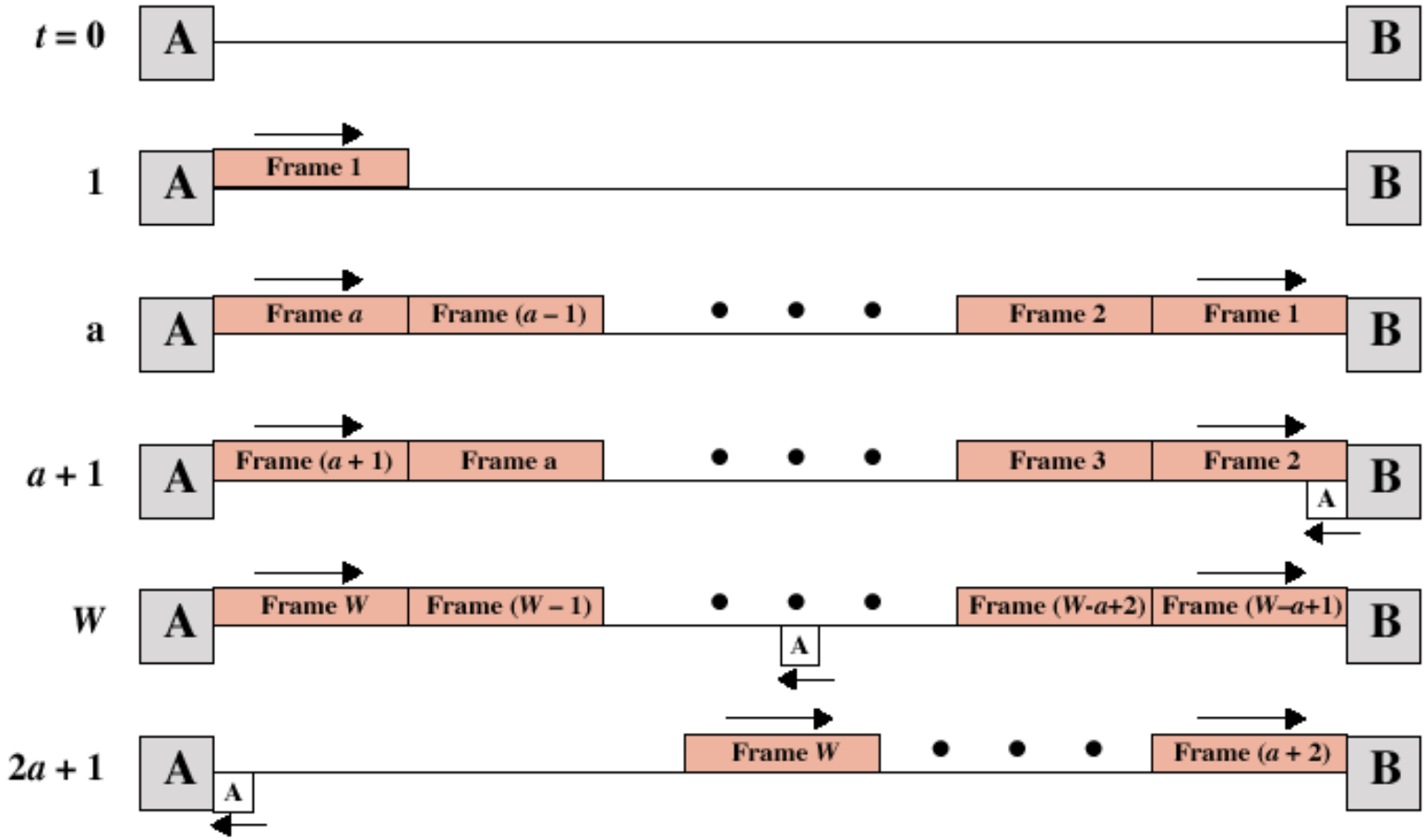


(a) $W \geq 2a + 1$

W为发送方/接收方窗口大小，a为链路长度

链路利用率：$U = 1$

滑窗的性能



(b) $W < 2a + 1$

W为发送方/接收方窗口大小，a为链路长度

链路利用率： $U = W / (2a + 1)$

数据链路控制协议

1. 流量控制

2. 差错控制

3. 高级数据链路控制

差错控制

- **差错控制**指的是用于检测和纠正帧传输过程中出现差错的机制。

- **主要针对以下两种类型的差错：**

  - **帧丢失**
    - 帧没有达到另一方

  - **帧损伤**
    - 帧到达，但是一些比特有差错

常用的差错控制技术

- **差错检测：**利用差错检测技术检测到某些帧出了错，并丢弃这些帧。
- **肯定确认：**终点为成功接收到的无差错的帧返回一个肯定确认。
- **超时重传：**在预定时间没有收到确认的情况下，源点会重新传输一个帧。
- **否定与重传：**终点为检测到差错的帧返回一个否认，源点重新传输这些帧。

→ 这些机制都称为**自动重传请求**（ARQ）。

自动重传请求

- **停止等待ARQ （stop-and-wait ARQ）**
  - 基于停止等待流量控制技术
- **返回N ARQ (go-back-N ARQ)**
  - 基于滑动窗口流量控制技术
- **选择拒绝 ARQ (selective-reject ARQ)**

停止等待ARQ

- **源点发送一个数据帧，等待ACK**
  - 保持一个发送帧的拷贝
  - 在终点确认返回前，源点不发送其他帧

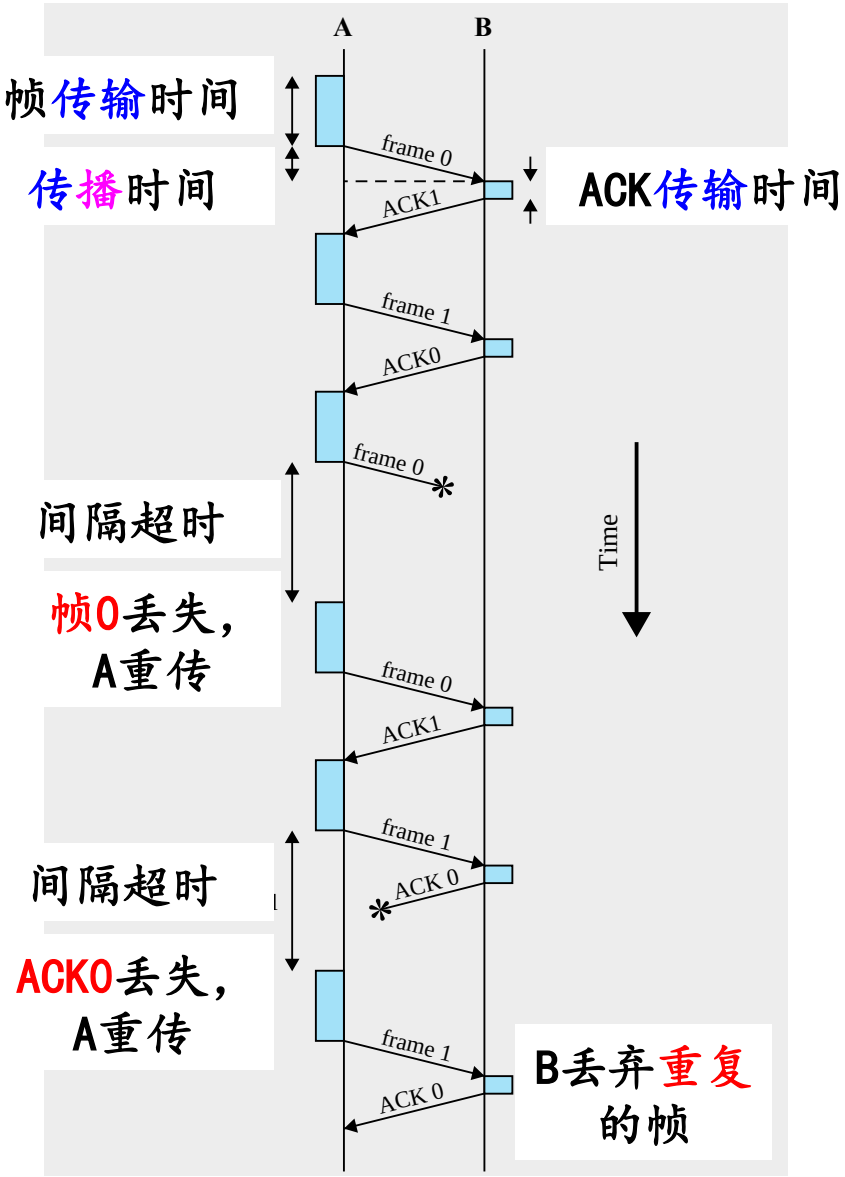
1）帧损伤

- 接收端检测到差错，丢弃该帧
- 发送端超时重传

2）确认（ACK）损伤

- 发送端超时重传
- 接收端收到用两份相同编号的帧
- 使用 ACK0 / ACK1 来确认希望接收的帧
  （ACK“i”是指接下来希望接收帧i）

停止等待ARQ


A
B
帧传输时间
传播时间
ACK传输时间
frame 0
ACK1
frame 1
ACK0
frame 0
间隔超时
帧0丢失，
A重传
Time
frame 0
ACK1
frame 1
ACK 0
间隔超时
ACK0丢失，
A重传
frame 1
ACK 0
B丢弃重复
的帧

返回N ARQ

- **基于滑动窗口流量控制机制**,没有收到确认的帧的最大数目取决于窗口大小
- 如果**没有差错**，使用肯定确认（RR）
- 如果**错误发生**，为错误帧发送**一个否认** (REJ)
  - **接收端丢弃**该帧和所有后来收到的帧，直到错误帧被正确接收
  - **发送端重传**有差错的帧和差错帧后所有已经传输过的帧

返回N ARQ对异常事件的反应

- 站点A正向站点B发送发送帧，且每次传输后均会为该帧设置确认计时器。现在假设B已成功接收到帧(i-1)，并且A刚刚传输了帧i。

- 返回N针对下述异常做出反应：

➢ **帧损伤** ：B接收到的帧时无效的，则B丢弃该帧，并且不对该帧做任何进一步的动作。

——具体分两种情况：接收端乱序、计时器超时

返回N ARQ对异常事件的反应

- **帧损伤—接收端乱序：**
  - 在合理的时间范围内，A继续发送帧(i+1)；
  - B接收到帧(i+1)后发现次序不对，于是发送REJ*i*；
  - A必须重传帧i和所有后继帧。

- **帧损伤—计时器超时：**
  - A传输一个RR帧，并在其中设置一个为1的P比特位；
  - B将RR中P比特位为1解释为一条命令；
  - B必须发送一个RR响应明确自己希望接收的下一帧，也就是帧i；
  - A收到B发送的RR后，重传帧i。

返回N ARQ对异常事件的反应

- **RR损伤**：B接收到帧i，但其回复的RR(i+1)损伤。

➢ **RR损伤—情况1**：

- B接收到帧i并发送RR(i+1)，而它在传输时丢失；
- 由于确认是累积的，A可能会受到下一个帧的RR；
- 这里下一个帧的RR可能在计时器超时前到达A。

（以为所有帧都正确接收了！）

➢ **RR损伤—情况2**：

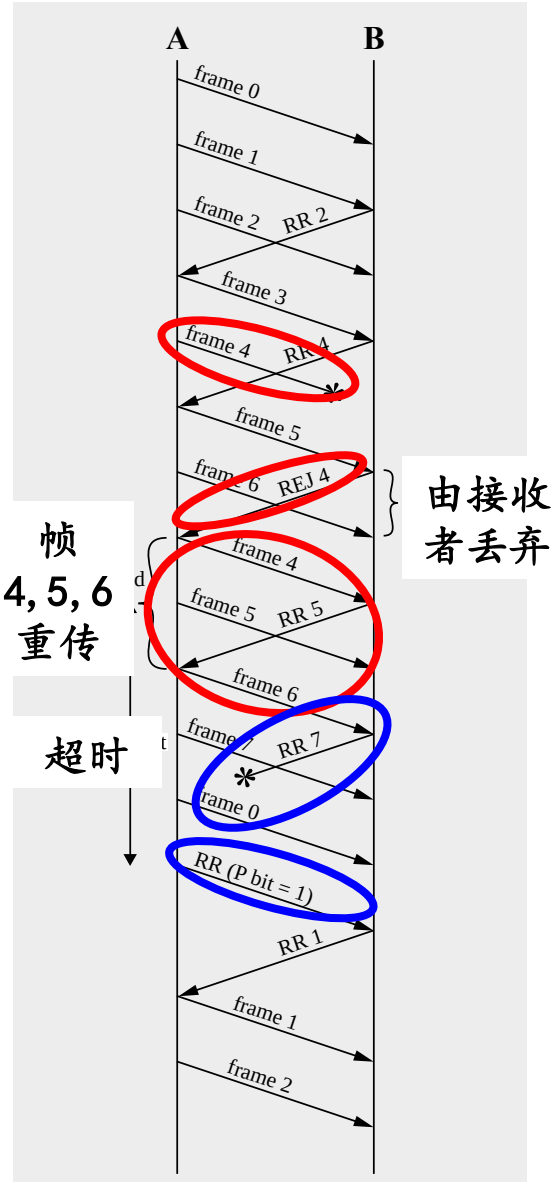
- A的计时器超时，它传输一个RR命令，含有前述P比特位；
- A还设置一个计时器，称为P比特计时器。
- 如果B没有响应RR命令或者响应损伤，P比特计时器会超时；
- A将发送新的RR命令，重启P比特计时器；
- 以上过程重复数次，当次数超过最大值后，还没每收到确认，A启动复位过程。

返回N ARQ对异常事件的反应

- **REJ损伤**：B接收到帧i并检测到有差错，希望重传帧i及其后的帧，但其回复的REJ(i+1)损伤。

  – 其情况等同于：帧损伤-计时器超时。

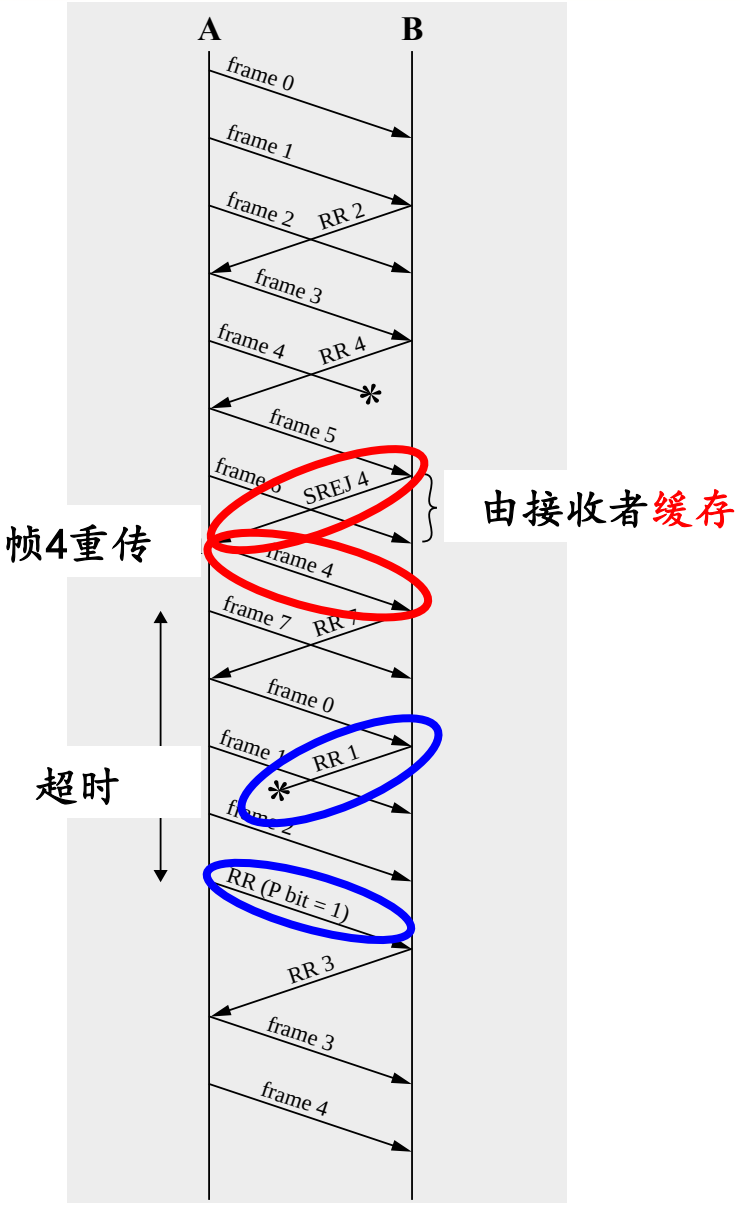
返回N ARQ示例


A
B
frame 0
frame 1
frame 2
RR 2
frame 3
frame 4
RR 4
frame 5
REJ 4
由接收者丢弃
帧4, 5, 6重传
frame 4
frame 5
RR 5
frame 6
RR 7
超时
frame 0
RR (P bit = 1)
RR 1
frame 1
frame 2

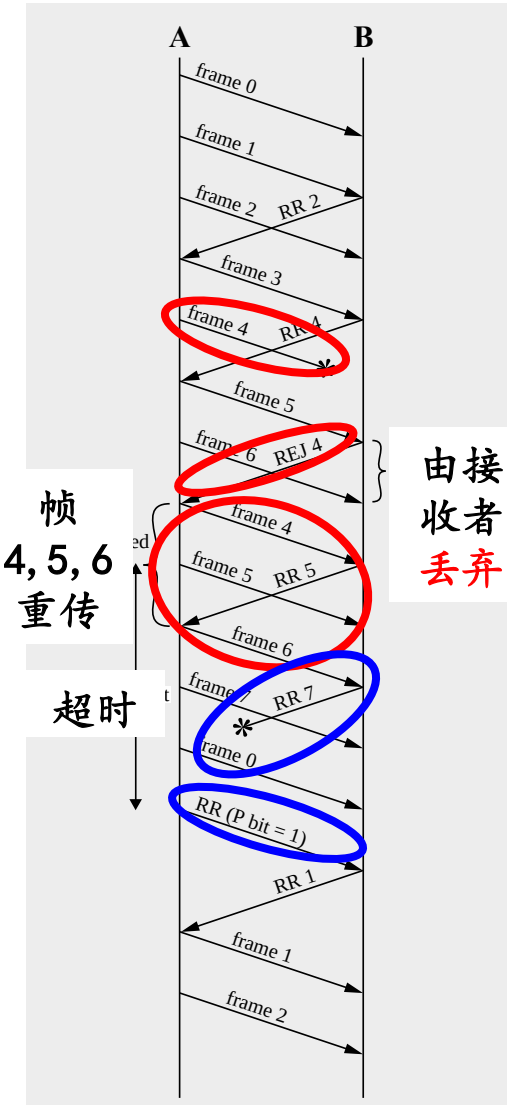
选择拒绝 (ARQ)

- 仅重传拒绝帧或超时帧，也叫做选择重传ARQ
  - 后续帧被接收端接收并缓存起来
  - 最小化重传帧的数量
- 接收端需要维护足够大的缓存
- 发送端和接收端逻辑更为复杂
  - 能够按照正确的顺序重组帧
  - 判断并仅发送失序帧
- 用于传播时延长的卫星链路

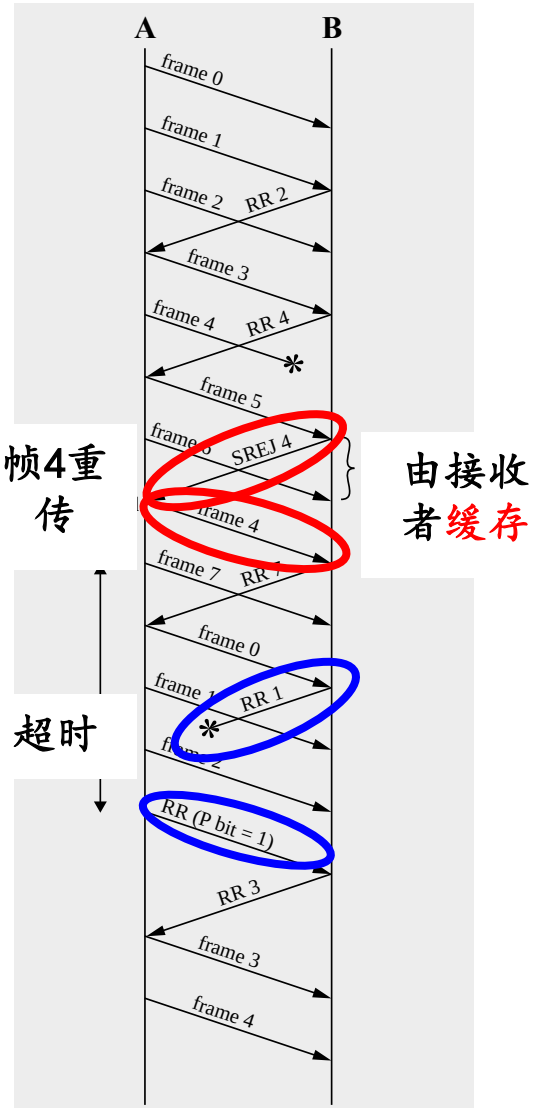
选择拒绝 (ARQ)


A
B
frame 0
frame 1
frame 2
RR 2
frame 3
frame 4
RR 4
frame 5
SREJ 4
frame 4
frame 7
frame 0
RR 1
RR (P bit = 1)
RR 3
frame 3
frame 4
帧4重传
超时
由接收者缓存

返回N ARQ和选择拒绝ARQ



返回N ARQ



选择拒绝ARQ

- 假设序号字段是n比特，最大的窗口大小是多少？

➢ 返回N ARQ

→ $2^n - 1$

➢ 选择拒绝ARQ

→ 2^{n-1}

返回N ARQ和选择拒绝ARQ

- 返回N ARQ
  - 发送方：窗口大小的缓存，用于备份重传
  - 接收方：无需缓存

- 选择拒绝 ARQ
  - 发送方：窗口大小的缓存，用于备份重传
  - 接收方：窗口大小的缓存

数据链路控制协议

1. 流量控制

2. 差错控制

3. 高级数据链路控制

高级数据链路控制 (HDLC)

- 最重要的数据链路控制协议，是其他重要数据链路控制协议基础。
- HDLC定义了3种类型的站点、2种链路设置以及3种数据传送运行方式。
- **站点：**
  - 主站：负责链路控制操作(命令)
  - 从站：在主站的控制下操作(响应)
  - 混合站：结合了主站和从站的特点（命令/响应）
- **链路设置：**
  - 非平衡设置：1个主站、多个从站（全双工/半双工）
  - 平衡设置： 2个混合站组成（全双工/半双工）

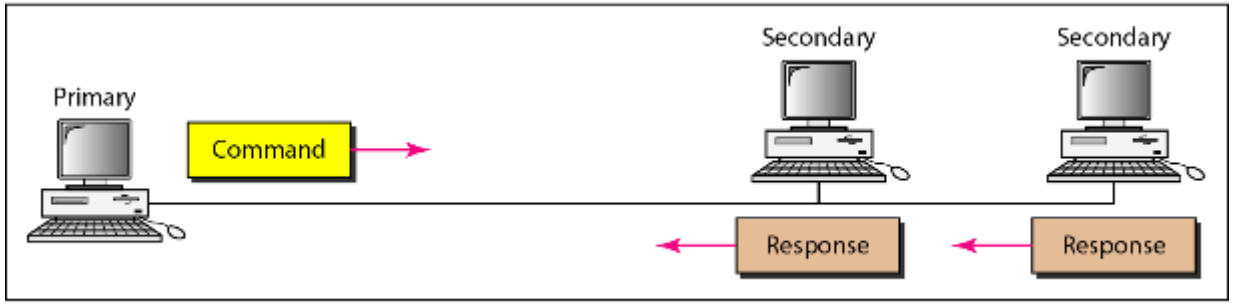
数据传送方式

- **正常响应方式** (NRM)
  - 非平衡设置
  - 主站能够发起到从站的数据传送
  - 从站只有在接收到主站的命令式才传输数据

主站

从站

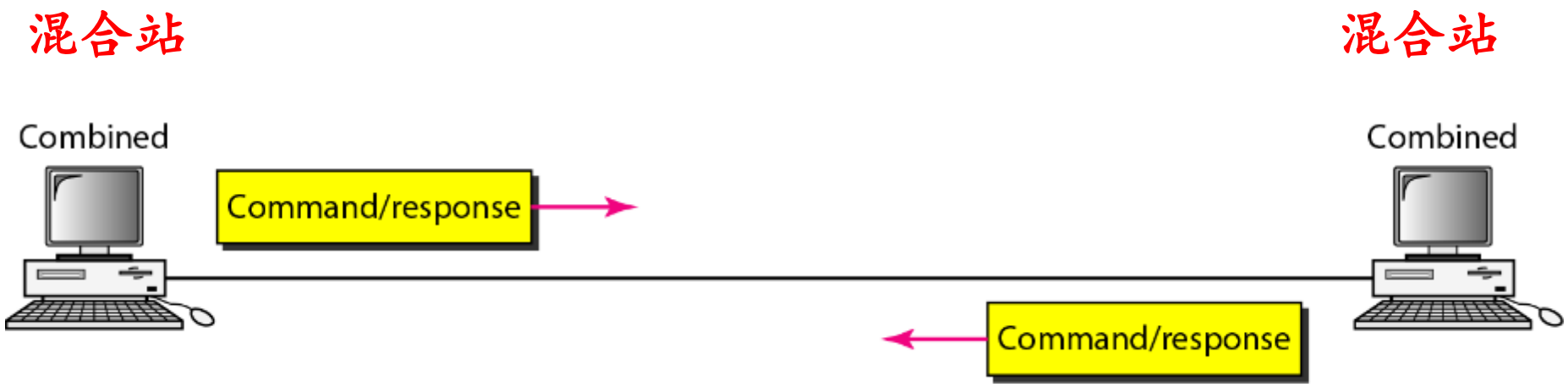
a. Point-to-point



b. Multipoint

数据传送方式

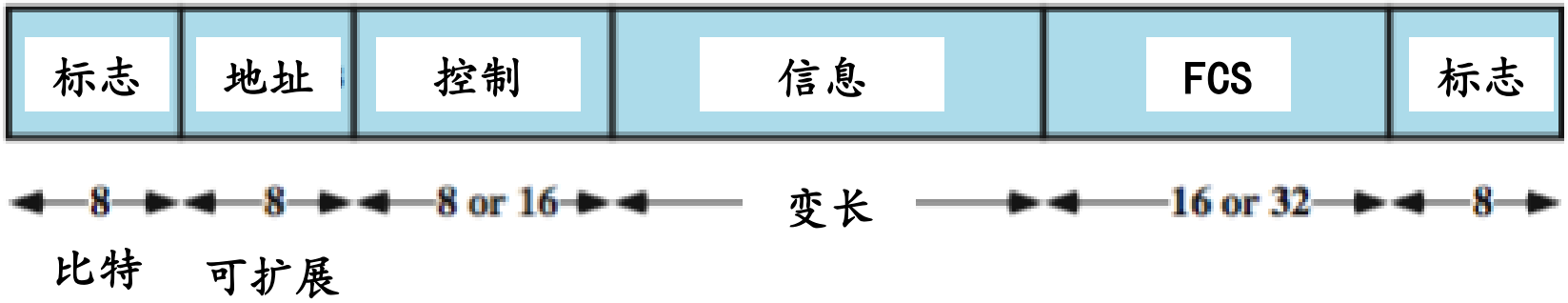
- **异步平衡方式** (ABM)
  - 平衡设置
  - 两个混合站都能够发起数据传输，不需要对方混合站的许可
  - 使用最广泛



数据传送方式

- **异步响应方式** (ARM)
  - 非平衡设置
  - 从站能够发起传输
  - 主站仍对链路全权负责，包括初始化、差错回复、链路逻辑断开等
  - 很少被使用

HDLC 帧结构



帧格式

- 使用同步传输
- 传输以帧的形式进行
- 一个帧格式满足所有数据和控制交换

标志字段和比特填充

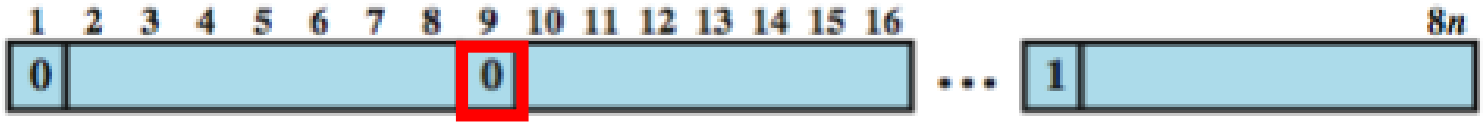
- 以 **01111110** 模式在帧的两端起定界作用
- 接收端不断搜索标志序列，用于一个**帧的同步**
- **比特填充**用于避免帧中间出现 01111110 导致误判
  - 每出现五个1，插入附加0
  - 如果接收端收到五个1，它检查下一个比特
    - 如果第6个比特为0，则该比特删除
    - 如果第6个比特为1，第7个为0，则为标志字段
    - 如果第6个和第7个比特是1，异常

原模式： 11111111111101111101111110

比特填充之后： 111110111110110111110101111010

地址域

- 标识传输或准备接收这个帧的**从站**
- 长度可以扩展
- 地址 11111111 用于**广播**



扩展的地址域

控制域

8比特控制字段格式

- **信息帧**：携带向用户传输的数据，采用ARQ机制，捎带了流量控制和差错控制数据
- **监控帧**：在未使用捎带技术时提供ARQ机制
- **无编号帧**：提供了增补的链路控制功能
- **P/F**：
  - **命令帧中P**：1代表向对等实体请求一个响应帧
  - **响应帧中F**：1代表对一个请求命令的响应

信息字段和检验序列字段

信息字段

- 只有I帧和U帧才具有
- 比特序列必须是八位组的整数倍
- 长度不固定

帧检验序列字段

- 用于差错检验
- 一般采用16 bit CRC或32 bit CRC

HDLC 运行方式

- 交换I帧, S帧和U帧
- 包含3个阶段

初始化

- 任何一方通过6个置位方式命令之一请求初始化（U帧）

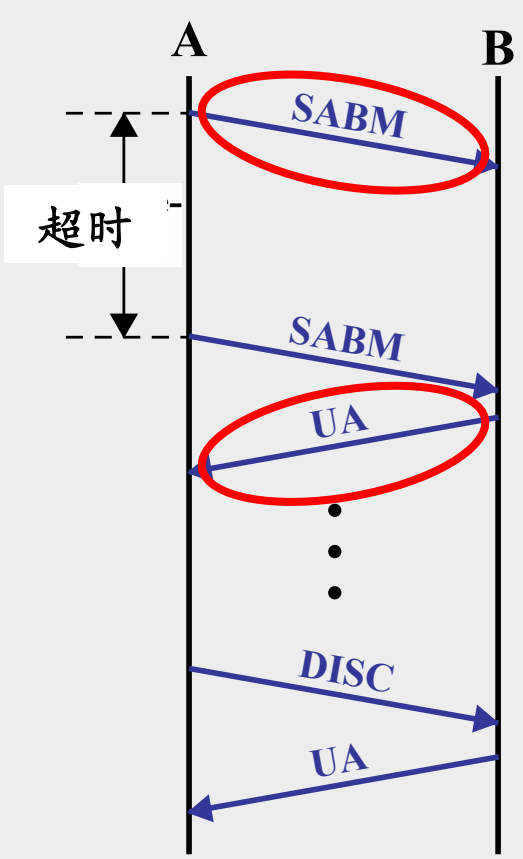
数据传送

- 支持流量控制和差错控制
- 使用I帧和S帧(RR，RNR，REJ，SREJ)

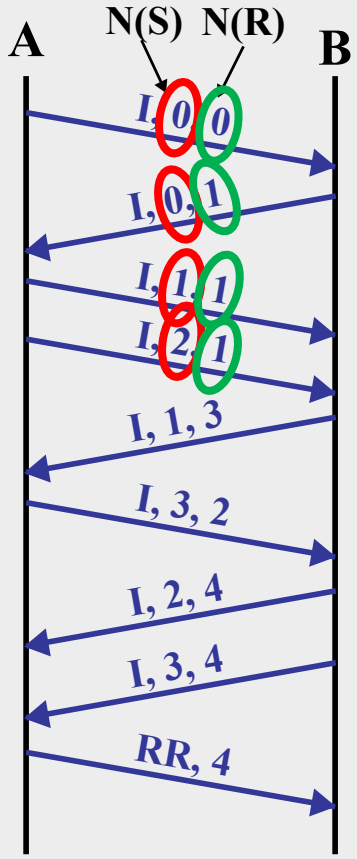
拆链

- 本身因某种错误引起中端，或是高层用户的请求
- 发送一个拆链帧(U帧，DISC)宣布连接终止

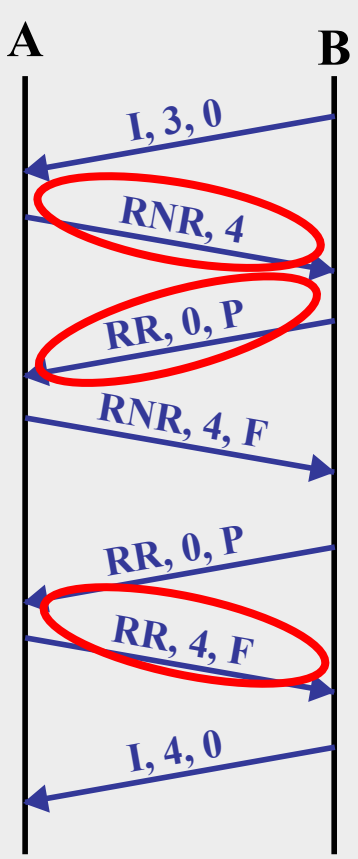
HDLC 运行方式举例



(a) 链路建立和拆链

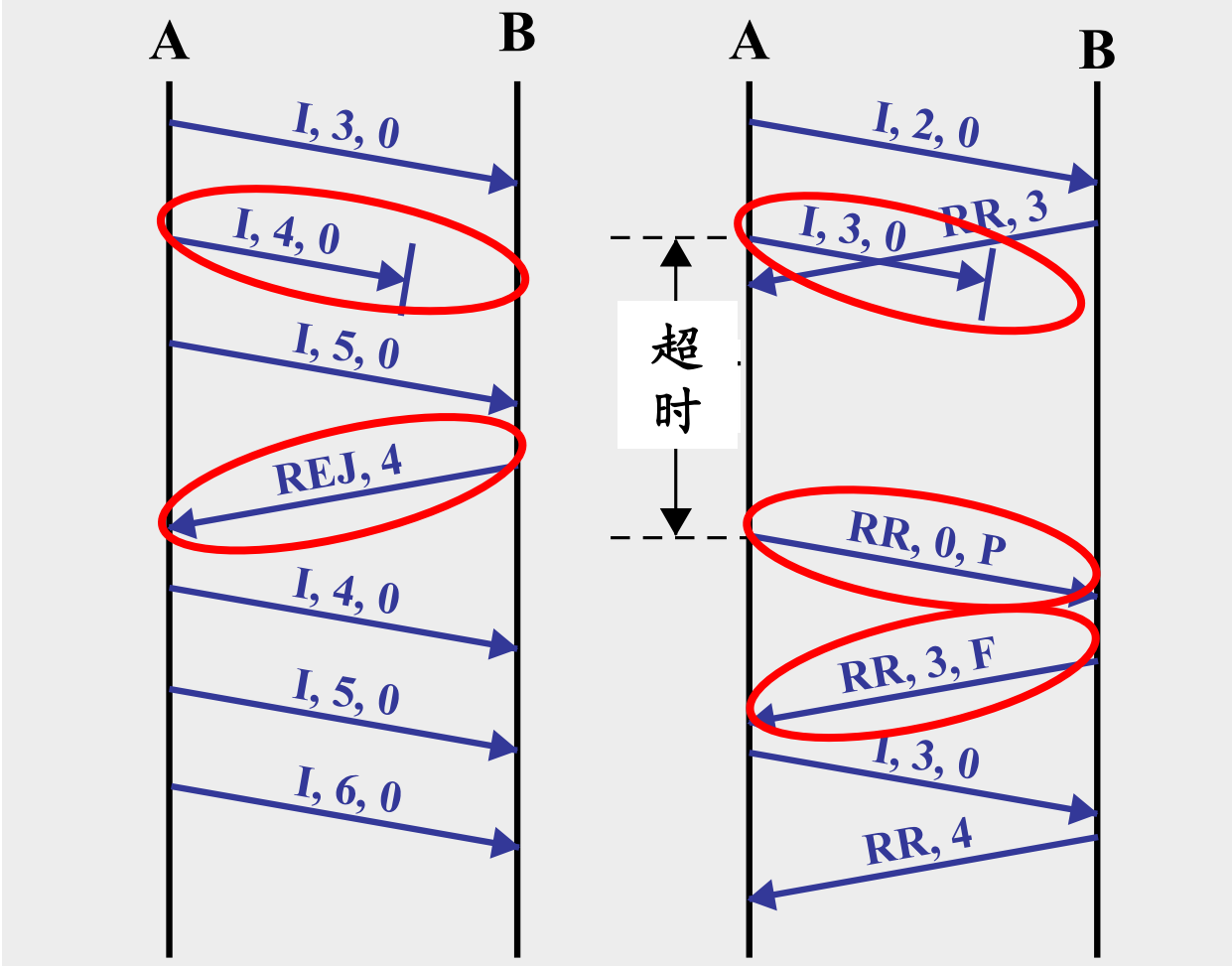


(b) 双向数据交换



(c) 出现忙的状态

HDLC 运行方式举例



(d) 拒绝的恢复　　(e) 超时的恢复

课程习题（作业）

课本（截止日期：习题课前/6月2日晚23:55）：

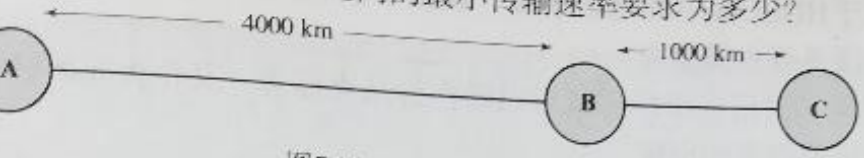
7.3 ；　7.6；　7.10 ；

提交方式：http://cslabcms.nju.edu.cn（本科教学支撑平台）

提交截止时间	2021年04月21日 星期三 23:55

- 命名：学号+姓名+第*章。
- 若提交遇到问题请及时发邮件或在下一次上课时反馈。

课程习题（作业）



度之间的关系，假设传播速度为2×10^8 m/s。

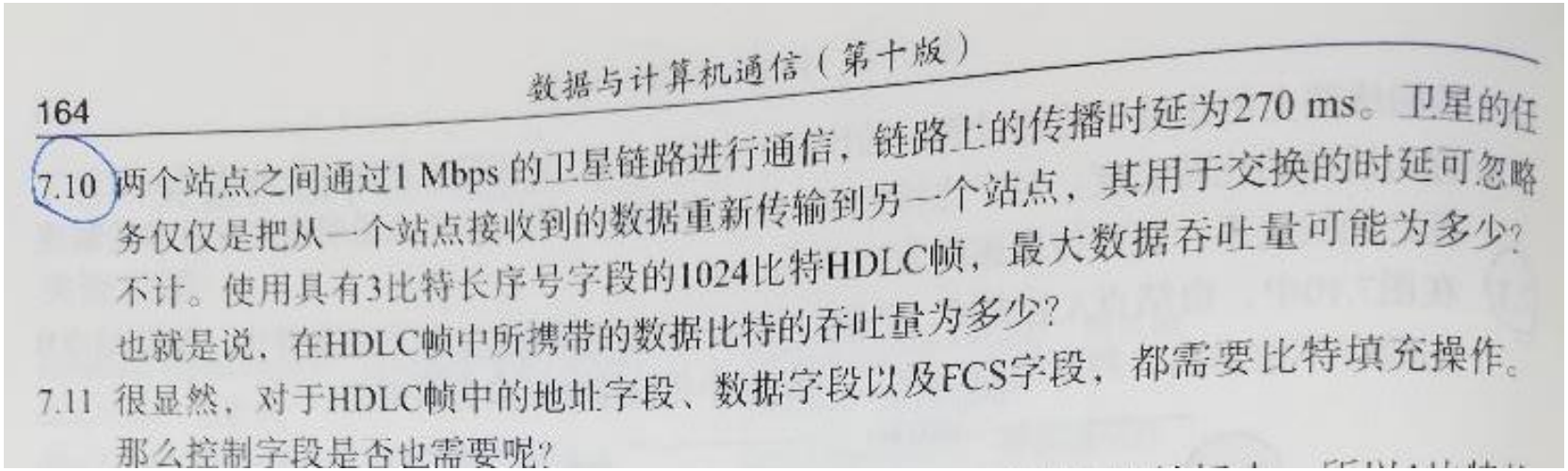
7.3 在图7.10中，由结点A生成帧，并通过结点B发送到结点C。在下述条件中，要使结点B的缓存不致溢出，判断结点B和C之间的最小传输速率要求为多少？

图7.10 习题7.3的配置图

- 结点A和B之间的数据率为100 kbps。
- 两条线路的传播时延都是5 μs/km。
- 节点之间的线路为全双工线路。
- 所有的数据帧都是1000比特长。ACK是独立的帧，长度可忽略不计。
- 在A和B之间，滑动窗口协议使用的窗口大小为3。
- 在B和C之间使用的是停止等待机制。
- 没有差错。

7.6 假设使用的是选择拒绝ARQ，且$W = 4$。举例说明序号字段的长度至少是3比特。

课程习题（作业）



164 数据与计算机通信（第十版）

7.10 两个站点之间通过1 Mbps的卫星链路进行通信，链路上的传播时延为270 ms。卫星的任务仅仅是把从一个站点接收到的数据重新传输到另一个站点，其用于交换的时延可忽略不计。使用具有3比特长序号字段的1024比特HDLC帧，最大数据吞吐量可能为多少？也就是说，在HDLC帧中所携带的数据比特的吞吐量为多少？

7.11 很显然，对于HDLC帧中的地址字段、数据字段以及FCS字段，都需要比特填充操作。那么控制字段是否也需要呢？

总结

问题？

殷亚凤
yafeng@nju.edu.cn
http://cs.nju.edu.cn/yafeng/
Room 901, Building of CS