



第 3 章 数据链路层

课程名称：计算机网络

主讲教师：姚烨

课程代码：**U10M11016.01**

E-MAIL : **yaoye@nwpu.edu.cn**

第 13-14 讲

2021 – 2022 学年第一学期



本节课程位置

1. 概述

2. 物理层

3. 数据链路层

4. 局域网

5. 网络层

6. 传输层

7. 应用层

8. 广域网

9. 网络新技术



第 1 章 概述

■ 学习目的

- 如何设计一个数据链路层通信协议，即支持可靠通信(差错控制+流量控制)，也支持不可靠通信。

■ 阅读材料

- 教材（HDLC规程，P37-41）+参考教材

■ 引导要点

- 网络协议三要素：语法、语义和同步
- 设计一个协议，即支持可靠通信，也支持不可靠通信
- 协议功能扩展

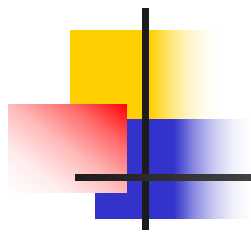
■ 编程任务

- 通信协议设计，见Project2



本节内容

- 1. HDLC 协议概述
- 2. HDLC协议设计
 - 2.1 HDLC 语法： 帧结构
 - 2.2 HDLC 语义： 帧类型
 - 2.3 HDLC 同步： 事件发生次序
- 3. 数据帧封装/拆封
- 4. 工程应用



**问题1:数据链路层协议
在设计时需要实现哪些
功能?**



1.1 数据链路层协议需求

- 数据链路层协议需求：
 - 发送方和接收方标识；
 - 支持可靠性通信，兼容不可靠通信；
 - 支持点到点和点到多点链路（支持广播）；
 - 支持半双工和全双工操作；
 - 支持主次站结构和复合站结构；



1.2 HDLC产生背景

- 面向字符数据链路层协议 (BSC) 缺点：
 - 发送节点和接收节点采用相同编码，不具有通用性。
 - 报文格式不一样：数据帧与控制帧格式不相同；
 - 停止-等待发送方式，传输效率低。
- 面向比特型协议的设计目标
 - 传输透明性好：数据可采用任何比特组合；具有通用性：数据块为比特流，不用考虑字符编码问题；
 - 数据帧与控制帧格式相同；
 - 可连续发送多个数据帧，传输效率高。



1.3 HDLC协议产生

- 1974年，IBM公司率先推出了**面向比特**的数据链路层访问控制规程SDLC (Synchronous Data Link Control)：**同步数据链路控制**规程。
- 后来，ISO在SDLC基础上修订，产生一个新版本，称为HDLC (High-level Data Link Control)，译为高级数据链路访问控制规程，已作为国际标准ISO-3309。
- CCITT则将 HDLC 修订后称为链路接入规程 LAP (Link Access Procedure)；不久，把 LAP 修订为 LAPB， “B”表示平衡型(Balanced)，所以 LAPB 叫平衡型链路接入规程。-X.25网络一部分
- **IEEE 802.2 LLC协议也源于HDLC**。这些通信规程尽管在一些细节上存在着差异，但总的来说是大同小异的。

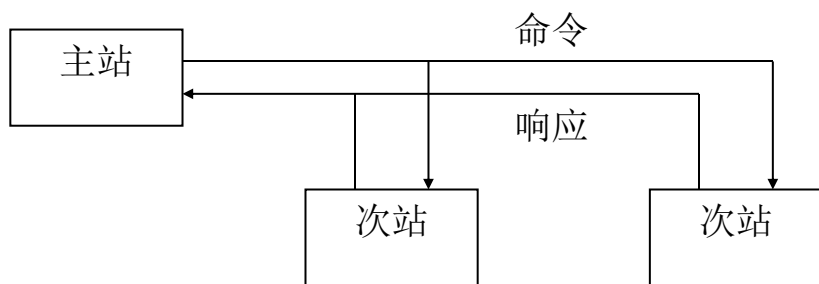


HDLC协议特点

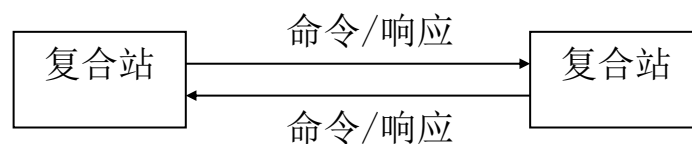
- HDLC规程被广泛采用
 - 高效率、高可靠性及能传输任意比特组合数据等特点。
 - 如X. 25, 以及IEEE 802局域网等采用部分技术。
- HDLC协议特征
 - 三种类型的站
 - 两种链路结构
 - 三种数据传送操作方式

(1) 三种类型站

- **主站**：负责对数据链路实行全面**管理**，包括链路建立，链路维护，数据发送，**差错控制**，**链路故障后恢复**等。
- **次站**：受控于主站，接收主站命令执行相应操作。
- **复合站**：具有主站和次站双重功能的站，两个复合站之间可以**完全对等地**进行通信。



(a) 非平衡式链路结构



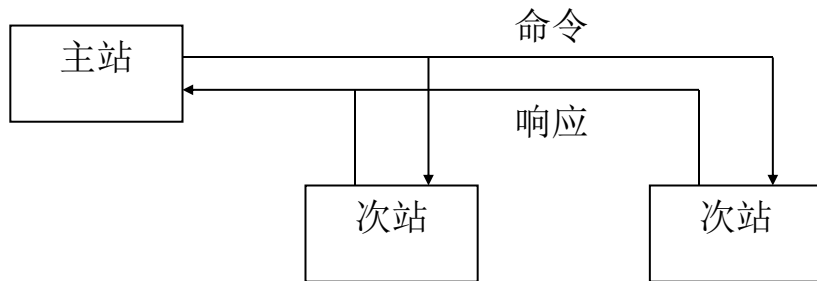
(b) 平衡式链路结构

链路结构

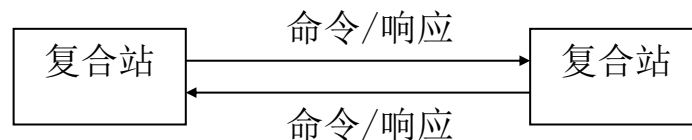
1.5 数据链路类型

■ 非平衡式链路结构

- 由一个主站和多个(一个)次站通过链路连接而成。
- 实现点到点或点到多点通信。
- 通信操作方式:半双工或全双工。
- 主站对链路实施控制, 发送命令帧, 将次站置于某种状态或适当操作方式。
- 次站响应主站的命令, 或执行相应的操作, 或向主站回送应答。
- 主站和次站之间通过交换数据和控制信息, 实现差错控制和故障恢复功能。



(a) 非平衡式链路结构

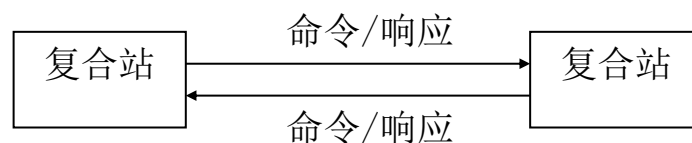
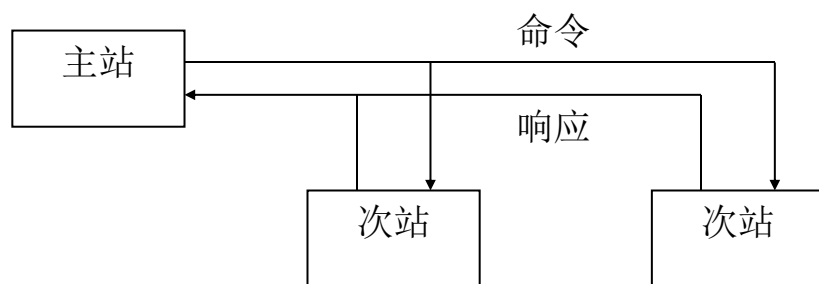


(b) 平衡式链路结构

1.5 数据链路类型

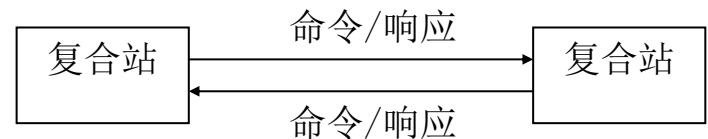
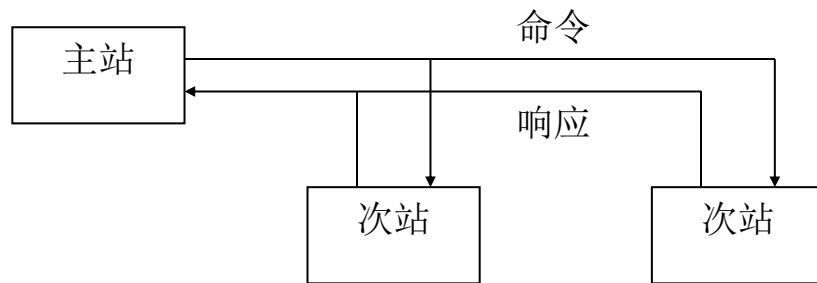
■ 平衡式链路结构

- 由两个复合站以点对点方式连接而成。
- 通信方式: 半双工或全双工。
- 两个复合站点都具有数据传送和链路控制能力。



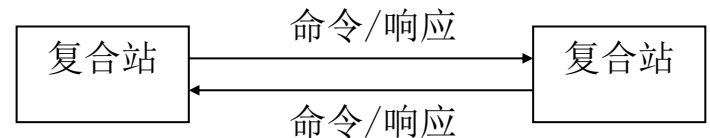
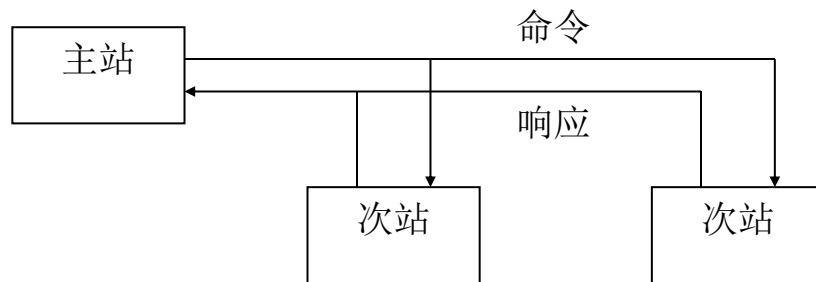
1.6 三种数据传送操作方式

- 正常响应方式NRM (Normal Response Mode);
 - 适用于**非平衡式链路结构**。
 - 次站只有在接收到来自主站的允许（传输数据帧命令）之后，才可传送一帧或多帧数据。
 - 次站在发出最后的数据帧后，将停止发送。



三种数据传送操作方式

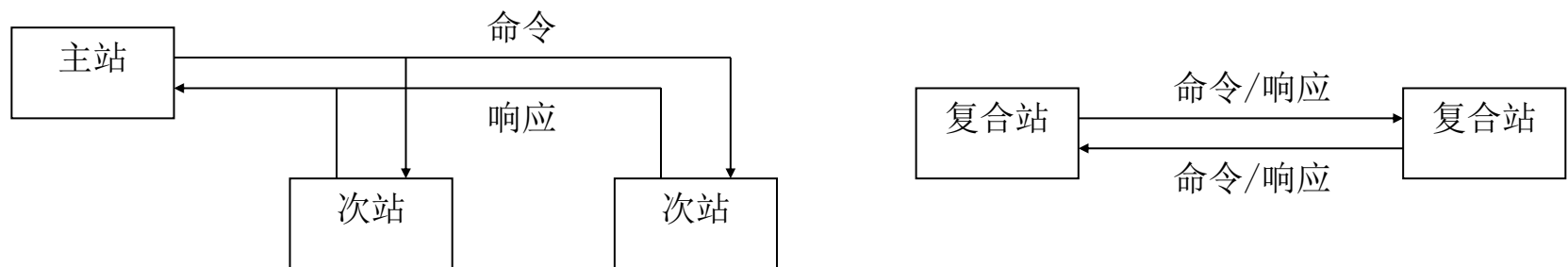
- 异步响应方式 (ARM: Asynchronous Response Mode)
 - 适用于**非平衡式链路结构**;
 - 次站不一定在接收到主站允许即可进行数据传输;
 - 次站可以一次发送一帧或多帧连续发送;
 - 次站可将自己状态信息发送给主站。



三种数据传送操作方式

■ 异步平衡方式ABM (Asynchronous Balance Mode)

- 适用于平衡式链路结构;
- 复合站自主决定数据传输, 不必得到另一个复合站允许;
- 一次可以传输一帧或多帧;
- 既可发送数据帧, 也可发送自己状态信息或命令帧。



目前互联网采用平衡链路结构?
非平衡链路结构就一点用处都没有吗?



本节内容

- 1. HDLC 协议概述
- 2. HDLC协议设计
 - 2.1 HDLC 语法： 帧结构
 - 2.2 HDLC 语义： 帧类型
 - 2.3 HDLC 同步： 事件发生次序
- 3. 数据帧封装/拆封
- 4. 工程应用

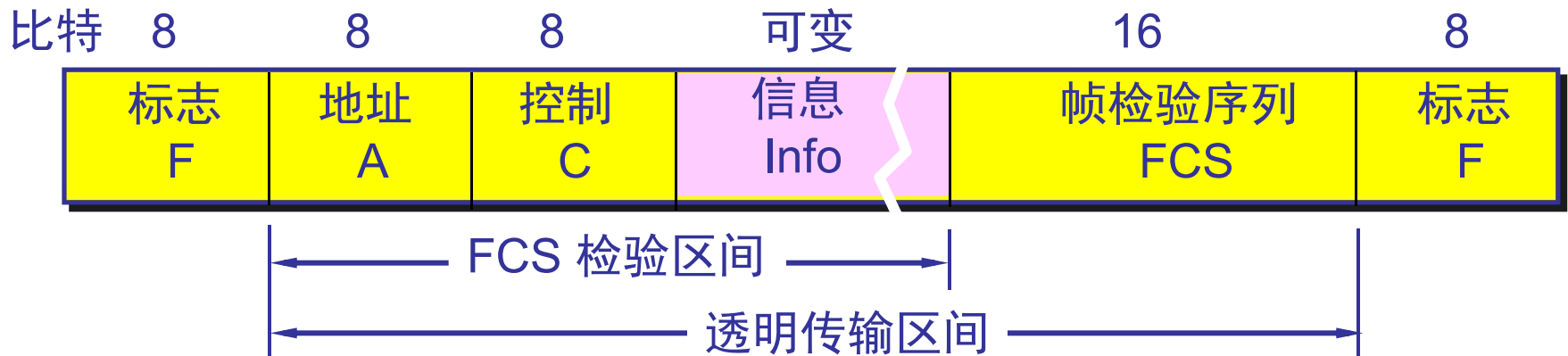


■ HDLC协议设计

- 语法：数据及命令格式
- 语义：数据帧类型
- 同步：事件顺序

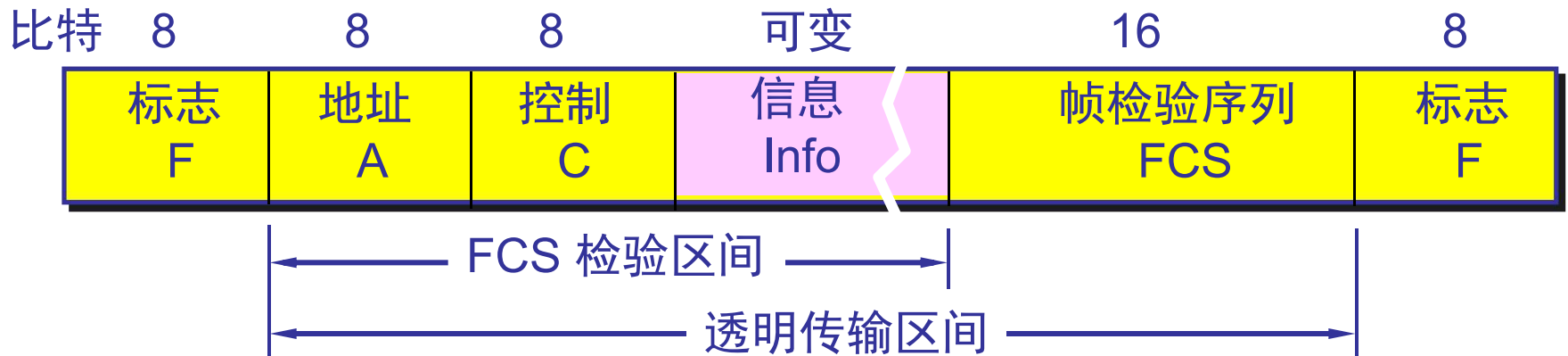
2.1 HDLC协议-语法：格式

- HDLC不论传送数据, 或控制信息, 以帧(Frame)为基本单位。



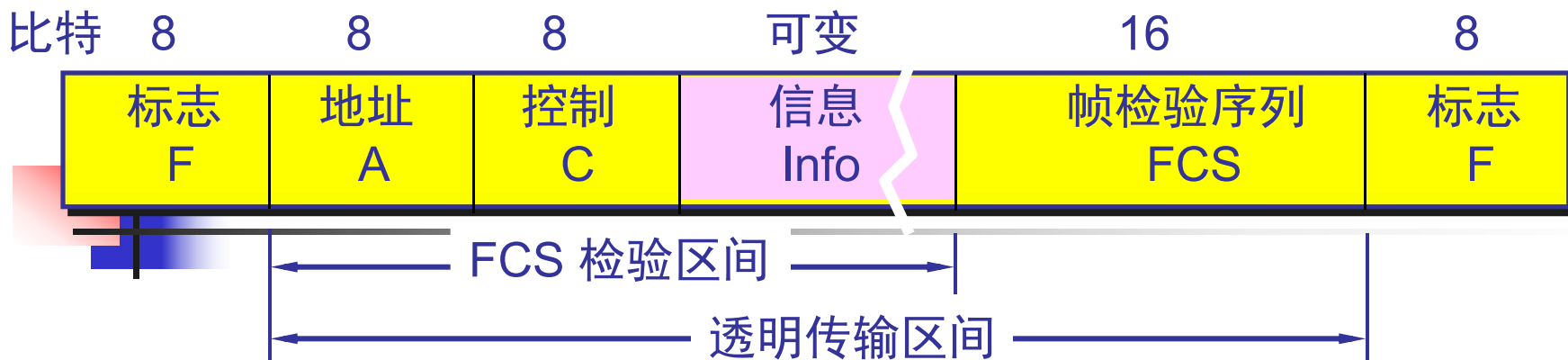
2.2 HDLC协议-语法： 格式

- HDLC不论传送数据, 或控制信息, 以帧(Frame)为基本单位。



- 标志字段 F (Flag)**: 为 6 个连续 1 加上两边各一个 0 共 8 bit (**01111110**)。在接收方只要找到标志字段就可确定一个帧的位置。
- 采用面向比特的同步传输方法。

2.2 HDLC协议-语法：格式



■ 地址字段 (Address)

- 对于**控制帧**，该字段为接收方地址；对于响应帧，该字段为发送方地址。

■ 地址字段扩展

- 如果地址字节的**第1位为0**，表示后面字节仍为地址字节，依次类推，直到出现一个**第1位为1的字节为止**，表明该字节就是地址字段的最后一个字节。
- 不扩展情况下，**第1位必为1**，地址字段为8比特位。

■ 特殊地址

- 组地址 (点到多点)：一个地址可分配给多个站点；
- 广播地址 (点到多点)：全“1”，链路上所有站点均可以收到。
- 地址字段全“0”：不分配给任何站点，仅作为测试用。

2.2 HDLC协议-语法：格式

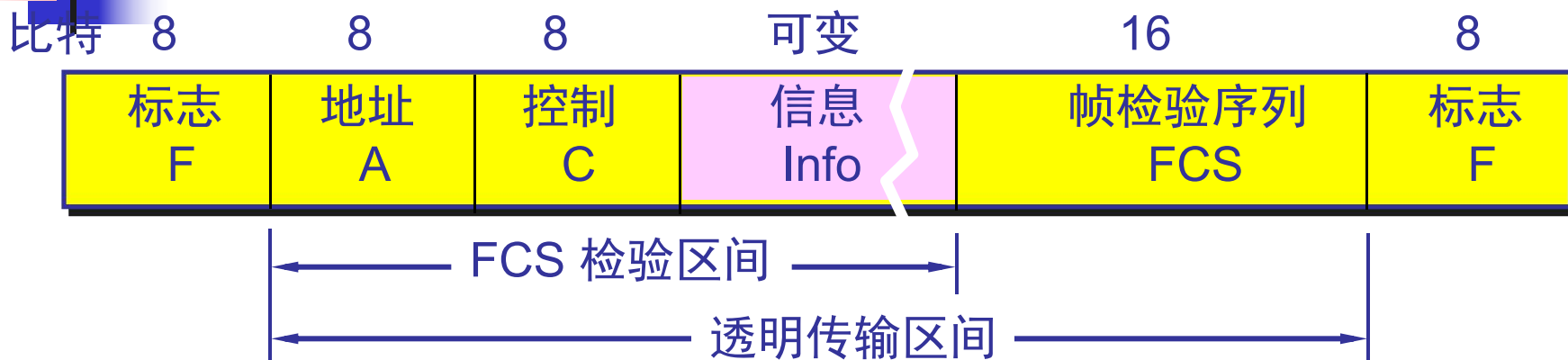


	1	2	3	4	5	6	7	8
I 帧:	0	N(S)			P/F	N(R)		
S 帧:	1	0	S		P/F	N(R)		
U 帧:	1	1	M		P/F	M		

HDLC 帧格式

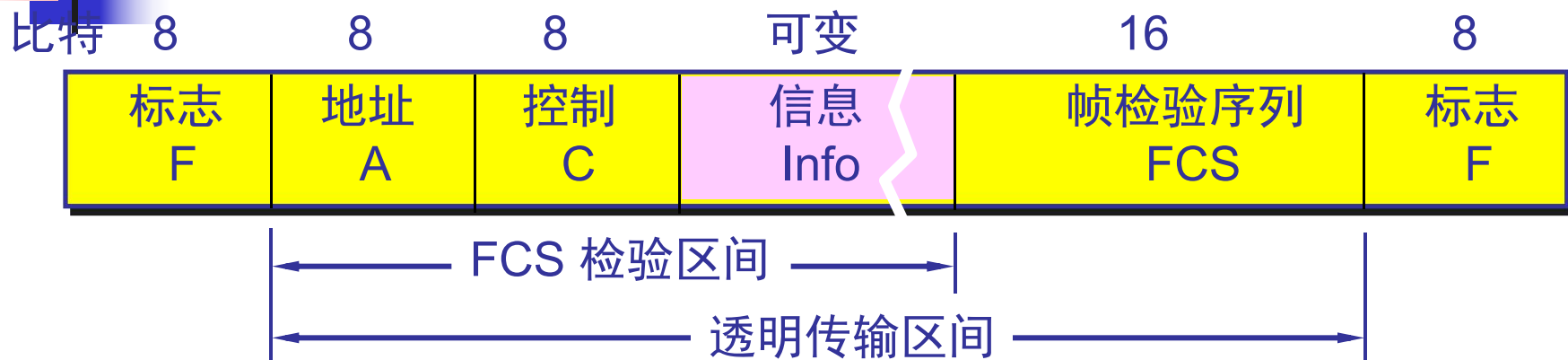
- 控制字段 (Control): 是HDLC协议关键字段, 设计各种控制帧或响应帧, 以便对数据链路监测 (差错) 和控制 (流量)。
- HDLC利用前两个比特位定义了三种类型帧。
 - I 帧 (信息帧): 可靠通信时数据帧;
 - S 帧 (监控帧): 实现流量控制和差错控制;
 - U 帧 (无编号U帧): 设置链路操作方式, 不可靠通信数据帧, 无编号应答帧;

2.2 HDLC协议-语法：格式



- 信息字段 (Information): 网络层协议数据单元, 为任意组合比特流。
- 信息字段长度是可变
 - 理论上不限长度, 但实际中要受到FCS校验能力、站点缓冲区大小等具体因素的限制, 国际上较常用信息字段最大长度: 1000~2000 byte。
 - 最小长度为0, 即无数据, 如S帧 (监控帧) 中无数据。

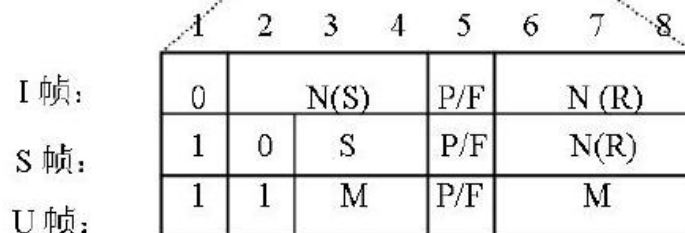
2.2 HDLC协议-语法： 格式



- 帧校验字段 (FCS)：采用16位的CRC冗余校验码，用于差错检测。
 - **CRC-CCITT**: $G(x) = x^{16} + x^{12} + x^5 + 1$
 - 标志字段不在校验范围类。

语义：帧类型

控制字段(C)



HDLC 帧格式

■ 信息帧(I 帧)

- 用于面向连接（提供可靠性）的数据传输。
- 发送站用N(S)表示所发送帧序号。
- N(R)表示捎带确认，即发送数据时确认；表示已正确接收到N(R)-1个帧，下一个要准备接收的帧序号为N(R)。

■ 探询/终止(P/F)位在传输时被置“1”或置“0”。

■ 两个站点相互传送信息帧，其帧序号相互独立。

■ 控制字段不扩展情况下，发送方在没有收到接收方应答确认的情况下，可连续发送I帧的数量最多少？(7, 4)。

语义：帧类型

控制字段(C)

8 位	8 位	8 位	n 位	16 位	8 位
标志	地址	控制	信息	帧校验	标志

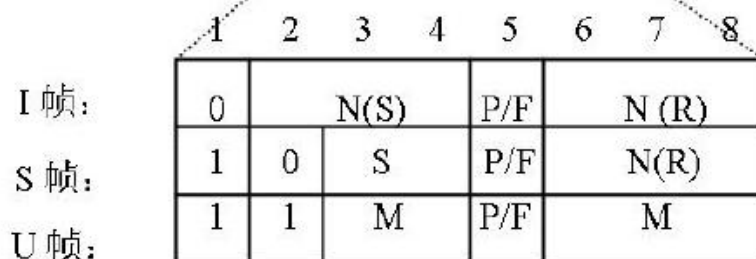
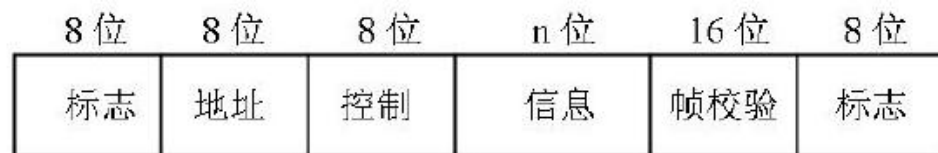
	1	2	3	4	5	6	7	8
I 帧:	0	N(S)			P/F	N(R)		
S 帧:	1	0	S		P/F	N(R)		
U 帧:	1	1	M		P/F	M		

- 监控帧(S帧)：用于在可靠性数据传输中链路监控功能(流量控制和差错控制)，如确认、请求暂停、重发等。
- S字段定义S帧的类型, 共有4种S帧; S帧没有数据, 不设发送序号。

3-4比特	帧 名	功 能	N(R) 含义
00	RR 接收准备好	ACK确认, 准备接受下一帧, N(R) 以前的各帧以接收	期望下一I帧序号
10	RNR 接收未准备好	确认, 暂停接收下一帧, N(R) 含义同上	N(R) 以前帧已接收
01	REJ 后退N帧	N(R) 以前的帧已接收, 以后 所有帧重发	重发帧开始序号
11	SREJ 选择重发	重发序号为N(R) 帧	重发帧序号

语义：帧类型

控制字段(C)

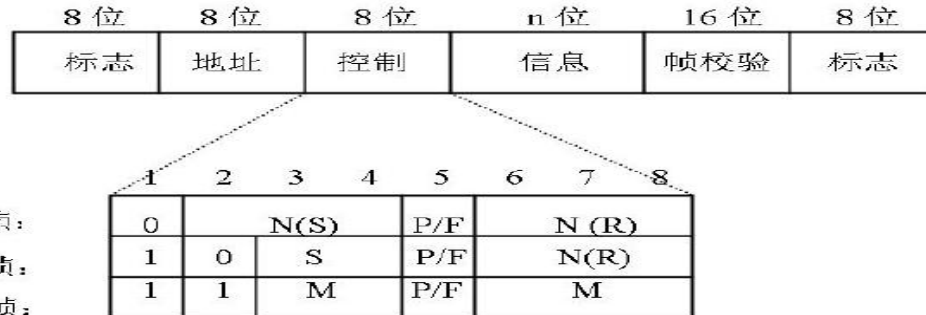


HDLC 帧格式

- 无编号帧(U帧): 由于无序号N(S), N(R)而得名。
 - 可靠数据传输中, 提供链路管理与维护功能, 如设置方式(建立链路)、链路维护(如链路复位、拆除链路等, 而**差错控制和流量控制**由S帧来完成);
 - 不可靠通信中数据帧(无编号UI帧) (**00 000**)
 - **M**字段用于定义**U帧的类型**, 可最多有32种, HDLC协议只用到**19种U帧**。

语义：帧类型

无编号帧



HDLC 帧格式

记忆符	名 称	B ₃ B ₄ B ₆ B ₇ B ₈
SNRM	置正常响应模式	00001
SARM/DM	置异步响应模式/断开方式	11000
SABM	置异步平衡模式	11100
SNRME	置扩充正常响应模式	11011
SARME	置扩充异步响应模式	11010
SABME	置扩充异步平衡模式	11110
DISC/RD	断链/请求断链	00010
SIM/RIM	置初始化/请求初始化方式	10000
UP	无编号探询	00100
UI	无编号信息	00000
XID	交换识别	11101
RESET	复位	11001
FRMR	帧拒绝	10001
UA	无编号确认	00110

语义：HDLC帧类型

帧类型	命 令	应 答	控制字段各位↵							
I	信 息		0	N(S)	P/F	N(R) ↵				
S	RR——接收准备好	RR	1	0	0	0	P/F	N(R) ↵		
	RNR——接收未准备好	RNR	1	0	0	1	P/F	N(R)↵		
	REJ——请求重发	REJ	1	0	1	0	P/F	N(R) ↵		
	SREJ——选择请求重发	SREJ	1	0	1	1	P/F	N(R) ↵		
U	SNRM——置 NRM 方式	DM—— 断开方式	1	1	0	0	P	0	0	1 ↵
	SARM——置 ARM 方式		<u>1</u>	<u>1</u>	1	1	P/F	0	0	0↵
	SABM——置 ABM 方式		1	1	1	1	P	1	0	0↵
	SNRME——置扩展 NRM 方式		1	1	1	1	P	0	1	1↵
	SARME——置扩展 ARM 方式		1	1	1	1	P	0	1	0↵
	SABME——置扩展 ABM 方式		1	1	1	1	P	1	1	0↵
	SIM——置初始化方式	RIM——请求初始化方式	<u>1</u>	<u>1</u>	1	0	P/F	0	0	0↵
	DISC——断开连接	RD——请求断开	<u>1</u>	<u>1</u>	0	0	P/F	0	1	0↵
	UI——无编号信息帧	UI——无编号信息帧	<u>1</u>	<u>1</u>	0	0	P/F	0	0	0 ↵
	UP——无编号探询		1	1	0	0	P	1	0	0↵
	RSET——复位		1	1	1	1	P	0	0	1
	XID——交换标识	XID——交换标识	<u>1</u>	<u>1</u>	1	1	P/F	1	0	1↵
		UA——无编号确认	<u>1</u>	<u>1</u>	0	0	F	1	1	0↵
		CMDR——命令帧拒收	<u>1</u>	<u>1</u>	1	0	F	0	0	1↵

语义：帧类型

控制字段(C)

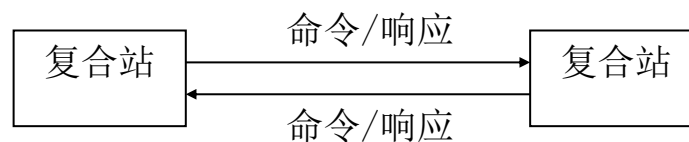
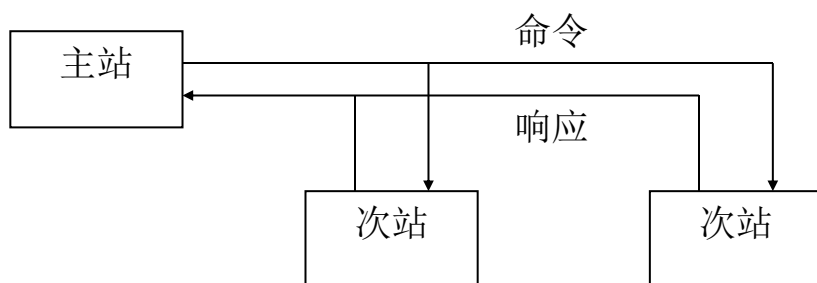
8 位	8 位	8 位	n 位	16 位	8 位
标志	地址	控制	信息	帧校验	标志

	1	2	3	4	5	6	7	8
I 帧:	0	N(S)			P/F		N(R)	
S 帧:	1	0	S		P/F		N(R)	
U 帧:	1	1	M		P/F		M	

HDLC 帧格式

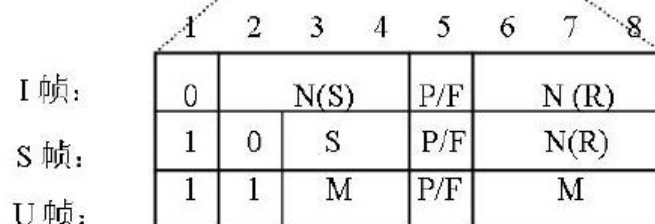
■ 探测/终止 (P/F) 位

- 探测位P: 主站用来请求次站应答或发送数据。
- 终止位F: 次站用来响应主站探测。



语义：帧类型

控制字段(C)



HDLC 帧格式

控制字段扩展

- 由原来一个字节可扩展为两个字节;
- 扩展后的控制字段主要增加了N(S)和N(R)的长度, 即由原来的3位增加到7位;
- 发送站在没有收到接收站应答确认的情况下, 可连续发送I帧的数量最多少(127, 64); **如何扩展?**
- 控制字段扩展是通过相应无编号U帧来设置。

■ 无效帧: 是指没有用两个标志(F)字段作边界的帧, 或者两个标志字段之间的位数小于32bit的超短帧。

■ 问题: 发送一个数据帧, 该帧有两个F字段, 如果连续发送多个帧, 会有几个F字段呢?

- 中间F字段可省略一个。

HDL C协议语义

三种控制帧类型总结

- 在HDL C协议中，控制字段定义了3类帧类型（大类型）：信息I帧、监控S帧和无编号U帧。

0	N(S) (3位序列号)	P/F	N(R) (准备接收序列号)
---	-----------------	-----	-------------------

I帧用于面向连接数据帧，并具有捎带确认功能

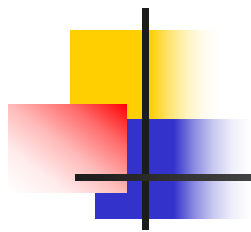
1	0	TYPE (2位)	P/F	N(R) (准备接收序列号)
---	---	--------------	-----	-------------------

S帧用于监控链路，包括差错控制和流量控制。

1	1	TYPE	P/F	TYPE
---	---	------	-----	------

U帧用于数据链路建立，维护(故障恢复)，拆除链路

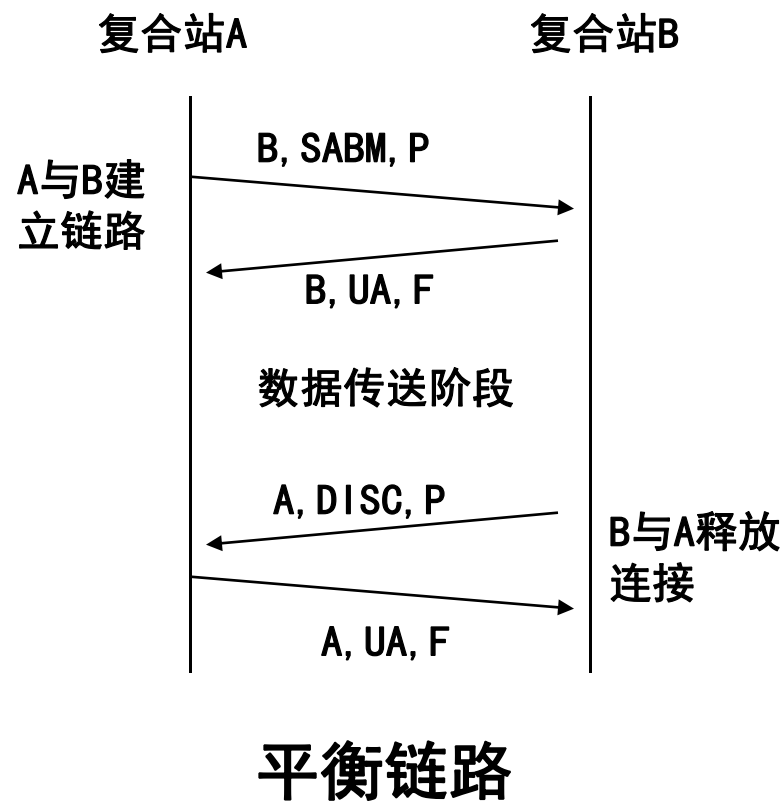
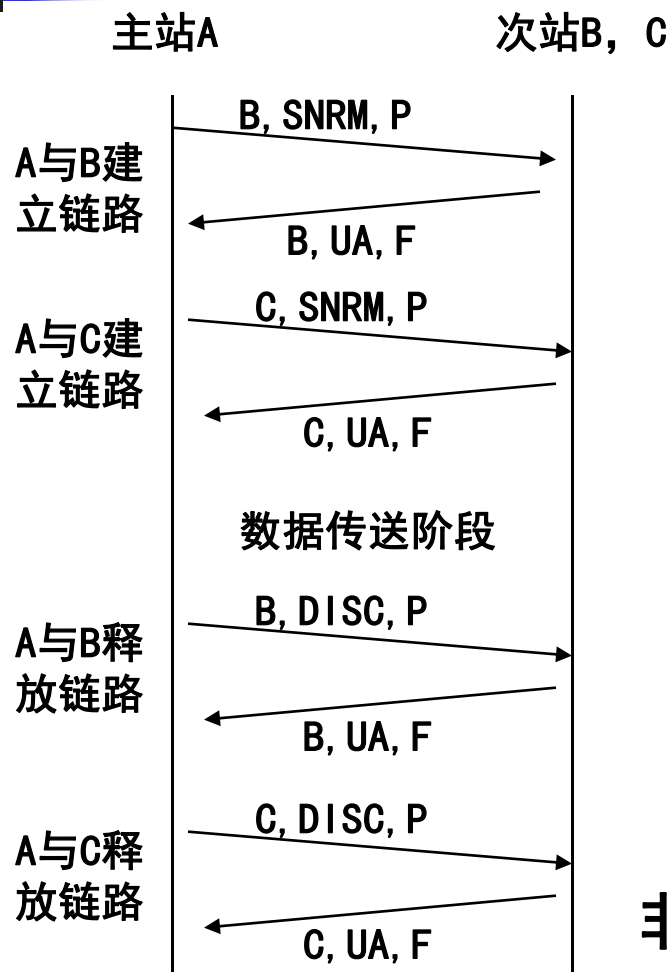
不可靠通信数据帧UI



2.3 HDLC协议-同步

事件发生顺序

(1) 链路建立和拆除



(2) 有确认可靠 通信服务

帧类型	命令	应 答	控制字段各位 ^{a)}
I	信息		0 N(S) P/F N(R) ^{a)}
S	RR——接收准备好	RR	1 0 0 0 P/F N(R) ^{a)}
	RNR——接收未准备好	RNR	1 0 0 1 P/F N(R) ^{a)}
	REJ——请求重发	REJ	1 0 1 0 P/F N(R) ^{a)}
	SREJ——选择请求重发	SREJ	1 0 1 1 P/F N(R) ^{a)}
U	SNRM——置 NRM 方式	DM—— 断开方式	1 1 0 0 P 0 0 1 ^{a)}
	SARM——置 ARM 方式		1 1 1 1 P/F 0 0 0 ^{a)}
	SABM——置 ABM 方式		1 1 1 1 P 1 0 0 ^{a)}
	SNRME——置扩展 NRM 方式		1 1 1 1 P 0 1 1 ^{a)}
	SARME——置扩展 ARM 方式		1 1 1 1 P 0 1 0 ^{a)}
	SABME——置扩展 ABM 方式		1 1 1 1 P 1 1 0 ^{a)}
	SIM——置初始化方式	RIM——请求初始化方式	1 1 1 0 P/F 0 0 0 ^{a)}
	DISC——断开连接	RD——请求断开	1 1 0 0 P/F 0 1 0 ^{a)}
	UI——无编号信息帧	UI——无编号信息帧	1 1 0 0 P/F 0 0 0 ^{a)}
	UP——无编号探测		1 1 0 0 P 1 0 0 ^{a)}
	RSET——复位		1 1 1 1 P 0 0 1
	XID——交换标识	XID——交换标识	1 1 1 1 P/F 1 0 1 ^{a)}
		UA——无编号确认	1 1 0 0 F 1 1 0 ^{a)}
		CMDR——命令帧拒收	1 1 1 0 F 0 0 1 ^{a)}

A发送无编号U帧请求建立链路



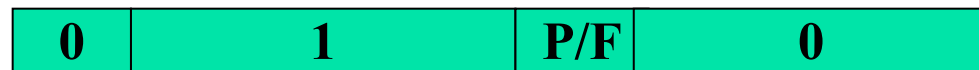
B发送无编号U帧确认建立链路



A发送序号为0的数据I帧



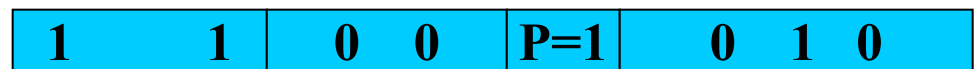
A发送序号为1的数据I帧



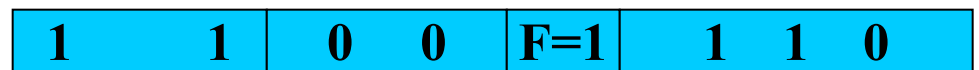
B发送序号为0的数据I帧，
捎带确认确认



A发送无编号U帧请求断开链路



B发送无编号U帧确认断开链路



(3) 选择重发

帧类型	命令	应 答	控制字段各位 ^{a)}					
			0	N(S)	P/F	N(R)	a)	
I	RR——接收准备好	RR	1	0	0	0	P/F	N(R)
	RNR——接收未准备好	RNR	1	0	0	1	P/F	N(R)
	REJ——请求重发	REJ	1	0	1	0	P/F	N(R)
	SREJ——选择请求重发	SREJ	1	0	1	1	P/F	N(R)
U	SNRM——置 NRM 方式	DM—— 断开方式	1	1	0	0	P	0 0 1
	SARM——置 ARM 方式		1	1	1	1	P/F	0 0 0
	SABM——置 ABM 方式		1	1	1	1	P	1 0 0
	SNRME——置扩展 NRM 方式		1	1	1	1	P	0 1 1
	SARME——置扩展 ARM 方式		1	1	1	1	P	0 1 0
	SABME——置扩展 ABM 方式		1	1	1	1	P	1 1 0
	SIM——置初始化方式	RIM——请求初始化方式	1	1	1	0	P/F	0 0 0
	DISC——断开连接		1	1	0	0	P/F	0 1 0
	UI——无编号信息帧		1	1	0	0	P	1 0 0
	UP——无编号探测		1	1	1	1	P	0 0 1
	RSET——复位		1	1	1	1	P/F	1 0 1
	XID——交换标识		1	1	0	0	F	1 1 0
		XID——交换标识	1	1	1	0	F	0 0 1
		UA——无编号确认	1	1	0	0	F	0 0 1
		CMDR——命令帧拒收	1	1	1	0	F	0 0 1

A发送编号为1的数据I帧

B发送编号为0的数据I帧，
并对A的第1帧进行确认

A发送编号为2的数据I帧，并
对B的第0帧进行捎带确认

A发送编号为3的数据I帧

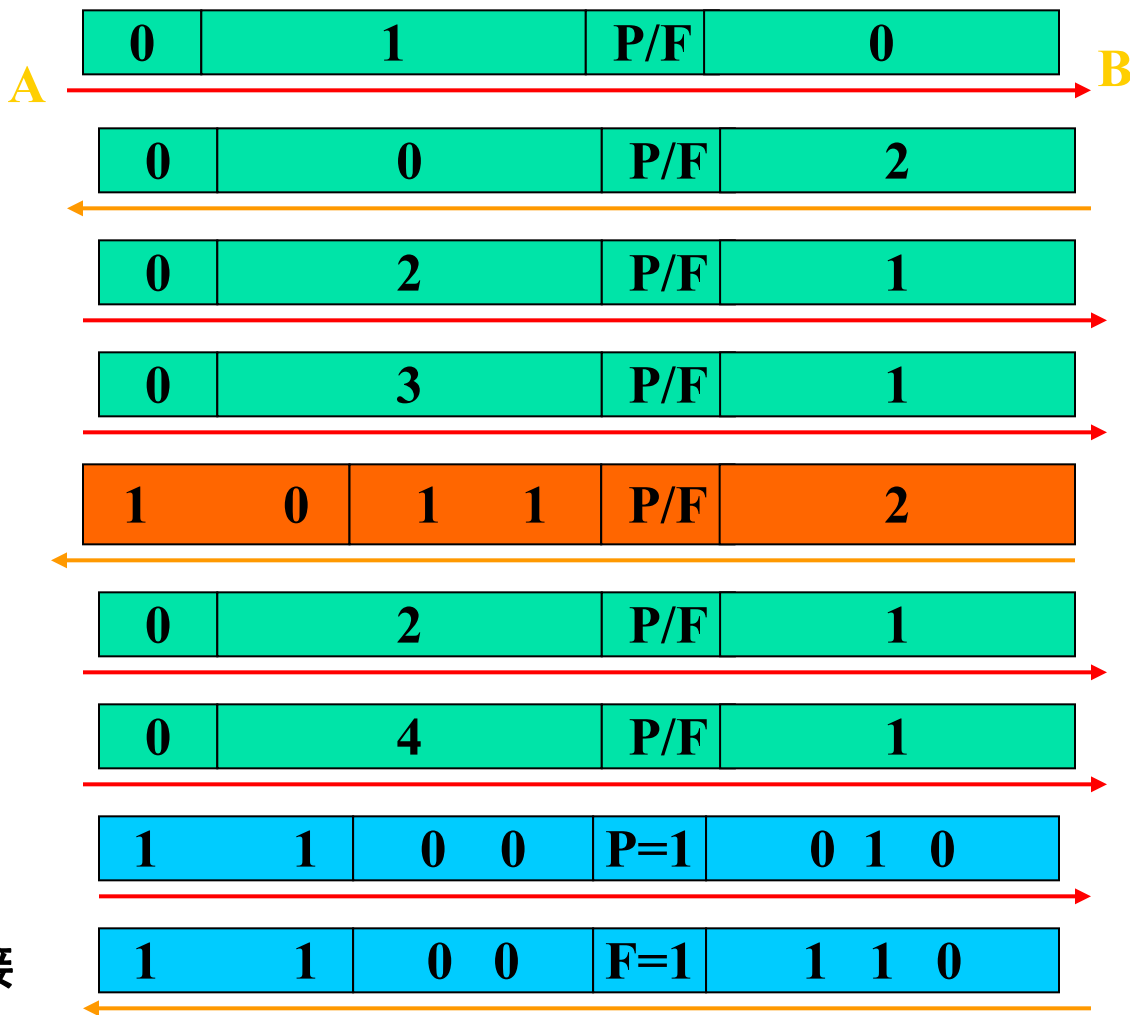
B发送监控S帧(SREJ)报告第
2帧出错，并要求选择重发

A重发编号为2的数据I帧

A继续发送编号为4的数据I帧

A发送无编号U帧请求断开连接

B发送无编号U帧确认断开连接



(4) 复合站工作流程

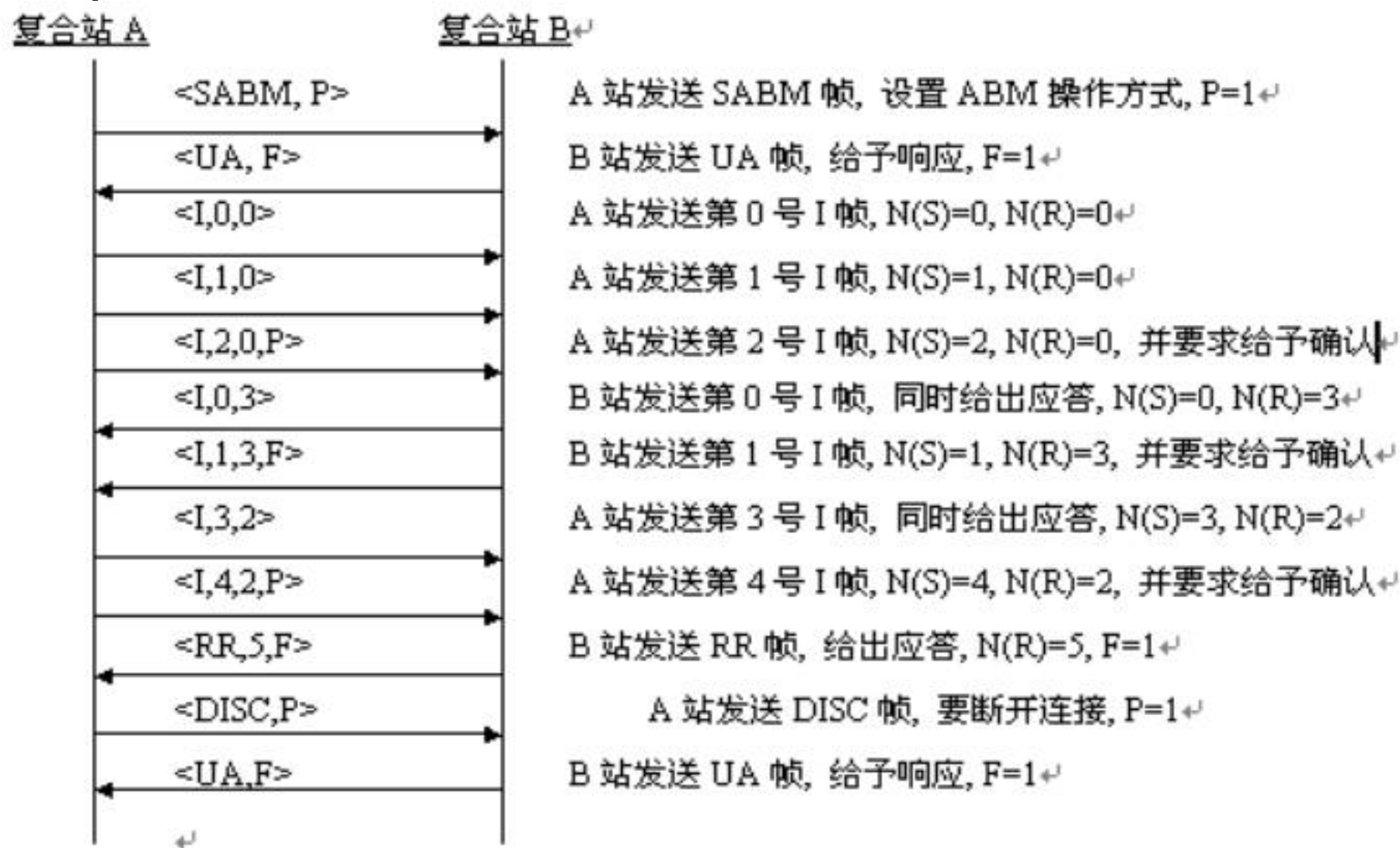
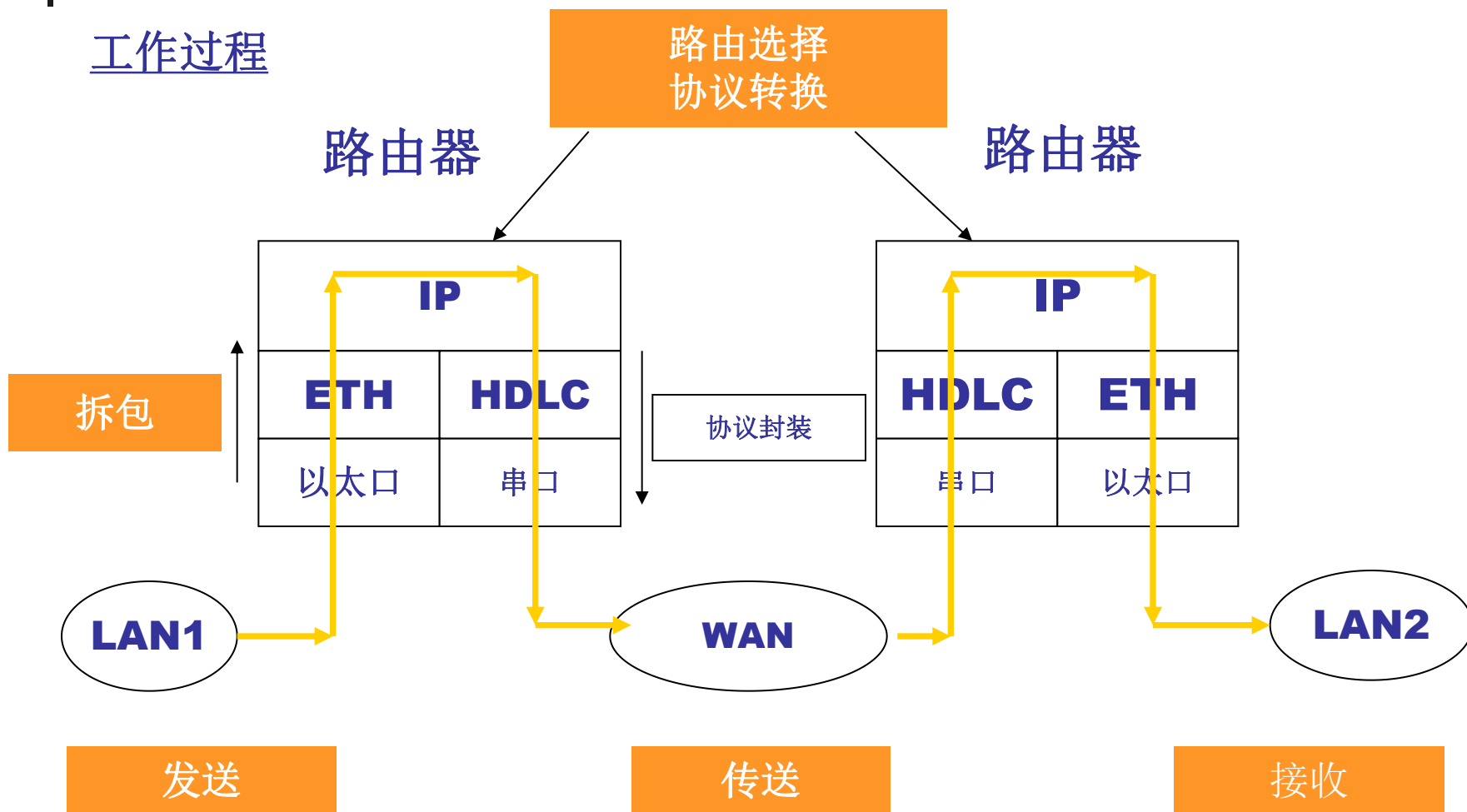


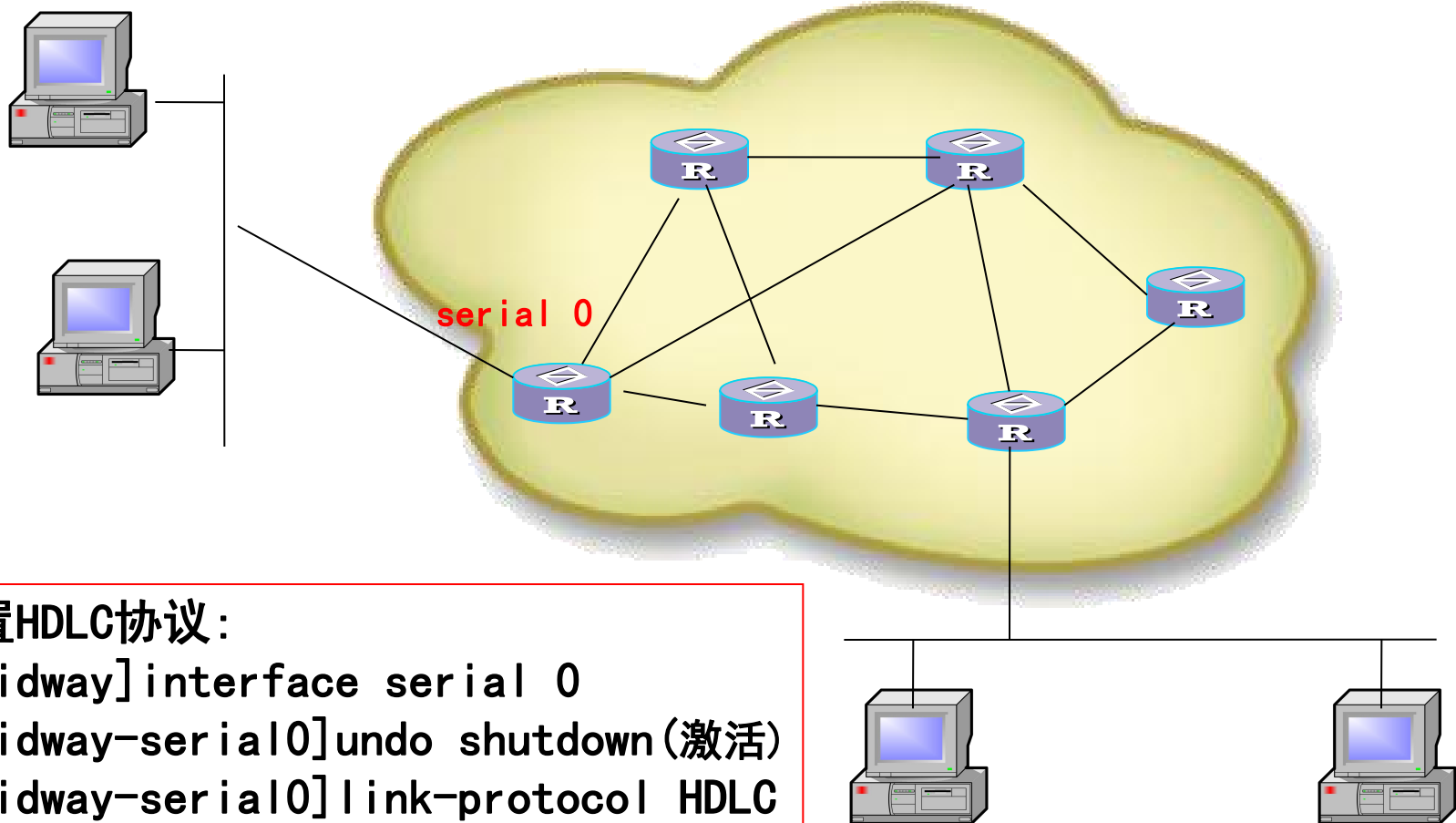
图 2.7 ABM 方式下的数据传输过程

3. 数据帧封装/拆封

工作过程



4. 实际工程中应用举例



配置HDLC协议：

```
[Quidway]interface serial 0
```

```
[Quidway-serial0]undo shutdown(激活)
```

```
[Quidway-serial0]link-protocol HDLC
```

6. 实际工程中应用举例

- [Quidway]display interface serial 0 （检查serial0运行状态）

Serial0 is UP, line protocol is UP

physical layer is synchronous, baudrate is 64000 bps,

Maximum Transmission Unit is 1500

Internet address is 10.0.1.2 255.255.255.0

Link-protocol is HDLC

5 minutes input rate 0.00 bytes/sec, 0.00 packets/sec

5 minutes output rate 0.00 bytes/sec, 0.00 packets/sec

Input queue :(size/max/drops)

0/75/0

FIFO queueing: FIFO

Output Queue :(size/max/drops)

0/75/0

4294967288 packets input, 0 bytes, 0 no buffers

0 packets output, 0 bytes, 0 no buffers

0 input errors, 0 CRC, 0 frame errors

0 overrunners, 0 aborted sequences, 0 input no buffers

DCD=DOWN DTR=DOWN DSR=UP RTS=DOWN CTS=UP



HDL C协议特点总结

- 适用于点-点(采用)或点对多点(已放弃)通信。
- 可用于半双工或全双工通信。
- 采用面向比特同步方式和滑窗技术传输, 传输效率高。
- 帧都有帧校验序列, 且按顺序编号, 可靠性较高。
- 采用比特位填充标识法成帧, 实现数据透明传输。
- 面向位流: 可传输任意长度和组合的二进制比特位。
- 采用统一的帧格式来传输数据、命令和响应, 非常利于程序的实现。

因此HDL C和相类似的协议获得了广泛的应用。



本节小结

- 1. HDLC 协议概述
- 2. HDLC协议设计
 - 2.1 HDLC 语法：帧结构
 - 2.2 HDLC 语义：字段含义
 - 2.3 HDLC 同步：时间发生次序
- 3. 数据帧封装/拆封
- 4. 工程应用



复习&预习&作业

- 本节内容

教材:PP36-PP41

- 下一节预习内容

教材: PP111-119

- 作业

教材:PP63:11,12,14,20.