



恒润科技
HIRAIN TECHNOLOGIES

CAN基础

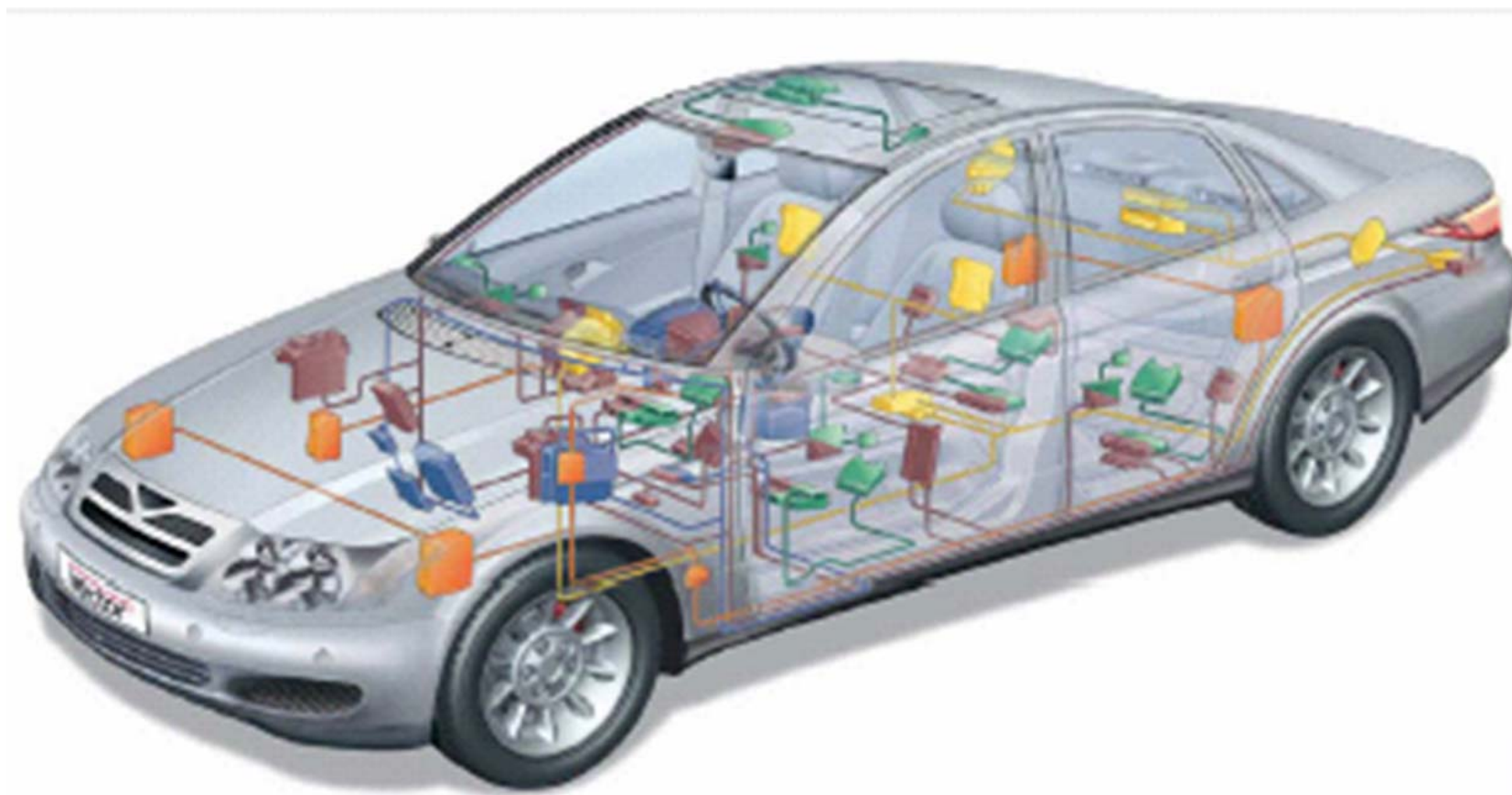
北京恒润科技有限公司

2008-7-5

- 概述
- 汽车总线与CAN标准
- CAN的通信机制
- 数据帧
- 错误检测与错误帧
- CAN的帧格式
- 位定时与同步
- 物理连接

■ CAN的起源

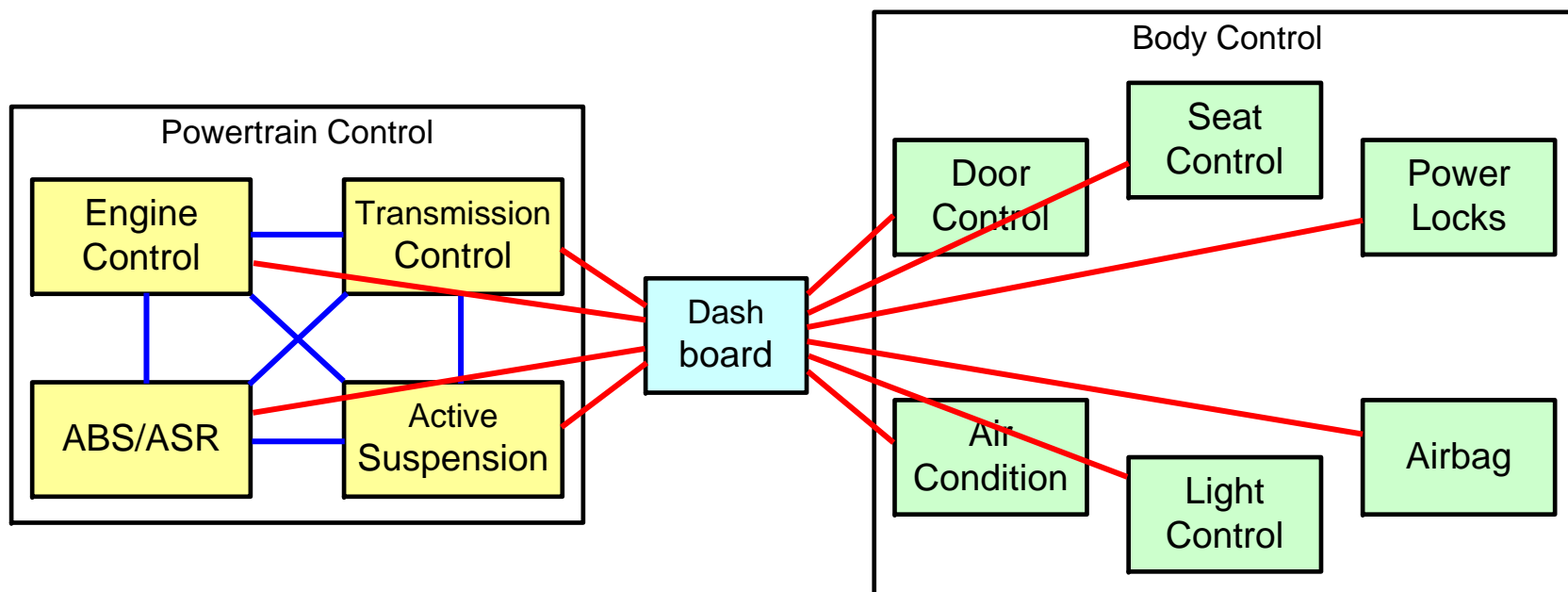
- ❖ CAN—Controller Area Network—是20世纪80年代初德国Bosch公司为解决现代汽车中众多控制单元、测试仪器之间的实时数据交换而开发的一种串行通信协议





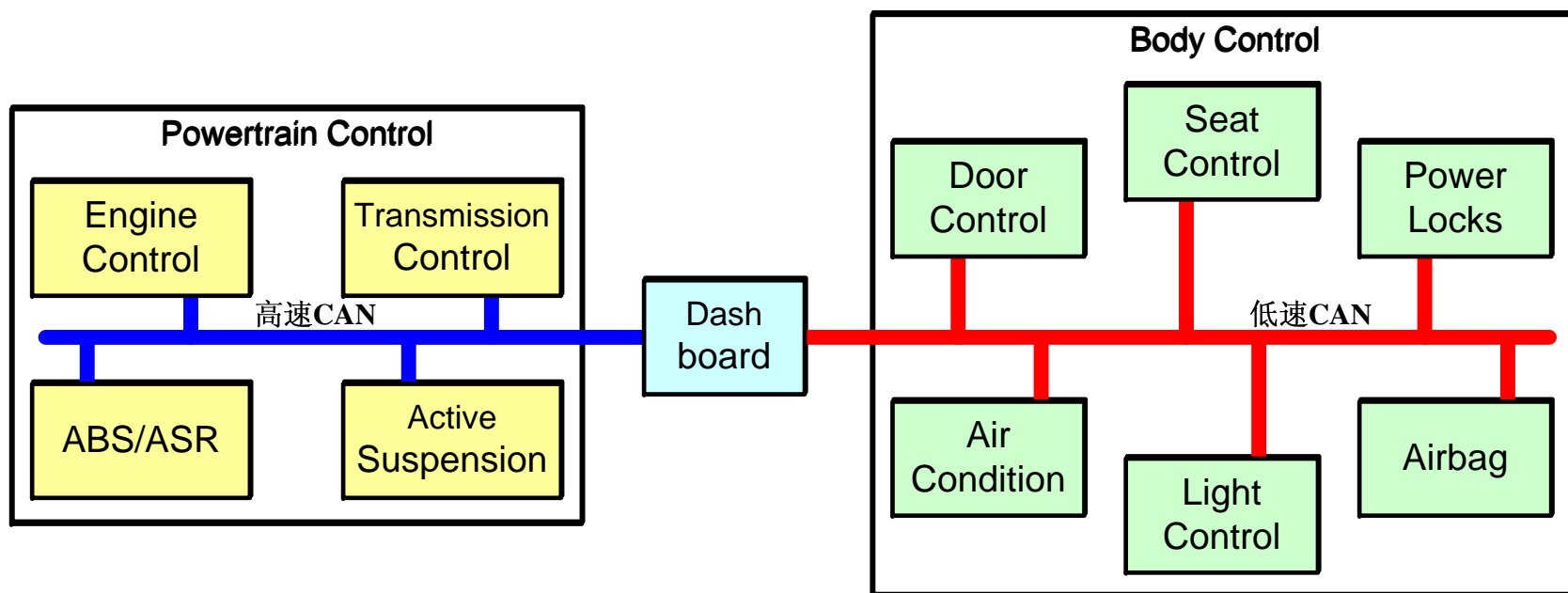
■ CAN的起源

❖ 传统的汽车线束连接



■ CAN的起源

❖ 汽车的CAN网络



■ CAN的历史

- ❖ 1983年，Bosch开始研究车上网络技术
- ❖ 1986年，Bosch在SAE大会公布CAN协议
- ❖ 1987年，Intel和Philips先后推出CAN控制器芯片
- ❖ 1991年，Bosch颁布CAN 2.0技术规范，CAN2.0包括A和B两个部分
- ❖ 1991年，CAN总线最先在Benz S系列轿车上实现

■ CAN的历史

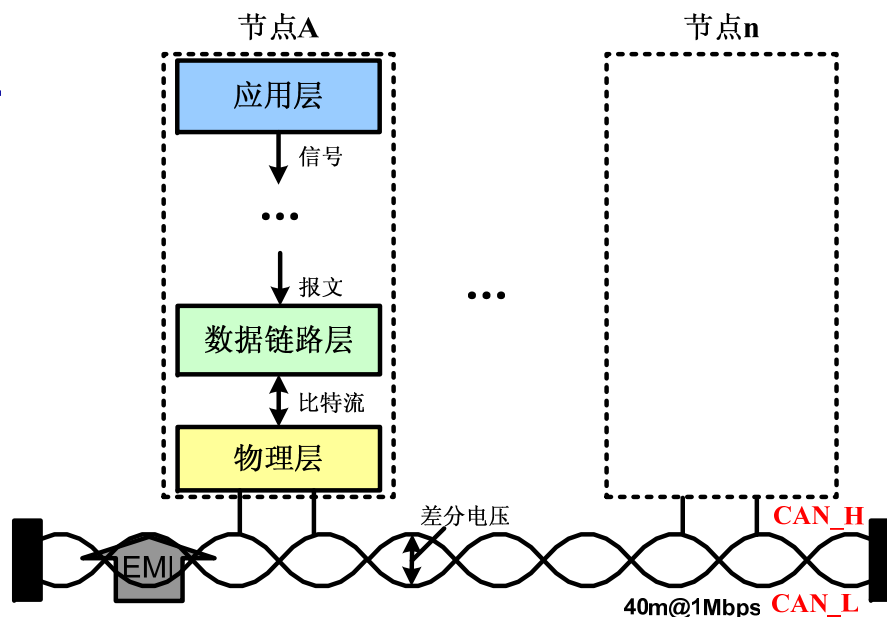
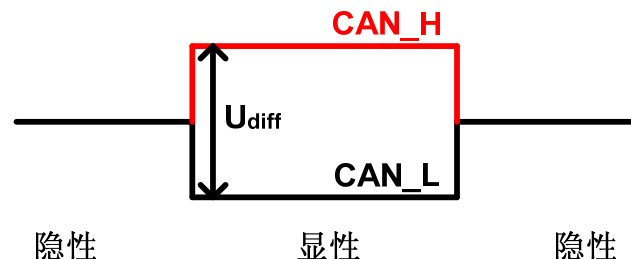
- ❖ 1993年，ISO颁布CAN国际标准 ISO-11898
- ❖ 1994年，SAE颁布基于CAN的J1939标准
- ❖ 2003年，Maybach发布带76个ECU的新车型（CAN，LIN，MOST）
- ❖ 2003年，VW发布带35个ECU的新型Golf

.....

- ❖ 未来，CAN总线将部分被FlexRay所取代，但CAN总线将仍会被持续应用相当长的时间

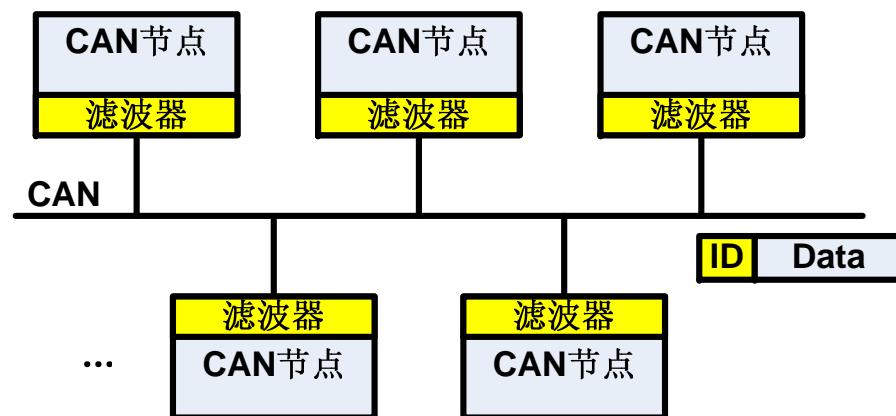
■ CAN的特性

- ❖ 采用**双线差分信号**
- ❖ 协议本身对**节点的数量没有限制**，**总线上节点的数量可以动态改变**
- ❖ **广播**发送报文，
报文可以被所有
节点同时接收



■ CAN的特性

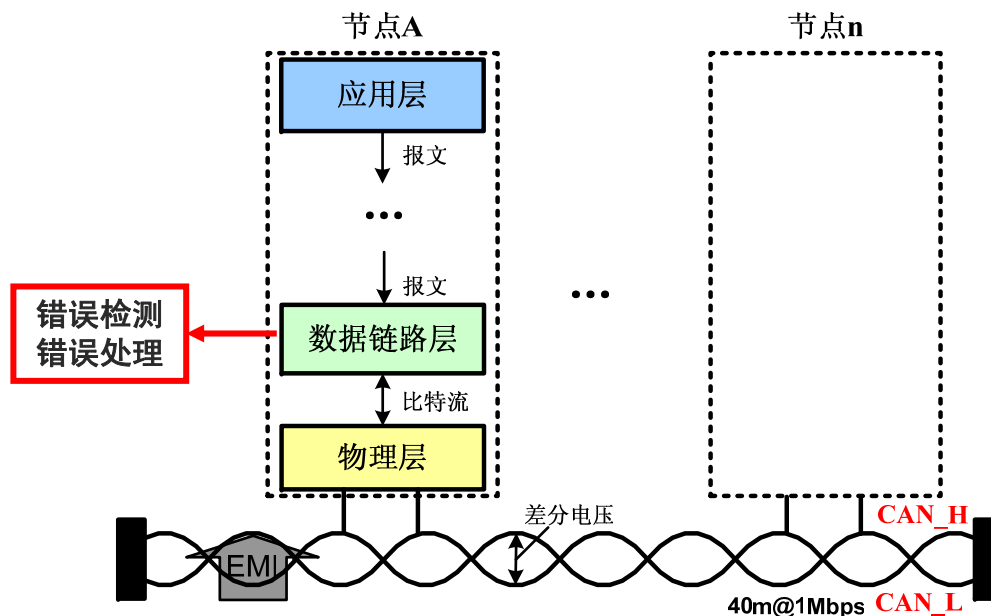
- ❖ 多主站结构，各节点平等，优先权由报文ID确定
- ❖ 每个报文的内容通过标识符识别，标识符在网络中是唯一的
 - 标识符描述了数据的含义
 - 某些特定的应用对标识符的分配进行了标准化
- ❖ 根据需要可进行相关性报文过滤



■ CAN的特性

❖ 保证系统数据一致性

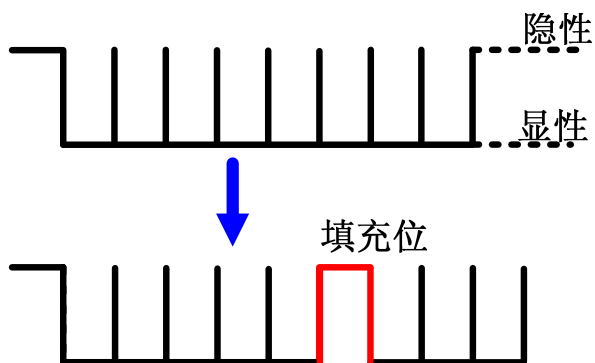
- CAN提供了一套复杂的**错误检测与错误处理**机制，比如CRC检测、接口的抗电磁干扰能力、错误报文的自动重发、临时错误的恢复以及永久错误的关闭



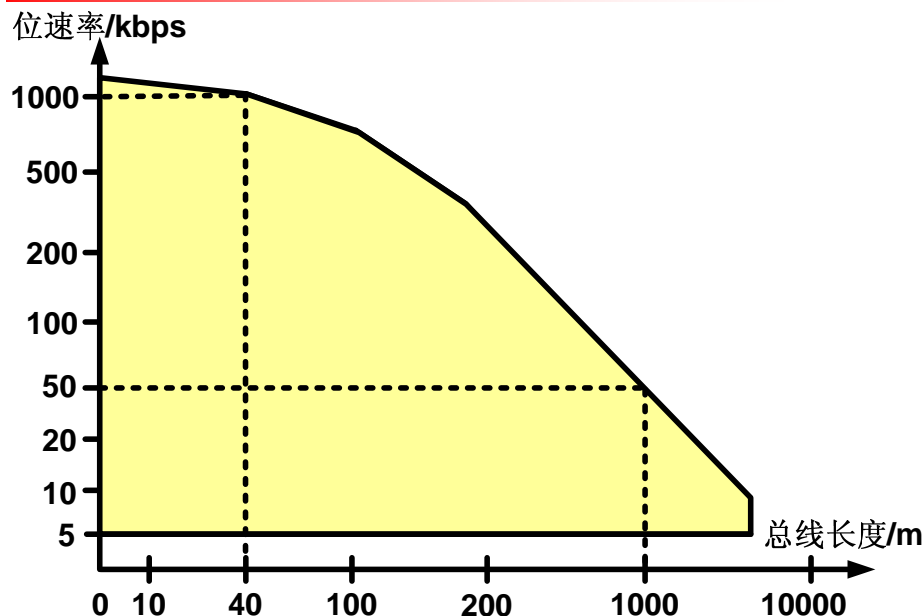
■ CAN的特性

- ❖ 使用双绞线作为总线介质，传输速率可达1Mbps，总线长度 ≤ 40 米
- ❖ 采用NRZ和位填充的位编码方式

NRZ和位填充



位速率与总线长度的关系



■ CAN的特性

- ❖ 总线访问—非破坏性仲裁的载波侦听多路访问/冲突避免**CSMA/CA** (Carrier Sense Multiple Access/Collision Avoidance)
 - **载波侦听(CS)**: 总线上各个节点在发送数据前都要侦听总线的通信状态
 - 总线有通信→不发送数据, 等待网络空闲
 - 总线空闲→立即发送已经准备好的数据
 - **多路访问(MA)**: 如果总线空闲, 则在同一时刻多个节点可同时访问总线(向总线发送数据)
 - **冲突避免(CA)**: 节点在发送数据过程中要不停地检测发送的数据, 确定是否与其它节点数据发生冲突, 并通过**非破坏性仲裁机制避免冲突**

- 概述
- 汽车总线与CAN标准
- CAN的通信机制
- 数据帧
- 错误检测与错误帧
- CAN的帧格式
- 位定时与同步
- 物理连接



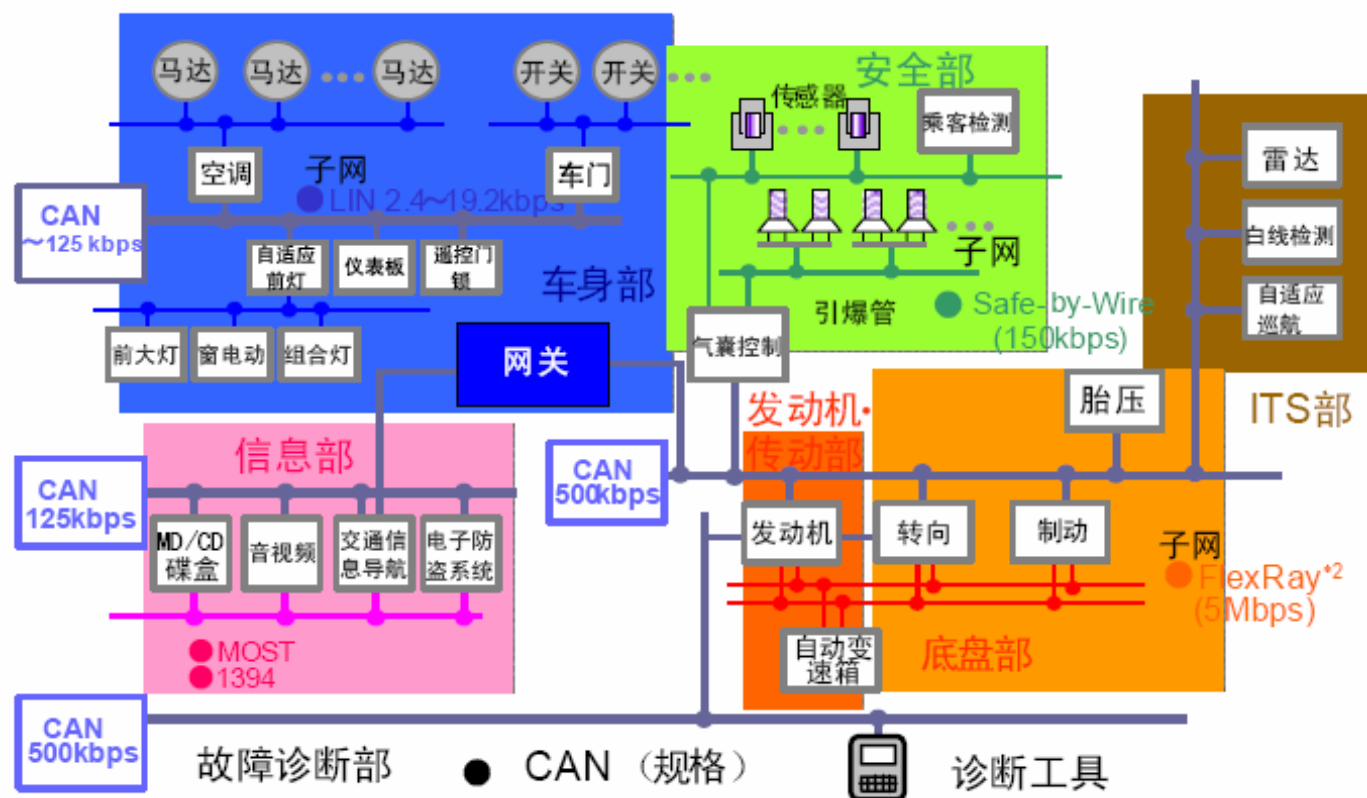
■ 汽车总线

❖ 汽车总线的分类

类别	位速率 /kbps	应用场合	应用范围	协议
A	~10	车身系统	电动门窗、座椅调节、灯光照明控制等	■LIN ■CAN
B	10~125	状态系统	电子仪表、驾驶信息、故障诊断、安全气囊、自动空调等	■J1850 ■VAN ■CAN
C	125~1000	实时控制系统	发动机控制、变速控制、ABS、悬架控制、转向控制等	■CAN
D	1000~	多媒体系统		■MOST ■FlexRay ■D2B ■IEEE1394

■ 汽车总线

❖ 汽车总线的应用



from Renesas

■ CAN标准

❖ CAN与OSI参考模型

OSI参考模型



- CAL, CANopen (CiA)
- DeviceNet (ODVA)
- SDS (Honeywell)
- NMEA-2000(NMEA)
- J1939(SAE)

汽车和工业自动化领域广泛应用



□ LLC, Logical Link Control

逻辑链路控制

□ MAC, Medium Access Control

媒介访问控制

□ PLS, Physical Signaling Sublayer

物理信令子层

□ PMA, Physical Medium Attachment

物理介质连接

□ MDI, Medium Dependent Interface

介质相关接口

■ CAN标准

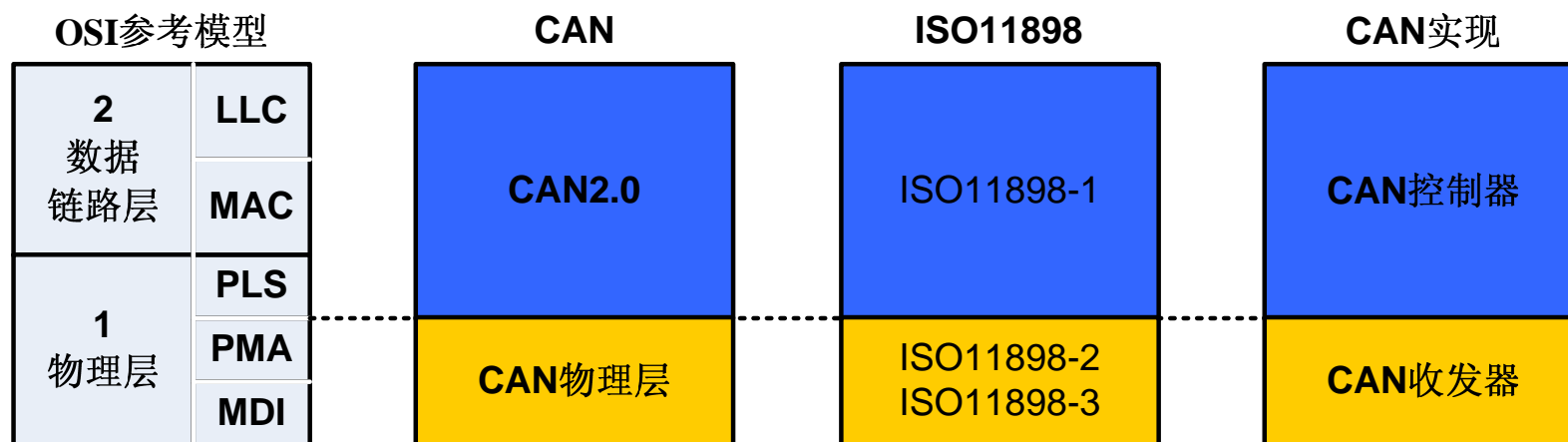
❖ CAN2.0版本

- ❑ 2.0A—将29位ID视为错误
- ❑ 2.0B被动—忽略29位ID的报文
- ❑ 2.0B主动—可处理11位和29位两种ID的报文

	11位ID数据帧	29位ID数据帧
CAN 2.0B Active	OK	OK
CAN 2.0B Passive	OK	容纳
CAN 2.0A	OK	总线错误

■ CAN标准

❖ ISO11898

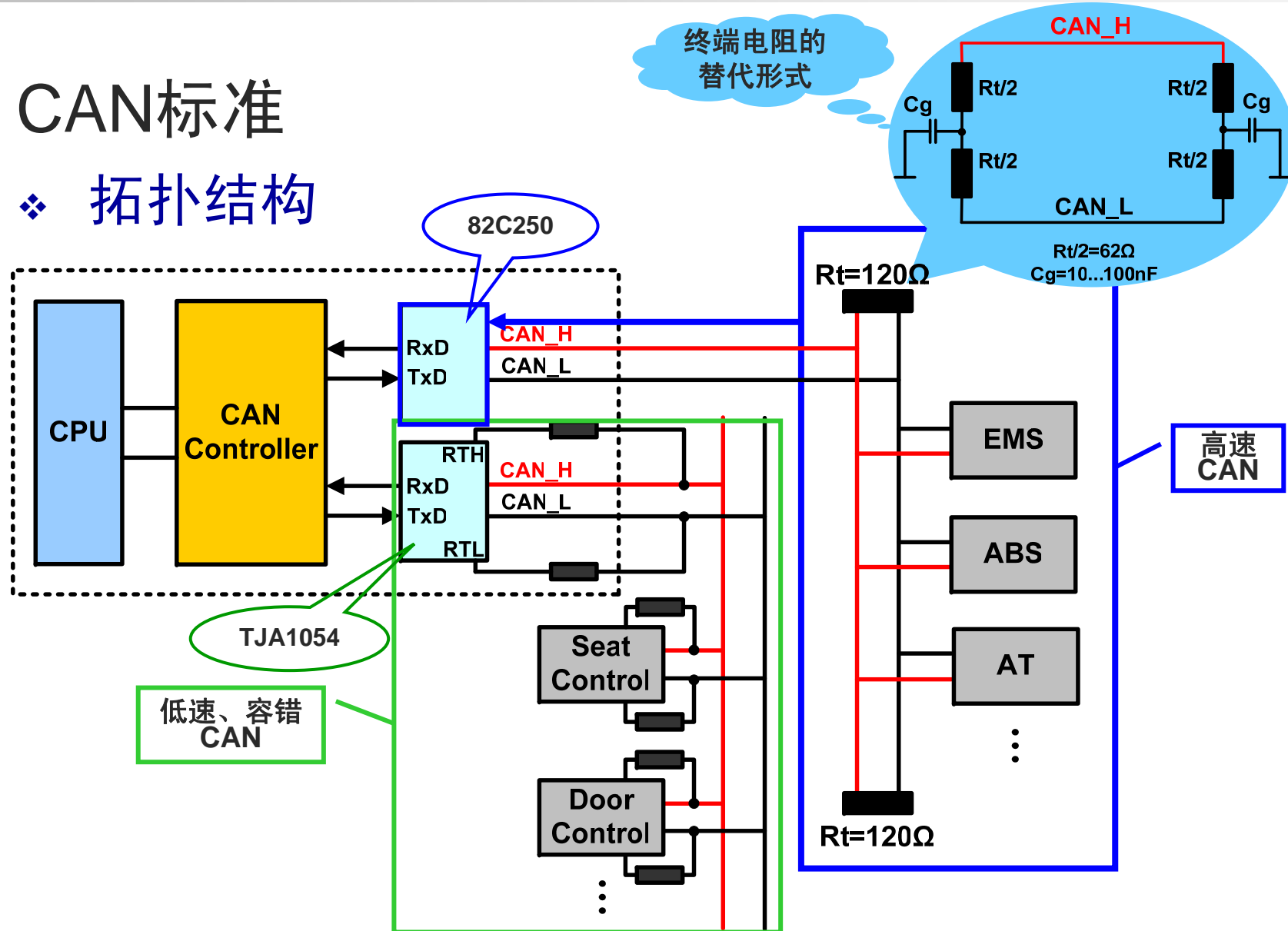


- ❑ ISO 11898-1: 2003 Road vehicles -- Controller area network (CAN) -- Part 1: Data link layer and physical signalling
- ❑ ISO 11898-2: 2003 Road vehicles -- Controller area network (CAN) -- Part 2: High-speed medium access unit
- ❑ ISO 11898-3: 2006 Road vehicles -- Controller area network (CAN) -- Part 3: Low-speed, fault-tolerant, medium-dependent interface
- ❑ ISO 11898-4: 2004 Road vehicles -- Controller area network (CAN) -- Part 4: Time-triggered communication
- ❑ ISO 11898-5 Road vehicles -- Controller area network (CAN) -- Part 5: High-speed medium access unit with low-power mode



■ CAN标准

❖ 拓扑结构

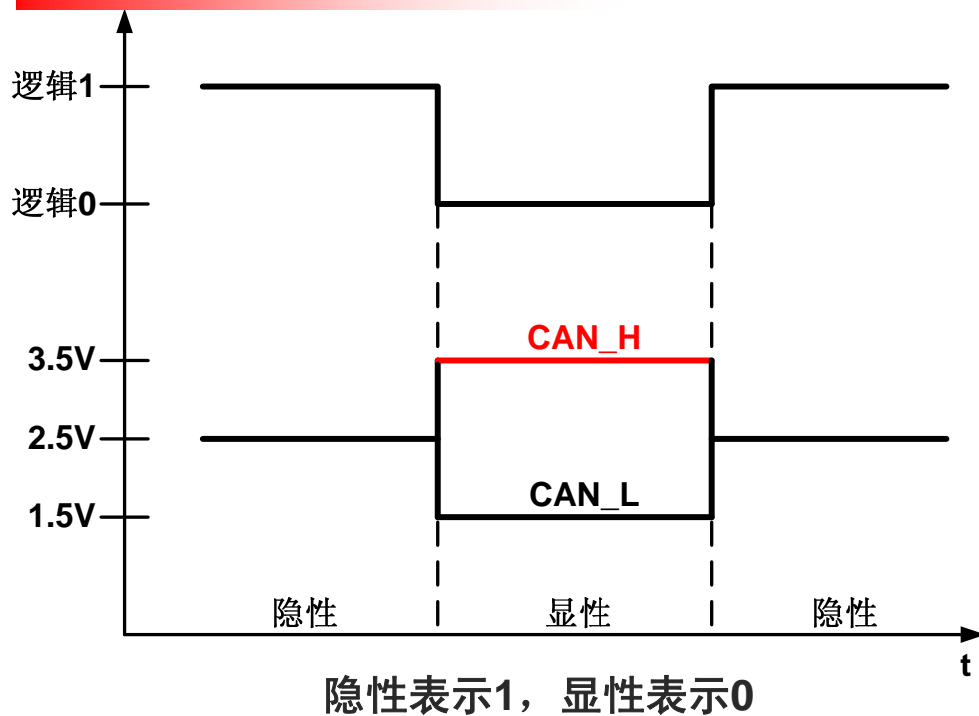


■ CAN标准

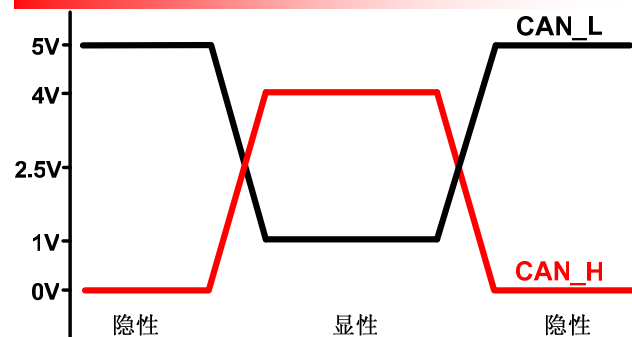
❖ 总线电平

差分电压

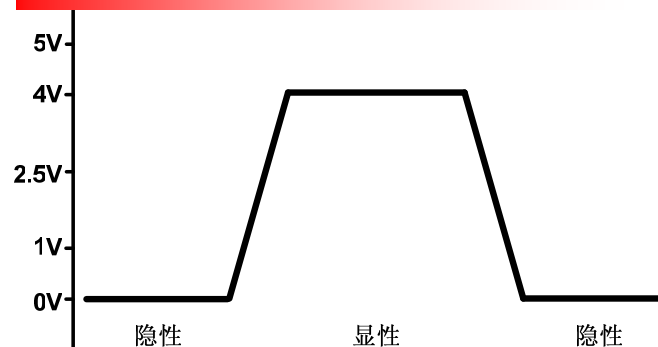
高速CAN电平



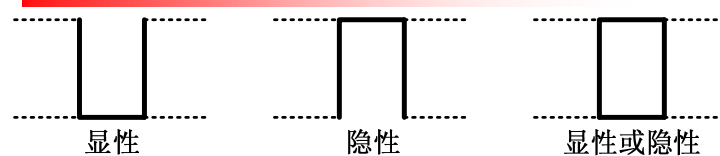
低速、容错CAN电平



单线CAN电平

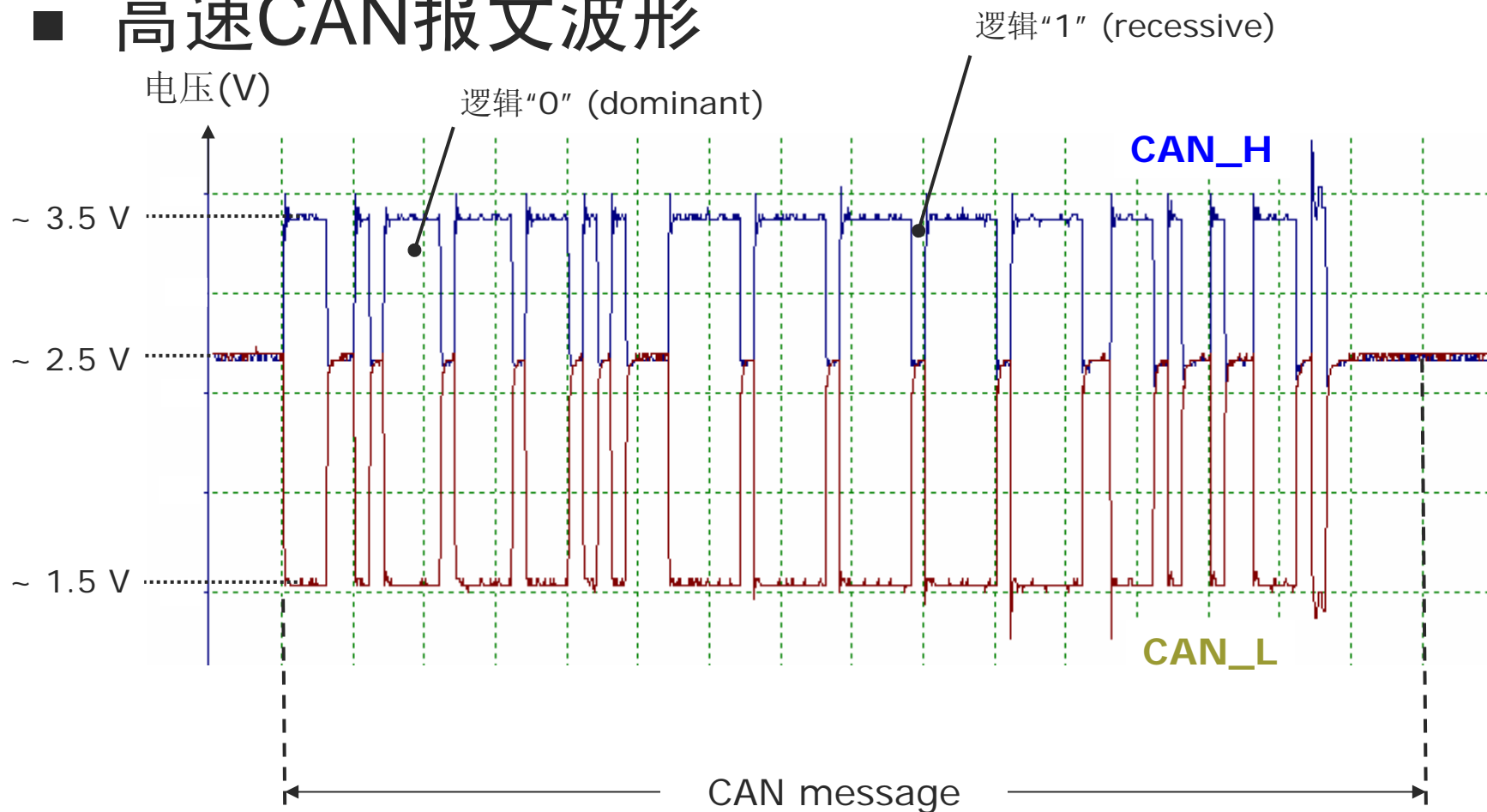


CAN逻辑表示





■ 高速CAN报文波形

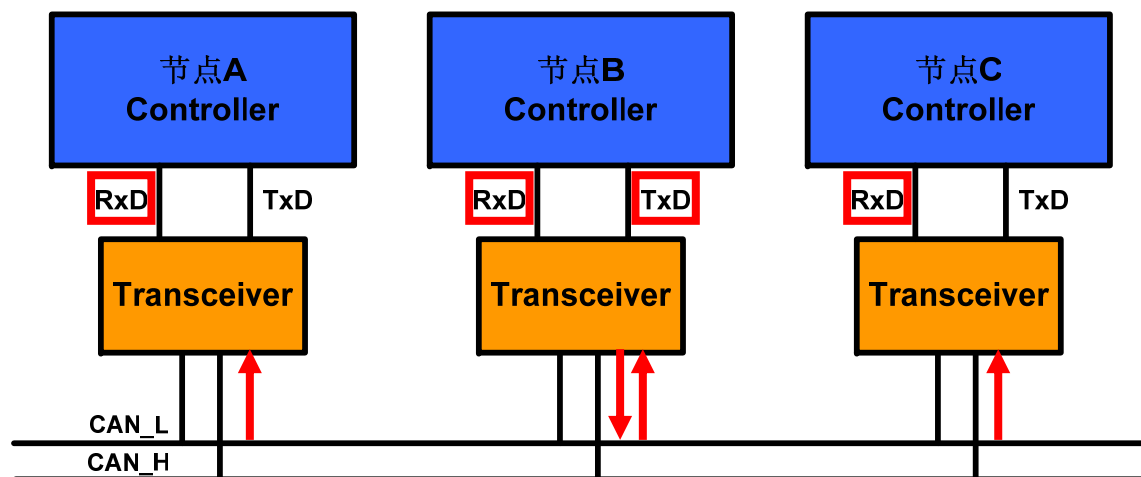


- 概述
- 汽车总线与CAN标准
- **CAN的通信机制**
- 数据帧
- 错误检测与错误帧
- CAN的帧格式
- 位定时与同步
- 物理连接

■ 报文发送

❖ 节点发送报文时要检测总线状态

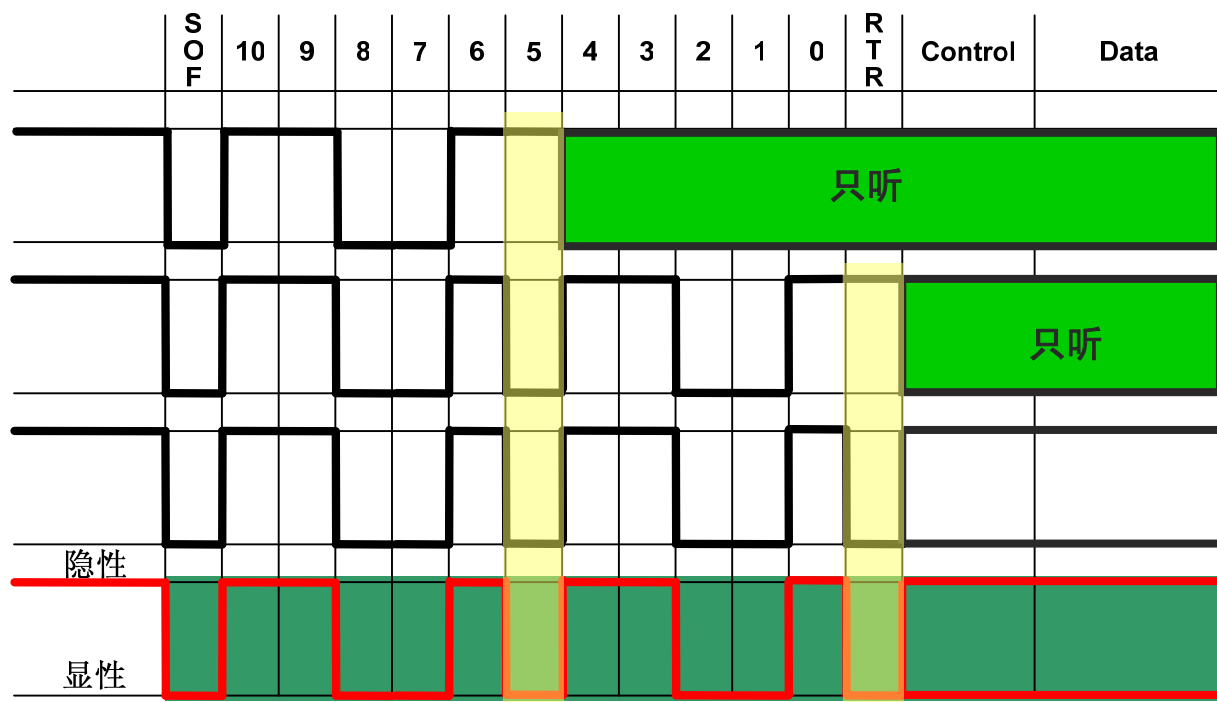
- 只有总线处于空闲，节点才能发送报文
- 在发送报文过程中进行“回读”，判断送出的位与回读的位是否一致



■ 报文发送

❖ “线”与“机制”

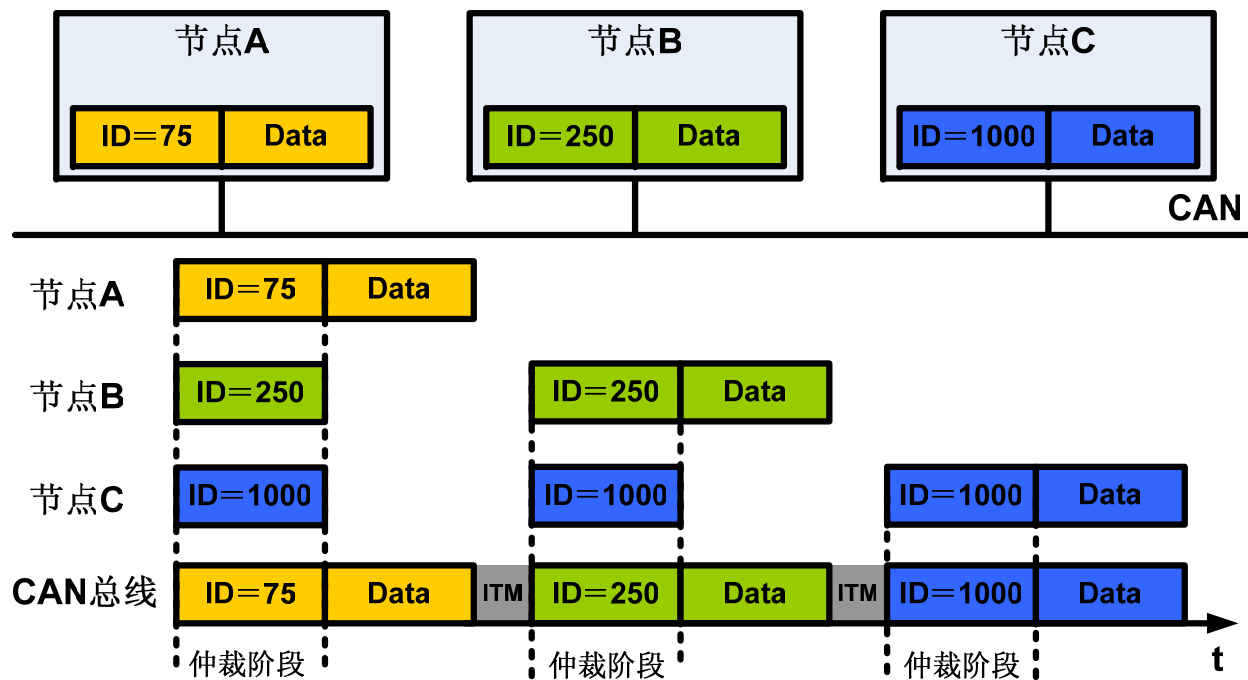
- 通过ID进行仲裁
- 显性位能够覆盖隐性位 → ID值越小，报文优先级越高



■ 报文发送

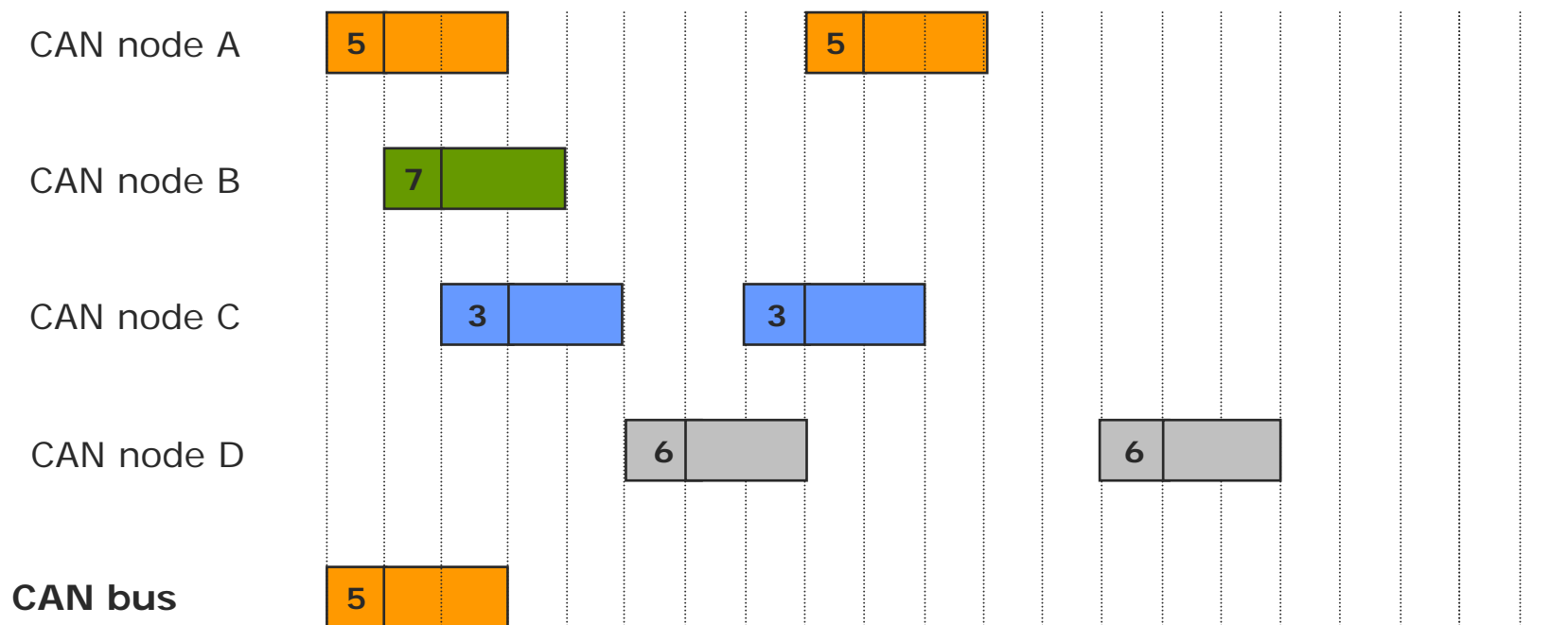
❖ 非破坏性仲裁

- 退出仲裁后进入“只听”状态
- 在总线空闲时进行报文重发



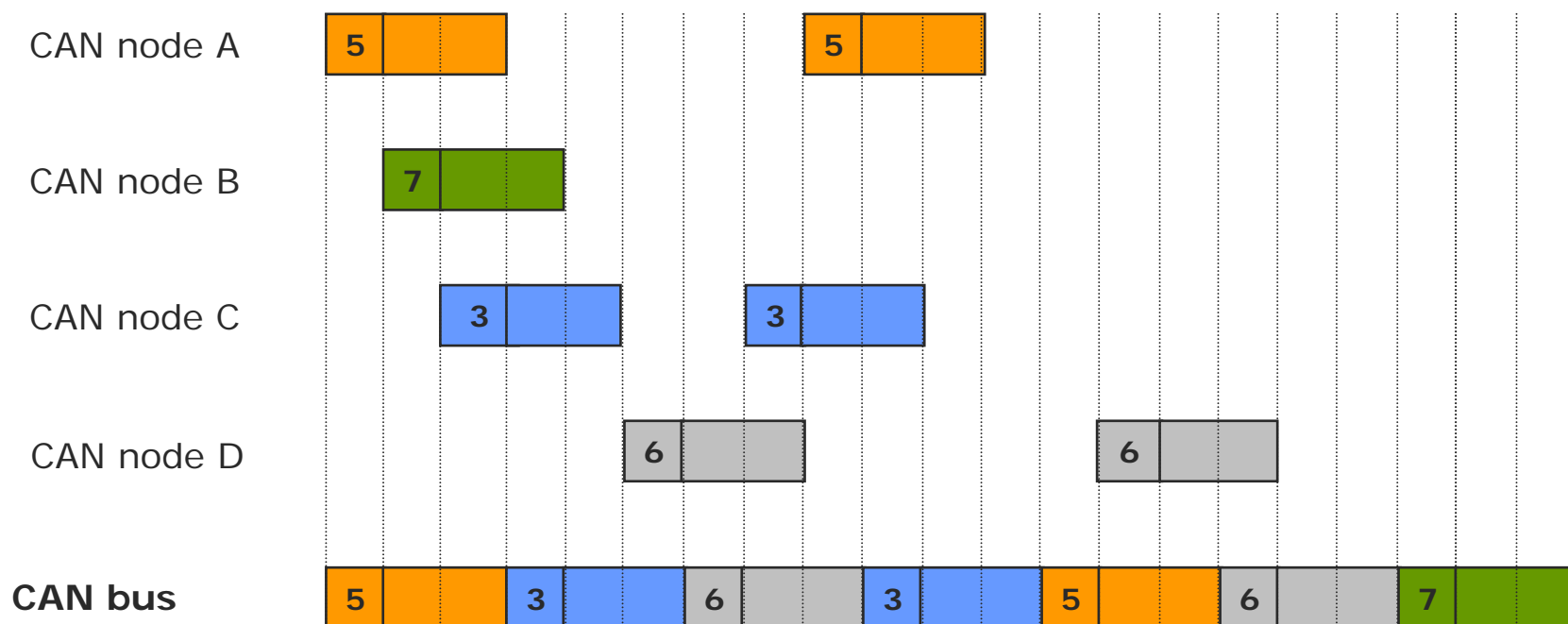
练习1：CAN总线访问仲裁机制

如图所示，A、B、C、D四个节点在不同的时刻分别往总线上发送ID为5、7、3、6的消息。请画出消息在总线上出现的顺序（假设每帧报文的传输时间占3格）。





练习1答案：CAN总线访问仲裁机制



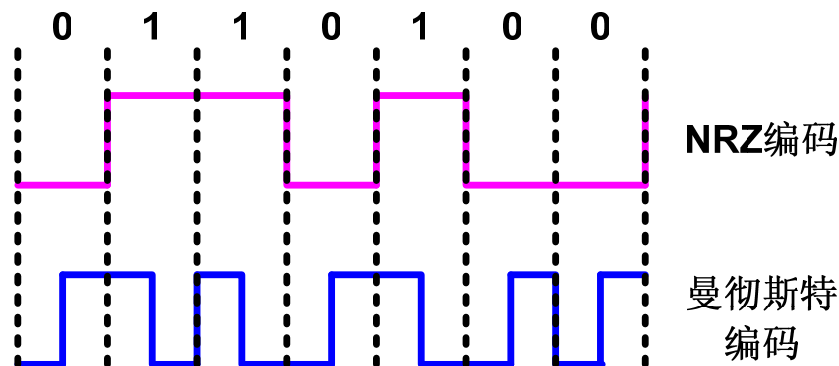


■ NRZ编码与位填充

❖ NRZ（非归零）编码

- NRZ编码确保报文紧凑→在相同带宽情况下，NRZ编码方式的信息量更大
- NRZ不能保证有足够的跳变延用于同步，容易带来节点间计时器误差的累积→**位填充**→保证有足够的跳变沿用于同步

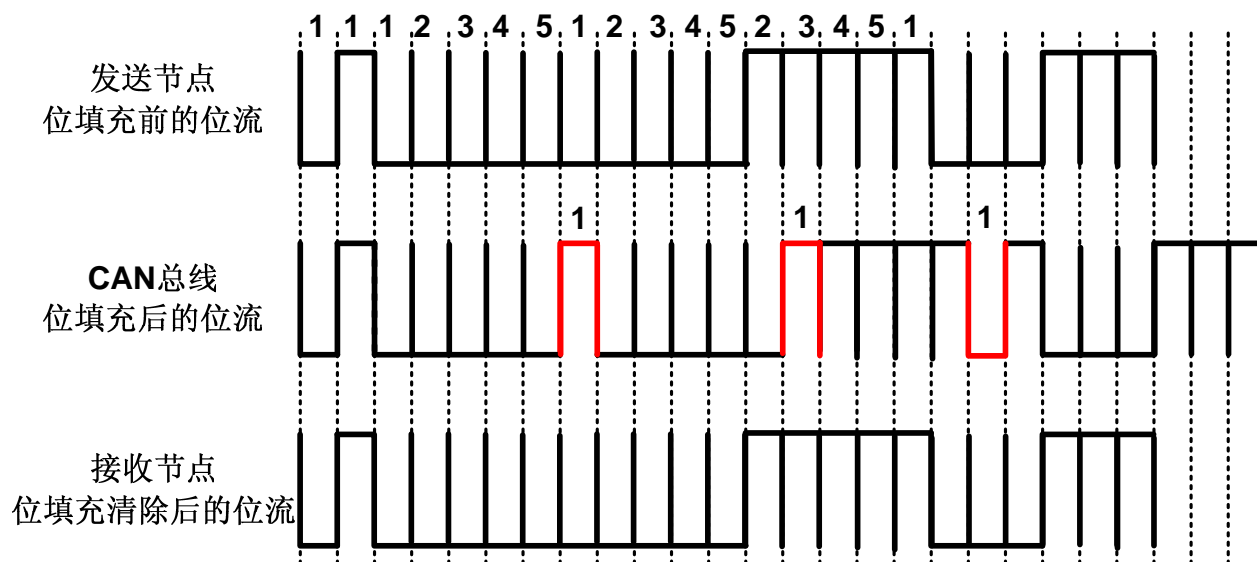
NRZ编码与曼彻斯特编码比较



■ NRZ编码与位填充

❖ 位填充

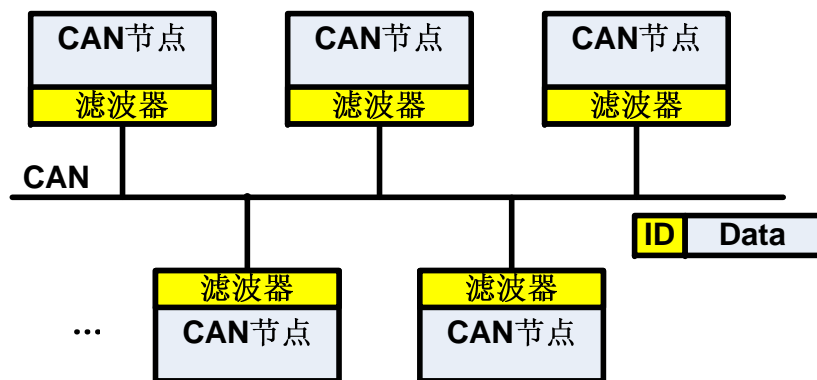
- 发送节点发送5个连续的相同极性位后，在位流中自动插入一个极性相反的位→位填充
- 接收节点对相同极性位的数量进行检测，从位流中将填充位去掉→清除填充



■ 报文接收过滤

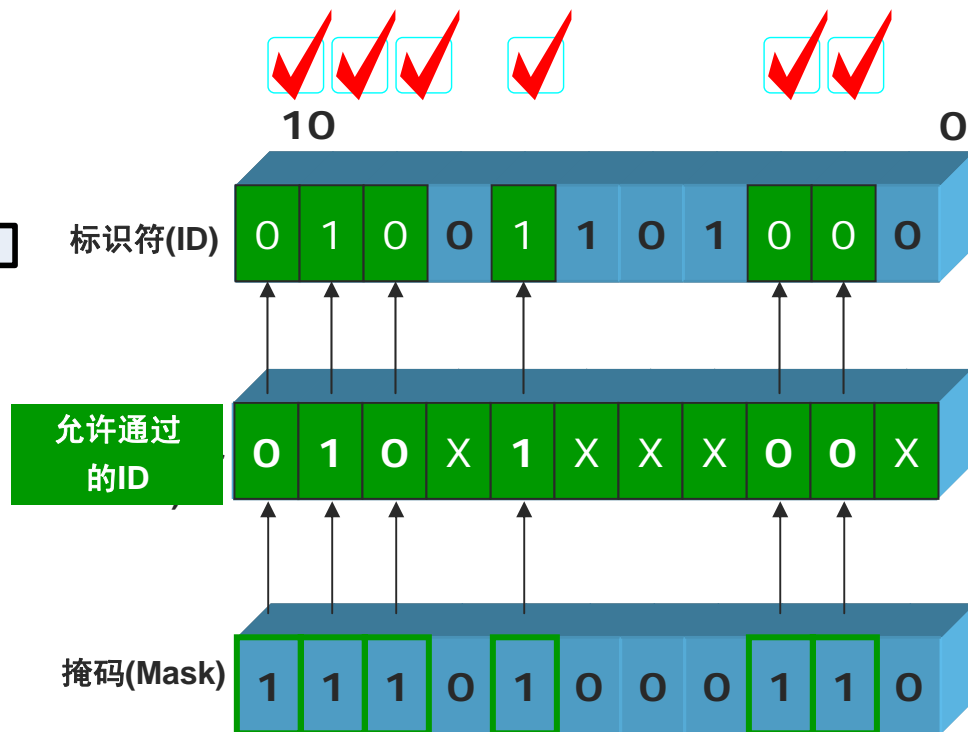
- ❖ 通过滤波器，节点可以对接收的报文进行过滤→如果报文相关就进行接收

接收滤波器(Acceptance Filter)



- 接收规则：比较消息ID与选择器中和接收过滤相关的位是否相同
- 接收过滤相关位：由掩码定义
 - ❖ 1 = 与消息过滤有关
 - ❖ 0 = 与消息过滤无关

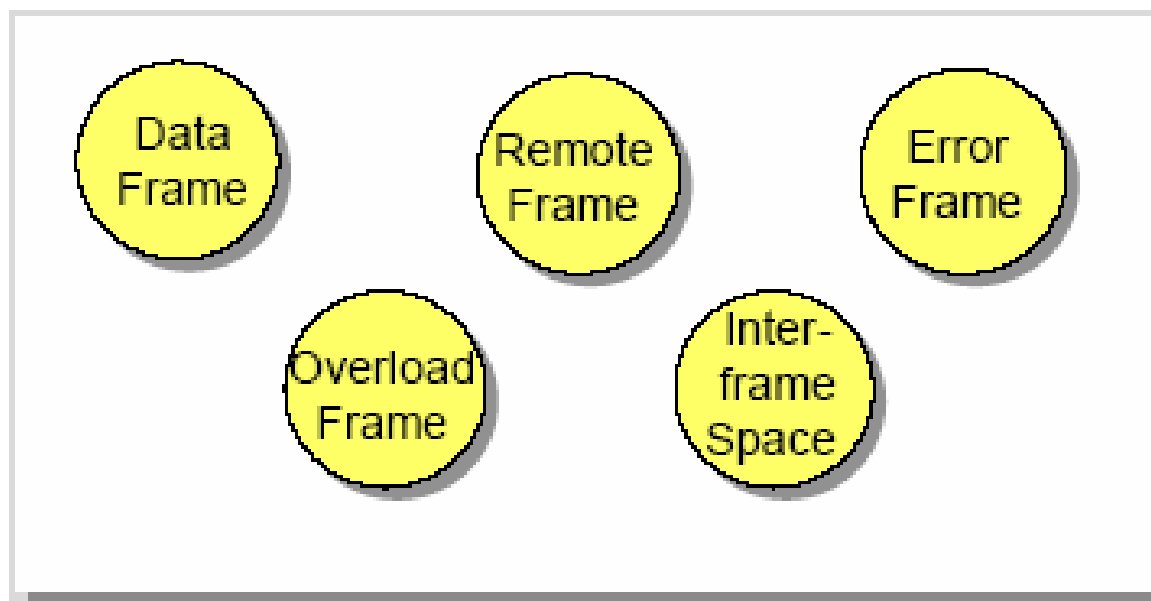
报文的过滤过程

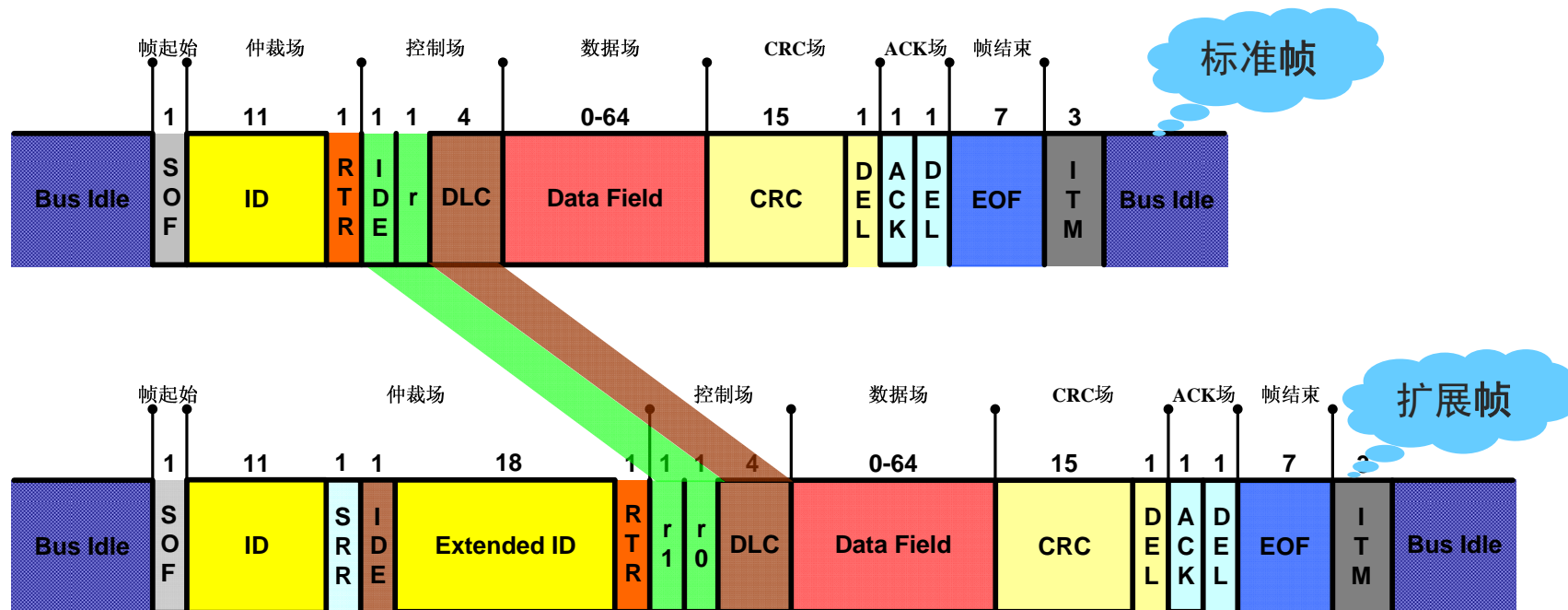


- CAN的发展
- 汽车总线与CAN标准
- CAN的通信机制
- 数据帧
- 错误检测与错误帧
- CAN的帧格式
- 位定时与同步
- 物理连接

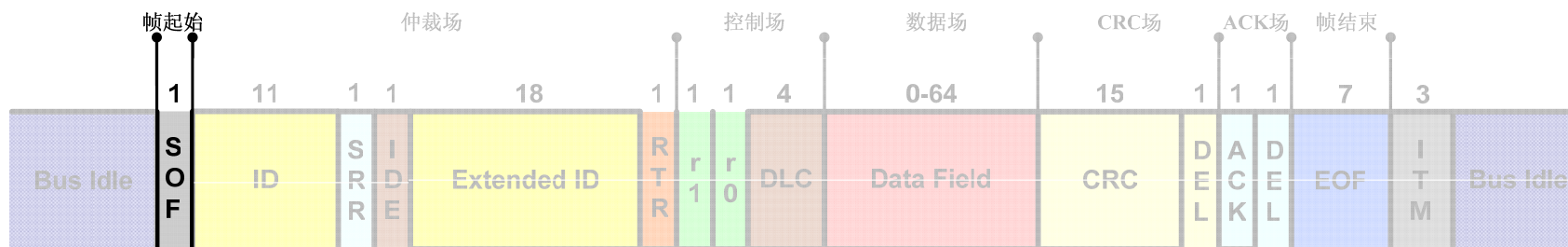


现有的帧格式



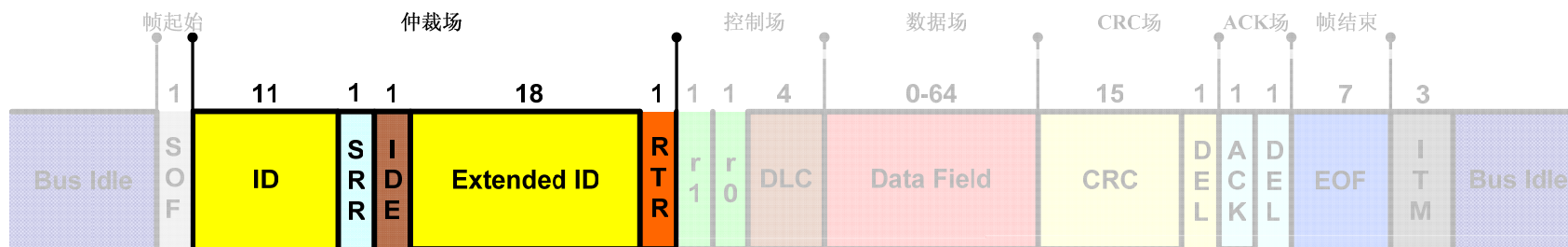


- | | | | |
|------------------------------------|--------|----------------------------------|--------|
| □ SOF, Start of Frame | 帧起始 | □ ACK, Acknowledgement | 应答 |
| □ RTR, Remote Transmission Request | 远程发送请求 | □ EOF, End of Frame | 帧结束 |
| □ IDE, Identifier Extension | 标识符扩展 | □ ITM, Intermission | 间歇场 |
| □ DLC, Data Length Code | 数据长度代码 | □ SRR, Substitute Remote Request | 代替远程请求 |

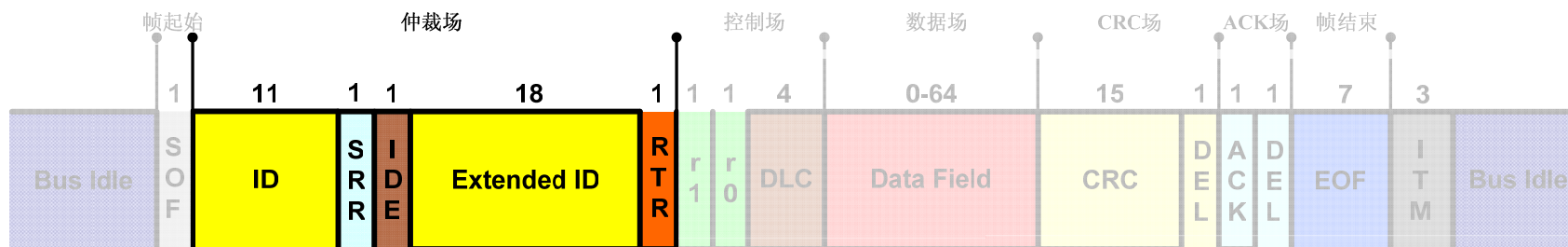


■ 帧起始

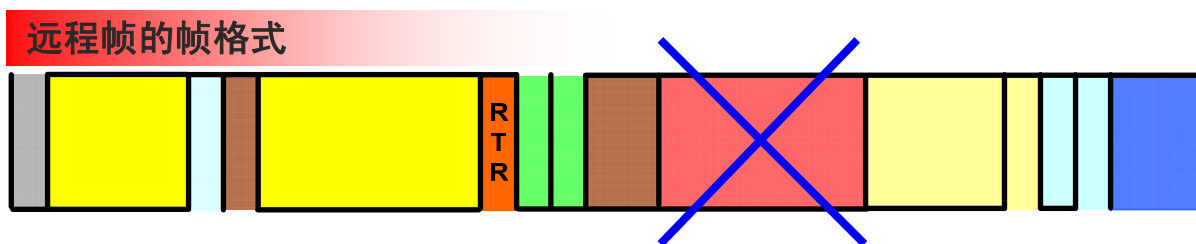
- ❖ 标识一个数据帧的开始，用于同步
- ❖ 一个显性位
- ❖ 只有在总线空闲期间节点才能够发送SOF

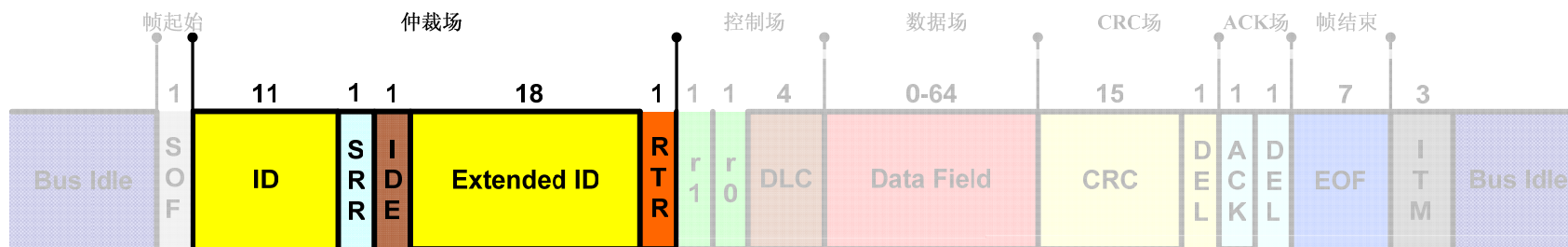


- 仲裁场→ID、RTR、IDE和SRR
- ID：标识符
 - ❖ 唯一确定一条报文，表明报文的含义和优先级
 - ❖ 标准帧→11位；扩展帧→29位

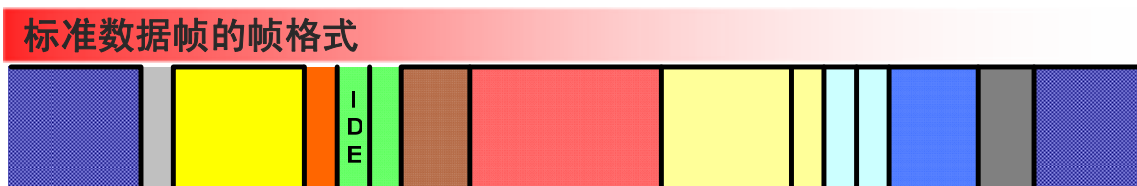


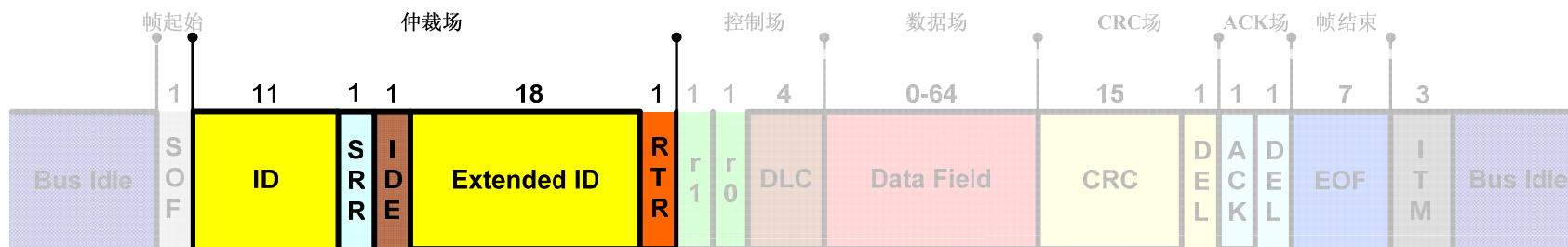
- 仲裁场→ID、RTR、IDE和SRR
- RTR—远程传送请求位
 - ❖ 数据帧，RTR=0;
 - ❖ 远程帧，RTR=1;



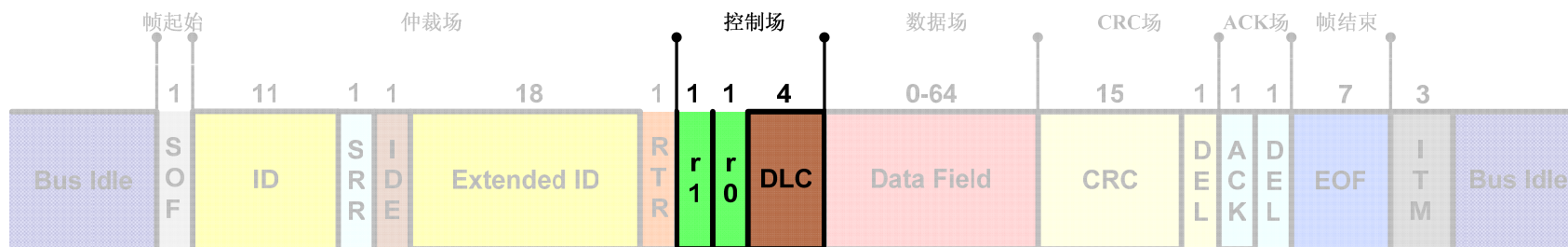


- 仲裁场→ID、RTR、IDE和SRR
- IDE—标识符扩展位
 - ❖ IDE=0→标准帧（11位ID）
 - ❖ IDE=1→扩展帧（29位ID）

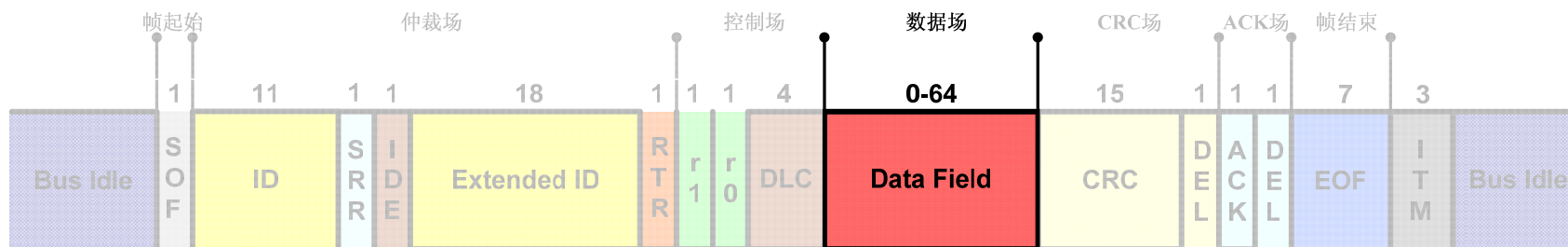




- 仲裁场→ID、RTR、IDE和SRR
- SRR—远程代替请求位
 - ❖ SRR=1

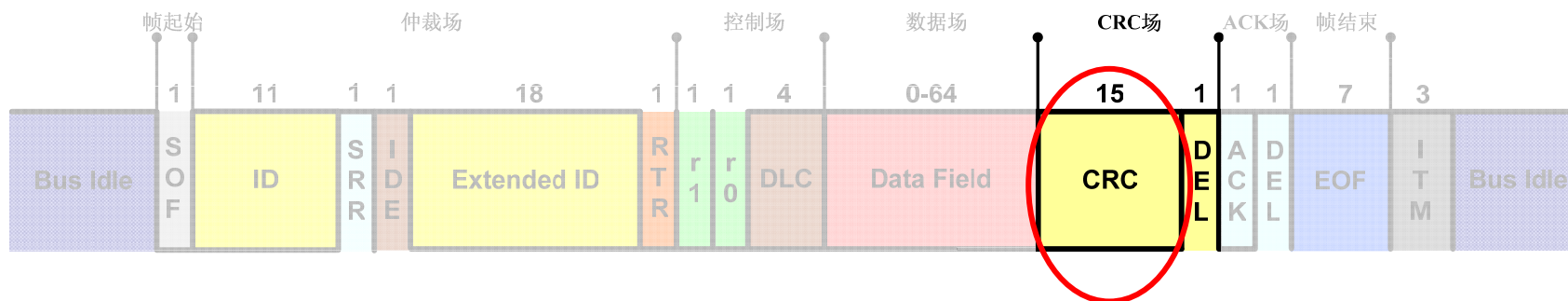


- 控制场→r0、r1和DLC
- r0、r1，保留位，置0
- DLC—数据长度码
 - ❖ 表示数据场的字节数
 - ❖ 有效的DLC：0-8；DLC9-15无效



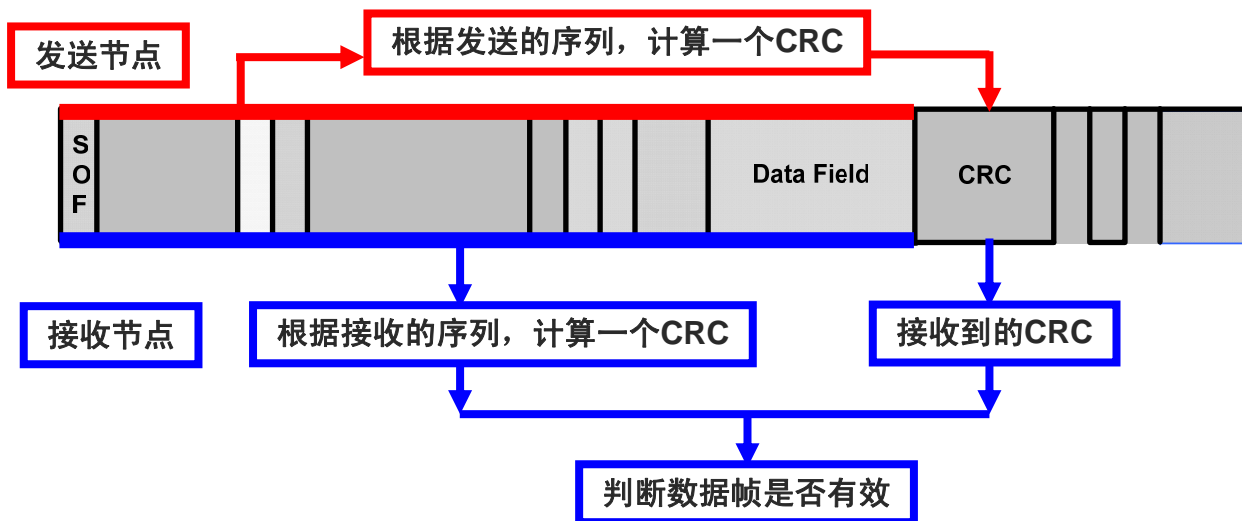
■ 数据场

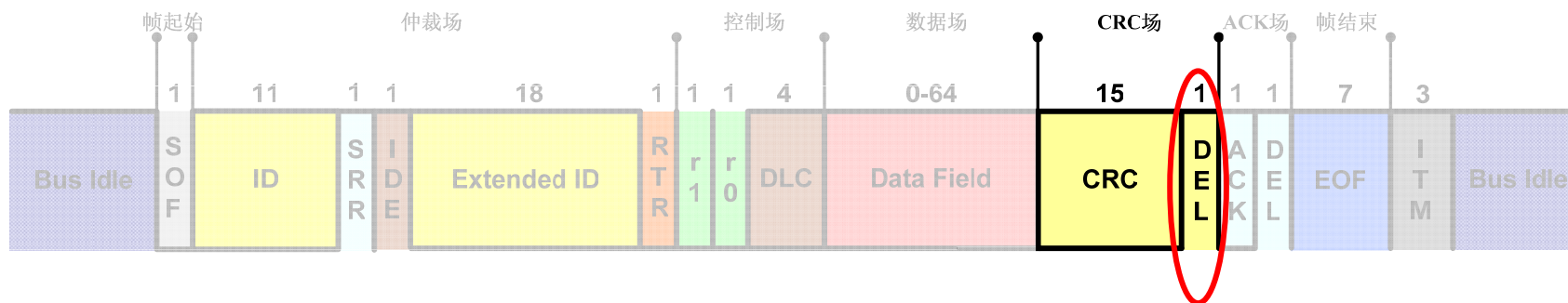
- ❖ 0-8个字节长度
- ❖ 包含CAN数据帧发送的内容



■ CRC

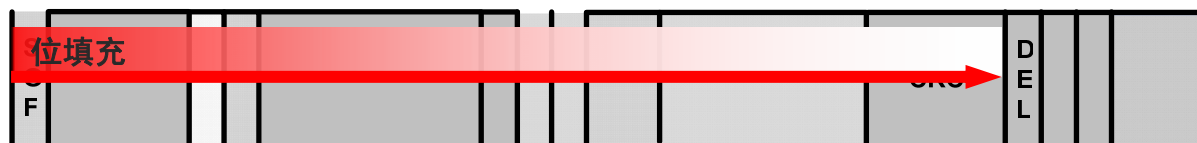
❖ 用于进行CRC校验

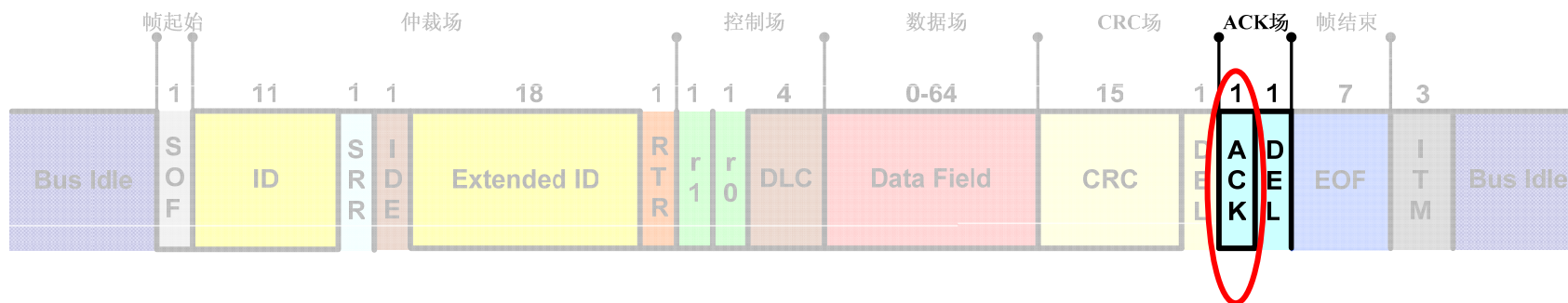




■ CRC界定符

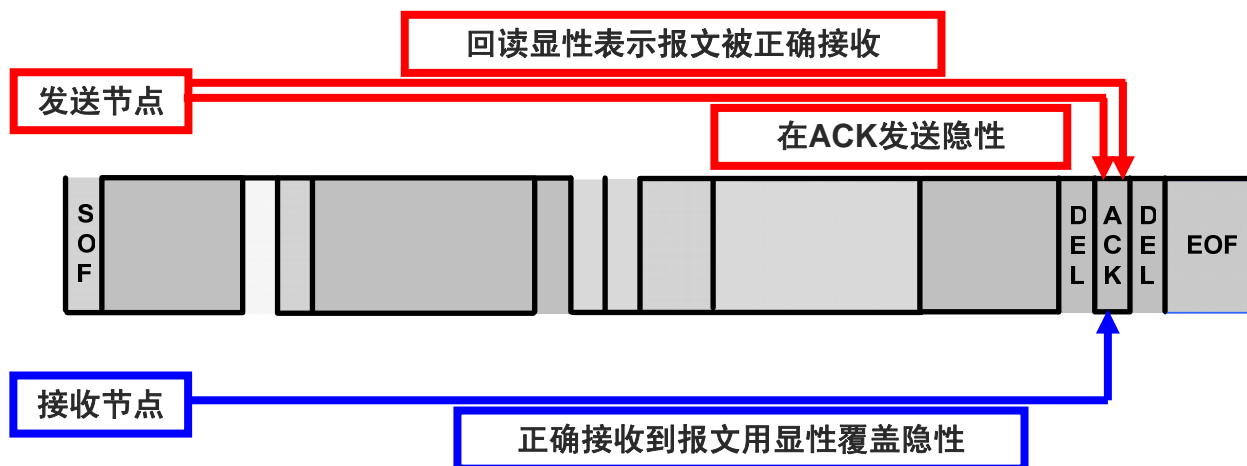
- ❖ 界定CRC序列
- ❖ 固定格式，1个隐性位
- ❖ CRC界定符之前进行位填充

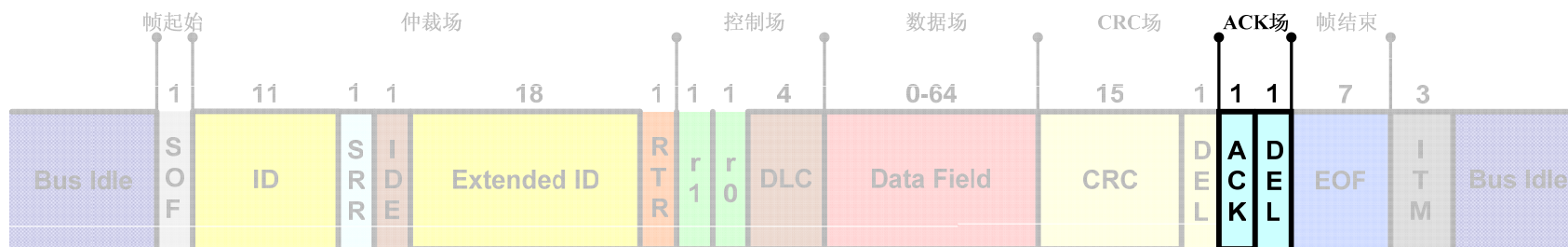




ACK

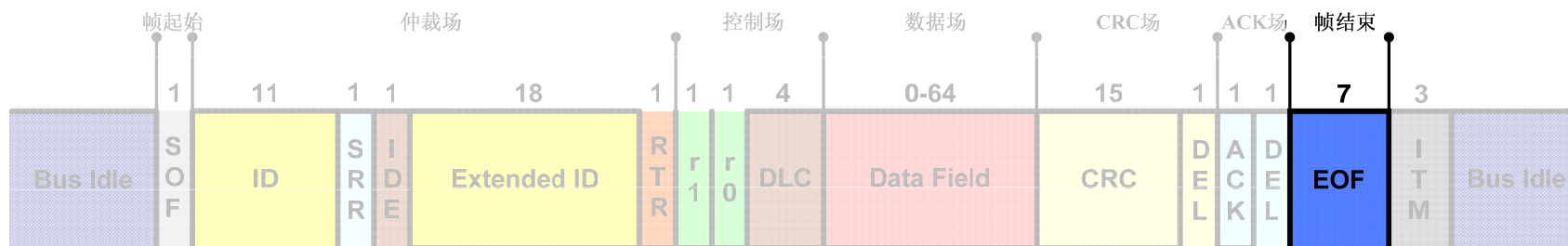
❖ 确定报文被至少一个节点正确接收





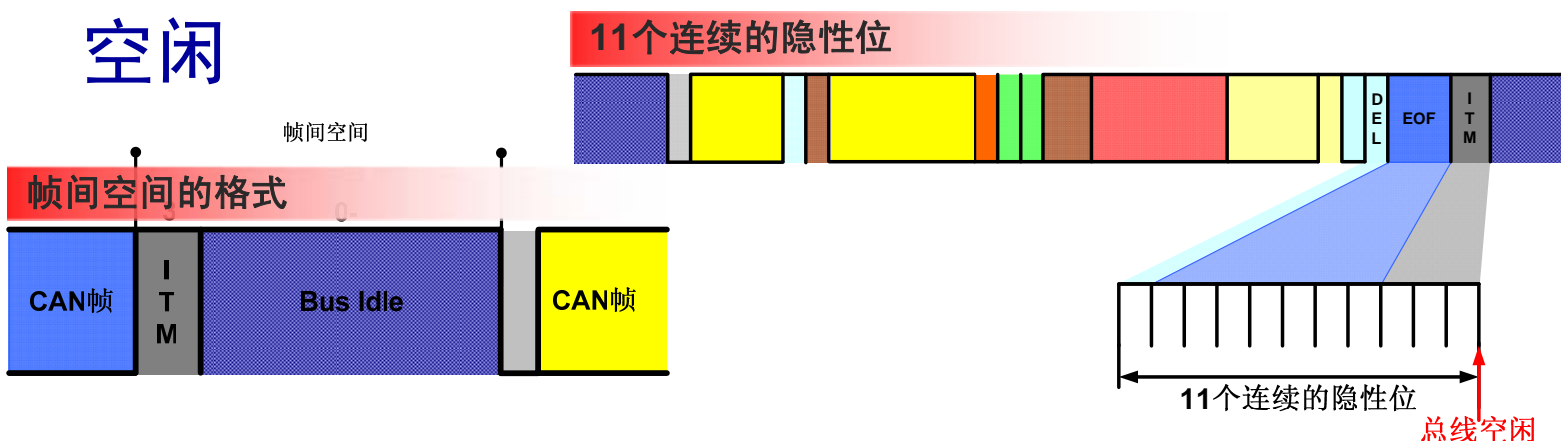
■ ACK场

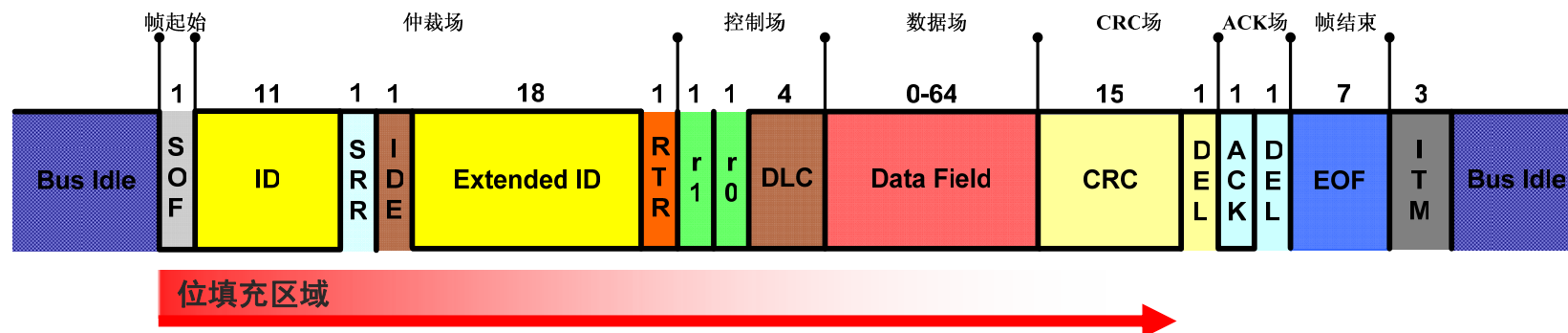
- ❖ 发送节点在ACK发送隐性位
- ❖ 正确接收到报文的节点→ACK发送显性
- ❖ 未正确接收到报文的节点→ACK发送隐性
- ❖ 发送节点检测应答位是否被显性覆盖
 - 没有→ACK错误



■ 帧结束

- ❖ 7个连续的隐性位，表示数据帧结束
- ❖ 节点在检测到11个连续的隐性位后认为总线空闲



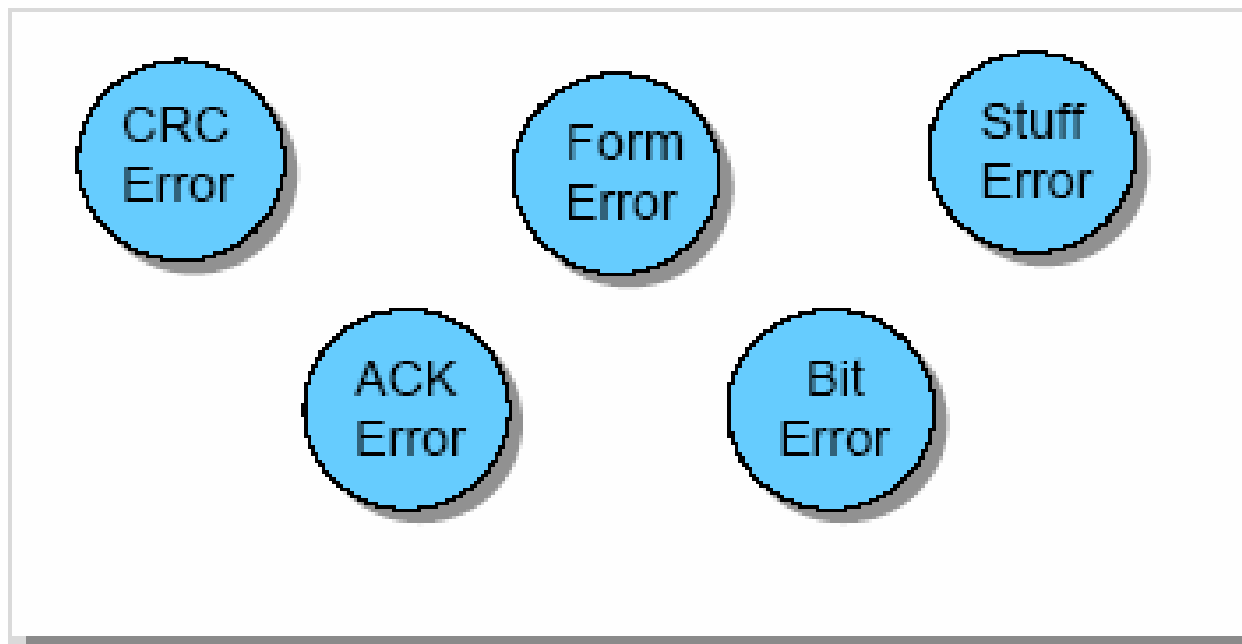


- SOF之前的总线空闲区域，不需要同步，无需进行位填充
- CRC之后的位域都是固定格式，不允许位填充操作

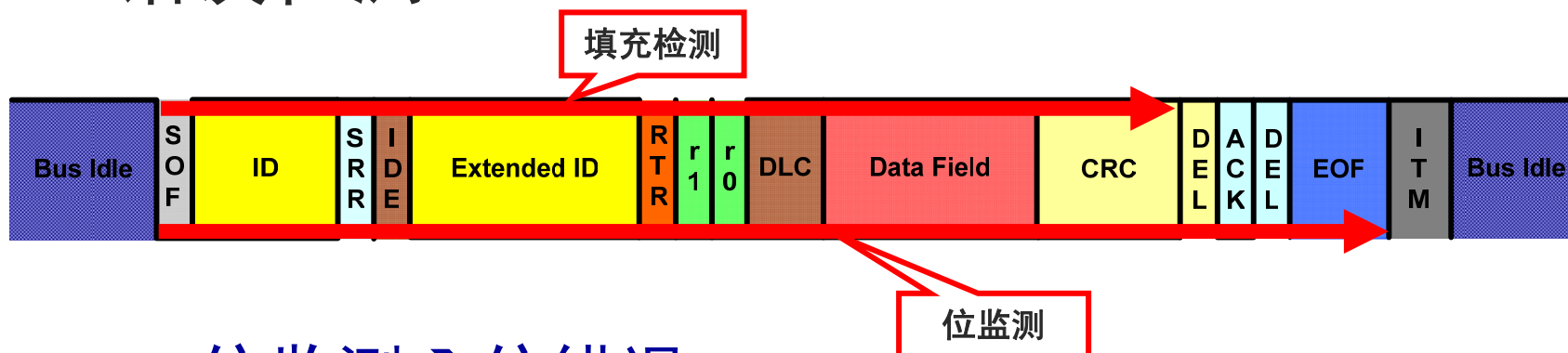
- 概述
- 汽车总线与CAN标准
- CAN的总线访问
- 数据帧
- 错误检测与错误帧
- CAN的帧格式
- 位定时与同步
- 物理连接



■ 可检测的错误



■ 错误检测



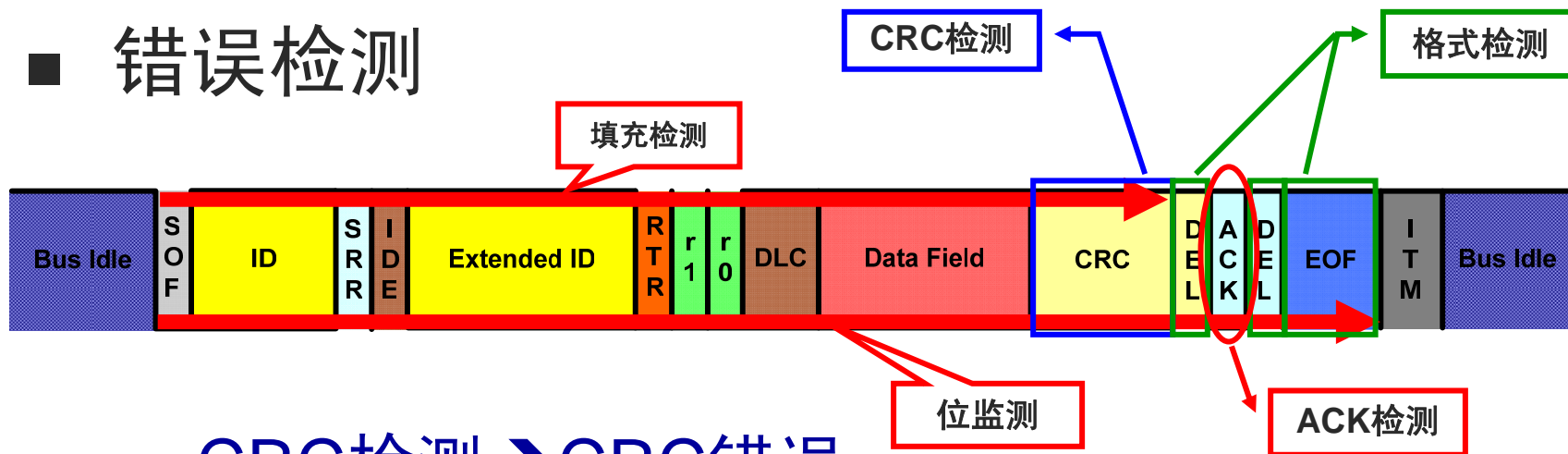
❖ 位监测 → 位错误

- ❑ 节点检测到的位与自身送出的位数值不同
- ❑ 仲裁或ACK位期间送出“隐性”位，而检测到“显性”位不导致位错误

❖ 填充检测 → 填充错误

- ❑ 在使用位填充编码的帧场（帧起始至CRC序列）中，不允许出现六个连续相同的电平位

■ 错误检测



❖ CRC检测→CRC错误

- ❑ 节点计算的CRC序列与接收到的CRC序列不同

❖ 格式检测→格式错误

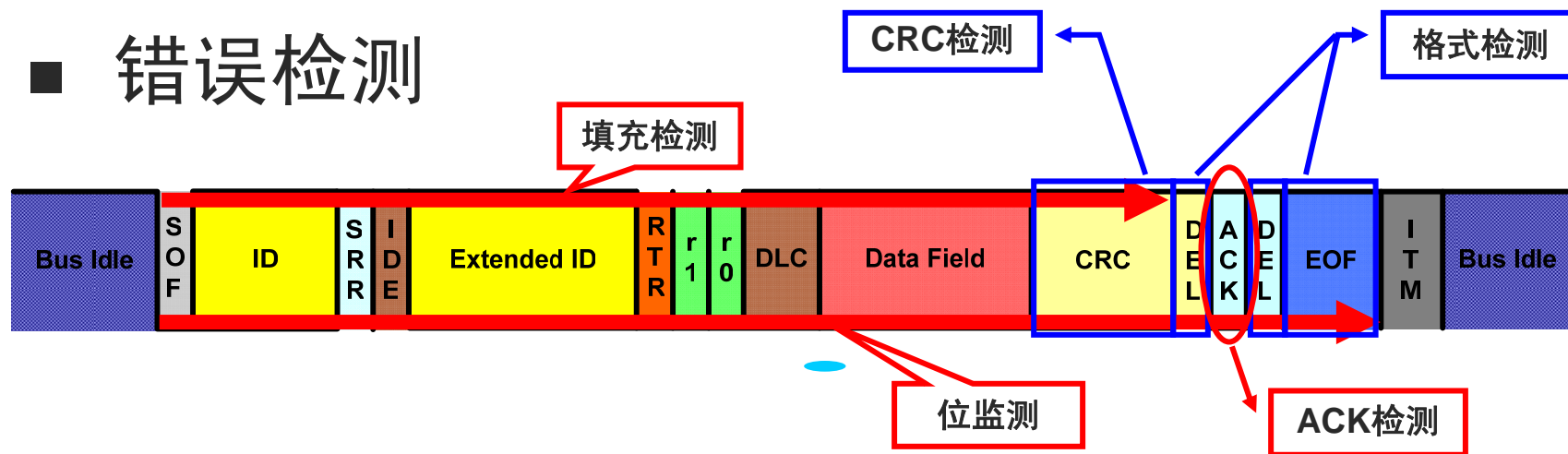
- ❑ 固定格式位场（如CRC界定符、ACK界定符、帧结束等）含有一个或更多非法位

❖ ACK检测→ACK错误

- ❑ 发送节点在ACK位期间未检测到“显性”位



■ 错误检测



- ❖ 发送节点 → 位错误、格式错误、ACK错误
- ❖ 接收节点 → 填充错误、格式错误、CRC错误

2000小时/年, 500kbps,
25%总线负载
每1000年才漏检一个错误

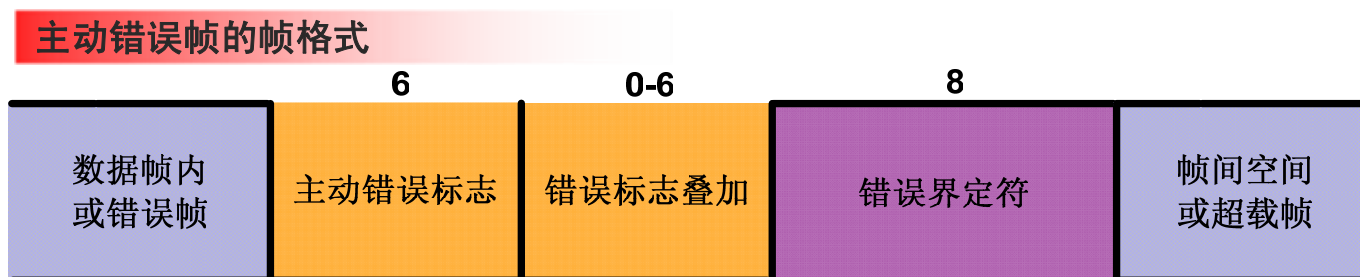


■ 错误帧

❖ 检测错误 → 发送错误帧 → 通知报文错误

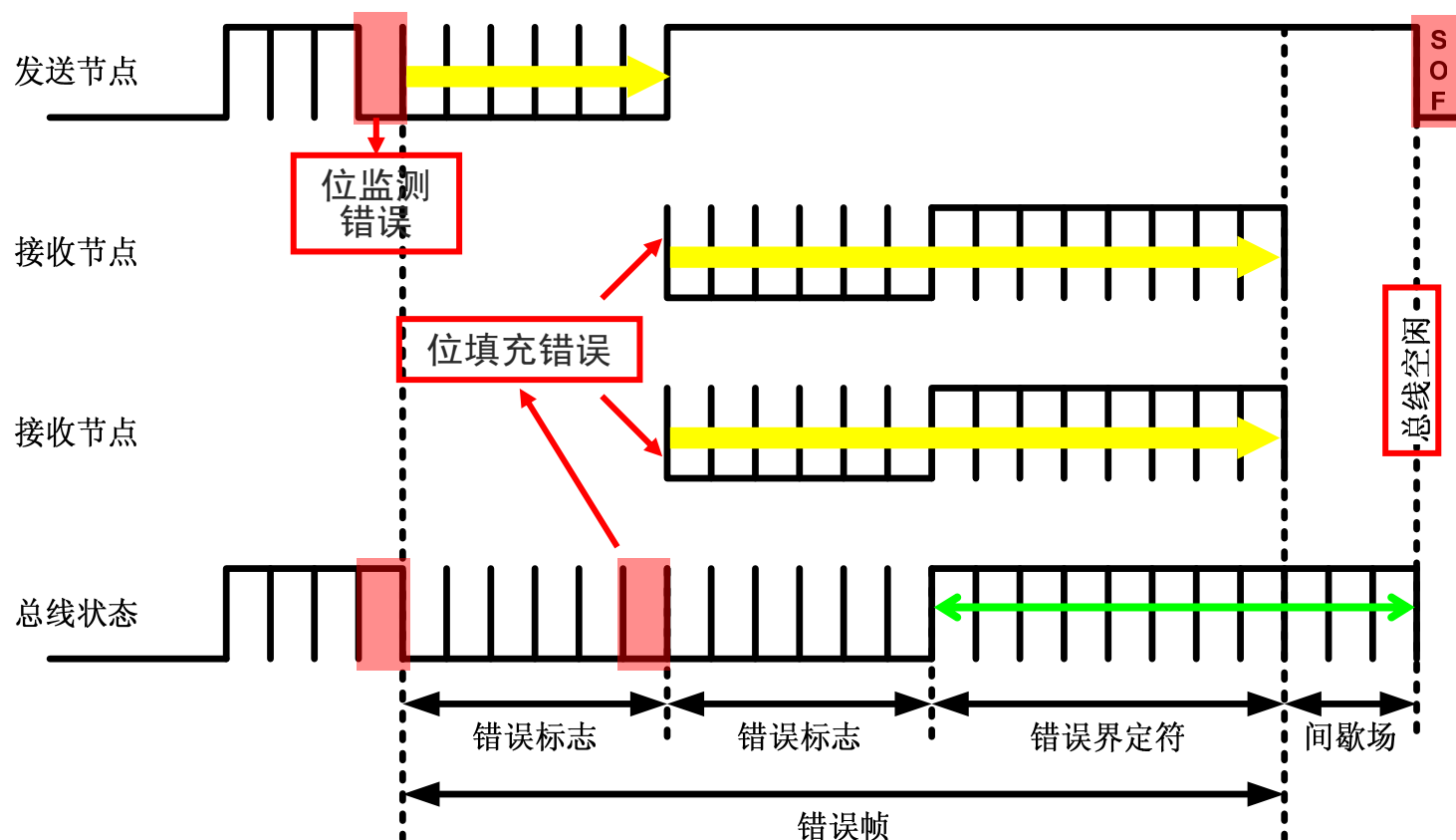
❖ 错误帧的发送

- 位错误、填充错误、格式错误或ACK错误产生后 → 当前发送的下一位发送错误帧
- CRC错误 → 紧随ACK界定符后的位发送错误帧
- 错误帧发送后 → 总线空闲时重发出错的数据帧



■ 错误帧

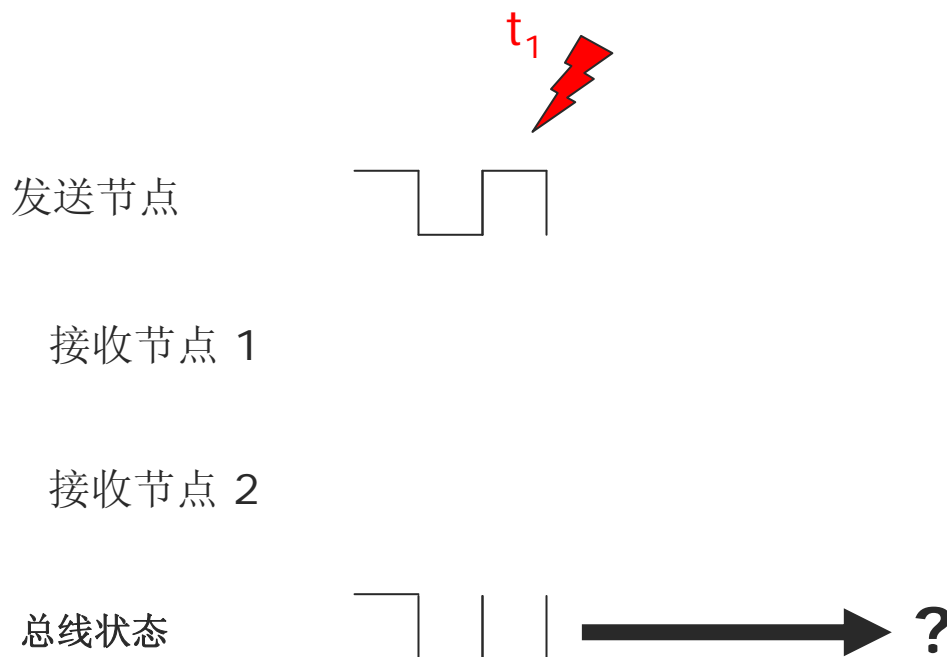
❖ 错误帧的发送（局部错误）





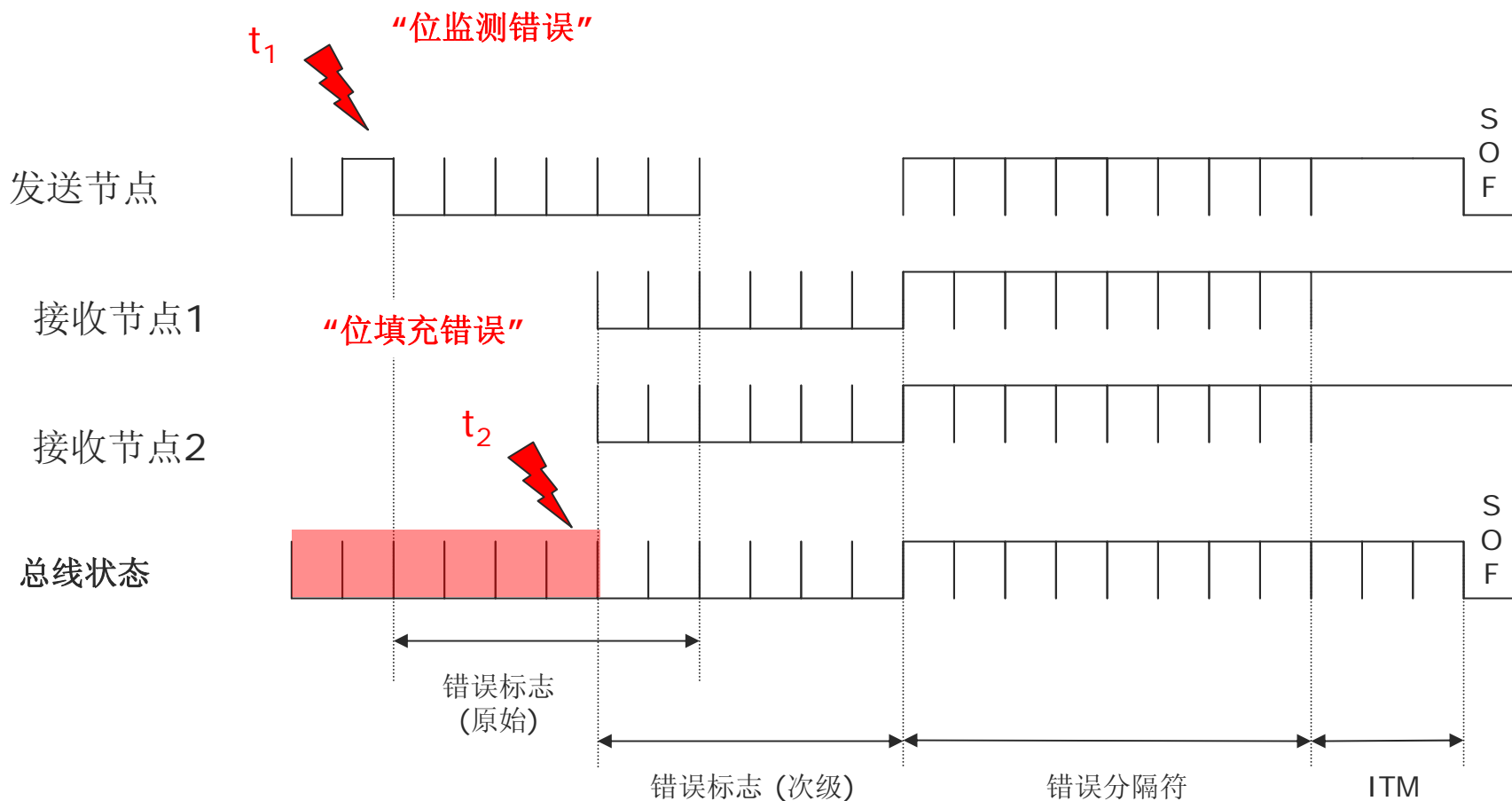
练习2：位错误

假设在时刻point t_1 发送节点检测到了一个位监测错误，
请完成两个接收节点的信号响应。



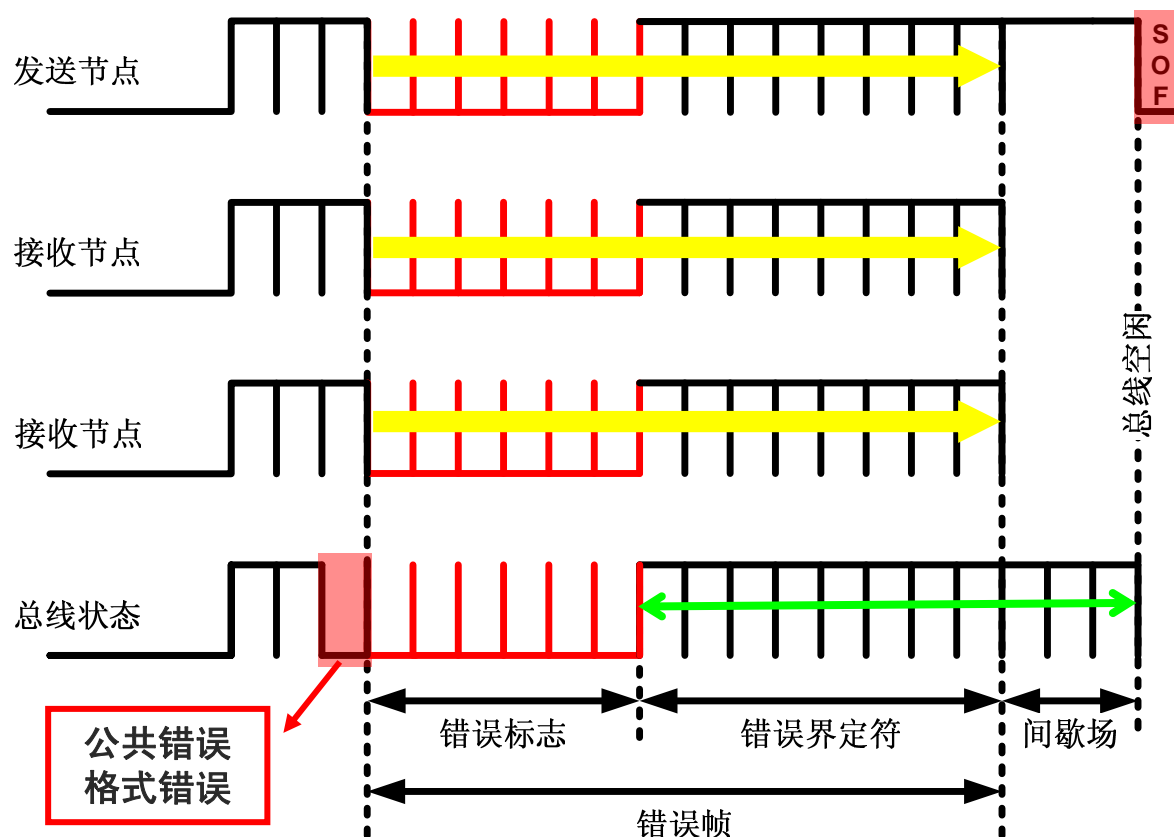


练习2：答案



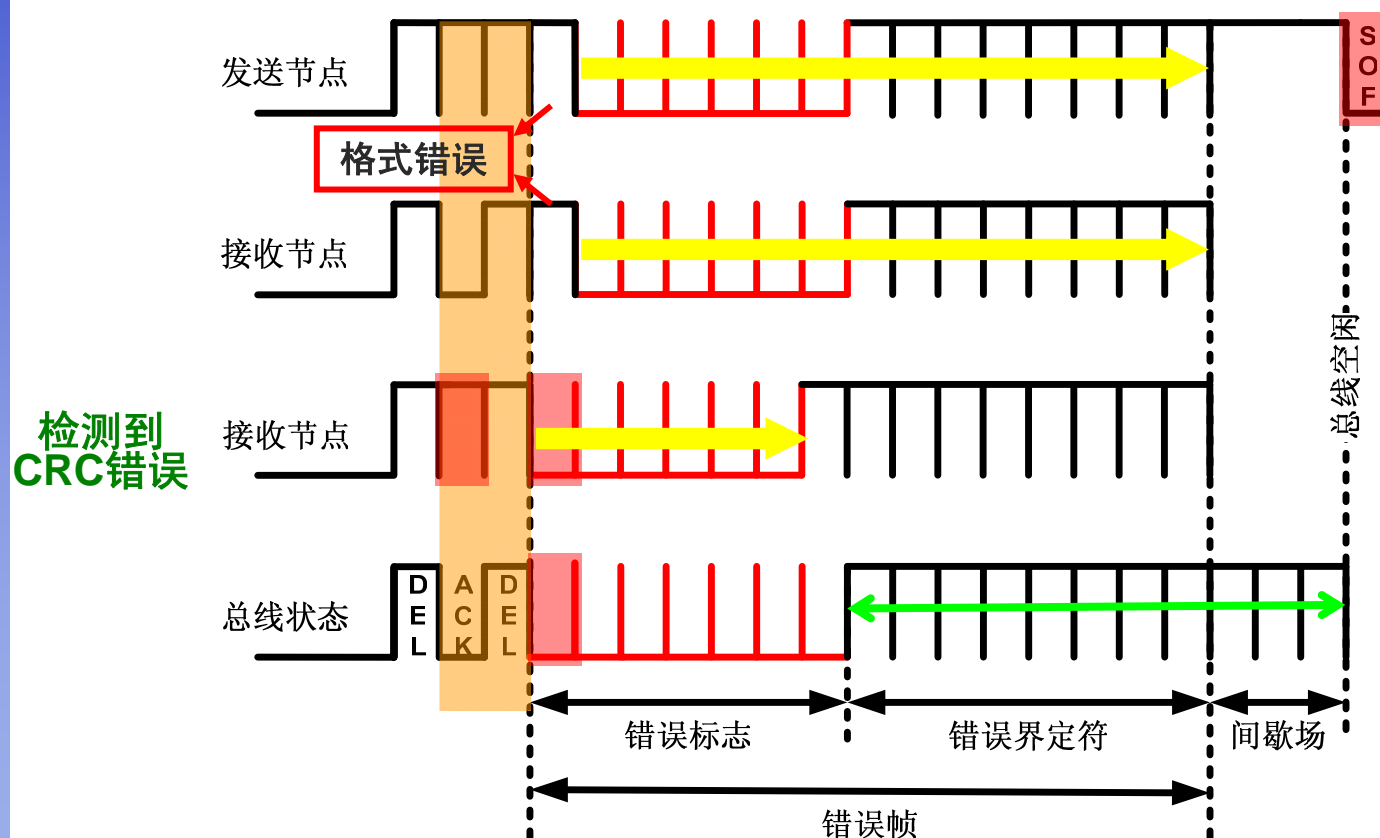
■ 错误帧

❖ 错误帧的发送（公共错误）



■ 错误帧

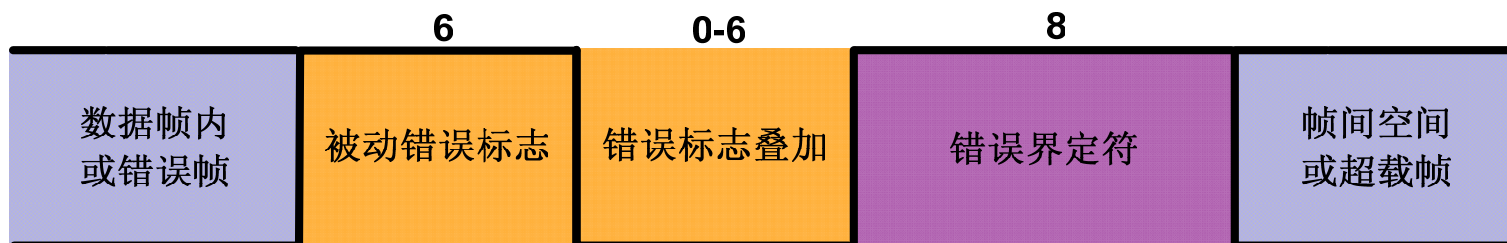
❖ 错误帧的发送（CRC错误）



■ 被动错误

- ❖ 被动错误状态的节点发送被动错误标志
- ❖ 被动错误标志由6个连续的隐性位组成，能够部分或全部被其它节点的显性位覆盖

被动错误帧的帧格式





■ 被动错误

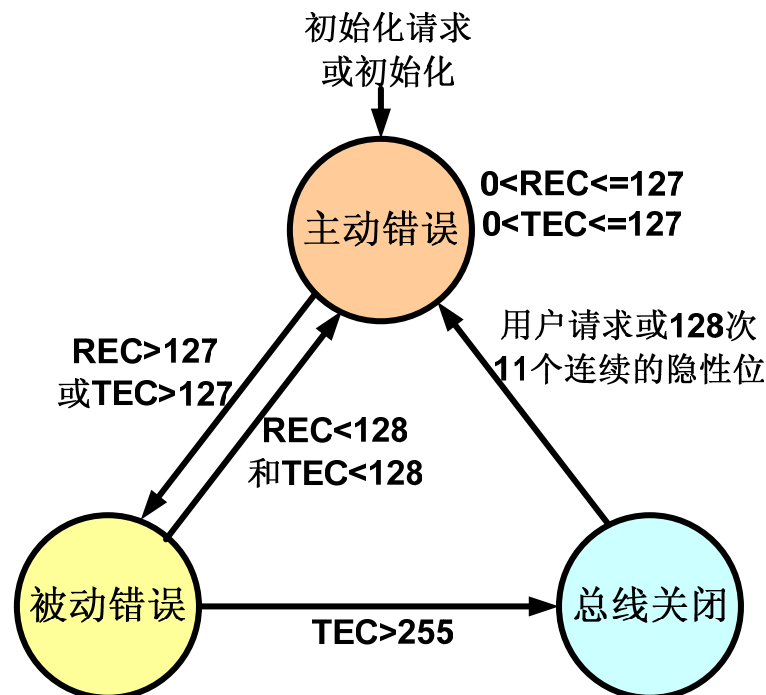
- ❖ 由发送节点发送的被动错误标志，会诱发接收节点发送错误标志，特例
 - 仲裁期间，ACK期间
- ❖ 由接收节点引起的被动错误标志不会诱发总线上的任何活动
- ❖ 被动错误节点检测到总线上6个连续相同的极性位后，认为错误标志已经送出

■ 节点的（错误）状态

由REC和TEC
的数值界定

数值随外界条件
增加或减少

- ❖ 主动错误状态 → 可收可发、使用主动错误标志
- ❖ 被动错误状态 → 可收可发、使用被动错误标志
- ❖ 总线关闭状态 → 不参与任何总线活动

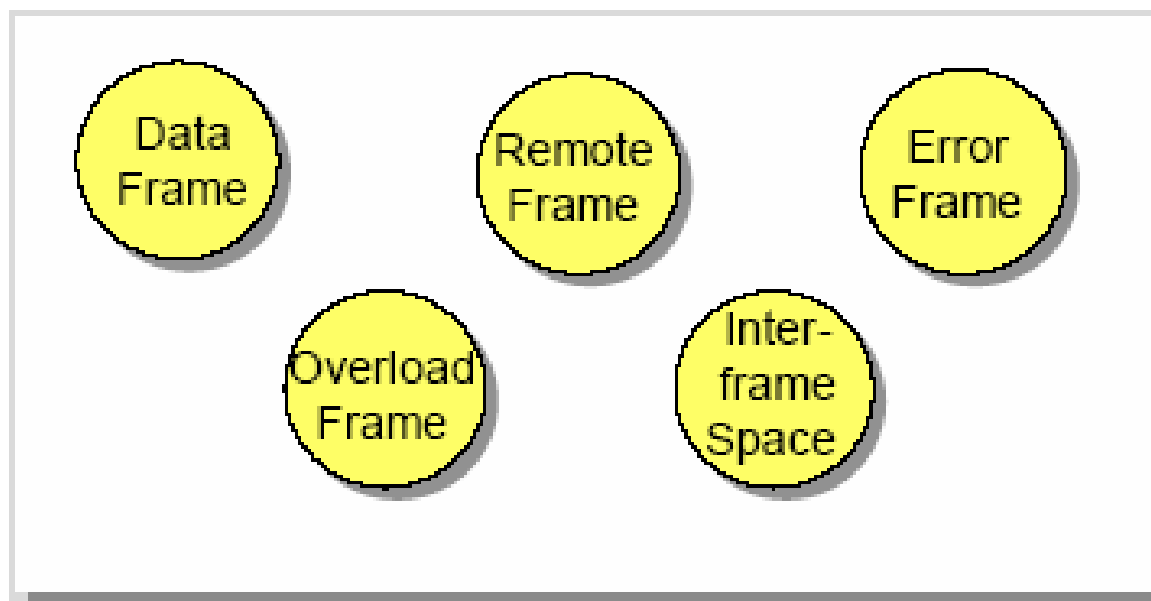


如果总线上只有一个节点，该节点发送数据帧后不会得到应答，此时该节点只是误状态不会进入总线关闭状态

- 概述
- 汽车总线与CAN标准
- CAN的通信机制
- 数据帧
- 错误检测与错误帧
- **CAN的帧格式**
- 位定时与同步
- 物理连接



现有的帧格式

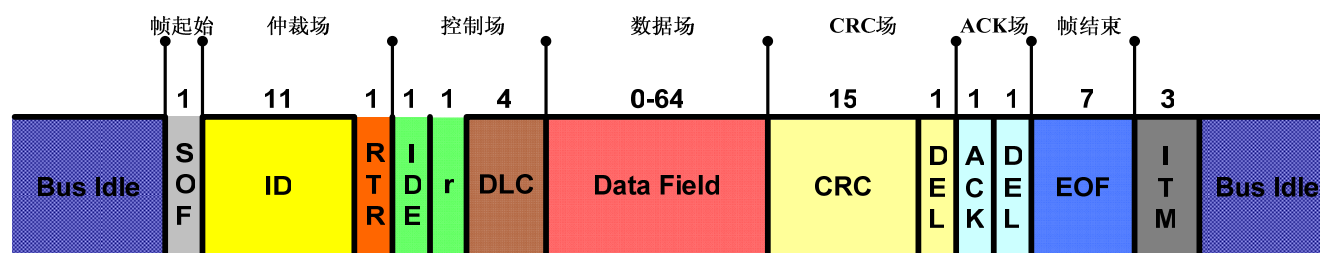


■ CAN的帧格式

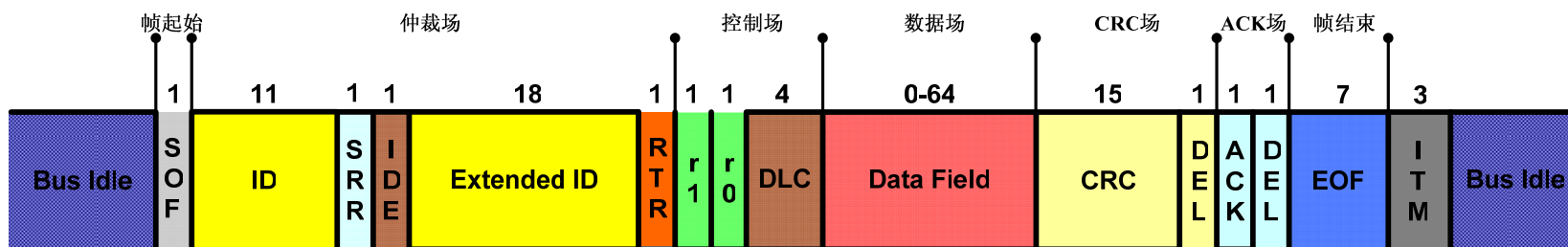
- ❖ **数据帧** → 携带从发送节点至接收节点的数据
- ❖ **远程帧** → 向其他节点请求发送具有同一标识符的数据帧
- ❖ **帧间空间** → 数据帧（或远程帧）通过帧间空间与前述的各帧分开
- ❖ **错误帧** → 节点检测到错误后发送错误帧
- ❖ **超载帧** → 在先行的和后续的数据帧（或远程帧）之间附加一段延时

■ 数据帧

❖ 标准帧

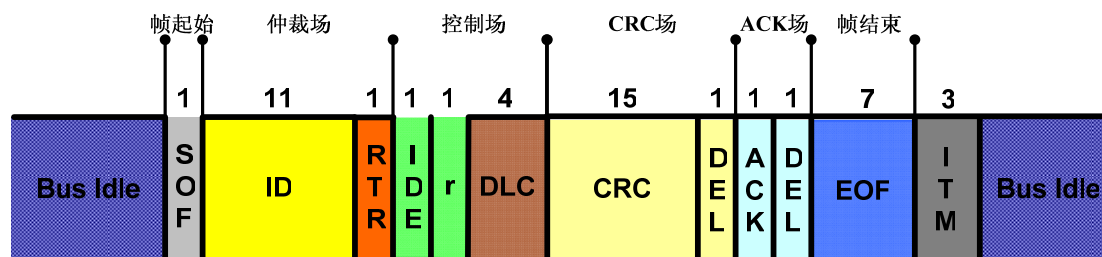


❖ 扩展帧

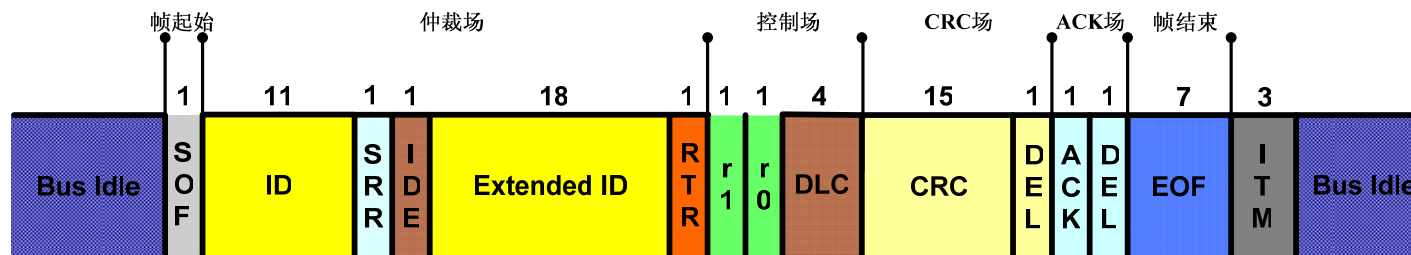


■ 远程帧

❖ 对应标准数据帧的远程帧



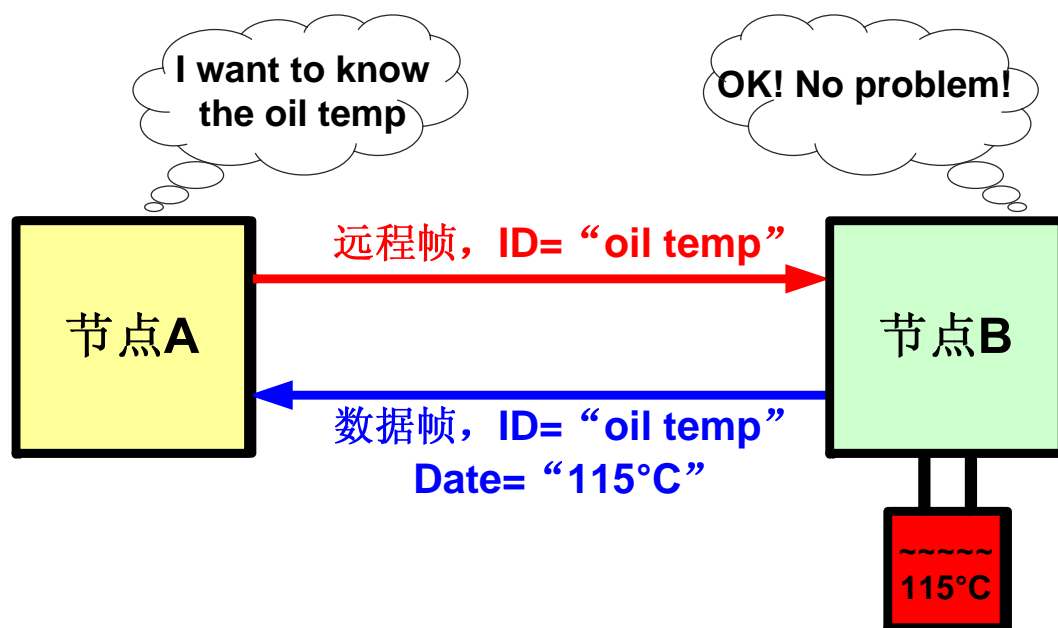
❖ 对应扩展数据帧的远程帧



■ 远程帧

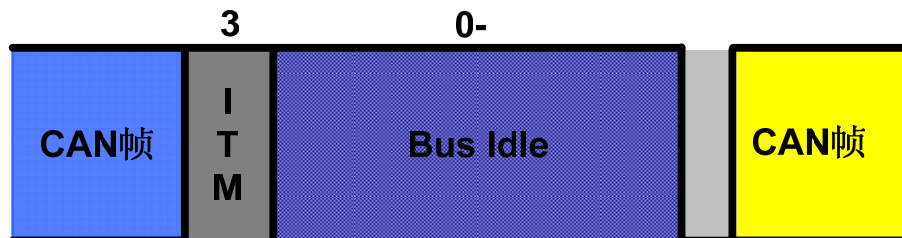
❖ 远程帧的使用

- 向其他节点请求发送具有同一标识符的数据帧

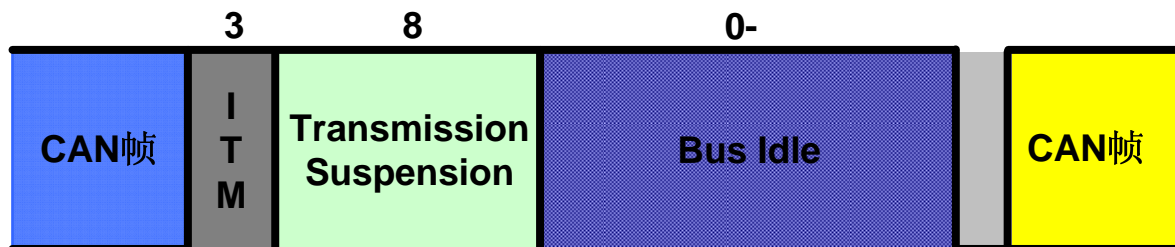


■ 帧间空间

❖ 主动错误节点使用的帧间空间格式

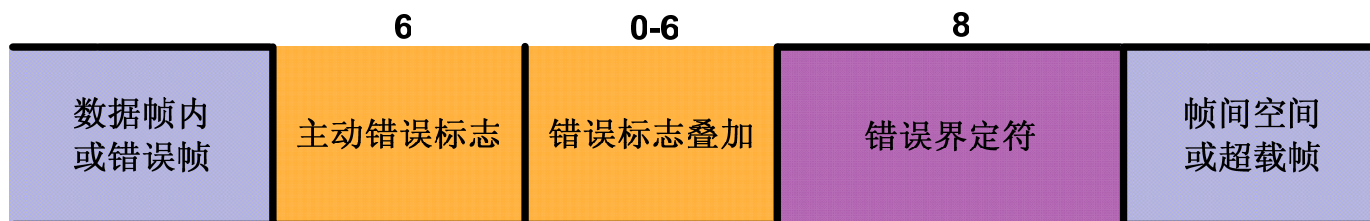


❖ 被动错误节点使用的帧间空间格式

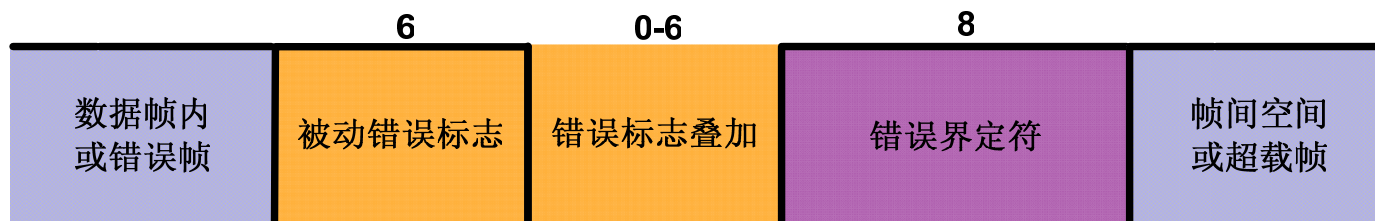


■ 错误帧

❖ 主动错误节点使用的错误帧格式

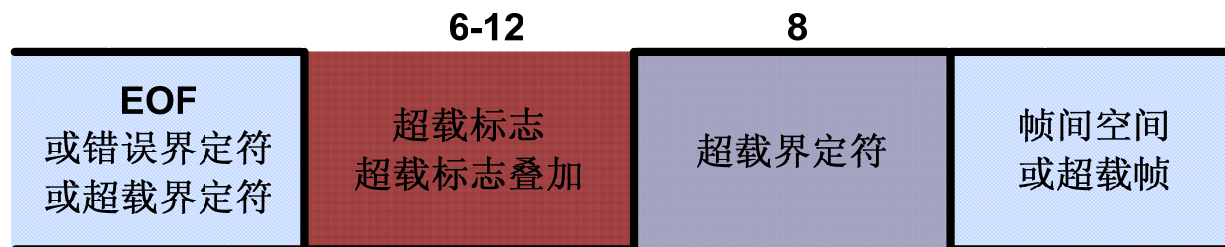


❖ 被动错误节点使用的错误帧格式



■ 超载帧

- ❖ 超载帧用以在先前的和后续的数据帧（或远程帧）之间提供一附加延时
- ❖ 大部分高层协议不使用超载帧

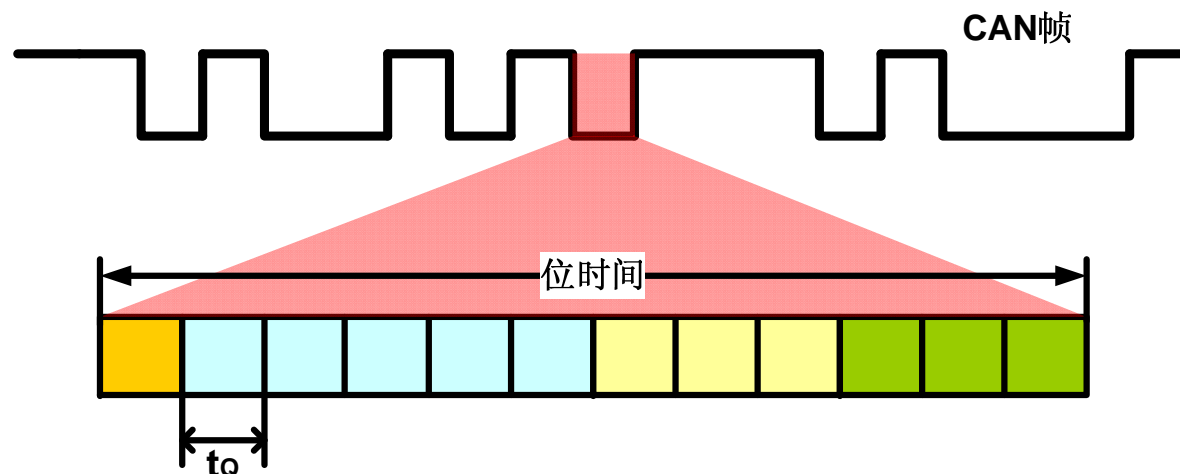


- 概述
- 汽车总线与CAN标准
- CAN的总线访问
- 数据帧
- 错误检测与错误帧
- CAN的帧格式
- 位定时与同步
- 物理连接

■ 位定时

❖ 波特率

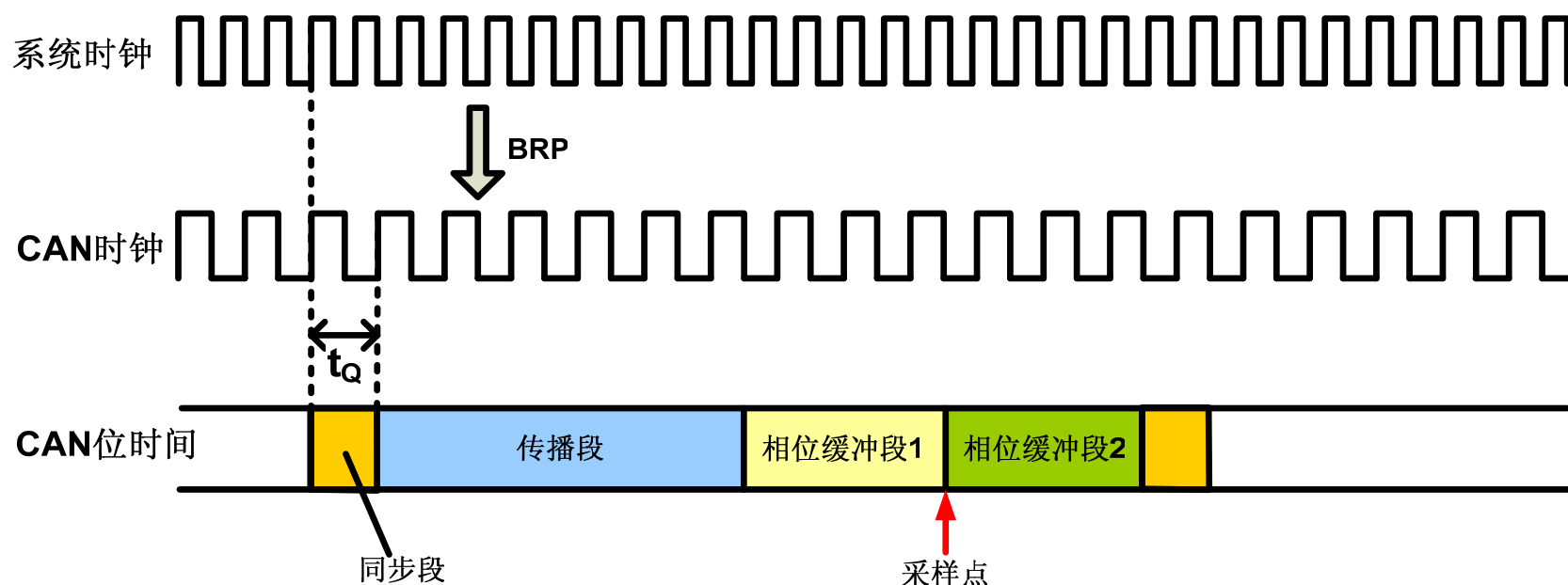
- 波特率可通过编程设置合适的时间量子长度和数量确定
- 波特率=1/位时间



■ 位定时

❖ 时间量子(Time Quantum)

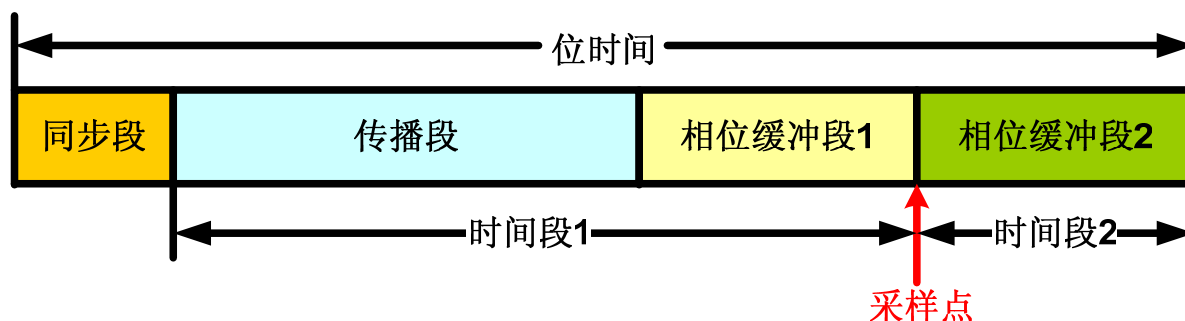
- 时间量子来源于对系统时钟可编程的分频



- **BRP=BaudRatePrescaler**

■ 位定时

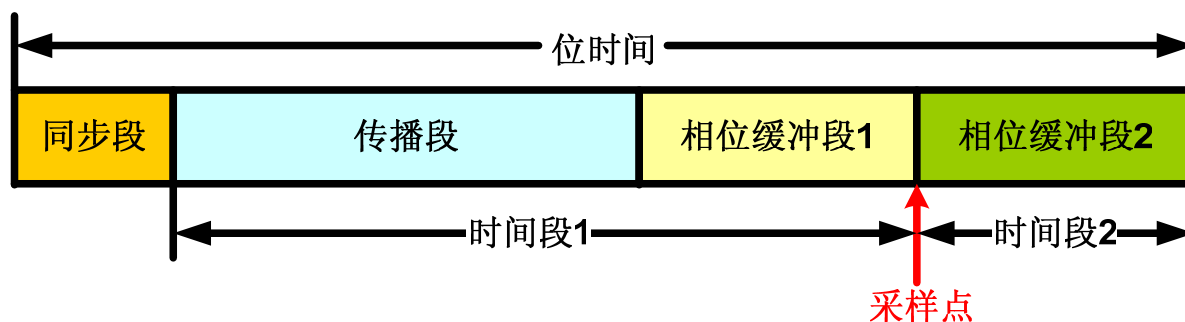
❖ 位时间的组成



- ❑ 一个位时间包含**4个时间段**，8-25个时间量子 (Time Quantum)
- ❑ 为方便编程，许多CAN模块将传播段和相位缓冲段1合并为一个时间段，即只有**3个时间段**

■ 位定时

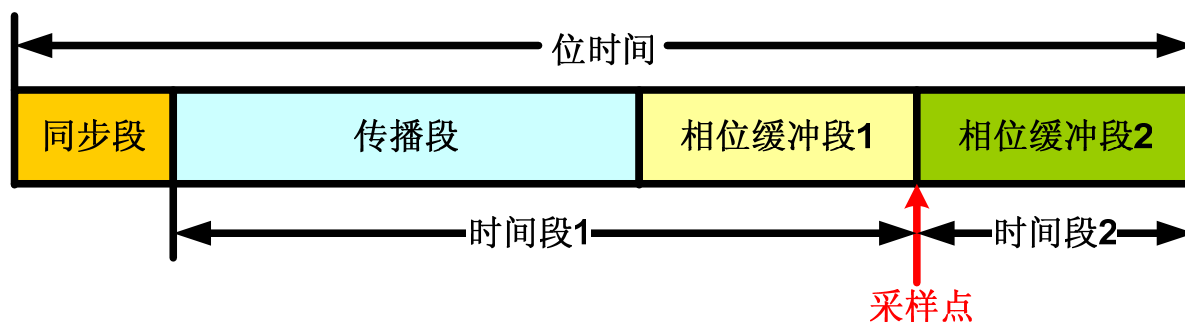
❖ 同步段—Synchronization Segment



- ❑ 一个位的输出从同步段开始
- ❑ 同步段用于同步总线上的各个节点，跳变沿产生在此段内
- ❑ 固定长度，1个时间量子

■ 位定时

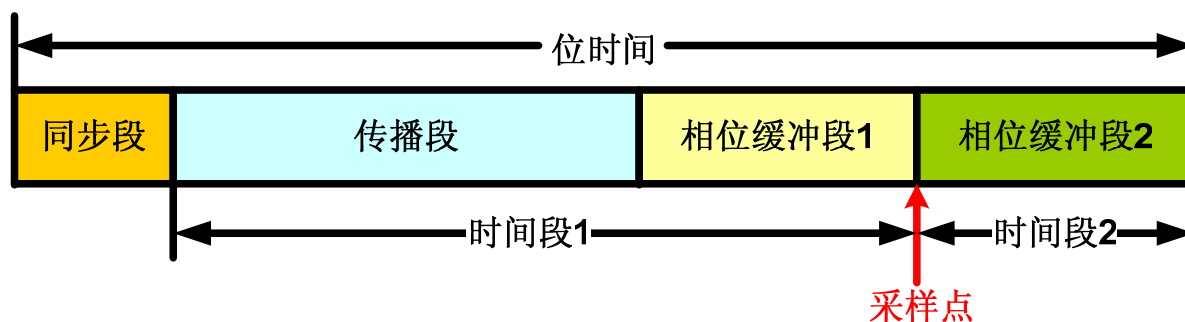
❖ 传播段—Propagation Segment



- ❑ 用于补偿信号通过网络和节点传播的物理延迟
- ❑ 传播段长度应能保证2倍的信号在总线的延迟
- ❑ 长度可编程（1...8个时间量子或更长）

■ 位定时

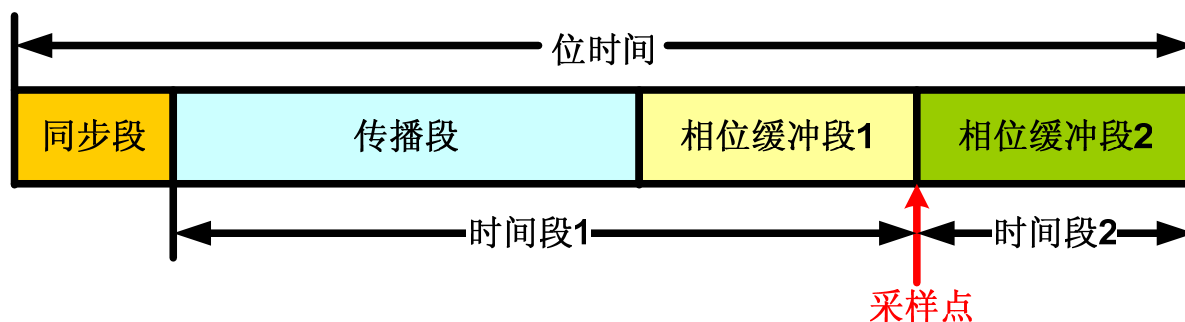
❖ 相位缓冲段1—Phase Buffer Segment1



- 用于补偿节点间的晶振误差
- 允许通过重同步对该段加长
- 在这个时间段的末端进行总线状态的采样
- 长度可编程（1...8个时间量子或更长）

■ 位定时

❖ 相位缓冲段2—Phase Buffer Segment2



- 用于补偿节点间的晶振误差
- 允许通过重同步对该段缩短
- 长度可编程，1...8个时间量子或更长

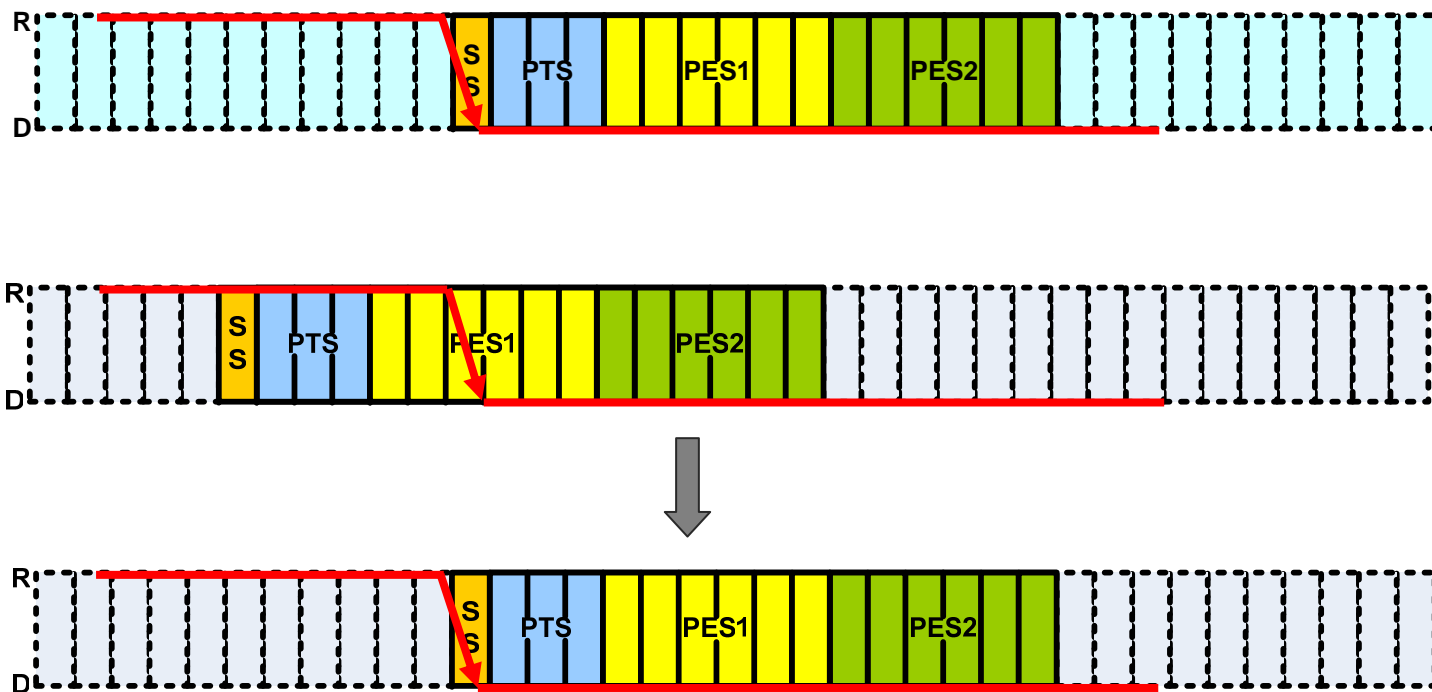


■ 同步

- ❖ CAN的同步包括**硬同步**和**重同步**两种同步方式
- ❖ 同步规则：
 - 一个位时间内只允许一种同步方式
 - 任何一个“隐性”到“显性”的跳变都可用于同步
 - **硬同步**发生在SOF→所有接收节点调整各自当前位的同步段，使其位于发送的SOF位内
 - 重同步发生在一个帧的其他位场内，当跳变沿落在了同步段之外
 - 在SOF到仲裁场有多个节点同时发送的情况下，发送节点对跳变沿不进行重同步

■ 同步

- ❖ 硬同步 发生在SOF位→所有接收节点调整各自当前位的同步段，调整宽度不限



■ 同步

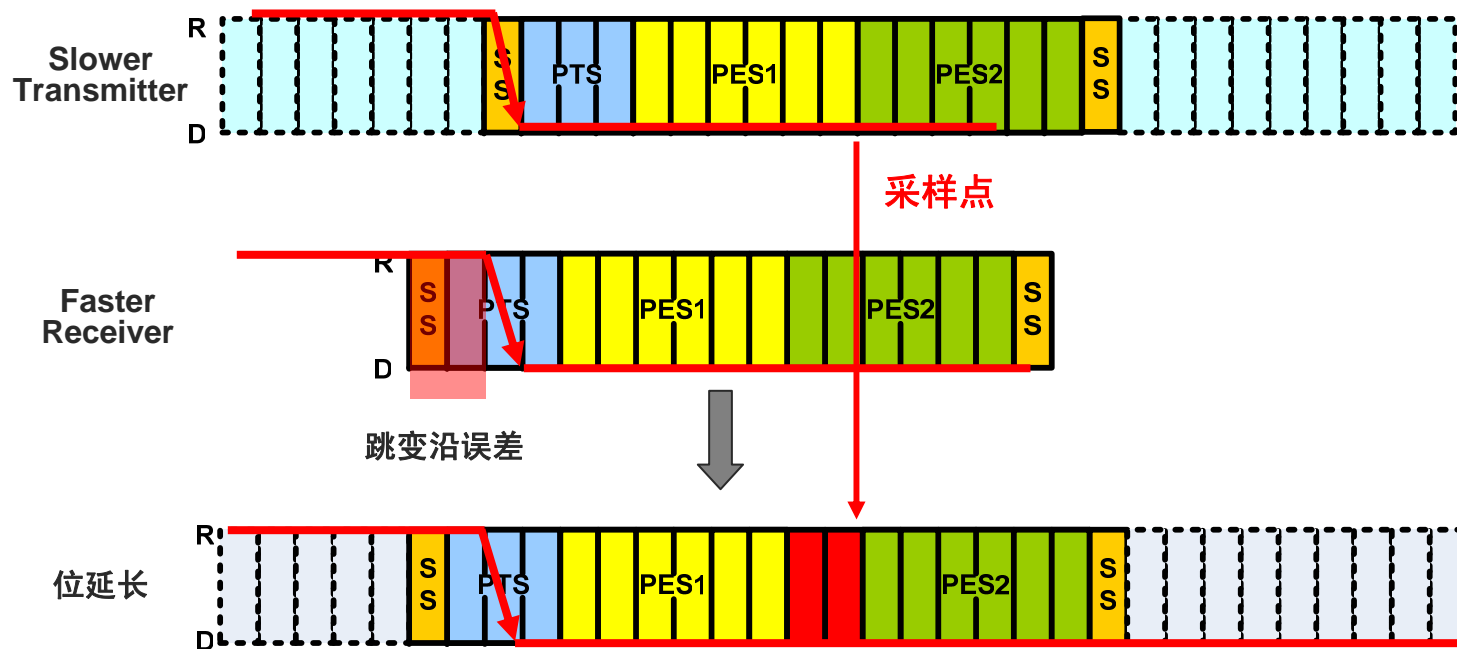
❖ 重同步

- 跳变沿与同步段的误差 $< SJW \rightarrow$ 重同步导致的相位缓冲段1延长或相位缓冲段2缩短，能够保证采样点位置的正确 \rightarrow 重同步和硬同步作用相同
- 跳变沿与同步段的误差 $\geq SJW$
 - 如果相位误差为正，沿位于采样点之前 \rightarrow 相位缓冲段1被增长 $= SJW$
 - 如果相位误差为负，沿位于前一个位的采样点之后 \rightarrow 相位缓冲段2被缩短 $= SJW$

■ 同步

❖ 重同步

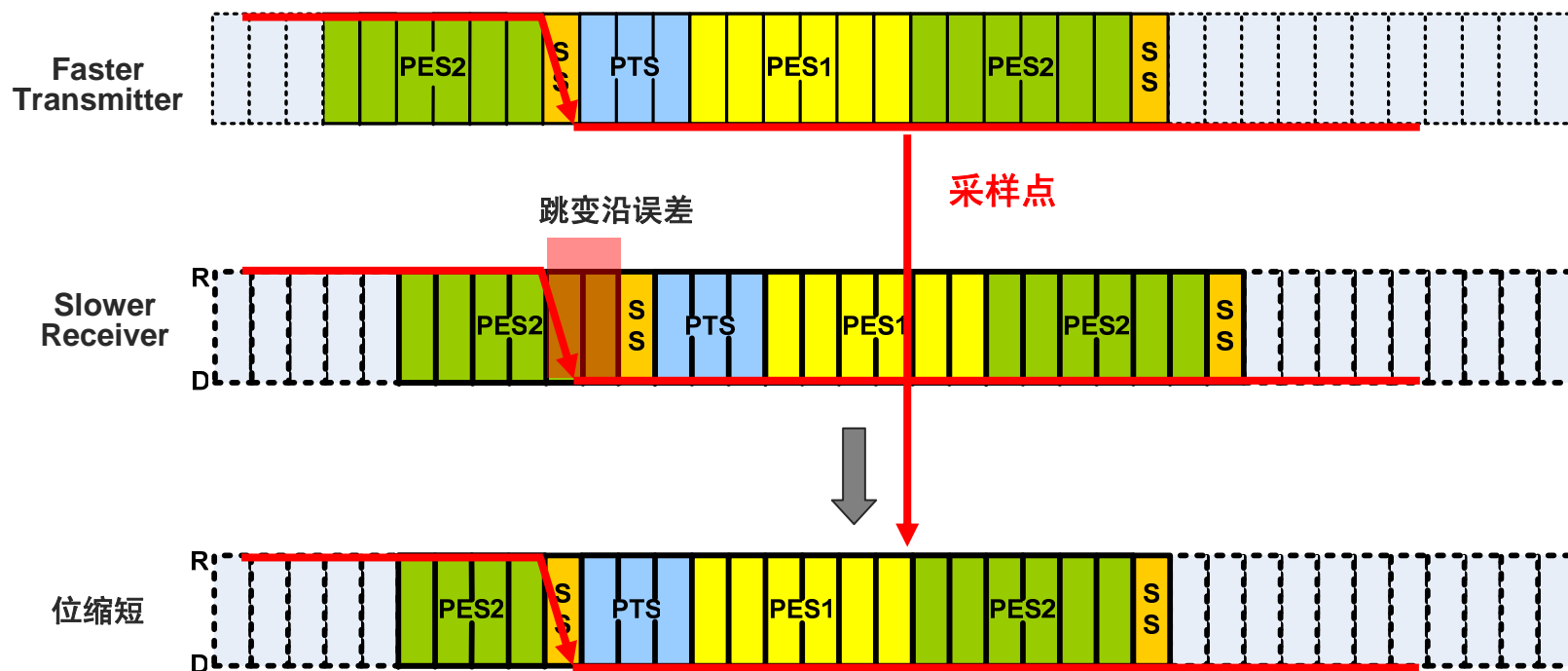
□ 相位缓冲段1延长 (SJW=4)



■ 同步

❖ 重同步

□ 相位缓冲段2缩短 (SJW=4)



■ 位定时

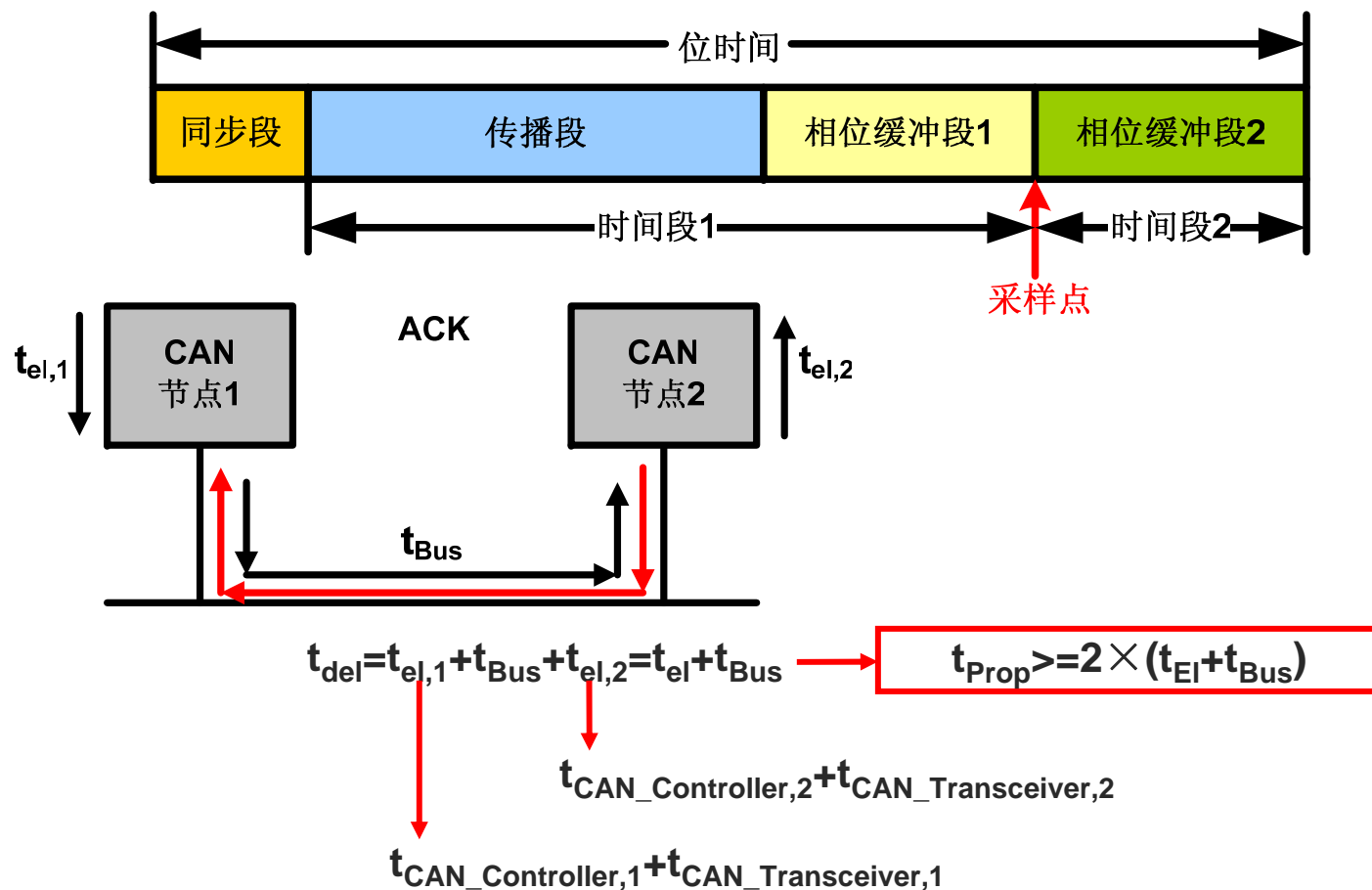
❖ 同步跳转宽度—Synchronization Jump Width

- **SJW**为**PES1**和**PES2**调整的最大长度
- **SJW**必须小于**PES1**和**PES2**的最小值



■ 传播段

❖ 延迟时间的确定



■ 位定时

❖ 位定时参数确定

- $T(\text{Bit}) = 1/\text{Baudrate}$
- $T(\text{tq}) = T(\text{Bit}) / \text{NBT}$
- $T(\text{Prop_Seg}) = 2 * (\text{DelayTransceiver} + \text{DelayBus})$
- $\text{Prop_Seg} = T(\text{Prop_Seg}) / T(\text{tq})$
- If $(\text{NBT} - 1 - \text{Prop_Seg}) / 2$ 为偶数
 $\text{Phase_Seg1} = \text{Phase_Seg2} = (\text{NBT} - 1 - \text{Prop_Seg}) / 2$
 else
 $\text{Phase_Seg1} = (\text{NBT} - 1 - \text{Prop_Seg}) / 2,$
 $\text{Phase_Seg2} = (\text{Phase_Seg1}) + 1$



■ 位定时

❖ 位定时参数确定

- $SJW = \min (\text{Phase_Seg1} , 4)$

- 验证晶振频率误差

$$Df \leq SJW / (2 \cdot 10 \cdot NBT)$$

$$Df \leq (\min(\text{Phase_Seg1}, \text{Phase_Seg2})) / (2 \cdot (19 \cdot NBT - \text{Phase_Seg2}))$$

■ 位定时

- ❖ 位定时确定的用例 给定，MCU晶振8MHz，位速率1Mbps，总线长度20m，单位总线延迟5ns/m，物理接口的发送接收延迟150ns@85C (*From Freescale AN1798*)
 - 1) 总线的物理延迟= $20 \times 5 = 100\text{ns}$
 $t_{\text{Prop}} = 2 \times (100 + 150) = 500\text{ns}$
 - 2) 选择BRP = 1, $t_Q = 125\text{ns}$, NBT = 8
 - 3) PROP_SEG = $500 / 125 = 4$
 - 4) NBT - PROP_SEG - 1 = 3, PHASE_SEG1 = 1, PHASE_SEG2 = 2

■ 位定时

❖ 位定时确定的用例

给定，MCU晶振8MHz，位速率1Mbps，总线长度20m，单位总线延迟5ns/m，物理接口的发送接收延迟150ns@85C

□ 5) $RJW = \min \{ 4, PHASE_SEG1 \} = 1$

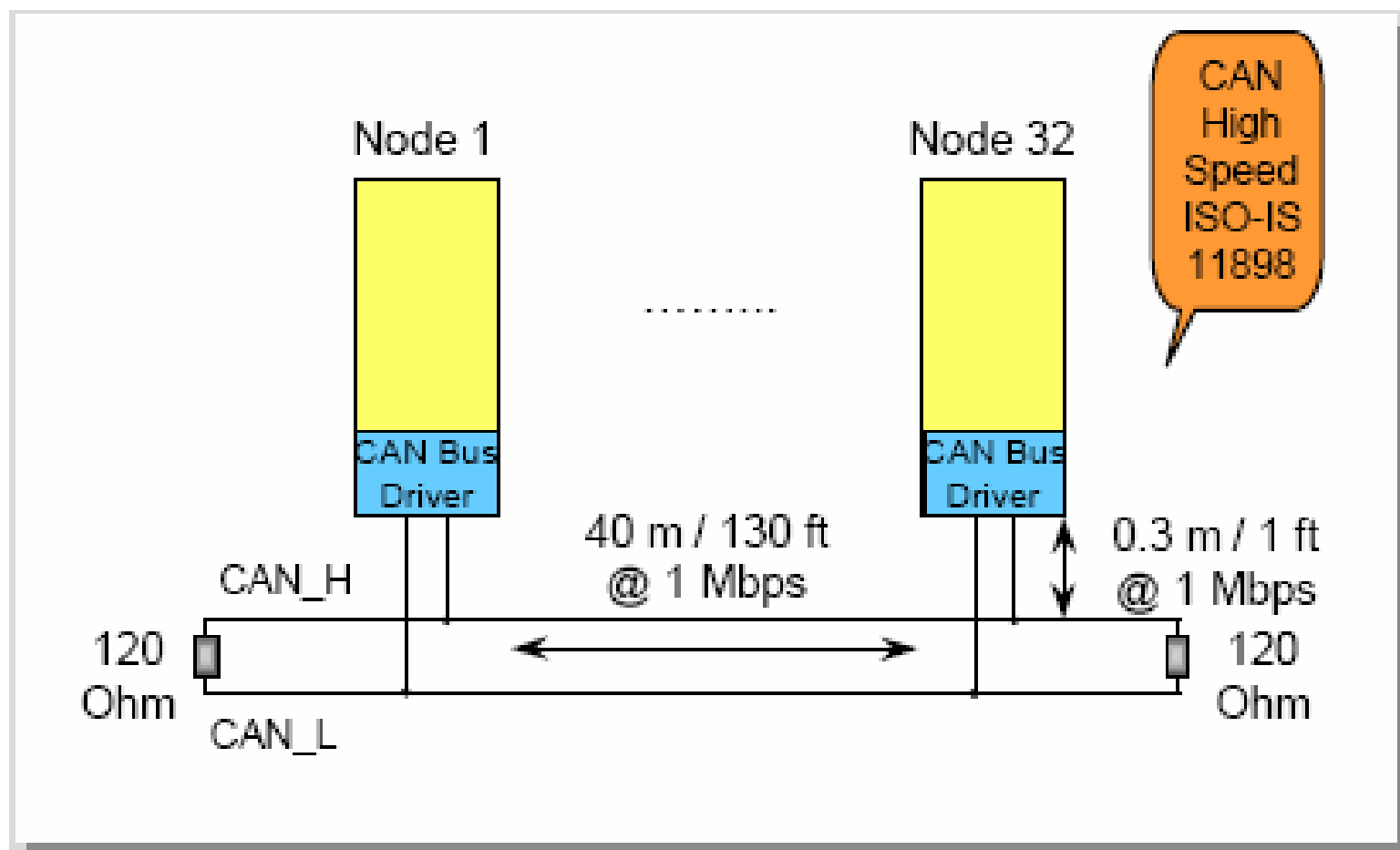
□ 6) $df \leq 1/2(13 \times 8 - 2) = 0.00490;$

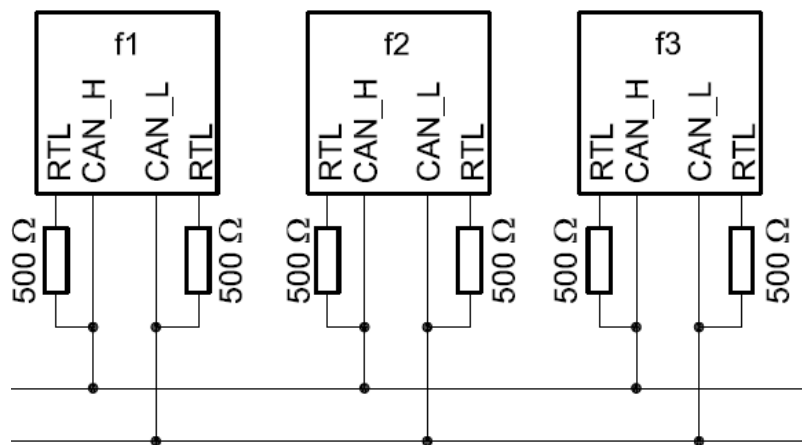
$df \leq 1/20 \times 8 = 0.00625$

- 概述
- 汽车总线与CAN标准
- CAN的总线访问
- 数据帧
- 错误检测与错误帧
- CAN的帧格式
- 位定时与同步
- 物理连接

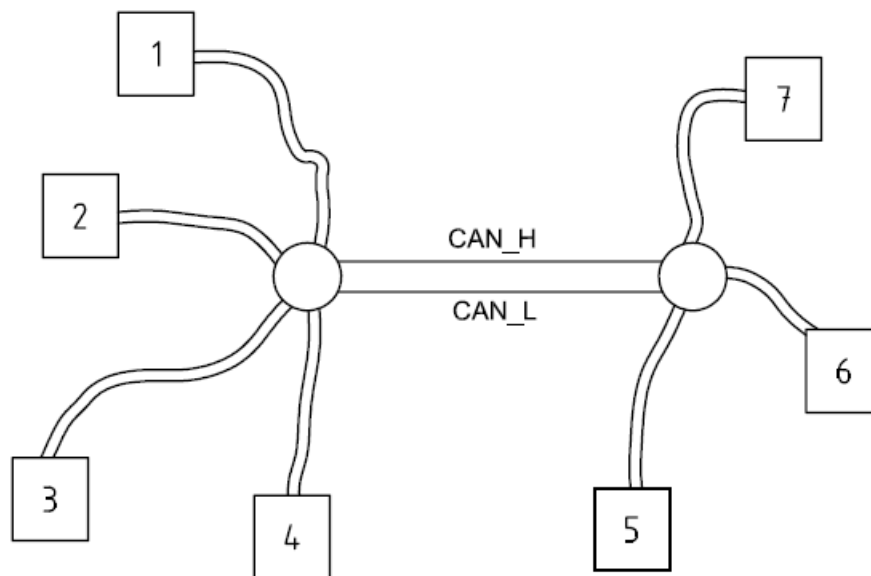


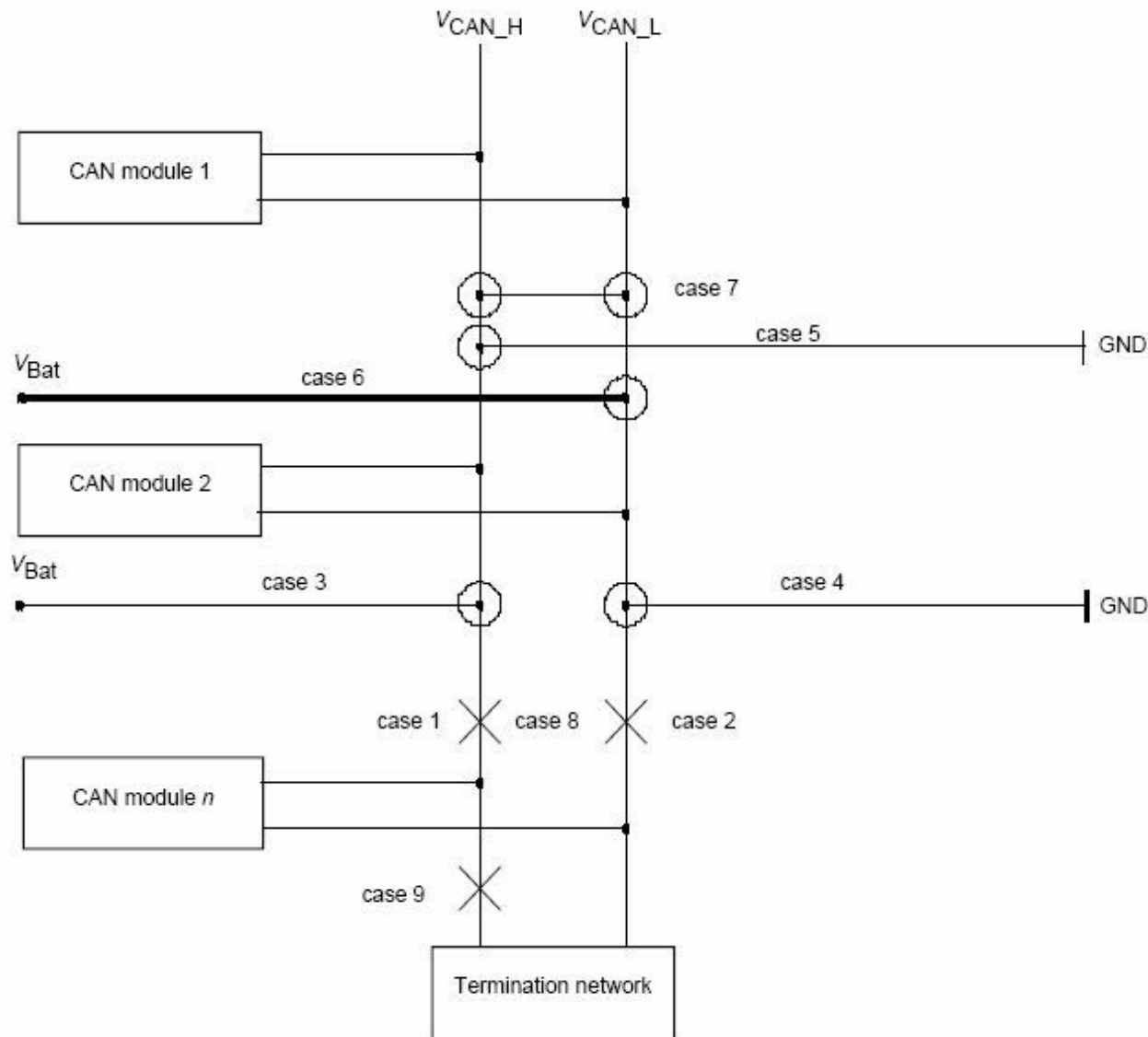
ISO11898-2规定的物理层





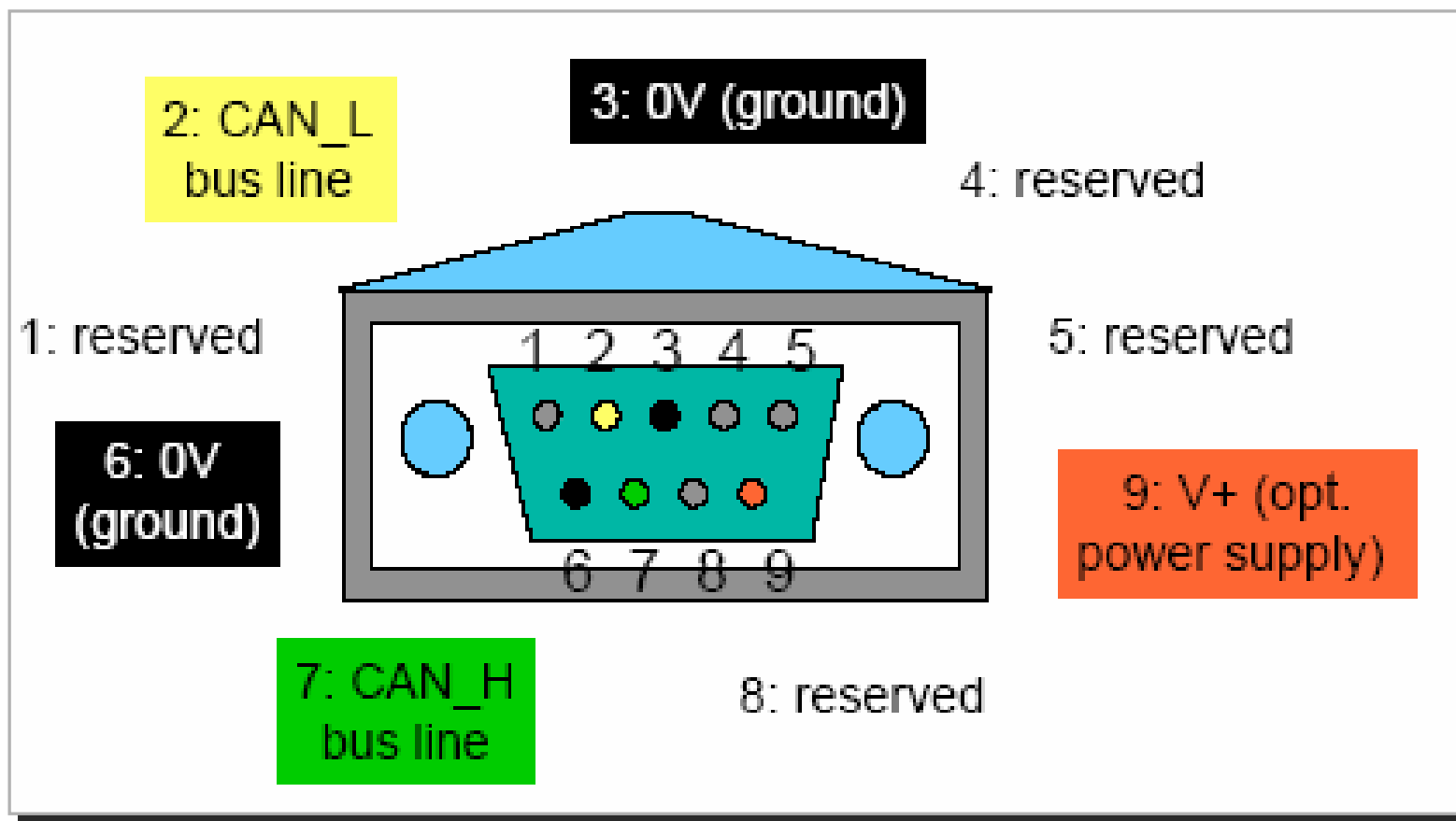
- $4.7K\Omega > R > 500\Omega$
- $R_{all} > 100\Omega$
- 可采用星型连接







CAN总线连接器



- 概述
- 汽车总线与CAN标准
- CAN的总线访问
- 数据帧
- 错误检测与错误帧
- CAN的帧格式
- 位定时与同步
- 物理连接

谢 谢