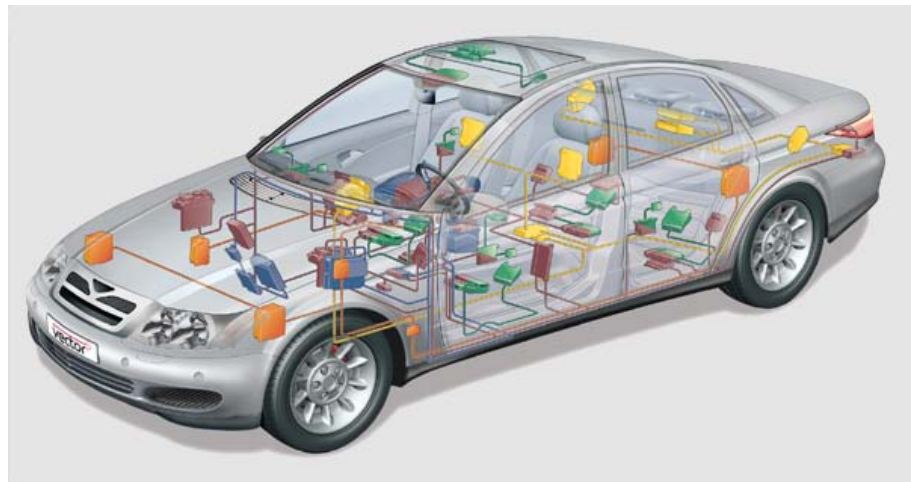


CAN总线快速入门



- CAN总线概述
- 数据链路层
- 物理层

- CAN总线概述
 - CAN总线起源
 - CAN发展历史
 - CAN总线相关标准
 - CAN总线概述
- 数据链路层
- 物理层

三个问题

□ 为什么需要总线？

□ 人类需要交流，ECU也需要

□ 人类的交流手段：书信、电话→网络

□ ECU的交流手段：线束→总线

□ 什么是CAN总线？

□ CAN (Controller Area Network)是二十世纪八十年代初德国Bosch公司为解决现代汽车中众多电控单元（ECU）之间的数据交换而开发的一种串行通信协议。

□ 为什么是CAN总线？

□ 技术完美 + 价格低廉 = 优胜劣汰

什么是汽车？

▣ 轮子上的沙发还是轮子上的计算机？



汽车——从机械液压到机械电子

□ 动力性

- 从空间的扩大到推背感

□ 经济性

- 从石油危机到日系汽车崛起

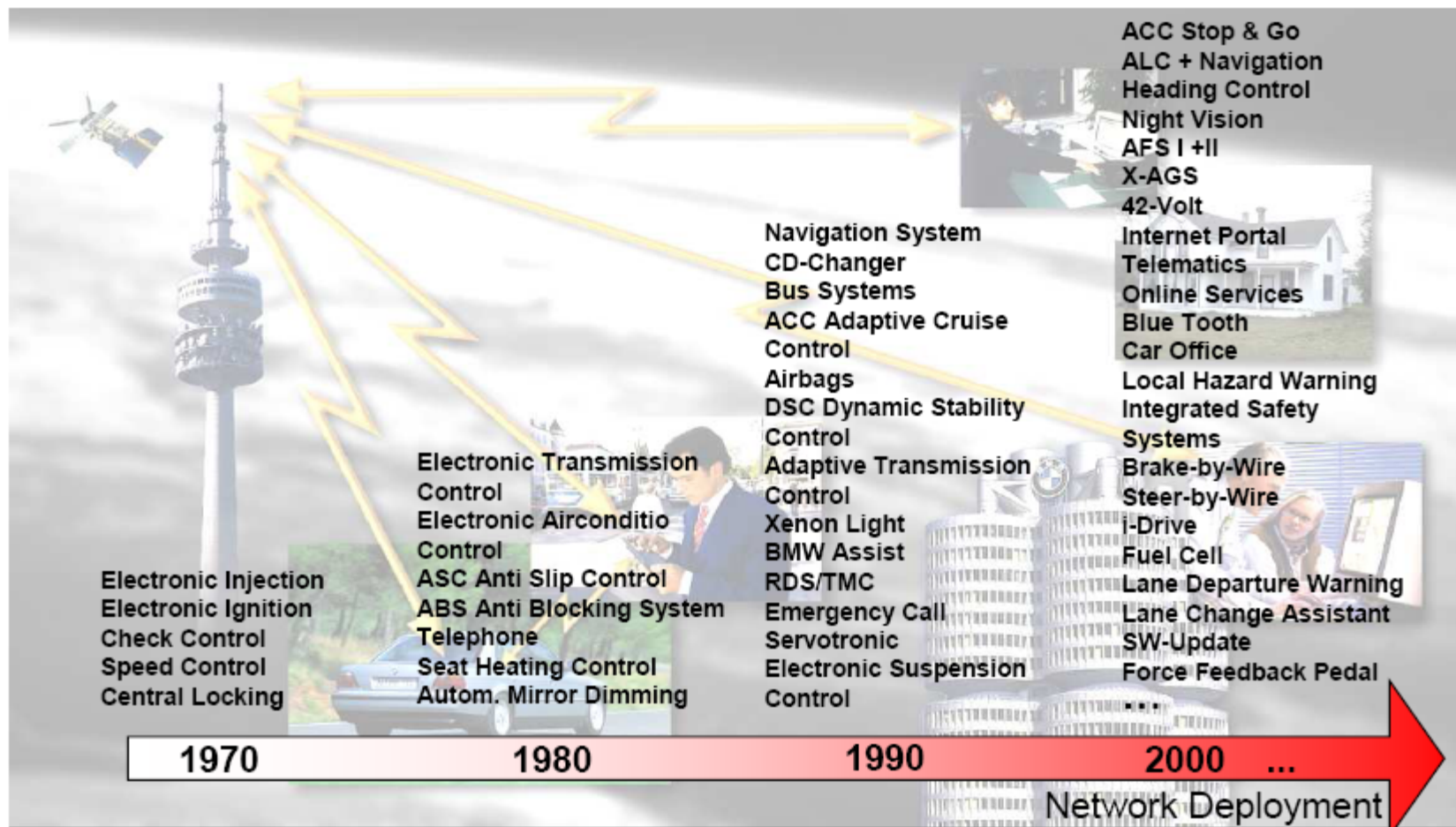
□ 排放

- 从温室效应到清洁（新能源）汽车

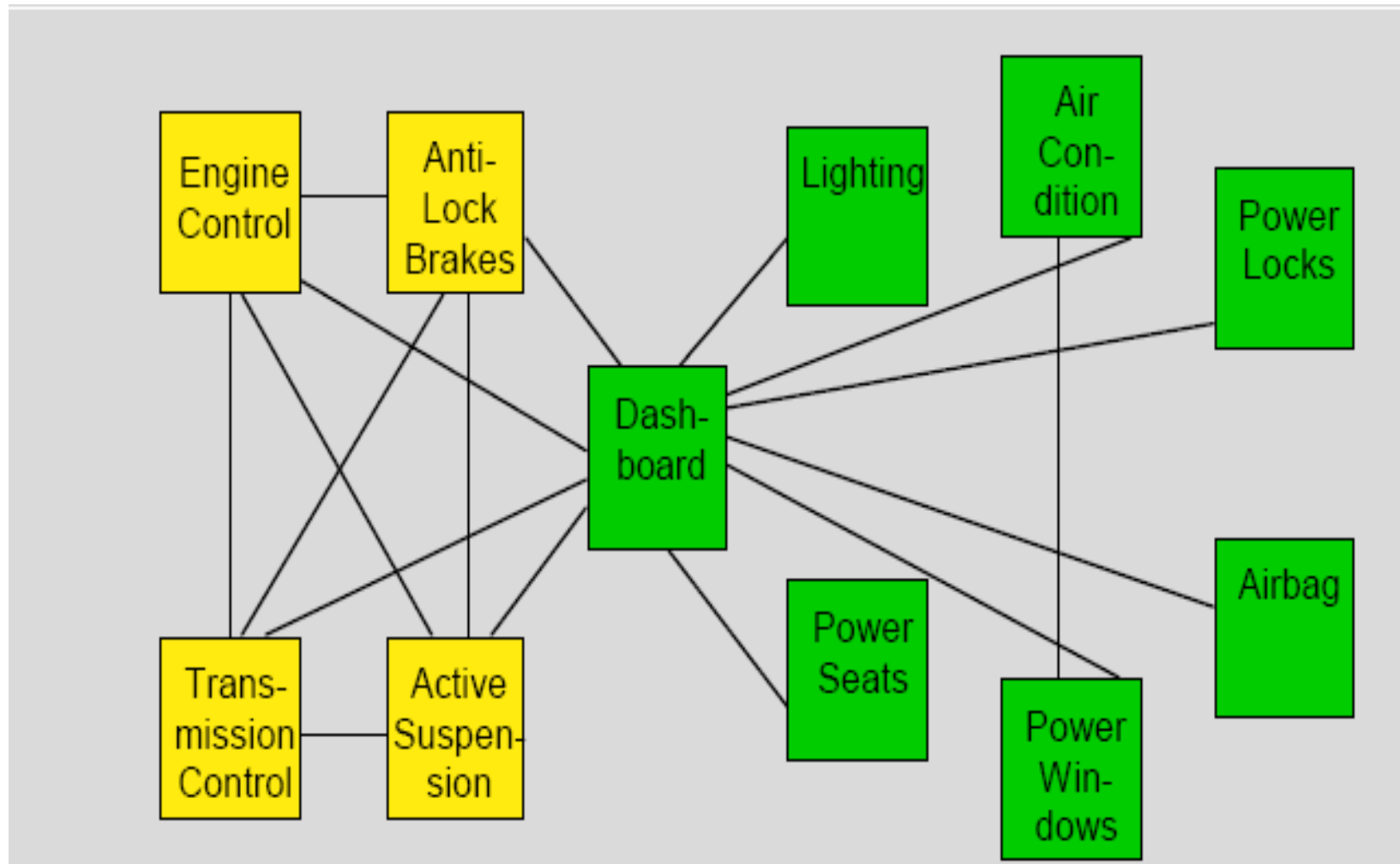
□ 舒适性

- 汽车仅仅是交通工具吗？

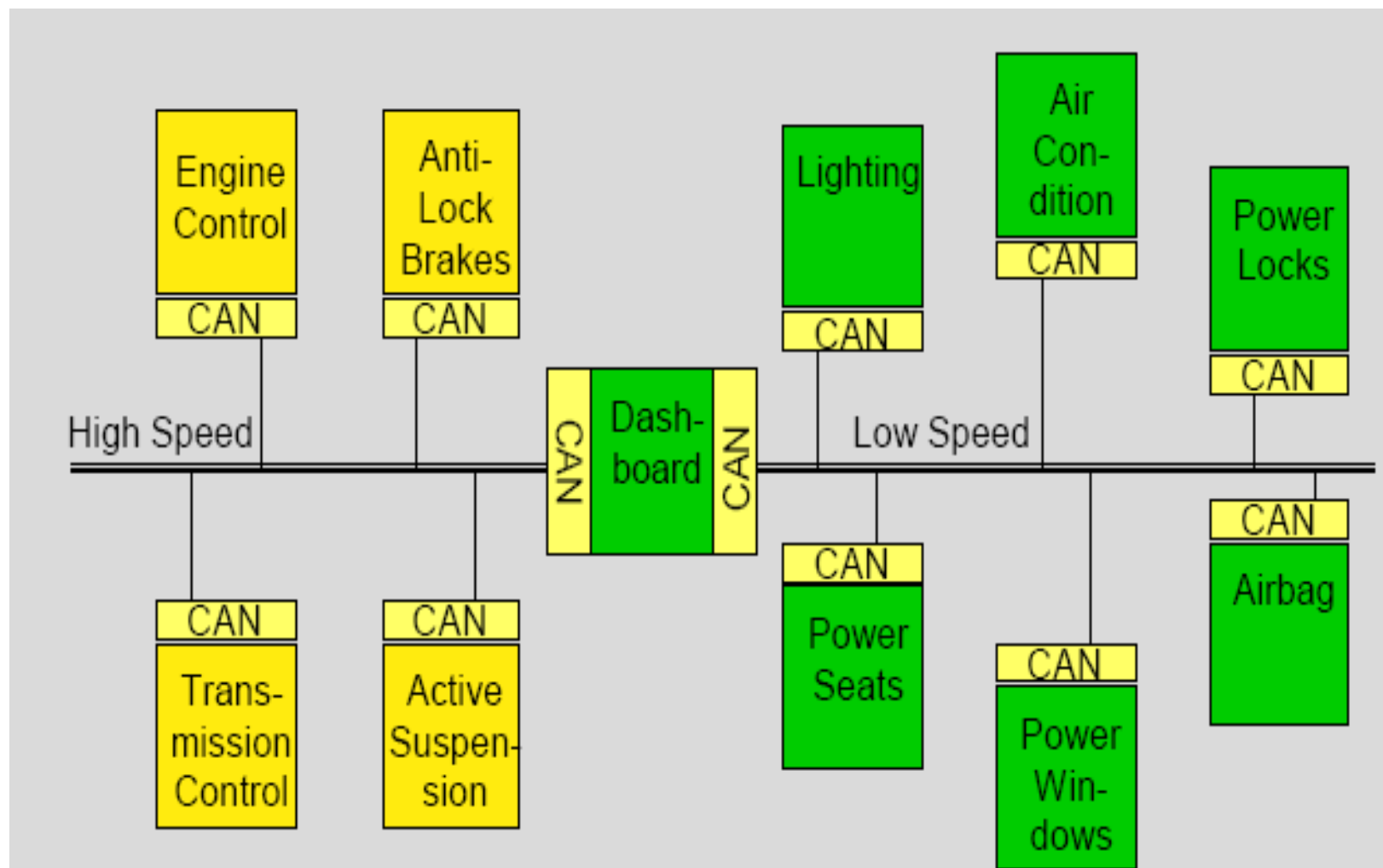
汽车电子发展趋势



早期的ECU通信



基于总线（CAN）的ECU通信



❑ 曾经的车用总线

- ❑ SAE J1850 (Class2)

- ❑ SAE J1708

- ❑ K-Line

- ❑ BEAN

- ❑ byteflight, K-Bus

- ❑ D2B

❑ 当前的车用总线

- ❑ CAN

- ❑ LIN

- ❑ FlexRay

- ❑ MOST

CAN总线的发展历史

- 1983 由Bosch和Intel共同开发
- 1987 第一块CAN控制器芯片（Intel）
- 1990 第一辆应用CAN的量产车：Mercedes S-Class
- 1991 CAN 2.0发布（PART A与PART B）
- 1993 CAN成为ISO标准（ISO 11898）

□ 与CAN总线相关的标准

- ISO 11898, ISO 16845

- SAE J1939, ISO 11783, NMEA 2000, CANopen...

- ISO 15765/14229

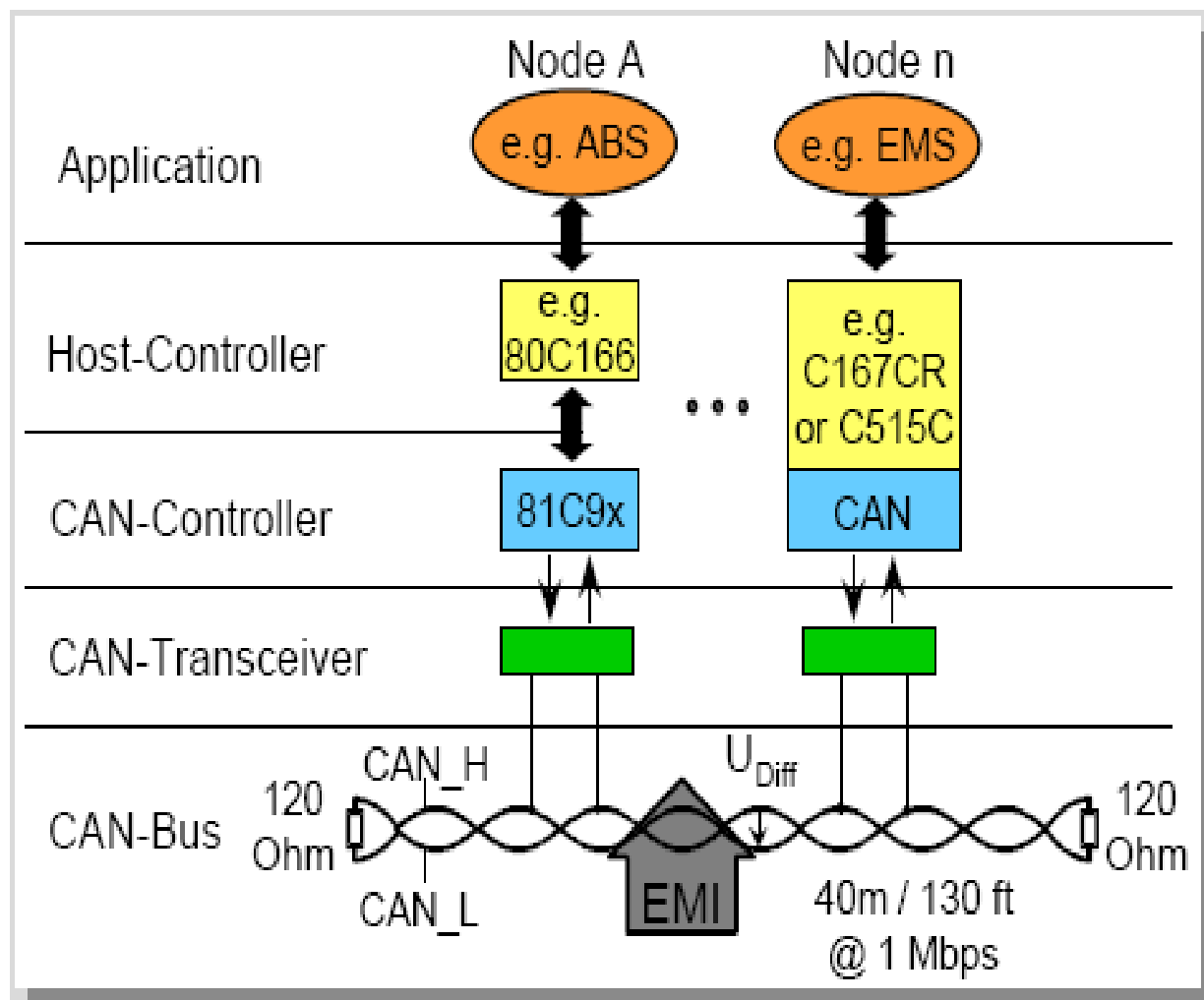
- ISO 17356/OSEK

- CCP (CAN Calibration Protocol)

- GMLAN, VWTP, FNOS, DCNet, MCNet

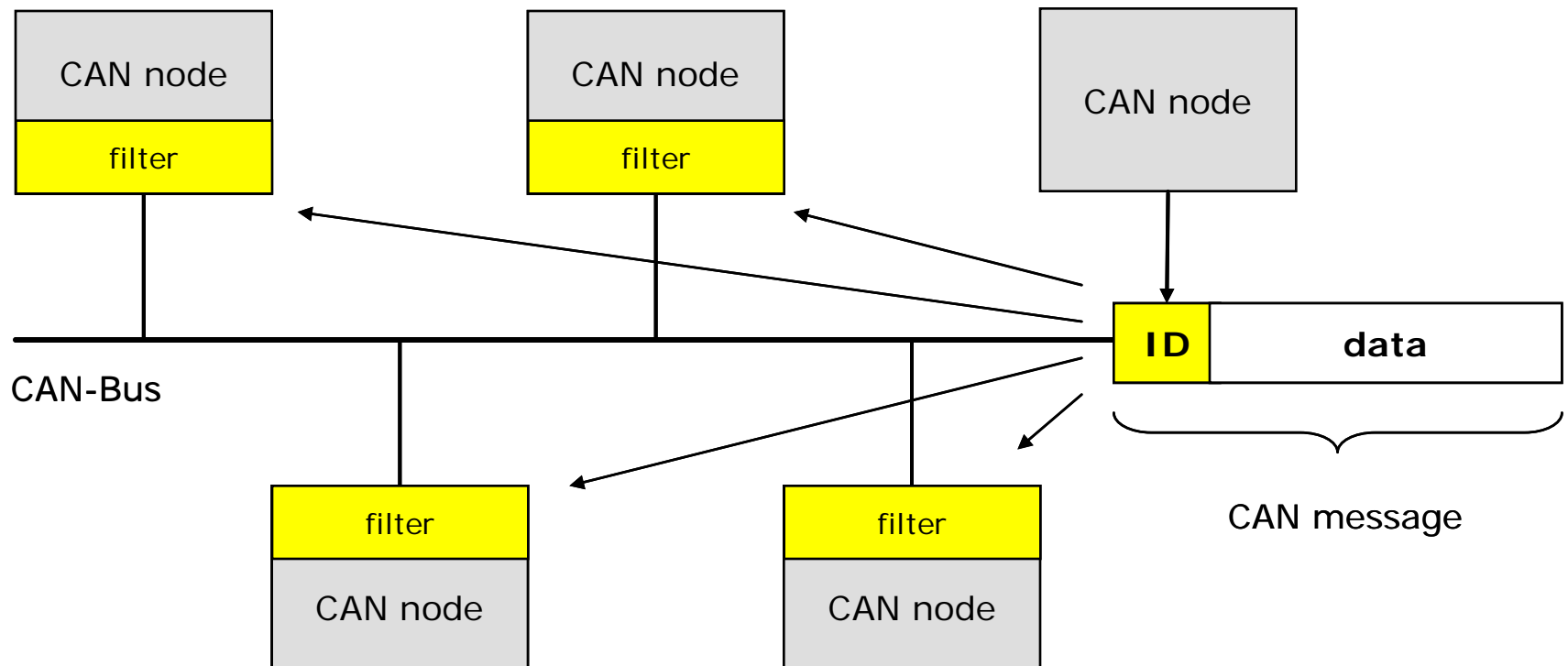
- 多主系统
- 最高速率1 Mbit/sec
- 短帧结构（每条报文最多8字节数据）
- 错误检测与处理机制
- 数据校验，帧内应答
- 总线型拓扑结构
- 广播发送
- 基于优先级的总线仲裁机制

CAN总线基本结构

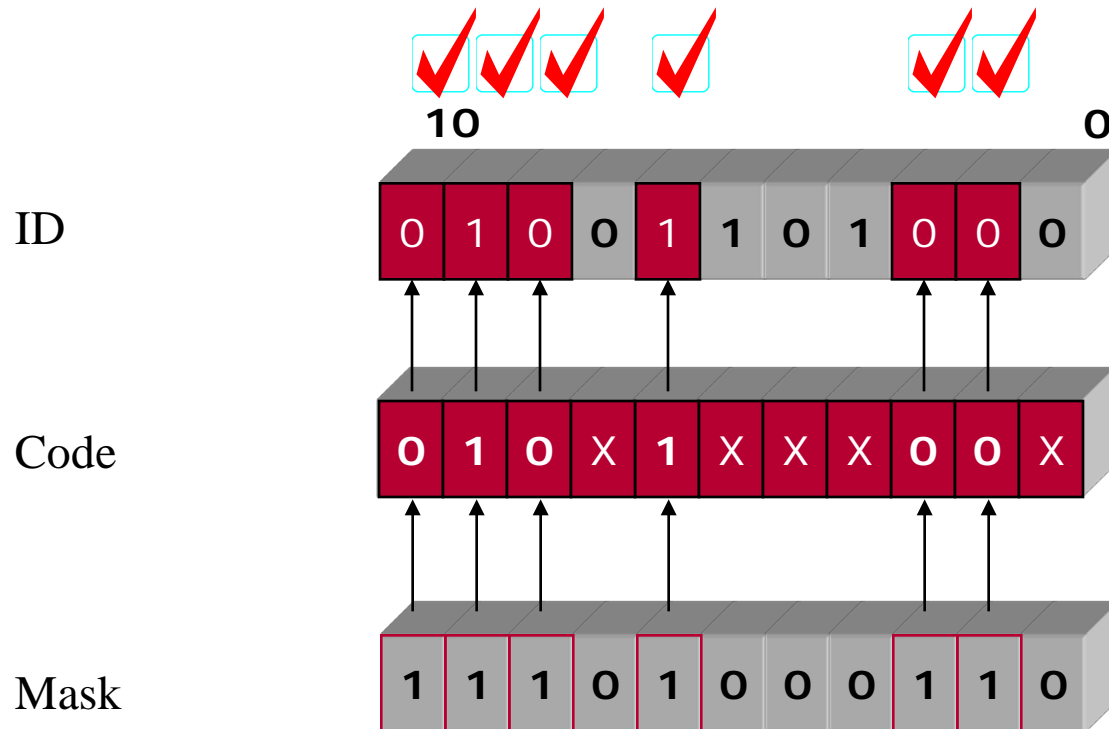


- ❑ 在CAN总线上传输的数据报文不包含发送节点和接收节点的信息
- ❑ 每个报文的内容通过标识符（ID）识别，标识符在网络中是唯一的
 - ❑ 标识符描述了数据的含义，同时也是决定优先级的主要因素
- ❑ 报文可以被所有节点同时接收（广播）
- ❑ 可以进行报文过滤

广播与过滤 (1/2)



广播与过滤 (2/2)

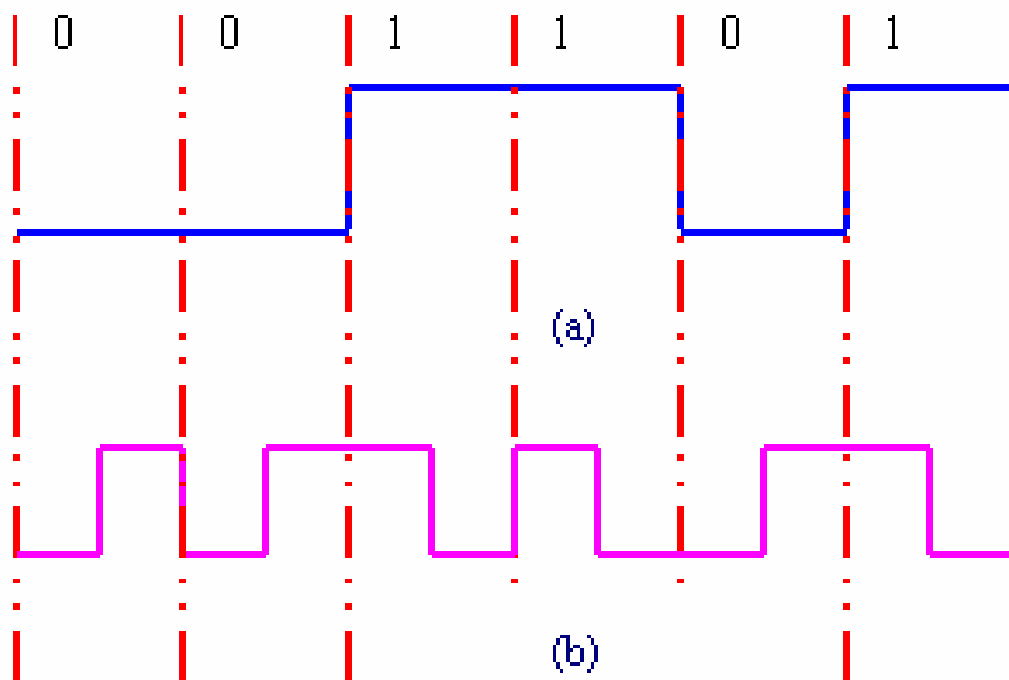


标识符（ID）

- ❑ 标识符是唯一的，它描述了数据的特定含义，也决定了报文的优先级
 - ❑ 标识符数值越小，优先级越高
- ❑ 最高优先级的报文在总线仲裁的过程中获得总线访问权
- ❑ 低优先级报文在下一个总线空闲自动重发

NRZ(Non-Return to Zero)编码

- 脉冲跳变最少，对外界干扰的抵抗能力强

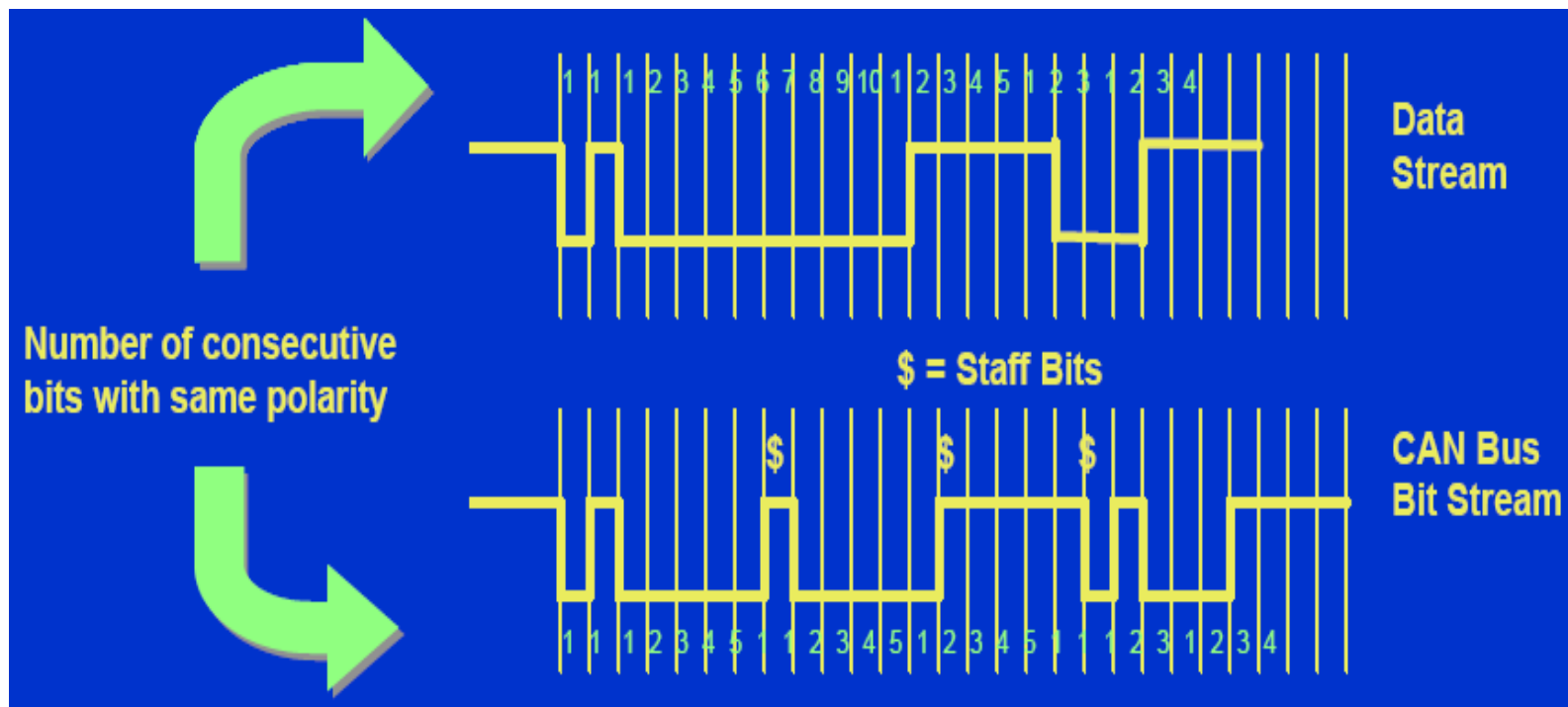


NRZ编码

曼彻斯特编码

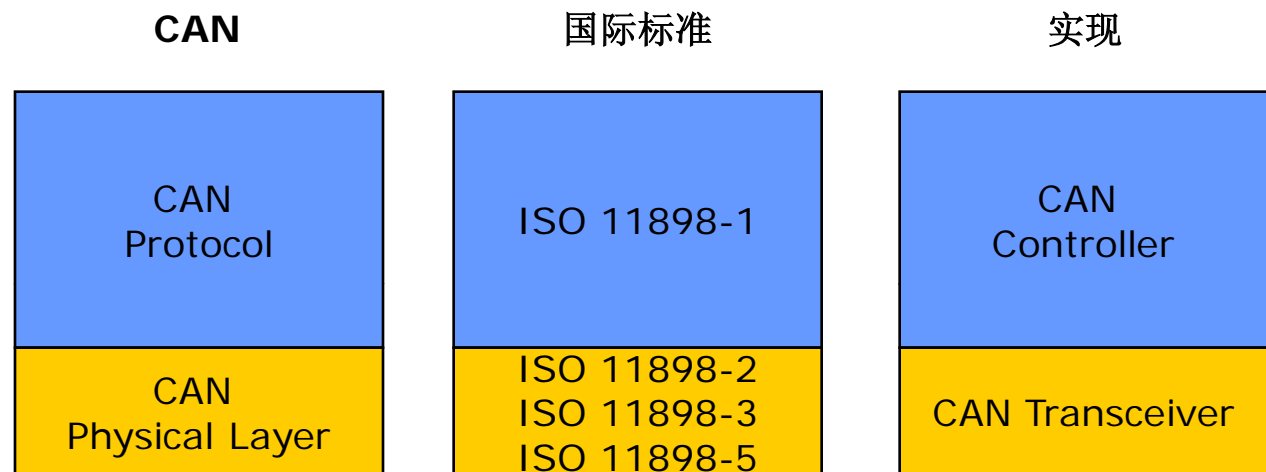
位填充

- 位填充是为了提供足够的跳变沿
 - 填充位出现在5个连续的相同极性的位之后
 - 填充位与其前面的位极性相反



□ ISO 11898

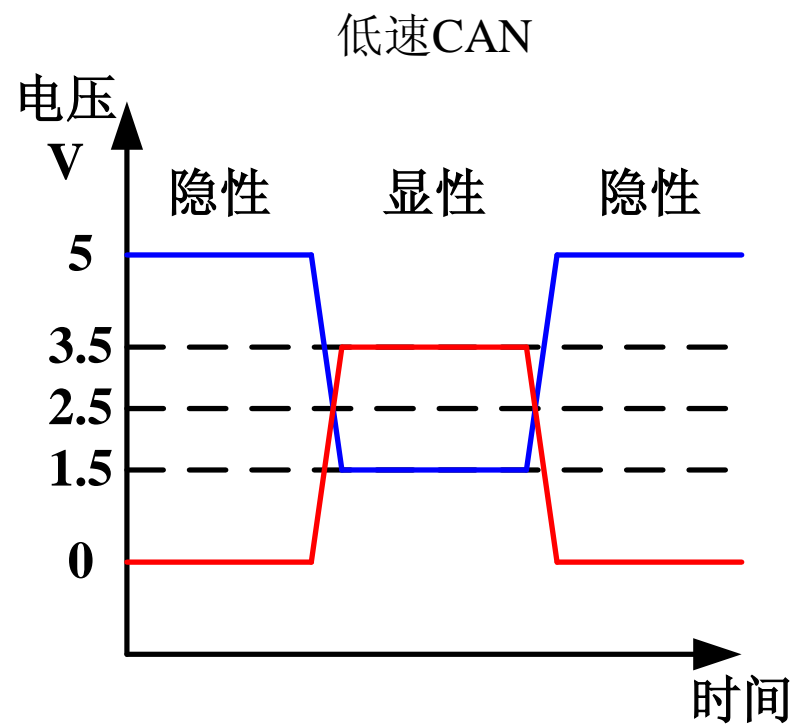
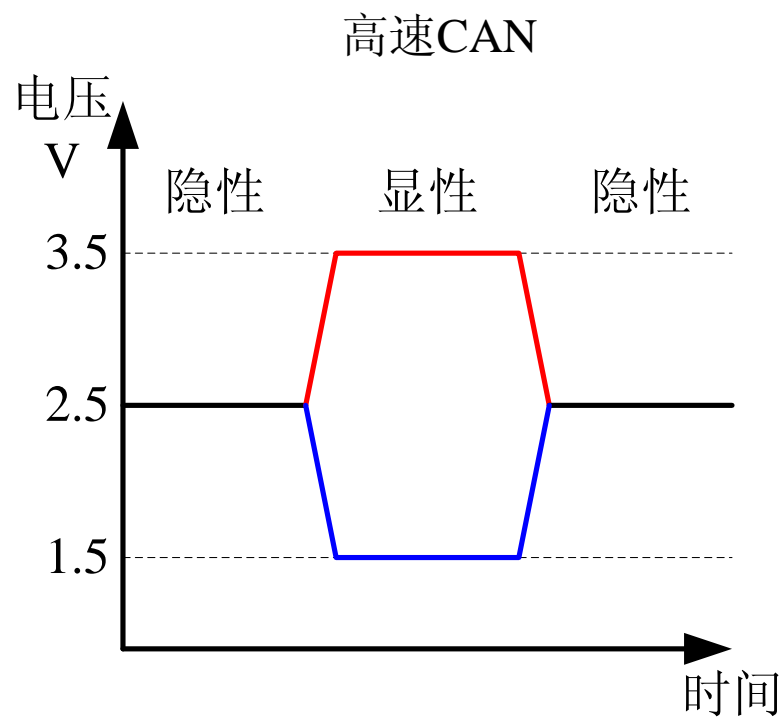
- 11898-1 Data link layer and physical signalling
- 11898-2 High-speed medium access unit
- 11898-3 Low-speed, fault-tolerant, medium-dependent interface
- 11898-5 High-speed medium access unit with low-power mode



显性位与隐性位

□ 显性位=0

□ 隐性位=1



Two logic states
possible on the bus:
"1" = recessive
"0" = dominant



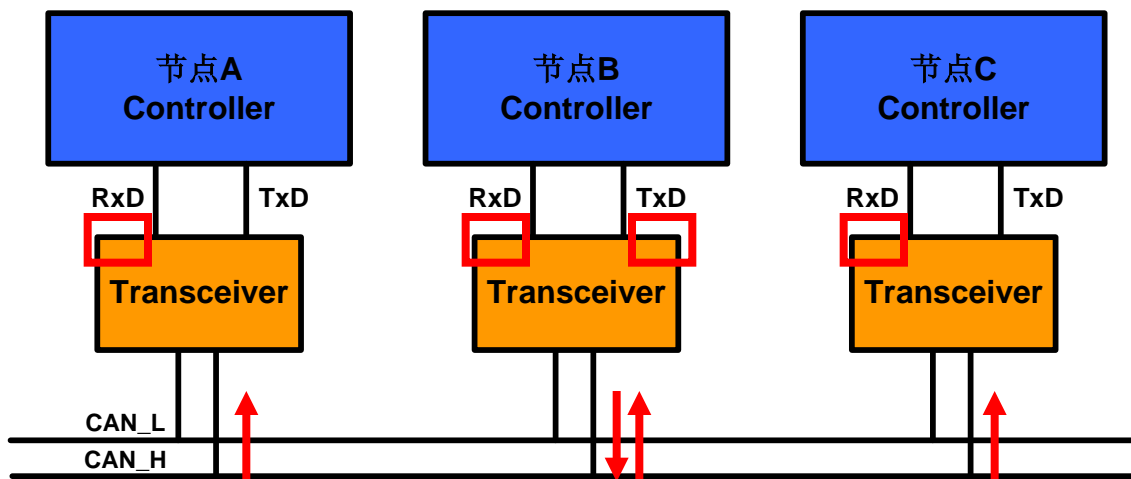
"1" = recessive
"0" = dominant

A	B	C	BUS
D	D	D	D
D	D	R	D
D	R	D	D
D	R	R	D
R	D	D	D
R	D	R	D
R	R	D	D
R	R	R	R

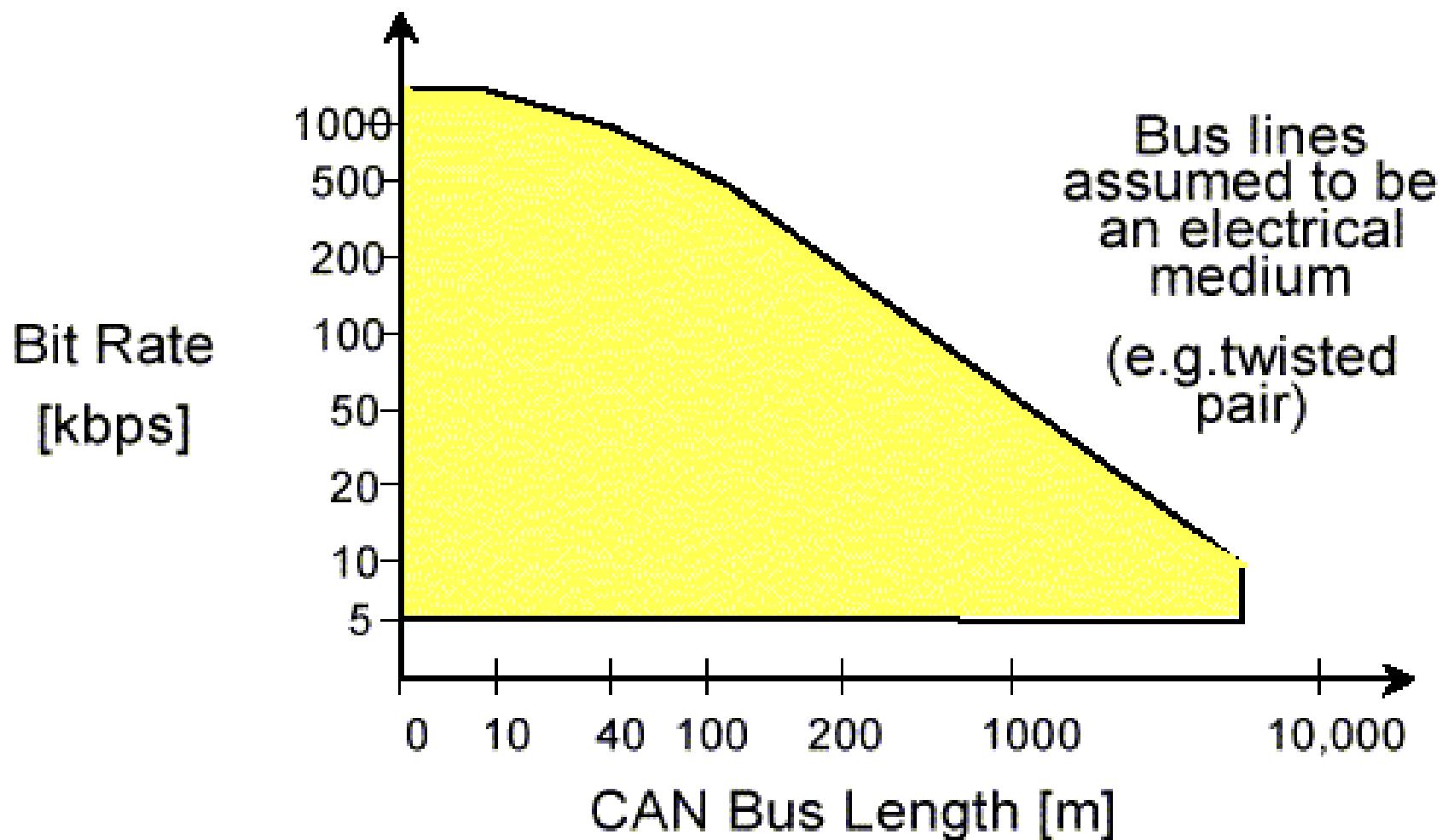
As soon as one node nodes transmits
a dominant bit (zero):
Bus is in the dominant state.

Only if all nodes transmit
recessive bits (ones):
Bus is in the recessive state.

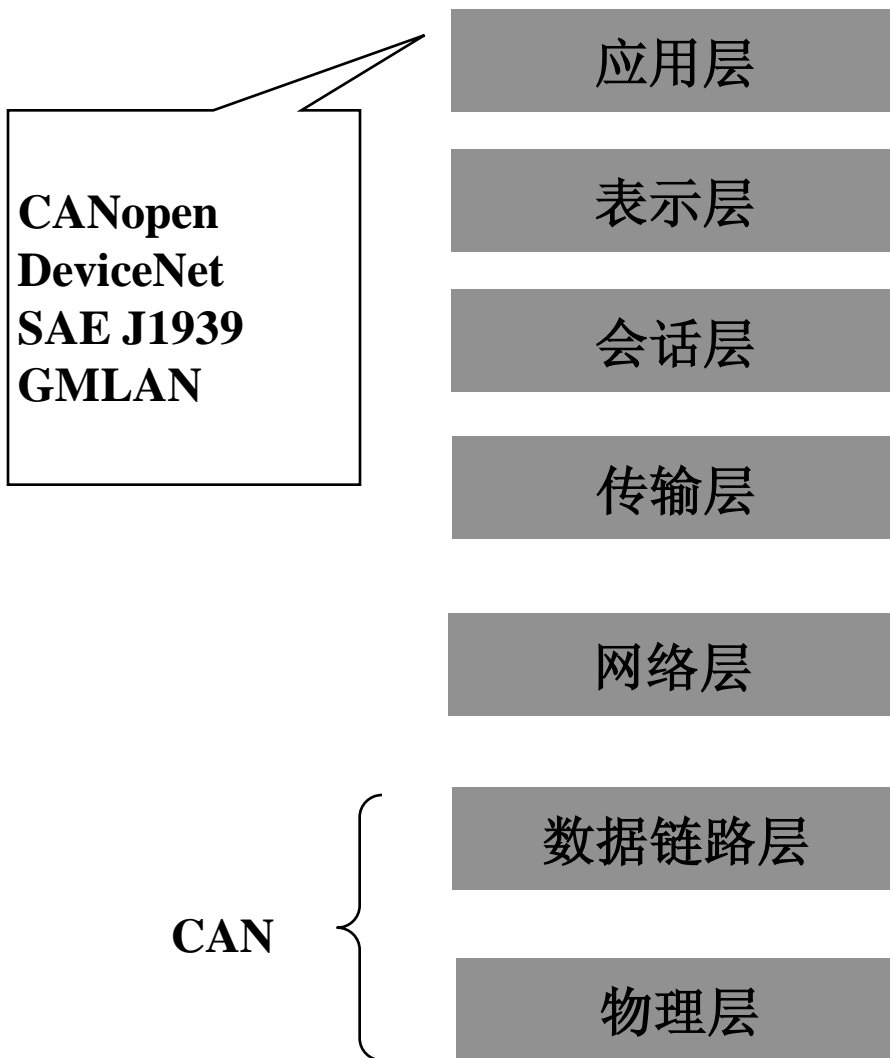
- 节点在发送每个位的同时读取总线上的电平信号



总线长度与波特率



OSI参考模型

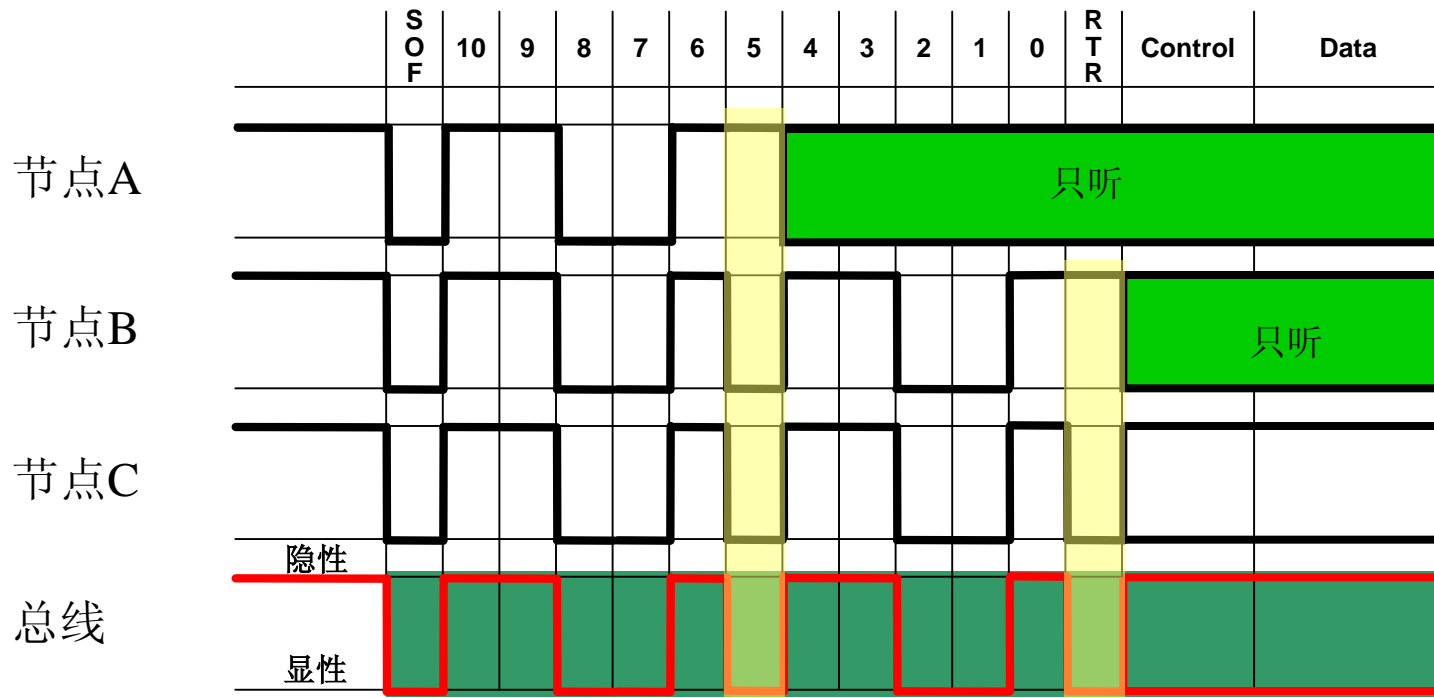


- CAN总线概述
- 数据链路层
 - 总线访问仲裁
 - 帧格式
 - 错误处理
 - 位定时与同步
- 物理层

□ 总线访问机制

- 载波侦听和带冲突检测协议的多路访问（CSMA/CD）
- CAN报文的优先级由标识符值决定
 - 标识符的数值在系统设计的初始阶段分配
 - 不同节点不允许发送相同ID报文（远程帧除外）
 - 标识符数值越小，优先级越高。
- 总线冲突通过非破坏性位序列仲裁解决。

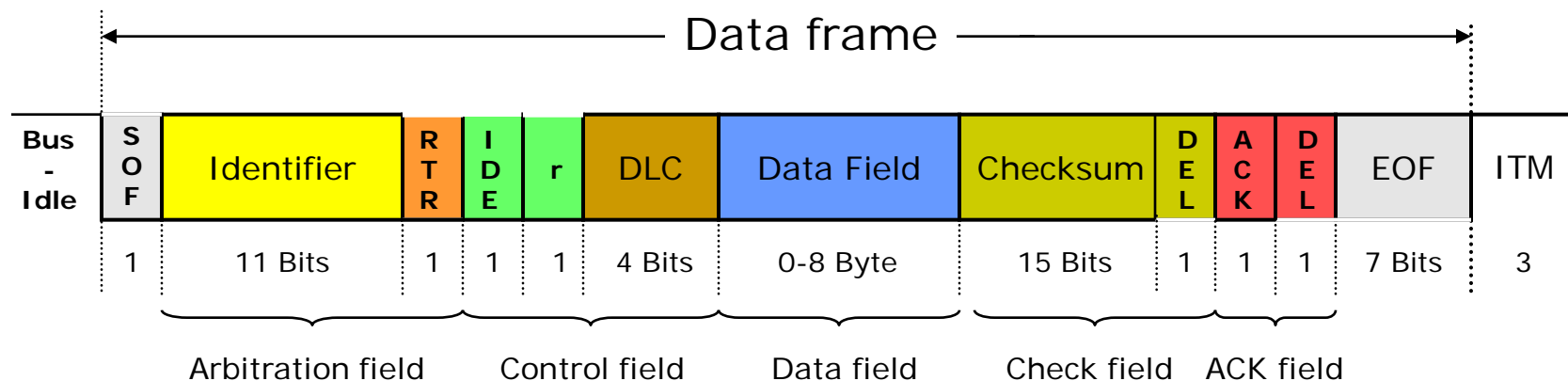
总线仲裁 (2/2)



帧格式

- 数据帧
- 远程帧
- 错误帧
- 过载帧
- 帧间空间

数据帧



SOF Start Of Frame
帧起始位

RTR Remote Transmission Request
远程传输请求位

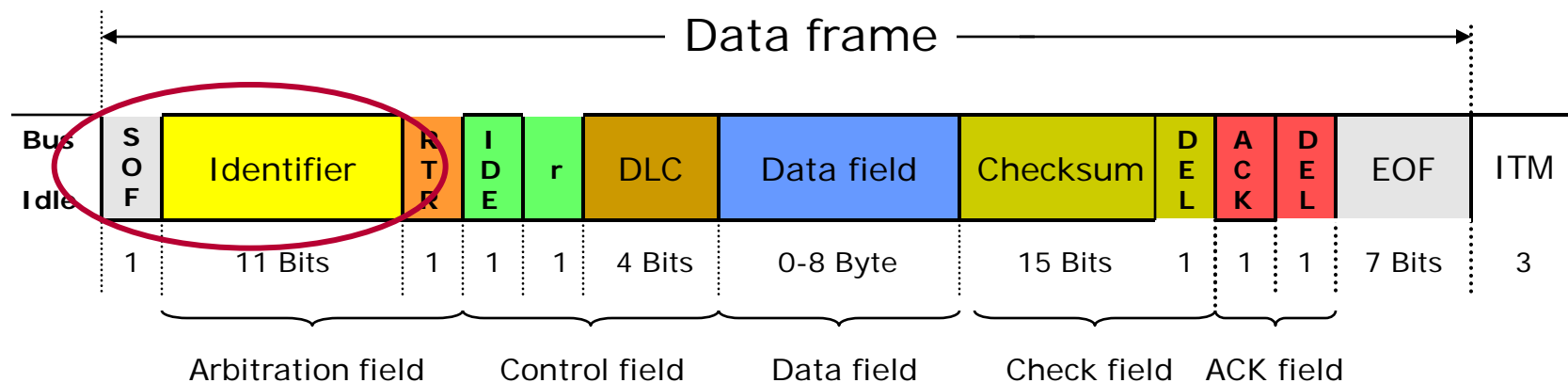
IDE Identifier Extension
标识符扩展

DLC Data Length Code
数据长度代码

ACK Acknowledgement
应答

EOF End Of Frame
帧结束场

SOF和标识符

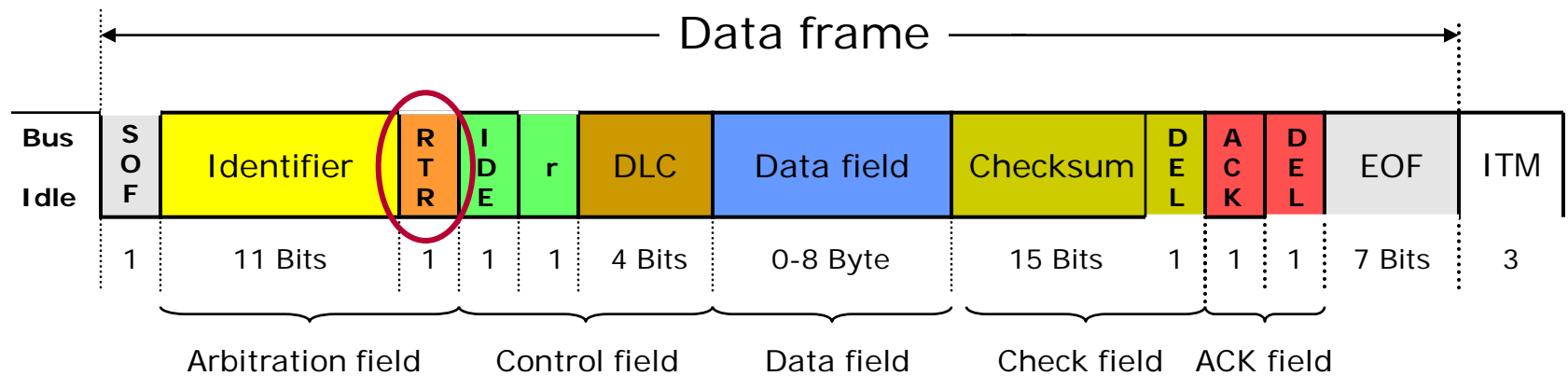


- **SOF**

- 一个显性位，表明一帧的开始

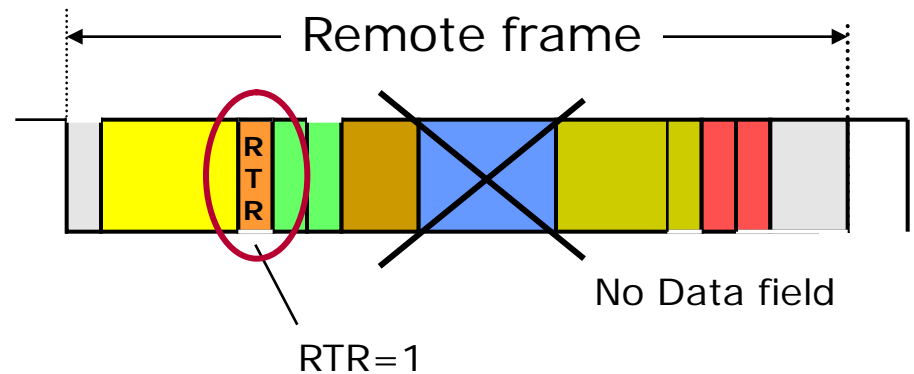
- **Identifier (ID)**

- 姓名
- 职务

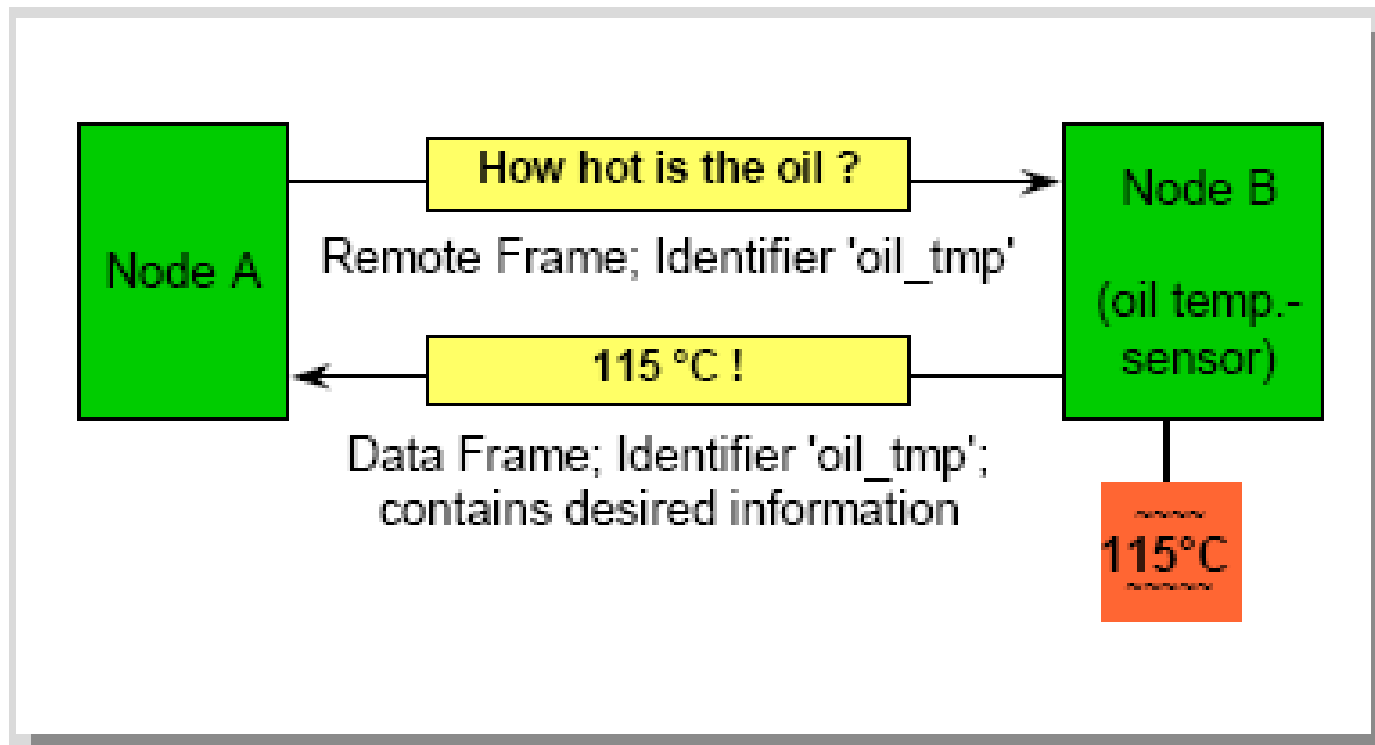


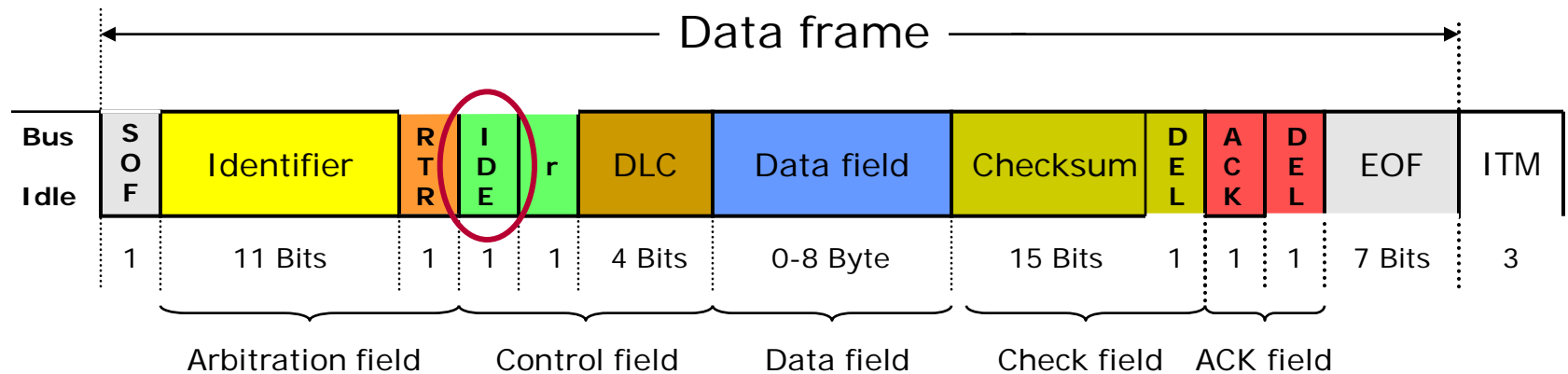
● RTR – 远程传输请求位

- RTR = 0: 数据帧
- RTR = 1: 远程帧



远程帧的使用



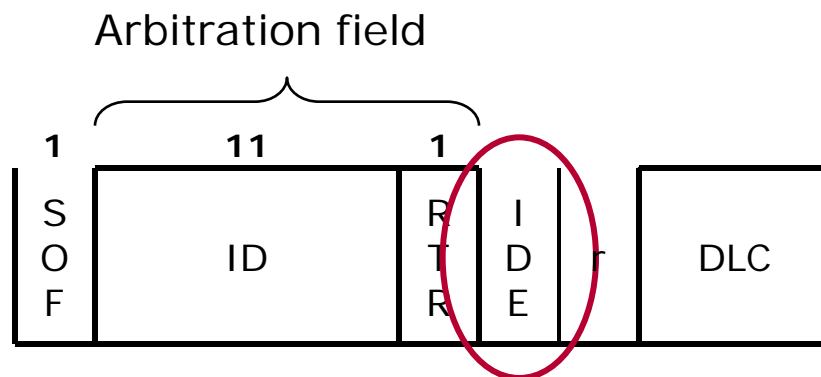


- **IDE** – 标识符扩展位

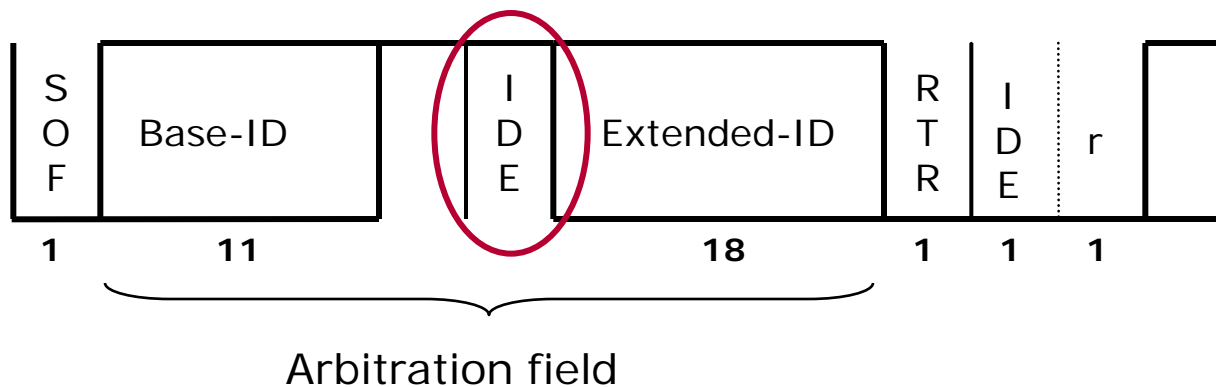
- IDE = 0: 11 bits → 标准CAN
- IDE = 1: 29 bits → 扩展CAN

标准CAN与扩展CAN

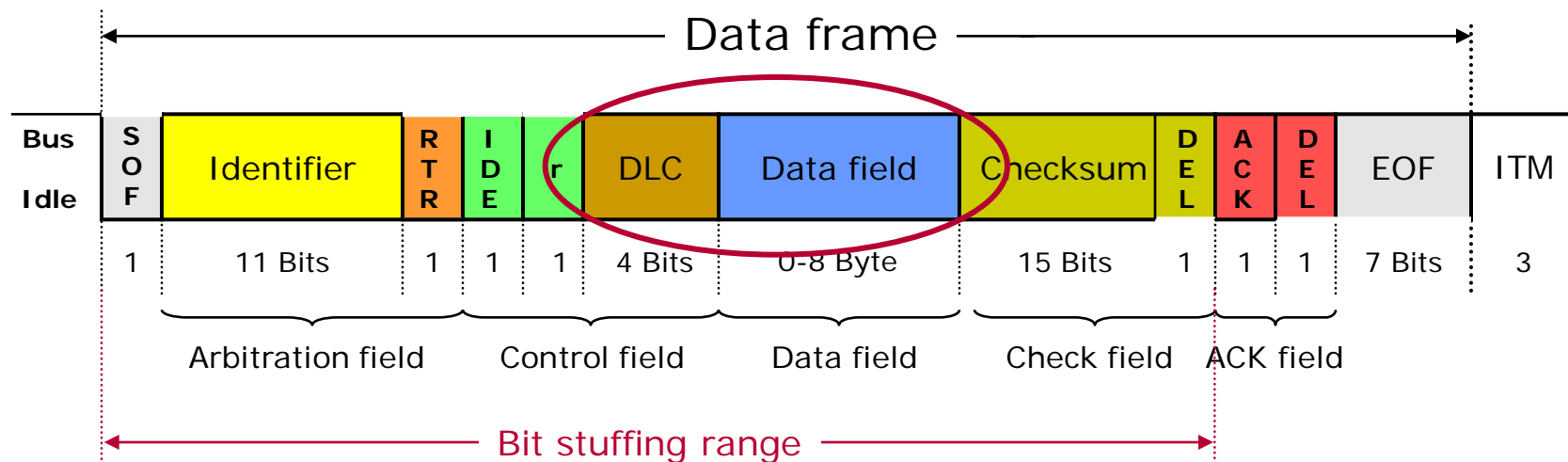
IDE=0:
标准CAN
2048个标识符



IDE=1:
扩展CAN
5亿多个标识符

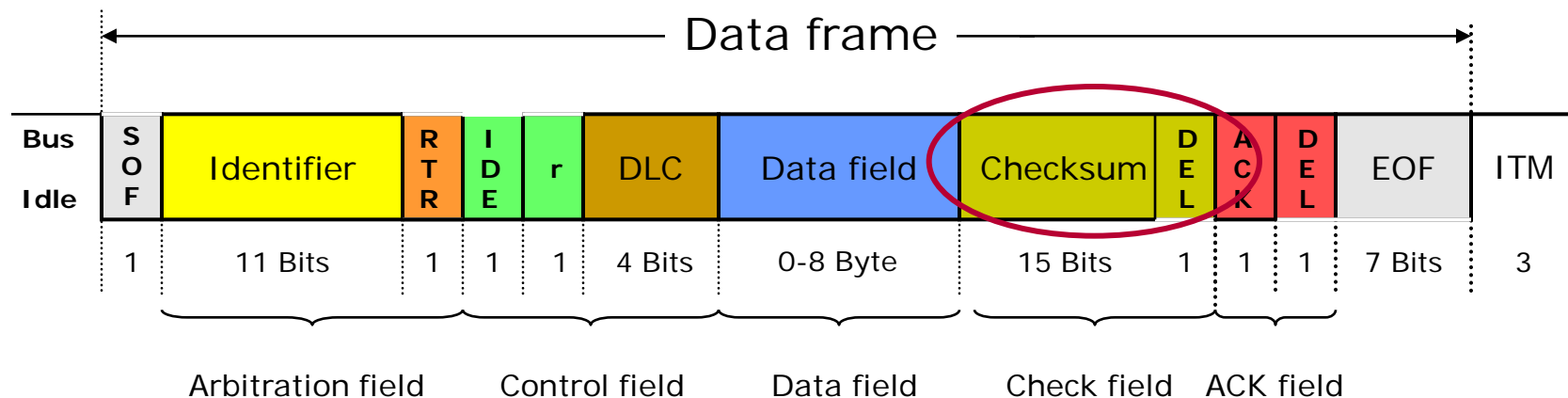


DLC与数据场



- **DLC – 数据长度代码**
 - 0~8

- **数据场**
 - 最大8个字节

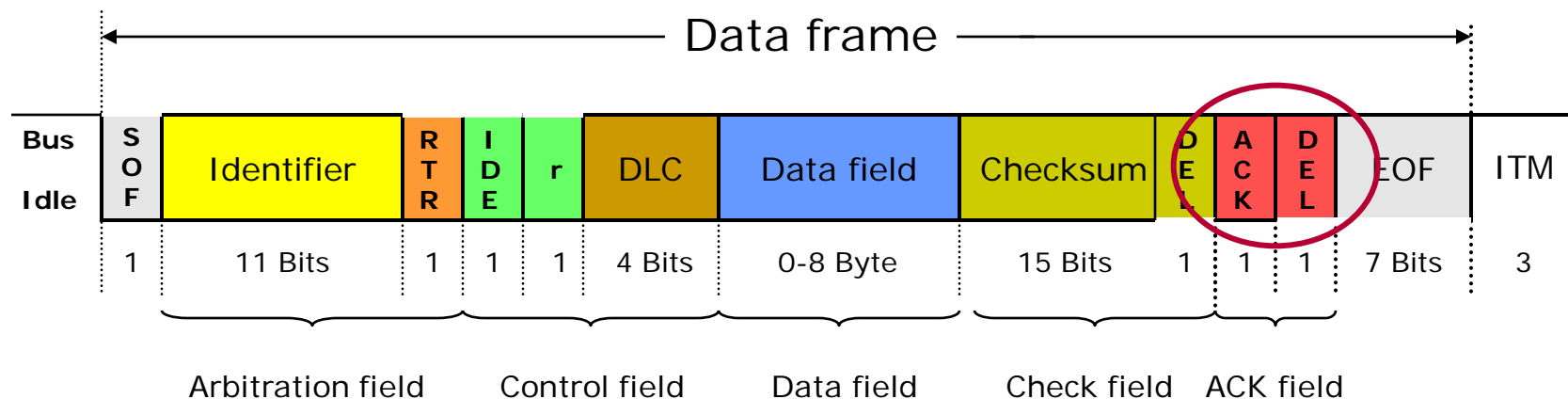


- 校验和

- $G(x) = x^{15} + x^{14} + x^{10} + x^8 + x^7 + x^4 + x^3 + 1$

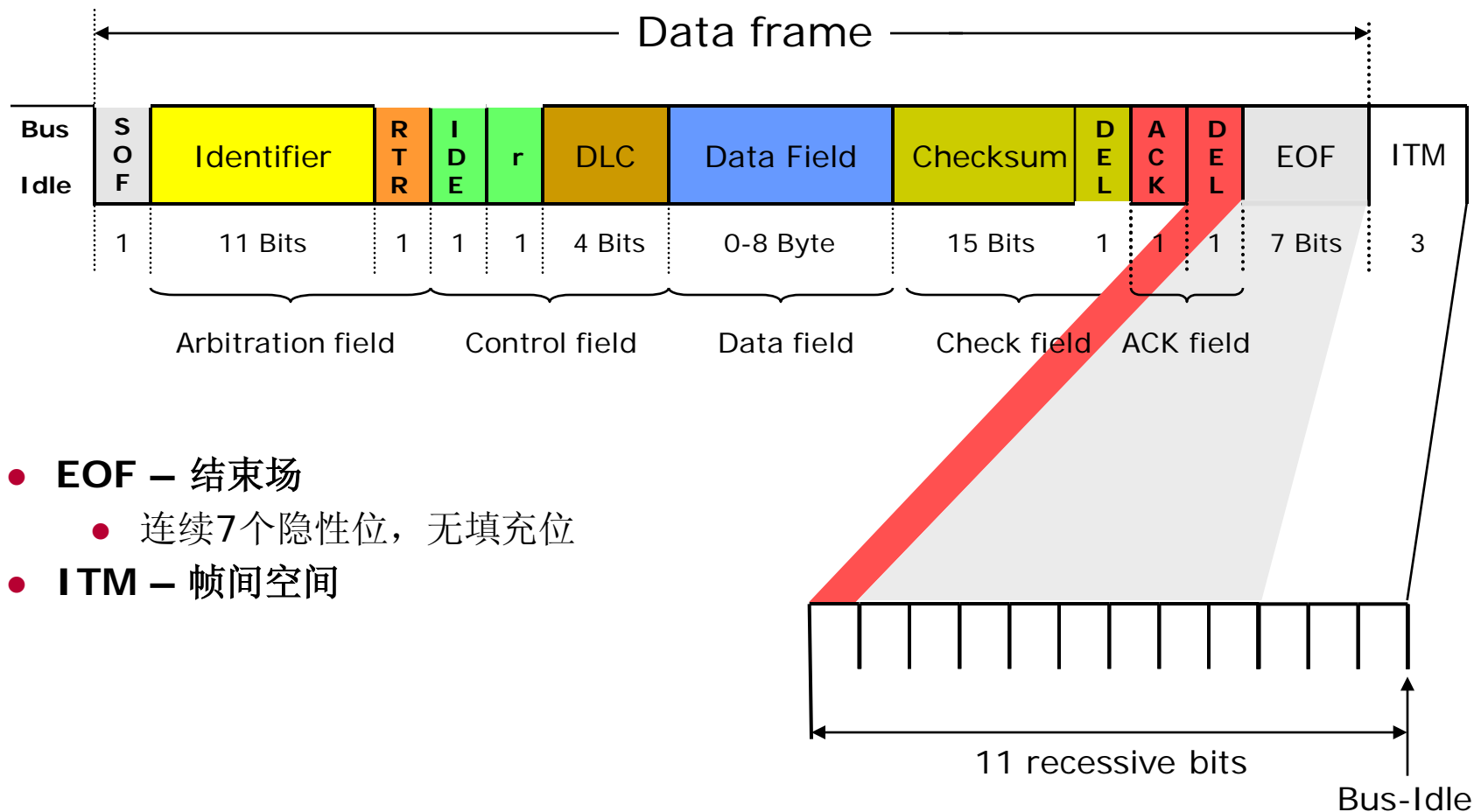
- 校验场分隔符

- 一个隐性位



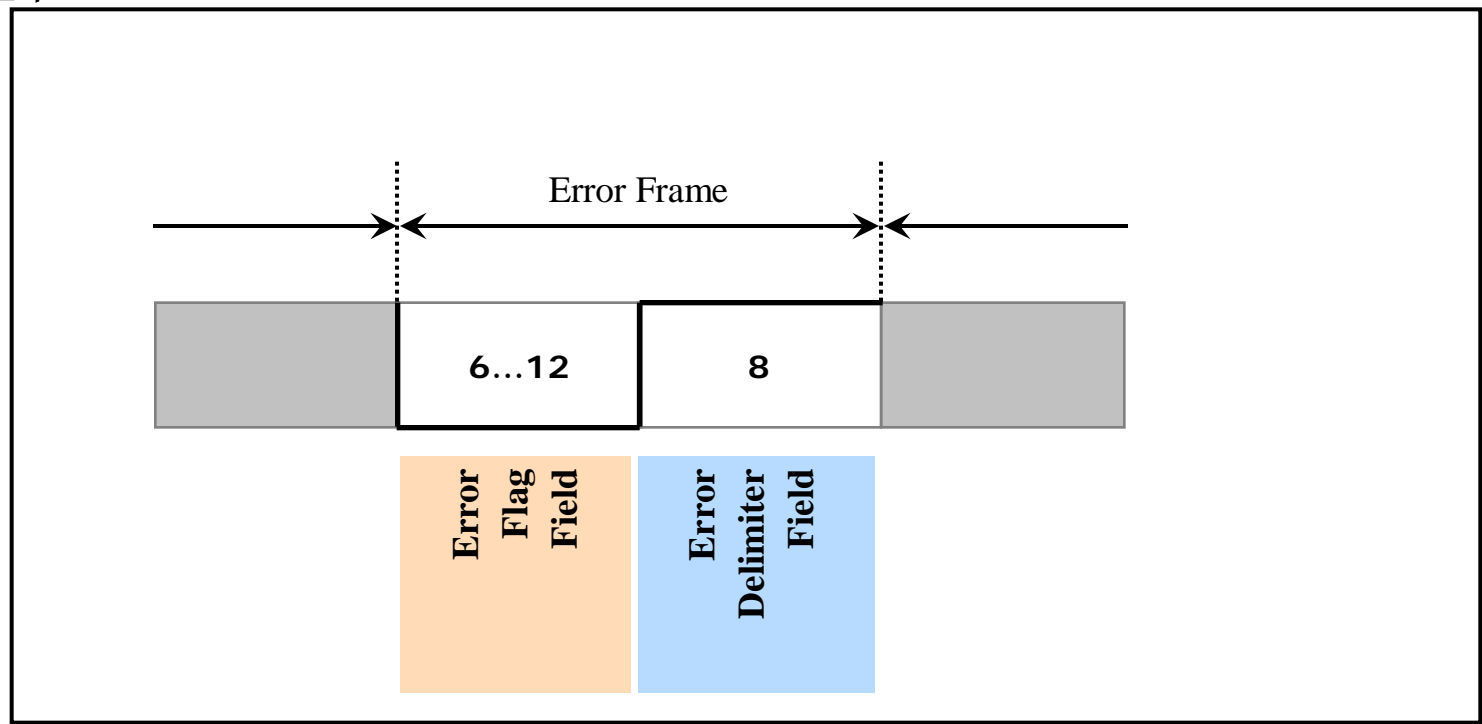
- 应答间隙(ACK Slot)
 - 发送节点: 1
 - 接收节点: 0
- 应答场分隔符(ACK Delimiter)
 - 一个隐性位

EOF与ITM

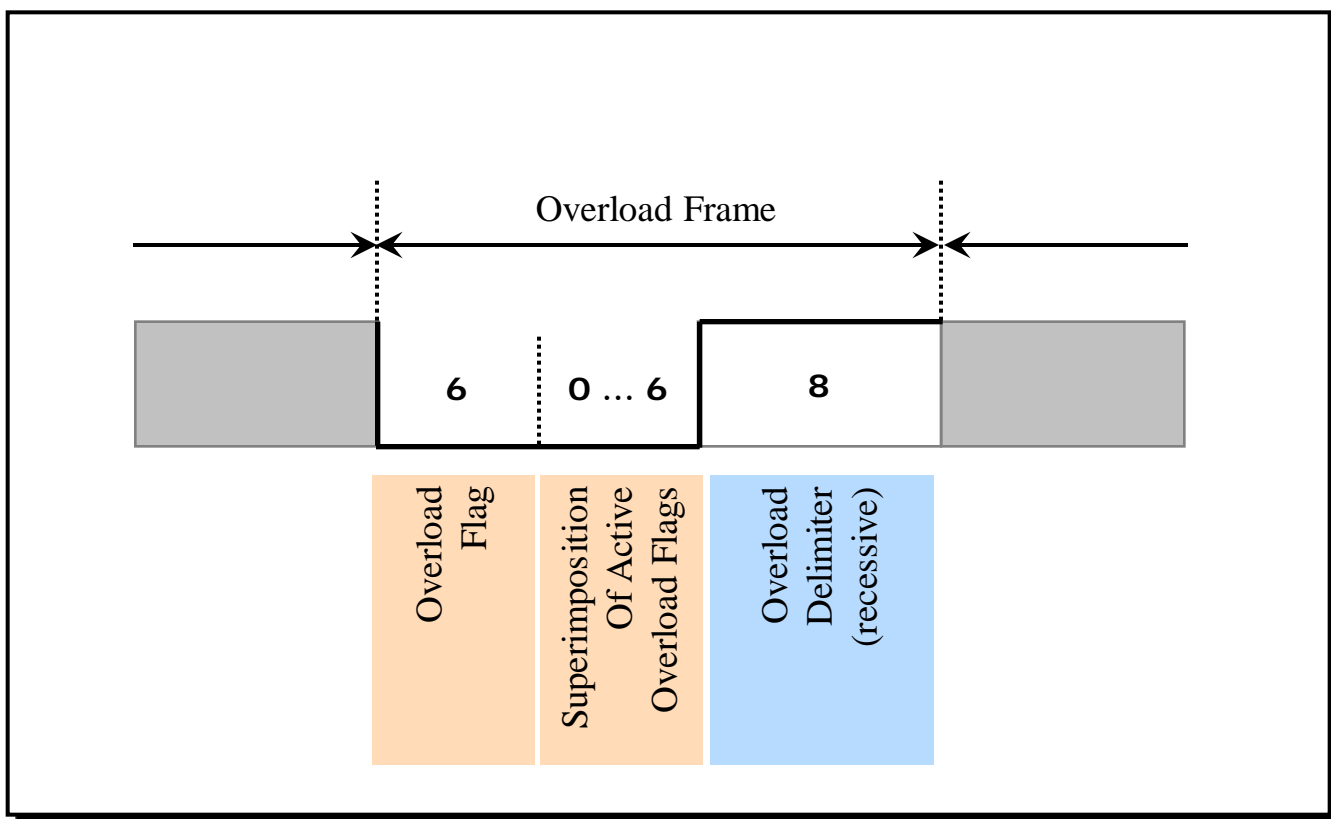


- **EOF** – 结束场
 - 连续7个隐性位，无填充位
- **ITM** – 帧间空间

- 用于表明发生了错误
 - 错误主动与错误被动
 - 无填充位



- 延缓其它节点发送报文
- 每个节点最多连续发送两条过载帧

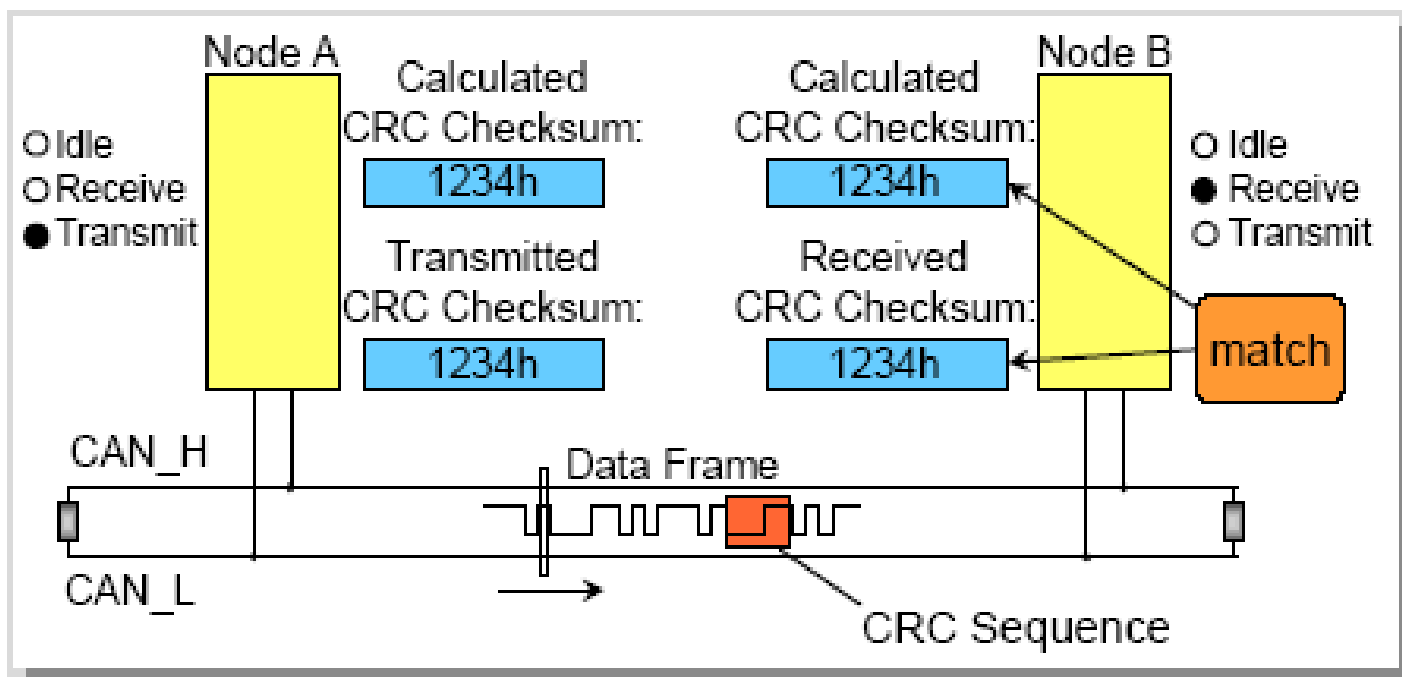


错误分类

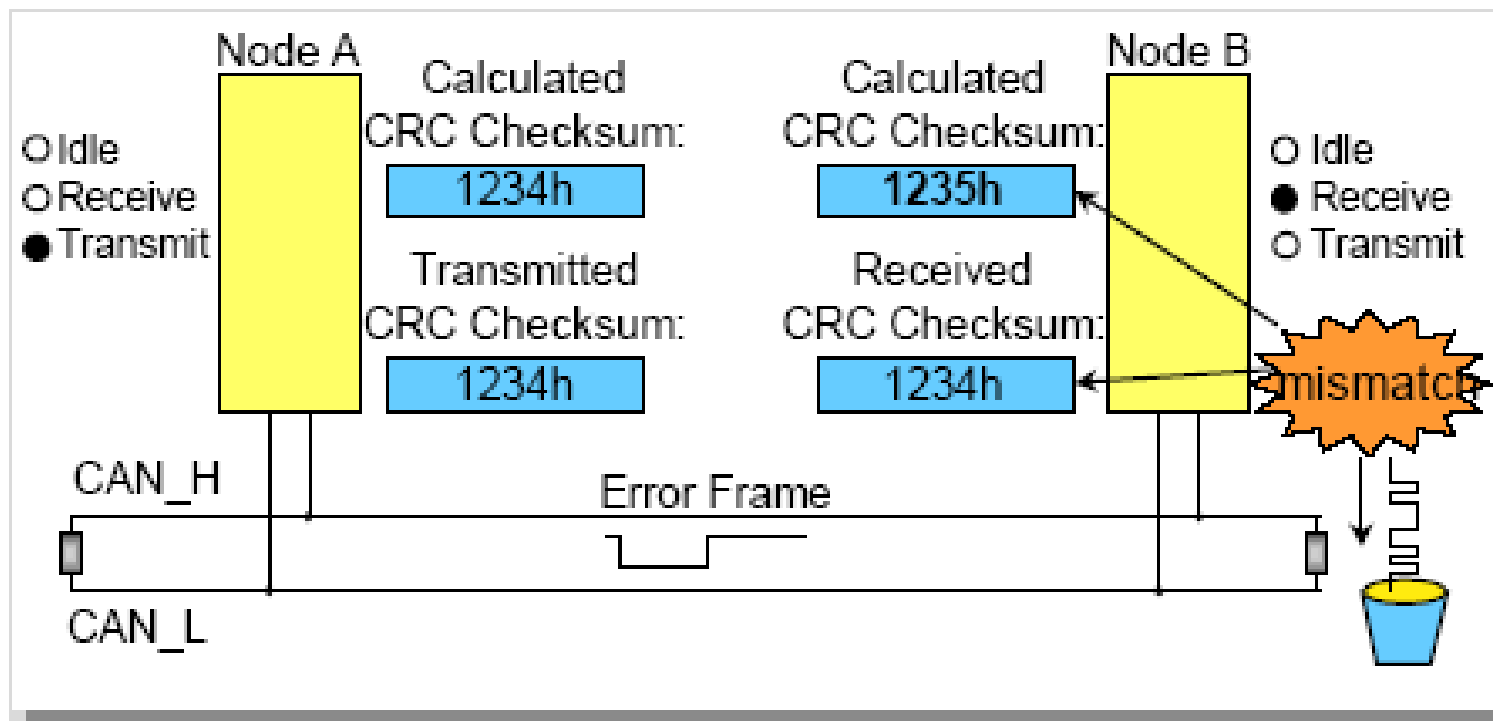
- ❑ CRC错误（CRC Error）
- ❑ 形式错误（Form Error）
- ❑ 填充错误（Stuff Error）
- ❑ 应答错误（ACK Error）
- ❑ 位错误（Bit Error）

CRC校验 (1/2)

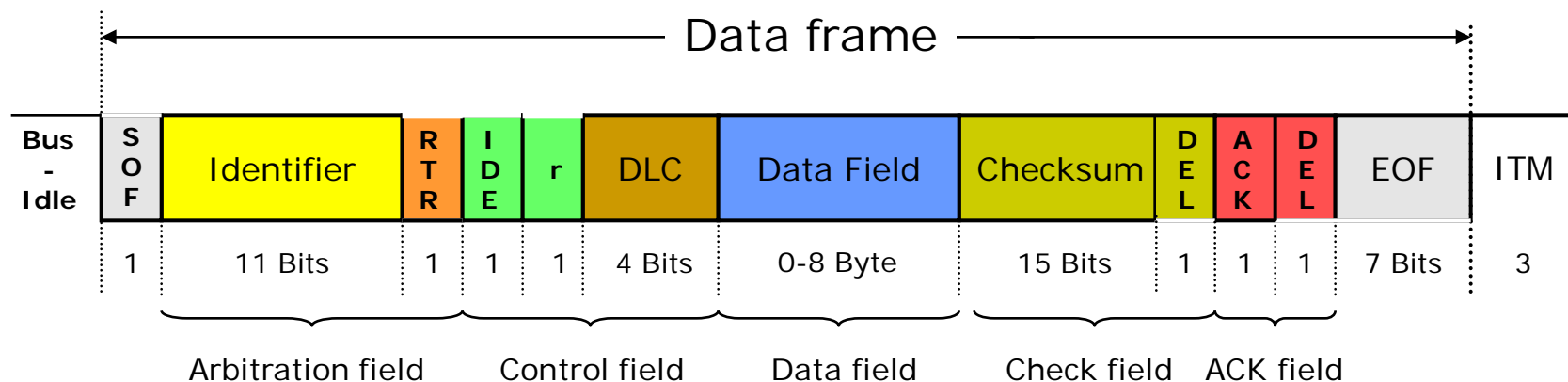
□ 接收到的校验值必须与计算出的校验值一致



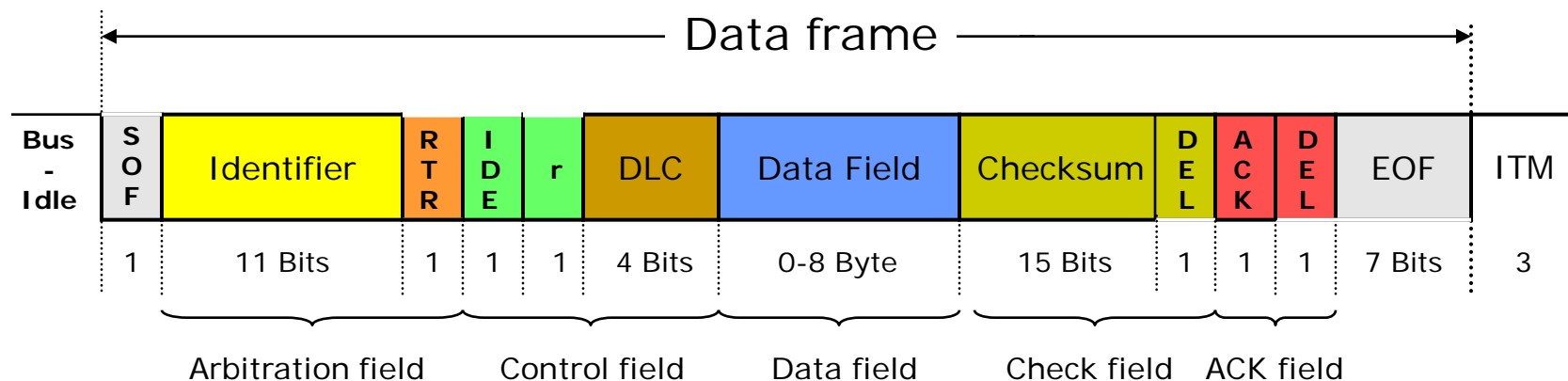
□ 否则，发送错误帧



- 在CRC分隔符、ACK分隔符、帧结束和帧间隔中不允许出现显性位

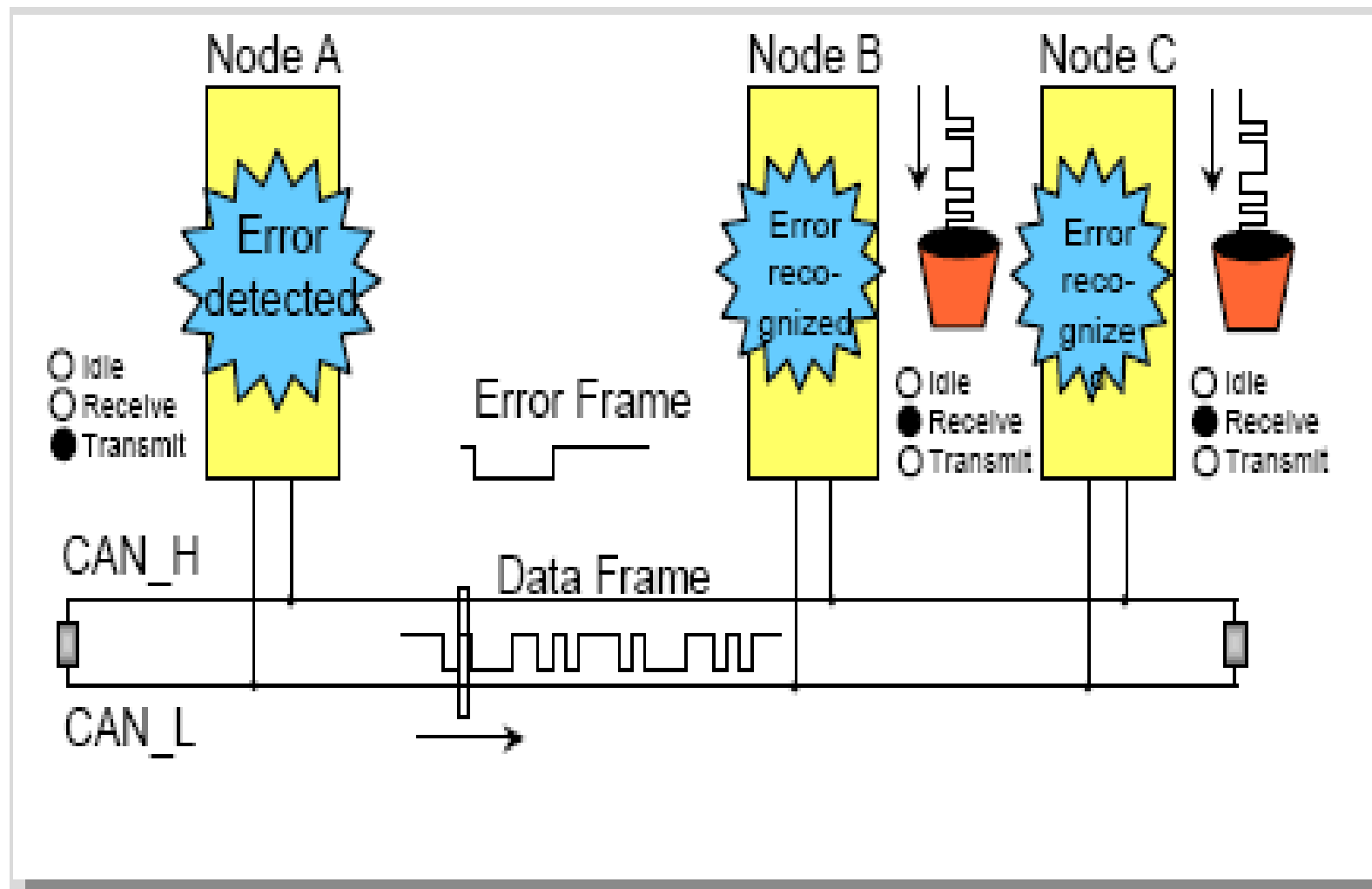


□ 检查区间从SOF到CRC分隔符

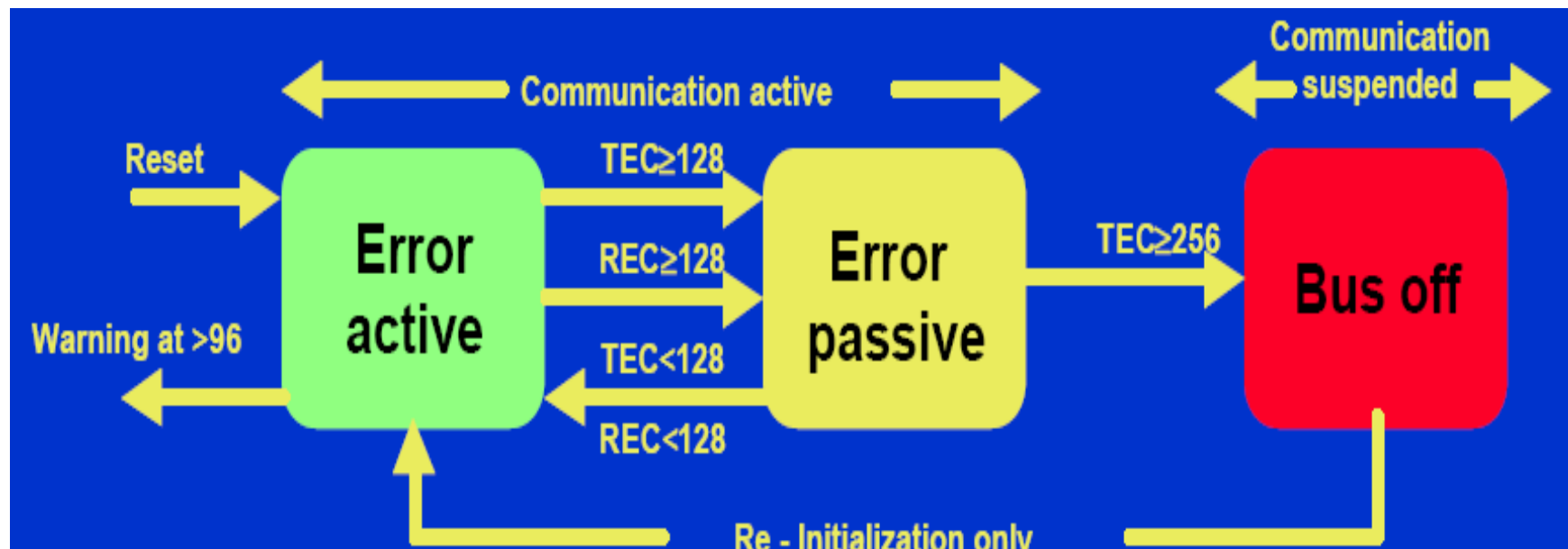


- 所发送的位必须从总线上正确回读，否则就是一个位错误
- 在仲裁场和应答间隙，隐性位可以被显性位重写

错误处理 (1/2)



□CAN控制器三种状态



错误计数器

- 1、在接收过程中发现一个错误，接收错误计数器加1

例外：在错误帧Flag或过载帧Flag发送过程中的位错误不计数

- 2、当接收节点发现错误帧Flag之后的第一个位为显性位，接收错误计数器加8

- 3、当一个发送节点发送一个错误帧，发送错误计数器加8

例外1：如果发送节点为被动错误状态，则当起没有检测到显性应答或发送错误帧时没有检测到显性位

例外2：如果发送节点发送错误帧是由于在仲裁过程中发生了位填充错误（这个填充位在RTR位之前，应该为隐性。这个填充位的确是被当作隐性发送出去的，但是通过对总线的检测发现是显性）

- 4、如果发送节点在发送主动错误Flag或过载Flag的过程中发现一个位错误，发送错误计数器加8

- 5、如果接收节点在接收主动错误Flag或过载Flag的过程中发现一个位错误，接收错误计数器加8

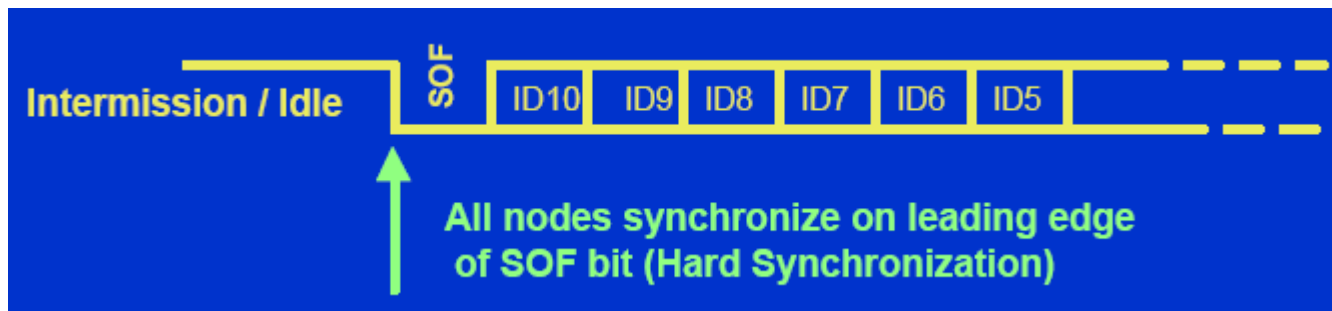
- 6、任何节点在发送主动错误Flag，被动错误Flag或过载Flag之后，最多能够容忍7个连续的显性位。如果出现了第8个显性位，则发送节点的发送错误计数器加8，接收节点的接收错误计数器加8：

- 7、当成功发送一个报文，发送错误计数器减1

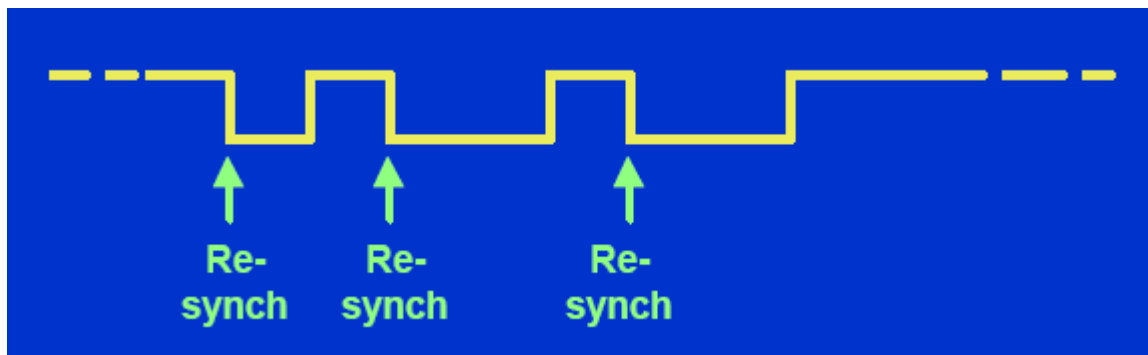
- 8、当成功接收一个报文，接收错误计数器减1（如果接收错误计数器在1到127之间）。如果接收错误计数器大于127，则接收错误计数器被置为119到127之间的一个值。

- 9、当节点进入Bus off状态时，如果总线上出现128个连续的11位隐性位，发送错误计数器和接收错误计数器被清零。

□ 硬同步：新闻联播！

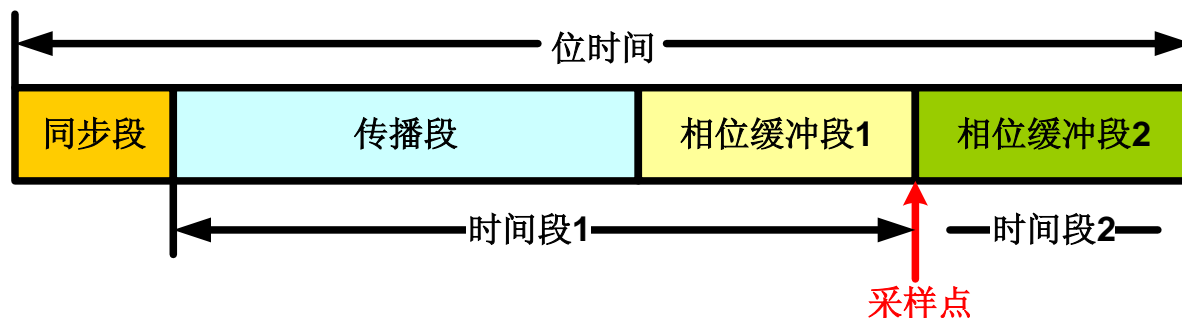


□ 重同步：一秒钟有多长？



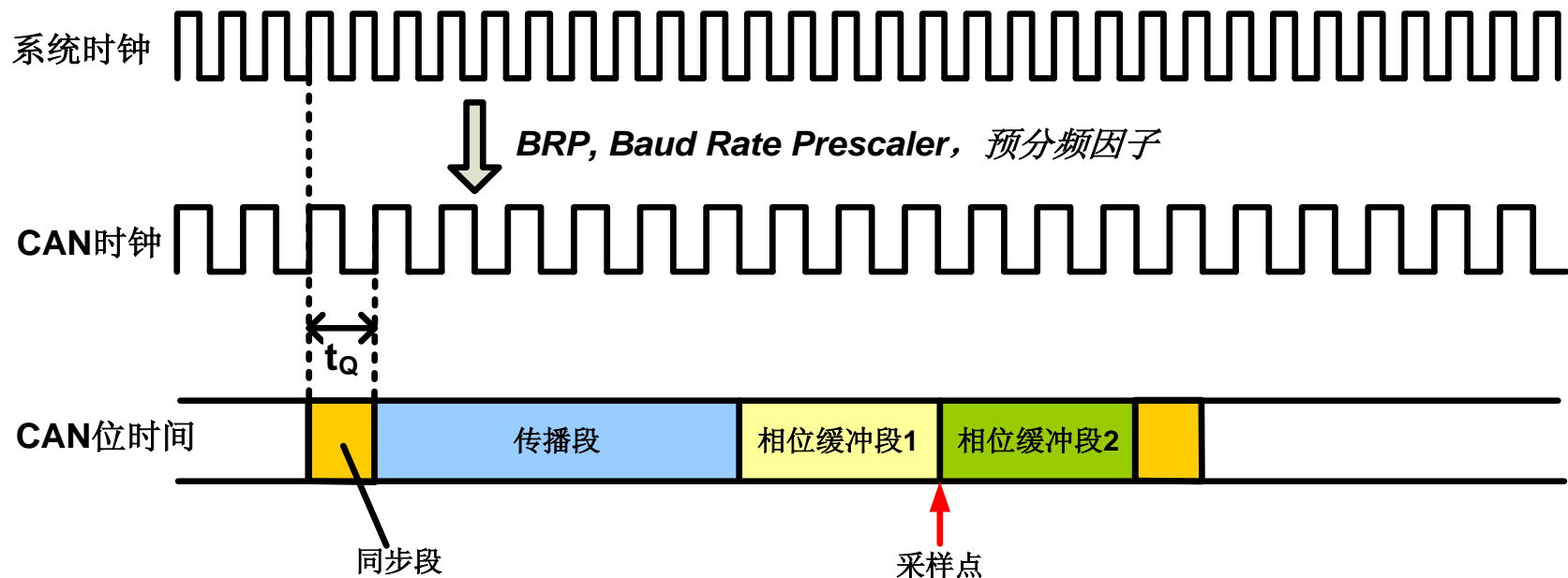
位定时的组成

- 一个位时间包含4个时间段，8-25个时间份Time Quantum(TQ)
- 为了编程方便，许多CAN模块将传播段和相位缓冲段1合并，称为时间段1；将相位缓冲段2称为时间段2



时间份额与波特率

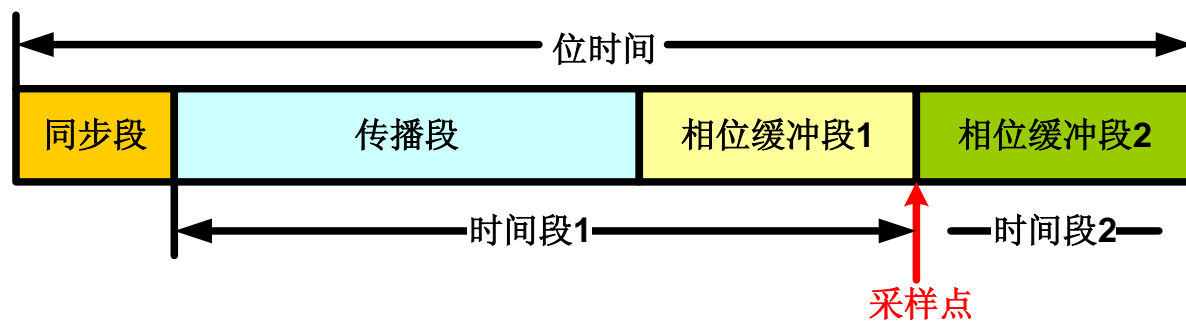
□ 时间份额来源于对系统时钟可编程的分频



□ 波特率=1/位时间=1/(n*t_q)

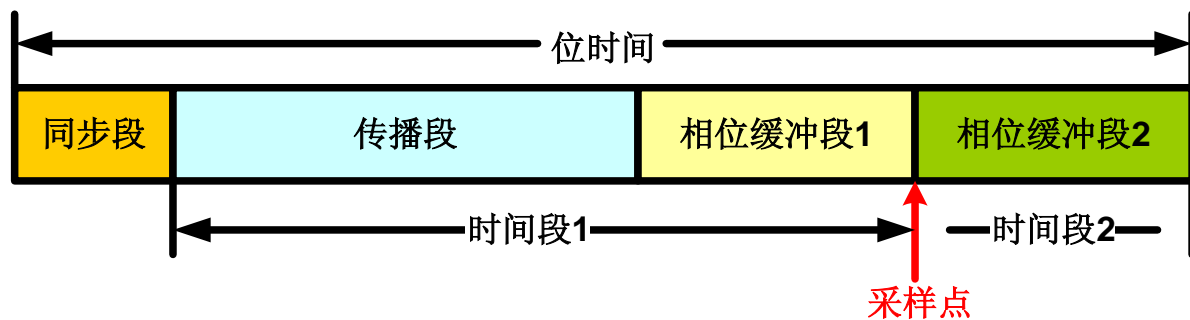
同步段

- 一个位的输出从同步段开始
- 用于同步各个节点，跳变沿产生在此段内
- 固定长度，1个TQ



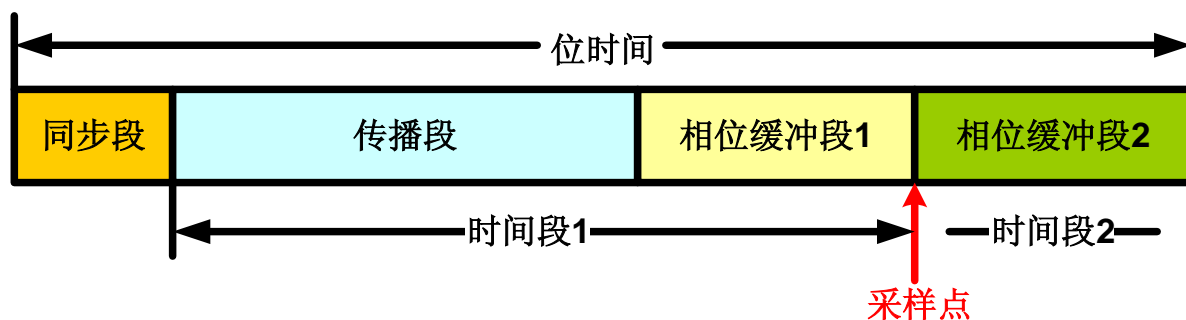
传播段

- 传播段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个时间份额或更长）



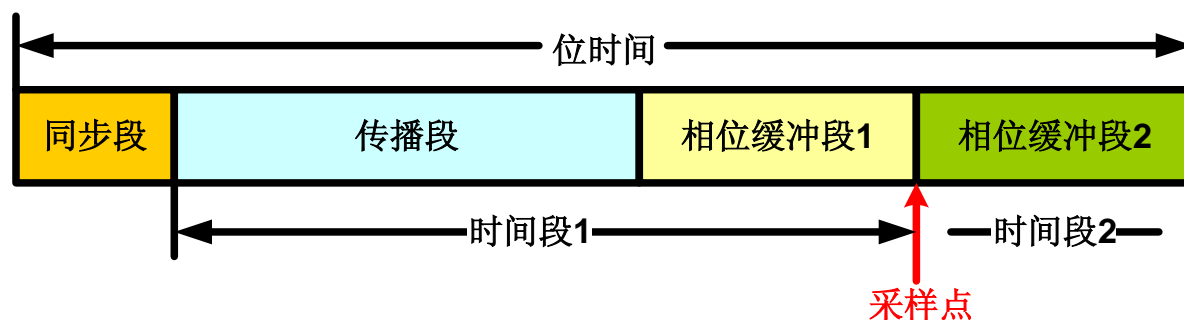
相位缓冲段1

- 允许通过重同步对相位缓冲段1加长
- 相位缓冲段1末端进行总线状态的采样
- 长度可编程（1…8个时间份额或更长）



相位缓冲段2

- 允许通过重同步对相位缓冲段2缩短
- 长度可编程（1...8个时间份额或更长）



□ SJW (Synchronization Jump Width)

- 相位缓冲段1可以被延长的时间份额数量
- 相位缓冲段2可以被缩短的时间份额数量

□ 同步跳转宽度大小

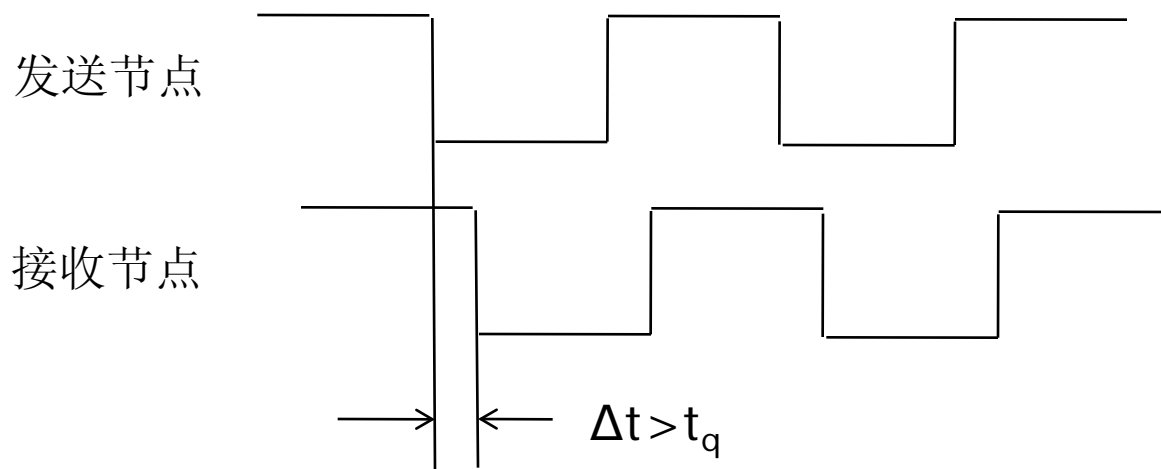
- 最短为1个时间份额，最长为4个时间份额
- 小于相位缓冲段2

同步条件

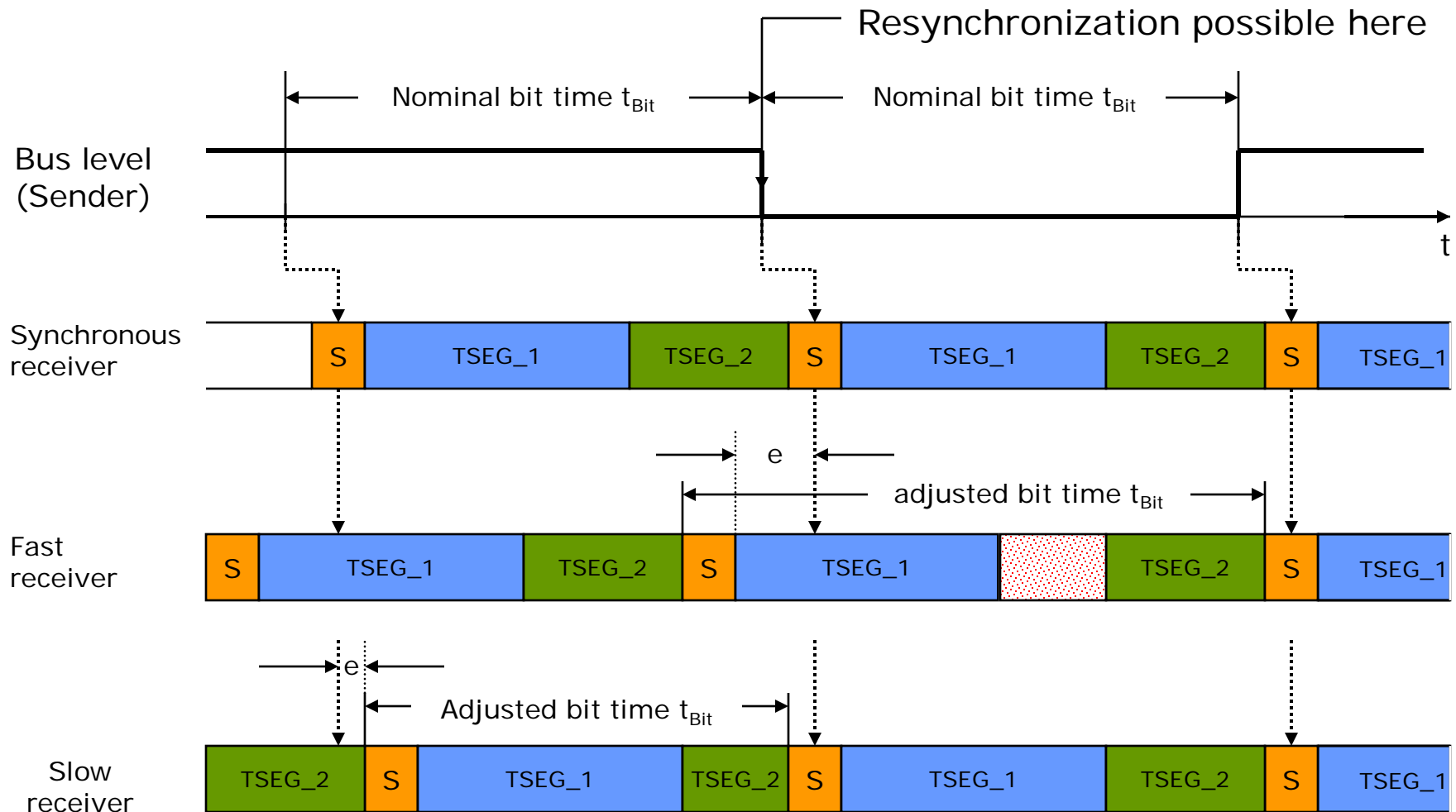
□ 从隐性到显性的跳变沿

□ 同步触发条件

□ 误差大于1个TQ

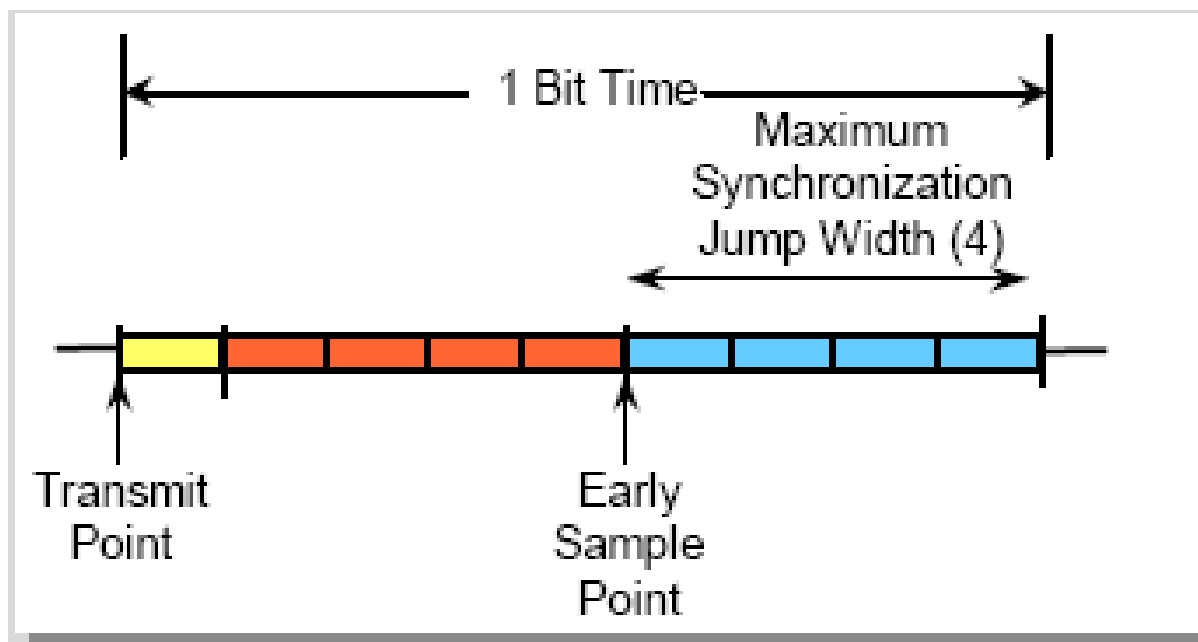


同步过程



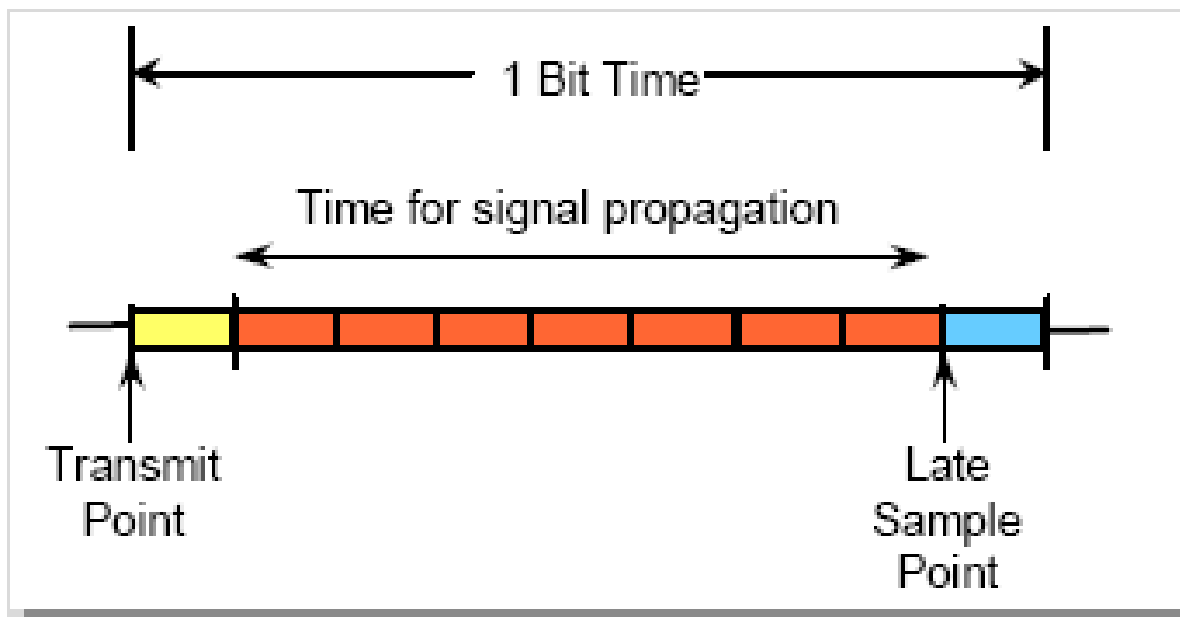
为什么要调整采样点的位置？

- 更大的同步跳转宽度
- 对于晶振的要求较低，可以使用价格低廉的晶振

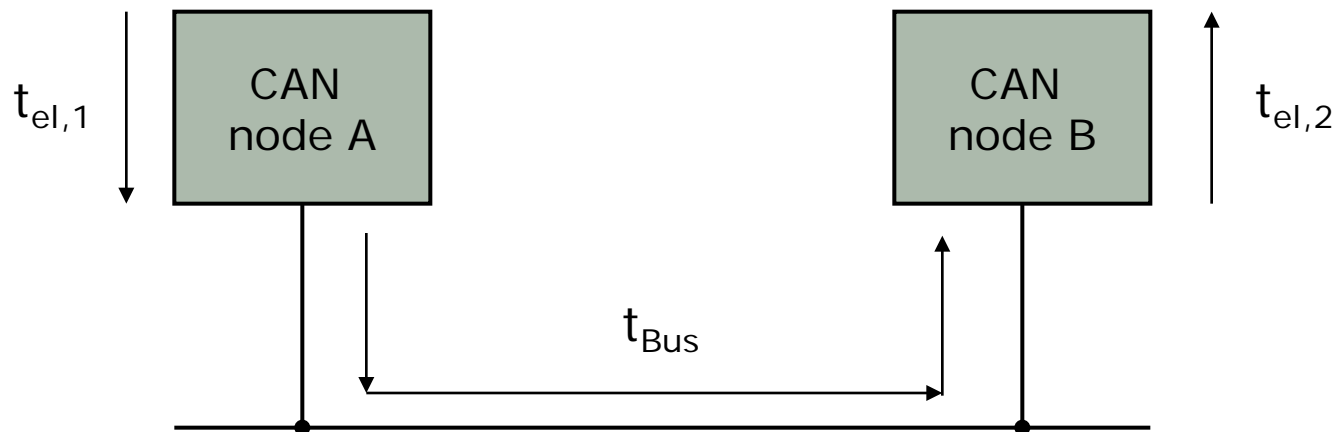


为什么要调整采样点的位置？

- 更长的传播补偿延时
- 适于处理更长的总线或不合理的总线拓扑结构



如何确定时间段1?



$$t_{del} = t_{el,1} + t_{Bus} + t_{el,2} = t_{el} + t_{Bus} \rightarrow$$

$$t_{TSEG_1} \geq 2 \cdot (t_{EI} + t_{Bus})$$

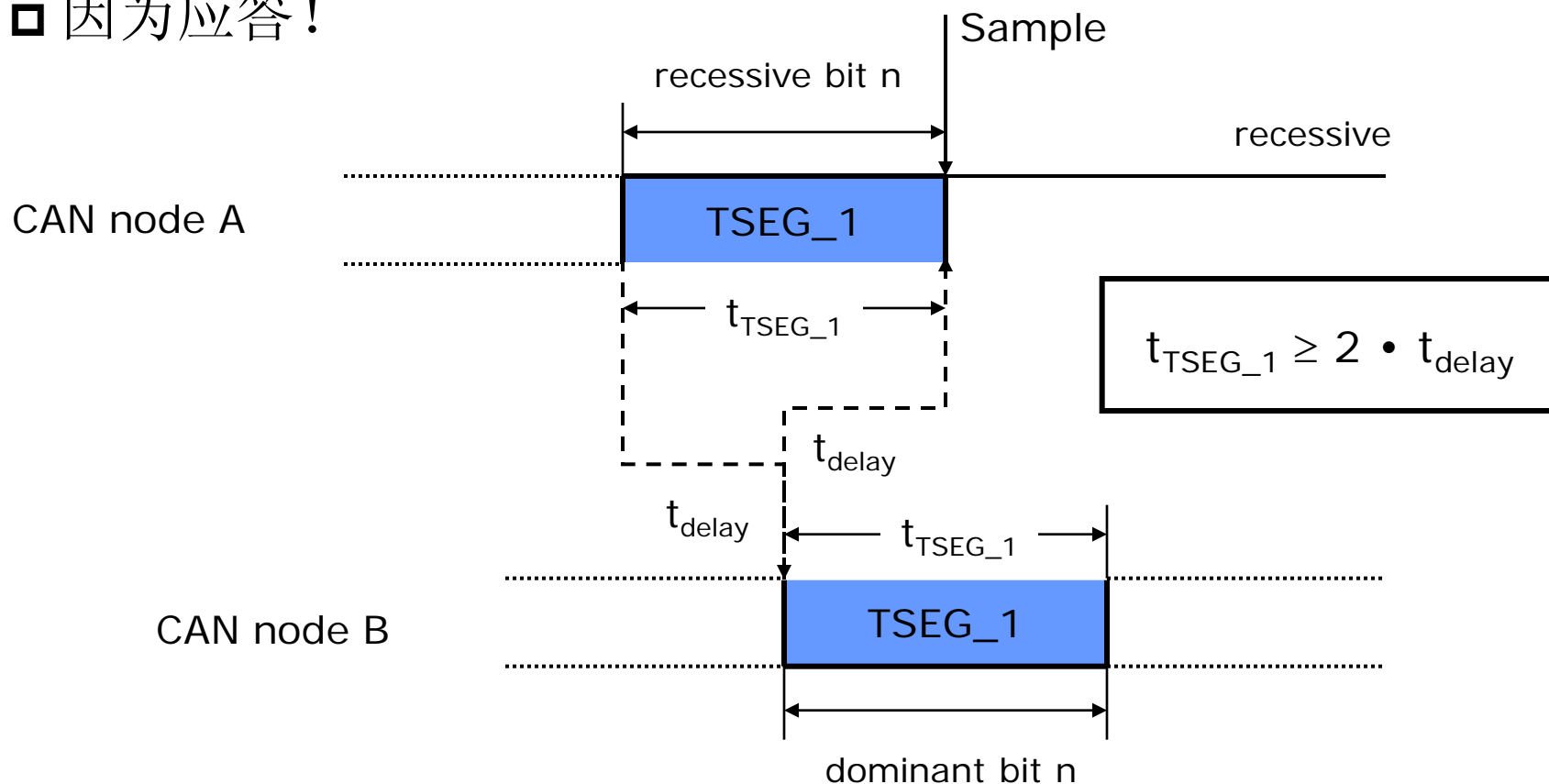
$$t_{el,2} = t_{CAN_Controller_2} + t_{CAN_Transceiver_2}$$

$$t_{el,1} = t_{CAN_Controller_1} + t_{CAN_Transceiver_1}$$

如何确定时间段1?

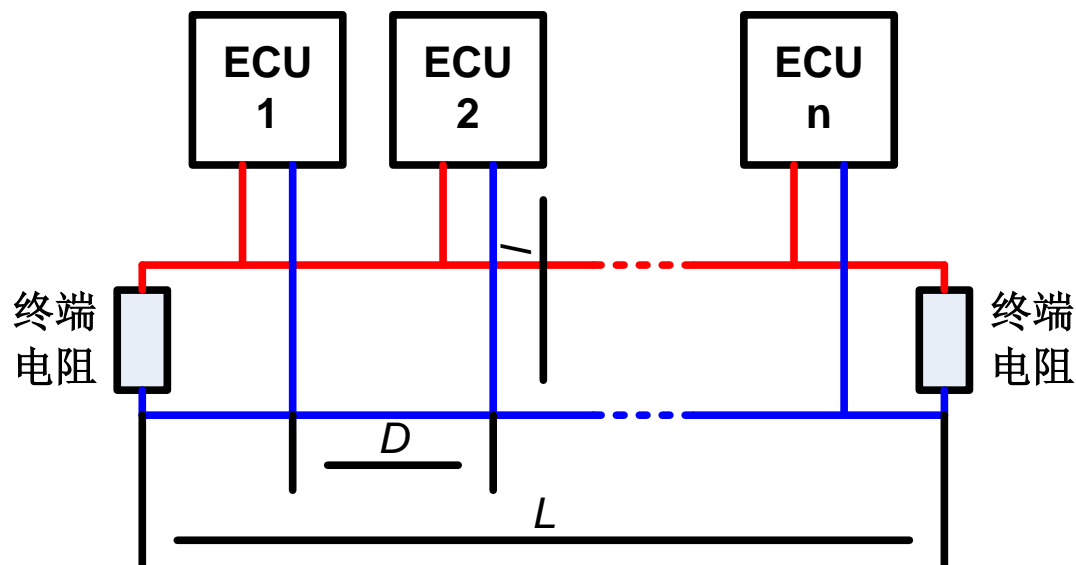
□ 为什么要乘2?

□ 因为应答!



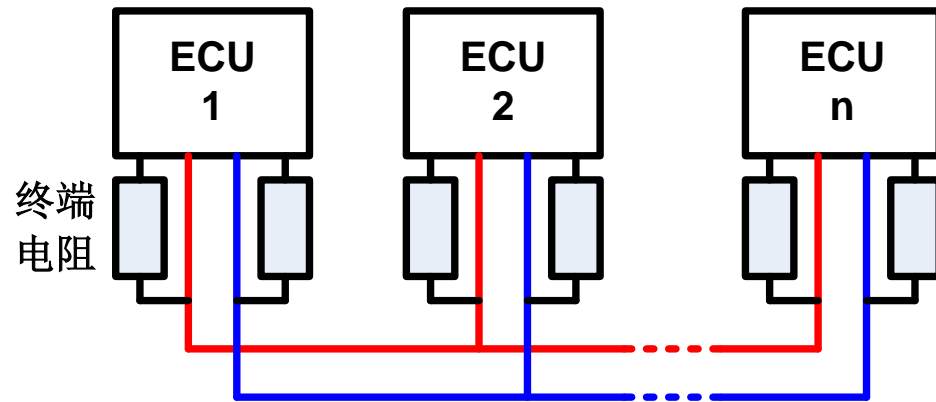
- CAN总线概述
- 数据链路层
- 物理层
 - 网络拓扑
 - 总线电平
 - CAN控制器

高速CAN拓扑结构



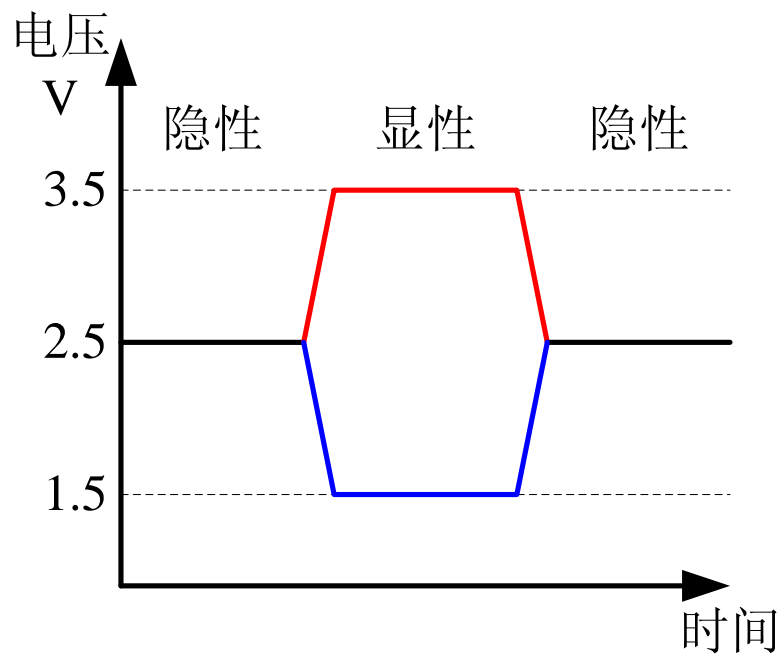
参数	符号	单位	数值		
			最小	名义	最大
总线长度	L	m	0		40
支线长度	l	m	0		0.3
节点距离	D	m	0.1		40

□ CAN线总长度小于40米



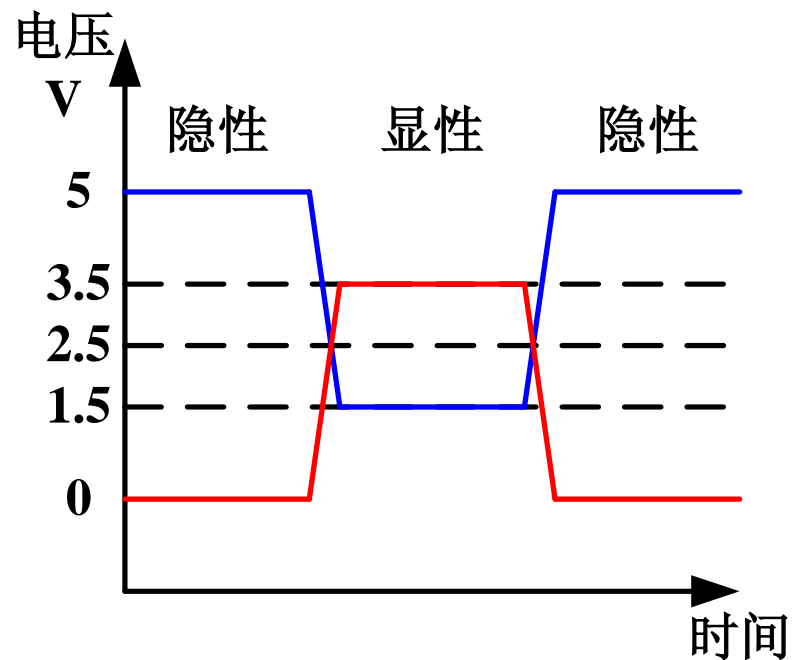
总线电平

高速CAN



PCA82C251

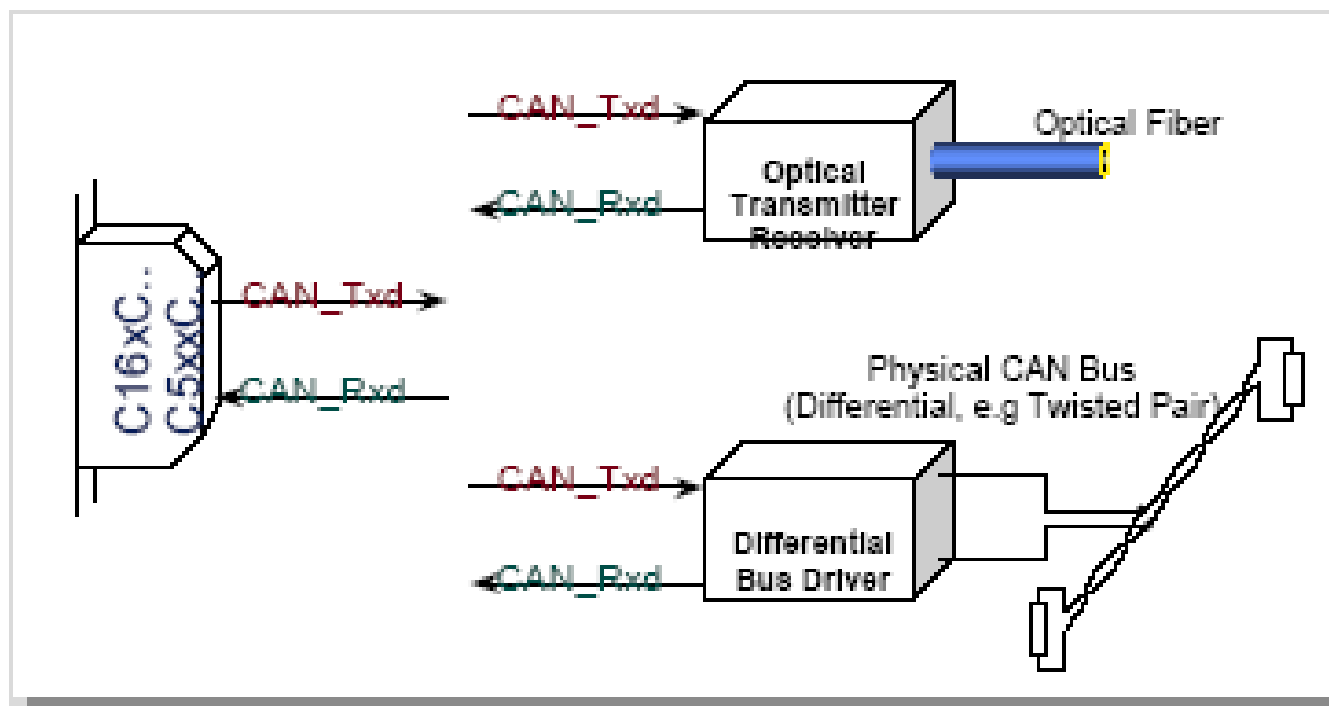
低速CAN



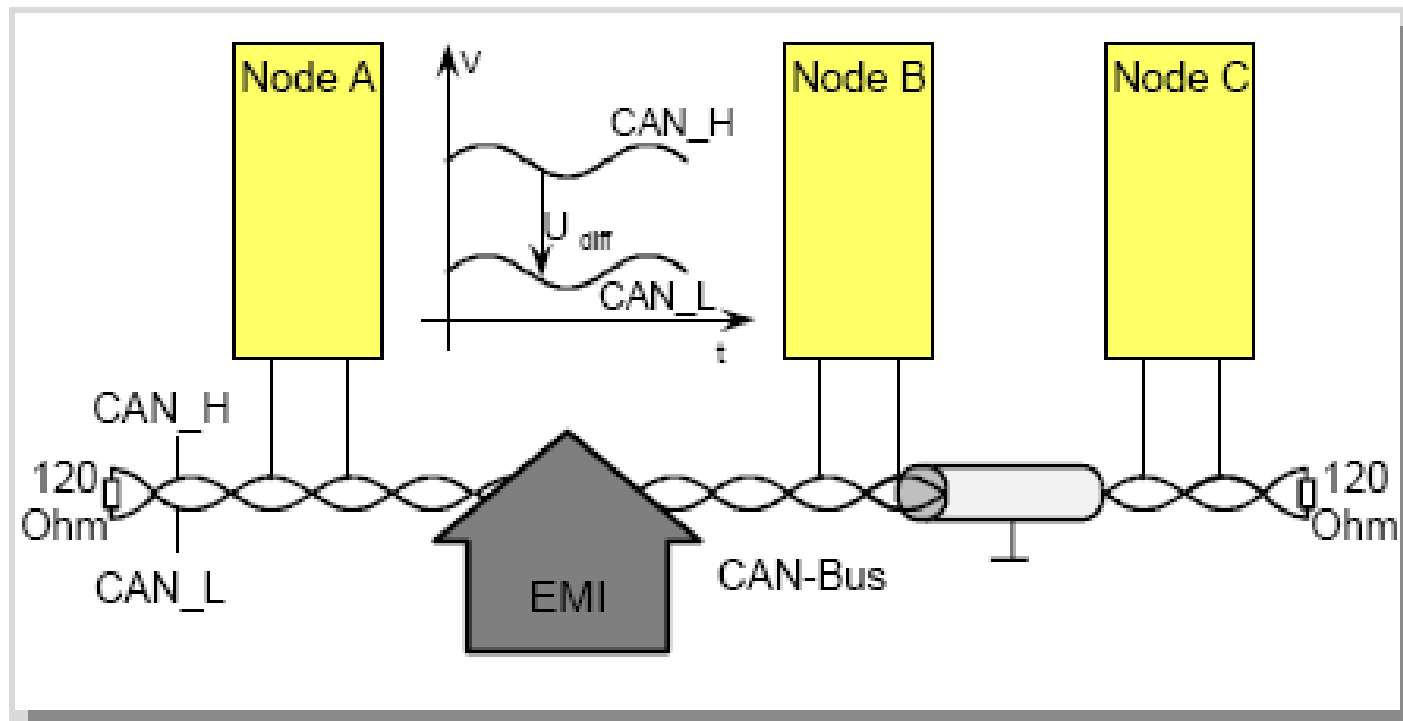
TJA1054

□ 双绞线

□ 光纤

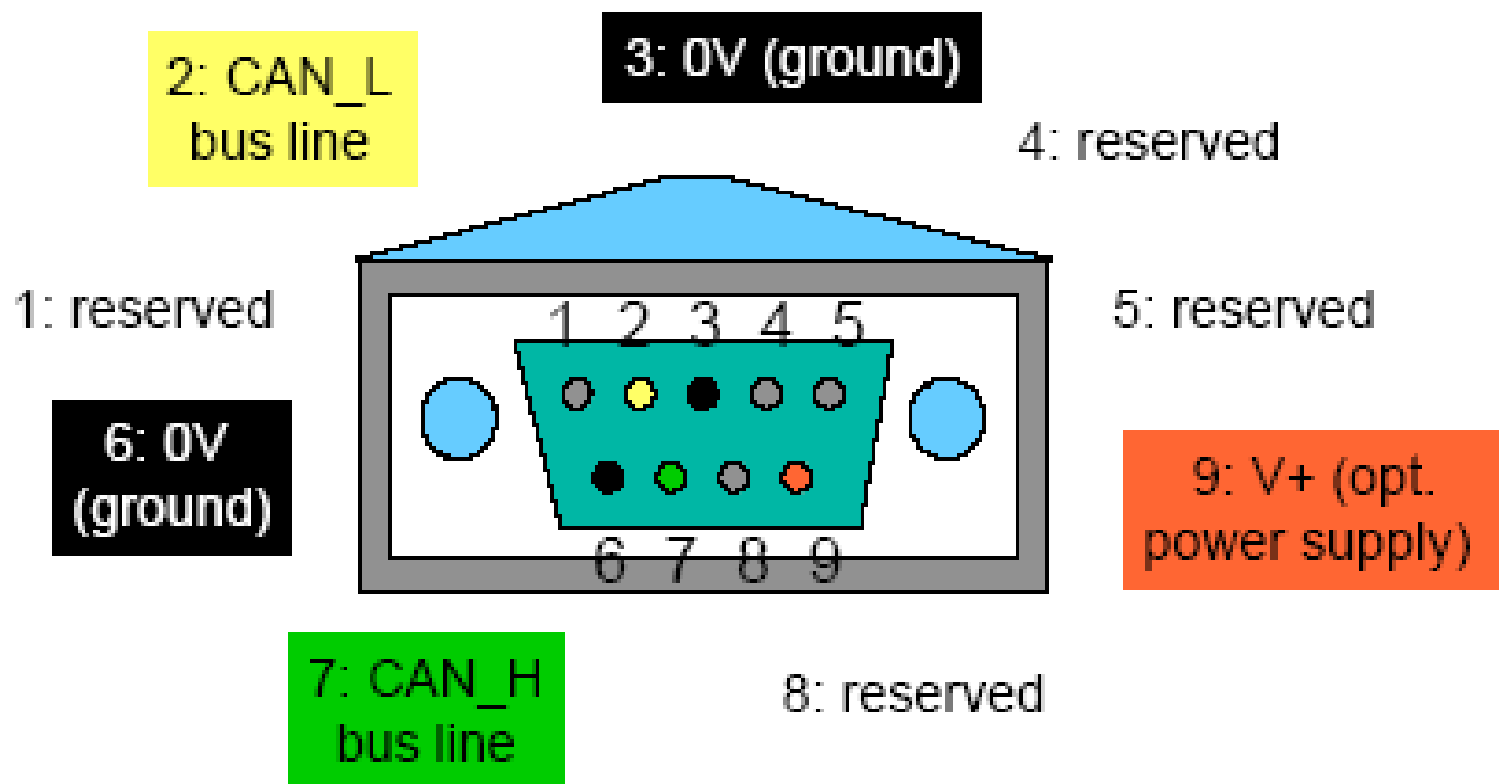


电磁干扰

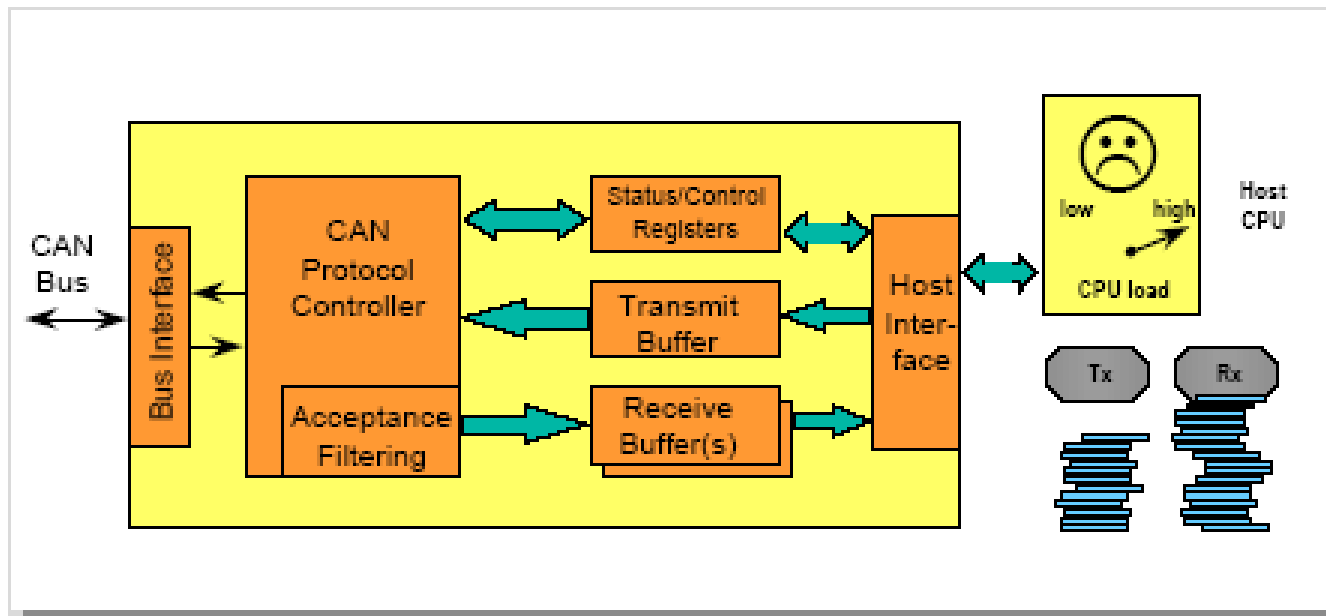


CAN总线连接器

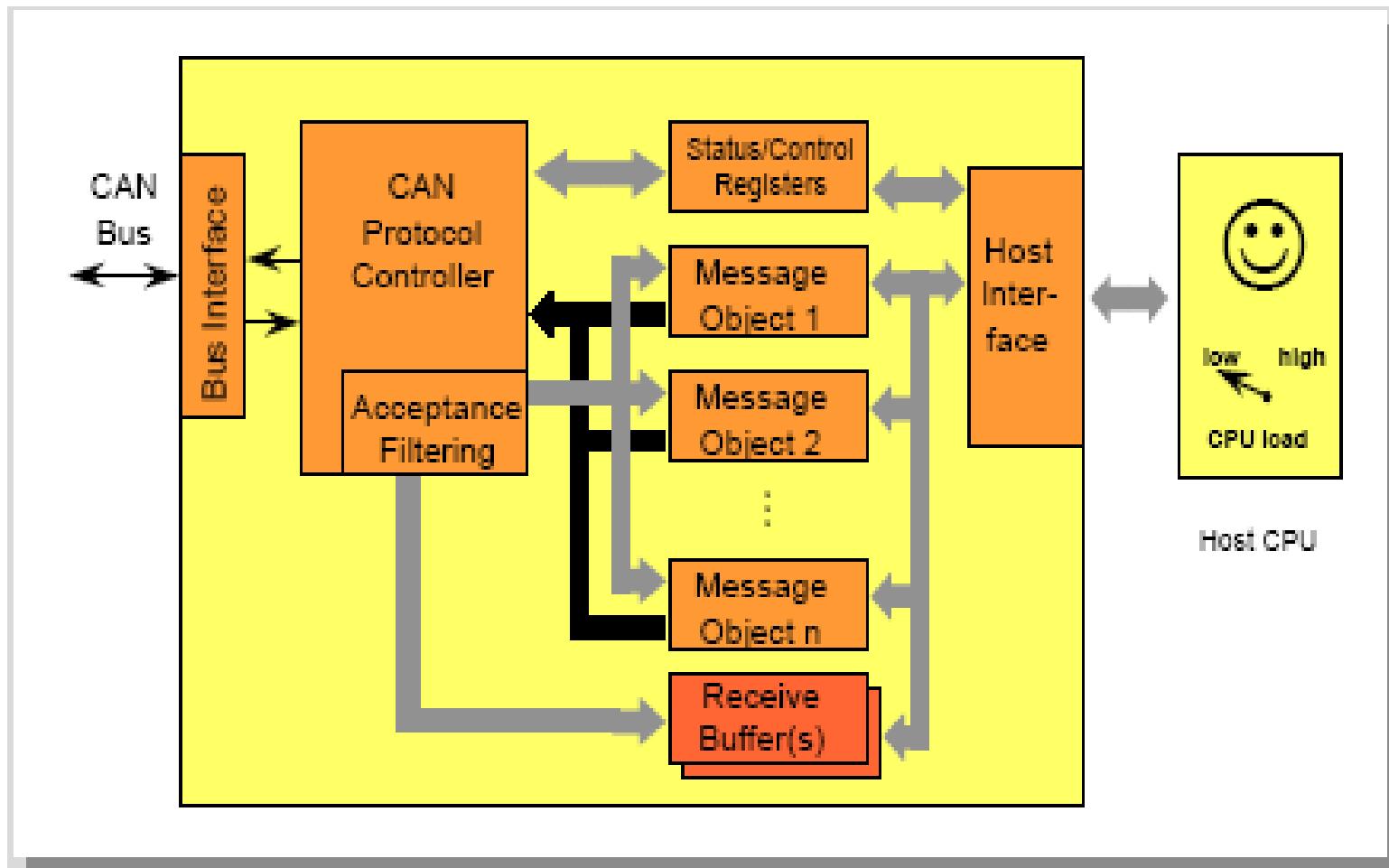
□ CiA-DS 102-1



Basic CAN控制器



Full CAN控制器



谢谢！

