

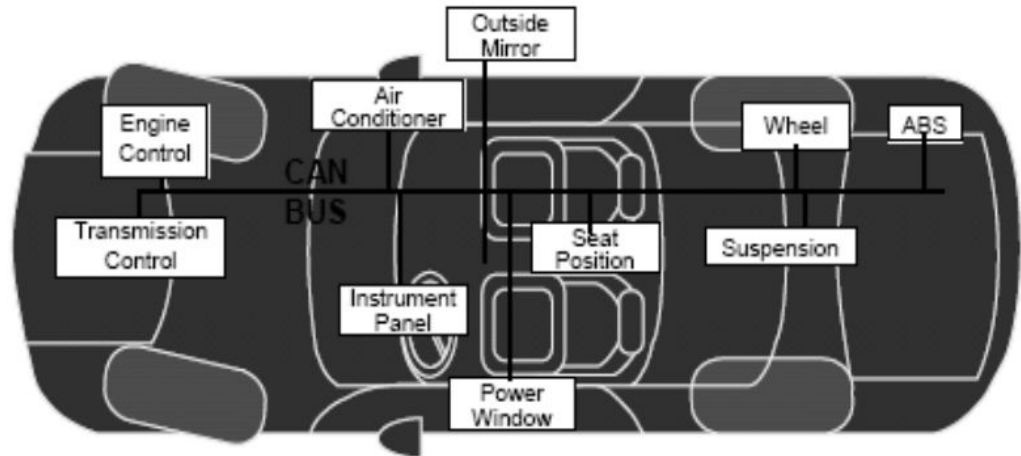
# CAN (Lý thuyết)

Phan Minh Thông

# 1. CAN

Controller Area Network (CAN) là giao thức giao tiếp **nối tiếp** hỗ trợ mạnh cho những hệ thống điều khiển **thời gian thực phân bố** (distributed realtime control system).

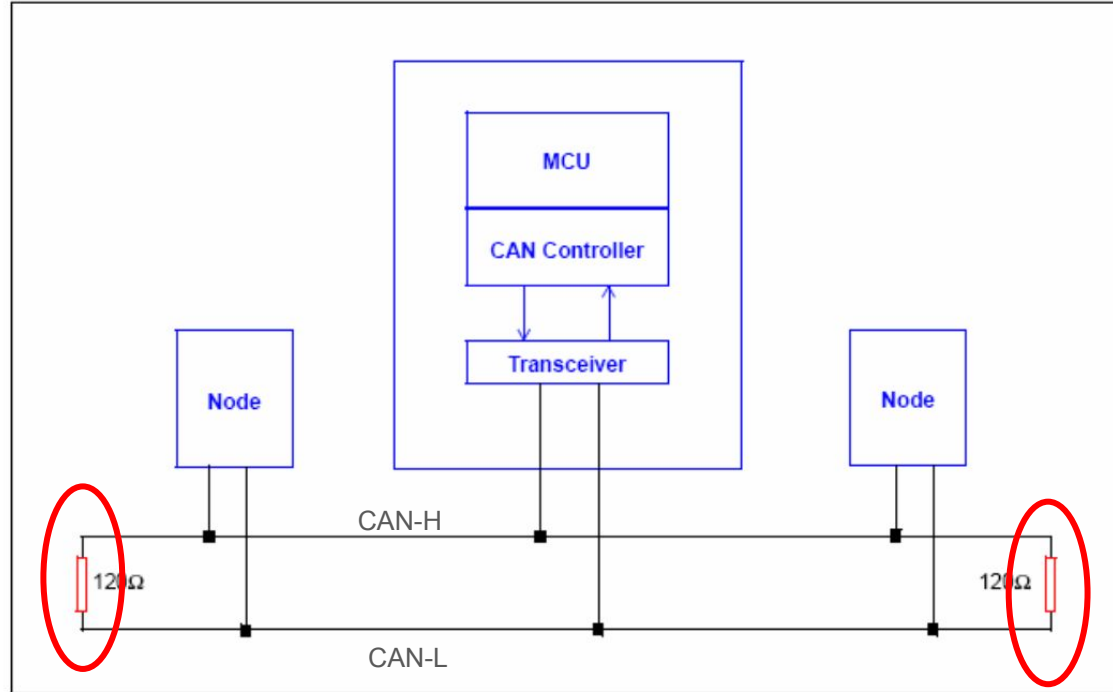
CAN đặc biệt được ứng dụng nhiều trong ngành công nghiệp Ô tô.



## 2. Mạng CAN

CAN có đường dây dẫn đơn giản gồm 2 dây CAN\_H và CAN\_L, tạo thành 1 Bus, các thiết bị được nối chung trên 2 dây này và gọi là 1 **node** trong mạng.

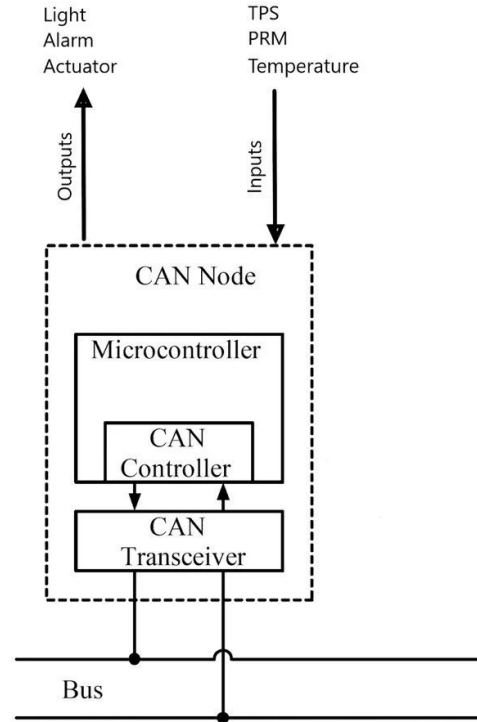
Sự truyền dữ liệu thực hiện nhờ tính toán vi sai trên cặp dây truyền tín hiệu, có nghĩa là chúng ta đo sự chênh lệch điện áp giữa CAN\_H và CAN\_L.



# CAN Node.

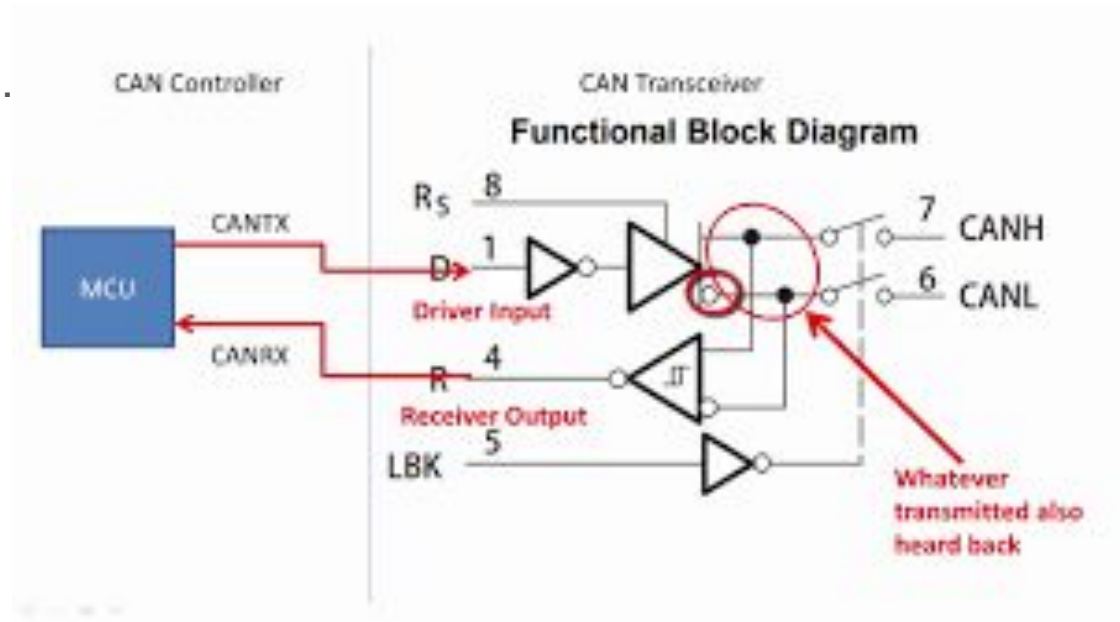
Mạng CAN được tạo thành bởi một nhóm các nodes. 1 node gồm:

- Một thiết bị hỗ trợ xử lý điện áp trên bus- CAN Transceiver.
- Một MCU hỗ trợ Can Controller, chính là giao thức CAN. MCU ngoài việc nhận và xử lý data còn thực hiện chức năng của node.



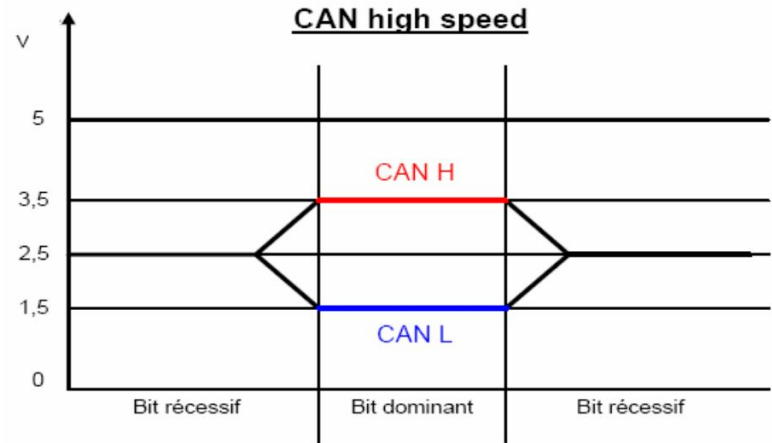
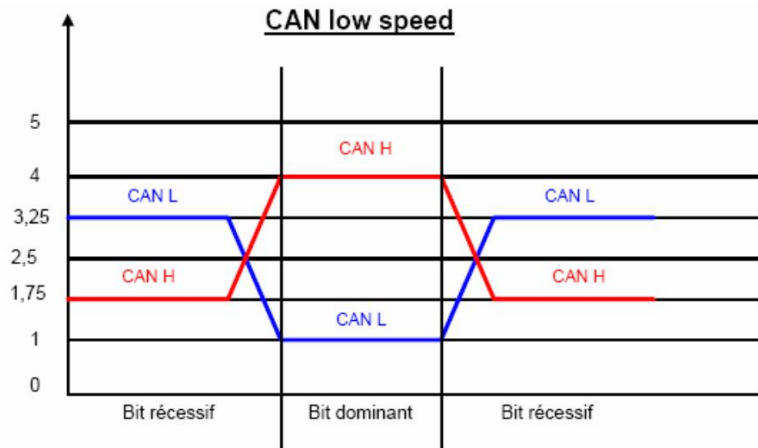
# Transceiver

Bộ transceiver hỗ trợ cho 1 node truyền dữ liệu lên Bus, đồng thời sẽ nhận lại tín hiệu vừa truyền đi.



# Trạng thái “dominant” và “recessive”

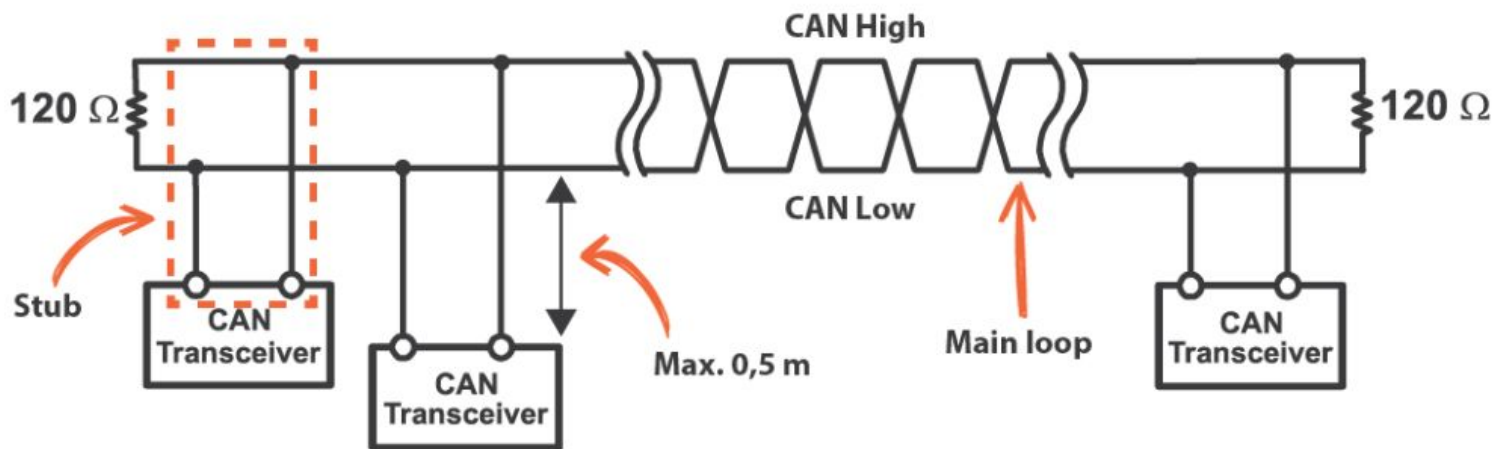
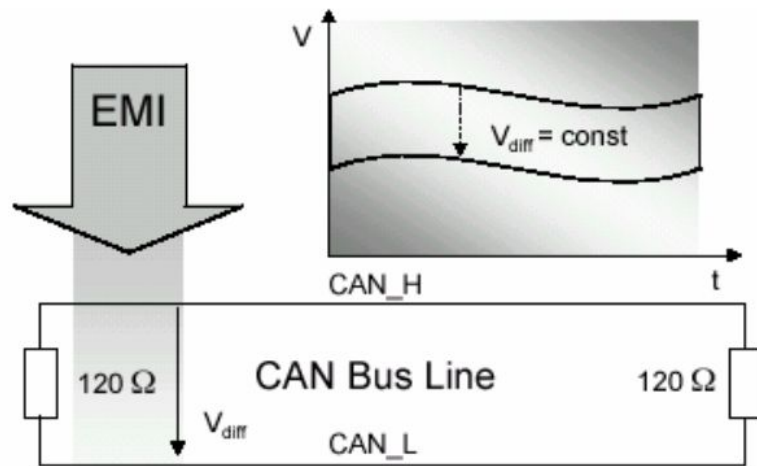
Bus CAN định nghĩa hai trạng thái điện áp là “dominant” và “recessive”, tương ứng với hai trạng thái bit là 0 và 1. Hai trạng thái này sẽ được xử lý bởi Transceiver của Node. Có 2 loại CAN tương ứng với các giá trị điện áp khác nhau.



# Tính chất

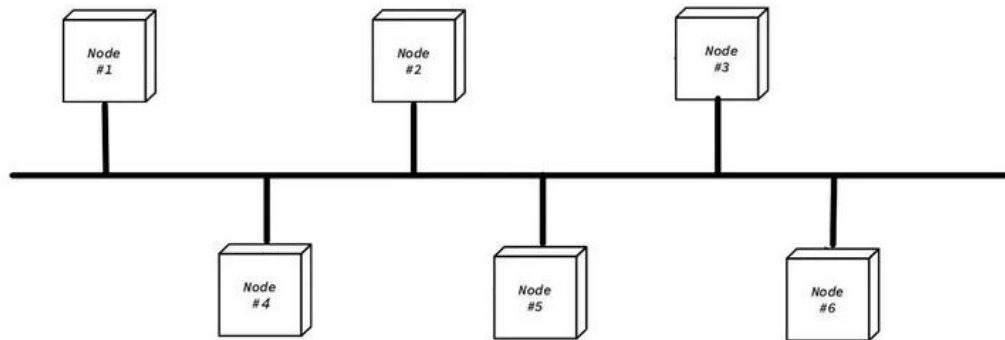
Vì tính chất vi sai trên đường truyền tín hiệu của bus CAN, tín hiệu nhiễu sẽ ảnh hưởng đến giá trị điện áp.

CAN bus thường được xoắn 2 dây vào nhau để triệt tiêu nhiễu.



# Nguyên tắc hoạt động

- Thông điệp dữ liệu được truyền từ bất kỳ nút nào trên bus CAN không chứa địa chỉ của nút truyền hoặc của bất kỳ nút nhận dự kiến nào.
- Thay vào đó, nội dung của thông điệp được gắn nhãn bởi một số nhận dạng (ID) là duy nhất trên toàn mạng.
- Tất cả các nút khác trên mạng đều nhận được thông điệp và mỗi nút thực hiện kiểm tra sự chấp nhận trên mã ID để xác định xem thông điệp có liên quan đến nút đó hay không. Nếu thông điệp có liên quan, nó sẽ được xử lý, nếu không thì nó bị bỏ qua

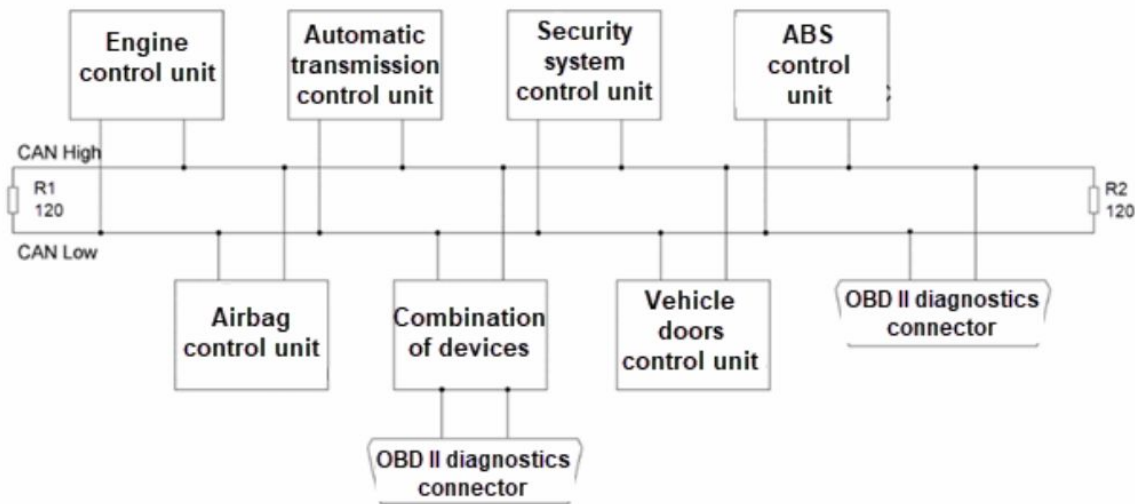




# Tranh chấp trên BUS

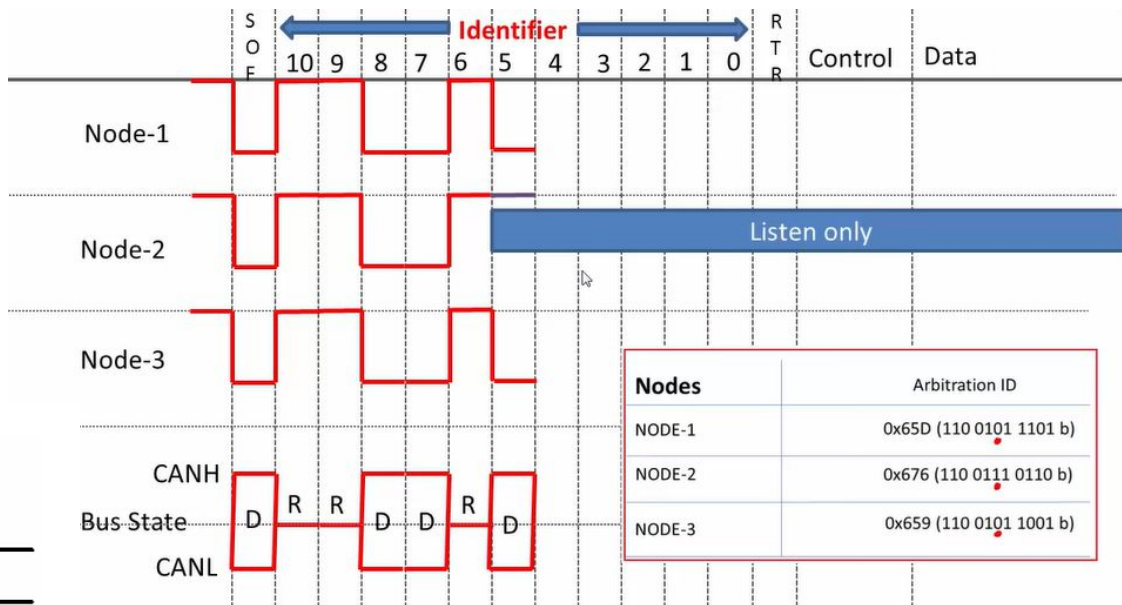
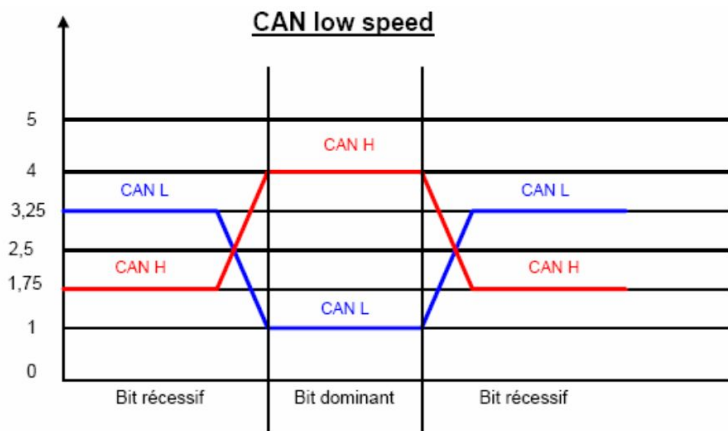
Giao thức CAN cho phép các nút khác nhau gửi dữ liệu cùng lúc. Một mạng Can có thể gồm nhiều node với lượng dữ liệu truyền lên Bus rất lớn. Chỉ 1 node được phép truyền tại 1 thời điểm.

Data Frame và Remote Frame làm việc theo cơ chế phân xử quyền ưu tiên của tín hiệu vì thế cấu trúc của chúng có vùng phân xử quyền ưu tiên - ID của Frame.



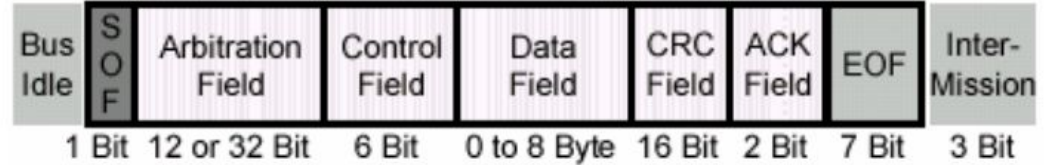
# Tranh chấp trên BUS

Khi phân xử, bit ID có giá trị 0 sẽ được ưu tiên hơn.  
Node sẽ dựa vào giá trị điện áp trên bus để quyết định.



# CAN Frame

Dữ liệu CAN được truyền dưới dạng các Frame (khung). Một khung có dạng chung như sau:



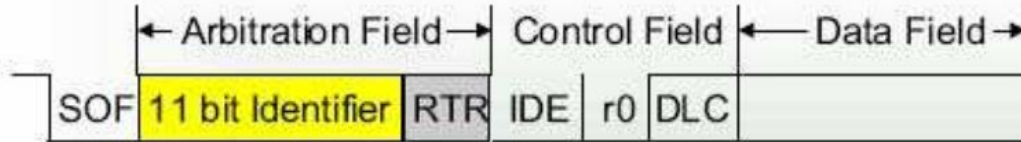
Chuẩn CAN định nghĩa bốn loại Frame:

- **Data frame** dùng khi node muốn truyền dữ liệu tới các node khác
- **Remote frame** dùng để yêu cầu truyền data frame.
- **Error frame và overload frame** dùng trong việc xử lý lỗi.

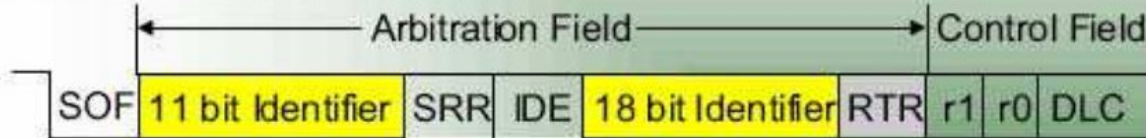
# Data frame:

Dùng để truyền dữ liệu. Có hai dạng: standard frame và extended frame

## Standard Frame Format

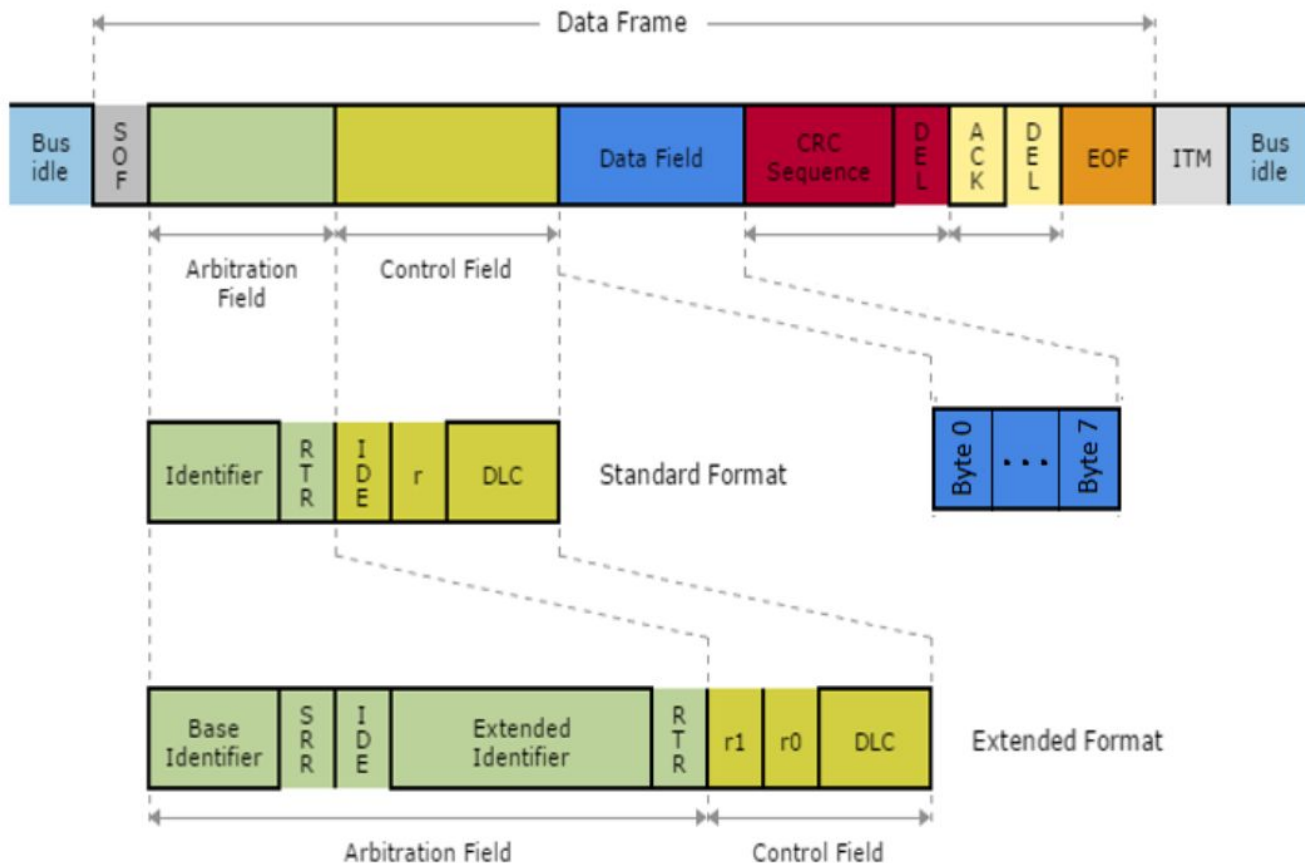


## Extended Frame Format



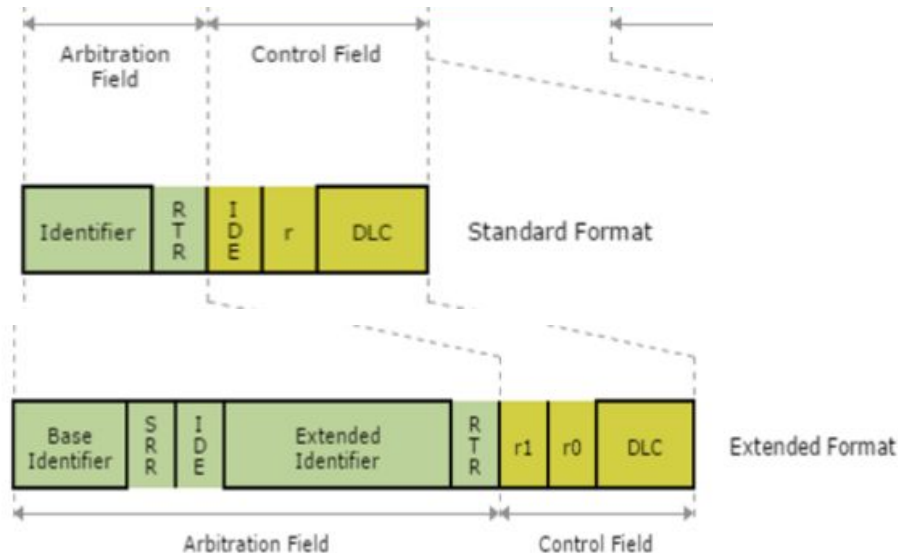
**Trade-off:** longer bus latency time (20 bit-times)  
longer frames (20 bit-times plus stuff-bits)  
reduced CRC performance

Cấu trúc của 1 data Frame có thể minh họa chi tiết như sau:



# Data Frame CAN bao gồm bảy trường bit khác nhau:

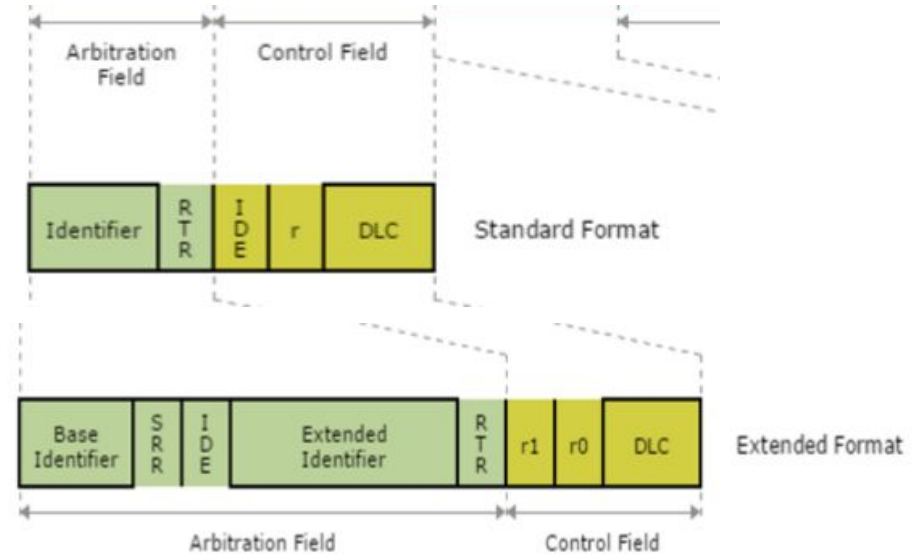
1. **Trường bắt đầu(Start Of Frame Field – SOF):** Độ dài 1 bit đầu của frame, giá trị Dominant báo hiệu bắt đầu của 1 frame.
2. **Trường xác định quyền ưu tiên (Arbitration Field):** Định dạng vùng xác định quyền ưu tiên
  - **Định dạng chuẩn:** vùng xác định quyền ưu tiên có độ dài 12 bit, bao gồm 11 bit ID và 1 bit RTR.
  - **Định dạng mở rộng:** trường xác định quyền ưu tiên có độ dài 32 bit, bao gồm có 29 bit ID, 1 bit SRR, 1 bit IDE và 1 bit RTR.



- **Bit RTR (Remote Transmission Request):** dùng để phân biệt khung là Data Frame hay Remote Frame (0- Data frame, 1- Remote frame).
- **Bit SRR (Substitute Remote Request):** chỉ có ở khung mở rộng, có giá trị là 1.
- **Bit IDE (Identifier Extension):** bit phân biệt giữa loại khung chuẩn và khung mở rộng:  
IDE = 0 - khung chuẩn, IDE = 1 - khung mở rộng. Bit này nằm ở trường xác định quyền ưu tiên với khung mở rộng và ở trường điều khiển với khung chuẩn.

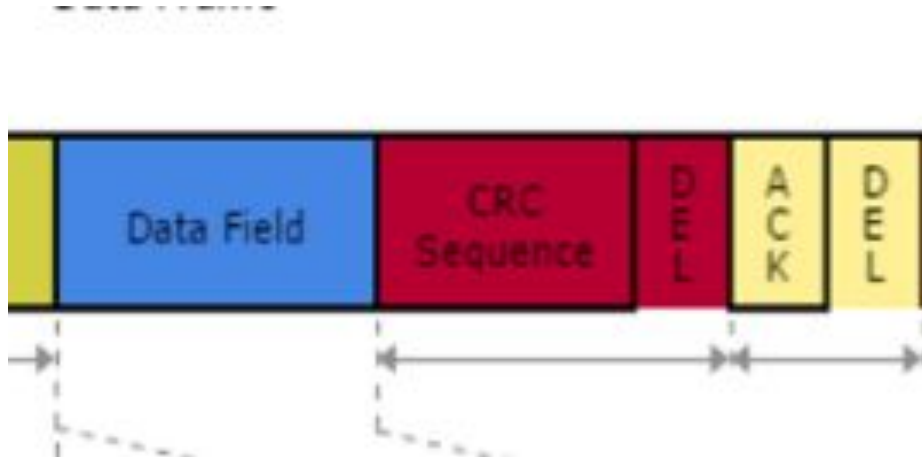
3. **Trường điều khiển (Control Field):** có định dạng khác nhau ở trường này:  
Khung chuẩn gồm IDE, r0 và DLC (Data Length Code).  
Khung mở rộng gồm r1, r0 và DLC.

- **Bit r0, r1** (hai bit dự trữ): phải được truyền là Recessive Bit bởi bộ truyền nhưng bộ nhận không quan tâm đến giá trị 2 bit này.
- **DLC (Data Length Code):** Có độ dài 4 bit quy định số byte của trường dữ liệu của Data Frame, Chỉ được mang giá trị từ 0 đến 8.

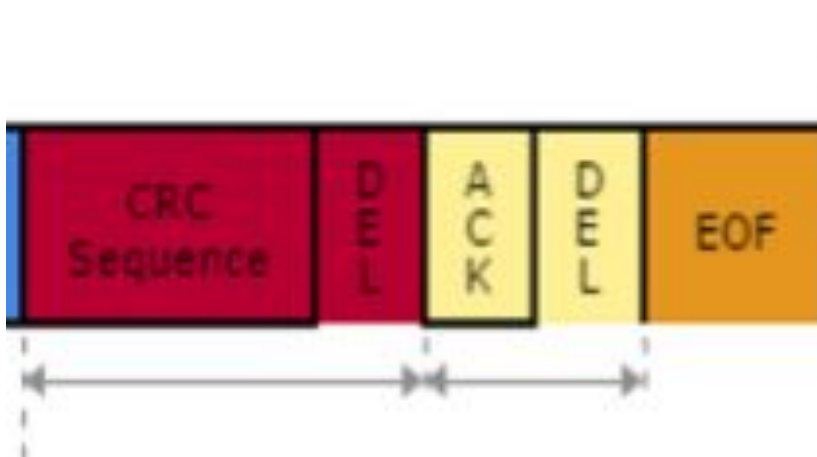




- 
- 4. **Trường dữ liệu (Data Field):** chứa data, có độ dài từ 0 đến 8 byte tùy vào giá trị của DLC ở trường điều khiển.
- 5. **Trường kiểm tra (Cyclic Redundancy Check Field – CRC):** gồm 16 bit và được chia làm hai phần:
  - **CRC Sequence:** gồm 15 bit CRC tuần tự
  - **CRC Delimiter:** là một Recessive Bit làm nhiệm vụ phân cách trường CRC với trường phía sau.



6. **Trường xác nhận (Acknowledge Field – ACK):** có độ dài 2 bit và bao gồm hai phần:
- **ACK Slot:** có độ dài 1 bit, Node truyền dữ liệu sẽ truyền bit này là Recessive. Khi một hoặc nhiều Node nhận chính xác giá trị thông điệp (không có lỗi và đã so sánh CRC Sequence trùng khớp) thì nó sẽ báo lại cho bộ truyền bằng cách truyền Dominant Bit ngay vị trí ACK Slot (tương tự việc kéo SDA trong I2C).
  - **ACK Delimiter:** có độ dài 1 bit, nó luôn là một Recessive Bit  $\Rightarrow$  ACK Slot luôn được đặt giữa hai Recessive Bit là CRC Delimiter và ACK Delimiter.
7. **Trường kết thúc (End Of Frame Field – EOF):** thông báo kết thúc một Data Frame hay Remote Frame. Trường này gồm 7 Recessive Bit.



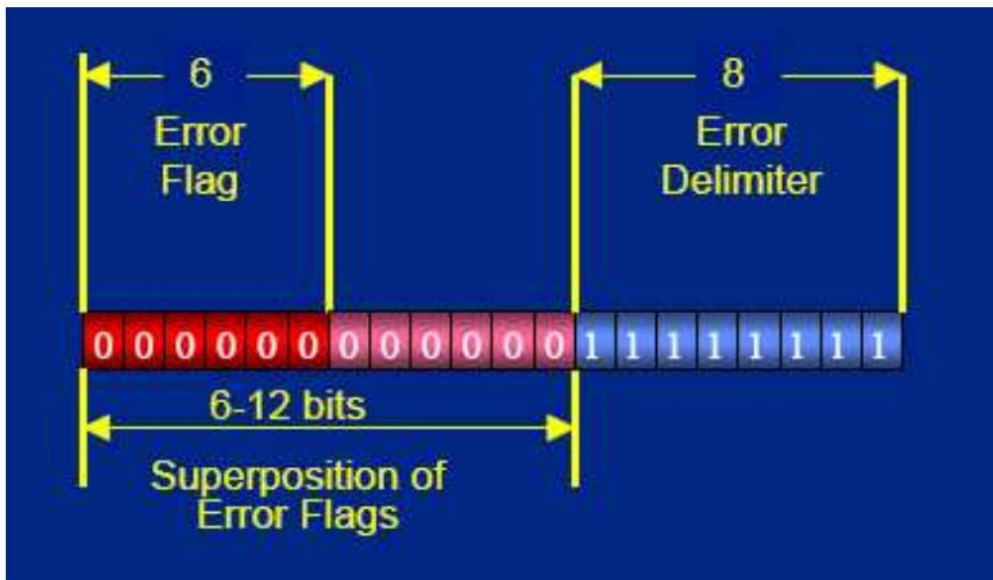
# Remote Frame

Cấu trúc 1 Remote Frame tương tự Data frame, tuy nhiên Remote Frame có Bit RTR =1 và không có trường data (Bit DLC =0).

## Error Frame

Frame lỗi bao gồm 2 phần:

- Cờ lỗi
- Phần delimiter



## Overload Frame

Dùng khi node bị tràn bộ đệm.