

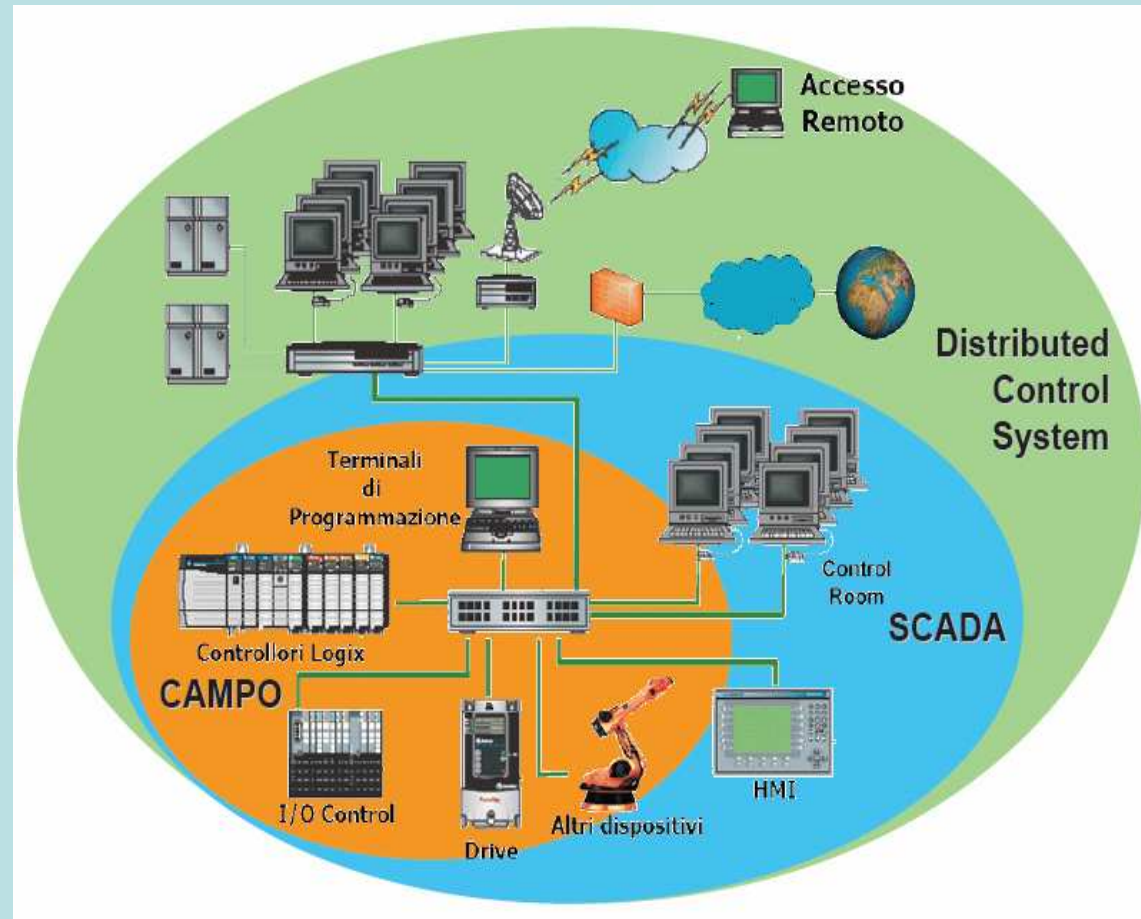
BUS DI CAMPO

Definizione

- “**Bus di campo**” è un’espressione generica che descrive una forma di comunicazione digitale dedicata ai sistemi a basso livello, quali sensori o attuatori, la quale si prevede sostituirà nell’industria la tecnologia analogica per collegamenti punto a punto basata su segnali a 4-20mA (PLC). L’architettura a bus consente una notevole riduzione dei cablaggi e dei costi. Il bus rappresenta il solo mezzo fisico utilizzato per trasportare i dati e quindi per interconnettere i vari dispositivi.

Scenario industriale

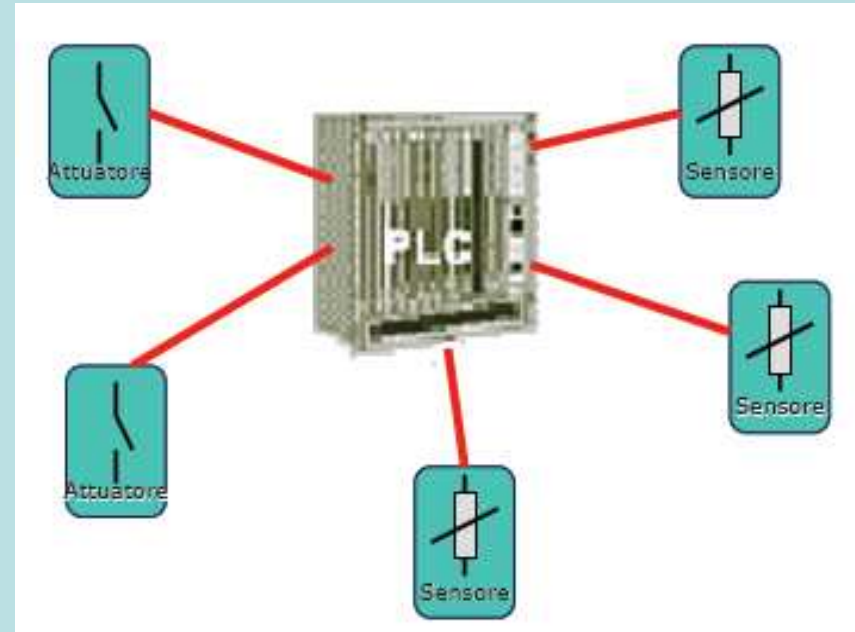
- Misura ed acquisizione delle grandezze di interesse
 - Sensori
 - Elettronica di condizionamento dei segnali
 - Conversione analogico/digitale
- Elaborazione (real-time) delle strategie di automazione
 - Sistemi embedded (dedicati)
 - Algoritmi di controllo
 - Software di controllo, segnalazione, diagnostica
- Attuazione dei comandi
 - Azionamenti



SCADA (Supervisory Control And Data Acquisition)

Architettura tradizionale

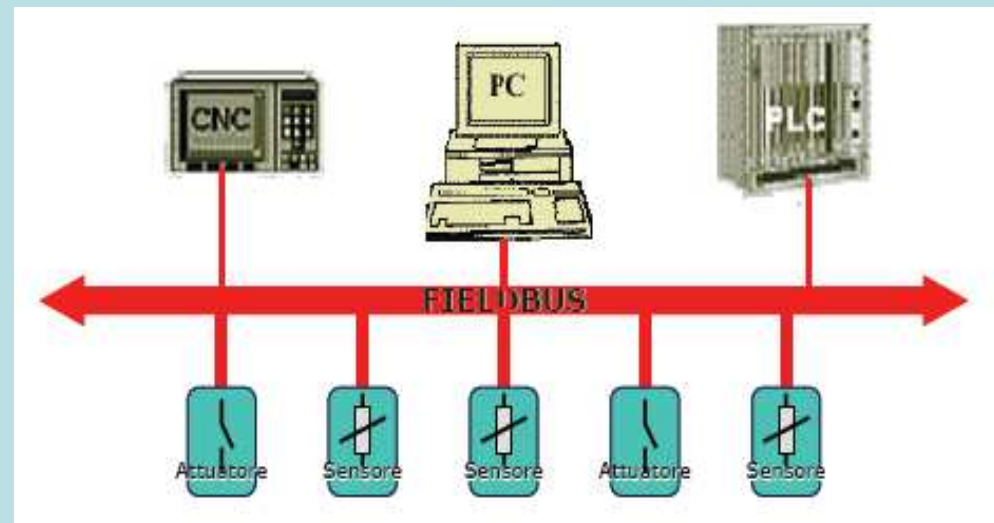
- **Caratteristiche**
 - Architettura centralizzata
 - Collegamento punto-punto
- **Svantaggi**
 - Elevato numero di collegamenti
 - Cablaggio costoso
 - Lavori di stesura e protezione fili molto costoso
- **Vantaggi**
 - Sistema affidabile e collaudato
 - Disponibilità di tutte le tipologie di strumentazione su mercato



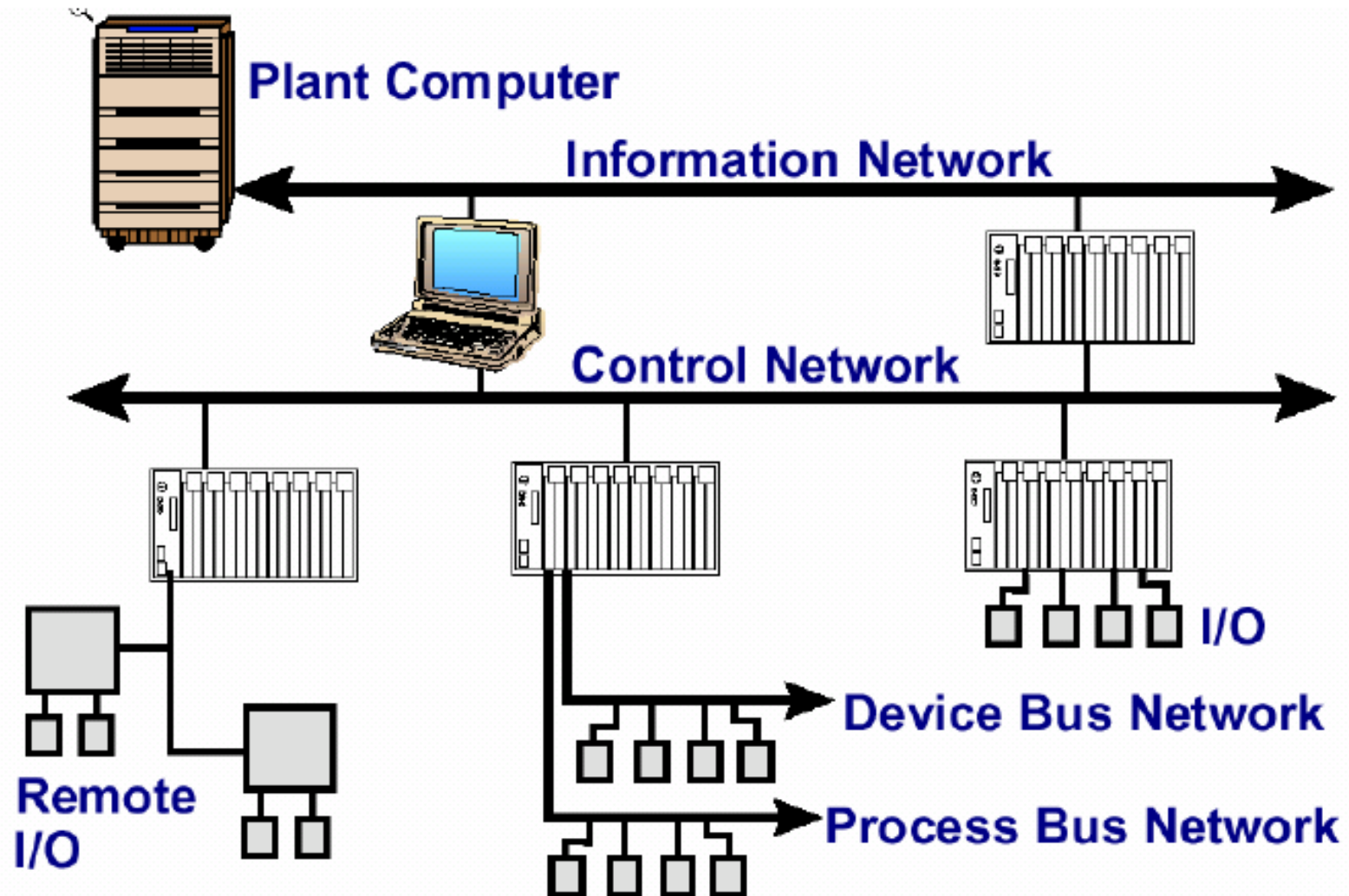
Il **PLC** (programmable logic controller) è un computer industriale specializzato nella gestione dei processi industriali. Il PLC esegue un programma ed elabora i segnali digitali ed analogici provenienti da sensori e diretti agli attuatori presenti in un impianto industriale.

Architettura a bus di campo

- **Caratteristiche**
 - Architettura a bus
 - Trasmissione digitale dei segnali
 - Strumenti intelligenti
- **Vantaggi**
 - Risparmio sui costi di cablaggio e installazione
 - Facilità di aggiunta o rimozione dei dispositivi
 - Riduzione degli errori d'installazione
 - Condivisione delle risorse
 - Tolleranza ai guasti
 - Flessibilità
- **Svantaggi**
 - Problemi di interoperabilità
 - Difficoltà di applicazione in aree pericolose
 - Sconvolgimento delle di progettazione

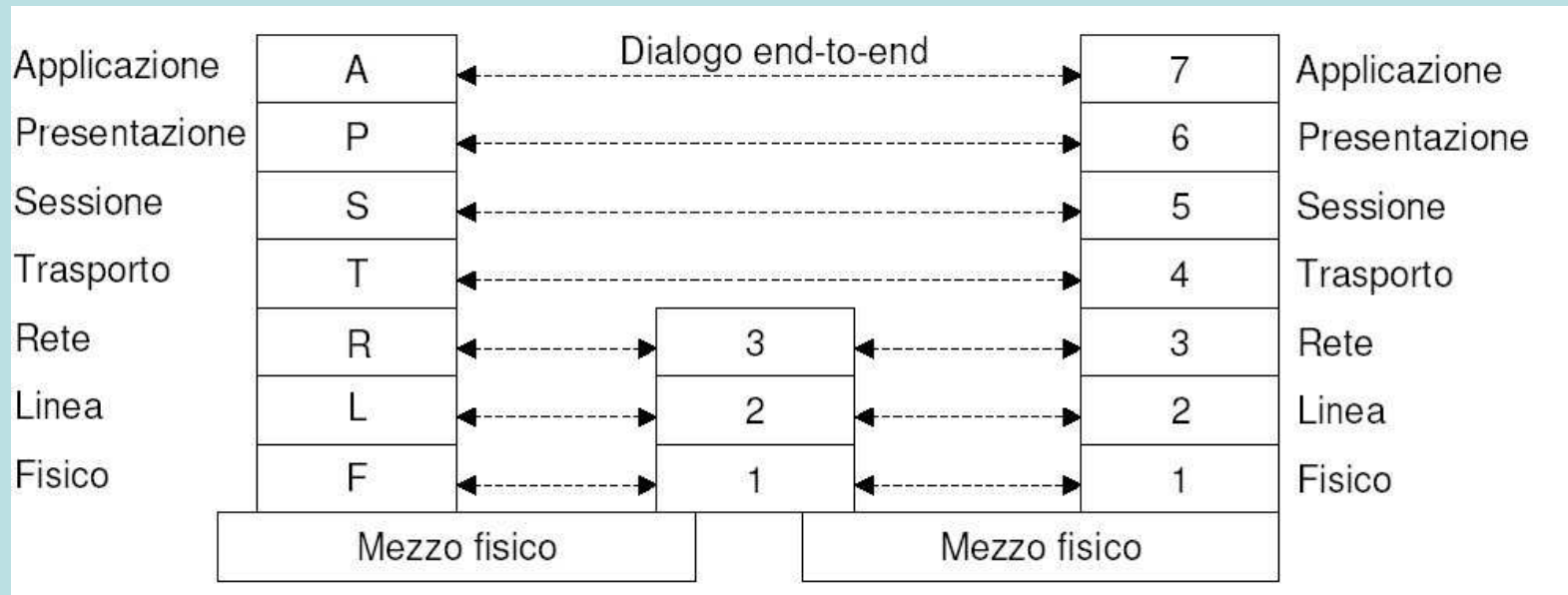


Interazione bus di campo



ISO/OSI

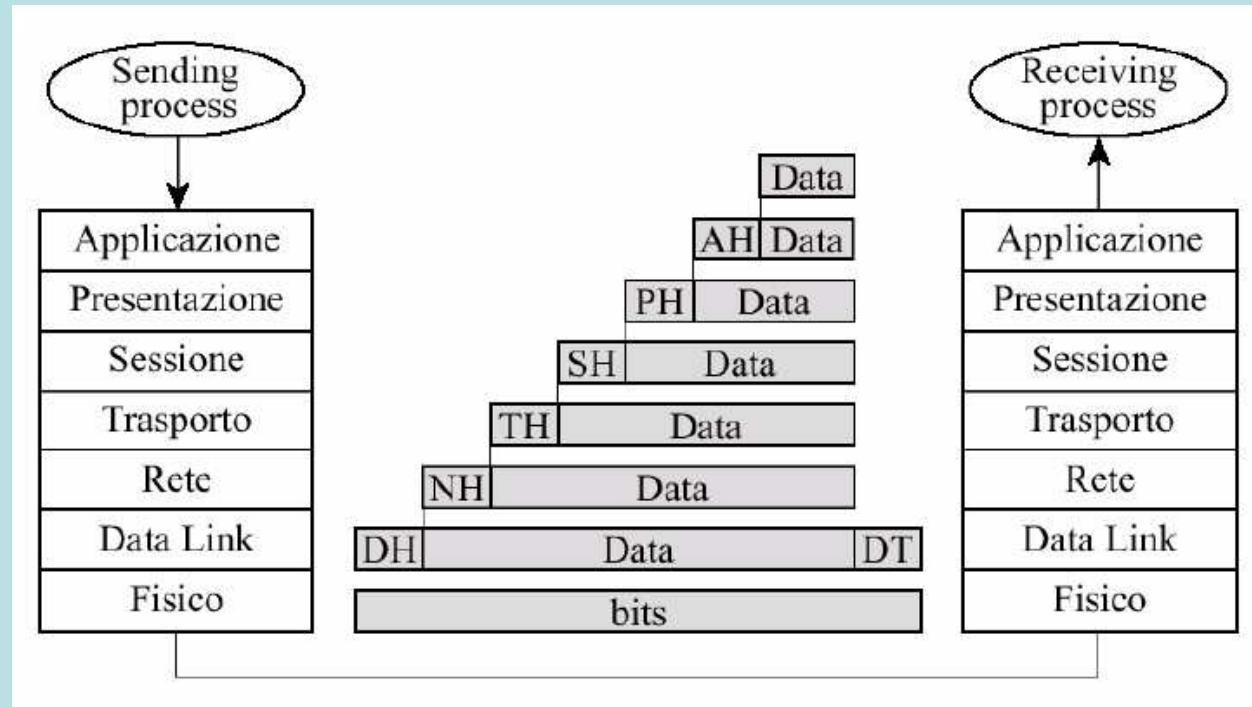
- OSI = Open System Interconnection
- ISO = International Standard Organization



I principi del modello OSI sono i seguenti:

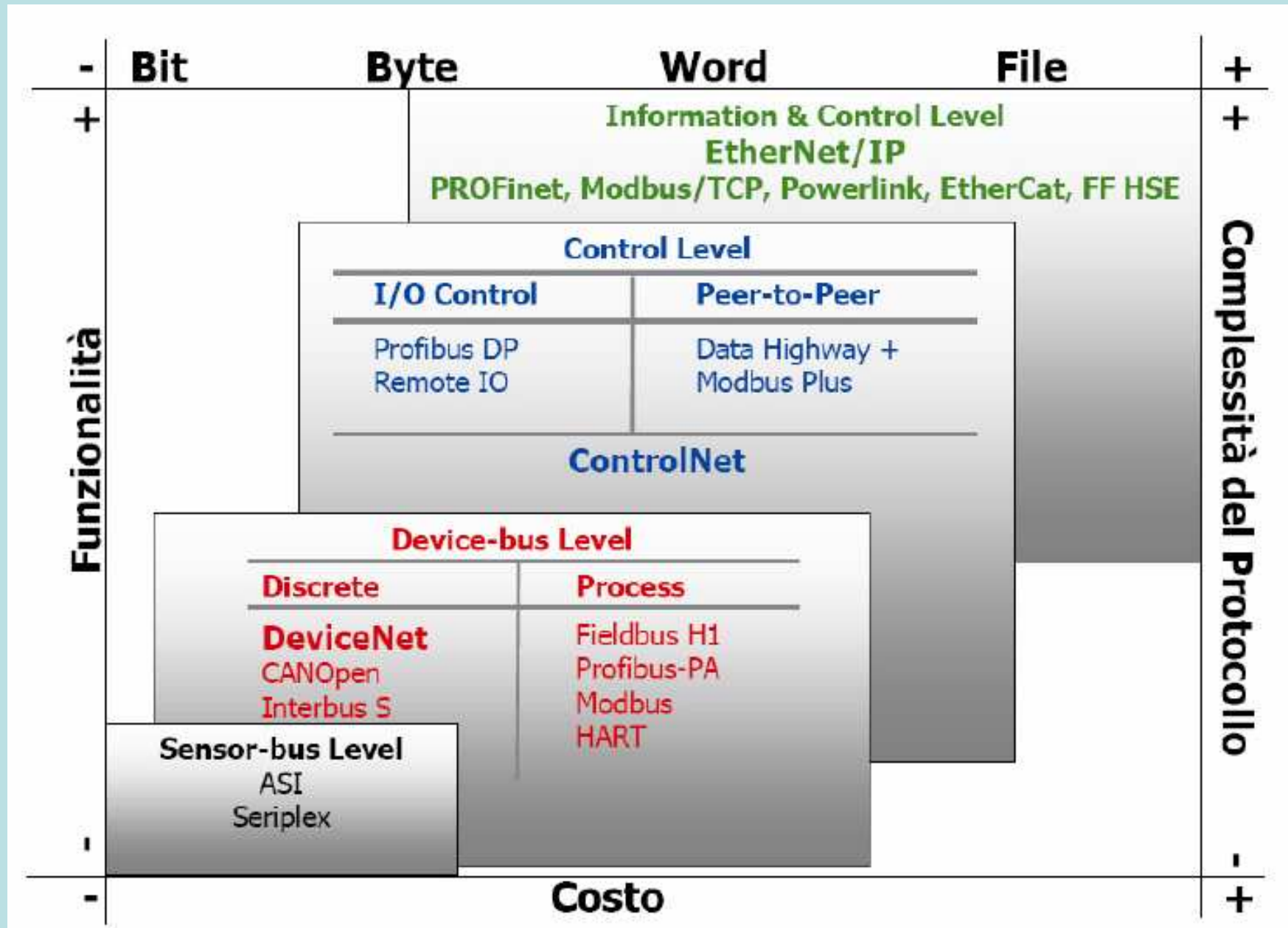
- Ogni livello deve avere una funzione ben definita.
- La scelta dei livelli deve: minimizzare il passaggio delle informazioni tra livelli.
- È necessario evitare troppe funzioni in un livello così come la definizione di troppi livelli.

ISO/OSI



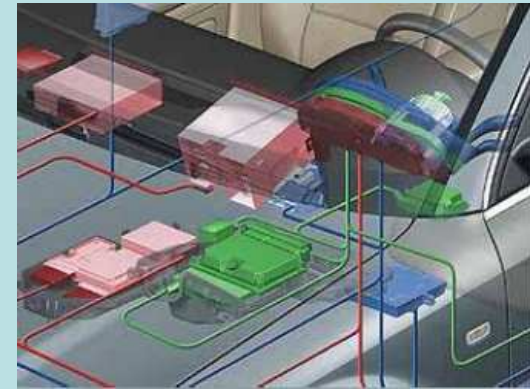
- **Simmetria:** i due utenti devono utilizzare le stesse funzioni logiche
- **Struttura gerarchica:** i vari livelli devono essere organizzati in una rigida gerarchia;
- **Modularità:** ogni livello deve avere definite le stesse funzioni interne e le stesse interfacce, in modo che possa essere sostituito da un altro

Classificazione bus di campo



CAN BUS

- Nel 1986 la Bosch, su richiesta del gruppo Mercedes e BMW, cercò di risolvere gran parte dei problemi che si riscontravano nella diffusione delle informazioni tra i sempre più numerosi dispositivi elettronici presenti all'interno di un'autovettura (air conditioning control, antiblock braking system, traction control etc.). Fu così che nacque il protocollo di rete **CAN (1.0)**



CAN BUS

- Il **CAN** (acronimo di **CONTROLLER AREA NETWORK**) è un protocollo di comunicazione seriale che assicura un elevato livello di sicurezza per controlli real-time.
- Il suo dominio applicativo spazia dalle reti ad alta velocità fino ad impianti elettronici di basso costo.
- Nel campo dell'automazione industriale motori, sensori e numerosi altri dispositivi elettronici sono connessi tramite CAN (con bit rate fino a 1 Mbit/s).

CAN e ISO/OSI

- Allo scopo di ottenere un modello trasparente ed un'implementazione flessibile, si potrebbe inquadrare il protocollo CAN in 3 differenti livelli:
 - il livello di OGGETTO
 - il livello di TRASFERIMENTO
 - il livello FISICO

LIVELLO DI OGGETTO

- filtraggio dei messaggi
- interpretazione dei messaggi

LIVELLO DI TRASFERIMENTO

- isolamento dei guasti
- rilevamento di errori
- segnalazione di errori
- acknowledgement
- arbitraggio
- suddivisione in pacchetti
- sincronizzazione

LIVELLO FISICO

- rappresentazione binaria
- trasmissione dell'informazione

CAN e ISO/OSI

- Ai primi due livelli (livello di **OGGETTO** e di **TRASFERIMENTO**) sono associati tutti i servizi e le funzioni del livello **DATA-LINK** definito nel modello ISO/OSI.
- Lo scopo principale del livello **DATA-LINK** è di trasformare una trasmissione grezza in una linea per il livello superiore che appaia libera da errori di trasmissione non segnalati.
- Visto che il livello **FISICO** accetta e trasmette sequenze di bit senza far riferimento al loro significato o alla loro struttura, è compito del livello data-link creare e riconoscere i limiti dei pacchetti.
- Il livello di **OGGETTO** è in grado inoltre di:
 - capire quali messaggi sono stati trasmessi
 - decidere quali messaggi ricevuti dal livello di trasferimento vanno processati

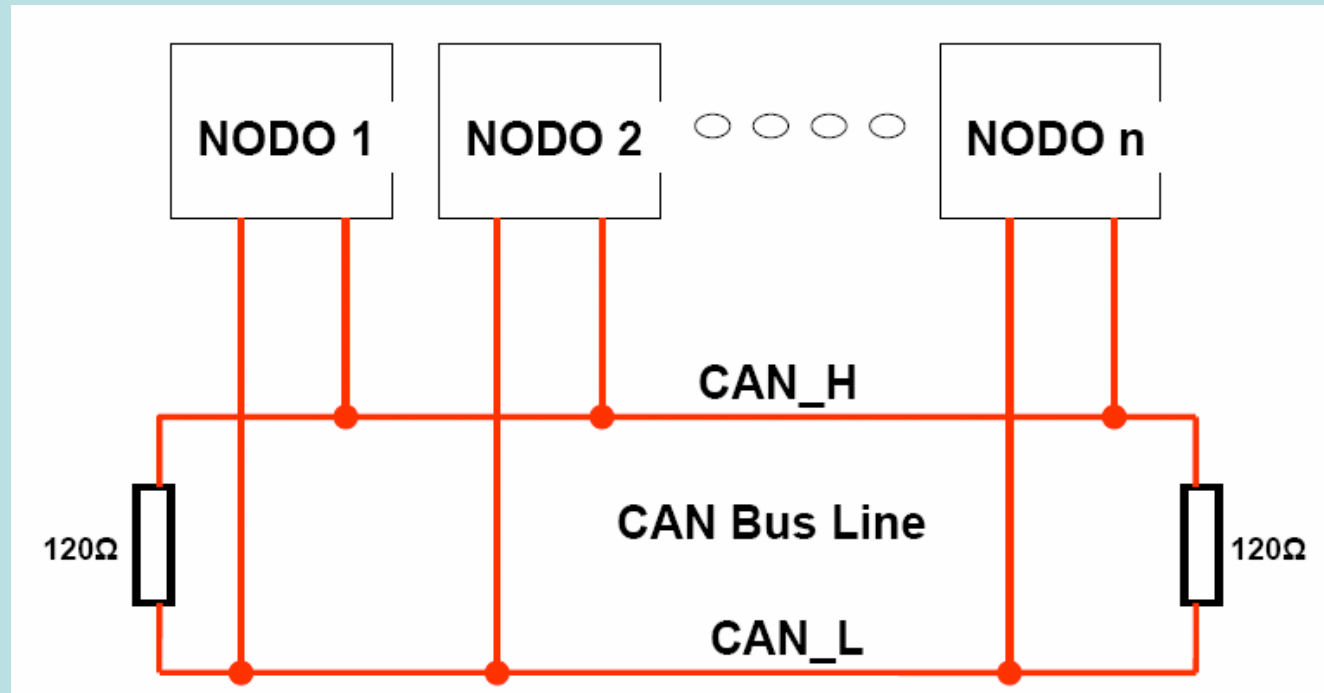
CAN e ISO/OSI

- L'obiettivo del livello di **TRASFERIMENTO** è principalmente il controllo dei pacchetti, il rilevamento di errori e l'invio di messaggi di errore.
- All'interno del livello di trasferimento si valuta sia quando il bus è libero per cominciare una nuova trasmissione sia quando dare inizio alla fase di ricezione.
- Il livello **FISICO** si occupa del trasferimento dei bit di informazione tra nodi differenti.
- Evidentemente il livello fisico, all'interno di una rete, deve essere lo stesso per tutti i nodi.

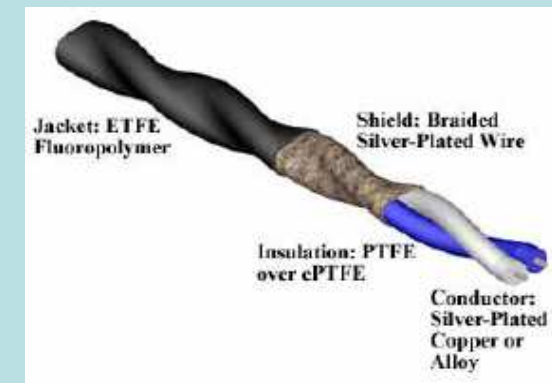
Proprietà principali CAN BUS

- gestione di vari livelli di priorità associati ai singoli messaggi
- garanzia di tempi di latenza
- flessibilità di configurazione
- ricezione multicast con sincronizzazione temporale
- rilevamento e segnalazione di errori
- ritrasmissione automatica di messaggi errati
- gestione di errori temporanei
- gestione di stati di malfunzionamento permanente
- isolamento automatico di nodi difettosi

Livello Fisico



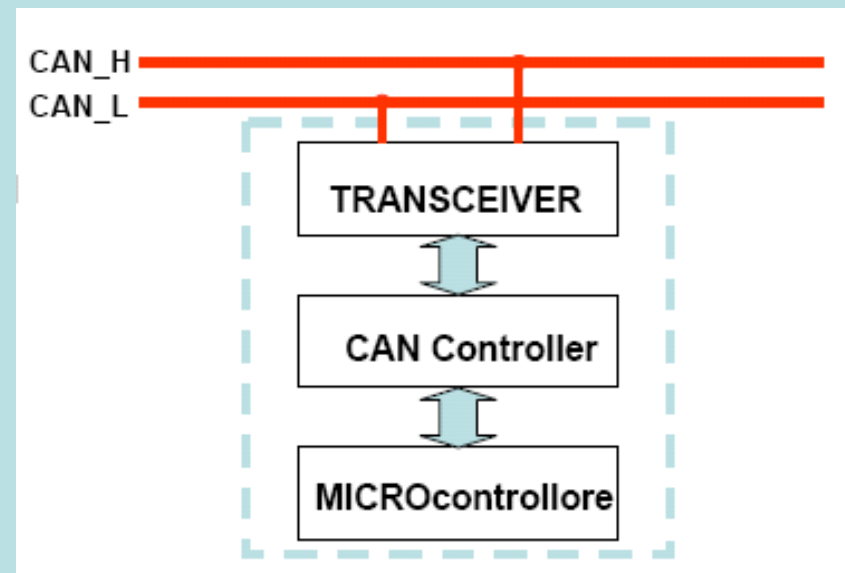
Il cavo trasmissivo è un doppino intrecciato (schermato o no a seconda delle applicazioni) terminato con impedenze di valore pari a 120Ω .



Livello fisico

Ogni nodo di una rete CAN è composto da 3 parti principali:

- **TRANSCEIVER:**
 - alimentato tipicamente a 5V, rileva lo stato del bus valutando la differenza di tensione tra le due linee CAN_H e CAN_L.
 - $\Delta < 0.5 \text{ V}$ – stato DOMINANTE
 - $\Delta > 0.9 \text{ V}$ - stato RECESSIVO
 - Interfaccia il Controllore CAN con il BUS
- **CAN CONTROLLER:**
 - trasmette e riceve dati via seriale al transceiver e comunica col microcontrollore.
- **MICROCONTROLLORE:**
 - gestisce le operazioni che deve svolgere il nodo.



Livello fisico (Power over Can)

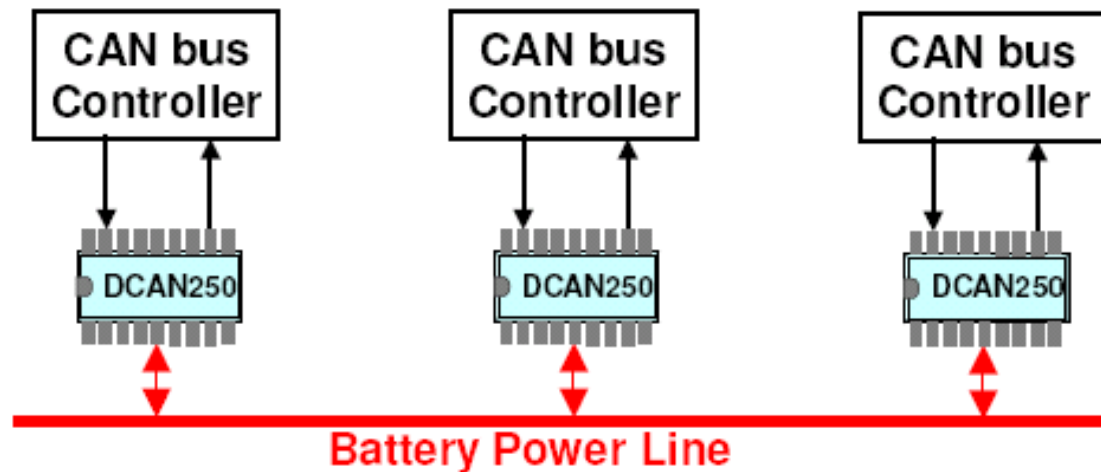
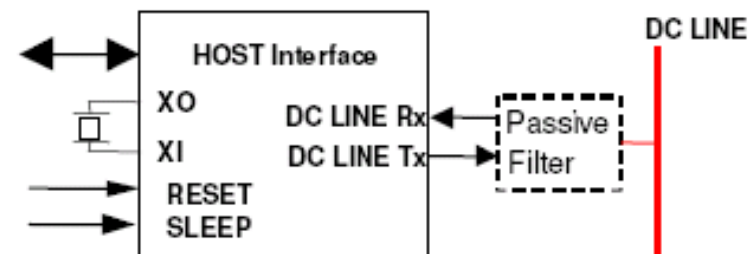


Figure 1 - Typical DCAN250 System

Characteristics

Packet data:	250Kbps
Modulation method:	DQPSK
Collision resolution:	Built in
Error correction codes:	Built in
Power save mode:	Built in
Packet size:	User defined
Interface:	CAN



Configurazione rete

- Un nodo CAN non usa informazioni sulla configurazione del sistema.
- Possono essere aggiunti nodi senza che vengano richiesti cambiamenti a livello hardware o software (**flessibilità del sistema**);

Messaggi

- Il contenuto dei messaggi è caratterizzato da un identificatore che **non** indica l'indirizzo di destinazione del pacchetto ma descrive semplicemente quanto vi è contenuto, affinché tutti i nodi della rete siano in grado di decidere attraverso il **filtraggio del messaggio** se “appropriarsene” o meno;
- la priorità è stabilita dal “tipo” di identificatore

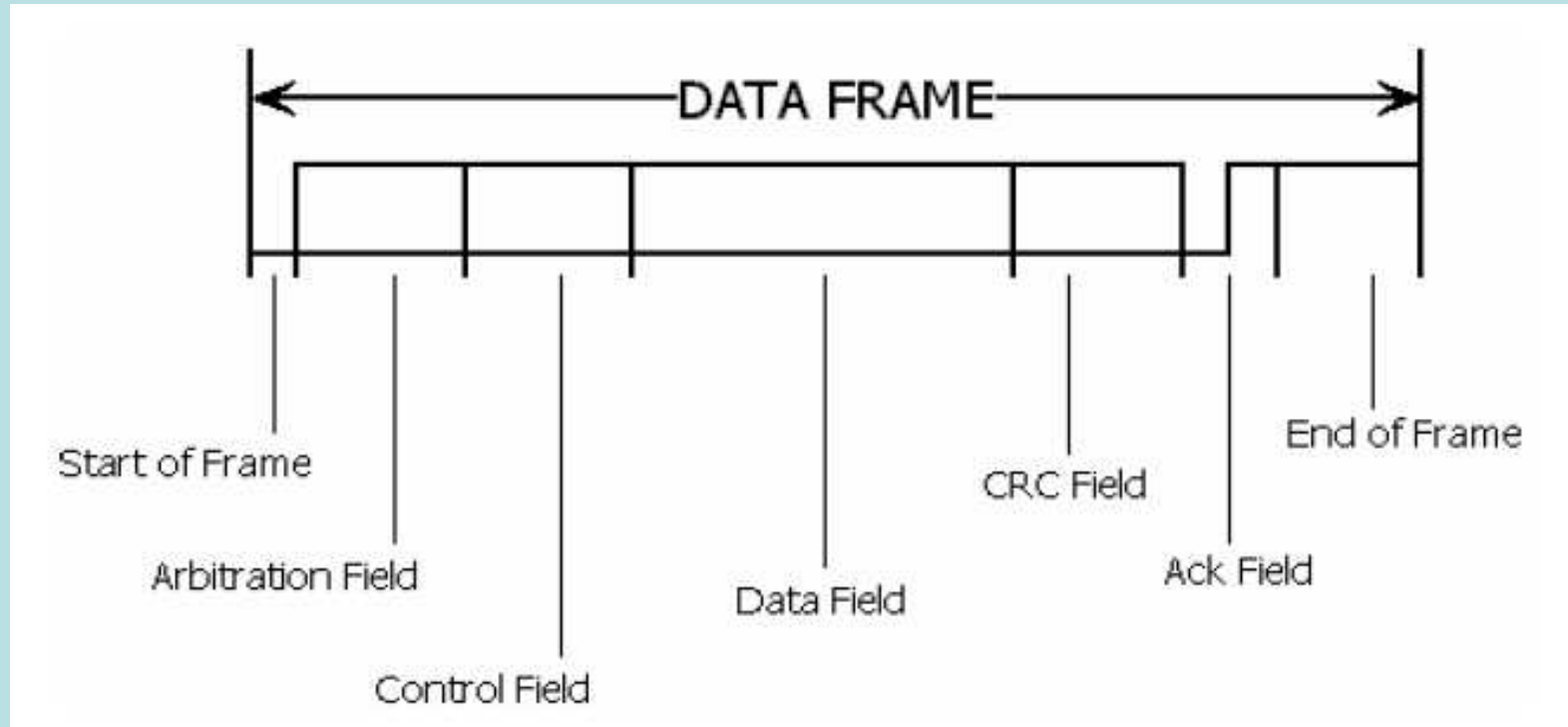
Accesso al canale

- **CSMA/CA** (Carrier Sense Multiple Acces with Collision Avoidance).
- Quando il bus è libero ogni unità è abilitata a trasmettere informazioni (**multimaster**).
- Solo l'unità a cui è associato il messaggio con priorità più alta si guadagna l'**accesso al canale**.
- Durante la trasmissione contemporanea di un **bit 1** ed un **bit 0**, il valore riscontrabile sul canale sarà **0**. Di conseguenza d'ora in poi si parlerà di bit '**dominante**' (bit 0) e bit '**recessivo**' (bit 1).

Remote frame

- Attraverso l'invio di un **REMOTE FRAME**, un nodo in attesa di informazioni può richiedere il **DATA FRAME** contenente i dati di interesse.
- Naturalmente il **DATA FRAME** ed il corrispondente **REMOTE FRAME** hanno lo stesso identificatore.

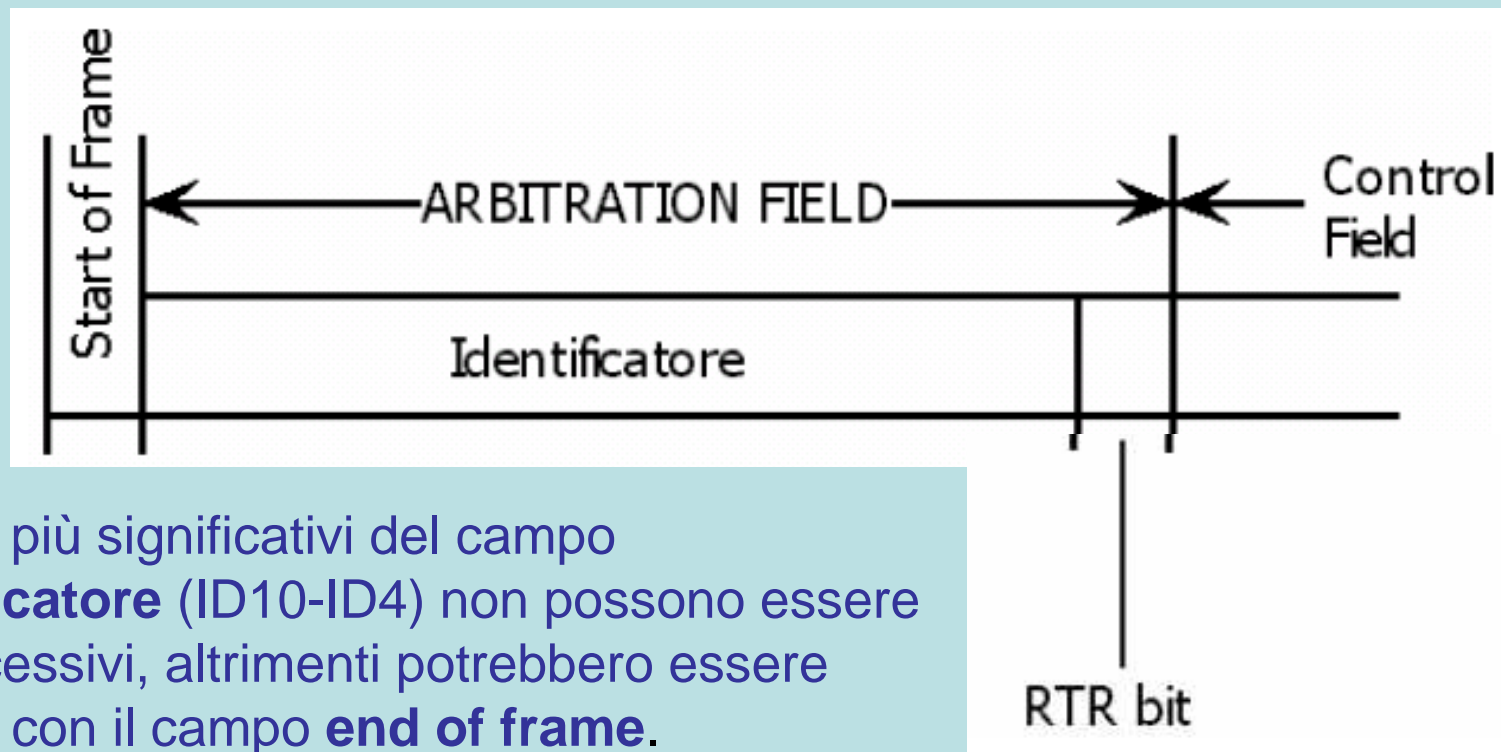
Data Frame



- Il campo **START OF FRAME** marca con un singolo bit dominante l'inizio di ogni data frame e remote frame.
- Il campo **END OF FRAME** è costituito da una sequenza di 7 bit recessivi, sia nei data frame che nei remote frame.

Arbitration field

- comprende un campo **identificatore** (11 bit) ed un **RTR** bit (Remote Transmission Request)



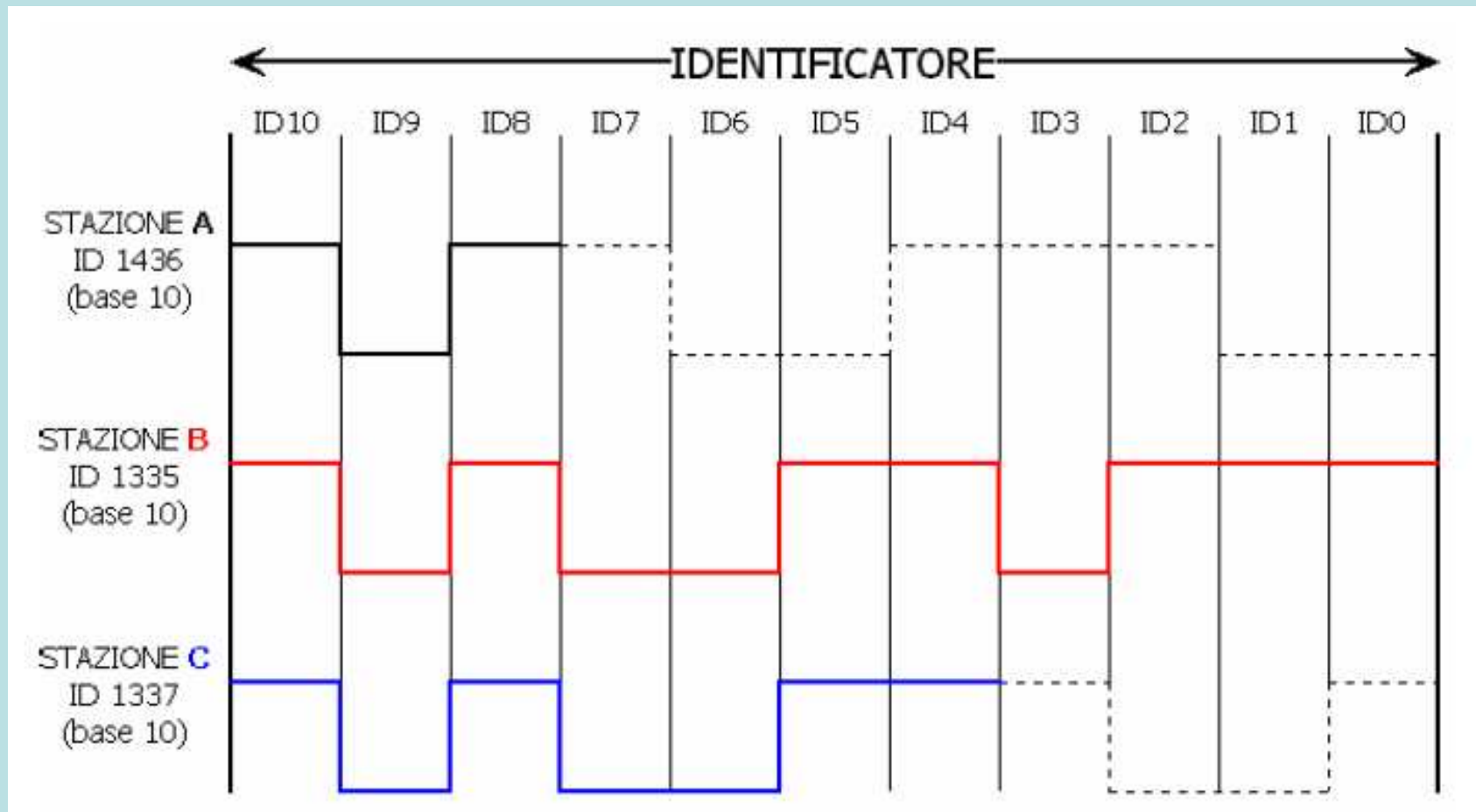
- I 7 bit più significativi del campo **identificatore** (ID10-ID4) non possono essere tutti recessivi, altrimenti potrebbero essere confusi con il campo **end of frame**.
- L'**RTR bit** assume valore dominante se si tratta di un data frame, valore recessivo, invece, nel caso di remote frame.

Arbitraggio

- Quando 2 o più unità stanno trasmettendo contemporaneamente, il conflitto è risolto con un meccanismo di **ARBITRAGGIO** che assicura che non siano perse né informazioni né tempo.
- Se un data frame ed un remote frame con lo stesso identificatore sono trasmessi contemporaneamente, il data frame prevale sul remote frame.
- Durante la fase di arbitraggio ogni trasmettitore confronta il livello del bit trasmesso con il livello monitorato sul canale.
- Se i 2 livelli coincidono l'unità continua a trasmettere. Quando il livello associato al bit è recessivo mentre sul canale si riscontra un livello dominante, l'unità interrompe immediatamente la trasmissione.

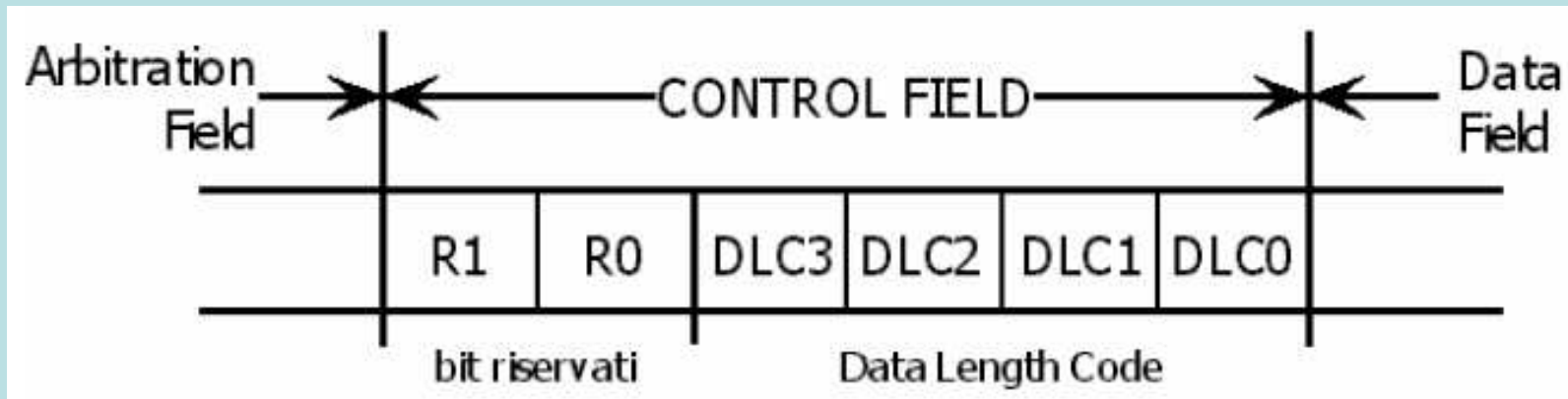
Arbitraggio

- Segue un esempio di **conflitto** tra 3 stazioni che si contendono l'accesso al canale.



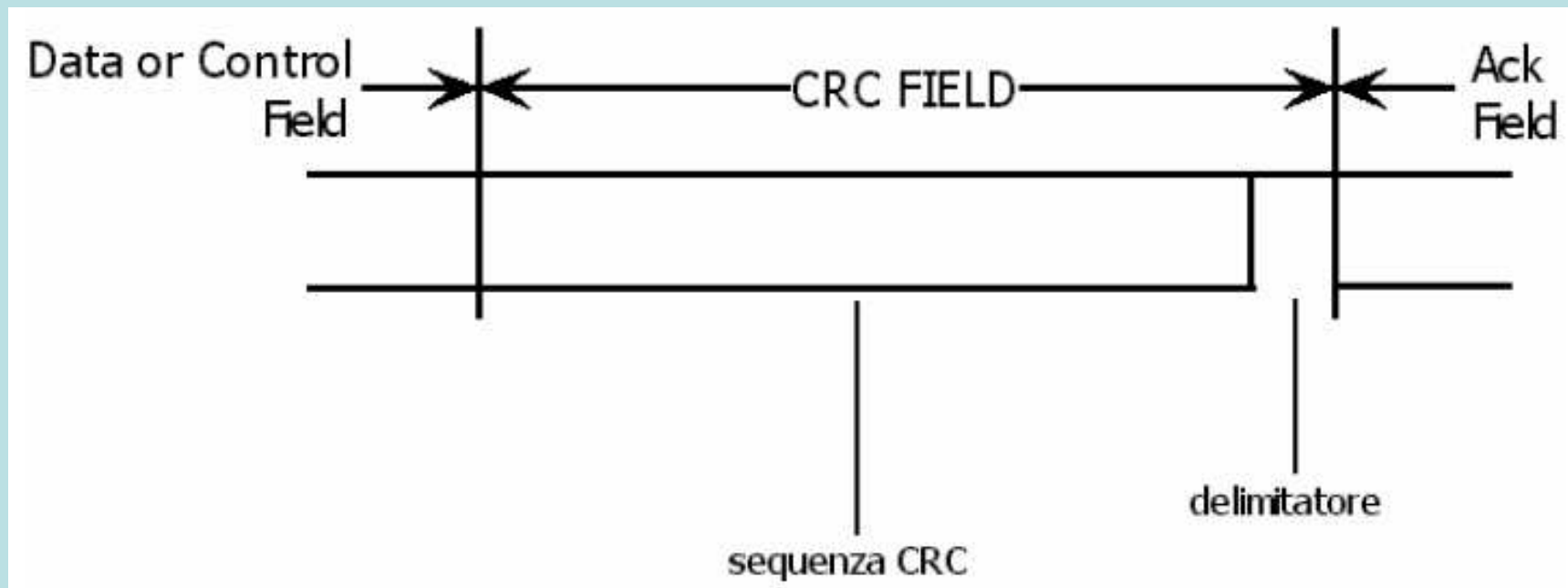
Control Field

- Il **CONTROL FIELD** occupa 6 bit, di cui 4 sono destinati al data length code e 2, entrambi dominanti, sono riservati per eventuali espansioni future.



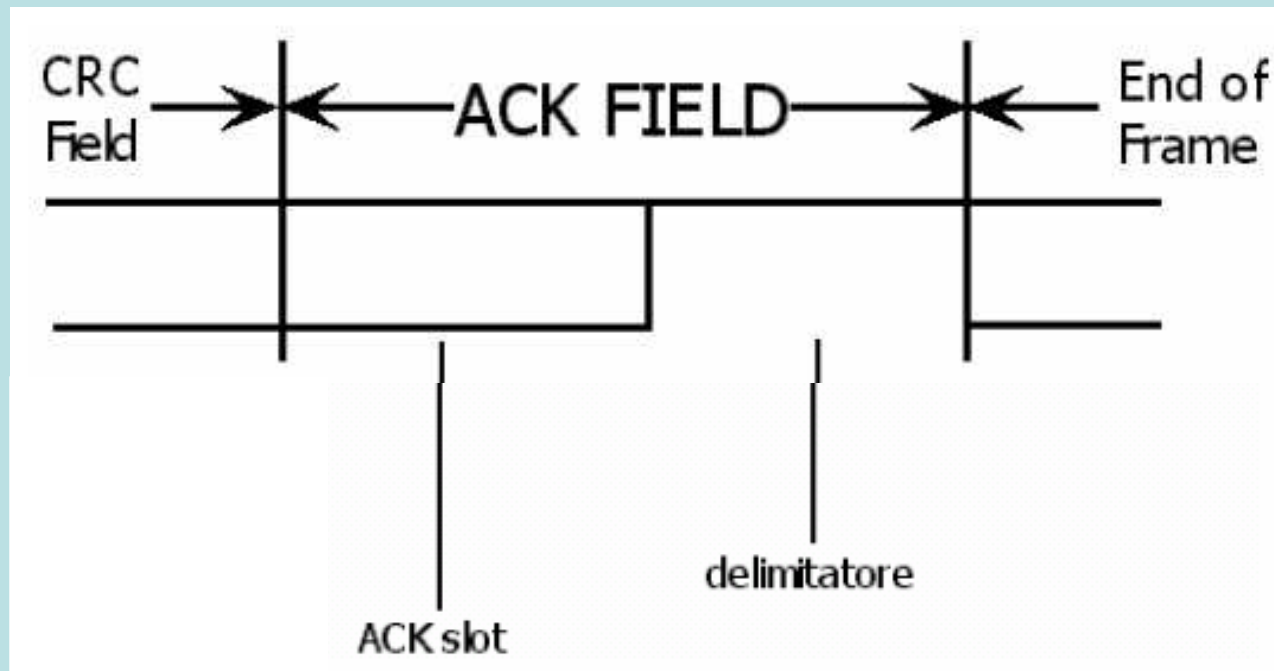
Data Field e CRC Field

- Il **DATA FIELD** contiene i dati da trasferire sul canale e può occupare da 0 ad 8 byte.
- Il **CRC FIELD** comprende 16 bit, di cui i primi 15 costituiscono una sequenza di controllo ottenuta da un **codice a ridondanza ciclica** e l'ultimo bit (recessivo) fa da **delimitatore** ($x^{15} + x^{14} + x^{10} + x^8 + x^7 + x^4 + x^3 + 1$).



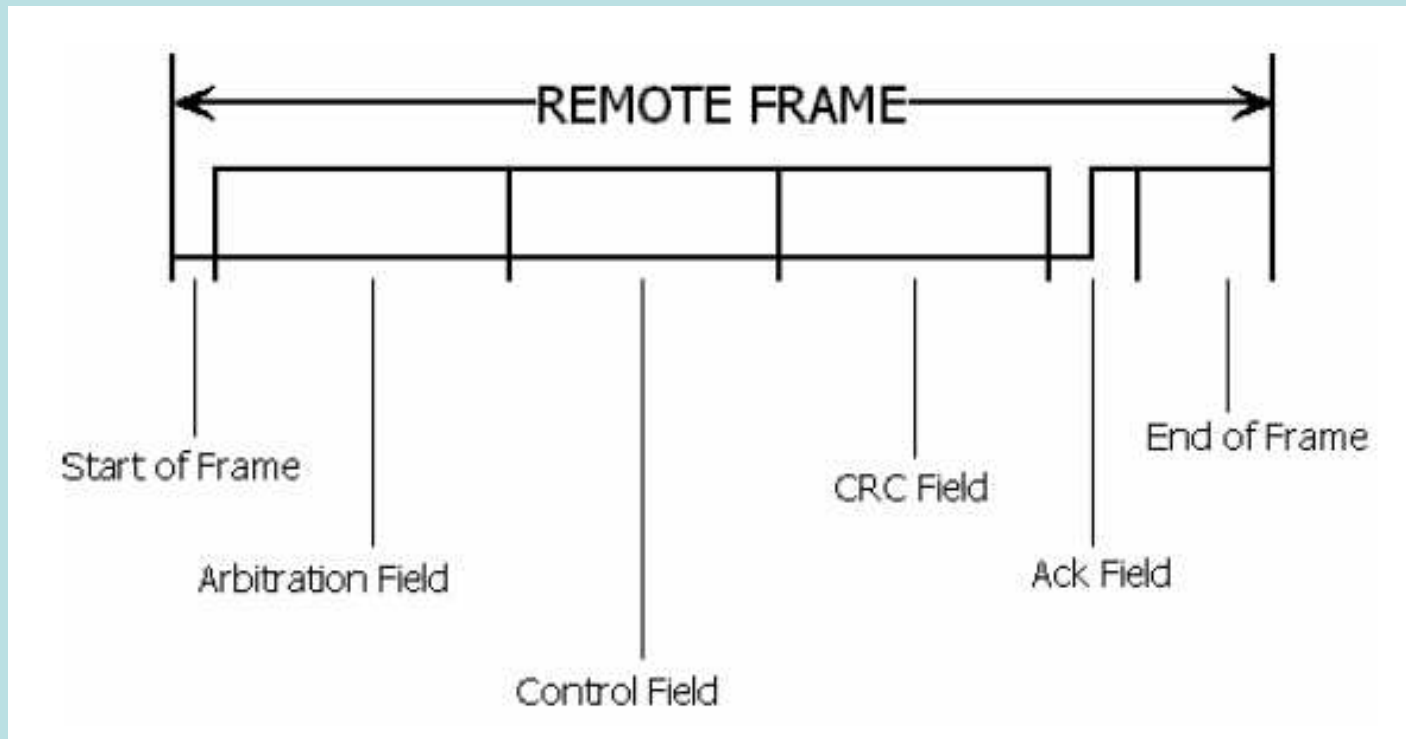
ACT Field

- L'**ACK FIELD** occupa 2 bit, **ACK slot** ed **ACK delimiter**, quest'ultimo sempre recessivo.
- Ogni stazione trasmittente invia un **ACK slot** contenente un bit recessivo.
- Un ricevitore che abbia ricevuto correttamente un messaggio, comunica l'evento al trasmettitore ponendo a 0 il bit associato all'**ACK slot**.



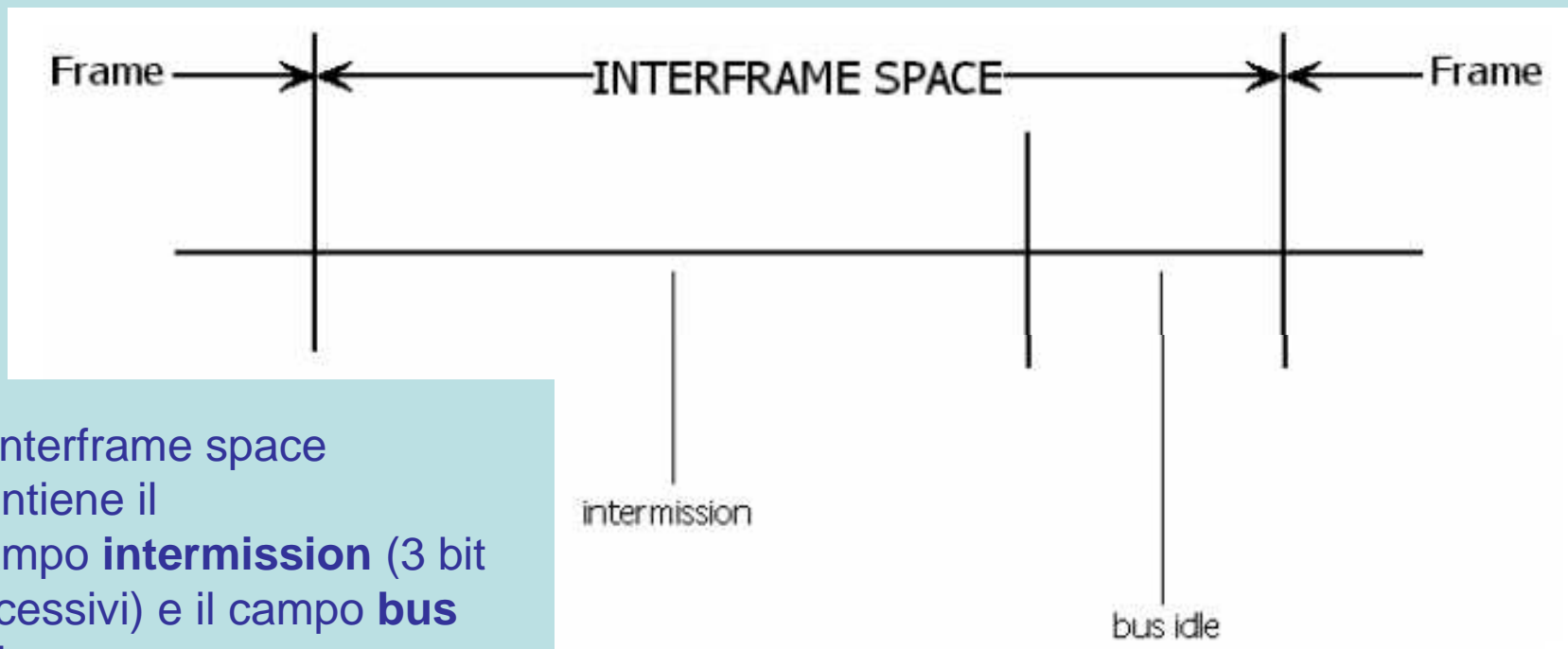
Remote Frame

- **Un remote frame è a tutti gli effetti un data frame privo del data field.**



Interframe Space

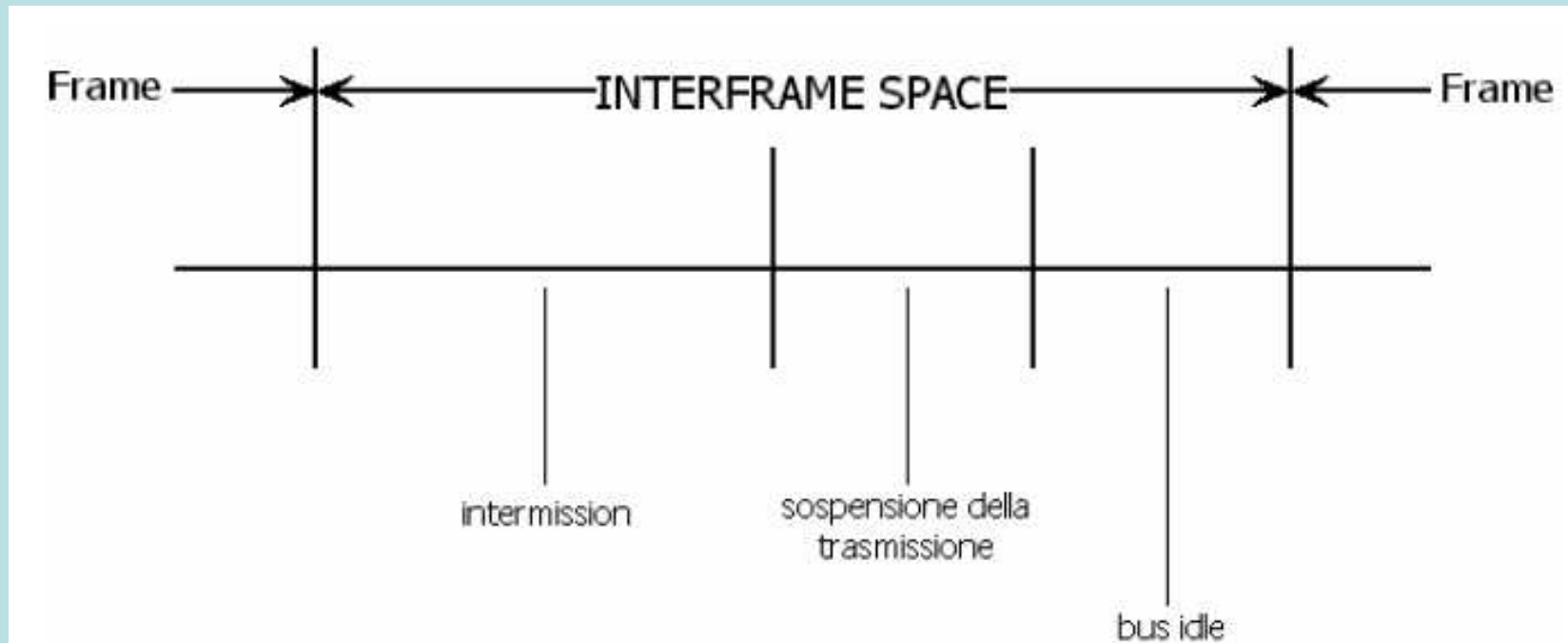
- **Data frame e remote frame** sono separati dai pacchetti che li precedono (data frame, remote frame, error frame, overload frame) da un campo di bit detto **INTERFRAME SPACE**.
- **Error frame ed overload frame** (anche multipli), invece, non sono preceduti da interframe space.



L'interframe space contiene il campo **intermission** (3 bit recessivi) e il campo **bus idle**

Suspend Transmission Field

- Per le stazioni **'error passive'** che hanno trasmesso il messaggio precedente esiste un ulteriore campo, detto **suspend transmission field**.



Durante la fase di **intermission** nessuna stazione è abilitata a cominciare la trasmissione di un eventuale data frame o remote frame.
L'unica azione possibile è la segnalazione di una **condizione di overload**.

Bus Idle

- La fase di **bus idle** può essere di lunghezza arbitraria.
- Ogni stazione che abbia qualcosa da trasmettere può accedere al bus.
- Un messaggio, la cui trasmissione è stata interrotta durante l'immissione sulla rete di un ulteriore messaggio, comincia ad essere ritrasmesso a partire dal primo bit dopo l'intermission field.
- Il rilevamento di un bit dominante sul canale è interpretato come **start of frame**.
- Dopo aver trasmesso un messaggio, una **stazione 'error passive'** manda a seguito dell'intermission field (esattamente prima di iniziare a trasmettere un ulteriore messaggio o prima dell'eventuale spazio di bus idle) 8 bit recessivi.
- La stazione è stata messa così in condizione di perdere la fase di arbitraggio e, se nel frattempo ha inizio una nuova trasmissione (causata da un'altra unità), la stazione diventerà automaticamente ricevente di tale messaggio.

Isolamento dei guasti

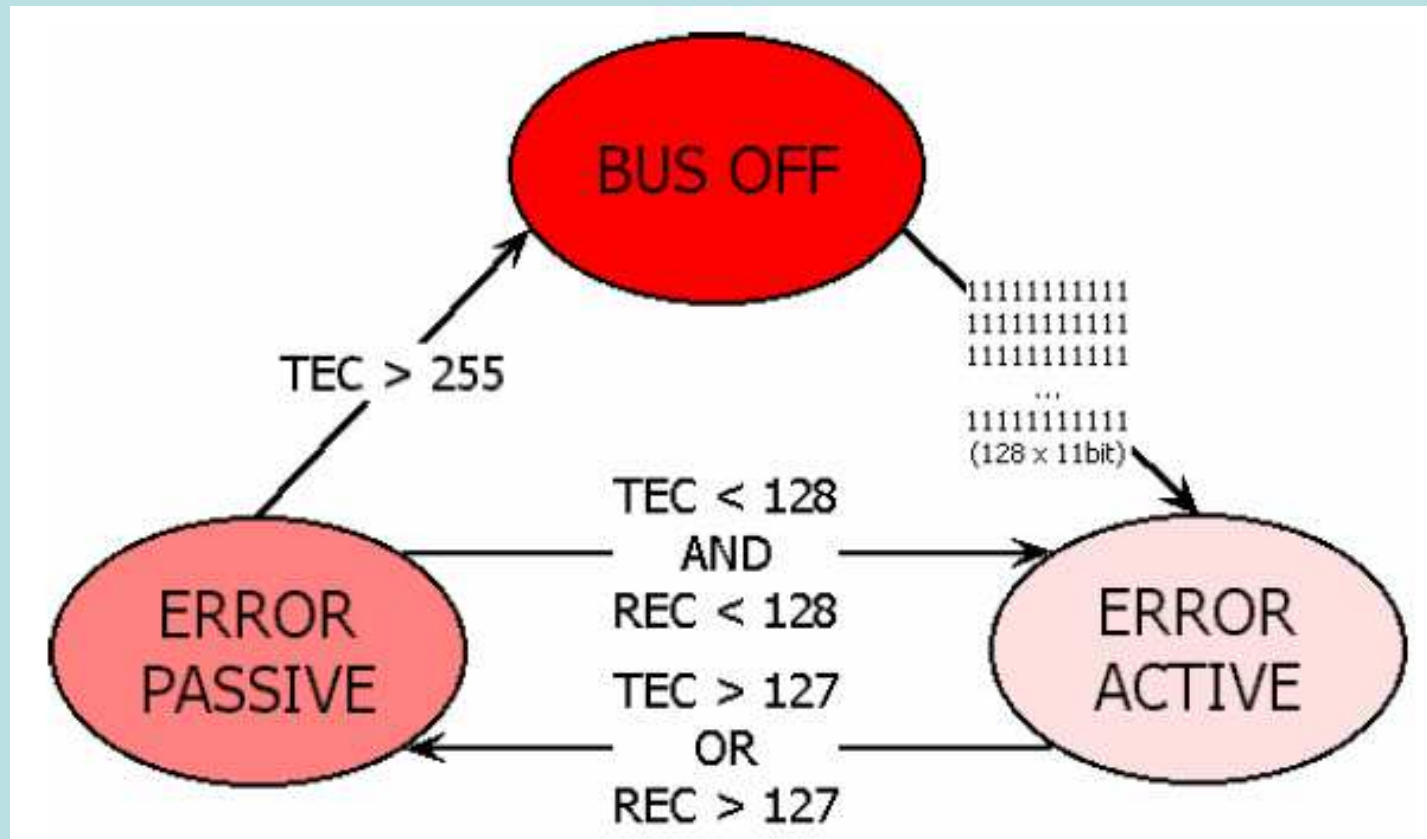
- Per quanto concerne l'isolamento dei guasti, ogni stazione può trovarsi in 3 stati differenti:
 1. ERROR ACTIVE
 2. ERROR PASSIVE
 3. BUS OFF
- Una stazione 'error active' al rilevamento dell'errore attiva l'**active error flag**.
- Una stazione 'error passive' comunica la presenza di un eventuale errore attivando il **passive error flag**.
- Una stazione in stato di '**bus off**' non può più inviare informazioni sul canale.
- Esistono 2 contatori a bordo di ogni stazione, entrambi responsabili di eventuali passaggi di stato:
 - il **TEC (transmit error counter)**
 - il **REC (receive error counter)**

Regole passaggio di stato

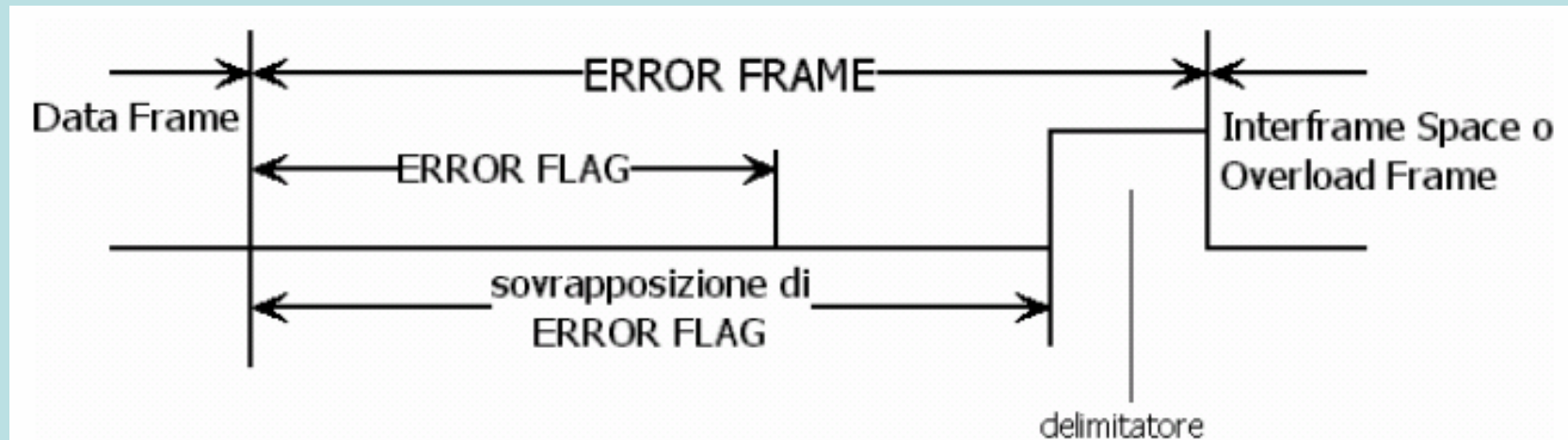
1. Quando un ricevitore rileva un errore il **REC** è incrementato di 1 unità, a meno che non si stia trasmettendo un active error flag o un overload flag.
2. Quando un ricevitore rileva un bit dominante come primo bit dopo aver trasmesso un error flag, il **REC** sarà incrementato di 8 unità.
3. Quando un trasmettitore invia un error flag, il **TEC** viene incrementato di 8 unità.
4. Se un trasmettitore rileva un bit error mentre sta inviando un active error flag o un overload flag, il **TEC** è incrementato di 8 unità.
5. Se un ricevitore rileva un bit error mentre sta inviando un active error flag o un overload flag, il **REC** è incrementato di 8 unità.

6. Ogni nodo tollera fino a 7 bit dominanti consecutivi dopo aver inviato un active/passive error flag o un overload flag. Dopo il quattordicesimo bit dominante consecutivo (in caso di active flag o di overload flag) o dopo aver rilevato l'ottavo bit dominante consecutivo dopo un passive error flag o dopo ogni altra sequenza di 8 bit dominanti consecutivi, ogni trasmettitore ed ogni ricevitore incrementa rispettivamente il TEC ed il REC di 8 unità.
7. Dopo una trasmissione riuscita di un messaggio (ack ricevuto e nessun errore rilevato) il **TEC** viene decrementato di 1 unità.
8. Dopo aver ricevuto correttamente un messaggio, ogni stazione decrementa il proprio **REC** di 1 unità se il valore precedente era compreso tra 1 e 127, lo riporta, invece, ad un valore compreso tra 119 e 127 se in precedenza era maggiore di 127.
9. Un nodo è 'error passive' quando almeno uno tra **TEC** e **REC** è maggiore o uguale di 128.

10. Un nodo è in stato di 'bus off' quando il TEC è maggiore o uguale di 255.
11. Un nodo in stato di '**bus off**' può ritornare '**error active**' (con entrambi i contatori settati a zero) dopo che sono state rilevate sul canale 128 occorrenze di 11 bit recessivi consecutivi.



Error Frame

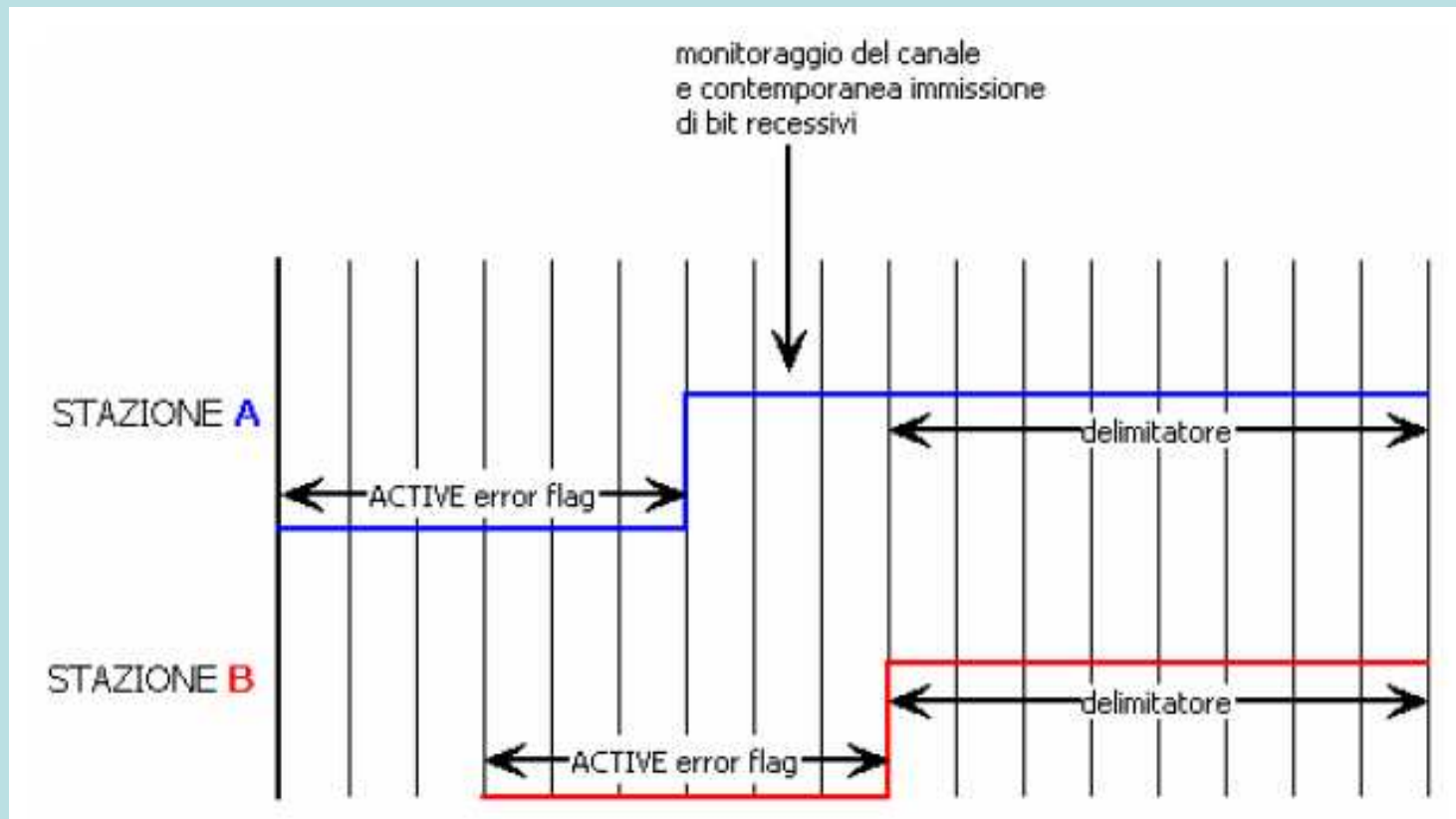


- Il primo non è altro che la **sovrapposizione di error flag** emessi da stazioni differenti, il secondo invece è un **delimitatore**.
- Ci sono 2 categorie di error flag: l'**active error flag** ed il **passive error flag**.
- Un **active error flag** è costituito da 6 bit dominanti (**000000**); un **passive error flag**, invece, è dato da una sequenza di 6 bit recessivi (**111111**), a meno di sovrascritture ad opera di bit dominanti provenienti da ulteriori nodi.

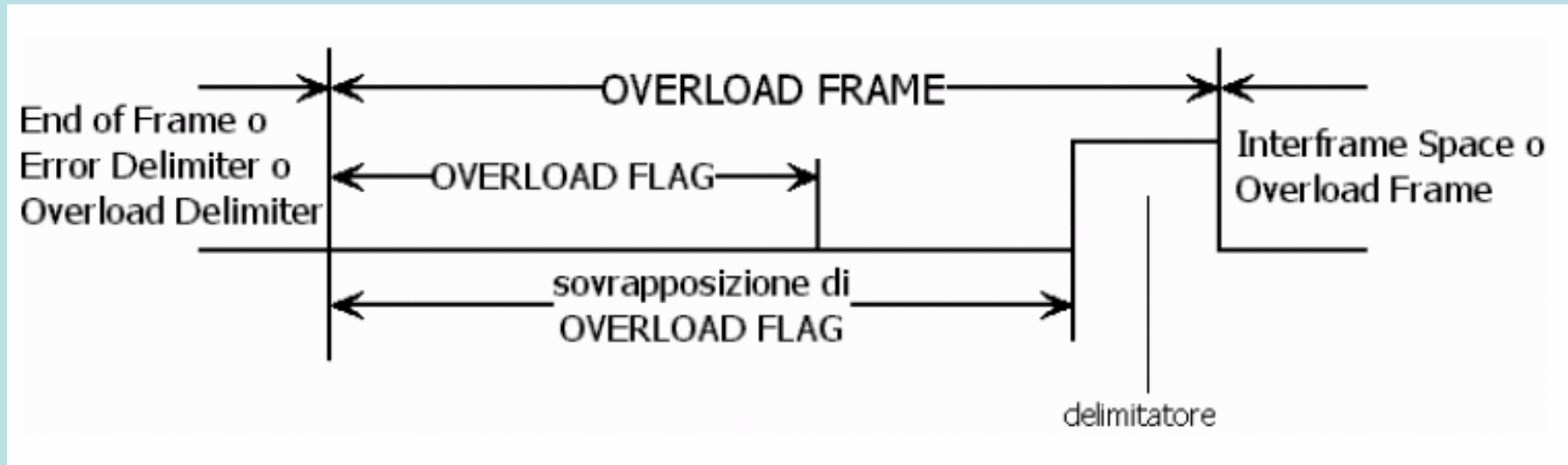
Active e passive error flag

- Una stazione '**error active**' segnala una condizione di errore trasmettendo un active error flag subito dopo il CRC field, distruggendo così ACK field e end of frame.
- Tutte le altre stazioni che si accorgono della condizione di errore segnalata danno inizio alla trasmissione del proprio error flag.
- Per effetto della sovrapposizione degli error flag, la sequenza di bit dominanti rilevabile sul bus può variare da un minimo di 6 ad un massimo di 12 bit.
- Una **stazione 'error passive'** tenta di segnalare una condizione di errore inviando un **passive error flag**.
- L'unità 'error passive', a partire dall'inizio della sequenza di **passive error flag**, si aspetta di monitorare sul canale 6 bit consecutivi dello stesso valore.
- Questo evento assicura alla stazione che la condizione di errore sia stata opportunamente diffusa sul canale.

- Il campo **delimitatore** è composto da 8 bit recessivi. Dopo la trasmissione di un error flag, ogni stazione invia bit recessivi e attende il primo bit recessivo sul canale. A questo punto si completa la trasmissione inviando i rimanenti 7 bit recessivi.



OVERLOAD Frame



- Ci sono 2 categorie **condizioni di overload**, che possono portare alla trasmissione di un overload flag:
 1. le **condizioni interne di un ricevitore**, che necessita di un ritardo della trasmissione del successivo data/remote frame
 2. Il rilevamento di un **bit dominante** durante la fase di **intermission** (caratteristica dell'**interframe space**)

Rilevamento errori

- Il meccanismo di rilevamento degli errori **garantisce che siano rilevati:**
 - tutti gli errori globali
 - tutti gli errori locali del trasmettitore
 - al massimo 5 errori distribuiti casualmente in un messaggio
 - un qualsiasi numero dispari di errori in un messaggio
- La probabilità di non rilevare messaggi contenenti informazioni errate è solitamente minore del valore della seguente espressione:

$$P < (4.7 \times 10^{-11} \times \text{error rate})$$

- Il tempo trascorso tra il rilevamento di un errore e la ritrasmissione del messaggio è di circa **29 bit time** (**bit time = 1 / bit rate**), se non ci sono altri errori.

Tipologie errori

- I campi start of frame, arbitration field, control field, data field e CRC field sono codificati col metodo del **BIT STUFFING**.
- **Ogni volta che il trasmettitore rileva 5 bit consecutivi identici, automaticamente inserisce un bit complementare nella sequenza da trasmettere.**
- Il protocollo CAN prevede **5 tipologie di errore** riscontrabili (e non mutuamente esclusive):
 1. **BIT ERROR**
 2. **STUFF ERROR**
 3. **CRC ERROR**
 4. **FORM ERROR**
 5. **ACKNOWLEDGEMENT ERROR**

Tipologie di errore

BIT ERROR

Una stazione trasmette un bit ma legge sul bus un valore differente.

Naturalmente questo tipo di errore non è riscontrato nella trasmissione di arbitration field, ack slot, passive error flag.

STUFF ERROR

È rilevato al sesto bit identico consecutivo in un campo che dovrebbe essere codificato con bit stuffing.

CRC ERROR

È rilevato dal ricevitore quando la sequenza CRC inviata dal trasmettitore risulta errata.

FORM ERROR

È rilevato quando un campo contiene valori illegali, quindi che non rispettano la forma fissata (ad esempio ack delimiter dominante anziché recessivo).

ACKNOWLEDGEMENT ERROR

È rilevato dal trasmettitore quando non riscontra sul canale un bit dominante all'interno del segmento ack slot (nessuna unità ha ricevuto il pacchetto).

Error Flag

- Quando una stazione rileva un **bit error, stuff error, form error o acknowledgement error**, questa comincia immediatamente la trasmissione dell'error flag.
- Se viene rilevato un **CRC error**, la trasmissione dell'error flag ha inizio subito dopo l'ack delimiter (a causa dei tempi di calcolo del CRC), a meno che un ulteriore error flag non sia già stato immesso sulla rete.

