

CAN Bus

Controller Area Network

Francesco Benzi
Dipartimento di Ingegneria Elettrica
Università di Pavia

CAN Controller Area Network

Il protocollo CAN è uno standard ISO standard (ISO 11898) per comunicazione seriale. È stato sviluppato da Bosch per applicazioni automotive. Attualmente è ampiamente usato anche per l'automazione industriale

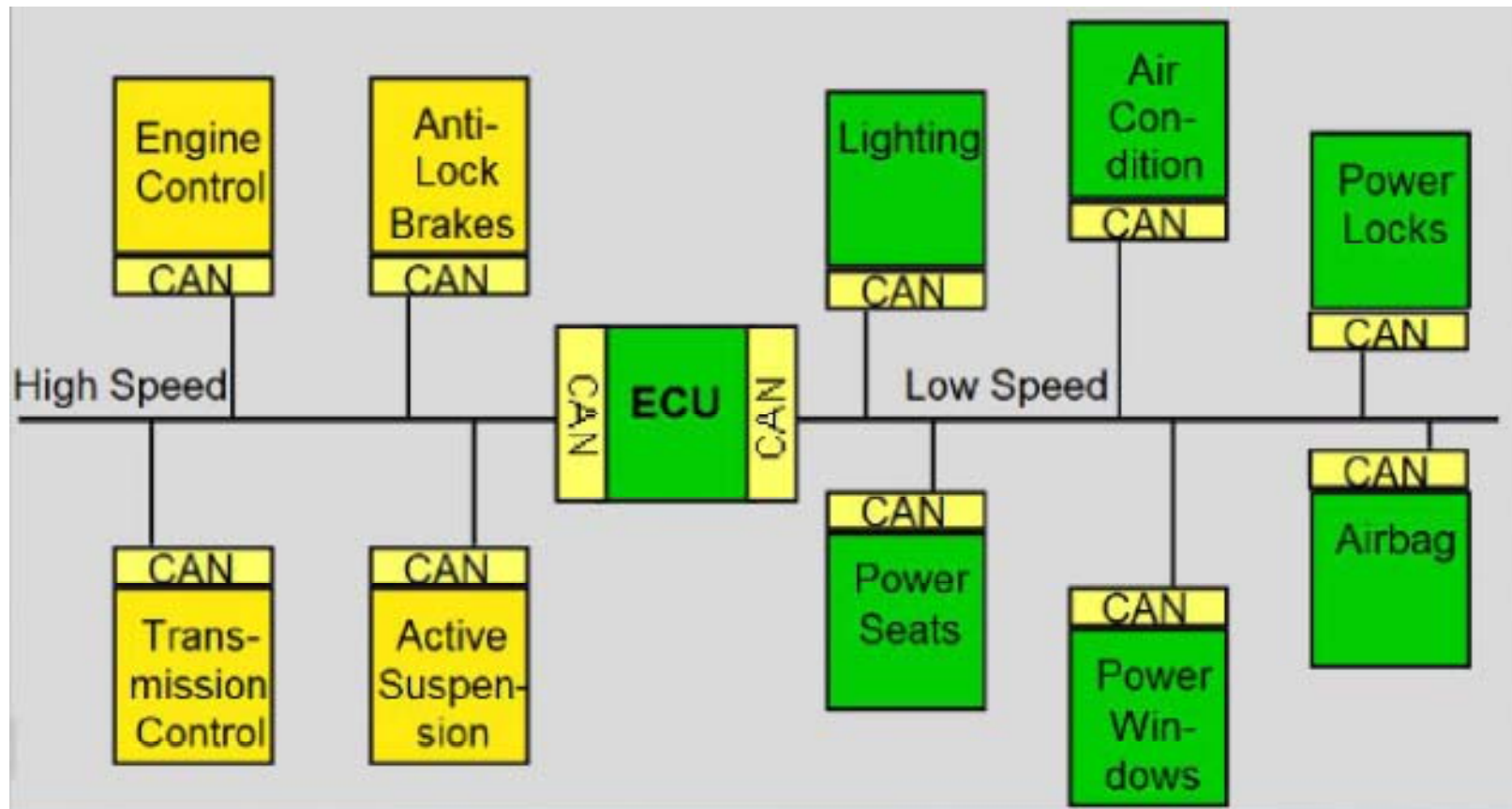
- **Standard diffuso dal 1986 basato sui soli primi 2 livelli ISO/OSI**
- **Principale mezzo di trasmissione è il doppino intrecciato**
- **Usa una tecnica di accesso al bus non distruttiva e quindi più efficiente**
- **Adotta una modalità efficace di riconoscimento e correzione dell'errore**
- **Consente di isolare eventuali nodi guasti senza che essi interferiscano col resto dell'impianto**
- **Nel 2012 Bosch ha pubblicato le specifiche di CAN FD (Flexible Data-Rate)**

Applicazioni di CAN Bus

- **Auto passeggeri Autocarri e bus**
- **Automezzi fuoristrada**
- **Treni e trasporto passeggeri**
- **Elettronica marittima**
- **Aerei e elettronica aerospaziale**
- **Automazione di fabbrica**
- **Controllo di macchine industriali**
- **Ascensori e scale mobili**
- **Automazione dell'edificio**
- **Elettronica e componenti biomedicali**
- **Controllo e dispositivi non-industriali**

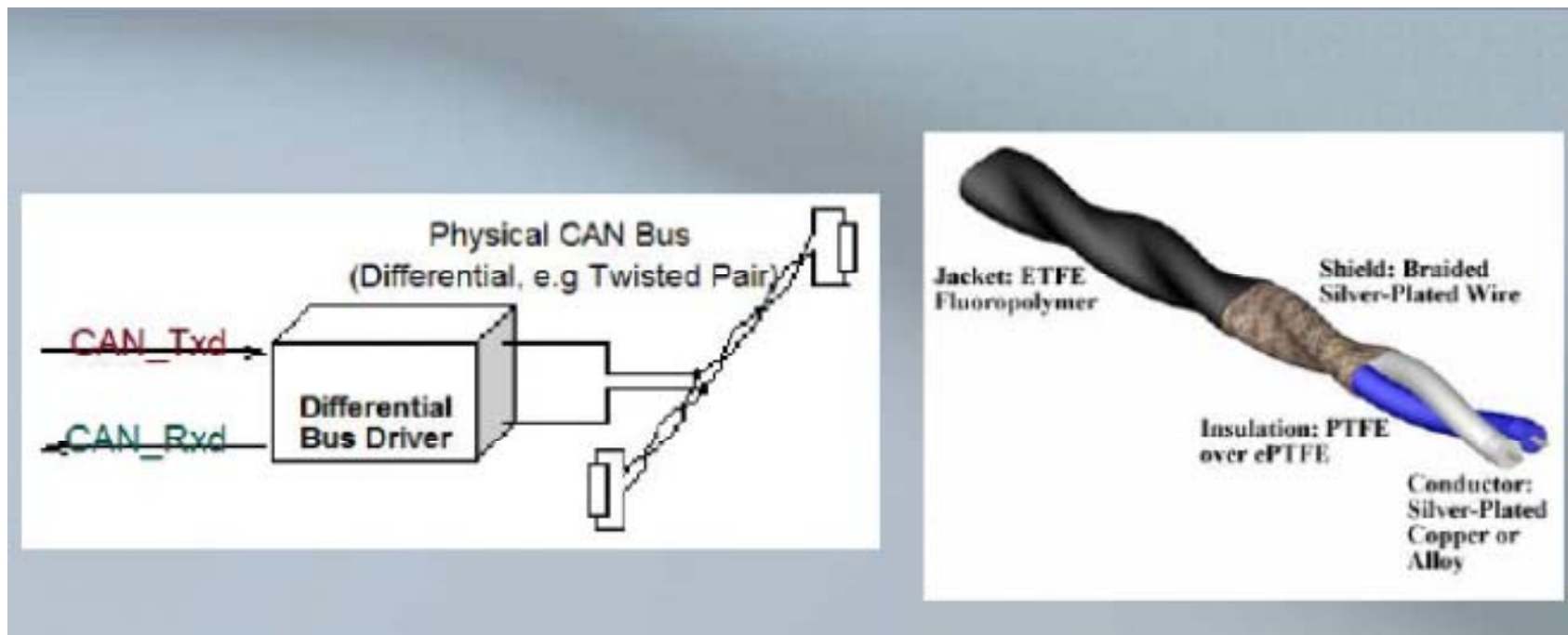
Applicazioni di CAN Bus

Rete CAN su un'automobile



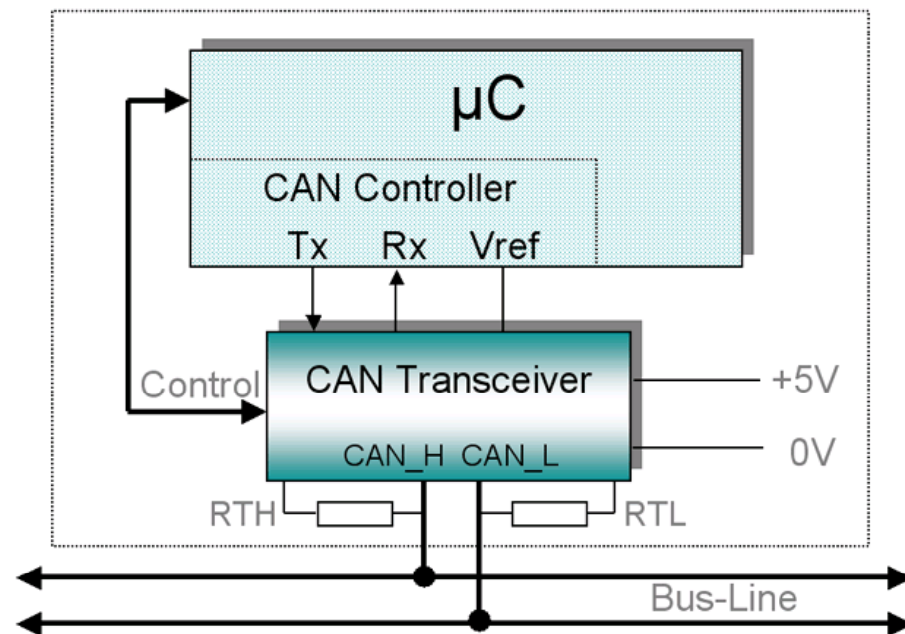
CAN – Mezzi di trasmissione

- Il più comune è definito dal CAN standard, ISO 11898: doppino in rame con segnale bilanciato ("high-speed CAN") per cancellare, compensandoli, gli eventuali disturbi che vengono indotti in modo equale sui due conduttori.
- Le due linee del doppino sono denominate una H e una L (high e low). Ad es. per rappresentare il bit 0, entrambe le linee restano a 2,5V (la loro differenza è di 0 volts). Il valore logico 1 viene rappresentato portando la linea H a 3,5V (rispetto a massa) e la linea L a 1,5V (Con d.d.p. tra le due linee di 2 volts).
- SAE J2411 definisce un collegamento single-wire (più massa) usato per applicazioni in auto (GM-LAN).



CAN – Mezzi di trasmissione

ISO standard 11519 definisce uno schema a due fili, bassa velocità (125 kbps), **resistente al guasto (fault tolerant)** (anche in caso di interruzione o messa a terra). ("low-speed CAN").



- Normalmente trasmette in modo differenziale.
- In caso di guasto di una linea può continuare a trasmettere su un solo conduttore
- In questo caso i nodi continuano a trasmettere con un livello ridotto S/N

Nodi o transceiver CAN prodotti da: **Philips**, Bosch, Infineon, Siliconix e Unitrode.

CAN – Velocità e segmenti

Velocità massima

- High speed Can 1 Mbit/s
- Low speed CAN fino a 125 kbit/s
- Su mezzi a filo singolo fino a 50 kbit/s

Lunghezza dei segmenti

- Limitazioni dovute alla necessità di una propagazione completa del segnale
- 1 Mbit/s 40 m
- 500 kbit/s 100 m
- 250 kbit/s 200 m
- 125 kbit/s 500 m
- 10 kbit/s 6 km

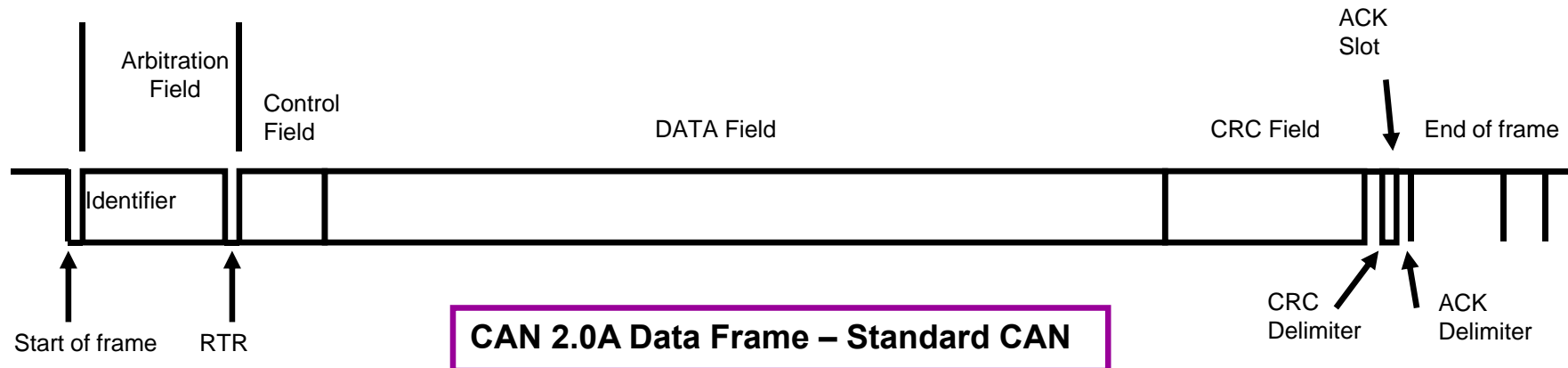
Caratteristiche dei cavi

- **Necessità di terminare il cavo per:**
 - Evitare segnali riflessi alle estremità
 - Assicurare un corretto livello in continua
- Impedenza nominale (di terminazione) di 120 Ω

4 Tipi di messaggio

- Data Frame
- Remote Frame
- Error Frame
- Overload Frame
- Interframe Space (precede ogni Data e Remote Frame e ha una funzione separatrice)

DATA Frame



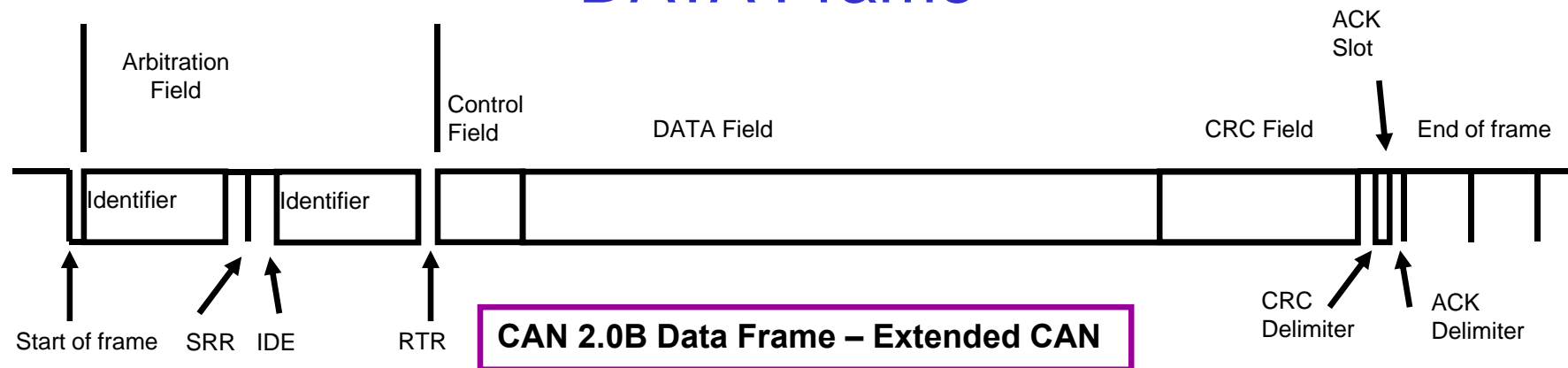
- **Arbitration Field**, determina la priorità del messaggio in caso di contesa. Contiene:

- Per CAN 2.0A (Standard CAN), un identificatore 11-bit e un Remote Transmission Request (RTR) bit, che è dominante per i data frames.

- Per CAN 2.0B (Extended CAN), un identificatore 29-bit e un RTR bit

- **Control Field**, è costituito da 6 bit, di cui 4 servono a specificare la lunghezza del Data Field (Data Length Code DLC) e 2 sono riservati per future espansioni del protocollo.

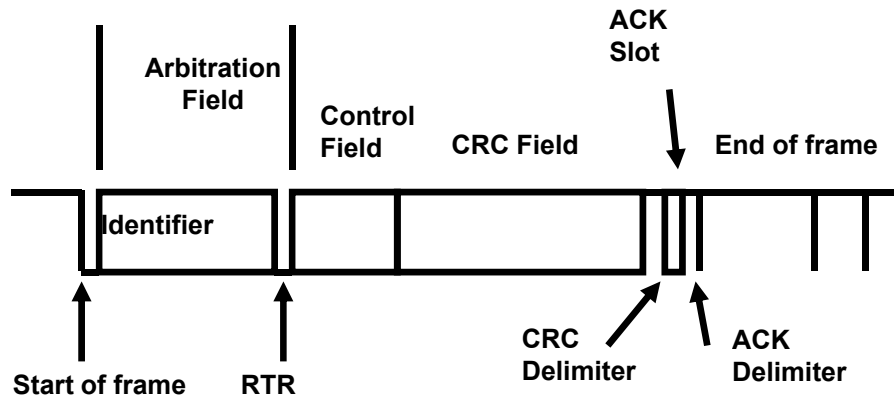
DATA Frame



- **Data Field** contiene da zero a 8 bytes di dati.
- **CRC (Cyclic Redundancy Check) Field** contiene 15-bit di verifica dell'errore sul messaggio.
 - È costituito da 16 bits, di cui i primi 15 contengono la sequenza di controllo mentre l'ultimo è un bit recessivo di delimitazione. Se il codice di ridondanza ciclica non rivela la presenza di un errore, il nodo invia un bit recessivo nel campo ACK del D.F.attuale

In assenza di almeno un segnale di conferma il segnale viene ritrasmesso.

REMOTE Frame



- È simile al Data Frame, ma non contiene dati ed ha un RTR recessivo
- È usato per sollecitare l'invio di un dato da parte di un data frame con lo stesso identificativo

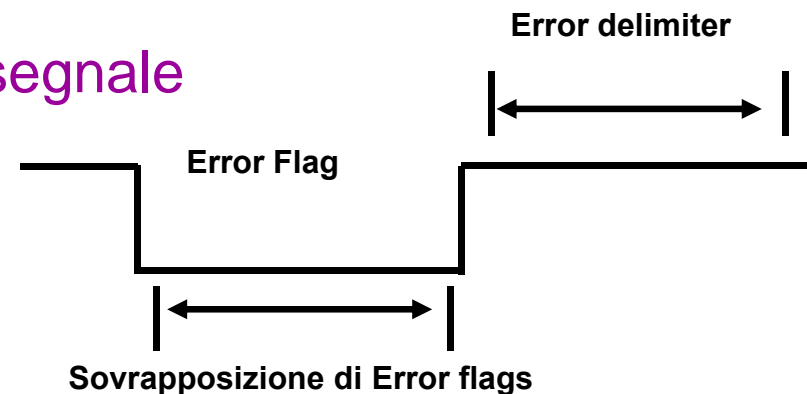
OVERLOAD Frame (presente nei chip più obsoleti)

- Serve a ritardare l'invio di dati successivo nel caso che un nodo sia impegnato da elaborazioni precedenti e non possa gestire altri dati.
- È costituito da 6 bit dominanti (che interrompono il traffico) seguiti da 8 bit recessivi (come per l'Error Frame)
- Un nodo non può inviare più di 2 Overload successivi.

ERROR Frame

- L' **Error flag** è inviato dal nodo che rileva un errore e comprende 6 bit dello stesso livello di tipo dominante o recessivo a seconda dell'effetto voluto sul traffico. Se i bit sono dominanti bloccano il traffico e impediscono al messaggio errato di giungere a destinazione. Possono inviare Error Flag dominanti solo nodi "affidabili" cioè che non risultano statisticamente responsabili di un numero elevato di errori.
- L' **Error delimiter** comprende 8 bit di valore recessivo (consente la sovrapposizione di altri messaggi di errore: ad es. alla fine della trasmissione di un errore se ne può verificare un altro che allunga il numero dei bit di errore).

- Il nodo interessato ritrasmette il segnale



Modalità Broadcast

Tutti i nodi sono in ascolto e recepiscono tutti i messaggi.

Non si può spedire un messaggio a un nodo specifico.

Sono previsti filtri hardware sui nodi per escludere dati non utili

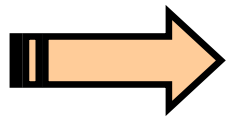
I messaggi sono brevi (non più di 94 bit utili)

Non è previsto un campo di indirizzamento

L'indirizzo può essere implicito nel contenuto del messaggio trasmesso

Il meccanismo di accesso al bus CSMA/BA

- Ogni nodo che vuole intervenire invia sul bus liberamente l'Arbitration Field. In caso di contesa prevale quello con bit 0 (cioè con identificativo più basso) che continua a trasmettere



- Ogni dispositivo con nodo CAN deve essere identificato **in maniera univoca**

Priorità e indirizzamento

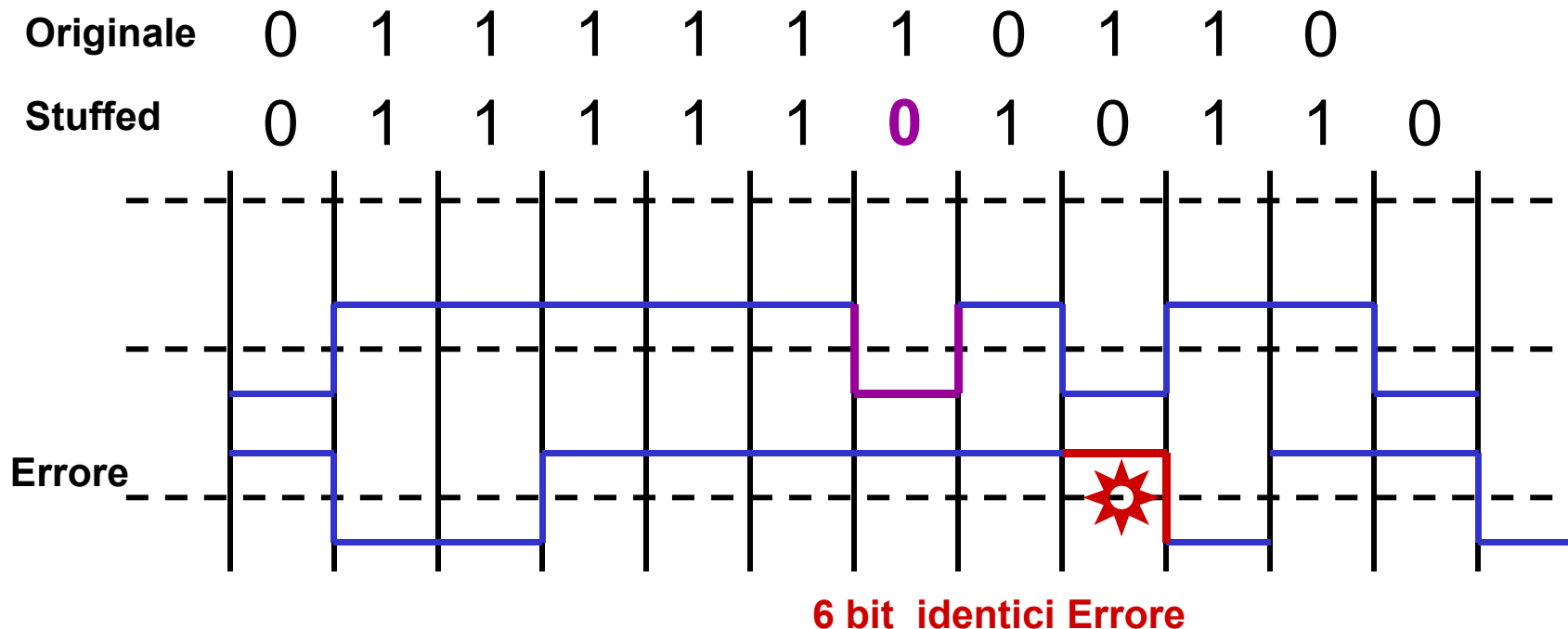
- **L'Arbitration Field** determina la priorità, non è un identificativo del dispositivo, anche se può essere usato come tale. Dipende dai livelli superiori. Lo standard SAE J1939 (destinato ai veicoli) prevede la definizione di parte del campo come Destination Address (NB Il source address è implicito data la natura di broadcast dovuta alla modalità Carrier Sensing.)
- Il suo valore può essere contenuto all'interno del data field del messaggio e come tale può essere filtrato e utilizzato come segnale di interesse per quel nodo

CAN - Codifica

La codifica è NRZ con *bit stuffing* (inserzione di un bit di valore opposto per interrompere una sequenza di 5 bit identici; il bit di stuffing viene automaticamente ignorato).

Il livello logico 0 è solitamente dominante.

L'allacciamento dei nodi è di tipo AND: se un unico nodo trasmette un bit dominante, tutto il bus è a quel livello indipendentemente dagli altri nodi



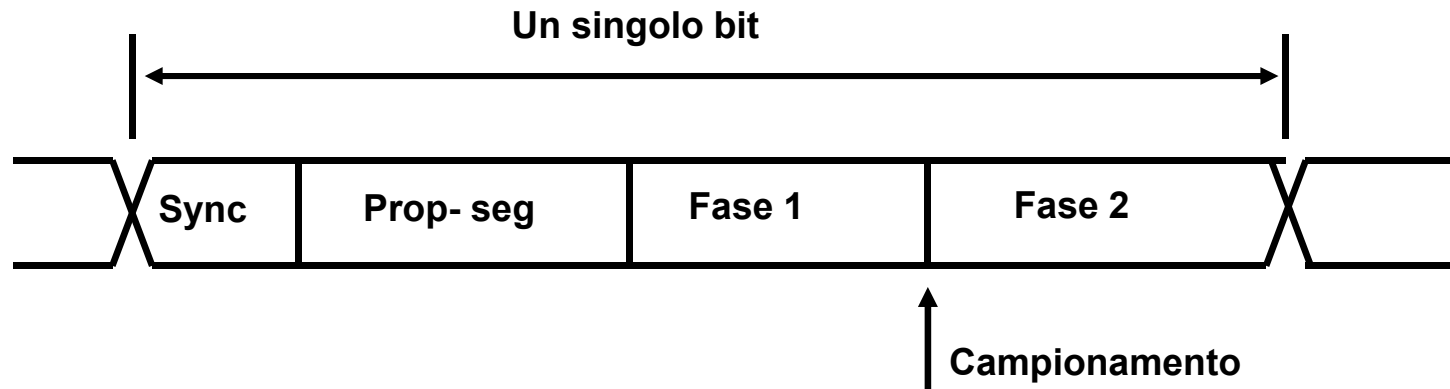
CAN – Temporizzazione del bit

La velocità di trasmissione dei dati sul bus è determinata dal **Nominal Bit Rate (NBR)**, definito come il numero di bit trasmessi al secondo in una trasmissione ideale; il suo inverso è il Nominal Bit Timing, che rappresenta il tempo occupato da un singolo bit ed è il reciproco del NBR.

Il NBT può essere diviso in 4 intervalli temporali di durata regolabile, espressa come multiplo di un quanto di tempo (TIME QUANTUM). Il **TIME QUANTUM** è a sua volta regolabile come multiplo dell'intervallo di un oscillatore locale, assunto come quanto minimo di tempo: **TIME QUANTUM = m * MINIMUM TIME QUANTUM**

Poiché il clock dei singoli nodi è locale, è possibile che sia necessaria una risincronizzazione dei vari nodi per tenere conto di eventuali differenze. Per questo scopo è prevista l'impiego di quantum variabili.

CAN – Temporizzazione del bit

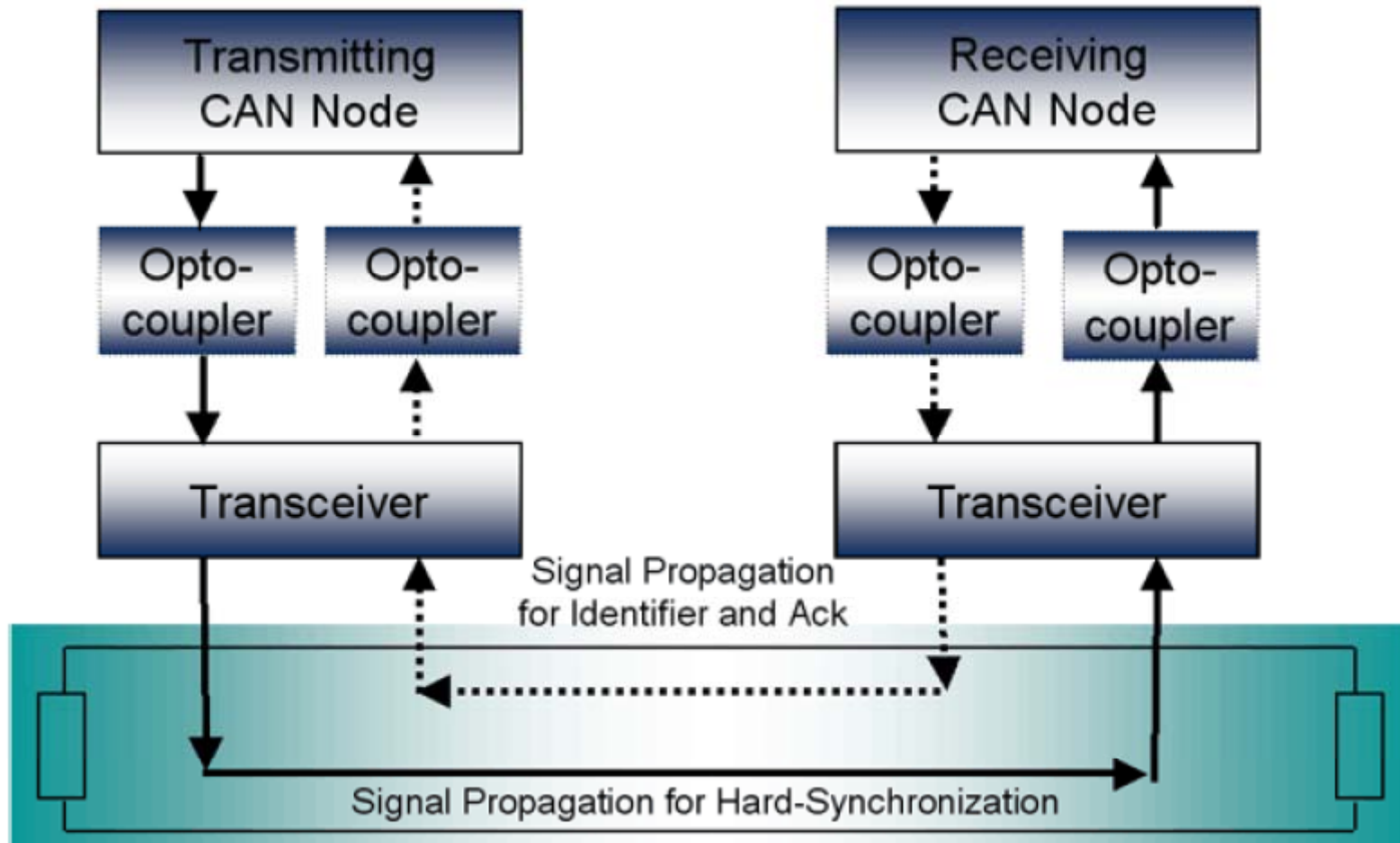


Sync – durata fissa (1 quantum). Segnale utilizzato per sincronizzare i vari nodi sul fronte di commutazione dal livello recessivo a quello dominante

Prop-seg – serve a compensare i ritardi di propagazione sulla linea. Può durare da 1 a 8 quantum.

Fase 1 e 2 – possono essere adattati in lunghezza per rispettare il tempo di clock. Tra la fase 1 (da 1 a 8 quantum) e la fase 2 (variabile) avviene la lettura del livello.

CAN – Temporizzazione del bit



CAN – Temporizzazione del bit

It is necessary to compensate for signal propagation delays on the bus line and through the electronic interface circuits of the bus nodes. The sum of the propagation delay times of controller, optional galvanic isolation, transceiver and bus line has to be less than the length of the Propagation Time Segment (Prop_Seg) within one Bit.

You have to add up the following delays depending on the selected components: CAN controller (50 ns to 62 ns), optocoupler (40 ns to 140 ns), transceiver (120 ns to 250 ns), and cable (about 5 ns/m).

These delays have to be considered twice, because after hard synchronization the most far away node is expect switching edges with delay of the propagation time, and the bit of the transmitter has to wait another propagation time to guarantee that the identifier bit or the Acknowledge slot bit of the Receiver is valid. Using ISO 11898 compliant transceiver and high-speed optocoupler you can reach a maximum bus length of 9 meters at 1 Mbit/s.

$t_{\text{propagation}} = 2 (t_{\text{cable}} + t_{\text{controller}} + t_{\text{optocoupler}} + t_{\text{transceiver}})$

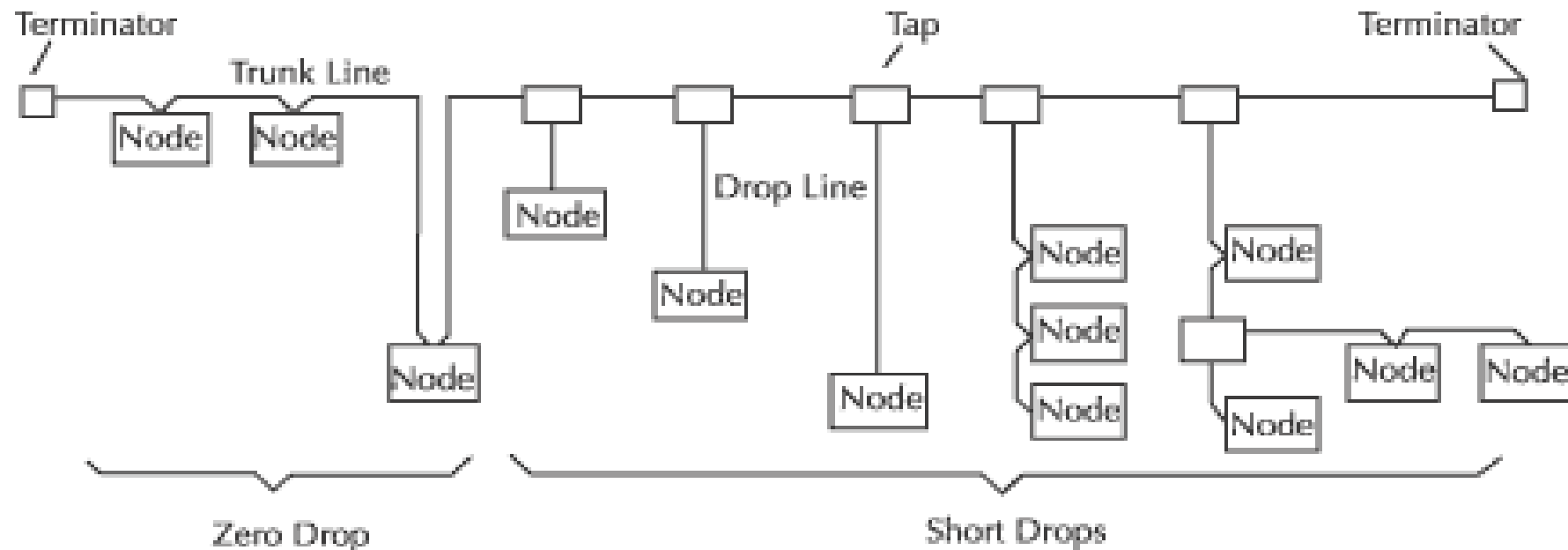
Estensioni di CAN ai livelli superiori

- Esistono diversi protocolli completi basati su CAN Bus
 - DeviceNet
 - CANopen
 - CANKingdom

DeviceNet

Network Size	Up to 64 Nodes
Network Length	Selectable end-to-end network distance varies with speed 125 kbps 500 m (1,640 ft) 250 kbps 250 m (820 ft) 500 kbps 100 m (328 ft)
Data Packets	0-8 bytes
Bus Topology	Linear (trunkline/dropline); power and signal on the same network cable
Bus Addressing	Peer-to-Peer with Multi-Cast (one-to-many); Multi-Master and Master/Slave special case; polled of change-of-state (exception-based)

DeviceNet



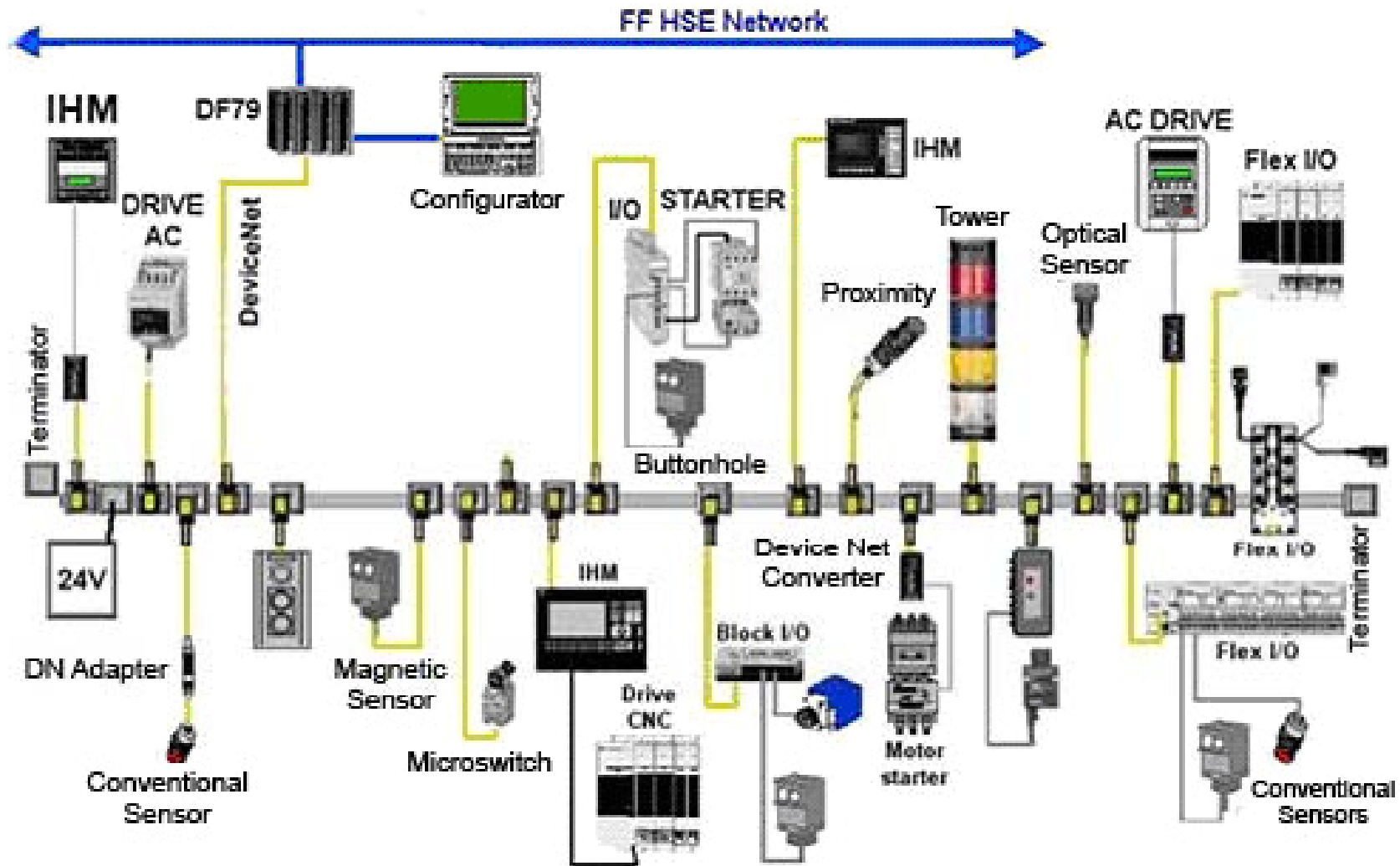
Livello fisico (Specifiche CAN)

Livello Data Link (Specifiche CAN)

Livello rete e trasporto (Modalità di comun. Connessa)

Livello Applicazione CIP (Common Industrial Protocol)

DeviceNet



DeviceNet: Object based

Ogni dispositivo DeviceNet è definito da proprietà in forma di Oggetti secondo lo standard CIP (Communications and Information Protocol)

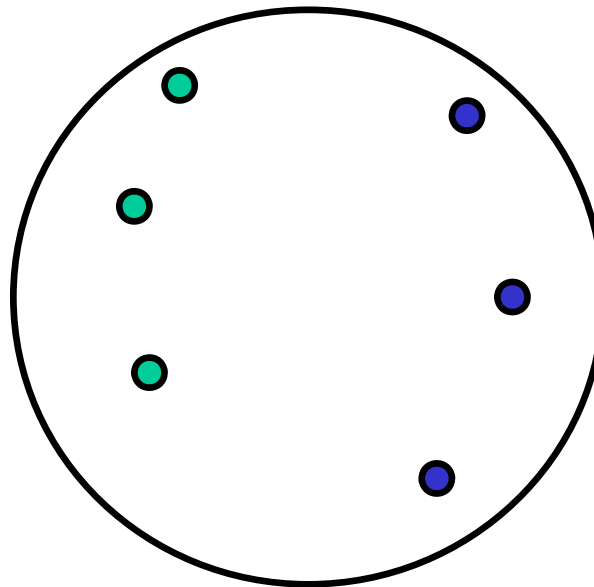
Oggetti obbligatori (Required Objects)

Identity Objects (Id. del fornitore, Serial N., dati del costruttore)

DeviceNet Objects (Baud Rate, N. Porta, MAC ID (Indirizzo DeviceNet, ...))

Connection Objects
(Attributi per la connessione esplicita o ciclica I/O dei dati, frequenza di trasmissione, percorsi, ...)

Message Router Objects
(Identificano i percorsi (target) di trasmissione)



**Dispositivo
DeviceNet**

Oggetti dell'applicazione (Application Objects)

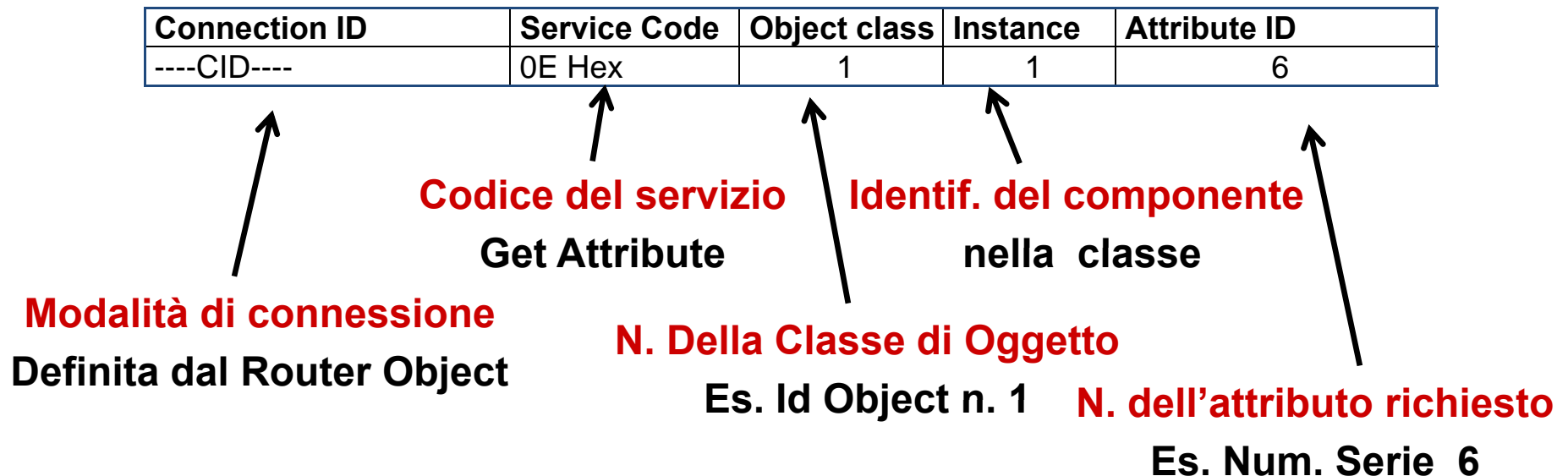
Oggetto Azionamento
(frequenza, corrente nominale, dati di targa del motore)

Oggetto I/O Analogico
(n. canali, risoluzione, livelli)

DeviceNet: Trasmissione dei messaggi

Explicit Messaging, sono messaggi di richiesta/risposta da un master a uno slave (Target) contenenti un codice di operazione e il target a cui si riferiscono.

Es. Per ottenere un attributo (Get Attribute) di un componente slave (es. il Numero di serie) si utilizza un messaggio esplicito:



DeviceNet: Trasmissione dei messaggi

I/O Messaging, sono tipi di messaggi generalmente impliciti fra un master e uno slave in cui la tipologia e frequenza dei dati è predefinita (ad es. usando un Explicit Message).

POLLING – Lo Scanner (Master) invia la richiesta di trasmissione a ciascuno slave con una frequenza di sua scelta.

CYCLIC - Un componente Slave trasmette un messaggio al suo Master ad una frequenza predefinita.

Change Of State (COS) - La trasmissione avviene solo al verificarsi di un evento (es. Un cambio di stato)

DeviceNet: Allocazione di uno Slave da parte di un Master (PLC/PC)

1. Il Master invia un messaggio Unconnected Open Request (tipo di messaggio che non prevede una connessione preliminare) indirizzato allo Slave che vuole asservire (es. Un sensore o un azionamento).
2. Lo Slave può rispondere accettando la richiesta e stabilendo una connessione in cui sono specificati indirizzi e attributi della connessione. (Uno Slave potrebbe non rispondere perché non è abilitato o ha già un altro Master. Un solo master per ogni Slave).
3. Utilizzando il canale aperto il Master configura I parametri della comunicazione che da quel momento diventerà attiva (frequenza, modalità, ecc.)

Esempio di applicazione a un azionamento

Si considera un azionamento basato su inverter IGBT, con modulazione vettoriale per la regolazione di motori asincroni fino a 132 kW

Funzioni dell'azionamento che implicano uno scambio di segnali:

- **Caratteristica V/f programmabile**
- **Compensazione dello scorrimento basato sulla corrente di carico**
- **Protezione termica basata sul monitoraggio della corrente**
- **Filtraggio delle frequenze per evitare risonanze meccaniche.**

Può essere utilizzato da solo o in configurazione multidrive per coordinare le operazioni in sincrono su celle di lavoro

La connessione è CAN in alternativa a una connessione seriale standard per le operazioni stand alone

Segnali tipici scambiati in un azionamento

Segnali di allarme (digital out on/off) per segnalazioni visive e acustiche

Start/Stop e comandi di emergenza (digital in/out off)

Scambio di dati interno al microcontrollore (sincronizzazione, interrupt)

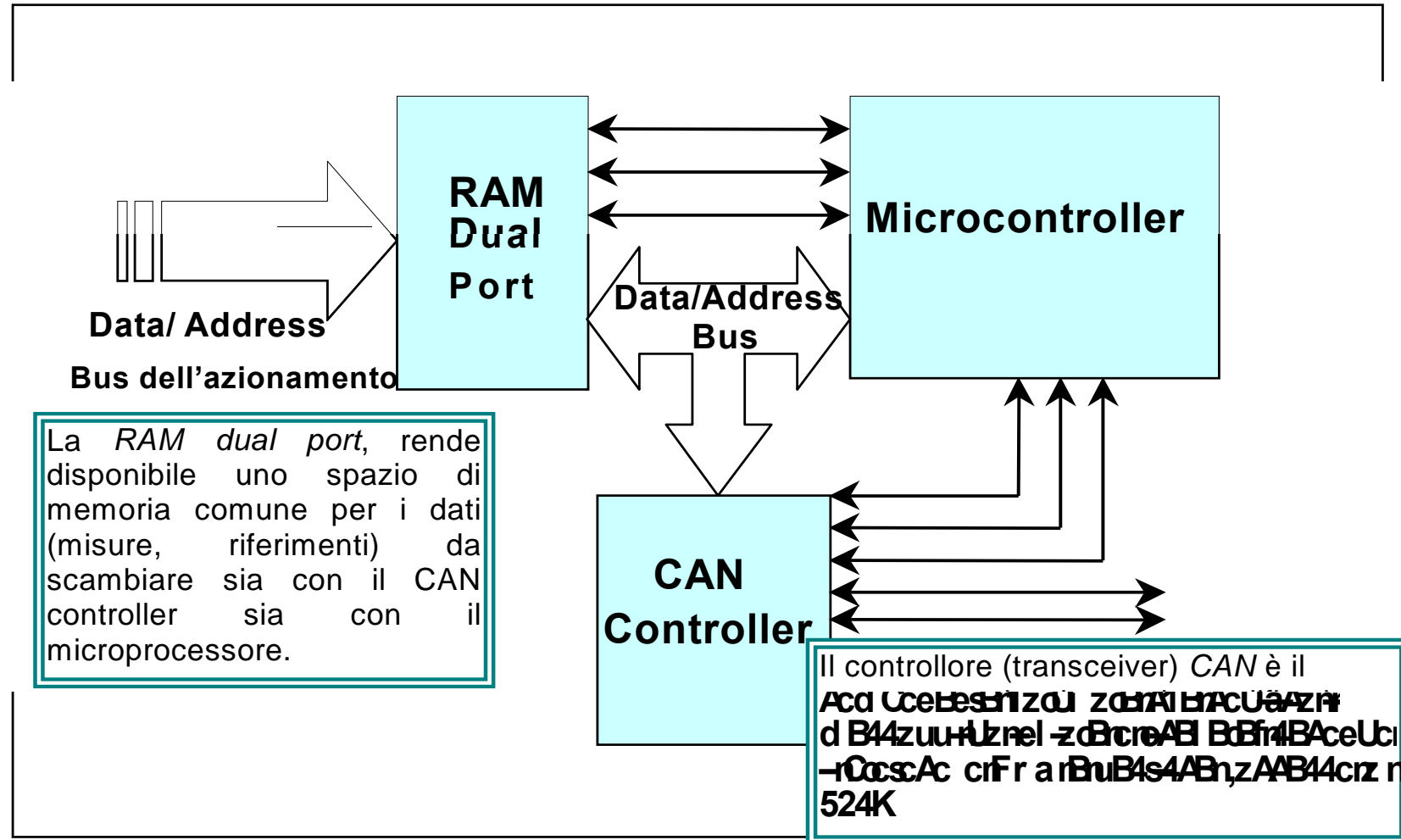
Segnali di riferimento (velocità, coppia, posizione) generati da sorgenti analogiche (sensori, manopole, resistori variabili) o digitali (tastiere, interfacce PC)

Segnali analogici o digitali provenienti da sensori necessari per la visualizzazione o il controllo

Misure disponibili all'azionamento con i tempi di aggiornamento

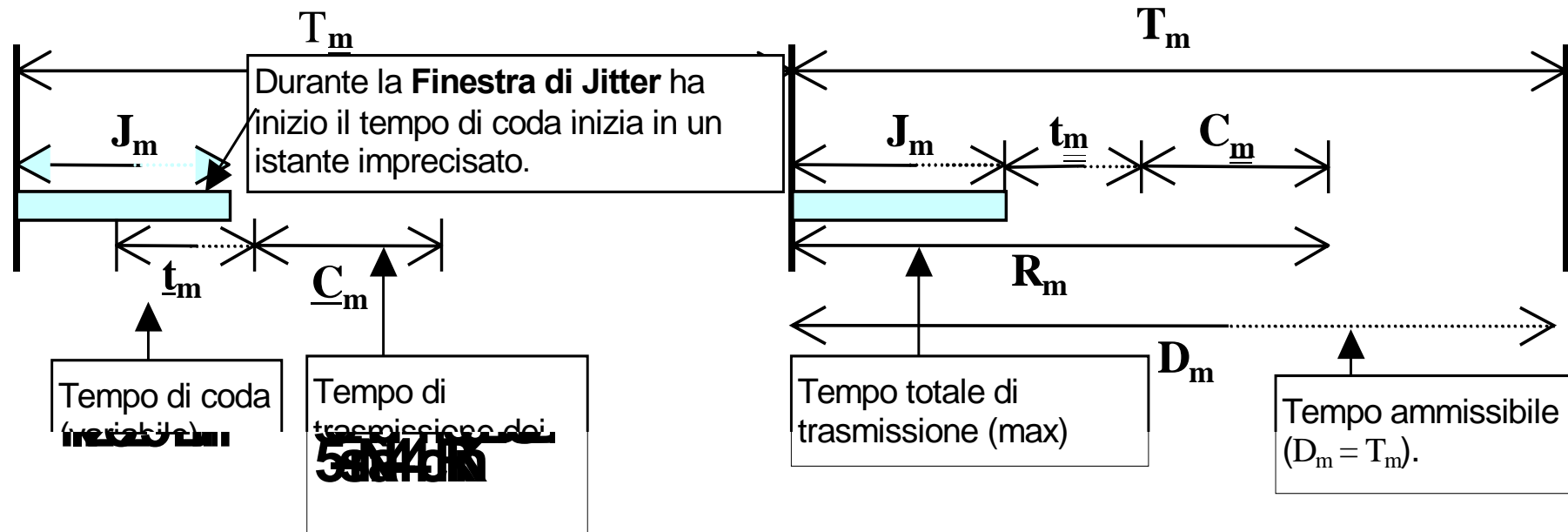
Message	D_m	Message	D_m
Speed reference	5 ms	Rated Motor Voltage	50 ms
Measured frequency	5 ms	Rated Motor Current	50 ms
Set Configuration Bit	5 ms	No-load Motor Current	50 ms
Clear Configuration Bit	5 ms	Setup Thermal Protection	50 ms
Measured current	10 ms	Motor Thermal Constant	50 ms
Measured voltage	10 ms	Motor Compensating Torque	50 ms
Current reference (4/20ma)	10 ms	Rated Motor Slip	50 ms
Configuration Word	20 ms	Set Limit. in acceleration	50 ms
Speed Reference Range	50 ms	Set Limit. at const. freq.	50 ms
Reference/Output Gain	50 ms	Max Current in acceleration	50 ms
Acceleration time	50 ms	Max Current at const. freq.	50 ms
Deceleration time	50 ms	Maximum Motor Frequency	50 ms
Rated Motor Frequency	50 ms	Minimum Motor Frequency	50 ms

CAN – Interfaccia azionamento – bus di campo



a

CAN – Temporizzazione della trasmissione



Tempi di trasmissione

J_m *jitter time* è il tempo di attesa prima che il messaggio programmato sia effettivamente gestito dal controllore. Il tempo di jitter (legato alle operazioni del microprocessore) non è completamente prevedibile ed è limitato superiormente da un valore massimo J_m

t_m *queuing time* è il tempo di attesa in coda legato alle priorità previste e implicite nel protocollo CAN. È diviso in due ulteriori tempi:

I_m necessario per completare I messaggi a priorità superiore;

B_m necessario per completare I messaggi a priorità inferiore, ma già in corso:

$$B_m = \max_{\forall k \in lp(m)} (C_k)$$

(il più lungo dei messaggi con priorità inferiore)

Tempi di trasmissione: calcolo di I_m

$$I_m = \sum_{\forall j \in hp(m)} \left[\frac{t + J_j + \tau_{bit}}{T_j} \right] C_j$$

hp(m) insieme dei messaggi a priorità più elevata

τ_{bit} il tempo richiesto per la trasmissione di un bit

t è il tempo durante il quale si calcola la sommatoria dei tempi dei singoli messaggi a priorità superiore. Poiché dipende dalla stessa coda che si sta calcolando la sua valutazione si svolge in forma ricorsiva:

$$t_m^{n+1} = B_m + \sum_{\forall j \in hp(m)} \left[\frac{t^n + J_j + \tau_{bit}}{T_j} \right] C_j$$

Funzione monotona crescente che inizia con $t_0=0$.

Il massimo tempo di risposta R_m : $R_m = J_m + t_m + C_m$.

Criteri di assegnazione delle priorità

I vari nodi hanno priorità legate all'urgenza dei tempi di ciclo D_m :
tempi di ciclo più brevi  priorità più elevate

Segnali periodici. (Riferimenti, misure) Richiedono un aggiornamento e lettura continui, per evitare che il nuovo valore ricopra il precedente.

Segnali occasionali. (Parametri, configurazione) Non hanno una scadenza specifica o urgente. Il tempo di ciclo è definito convenzionalmente in tempi ragionevoli di variazione dei valori. (es. 50 ms)

Sistemi multidrive. Se sono presenti più azionamenti con caratteristiche simili, si deve decidere un ordine di priorità convenzionale, al cui interno rimangono comunque privilegiati i segnali periodici, rispetto agli occasionali

Gestione degli errori (1)

In caso di errore il segnale deve essere ritrasmesso, preceduto da un Error Frame, per cui l'equazione viene modificata come segue

$$t_m = B_m + \sum_{\forall j \in hp(m)} \left[\frac{t_m + J_j + \tau_{bit}}{T_j} \right] C_j + E_m (t_m + C_m)$$

Il tempo aggiunto E_m è definito sulla base di un modello statistico. Sia n_{error} il *numero di errori previsti* in un intervallo generico e T_{error} l'*intervallo dell'errore residuo*, ovvero dopo che si sono verificati n_{error} un ulteriore errore non è previsto prima che passi T_{error} comunque con una frequenza non superiore a $1/T_{error}$.

Gestione degli errori (2)

Il massimo numero di errori previsti in un periodo è quindi:

$$N_{Merror} = n_{error} + \left[\frac{t}{T_{error}} \right] - 1$$

Il tempo necessario per rilevare e ritrasmettere un messaggio è:

$$n_{error_frame} \cdot \tau_{bit} + \max_{\forall k \in hp(m) \cup \{m\}} (C_k)$$

Dove n_{error_frame} è il numero di bit dell'error frame, e il secondo termine indica il massimo tempo di ritrasmissione per il messaggio m o con più elevata priorità.

Il tempo complessivo di gestione dell'errore è infine:

$$E_m(t) = \left(n_{error} + \left[\frac{t}{T_{error}} \right] - 1 \right) \left(n_{error_frame} \cdot \tau_{bit} + \max_{\forall k \in hp(m) \cup \{m\}} (C_k) \right)$$

Prestazioni del bus di campo per l'azionamento

La valutazione delle prestazioni è stata realizzata riportando in un foglio elettronico tutti parametri indicati con le rispettive priorità.

Per ogni parametro si calcola nelle colonne successive il tempo necessario alla trasmissione del dato utilizzando le formule indicate, e si confronta il tempo necessario con il D_m di ogni parametro.

Se il tempo di trasmissione è inferiore a D_m la trasmissione è riuscita.

L'analisi è ripetuta:

- per un singolo azionamento;
- per tre azionamenti in multidrive;
- considerando la gestione dell'errore.

Simulazione. Tempi di trasmissione per alcune variabili

Prior.	Signal	T _m (ms)	D _m (ms)	I _m (ms)	C _m (ms)	B _m (ms)	R _m (ms)
1 Mbps							
1	Speed reference	5	5	0	0.06	0.07	0.13
2	Measured frequency	5	5	0.06	0.06	0.07	0.14
3	Clear Configuration Bit	50	5	0.12	0.06	0.07	0.20
4	Set Configuration Bit	50	5	0.18	0.06	0.07	0.31
5	Measured current	10	10	0.24	0.06	0.07	0.37
6	Measured voltage	10	10	0.3	0.06	0.07	0.43
99	Setup Thermal Protection	50	50	6.466	0.07	0.07	6.59
102	No-load Motor Current	50	50	6.656	0.06	0.07	6.77
125 kbps							
1	Speed reference	5	5	0	0.492	0.57	1.06
2	Measured frequency	5	5	0.492	0.492	0.57	1.55
3	Clear Configuration Bit	50	5	0.984	0.492	0.57	2.04
4	Set Configuration Bit	50	5	1.477	0.492	0.57	2.53
5	Measured current	10	10	1.969	0.492	0.57	3.03
6	Measured voltage	10	10	2.461	0.492	0.57	3.52
99	Setup Thermal Protection	50	50	52.969	0.492	0.57	54.03
102	No-load Motor Current	50	50	54.523	0.492	0.57	55.50

Simulazione. Percentuale di esiti positivi

Tutti i parametri sono trasmessi entro il tempo necessario per velocità pari a 250 kbps o superiori. A 125 kbps il 10% dei segnali (non prioritari) non è trasmesso in tempi accettabili.

La situazione degrada nel caso di 3 azionamenti.

Se si considera la gestione dell'errore l'effetto non cambia in modo significativo (per il caso di 1 azionamento).

Transmission speed	Single Drive	Three Drives	Error Management
1 Mbps	100%	100%	100%
500 Kbps	100%	90%	100%
250 Kbps	100%	43%	100%
125 Kbps	75%	20%	54%

Bibliografia

1. E. Bassi, F. Benzi, G.S. Buja, L. Luseti: *Communications Protocols for Electrical Drives*. Proceedings of IECON'95, Orlando (USA), November 1995
2. *Bosch Controller Area Network (CAN) Version 2.0 Protocol Standard*. Motorola, 1997
3. K. W. Tindell, A. Burns and A. J. Wellings: *Calculating Controller Area Network (CAN) Message Response Times*, Control Engineering Practice, Vol.3, No 8, pp 1163-1169, 1995.
4. N. C. Audsley, A. Burns, M. Richardson and A. J. Wellings: *Hard Real-Time Scheduling: The Deadline Monotonic Approach* - Proceedings of IEEE Workshop on Real-Time Operating Systems and Software, pp. 133-137, May 1991