

Reti di campo, Controller Area Network (CAN), CANopen



Davide Quaglia



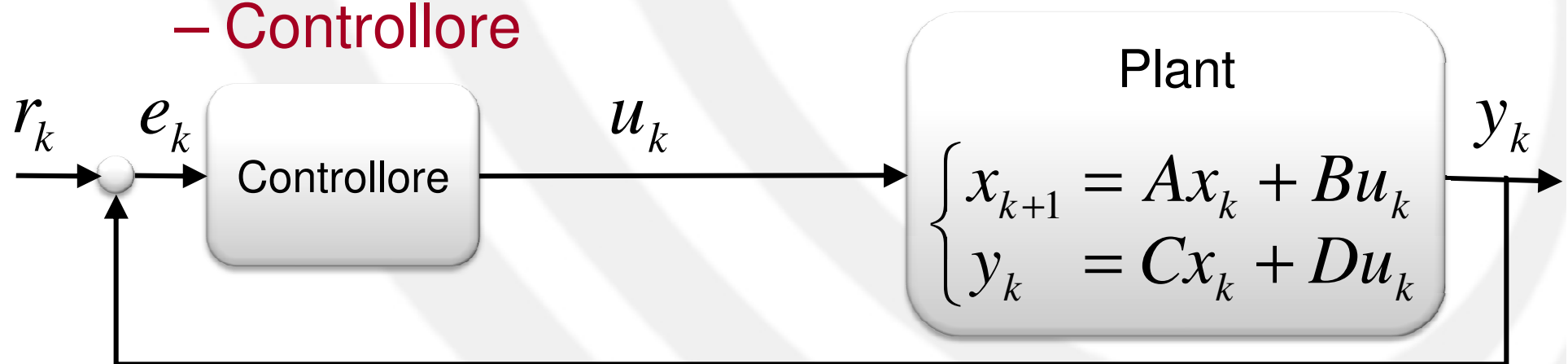
Sommario

- Reti di campo
- CAN
- CANopen

RETI DI CAMPO

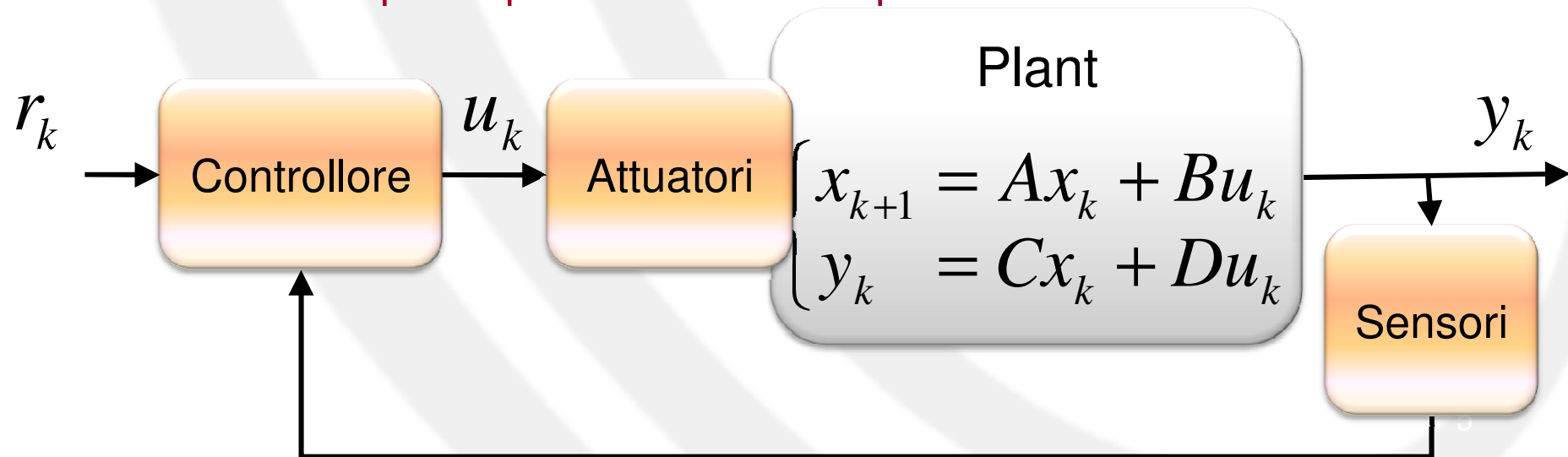
Sistemi di automazione

- Dal punto di vista della disciplina dei controlli automatici un sistema di automazione è costituito da
 - Sistema da controllare (Plant)
 - Controllore



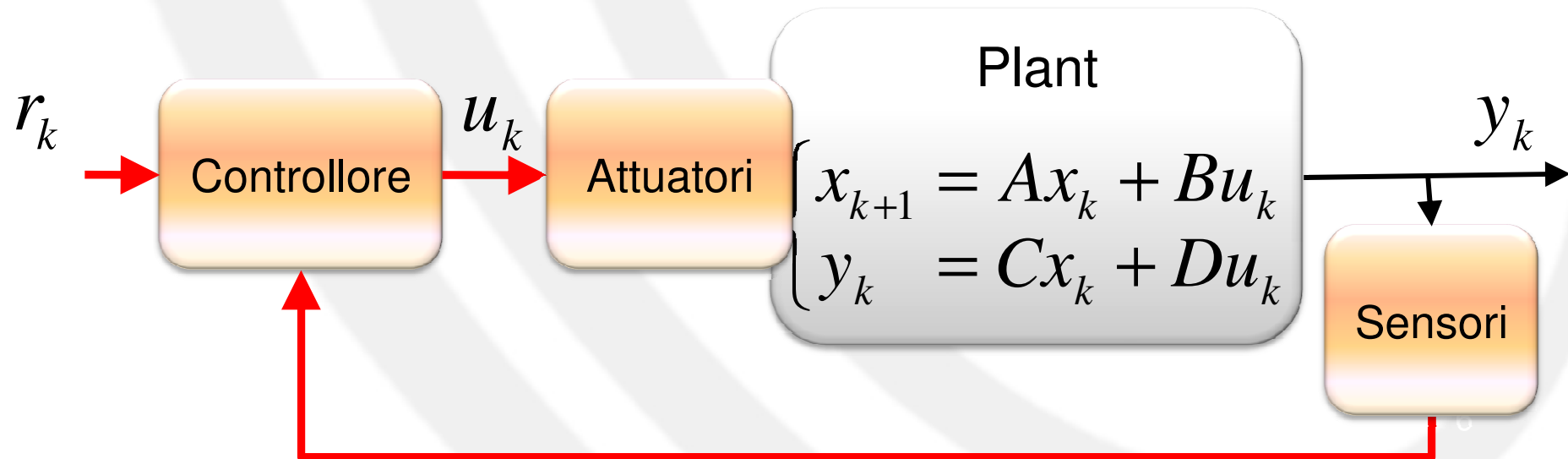
Architettura di un sistema di automazione

- Il plant è un sistema fisico che ci è dato
- Gli elementi costruiti dall'uomo sono:
 - Controllore = programma che implementa la strategia di controllo (calcola anche e_k)
 - Sensori per leggere informazioni dal plant
 - Attuatori per impartire comandi al plant



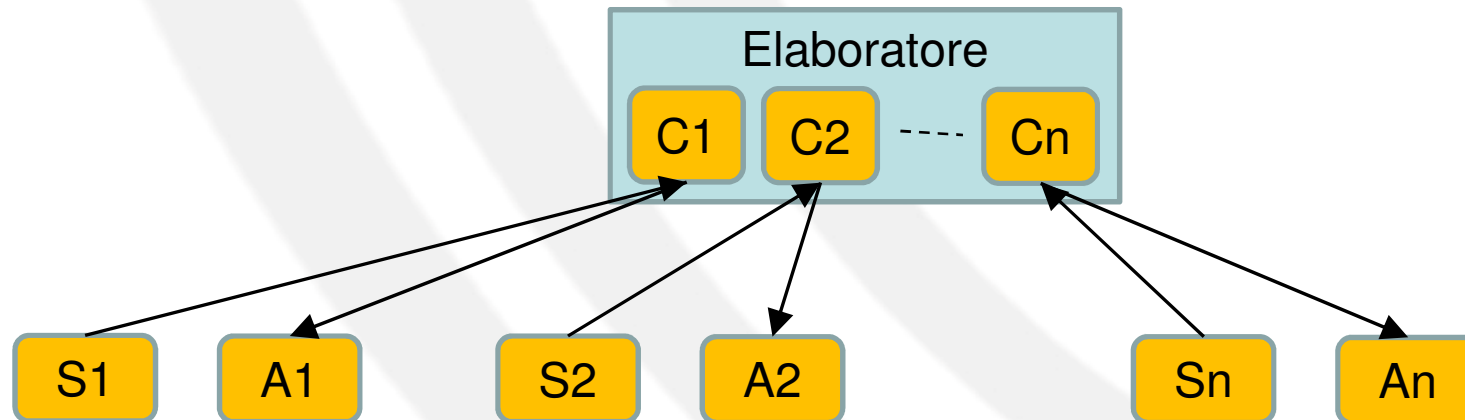
Comunicazioni un sistema di automazione

- Siccome gli elementi costruiti dall'uomo sono digitali essi sono collegati da linee di comunicazione digitale (in rosso in fig.)



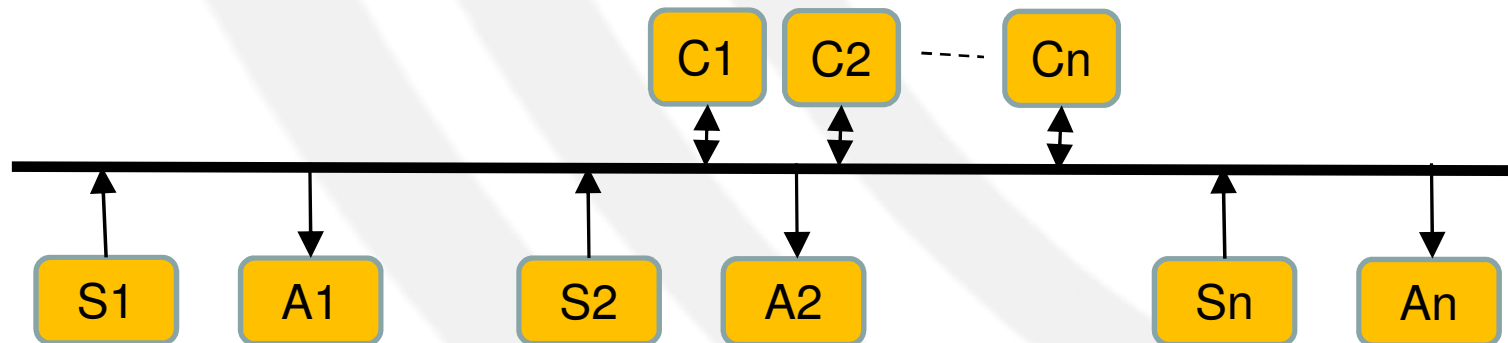
Prospettiva storica: ieri

- Alto costo dei processori
- I controllori di un sistema automatico erano tutti implementati in un unico elaboratore
- Sensori e attuatori molto semplici
- Linee seriali punto-punto tra elaboratore e sensori/attuatori



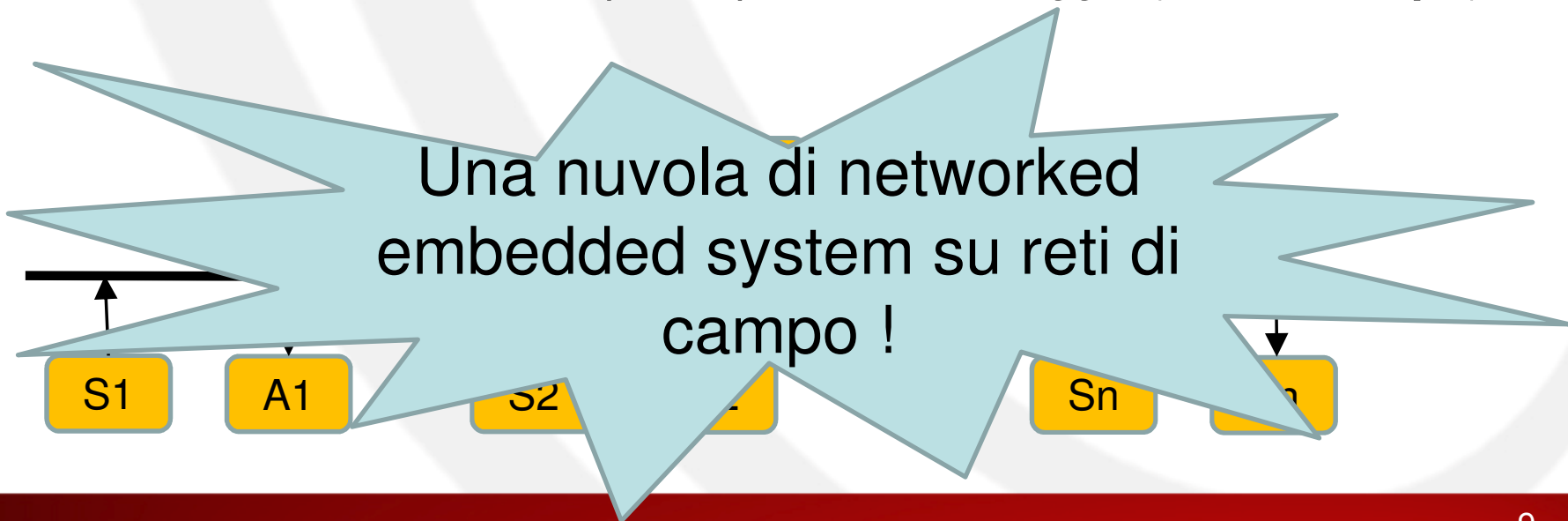
Prospettiva storica: oggi

- Microprocessori economici
- Incremento del numero di funzioni di controllo
- Architettura distribuita
 - Ciascun controllore su un sistema embedded dedicato
 - Sensori e attuatori possono svolgere alcune elaborazioni accessorie
- Unico canale condiviso per risparmiare cablaggio (**rete di campo**)



Prospettiva storica: oggi

- Microprocessori economici
- Incremento del numero di funzioni di controllo
- Architettura distribuita
 - Ciascun controllore su un sistema embedded dedicato
 - Sensori e attuatori possono svolgere alcune elaborazioni accessorie
- Unico canale condiviso per risparmiare cablaggio (**rete di campo**)



Reti office e reti di campo

- Office (Internet)
 - Grandi quantità di informazioni in minuti
 - Accesso al canale fisico di tipo non deterministico
 - Poca probabilità di errore fisico (tranne se wireless) → affidabilità affidata ai livelli alti (TCP)
- Campo
 - Pochi bit (segnali) in microsecondi
 - Bassa latenza (possibilmente deterministica) di accesso al canale
 - Canale rumoroso e applicazioni critiche → affidabilità gestita già dal livello fisico

Requisiti degli standard per reti di campo

- Protocolli semplici per non gravare su processore e memoria dei sistemi embedded
- Basso spreco di bit sul canale
 - Pochi tipi di pacchetti e interazioni semplificate
 - Solo liv. OSI-1 e OSI-2 (talvolta OSI-7)
- Affidabilità
 - Valori di comandi o misure alterate possono creare situazioni di pericolo

Requisiti degli standard per reti di campo (2)

- Politiche di accesso al canale
 - Bassa latenza
 - Basso o nullo non-determinismo
 - In certi controlli non serve un ritardo d'anello estremamente basso ma costante
- Diverse priorità di trasmissione
 - Diversi sistemi da controllare con diversi livelli di criticità e di requisiti temporali

Parte del materiale è stato prodotto da
Daniele Gallinella – Università degli Studi Roma3

CONTROLLER AREA NETWORK (CAN)

Controller Area Network (CAN)

- Bus seriale di comunicazione per applicazioni real-time
- Sviluppato da Bosch nel 1986 per Mercedes
- Facilità di configurazione e modifica
- Velocità di interfaccia fisica fino a 1 Mb/s
- Affidabilità
 - Funzionamento in ambienti ostili
 - Rilevamento automatico degli errori di trasmissione

Architettura dei protocolli

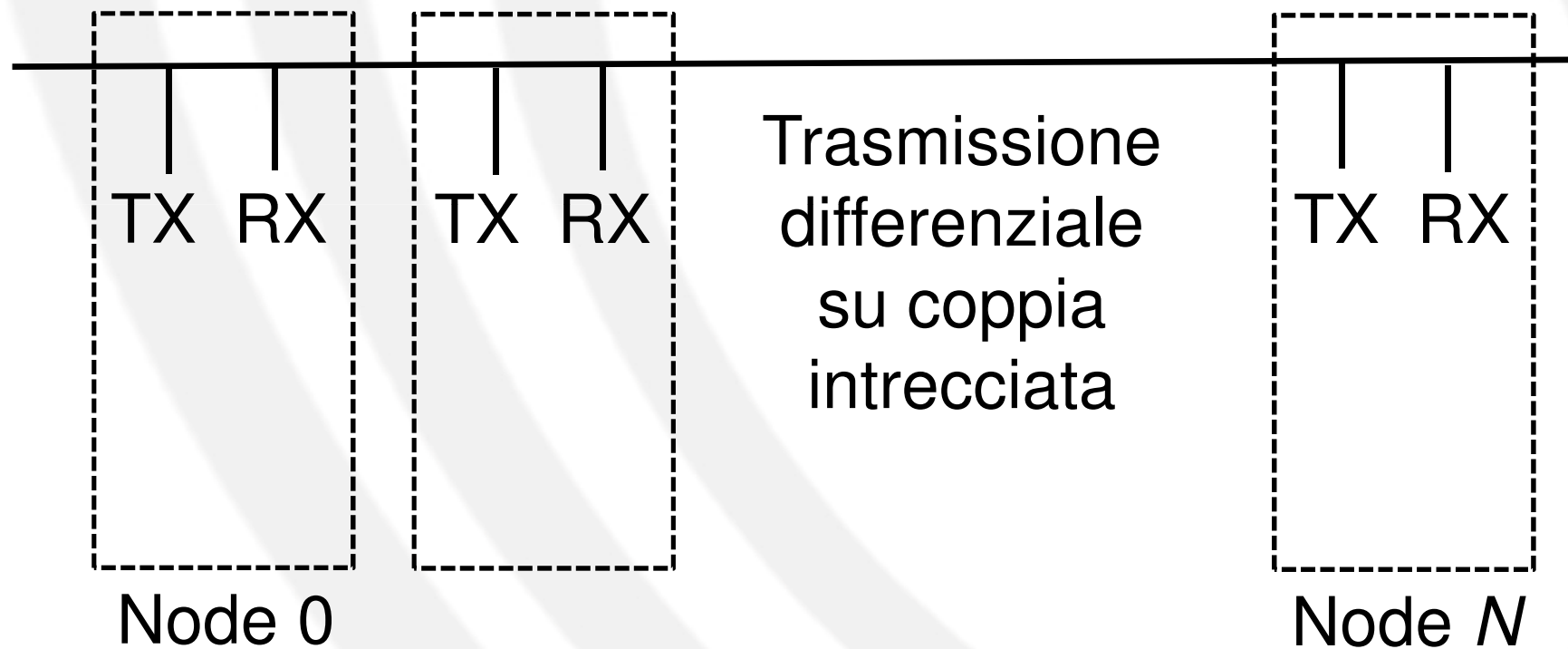
- Livello fisico
 - Cablaggi
 - Livelli di tensione elettrica
- Livello datalink
 - Politica di accesso al canale
 - Indirizzamento
 - Tipi di formato di pacchetti
 - Rilevazione errori
- Livelli superiori
 - Solo il livello applicazione chiamato CANopen

LIVELLO FISICO

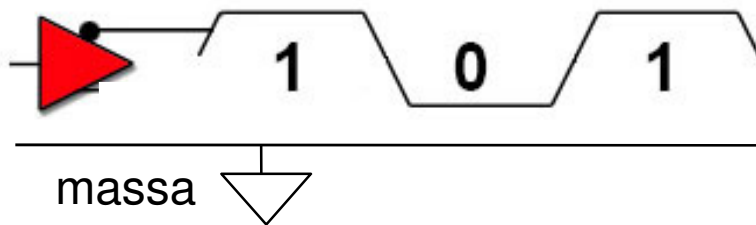
Mezzo trasmissivo

- Doppino intrecciato (schermatura opzionale) e trasmissione differenziale o bilanciata
- Tutte le stazioni si agganciano allo stesso cavo
- Impedenza di terminazione 120 ohm
- Lunghezza massima del cavo dipende dalla velocità di trasmissione scelta
 - Occorre che tutti possano sentire un bit scritto da un nodo prima che questo ne termini la scrittura

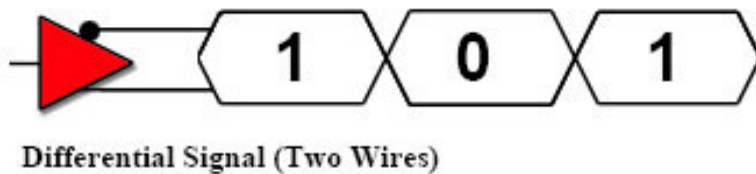
Topologia fisica



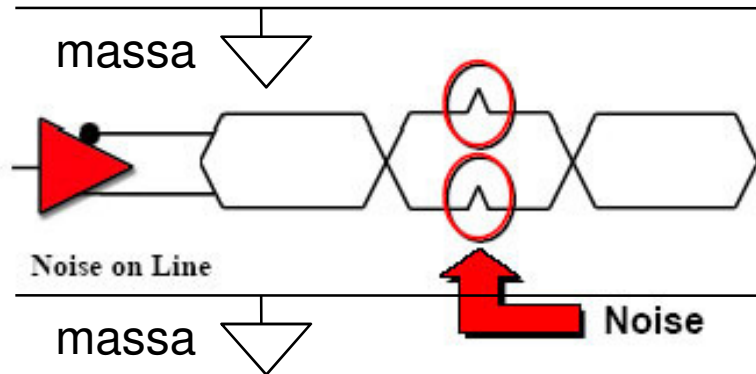
Trasmissione sbilanciata vs. bilanciata o differenziale



Trasmissione sbilanciata



Trasmissione differenziale



Esempio di annullamento dell' effetto dell'interferenza

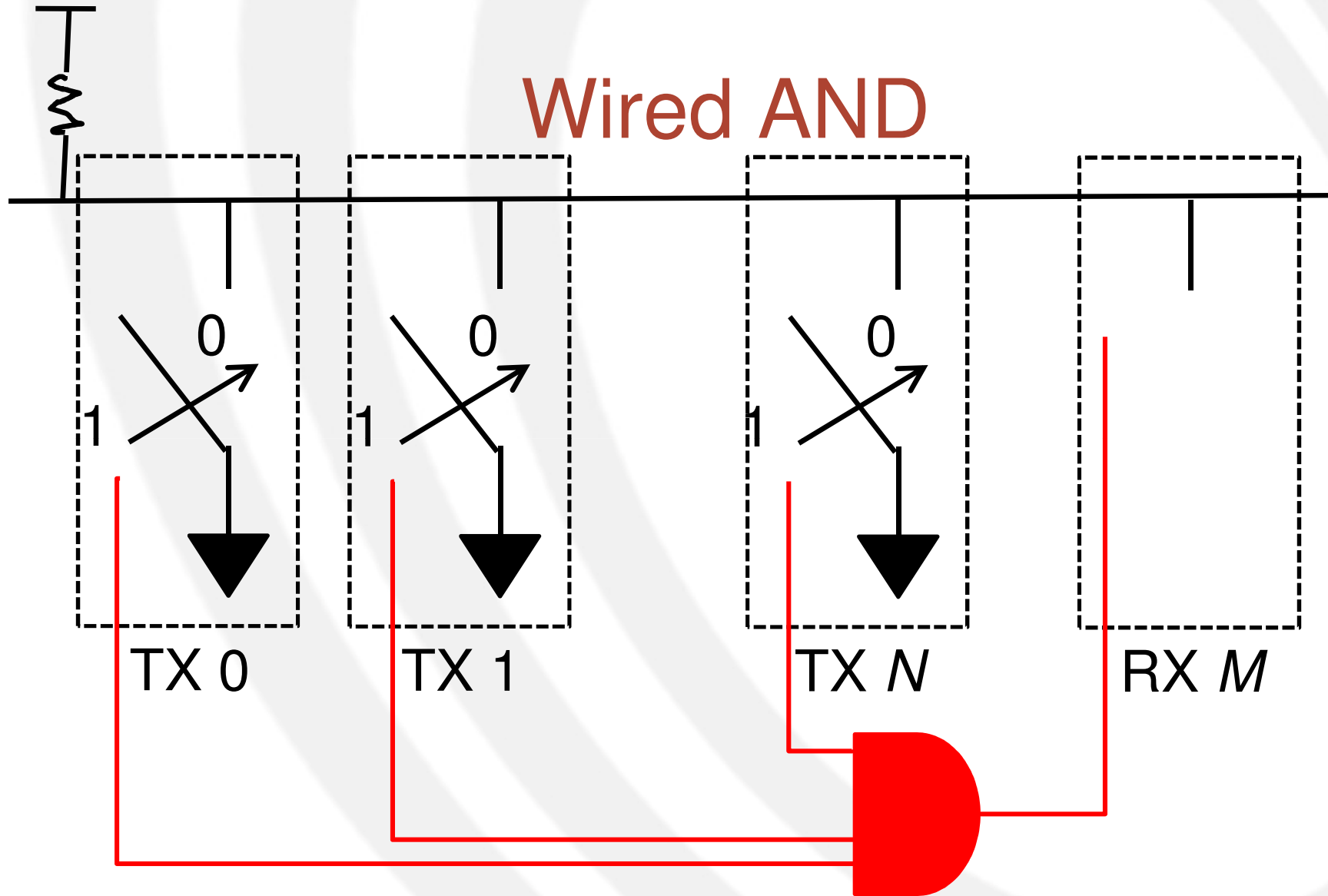
Relazione lunghezza cavo e velocità

Lunghezza del bus (metri)	Max bit rate (bit/sec)
40	1 Mbit/s
100	500 Kbit/s
200	250 Kbit/sec
500	125 Kbit/s
6K	10 Kbit/s

Scrittura sul canale

- Tutti i nodi possono scrivere sul canale
- Sovrapposizione di bit (collisione)
- Meccanismo del *wired AND*
- lo stato dominante (0 logico) sovrascrive lo stato recessivo (1 logico).

Wired AND



LIVELLO DATA LINK

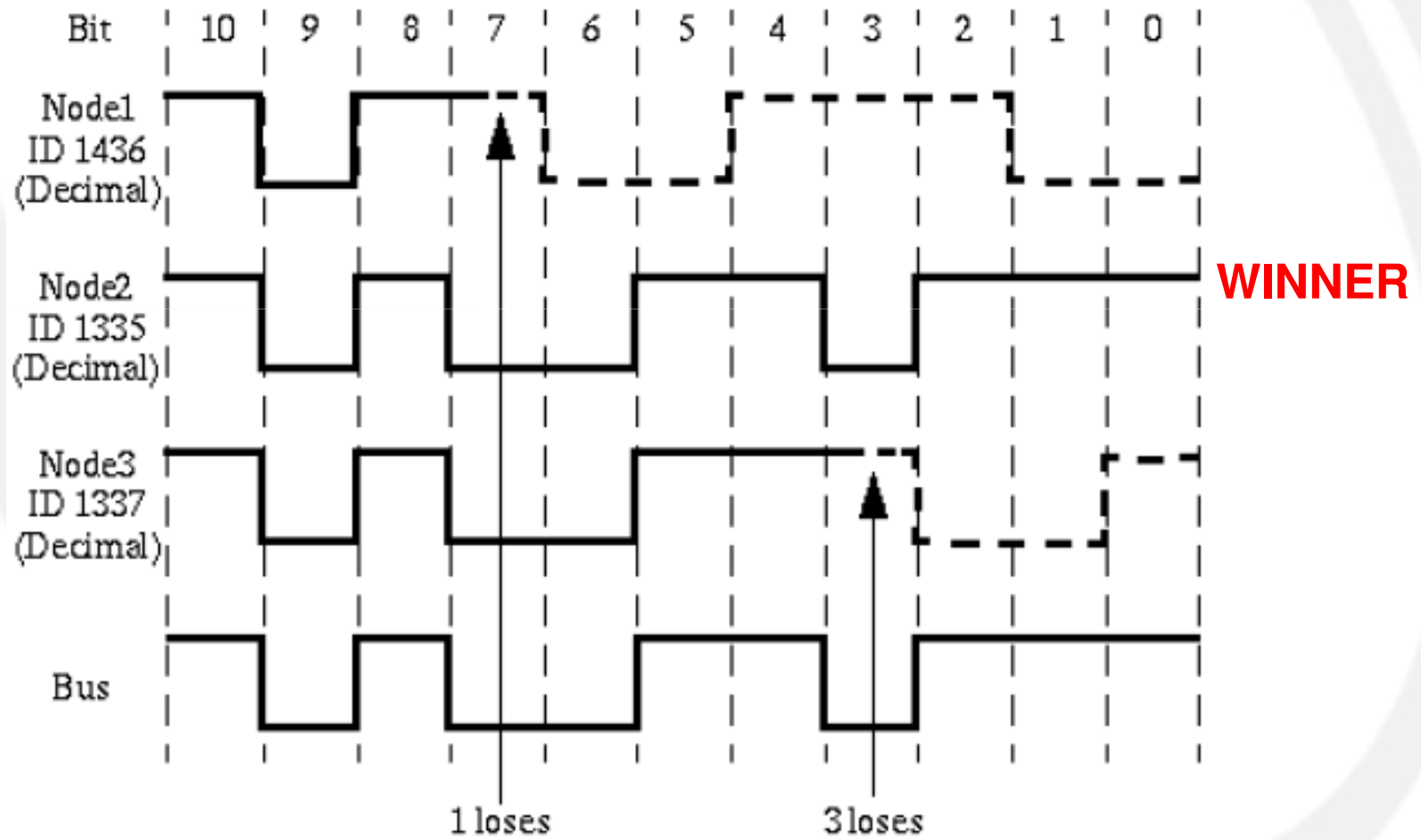
Politica di accesso del canale

- CSMA/CD modificato
- Chi deve trasmettere prima ascolta il canale
 - Se occupato ne aspetta la liberazione
 - Se libero (idle) allora inizia a trasmettere
- Durante la trasmissione la stazione continua ad ascoltare il canale
 - Quando legge un bit diverso da quello scritto smette di trasmettere

Politica di accesso del canale (2)

- Non si crea corruzione di bit perché
 - Il primo campo scritto sul canale è l'identificatore del messaggio (11 o 29 bit)
 - Si comincia a scrivere dal bit più significativo
 - Il meccanismo del wired AND preserva l'identificatore con valore più basso
- Vantaggi
 - Assenza di tempi morti dovuti ad attese casuali (che aumentano latenza e non-determinismo)
 - si crea una trasmissione con priorità

Esempio di arbitraggio del canale



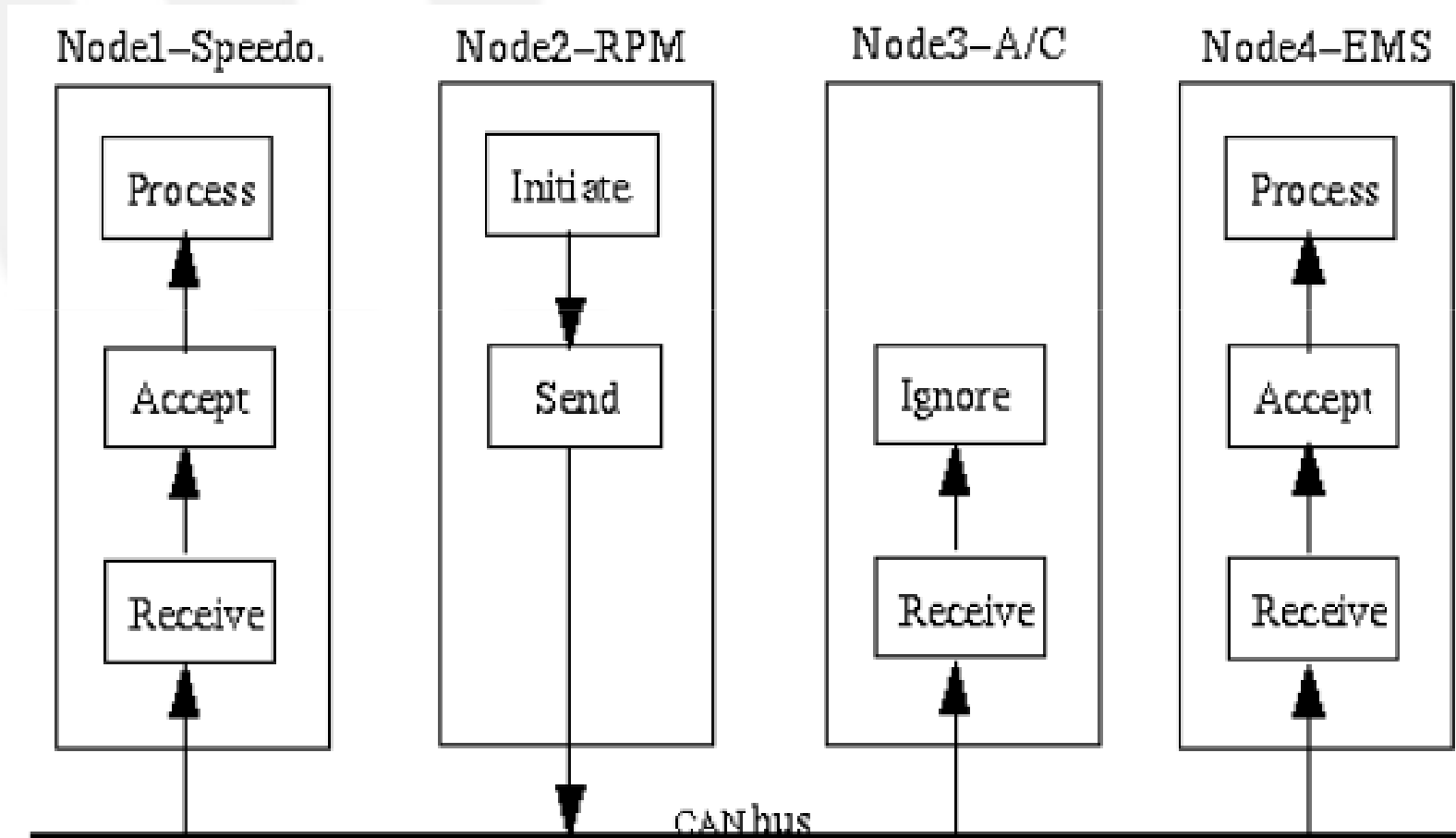
Indirizzamento

- Assenza di indirizzi mittente/destinatario
- Data-driven addressing
 - Ogni pacchetto di dati inizia con un identificatore del tipo di dati (es. giri/min, temperatura)
- Ogni potenziale ricevitore analizza l'identificatore e decide se continuare a leggere oppure no
- Approccio multicast

Indirizzamento (2)

- Vantaggi
 - nuove stazione possono essere aggiunte senza rischio di indirizzi duplicati
 - Se sono solo ricevitori non devo definire nuovi identificatori e quindi non serve modificare il SW degli altri nodi
 - Nessun meccanismo run time di assegnazione/gestione di indirizzi
 - Nessun frame duplicato in caso di trasmissione a più destinatari
- Svantaggi
 - Non ci deve essere più di 1 produttore per identificatore
 - Necessità di standardizzare gli identificatori (livello applicazione)

Esempio di trasmissione multicast



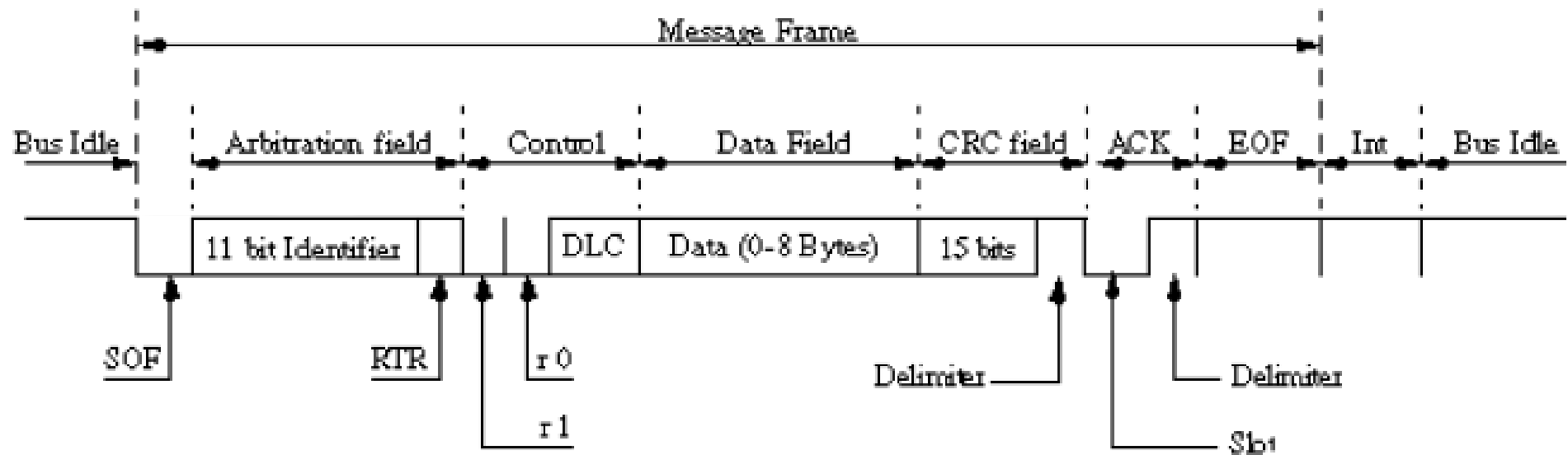
Tipi di frame

- 4 tipi di messaggi (frame)
 - Data frame
 - Remote frame
 - Error frame
 - Overload frame

Data/remote frame

- Data frame e remote frame hanno la stessa struttura eccetto per il payload assente nel remote frame.
- Il data frame serve per trasmettere informazioni relative ad un certo identificatore (es. giri/min)
- Il remote frame è la richiesta da parte di un nodo del data frame corrispondente (avente lo stesso identificatore)
 - Information solicitation vs. periodic advertising

Formato del data/remote frame



Standard CAN (Versione 2.0 A)

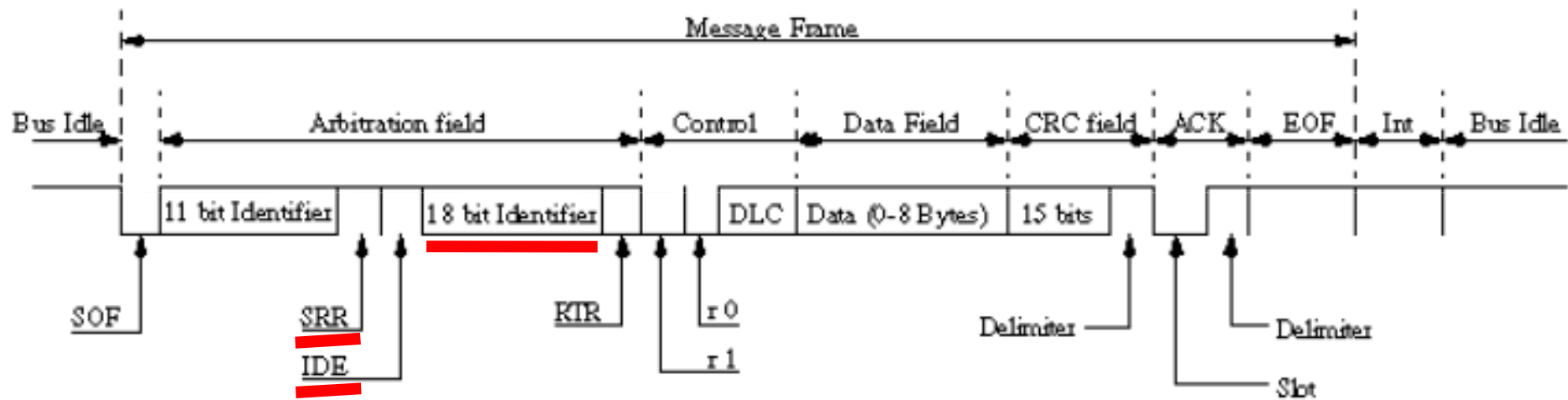
Formato del data/remote frame

- Start Of Frame (SOF): bit dominante (0 logico) che indica l'inizio di un frame. Il rilevamento di un bit dominante durante il Bus Idle è sempre interpretato come un SOF
- Arbitration Field, contenente 11 bit di identificatore
- Remote Transmission Request (RTR) bit: 0=Data Frame, 1=Remote Frame
- Control Field: 6 bit di cui 2 bit riservati per usi futuri e 4 bit di Data Length Code (DLC) che indica il numero dei byte nel Data Field seguente
- Data Field: può variare da 0 a 8 byte
- CRC Field: 15 bit di cyclic redundancy check code + 1 bit recessivo come delimitatore
- Ack: 2 bit di cui il primo è lo Slot Ack che è posto a recessivo da TX ma è sovrascritto con un dominante da ogni stazione che riceve correttamente il messaggio mentre il secondo bit è recessivo e svolge il compito di delimitatore
- End Of Frame (EOF): 7 bit di valore recessivo

Contesa tra Data e Remote Frame

- Remote Transmission Request (RTR) bit:
0=Data Frame, 1=Remote Frame
- Il simbolo 1 è recessivo
- Se un Data Frame e un Remote Frame con lo stesso identificatore (=stesso tipo di dato) sono trasmessi contemporaneamente, vince il Data Frame e il generatore di Remote Frame non deve far altro che ascoltare il campo dati del Data Frame per avere la risposta alla sua domanda

Formato del data/remote frame



Extended CAN (Versione 2.0 B)

Formato del data/remote frame

- L'identificatore è diviso in Base ID lungo 11 bit per garantire la compatibilità con la versione A e in Extension ID di 18 bit.
- Substitute Remote Request (SRR): 1 bit spedito sempre recessive in modo che in caso di competizione con un data frame della versione A quest'ultimo vinca sempre.
- Identifier Extension (IDE) : 1 bit spedito recessive se il nodo sta spedendo messaggi nel formato Extended o dominante se spedisce messaggi nel formato Standard.
- Esistono tre tipi di nodi CAN:
 - Nodi 2.0A che sono capaci di spedire solo messaggi di formato Standard, restituendo errore nel caso ricevano in formato Extended;
 - Nodi 2.0B passive che sono in grado di spedire solo in formato Standard, ma possono ricevere in formato Extended ignorandoli;
 - Nodi 2.0B che possono funzionare in entrambe le modalità.

Acknowledge dei dati

- I bit di un Data/Remote Frame vengono letti dai RX man mano che vengono trasmessi
 - Controlli di conformità
- Se la sequenza di bit passa i controlli allora viene forzato a 0 (dominante) il bit Slot Ack
- Il TX vede tale bit a zero e interpreta questo come corretta ricezione (da almeno uno dei nodi RX)

Processo di segnalazione di errori

- Un nodo CAN rileva un errore (in trasmissione o in ricezione)
- Un Error Frame viene immediatamente trasmesso
- Il messaggio incriminato viene ignorato da tutti i nodi
- Viene aggiornato lo stato del nodo CAN
- Il messaggio viene ritrasmesso, eventualmente competendo con altri

Rilevazione di errore a livello di bit

- Bit Stuffing Error
 - normalmente un nodo in trasmissione inserisce dopo 5 bit consecutivi della stesa polarità un bit di polarità opposta; ciò è chiamato bit stuffing. Un nodo che riceve più di 5 bit consecutivi di segno uguale rileverà un errore di questo tipo.
- Bit Error
 - un nodo in trasmissione ascolta sempre il bus per verificare la corrispondenza con ciò che sta trasmettendo: se esso ascolta un bit diverso dal suo (e non ci troviamo nell'Arbitration Field né nell'Ack Slot) verrà segnalato un errore

Rilevazione di errore a livello di frame

- **Checksum Error**
 - ogni nodo ricevente ricalcola il CRC in base a ciò che ha ricevuto
- **Frame Error**
 - vengono violati alcuni campi fissi del pacchetto (bit che devono essere spediti sempre dello stesso tipo)
- **Acknowledgement Error**
 - il trasmettitore non rileva alcun riscontro al frame appena inviato.

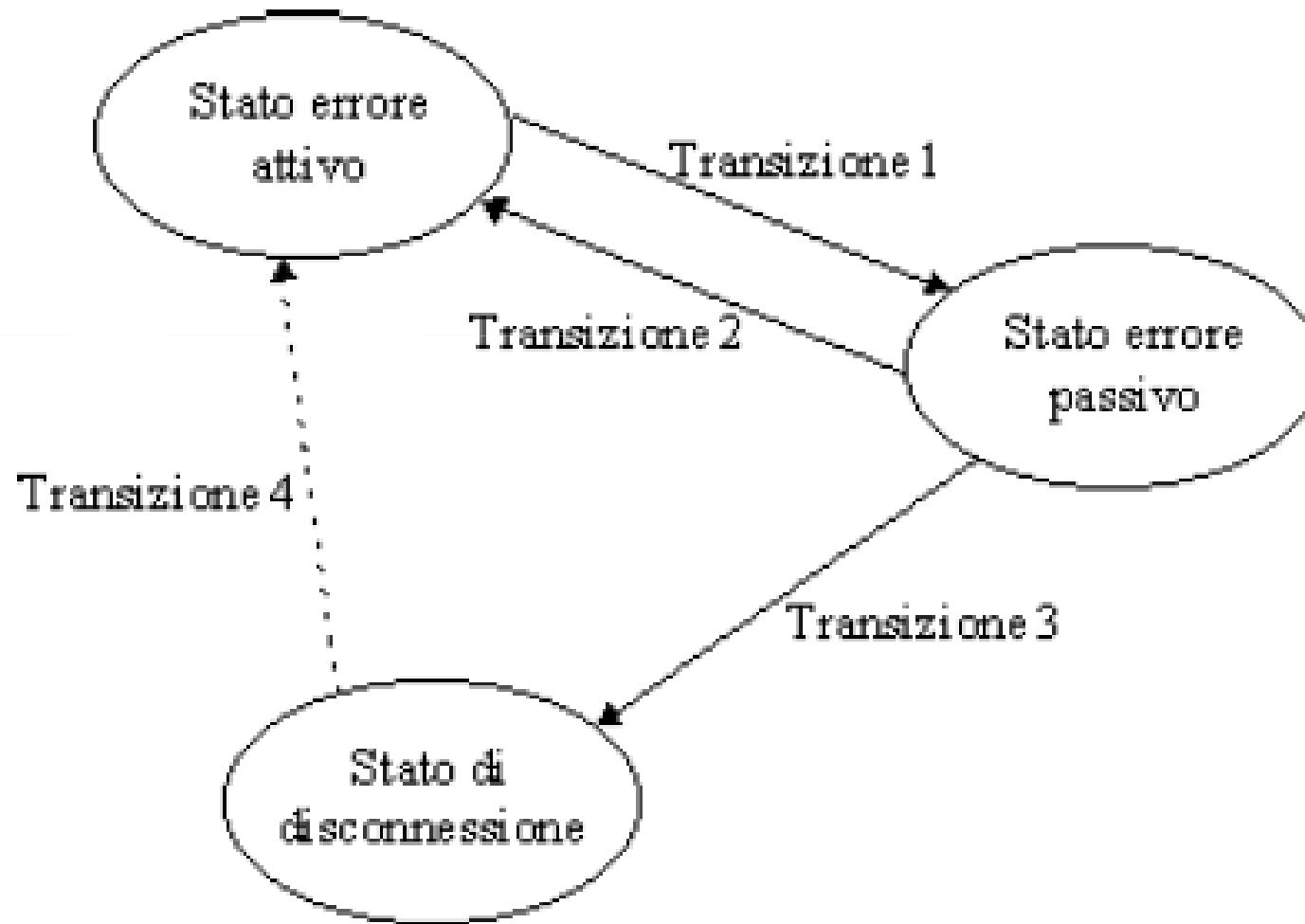
Error frame

- Un Error Frame è costituito da
 - Error Flag lungo almeno 6 bit dello stesso segno e viola volontariamente la regola dello bit stuffing in modo che tutte le altre stazioni rilevino un errore e spediscono anch'esse un Error Flag. Per questo motivo il campo Error Flag nel pacchetto è di lunghezza variabile (max 12 bit) dato dalla sovrapposizione di tutti gli Error Flag spediti.
- Error Delimiter costituito da 8 bit recessive.

Autodiagnostica

- CAN è in grado di distinguere tra condizioni di guasto transitorie (sbalzi di tensione) e guasti permanenti (cattive connessioni, cavi rotti)
- Ogni nodo mantiene due contatori di errori di trasmissione e uno di ricezione
 - Inizializzati a 0
 - Incrementati
 - +1 per un errore in ricezione,
 - +8 per un errore in trasmissione
- Ogni stazione può trovarsi in tre stati

Autodiagnostica



Stato errore attivo

- Nessuno dei due contatori ha superato il valore di 127.
- Il nodo è nel pieno delle sue funzionalità e decrementa di 1 i contatori ogni volta che riceve un messaggio andato a buon fine.
- Quando rileva un errore spedisce un Error Frame costituito da 6 bit dominanti in modo da interrompere sempre la trasmissione corrispondente

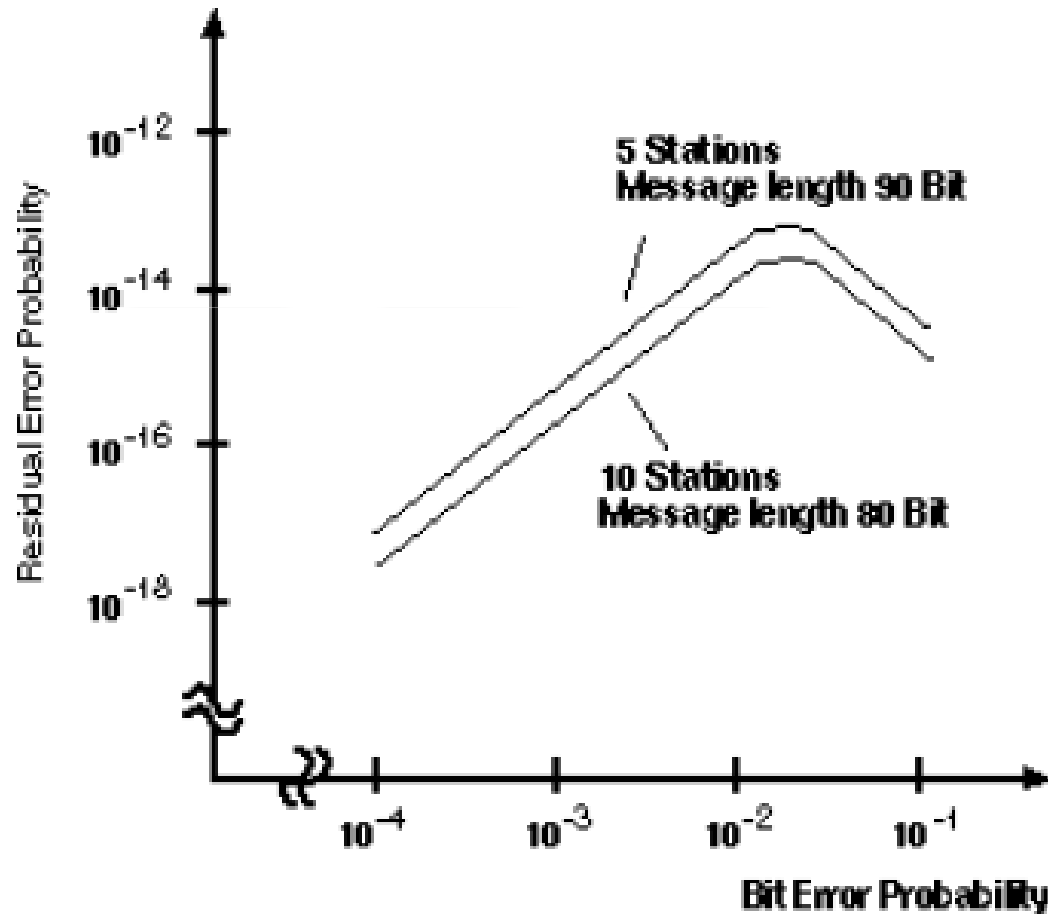
Stato errore passivo

- Almeno uno dei due contatori ha superato 127.
- Il nodo è ancora in grado di eseguire tutte le sue funzioni, ma è probabile che esso presenti dei disturbi o condizioni di guasto.
- Quando esso rileva un errore spedisce un Error Frame di 6 bit recessive che vengono interpretati come errore solo se nessuna stazione sta spedendo un suo proprio messaggio (i bit recessive vengono sovrascritti).

Stato di disconnessione

- Almeno uno dei contatori supera 255
- Il nodo si stacca dal bus e non partecipa alla comunicazione lasciando gli altri nodi nella possibilità di continuare a scambiarsi informazioni.
- Sintomo di un problema permanente che necessita di un intervento esterno per ripristinare il perfetto funzionamento.
 - Alcune implementazioni consentono al nodo di tornare in Stato Errore Attivo dopo che esso abbia ricevuto 128 occorrenze di 11 bit recessive consecutivi (128 messaggi andati a buon fine)
 - altre necessitano di un reset hardware.

Affidabilità di CAN



Probabilità di errore residua

- Probabilità che un messaggio corrotto non sia rilevato tale

Esempio

- Messaggio di 90 bit
- Prob. Err. Residua 10^{-13}
- 1 Mb/s
- Al max 1 msg errato non rilevato ogni 115 gg

Intermission field

- Separa tra loro Data/Remote Frame dai frame precedenti
- Tre bit recessivi
- Dopo l'Intermission Field il bus diventa Idle e una nuova trasmissione può avvenire
- Error Frame e Overload Frame non devono essere distanziati dal frame precedente

Overload Frame

- Simile ad un Error Frame
 - Consiste di un Overload Flag e di un Overload Delimiter
- Può essere generato da due condizioni:
 - Quando un ricevitore ha bisogno di più tempo per processare i dati correnti prima che altri vengano ricevuti
 - Il rilevamento di un bit dominante durante l'Intermission Field
- Un Overload Frame non richiede la ritrasmissione del messaggio che l'ha generato

Profili di realizzazione dei nodi

- Basic CAN
- Full CAN
- Serial Link I/O device

Basic CAN

- Implementazione più economica
- Solo interfaccia di rete (va aggiunto microcontrollore)
- Buffer di ricezione e di trasmissione gestiti con politica FIFO
 - un messaggio può essere ricevuto su un buffer mentre il microcontrollore sta leggendo su un altro buffer;
 - se arriva un messaggio quando tutti i buffer sono pieni, esso viene scartato
 - possono andare perse delle informazioni nell'eventualità che il microcontrollore non sia abbastanza veloce.
 - Un messaggio è inviato scrivendolo in un buffer di trasmissione.

Basic CAN

- I messaggi a cui il nodo è interessato sono filtrati usando due registri che operano sull'identificatore del messaggio
 - Ogni bit della maschera può essere settato ad 1, a 0 o Don't Care; la scelta va fatta molto accuratamente perché ogni messaggio che supera tale controllo viene mandato al microcontrollore e il filtraggio finale verrà effettuato via software, generando carico di lavoro non necessario
- Non supporta la risposta automatica ai Remote Frame che deve essere gestita in software

Full CAN

- Più costoso
- Include anche il microcontrollore
- Insieme di buffer chiamati *mailbox*
 - a ciascuno viene assegnato l'identificatore a 11 bit
 - ad ogni tipo di messaggio compete il proprio buffer specifico
- Quando viene ricevuto un messaggio vengono controllati tutti i buffer di ricezione
 - il filtraggio avviene interamente in hardware

Full CAN

- In trasmissione, il messaggio viene memorizzato nel buffer che gli compete
 - viene attuata anche una politica di selezione del messaggio da trasmettere, favorendo il più prioritario e non una semplice FIFO come nel Basic CAN
- Quando viene ricevuto un Remote Frame, il nodo verifica se esiste un buffer di trasmissione con lo stesso identificatore: se sì, il Data Frame corrispondente viene subito inviato senza chiamare in causa il microcontrollore
 - rischio di invio di dati ormai vecchi rimasti nel buffer

Serial Link I/O device (SLIO)

- Sono nodi CAN che implementano
 - Interfaccia di rete verso il bus CAN
 - Porte di I/O analogiche e digitali verso sensori/attuatori
- Essi sono slave amministrati da un altro nodo attraverso la rete CAN
- Si possono interpretare come porte I/O aggiunte ad un nodo CAN intelligente attraverso il bus CAN

Parte del materiale è stato prodotto da:
Marco Bonato, David Toso – Università di Verona
H. Boterenbrood – NIKHEF Amsterdam

CANOPEN

Motivazione

- È necessario definire un protocollo sopra CAN per standardizzare l'uso di
 - identificatori a 11 bit
 - campo Dati da max 8 byte
- Risultato
 - Livello Applicazione = insieme di servizi e protocolli
 - Communication profile = configurazione dei nodi e modalità di scambio dei dati
 - Device profiles = profili di comportamento per specifici dispositivi (acquisizione di dati, controllori di moto, encoder, ecc...)

CAL + CANopen

- CAN Application Layer (CAL)
 - Creato da Philips Medical Systems
 - Adottato dal consorzio CAN-in-Automation (CiA)
 - Definisce servizi e protocolli
- CANopen
 - Utilizza un sotto-insieme dei servizi CAL
 - Definisce il contenuto dei dati trasmessi

Informazioni generali

- Vengono introdotti indirizzi di nodo (Node-ID)
 - Il nodo 0 ha funzioni di master
 - I nodi “utenti” vanno da 1 a 127
- Object Dictionary (OD)

Object dictionary

- L'Object Dictionary rappresenta il cuore del protocollo
- Si tratta di una sorta di database in cui sono definiti tutti gli oggetti, i tempi, i dati, i modi che un dispositivo userà per comunicare
 - Ogni oggetto è definito da un indice di 16 bit e un eventuale sotto-indice di 8 bit
- Non necessariamente memorizzato completamente in un nodo
 - Molte sue sezioni rimangono come specifica cartacea dello standard e dei profili manufacturer-specific
- L'OD è diviso in sezioni (profili), ciascuno dedicato ad un preciso ambito

Object dictionary

CANopen Object Dictionary		
Index		Object
0000		<i>not used</i>
0001	- 001F	Static Data Types (standard data types, e.g. Boolean, Integer16)
0020	- 003F	Complex Data Types (predefined structures composed of standard data types, e.g. PDOCommPar, SDOPParameter)
0040	- 005F	Manufacturer Specific Complex Data Types
0060	- 007F	Device Profile Specific Static Data Types
0080	- 009F	Device Profile Specific Complex Data Types
00A0	- 0FFF	<i>reserved</i>
1000	- 1FFF	Communication Profile Area (e.g. Device Type, Error Register, Number of PDOs supported)
2000	- 5FFF	Manufacturer Specific Profile Area
6000	- 9FFF	Standardised Device Profile Area (e.g. "DSP-401 Device Profile for I/O Modules" [3]: Read State & Input Lines, etc.)
A000	- FFFF	<i>reserved</i>

Profili

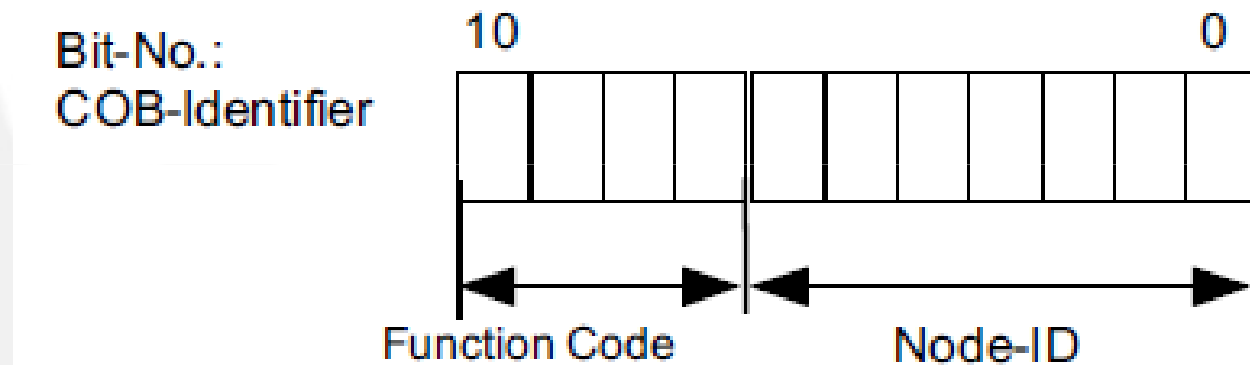
- L'OD è definito sotto forma di documenti chiamati profili:
- Il **communication profile** descrive la forma generale dell'OD e gli oggetti nella Communication Profile Area (communication parameters). Gli oggetti che contiene sono gli stessi per tutti i dispositivi
- Il **device profile** descrive le funzionalità specifiche del dispositivo. I dispositivi sono categorizzati per classi, e dei profili predefiniti sono disponibili per alcune di esse
- Questi profili sono estendibili definendo nuovi oggetti nella Manufacturer Specific Profile Area

Pre-Defined Connection Set

- Configurazione predefinita contenuta nello standard per facilitare l'installazione di reti CANopen semplici

Generazione degli identificatori CAN (11 bit)

- COB-ID = identificatore a 11 bit del protocollo CAN



- Il Function Code è usato per
 - Designare tipi speciali di messaggi
 - Creare delle “well-known ports” per lo scambio di dati

Pre-Defined Connection Set

object	function code (binary)	resulting COB-ID	Communication Parameters at Index
NMT	0000	0	-
SYNC	0001	128 (80h)	1005h, 1006h, 1007h
TIME STAMP	0010	256 (100h)	1012h, 1013h

object	function code (binary)	Resulting COB-IDs	Communication Parameters at Index
EMERGENCY	0001	129 (81h) – 255 (FFh)	1014h, 1015h
PDO1 (tx)	0011	385 (181h) – 511 (1FFh)	1800h
PDO1 (rx)	0100	513 (201h) – 639 (27Fh)	1400h
PDO2 (tx)	0101	641 (281h) – 767 (2FFh)	1801h
PDO2 (rx)	0110	769 (301h) – 895 (37Fh)	1401h
PDO3 (tx)	0111	897 (381h) – 1023 (3FFh)	1802h
PDO3 (rx)	1000	1025 (401h) – 1151 (47Fh)	1402h
PDO4 (tx)	1001	1153 (481h) – 1279 (4FFh)	1803h
PDO4 (rx)	1010	1281 (501h) – 1407 (57Fh)	1403h
SDO (tx)	1011	1409 (581h) – 1535 (5FFh)	1200h
SDO (rx)	1100	1537 (601h) – 1663 (67Fh)	1200h
NMT Error Control	1110	1793 (701h) – 1919 (77Fh)	1016h, 1017h

Tipi di messaggi CANopen

- Service data object
 - Query dell'OD di un nodo
 - Può occupare più frame CAN
- Process data object
 - Trasferimento di dati in tempo reale
 - Un solo frame CAN
- Administrative messages (NMT)
 - Network management e network monitoring
- Special function objects

Service data object

- Lettura dell'Object Dictionary di un nodo (ad es. per ottenere informazioni di configurazione)
- Politica client-server
 - Il client chiede e il server manda risposta
- La risposta
 - Può occupare più data frame
 - Conferma di ogni frame
- L'identificatore è utilizzato per creare delle “well-known ports” tra nodi

Process data object

- Dati real-time (letture di dispositivi)
- Inviati secondo la politica produttore-consumatore
 - 1 nodo manda informazioni a molti senza frame di conferma
- L'identificatore è utilizzato per creare delle “well-known ports” tra nodi

Messaggi di tipo Administrative (NMT)

- Modalità master-slave
 - Il master sollecita azioni verso gli slave
- Tipi
 - Module control
 - Node guarding
 - Heartbeat
 - Boot-up

NMT Module control

- Data Frame da master a slave

COB-ID	Byte 0	Byte 1
0x000	CS	Node-ID



Command Specifier	NMT Service
1	Start Remote Node
2	Stop Remote Node
128	Enter Pre-operational State
129	Reset Node
130	Reset Communication

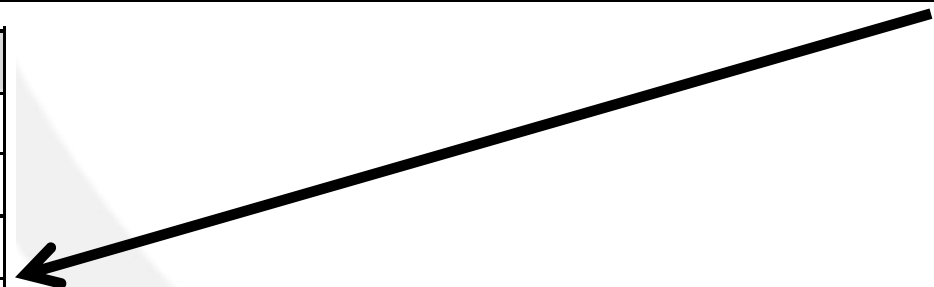
NMT Node guarding

- Remote frame da master a slave
- Data frame di risposta

COB-ID
0x700 + Node_ID

COB-ID	Byte 0
0x700 + Node_ID	bit 7: <i>toggle</i> , bit 6-0: <i>state</i>

Value	State
0	Initialising
1	Disconnected *
2	Connecting *
3	Preparing *
4	Stopped
5	Operational
127	Pre-operational



NMT Hearthbeat

- Messaggio periodico dai nodi al master

COB-ID	Byte 0
0x700 + Node_ID	<i>state</i>

<i>state</i>	Meaning
0	Boot-up
4	Stopped
5	Operational
127	Pre-operational

NMT Boot-up

- Data Frame da slave a master

COB-ID	Byte 0
0x700 + Node ID	0

- Serve per comunicare che il node è entrato nello stato *pre-operational*

Messaggi Special Function Objects

- SYNC
 - Messaggi periodici di sincronizzazione
- TIMESTAMP
 - Messaggi periodici per fornire la nozione di tempo
- EMERGENCY
 - Messaggi generati da un nodo quando si verifica un errore interno