

Laurea Magistrale in Ingegneria Informatica  
Laurea Magistrale in Ingegneria Elettronica e Telecomunicazioni per  
lo Sviluppo Sostenibile

## **Sistemi e Tecnologie per l'Automazione LM**

### **RETI INFORMATICHE NELL'AUTOMAZIONE**

**Ing. Gianluca Palli**

**DEI - Università di Bologna**

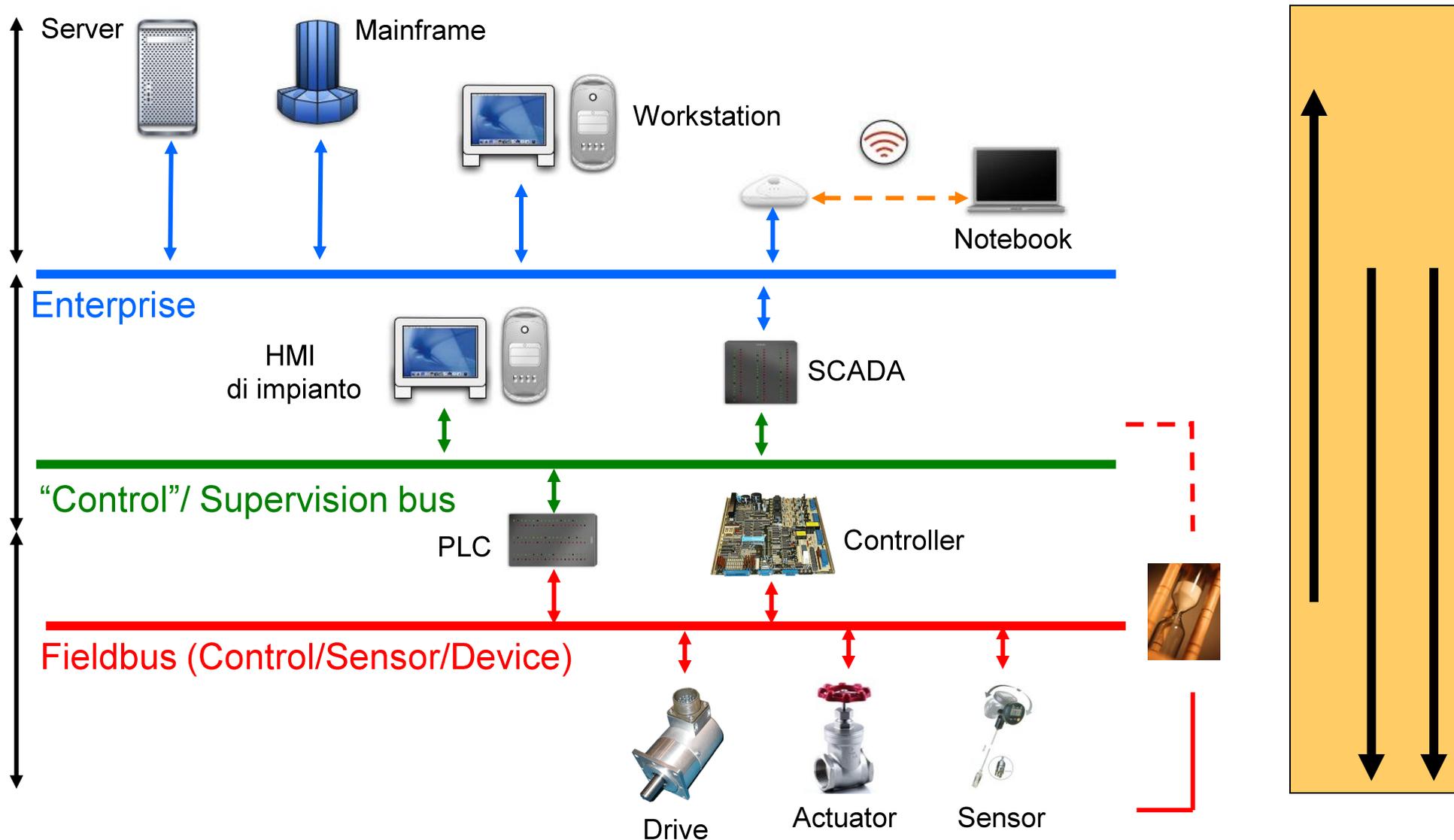
**Tel. 051-2093186**

**E-mail: [gianluca.palli@unibo.it](mailto:gianluca.palli@unibo.it)**

**<http://www-lar.deis.unibo.it/people/gpalli/>**

**Revisionato il 25/11/2015**

# Comunicazione nella Piramide dell'Automazione



## Diverse specifiche ai diversi livelli:

### ■ Livello di gestione (rete aziendale di stabilimento)

- Reti informatiche classiche: LAN Ethernet e TCP/IP, spesso connesse a Internet
- Applicazioni (livello 7): classiche (HTTP, FTP....)

### ■ Livello di supervisione

- Tipicamente ancora reti informatiche classiche: LAN Ethernet e TCP/IP, in alcuni casi connesse a Internet
- Applicazioni (livello 7): particolari per la supervisione di impianto
  - SCADA, OPC
  - Specifiche Real time molto blande, gestite a livello di applicazione

### ■ Livello dei controlli

- Reti specifiche per tale ambito
  - Gestione specifiche real time ed altro
- Applicazioni: specifiche per tale ambito

---

# RETI INFORMATICHE PER IL LIVELLO DEI CONTROLLI FIELD-BUS

## Livello dei controlli della PA:

- Reti utilizzate in tale ambito dette: Fieldbus
  - **Sensorbus, Controlbus, Devicebus, Motionbus....**
- Antagonisti:
  - **A tale livello resistono ancora:**
    - Trasmissioni analogiche (intrinsecamente punto-punto)
    - Trasmissioni punto-punto con linee digitali dedicate
- Vantaggi delle soluzioni fieldbus:
  - **Semplicità cablaggio**
  - **Flessibilità**
  - **ATTENZIONE: aumenta la complessità di gestione!**
    - Informazioni circolano su elementi condivisi: in quanto tempo?

## Specifiche Principali per FieldBus

- **Robustezza meccanica**
- **Robustezza elettromagnetica**
  - **Ambiente EM ostile**
  - **Dati non si devono corrompere troppo spesso!**
- **Soddisfacimento vincoli di tipo real-time**
  - **I task del livello dei controlli che comunicano su tali reti hanno vincoli realtime**

## Tipologie di comunicazioni e di traffico sui FieldBus

### ■ Derivanti da

- **Caratteristiche funzionali del livello dei controlli della PA**
- **Scelta delle unità di elaborazione**
- **Mappatura in tasks delle funzioni**
- **Scelta di Sensori e Attuatori (intelligenti)**

### ■ Tipicamente:

- **Dati per le applicazioni di *piccole dimensioni (pochi byte)***
- **Tipologia di comunicazioni:**
  - **Misure da sensori**
  - **Comandi per attuatori**
  - **Comandi e “misure virtuali” tra applicazioni**
  - **Allarmi**
  - **Messaggi per o da HMI o simili**
- **Tutte le comunicazioni dette si possono dividere in due grandi classi:**
  - **State Message**
  - **Event Message**

## Tipologie di comunicazioni e di traffico sui FieldBus

### ■ State messages

- **Messaggi fotografano completamente la condizione (stato) di interesse in un certo istante**
- **Istante di generazione deciso da a livello di applicazione di ctrl in modo arbitrario rispetto alle variazioni di stato**
  - **Istanti in cui stato è variato stato non sono rilevanti, è rilevante il valore attuale**
- **Non è necessario combinarli con i messaggi passati per conoscere la condizione (stato) corrente**

### ■ Event Messages

- **Messaggi legati alla variazione di stato di una informazione di interesse**
- **Istante di generazione legato all'istante di variazione dello stato**
  - **ISTANTE DI VARIAZIONE E' IMPORTANTE**
  - **Si può trasmettere solo la variazione**
    - ◆ **Per ricostruire la condizione (stato) è necessaria la conoscenza di tutta la sequenza dei messaggi evento (da un certo stato iniziale)**
  - **Si può trasmettere tutto lo stato**
    - ◆ **Comunque è rilevante l'istante di variazione (non si ricade in state messages)**

## Tipologie di comunicazioni e di traffico sui FieldBus

### ■ State messages

#### ▫ Caratteristiche in termini di traffico su fieldbus

##### • Di solito periodici con periodo noto

- ◆ Campionamento per Ctrl digitale
- ◆ Ctrl logico su RTOS Time Driven come appl. pseudotrasf ciclica
- ◆ Scambio periodico con HMI

##### ⇒ TRAFFICO PERIODICO NOTO CON VINCOLI RT E NON

- Periodo deciso in funzione di accuratezza temporale necessaria su informazione

- ◆ Da abbinare ai tempi di risposta sui nodi

- Di solito non serve acknowledgment fra ricevitore e trasmettitore

- ◆ Se non arriva non serve ritrasmettere

→ o informazione era critica, quindi eccezione da gestire da lato rx

→ o si usa vecchia informazione e si attende la trasmissione successiva

- ◆ Riduzione del traffico sulla rete

## Tipologie di comunicazioni e di traffico sui FieldBus

### ■ State messages (indicano valori, non eventi)

#### ▫ **Caratteristiche in termini di traffico su fieldbus (cont'd)**

##### ⇒ **TRAFFICO PERIODICO NOTO CON VINCOLI RT**

In certe applicazioni dove si deve

- ottimizzare prestazioni temporali complessive
- utilizzare tutti i dati mandati ciclicamente correttamente

può essere necessario

#### **SINCRONIZZARE TRASMISSIONE PERIODICA CON ESECUZIONE CICLICA DEI TASK sui nodi TX e RX**

- ◆ Si minimizza la latenza tra generazione in task TX e utilizzo in task RX
- ◆ Si evitano i drift tra i clock locali di Tx e Rx
- ◆ Tipico per
  - Ctrl diretto di variabili temporali e Sensori e Attuatori
  - Interazione tra Gen Traiettorie e Azionamenti in Motion Control

## Tipologie di comunicazioni e di traffico sui FieldBus

### ■ Event message

#### ▫ Caratteristiche in termini di traffico su fieldbus

- Di solito sporadici

  - ◆ Scatenati in relazione alle variazioni

⇒ TRAFFICO SPORADICO NON NOTO A PRIORI CON VINC. RT E NON

- Necessario lettura di ogni msg (e notifica con Ack)

- Necessario evitare doppia lettura

- Problema di ricostruzione dell'ordinamento temporale di eventi e sincronizzazione su nodi diversi

  - può cambiare stato ricostruito

## Tipologie di comunicazioni e di traffico sui FieldBus

### ■ State Messages vs. Event Messages in livello controlli di PA

#### ▫ Tipicamente

#### • State messages preferiti in controlli (e quindi in FBus)

- ◆ Pro: Gestione più semplice (stato è nel messaggio)

Perdita msg può essere tollerata (almeno in parte)

Quando sono ciclici → info temporale implicita sugli istanti di variazione

- ◆ Contro: possibili molte comunicazioni inutili (si conferma lo stesso stato)

**TIPICA FREQUENZA DI REFRESH DATI: da 100Hz a 10kHz**

- ◆ Piuttosto elevata (non confondere con quella di livello fisico delle reti)

#### • Event message usati:

- ◆ In comunicazioni non critiche in senso RT in cui si ottiene forte riduzione info da trasmettere

Tipicamente in comunicazioni HMI o simili

- ◆ IN SITUAZIONI CON VINCOLI RT IN CUI E' IMPORTANTE ISTANTE TEMPORALE PRECISO IN CUI AVVIENE LA VARIAZIONE DI CONDIZIONE

Tipicamente: **Allarmi distribuiti** che devono essere composti per risalire a causa scatenante.

- Gestione con messaggi a stato ciclici troppo pesante ...

→ Problema della marcatura temporale univoca per tutti i nodi ...

## Approfondimento Specifiche per Vincoli RealTime:

### 1. Ritardo di trasmissione contenuto

- **Ritardi negli anelli di controllo sono deleteri!!**
- **Utile avere elevata “efficienza della rete”**
  - **Efficienza:** indica overhead di bit di controllo rispetto ai dati

### 2. Determinismo (o Predicibilità)

- **I dati devono essere consegnati entro un certo tempo fissato**
  - **Si veda in seguito**

### 3. Possibilità di sincronizzare i nodi

- **Primitive di comunicazione di rete per sincronizzare innesco dei tasks (ciclici) su nodi diversi**
  - **Complesso, ma dà maggiori possibilità**
    - ◆ **Vedi state msg con vincoli di latenze minime e non perdita..**
- **Primitive che allineano orologi dei nodi (“tempo globale”...)**
  - **Magari non si sincronizzano tasks, ma conteggio del tempo**
    - ◆ **Vedi msg evento in cui è importante istante**

---

# LIVELLO FISICO NEI FIELDBUS

## Caratteristiche più diffuse per i livelli fisici dei fieldbus:

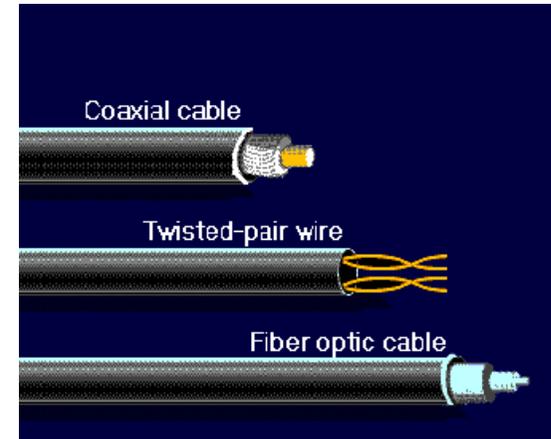
- Tipicamente supporto fisico del canale di trasmissione è “solido”:

(supporto guidato)

- **Rame:**

- coppie cavi twistati (doppini)
  - ◆ **Trasmissione sigle ended o differenziale**
- cavi coassiali

- **Fibra ottica**



- Tipicamente il canale fisico non è multiplexato in TDM o FDM (no canali virtuali)



## Particolare Livello Fisico usato in Fieldbus CAN:

- **CAN: Controller Area Network**
  - **Nasce in Automotive**
- **Broadcast a bus, Differenziale**
- **Codifica**
  - **2 livelli**
- **Struttura WIRED-AND**
  - **Simile a struttura open collector**
  - **Esiste una configurazione (lo zero logico) dominante**
  - **Struttura che consente l'arbitraggio e la risoluzione dei conflitti in HW a livello di singolo BIT**
  - **Adatto a realizzare MAC CSMA/CR**
    - **Detta in tal caso anche CSMA/BA (Bit-wise Arbitration)**

---

# LIVELLO DATA LINK NEI FIELDBUS

### Funzionalità più diffuse per i livelli data-link dei FieldBus:

- **Divisione in pacchetti praticamente assente**
  - **I blocchi di dati generati dal livello applicazione sono dimensionati per il pacchetto base del data-link layer**
  - **Meno astrazione, ma più efficienza**
- **Meccanismi aggiuntivi che possono essere presenti:**
  - **Verifica dell'Errore in Ricezione (esempio: CRC)**
  - **Notifica di Ricezione di Pacchetto (ACK)**
  - **Ritrasmissione (per errori o mancato ACK)**
  - **Meccanismi di Controllo di Flusso (se ricevitore “è pieno”)**
  - **In alcuni casi sono presenti, ma devono essere “molto snelli” per ridurre i tempi**

## Funzionalità più diffuse per i livelli data-link dei FB (cont'd):

### ■ MAC (Media Access Control)

- **Allocazione Statica o Allocazione Dinamica dell'accesso al media**
  - mirata a eliminare/gestire i conflitti
- **Necessario per determinismo (come visto)**
- **NON SEMPRE GESTITO IN MODO OTTIMALE RISPETTO AL DETERMINISMO (tmax facilmente noto e basso jitter)**
  - i.e. non tutti i fieldbus sul mercato sono ugualmente performanti rispetto a specifiche Real-Time

## Accesso al Canale

- **Quando una stazione deve inviare dati, ascolta il canale**
  - **se il canale è libero inizia a trasmettere immediatamente**
  - **se il canale è occupato, ritenta successivamente**
    - **tempi di attesa fissi**
    - **tempi di attesa variabili (casuali) da evitare in automazione**
  - **se dopo aver trovato il canale libero due o più stazioni iniziano a trasmettere “contemporaneamente” avviene una collisione**
    - **il ritardo di propagazione del messaggio sulla linea definisce il concetto di contemporaneità, e determina la banda ottenibile sul canale**
  - **Mentre trasmette, il nodo “ascolta” il canale per rilevare la collisione**

## Rilevamento della Collisione

### ■ Il nodo ascolta il canale per rilevare la collisione

#### ▫ **protocolli a rilevamento della collisione (Collision Detection, CD)**

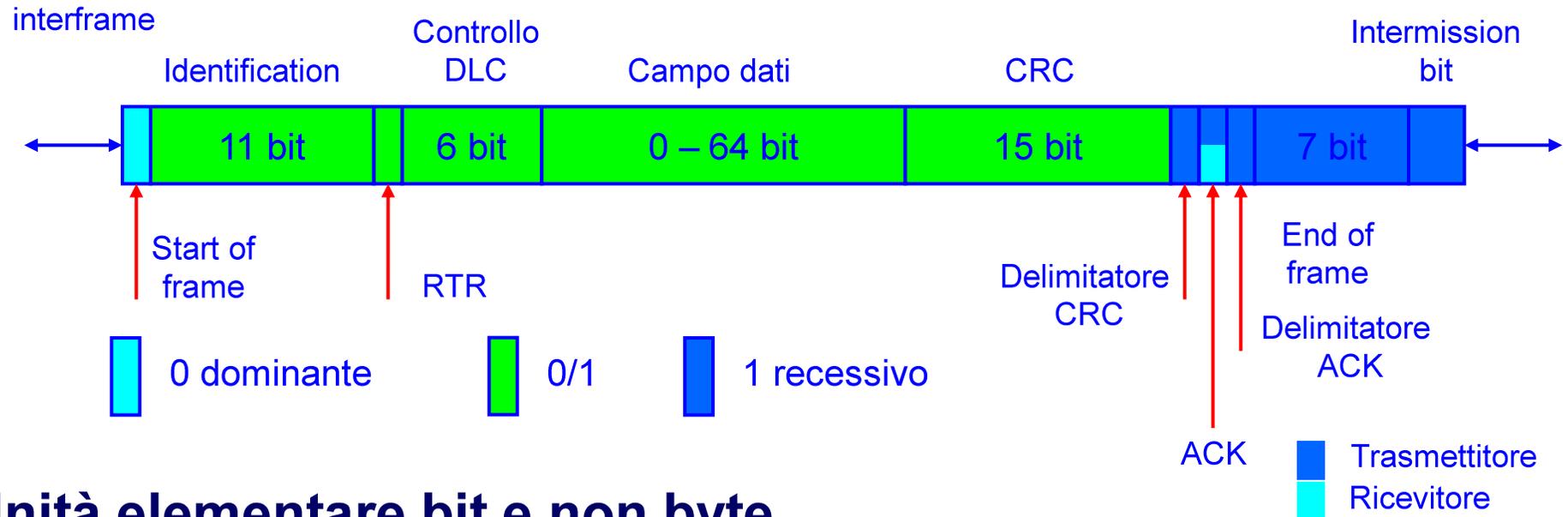
- quando avviene la collisione, i messaggi vengono corrotti
- tutti smettono di trasmettere e ritentano successivamente
  - ◆ **Attesa di tempi anche random (da evitare)**
- può capitare che il tempo di latenza non sia limitato (trasmissione non garantita)
  - ◆ **Es. Ethernet (algoritmo backoff, da evitare in automazione)**
  - ◆ **Per questo motivo Ethernet standard non viene utilizzata in automazione**
  - ◆ **Esistono varianti di Ethernet appositamente studiate per l'automazione (vedi in seguito)**

## Rilevamento della Collisione

### ■ Il nodo ascolta il canale per rilevare la collisione

- **protocolli a risoluzione della collisione (Collision Resolution, CR)**
  - uno tra i messaggi trasmessi determina lo stato del canale
    - ◆ Ogni messaggio e/o nodo ha una priorità specifica per gestire l'arbitraggio in caso di collisione
  - le stazioni che rilevano uno stato diverso da quello voluto smettono di trasmettere
  - la stazione che vince l'arbitraggio ed il cui messaggio non viene corrotto prosegue nella trasmissione fino al completamento
    - ◆ Esiste sempre un messaggio che viene trasmesso
    - ◆ Es. Controller Area Network (CAN)

## Struttura del frame CAN



### ■ Unità elementare bit e non byte

- **Arbitraggio a livello di bit (CSMA/BA, Bitwise Arbitration)**
  - Si deve risolvere nella zona di “identification”
- **Si sfrutta inoltre wired-and anche per semplificare acknowledge**
- **Nel caso due stazioni stiano cercando di trasmettere contemporaneamente, vince quella che trasmette la codifica dominante**
  - Ovvero con indirizzo (ID) più basso (più prioritario)
  - Priorità a livello di nodo (menzione a CanOpen)

Particolare Livello Fisico usato in Fieldbus CAN:

Esempio di driver CAN

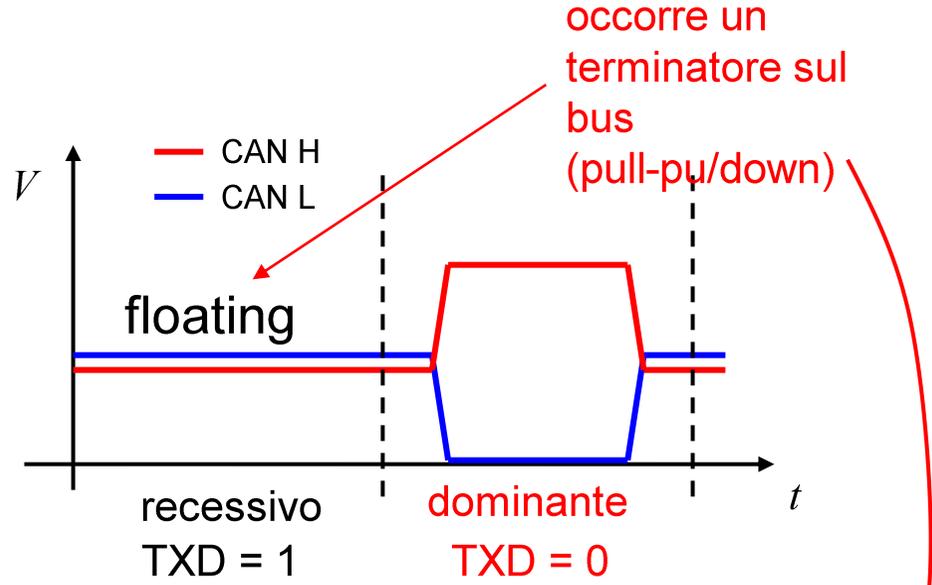
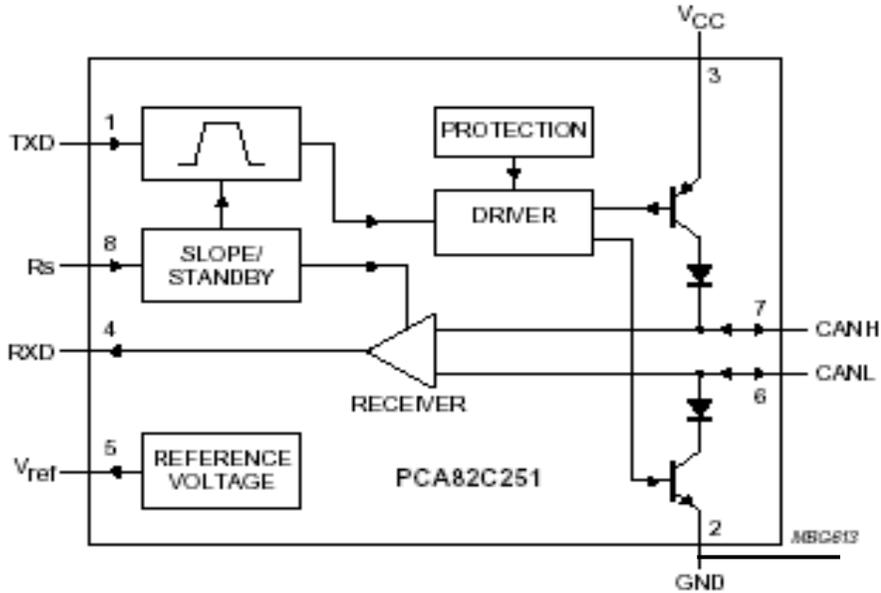
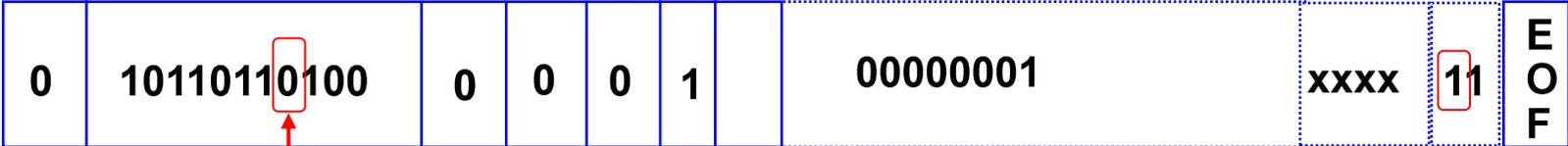


Tabella di verità

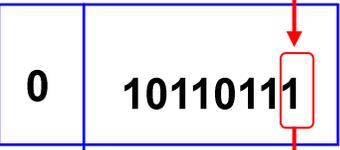
TXD	CANH	CANL	BUS STATE	RXD
0	HIGH	LOW	dominant	0
1 (or floating)	floating	floating	recessive	1(2)

# Esempio di contesa

Trasmissione nodo 1



Trasmissione nodo 2



Il nodo 2 perde l'arbitraggio e interrompe la trasmissione

come tutti gli altri nodi invia comunque il campo ACK



msg. ricevuto correttamente

Stato della linea



Campo di arbitraggio

## Limitazioni

- Poiché l'arbitraggio (e la sincronizzazione) devono avvenire durante la trasmissione del bit, la limitazione introdotta dal tempo di ritardo dovuto alla propagazione diviene

$$2\tau + T_r < T_{bit}$$

- Tempi di bit relativamente elevati (in particolare rispetto a CSMA/CD)

- Per avere velocità elevate necessario ridurre i  $\tau$

- A parità di velocità di propagazione  $\Rightarrow$  ridurre distanze

- **Esempio CAN**

- Configurazioni ammesse

Velocità	1 Mbit/s	500 kbit/s	250 kbit/s	125 kbit/s	100 kbit/s	50 kbit/s
Lunghezza	< 25 m	< 100 m	< 250 m	< 500 m	< 650 m	< 1 km

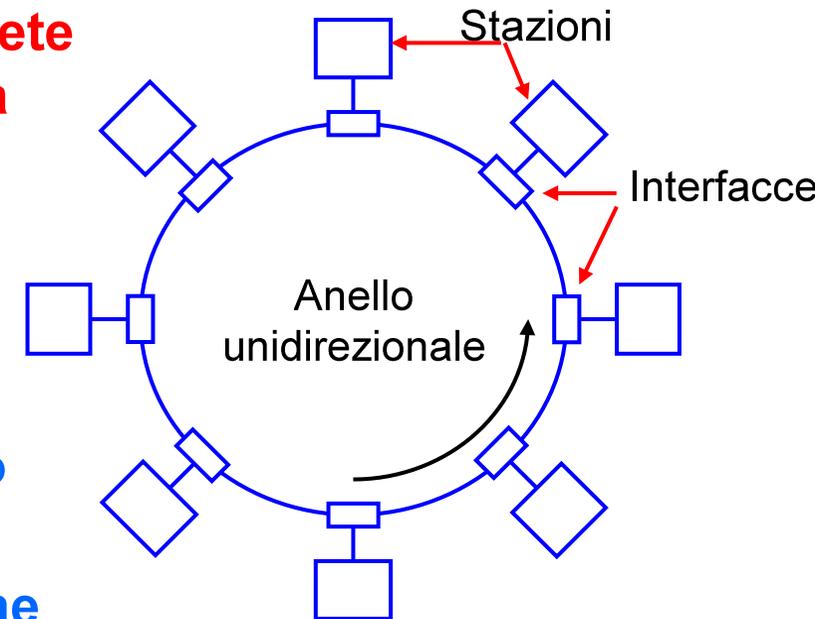
## Token Ring

### ■ Meccanismo di accesso al mezzo mediante Token

- sequenza speciale di bit che circola sulla rete quando tutte le interfacce sono in modalità ascolto

### ■ Invio di un Frame

- Per trasmettere, il nodo rimuove il token dall'anello e inizia a trasmettere il frame
  - Il frame viene rimosso quando tornano alla trasmittente
  - tutte le stazioni ascoltano quindi il frame
    - ◆ rete broadcast anche se connessione punto a punto
- Al termine, ritrasmette il token che ricomincia a circolare fino a che un'altra stazione non lo rimuove



## Mater-Slave o Polling

### ■ Protocollo di tipo token centralizzato

- **Un nodo speciale (master o arbitro) invia un messaggio esplicito (token) al nodo che ha il diritto di trasmettere**
  - Due messaggi per ciascuna dato da trasmettere
  - Limitazione della capacità del canale
- **Per garantire il determinismo alla rete e aumentare la capacità del canale, è necessario limitare la dimensione massima dei messaggi**
- **Soluzione dinamica, centralizzata e senza collisioni**
- **Affidabilità del protocollo**
  - Guasto del nodo master...

## Time Division Multiple Access (TDMA)

- **Divisione del tempo di accesso al mezzo in slot, assegnati ai nodi in base a una certa strategia**
  - **Gli slot possono essere tutti uguali o diversi**
  - **TDMA sincrono**
    - L'accesso è assegnato periodicamente a tutti i nodi in modo ciclico
  - **TDMA asincrono**
    - L'accesso viene assegnato alle stazioni in base alle loro necessità
      - ◆ Se un nodo non ha necessità di trasmettere, non utilizza il suo slot
  - **Sincronizzazione dei clock dei nodi**
    - È fondamentale per definire gli istanti di inizio della trasmissione di ogni nodo
      - ◆ Può essere centralizzata o distribuita

## Confronto tra MAC

### ■ CSMA-CD (Usato in Ethernet)

#### ▫ Non deterministico

- In caso di collisione, un nodo può “catturare” la rete per un tempo indefinito
- Legato alla casualità del meccanismo di backoff
- Difficile determinare il latency delay nel caso peggiore, a traffico elevato

#### ▫ Elevato jitter

#### ▫ Nessun supporto intrinseco alla sincronizzazione dei nodi

#### ▫ Buone prestazioni per traffico basso (ciclico o aciclico) rispetto alla “banda” del Livello 1

- Collisioni ridotte
- Nel caso di Ethernet velocità molto elevata

## Confronto tra MAC

### ■ CSMA-CR (Usato in CAN)

- **“Potenzialmente” deterministico**
  - Massimo latency delay per ogni messaggio è calcolabile
  - Però VALUTAZIONE COMPLESSA: fortemente dipendente da numero di nodi e dal tipo di traffico (ed in modo complicato)
- **Spesso limitazioni di velocità e dimensioni massime**
  - Tipicamente, come in CAN, si usa CR di tipo BA
- **Jitter può essere elevato**
  - dipendente comunque da traffico del momento
- **No supporto intrinseco alla sincronizzazione tra nodi**
- **Più adatto a gestione di traffico ciclico noto**
  - La valutazione del massimo latency delay è più facile in tale scenario

## Confronto tra MAC

### ■ Token (Ring o Bus)

- **Determinismo**
  - Facilmente valutabile il caso peggiore
- **Jitter basso soprattutto con traffico ciclico noto**
- **Possibile supporto alla sincronizzazione ai nodi**
  - Circolazione Token... non banale
- **Traffico ciclico noto è preferibile**

#### Ulteriori note:

- **Elevata adattabilità**
- **Non efficiente a basso traffico e con molti nodi**
  - Trasmissione del token
- **Il tempo di rotazione del token deve essere inferiore alla deadline minima**
  - Problemi nel caso di sistemi con deadline molto differenti fra loro
- **In caso di perdita del token, tempi di ripristino lunghi**

## Confronto tra MAC

### ■ Master-Slave (detta anche Polling)

#### ▫ **Determinismo**

- Facile calcolare il massimo latency delay di un messaggio per una data configurazione di nodi e di traffico

#### ▫ **Ridotto jitter se si limita la variabilità di lunghezza massima dei messaggi**

#### ▫ **Parziale supporto alla sincronizzazione tra nodi**

- Messaggi da master strutturalmente presenti si possono usare a tale fine

#### ▫ **Buone prestazioni per poche stazioni e traffico periodico**

- Efficienza bassa a causa dei messaggi di “autorizzazione” da master a slave
- Per traffico sporadico:
  - ◆ Master deve sempre autorizzare periodicamente slave che potrebbe non avere msg.

## Confronto tra MAC

### ■ TDMA

- **Deterministico e jitter ridotto, con buona sincronizzazione dei clock di interfacce di rete locali**
  - Serve misura di tempo allineata tra le interfacce di rete
    - ◆ **Tempo globale....**
- **Supporto intrinseco alla sincronizzazione dei nodi**
  - Le interfacce di rete devono essere sincronizzate
  - Segnali di sincronismo “esportabili” ai nodi
- **Problematiche legate a allocazione statica del mezzo**
  - Scarsa efficienza
  - Soluzioni ATDMA per migliorare
- **Ottimale per traffico ciclico noto**
  - Allineamento con le slot temporali definite nel bus

---

# CENNI A FIELDBUS COMMERCIALI E PROSPETTIVE

DeviceNet	Foundation Fieldbus
ControlNet	Profibus
Mechatrolink	Sercos
Interbus	Swiftnet
Can	CanOpen
ModBus	

**Alcuni Fieldbus non definiscono tutti i livelli (1,2,7)**

- **Esempio: ModBus e CanOpen**

**Alcuni Fieldbus definiscono livelli che possono andare oltre al 7**

- **Esempio: CanOpen**
  - **Obiettivo intercambiabilità e interoperabilità di nodi di diversi costruttori**
    - ◆ **Es: Azionamenti elettrici di diversi costruttori, rimpiazzabili e parametrizzabili in modo trasparente... chimera... per ora...**

# Alcune Caratteristiche di Fieldbus Commerciali

## I Fieldbus si differenziano per applicazione

- **Alcuni FBus sono più adatti alle reti a livello controllo/supervisione**
  - **Es: Modbus**
    - ◆ **CSMA/CD (basato su Ethernet TCP/IP vedi dopo..)**
- **Alcuni sono specifici per il motion**
  - **Elevato determinismo (bassissimo jitter)**
  - **Sercos, Mechatrolink**
    - ◆ **TDMA**
- **FBus per sensori e attuatori generici**
  - **Profibus → Master-Slave o Token Bus di tipo address-oriented**
  - **CAN → CSMA/CR (... magari con Master-Slave a livello 7) di tipo message-oriented**

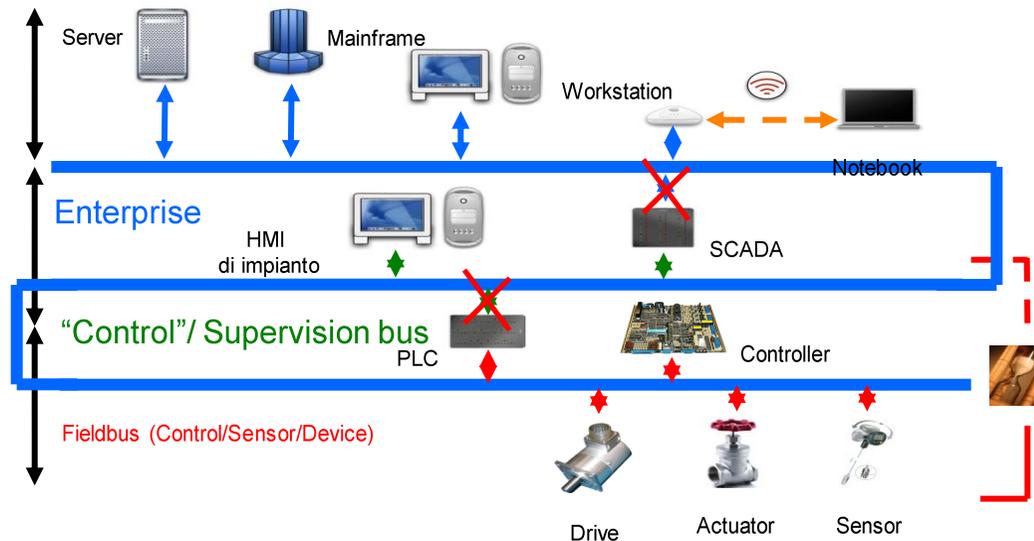
## Standard IEC61158 (04/2003): fieldbus standard for use in industrial control systems

- **Definisce una struttura standard di Fieldbus a cui tutte le realizzazioni commerciali dovrebbero ispirarsi**
- **Alcuni ci si avvicinano, altri sono molto lontani**
  - **Uniformità tra i fieldbus commerciali è una chimera**
- **Comunque utile pietra di paragone**
  - **(come ISO/OSI per le reti informatiche in generale)**

## ■ Trend: Ethernet come FieldBus

### □ Motivazione di fondo:

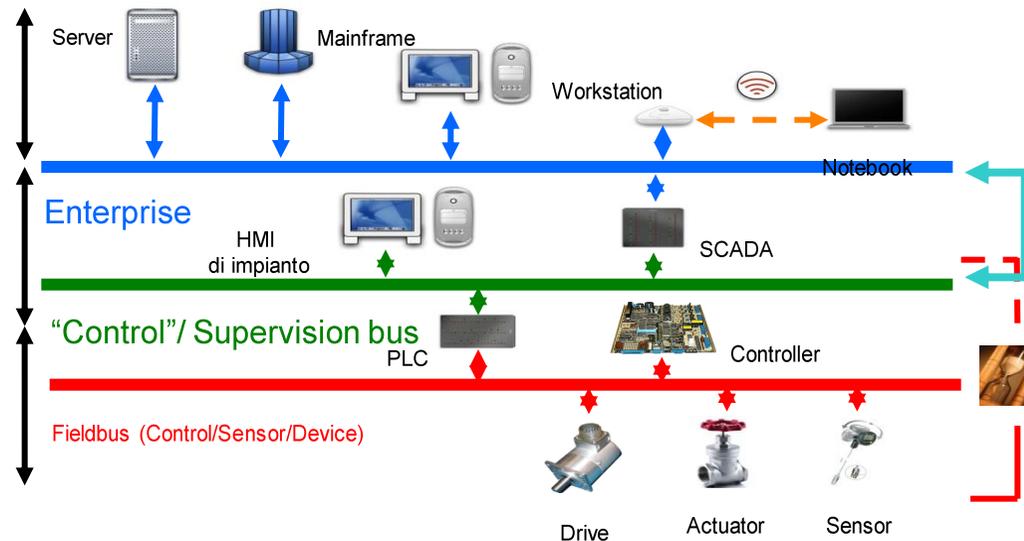
- Ethernet è veloce (e a costo contenuto, vista l'elevata diffusione)
- **ATTENZIONE: Si confonde Ethernet con Internet!!!!**
- **SI VORREBBE AVERE UNA UNICA RETE PER TUTTA LA PIRAMIDE DELL'AUTOMAZIONE!!!**
  - ◆ Anche se i reali vantaggi funzionali non sono chiari... è una moda!



## ■ Trend: Ethernet come FieldBus

### □ Alcune conclusioni:

- Ethernet + TCP/IP può essere utile per rete controllo/supervisione
- Poche richieste di Real time
- Facilità di interfacciamento con livello gestione (rete enterprise)
  - ◆ TCP/IP dà una gestione uniforme



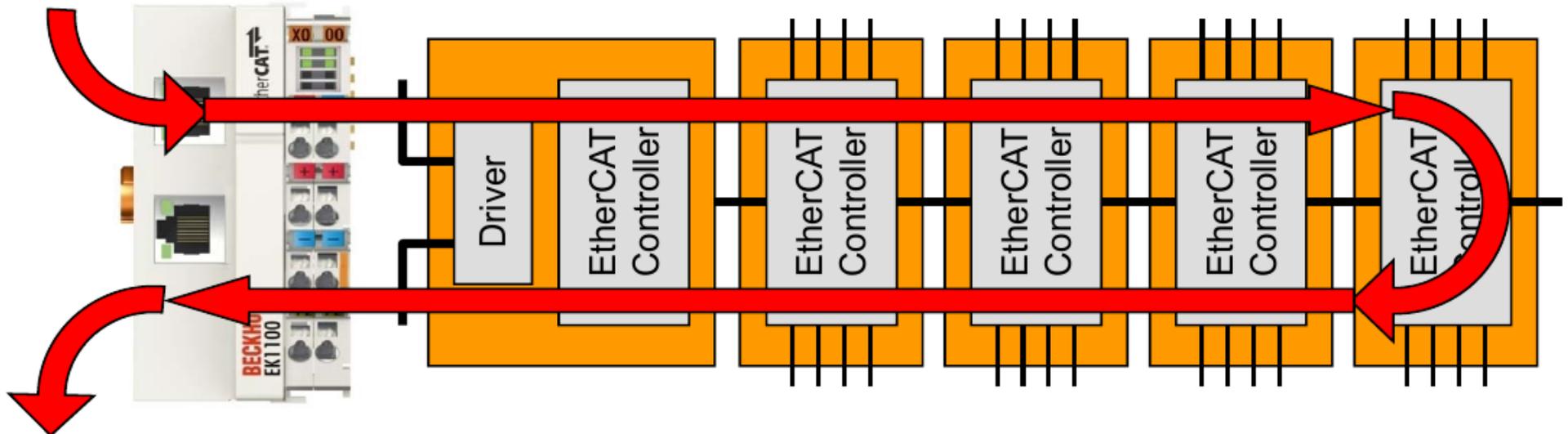
## ■ Trend: Ethernet come FieldBus

### ▫ EVOLUZIONI PER L'AUTOMAZIONE → ETHERNET INDUSTRIALE:

- Livello 1 e 2 di Ethernet integrati/rivisti per
  - elevata robustezza meccanica e EMC
  - elevato determinismo (no CSMA/CD)
- Alcune soluzioni:
  - Profinet
  - EtherCat
  - PowerLink
  - Ethernet/IP (industrial Process)
  - .....
- Hanno poco a che fare con Ethernet “da ufficio” e non sono conformi a TCP/IP
  - ◆ Sono veri Fieldbus
  - ◆ A livello 7 (applicazione) si possono “riciclare” livelli 7 di altri Fieldbus
  - ◆ Alcune varianti tentano di far coesistere TCP/IP.  
Allocazione slot temporale

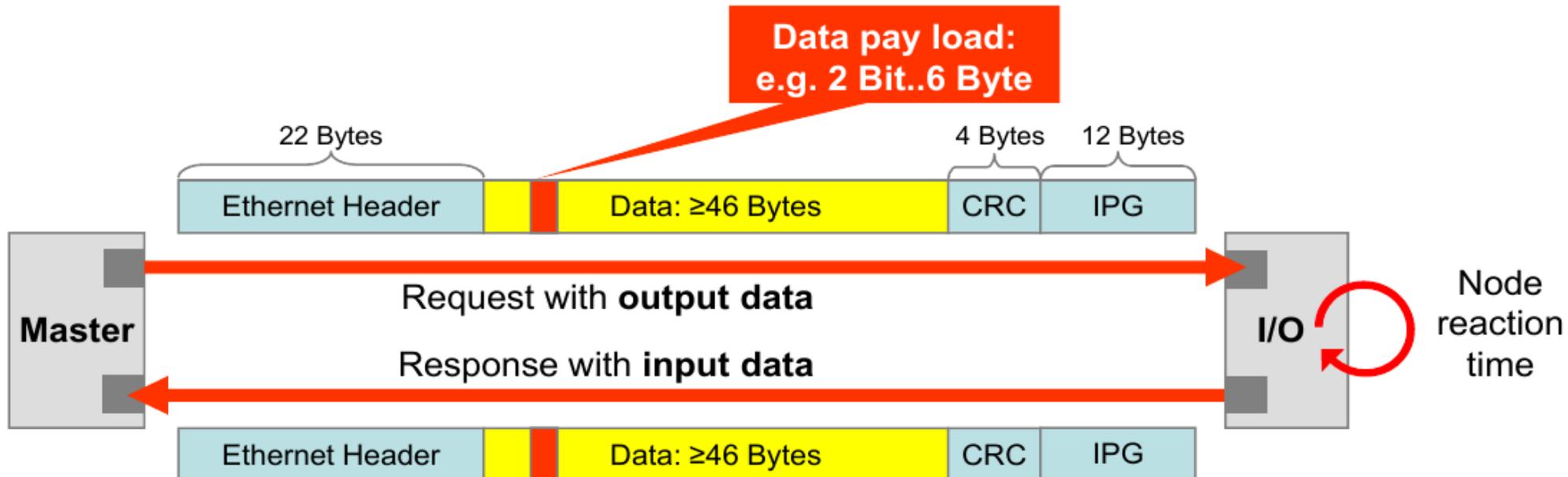
## ■ Il protocollo EtherCAT

- **Basato su Ethernet (SOLO LIVELLO FISICO)**
- **Velocità di Trasmissione:**
  - 2 x 100 Mbit/s (Fast Ethernet, Full-Duplex)
- **Il frame trasmesso ha struttura costante**
  - viene trasmesso dal master
  - attraversa tutti gli slave (tranne l'ultimo) in andata e ritorno fino a tornare al master



## ■ Il protocollo EtherCAT

- **Modello shift-register**
- **Il frame rappresenta l'immagine della memoria di processo**
  - **Contiene i dati di input e output verso i nodi**
- **Gli indirizzi di accesso ai dati nella memoria corrispondono alla posizione dei rispettivi dati nel frame**
  - **Ogni nodo legge e scrive nel frame le informazioni a lui riservate**



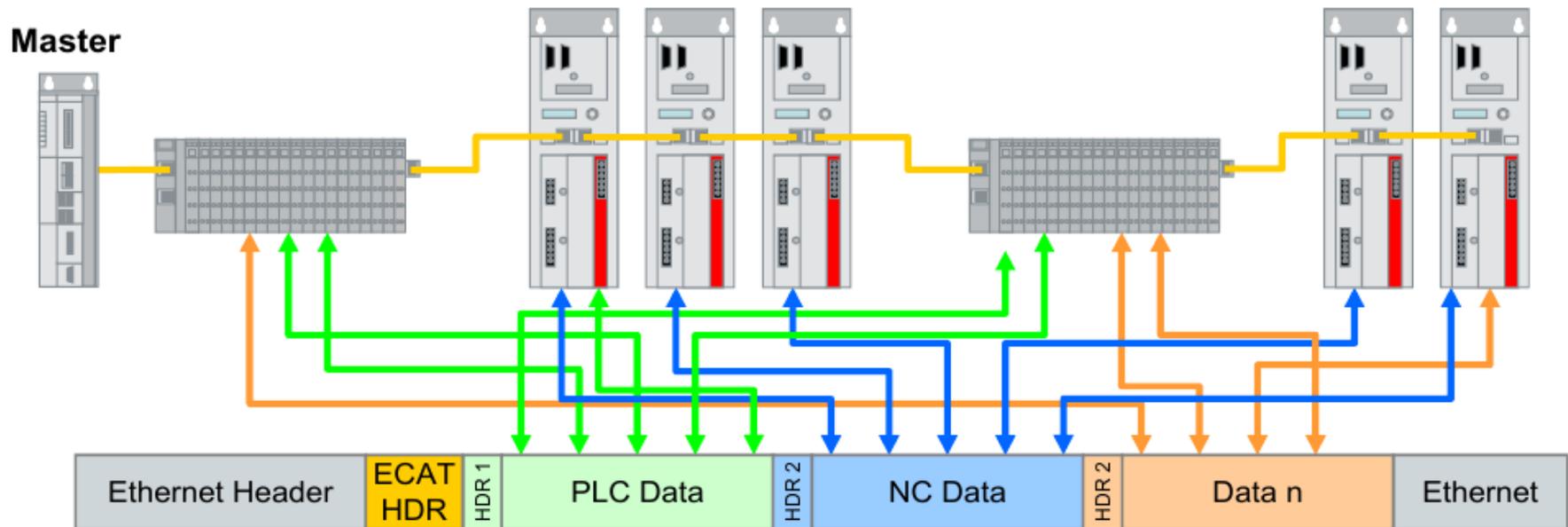


## ■ EtherCAT: principio di funzionamento

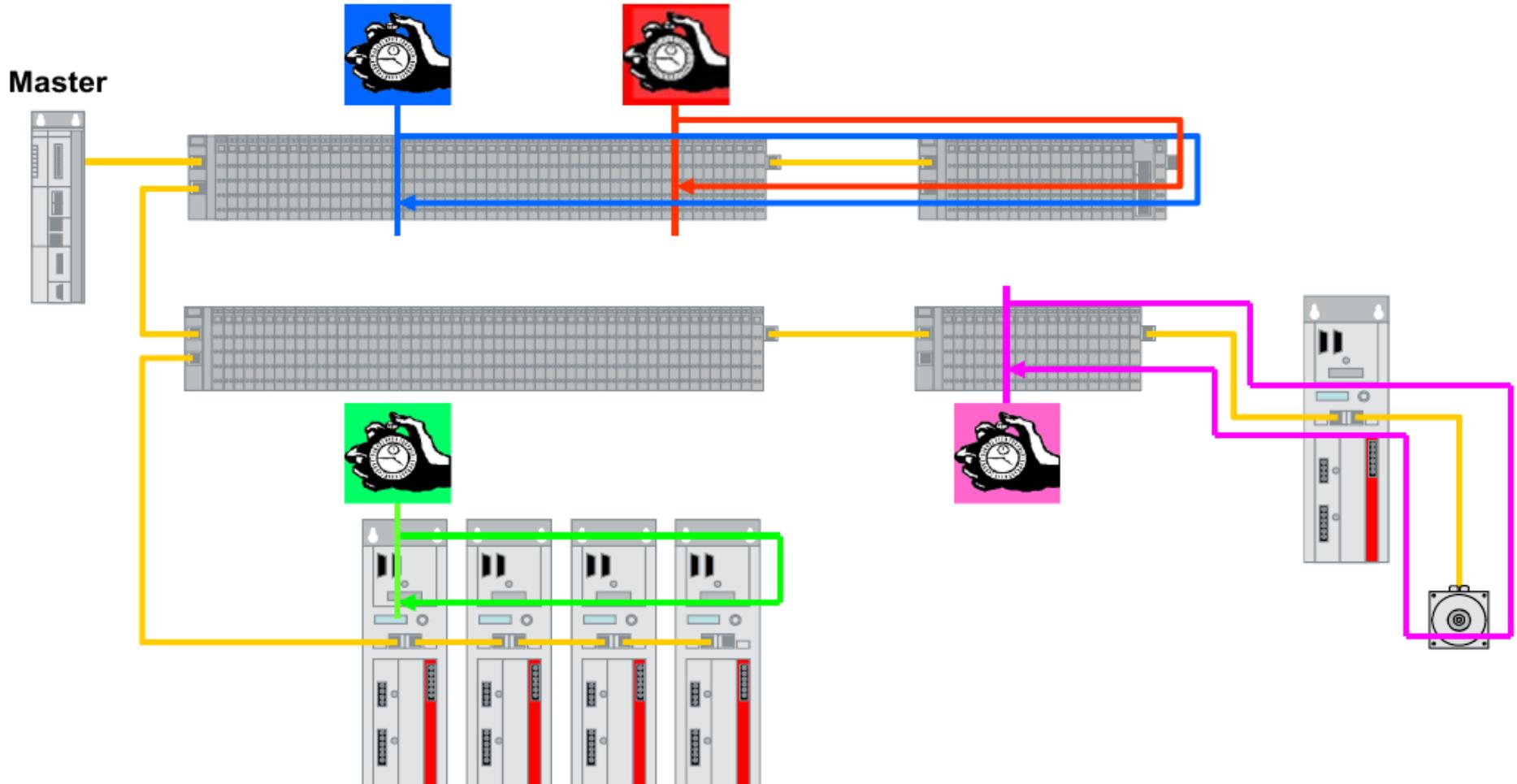
- **“Il Treno” (Frame Ethernet ) non si ferma**
  - Persino osservando il “Treno” attraverso una sottile finestra , si è in grado di veder passare il “Treno” per intero
- **“Vagoni” (Sub-Telegram) hanno una lunghezza variabile**
  - Si possono “estrarre” o “inserire” singole “persone” (Bits) o interi “gruppi” (Bytes) – persino gruppi multipli per treno

## ■ Minimo Overhead del Protocollo mediante l'Indirizzamento

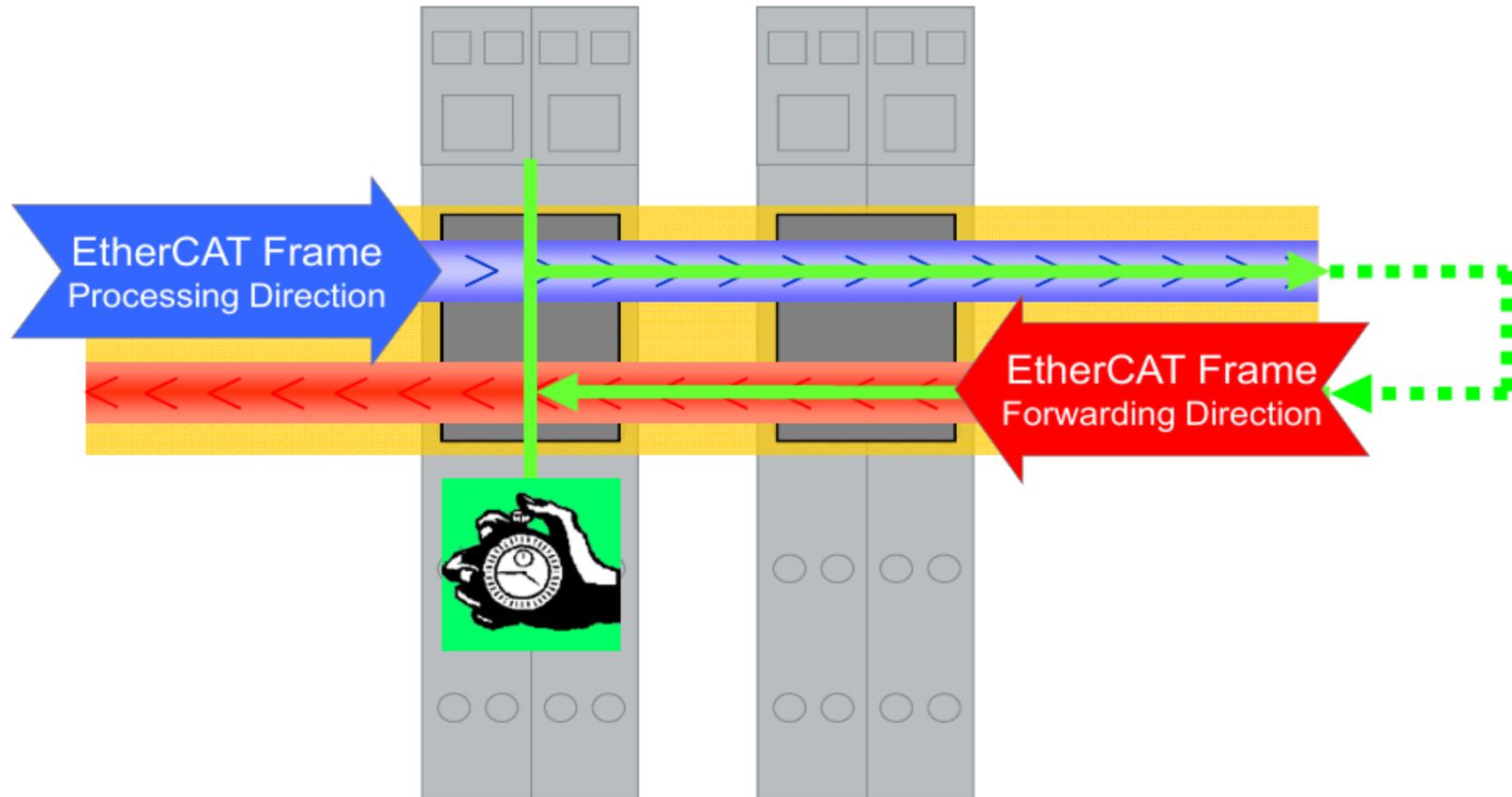
- **Strutture Dati del Telegramma Ottimizzate per gestire l'I/O**
  - Comunicazione realizzata completamente in hardware:
  - massimizzazione delle prestazioni (+ deterministiche!)
- **Nessun switches è necessario se in rete vi sono solo dispositivi EtherCAT**
  - Notevoli caratteristiche di diagnostica
  - Compatibilità Ethernet conservata



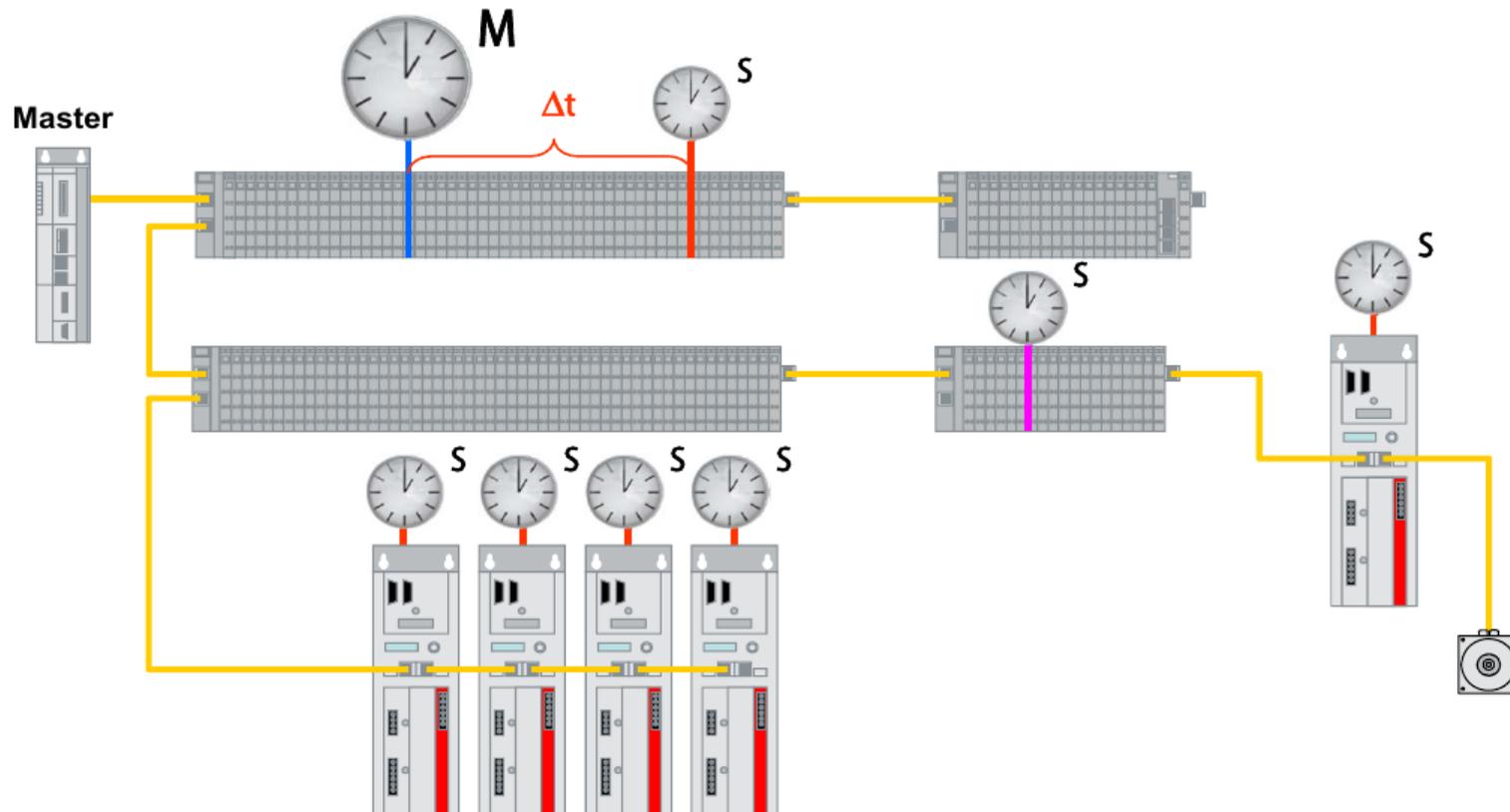
- Ogni nodo EtherCAT può misurare la differenza temporale tra il Frame di Andata e quello di Ritorno



- Ogni nodo EtherCAT può misurare la differenza temporale tra il Frame di Andata e quello di Ritorno
  - Ogni nodo ha un proprio clock interno che viene sincronizzato periodicamente grazie alla misura del ritardo introdotto dalla rete



- Ogni nodo EtherCAT può misurare la differenza temporale tra il Frame di Andata e quello di Ritorno
  - Ogni nodo ha un proprio clock interno che viene sincronizzato periodicamente grazie alla misura del ritardo introdotto dalla rete
    - Accurata Sincronizzazione ( $\ll 1 \mu\text{s}$ !) mediante l'allineamento dei Distributed Clocks



## ■ **Fieldbus Wireless.....**

- **Affidabilità?**
- **Collisioni più complesse**
  - **ci possono essere collisioni locali e non globali sul mezzo**
- **Sicurezza rispetto alle intrusioni dall'esterno?**
- **Alimentazione nodi?**
- **.....**

■ .....

Laurea Magistrale in Ingegneria Informatica  
Laurea Magistrale in Ingegneria Elettronica e Telecomunicazioni per  
lo Sviluppo Sostenibile

**Sistemi e Tecnologie per l'Automazione LM**

**RETI INFORMATICHE NELL'AUTOMAZIONE**

**FINE**

**Ing. Gianluca Palli**

DEI - Università di Bologna

Tel. 051-2093186

E-mail: [gianluca.palli@unibo.it](mailto:gianluca.palli@unibo.it)

<http://www-lar.deis.unibo.it/people/gpalli/>