

**ACHILLE  
CANNAVALE**

# **APPUNTI**

## **MISURE DISTRIBUITE**

**2024**

**CIAO! QUESTI APPUNTI SONO  
FRUTTO DEL MIO STUDIO E  
DELLA MIA INTERPRETAZIONE,  
QUINDI POTREBBERO  
CONTENERE ERRORI, SVISTE O  
COSE MIGLIORABILI. BUONO  
STUDIO 📖 ✎**

# **Sistemi di Misura Distribuiti**

Riassunto da

**Achille Cannavale**

# Indice

<b>1</b>	<b>Introduzione</b>	<b>3</b>
1	Nodo di Misura . . . . .	3
1.1	Sensore . . . . .	3
1.2	Condizionamento Sensori Passivi . . . . .	4
1.3	Condizionamento Circuiti Attivi . . . . .	4
1.4	Eleaborazione e Trasmissione dell'Informazione . . . . .	4
2	Nodo di Misura per più Misure . . . . .	4
2.1	Modo 1 . . . . .	4
2.2	Modo 2 . . . . .	5
3	Architetture Tipiche di Sistemi di Misura . . . . .	6
3.1	Classica . . . . .	6
3.2	Unità di controllo in remoto . . . . .	7
3.3	Senza Rete Locale . . . . .	7
<b>2</b>	<b>Tecnologie di Comunicazione</b>	<b>8</b>
1	Integrazione con diverse Tecnologie di Comunicazione . . . . .	8
2	Lista di alcune Tecnologie di Comunicazione . . . . .	9
2.1	Ethernet . . . . .	9
2.2	IEEE-488 . . . . .	9
2.3	LTE (Long Term Evolution) . . . . .	9
2.4	Bluetooth . . . . .	9
2.5	WiFi . . . . .	9
2.6	ZigBee . . . . .	9
<b>3</b>	<b>Richiami delle Reti Informatiche</b>	<b>10</b>
1	Lista di tipologie di Reti in base alla loro Dimensione . . . . .	10
2	Comunicazione nelle Reti . . . . .	10
3	Tipi di Rete in base alla Topologia . . . . .	10
3.1	Bridge di protocollo . . . . .	11
3.2	Soluzioni software . . . . .	11
4	Soluzioni web-based . . . . .	11
5	Reti Informatiche . . . . .	11
5.1	Tipi di reti . . . . .	11
5.2	Topologie di rete . . . . .	11
6	Dispositivi di rete . . . . .	11
7	Cablaggio di rete . . . . .	11
<b>4</b>	<b>Le Tecnologie di Comunicazione</b>	<b>12</b>
1	Applicazioni in Tempo Reale . . . . .	12
2	CANBus . . . . .	12
3	Profibus e Foundation FieldBus . . . . .	12
4	Migrazione verso Ethernet . . . . .	13
5	Problemi di Collisione su Ethernet . . . . .	13
6	Miglioramento delle Performance . . . . .	13

<b>5</b>	<b>Tecnologie Wireless</b>	<b>14</b>
1	Introduzione . . . . .	14
2	Reti di Comunicazione Cellulare . . . . .	14
3	Canali Radio Dedicati . . . . .	14
4	Tecnologie Wireless a Corto Raggio . . . . .	14
5	La Tecnologia ZigBee . . . . .	14
6	Confronto tra ZigBee e Bluetooth . . . . .	15
7	Nuove Tecnologie Wireless: BLE e Wi-Fi 6 . . . . .	15
8	LoRa e LoRaWAN . . . . .	15
9	Problematiche di Coesistenza nelle Reti Wireless . . . . .	15
10	Conclusioni . . . . .	16
<b>6</b>	<b>Lo Standard IEEE-488</b>	<b>17</b>
1	Bus Standard . . . . .	17
2	Sistema di Interfaccia . . . . .	17
2.1	Il Ruolo dell'Interfaccia . . . . .	17
2.2	Livello Fisico . . . . .	17
2.3	Livello Temporale . . . . .	17
2.4	Livello di Rete . . . . .	17
2.5	Livello di Trasporto . . . . .	18
3	Suddivisione . . . . .	18
4	Il Bus Cable . . . . .	18
4.1	Data Bus . . . . .	18
4.2	General Interface Management . . . . .	18
4.3	Data Byte Transfer Control . . . . .	18
4.4	Temporizzazione della Comunicazione . . . . .	19

# Introduzione

## Sistemi di Misura Distribuiti

Insieme di dispositivi di misura e/o controllo posizionati in punti geografici differenti, tra loro connessi attraverso un’opportuna tecnologia software/hardware di comunicazione

Una delle caratteristiche più importanti dei sistemi di misura distribuiti è quella di poter avere accesso alle strumentazioni, per un qualsiasi motivo (taratura, monitoraggio, misura etc) pur non essendo "sul posto", vicino alla strumentazione.

## 1 Nodo di Misura

Un nodo di misura in un sistema di misura distribuito è in grado di rilevare la grandezza da misurare per poi trasmetterla attraverso un’opportuna rete di comunicazione, ad un utente finale, che non deve per forza trovarsi vicino al punto di misurazione.

In un nodo di misura possiamo trovare quattro blocchi fondamentali:

- **Trasduttore:** converte una grandezza fisica in ingresso, in una grandezza elettrica in uscita
- **Condizionamento:** adatta la grandezza elettrica in ingresso per metterla in ingresso al digitalizzatore (legge solo tensioni)
- **Digitalizzazione:** converte la grandezza da analogico a digitale
- **Elaborazione e Trasmissione dell’Informazione**

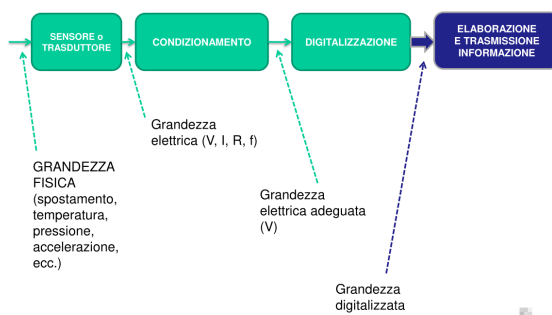


Figura 1.1: Schema di un Nodo di Misura

## 1.1 Sensore

### Sensore

Dispositivo in grado di fornire un segnale di uscita (output) in risposta ad uno specifico segnale di ingresso (input). Il segnale in uscita è sempre una grandezza elettrica, il segnale in entrata può

essere una qualunque quantità, proprietà o condizione chimico-fisica.

La curva caratteristica del sensore (o trasduttore) mi permette di passare dalla grandezza elettrica alla grandezza fisica di interesse per l'utente finale.

Essi, inoltre, si possono classificare a seconda della loro natura, in:

- **Sensori Attivi**, se l'effetto fisico su cui è basato assicura la trasformazione in energia elettrica dell'energia propria del misurando (es: pannelli fotovoltaici)
- **Sensori Passivi**, se l'effetto del misurando si traduce in una variazione di potenziale sull'elemento sensibile

## 1.2 Elaborazione e Trasmissione dell'Informazione

Un particolare tipo di Nodo di Misura è lo Smart Sensor, che è un oggetto con una particolare architettura standardizzata.

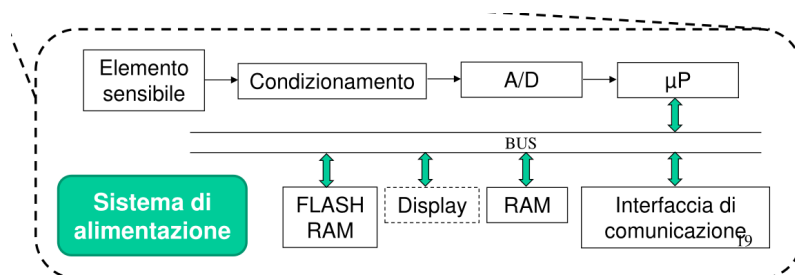


Figura 1.2: Schema di uno Smart Sensor

## 2 Nodo di Misura per più Misure

### 2.1 Modo 1

Creare una catena di blocchi per ogni misura sarebbe davvero inefficiente, quindi cerchiamo di sviluppare il blocco Elaborazione e Trasmissione Informazione e inserendo un multiplexer digitale:

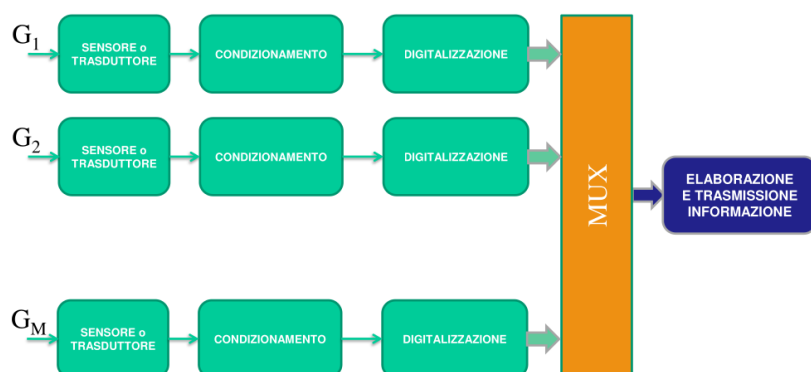


Figura 1.3: Schema più misure (Modo 1)

In particolare, essendoci M digitalizzazioni, tutte le misure saranno campionate allo stesso istante. (es: Positivo per il calcolo dello sfasamento tra due sinusoidi)

### 2.2 Modo 2

Un altro modo per misurare più grandezze e rendere l'architettura più efficiente ed economica è quella di mettere in comune anche la digitalizzazione:

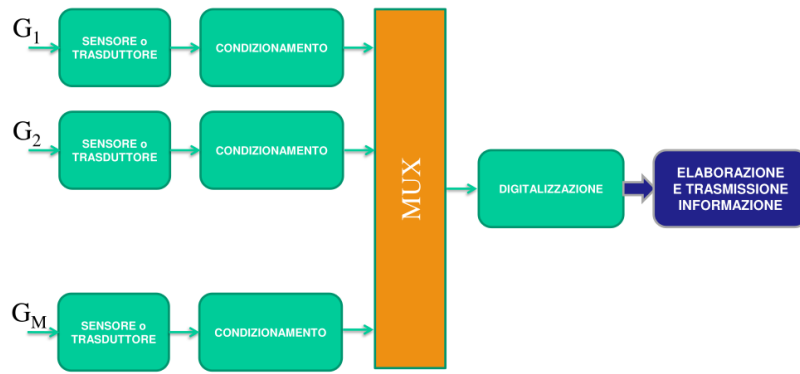


Figura 1.4: Schema più misure (Modo 2)

A questo punto però, i tempi di campionamento saranno diversi, generando il cosiddetto Inter Channel Delay.

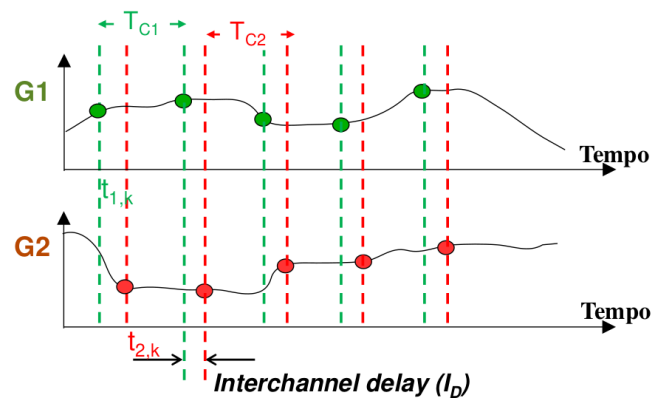


Figura 1.5: a nice plot

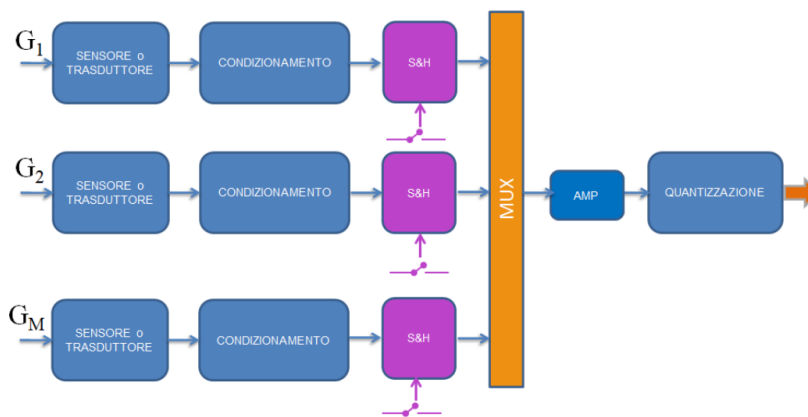
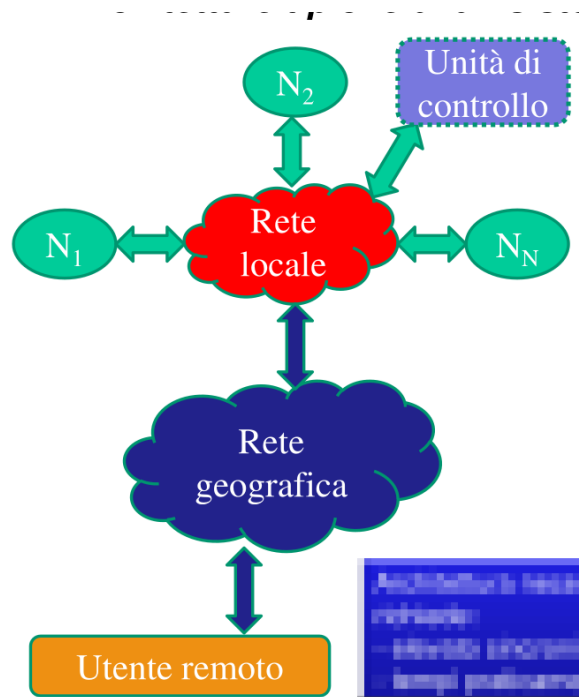


Figura 1.6: a nice plot

Per risolvere il problema dell'Inter Channel Delay possiamo utilizzare dei dispositivi di Sample and Hold, che verranno azionati tutti nello stesso istante, facendo in modo che la quantizzazione verrà fatta ancora con un delay temporale, tuttavia il campionamento sarà fatto nello stesso istante.

### 3 Architetture Tipiche di Sistemi di Misura

#### 3.1 Classica



Il concetto principale dei sistemi di misura distribuiti risiede nella distinzione di alcune entità e reti come:

- Nodi di Misura
- Rete locale: un punto di accesso che collega tutti i nodi di misura, l'Unità di Controllo e gestisce l'interfacciamento con la Rete Geografica.
- L'unità di Controllo: ovvero un sistema (es un pc) che ha come scopo la gestione della procedura di misura.

In uno schema del genere, l'Utente Remoto accede ai nodi tramite l'Unità di Controllo. Quindi funge anche da "filtro" nei confronti dei nodi di misura.

Figura 1.7: a nice plot

#### 3.2 Unità di controllo in remoto

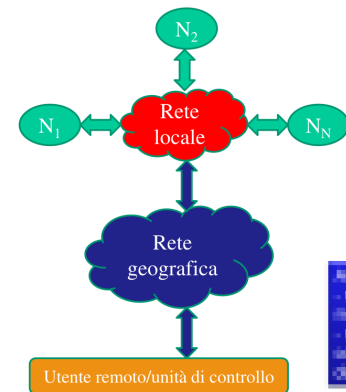


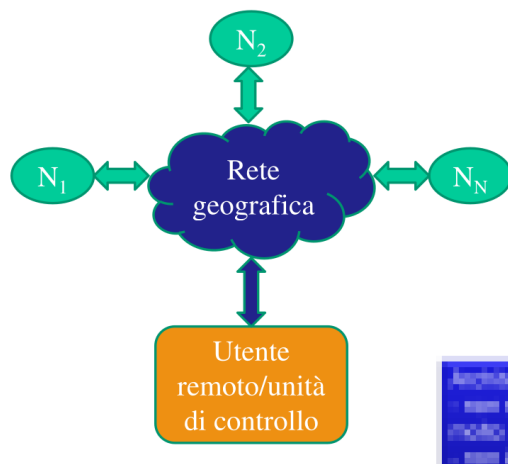
Figura 1.8: a nice plot

In questa architettura, la rete locale collega direttamente i nodi alla rete geografica (ad es. Internet), senza un'unità di controllo locale. È adatta a situazioni in cui non sono richiesti vincoli temporali stringenti o coordinamento tra i vari nodi.

#### 3.3 Senza Rete Locale

In questa configurazione è assente la Rete Locale, quindi, l'unità di controllo in remoto ha la possibilità di accedere direttamente ai nodi.





In particolare, ciascun nodo è dotato tecnologie di comunicazione di rete geografica. La rete geografica introduce alcuni ritardi che non sono programmabili, quindi questa configurazione è adatta per quelle applicazioni che non hanno necessità di una risposta con bassa latenza.

Figura 1.9: a nice plot

## 4 Bridge di Protocollo

I **bridge di protocollo** (o convertitori di protocollo) consentono di collegare dispositivi con interfacce e protocolli differenti senza modificare il livello applicativo. Questi dispositivi sono utili per mantenere l'integrità delle applicazioni esistenti, permettendo la comunicazione tra sistemi che utilizzano protocolli diversi.

## 5 Nodo di Misura e Capacità di Networking

I bridge di protocollo (o convertitori di protocollo) consentono di collegare dispositivi con interfacce e protocolli differenti senza modificare il livello applicativo. Questi dispositivi sono utili per mantenere l'integrità delle applicazioni esistenti, permettendo la comunicazione tra sistemi che utilizzano protocolli diversi.

# Tecnologie di Comunicazione

La prima distinzione da fare tra le tecnologie di comunicazione è tra cablate e wireless. Possiamo poi classificare ulteriormente le varie tecnologie attraverso i seguenti parametri:

- **Estensione Geografica**
- **Topologia di Rete**
- **Numero di dispositivi**
- **Performance Raggiungibili** (es data rate)

Qualitativamente per estensione geografica intendiamo l'area di copertura di queste tecnologie, che può variare da qualche metro a migliaia di km. Per quanto riguarda la topologie di rete, possiamo far riferimento a due topologie principali:

- Rete a stella
- Rete peer-to-peer

Ed infine, come esempio di Performance Raggiungibile possiamo prendere il data rate, ovvero un indice che mostra la quantità di informazione trasmessa in un secondo.<sup>1</sup>

## 1 Integrazione con diverse Tecnologie di Comunicazione

Per far comunicare diversi dispositivi con diverse tecnologie di comunicazione sono utili i cosiddetti "bridge" di protocollo che consentono di mantenere tutto inalterato a livello di applicazione.

---

<sup>1</sup>Più importante è il payload rate

## 2 Lista di alcune Tecnologie di Comunicazione

### 2.1 Ethernet

Ethernet è lo standard delle reti locali ed ha:

- **Estensione Geografica:** praticamente illimitata
- **Numero di dispositivi:** praticamente illimitato
- **Data rate:** fino a 100 Mbps un'estensione geografica praticamente illimitata

### 2.2 IEEE-488

- **Estensione Geografica:** 20m
- **Numero di dispositivi:** fino a 15
- **Data rate:** fino a 1 Mbps

### 2.3 LTE (Long Term Evolution)

- **Estensione Geografica:** praticamente illimitata
- **Numero di dispositivi:** praticamente illimitati
- **Data rate:** fino a 326,4 Mbps

### 2.4 Bluetooth

- **Estensione Geografica:** fino a 100 m
- **Numero di dispositivi:** fino a 64
- **Data rate:** fino a 1 Mbps

In particolare il Bluetooth lavora su una cosiddetta **Banda Non Licenziata**, ovvero una banda su cui i dispositivi possono comunicare gratuitamente.

### 2.5 WiFi

- **Estensione Geografica:** fino a 100 m
- **Numero di dispositivi:** fino a 32
- **Data rate:** fino a 54 Mbps

### 2.6 ZigBee

- **Estensione Geografica:** fino a 100 m
- **Numero di dispositivi:** fino a 65536
- **Data rate:** fino a 250 kbps

## Richiami delle Reti Informatiche

### **Rete Informatica**

Una rete informatica è un insieme di PC e di altri dispositivi che sono collegati tra loro tramite una serie di componenti hardware (cablaggi, ripetitori, ecc.) e componenti software che forniscono gli strumenti per la gestione della rete.

Il sistema deve essere in grado di far comunicare i dispositivi tra loro e possiamo suddividere le reti in base alla loro grandezza.

### **1 Lista di tipologie di Reti in base alla loro Dimensione**

- **LAN** (Local Area Network)
- **WAN** (Wide Area Network)
- **WLAN** (Wireless Local Area Network) [es: WiFi]
- **WPAN** (Wireless Personal Area Network) [es: BT, ZigBee, UWB]

### **2 Comunicazione nelle Reti**

I pacchetti di dati che vengono scambiati tra i dispositivi di una rete, vengono etichettati dagli indirizzi del destinatario e del mittente. Per controllare questo percorso si utilizzano vari protocolli (es: TCP/IP).

### **3 Tipi di Rete in base alla Topologia**

- **Reti Centralizzate**
- **Reti Client-Server**
- **Reti Peer-to-Peer**

Le topologie di rete possono essere suddivise anche in:

- **Topologia Fisica:** definita dalla disposizione fisica del cablaggio e dai dispositivi
- **Topologia Logica:** definita dalla logica di controllo dell'accesso al mezzo trasmissivo e dal modo in cui i dispositivi gestiscono il traffico dei dati in rete

I sistemi di misura distribuiti integrano più tecnologie di comunicazione tramite nodi di misura che si interfacciano con protocolli differenti come RS-232, IEEE-488, Ethernet, ecc.

### 3.1 Bridge di protocollo

- I *bridge* di protocollo permettono l'integrazione di tecnologie wired (USB, Ethernet) e wireless (WiFi, GSM, LTE).
- Questi consentono di mantenere inalterato il livello applicativo e facilitano la comunicazione tra nodi con interfacce diverse.

### 3.2 Soluzioni software

Sono disponibili tre principali soluzioni software per la comunicazione:

1. Software proprietario dedicato, che garantisce determinismo ma poca flessibilità.
2. Software proprietario multipurpose, come LabView, che offre maggiore adattabilità ma richiede licenze.
3. Software open-source, che richiede competenze avanzate ma offre prestazioni ottimizzate senza costi di licenza.

## 4 Topologie di rete

- **Bus:** i dispositivi condividono un singolo cavo, con terminatori alle estremità.
- **Stella:** tutti i dispositivi sono collegati a un nodo centrale (hub).
- **Anello:** i dispositivi sono connessi in un circuito chiuso, con dati che viaggiano in un'unica direzione.
- **Ibrida:** come le applicazioni web-based sono accessibili tramite un semplice browser e non richiedono installazioni aggiuntive, permettendo l'accesso remoto da qualsiasi parte del mondo. bina topologie diverse come stella e bus.

## 5 Dispositivi di rete

- **Hub:** passivi, duplicano i segnali senza elaborazione.
- **Bridge e Switch:** trasferiscono dati e svolgono elaborazioni aggiuntive.
- **Router:** gestiscono traffico tra diverse reti LAN e WAN.

## 6 Cablaggio di rete

- **Cavi coassiali:** disponibili in versioni *Thicknet* e *Thinnet*, con lunghezze massime di 500 e 185 metri rispettivamente.
- **UTP/STP:** cavi a coppie ritorte, protetti o meno dalle interferenze elettromagnetiche.
- **Fibra ottica:** il mezzo di trasmissione più veloce, con velocità che vanno da 100 Mbps a oltre 200 Gbps.

# Le Tecnologie di Comunicazione

## 1 Applicazioni in Tempo Reale

Per Tempo reale possiamo trovare due definizioni, diverse in base ad alcune sfumature:

- **Soft Real Time:** dove i risultati delle elaborazioni devono essere disponibili il più vicino possibile al tempo di scadenza
- **Hard Real Time:** sono tutte quelle applicazioni dove ogni singolo evento deve essere necessariamente gestito all'interno dei limiti temporali

Analizziamo ora alcune parole chiave:

- Il **determinismo**, si riferisce alla capacità di un sistema di rispondere a una richiesta entro un tempo massimo predeterminato. Questo significa che il sistema è in grado di garantire che ogni operazione verrà completata in un tempo limite noto a priori, indipendentemente dalle condizioni o dal carico del sistema.
- Il concetto di "**massima latenza nota**" è fondamentale perché permette di sapere, in anticipo, qual è il tempo massimo entro cui si otterrà una risposta.
- L'**isocronia** si riferisce alla capacità di un sistema di eseguire operazioni in modo strettamente ripetitivo nel tempo, con una variazione minima nel comportamento temporale.
  - Questo implica un "**basso jitter**", dove il jitter rappresenta la variazione o fluttuazione del tempo tra due eventi successivi. In pratica, un sistema isocrono esegue le operazioni con intervalli temporali regolari, minimizzando le deviazioni.

## 2 CANBus

Il CANBus (Controller Area Network) è un protocollo di comunicazione seriale sviluppato per permettere a diversi dispositivi elettronici di scambiarsi dati all'interno di un veicolo senza la necessità di un computer centrale. Introdotto negli anni '80 da Bosch, il CANBus è particolarmente adatto per applicazioni automobilistiche, dove dispositivi come centraline, sensori e attuatori devono comunicare tra loro in modo affidabile e tempestivo.

Il protocollo si basa su una rete bus multi-master, permettendo a più nodi di inviare e ricevere messaggi senza rischio di collisioni, grazie a un meccanismo di arbitraggio delle priorità. Il CANBus garantisce inoltre elevata affidabilità, tolleranza ai guasti e un utilizzo efficiente della larghezza di banda.

## 3 ProfiBus e Foundation FieldBus

Il livello fisico di queste tecnologie segue lo standard IEC 61158, con una velocità di 31.25 kbps e supporto per un massimo di 32 dispositivi per segmento. La rete può essere estesa oltre i 2 km utilizzando ripetitori. ProfiBus utilizza una trasmissione differenziale, con alimentazione disponibile direttamente sul bus. La comunicazione è permanente e ciclica, dove il master comunica con gli slave

in modo sequenziale. Questa tecnologia è particolarmente diffusa in Europa per la sua semplicità e rapidità d'uso.

## 4 Migrazione verso Ethernet

Negli ultimi anni, si è osservata una forte migrazione verso Ethernet, che permette di integrare le reti industriali con quelle aziendali. La tecnologia Ethernet offre diverse velocità: 10 Mbps, 100 Mbps (Fast Ethernet) e 1000 Mbps (Gigabit Ethernet), consentendo la creazione di reti con un numero virtualmente illimitato di dispositivi grazie all'uso di hub, switch e router. Ethernet è ormai lo standard de-facto nel networking, sia in ambito industriale che aziendale, grazie alla sua diffusione, al supporto su varie piattaforme e ai bassi costi di installazione e manutenzione.

## 5 Problemi di Collisione su Ethernet

Le reti Ethernet usano il protocollo CSMA/CD (Carrier Sense Multiple Access/Collision Detection) per evitare che più dispositivi trasmettano simultaneamente, cosa che causa collisioni. Quando si verifica una collisione, il dispositivo attende un intervallo di tempo e ritenta la trasmissione. Questo processo può essere ripetuto fino a 16 volte prima di generare un messaggio di errore. Tuttavia, questo meccanismo è considerato una delle principali cause del non-determinismo di Ethernet.

## 6 Miglioramento delle Performance

Per migliorare le performance di Ethernet, è utile adottare Fast Ethernet o Gigabit Ethernet, che riducono i tempi di attesa e la probabilità di collisione. Si consiglia di minimizzare il numero di nodi per dominio di collisione e di trasmettere pacchetti di dimensioni ridotte. Inoltre, l'uso di protocolli di tipo "event-driven" può contribuire a limitare il traffico non necessario sulla rete.

# Tecnologie Wireless

## 1 Introduzione

Negli ultimi anni, le tecnologie di comunicazione wireless sono diventate onnipresenti, grazie soprattutto all'elettronica di consumo. Esse offrono numerosi vantaggi per i sistemi di misura distribuiti, come la riduzione dei costi di cablaggio e installazione, una maggiore mobilità dei nodi di misura e minori ingombri. Le principali soluzioni includono la trasmissione dati tramite reti cellulari, canali radio dedicati e collegamenti a corto raggio come infrarossi o radiofrequenze.

## 2 Reti di Comunicazione Cellulare

Le reti cellulari utilizzano tecnologie come GSM, GPRS, UMTS e LTE, con diverse velocità e caratteristiche. GSM, progettato per la comunicazione mobile digitale, ha un data rate limitato di 9.6 kbps, mentre GPRS offre una connessione sempre attiva e data rate più elevati. UMTS integra la rete satellitare, permettendo l'accesso a banda larga. LTE, con velocità di download fino a 150 Mbps, migliora ulteriormente le prestazioni della rete cellulare.

## 3 Canali Radio Dedicati

La trasmissione tramite canali radio dedicati è utilizzata per comunicazioni su distanze comprese tra 100 metri e 100 km. Il radiomodem è l'elemento centrale di questa architettura, e la potenza richiesta per la trasmissione può determinare la necessità di una licenza. Generalmente, potenze inferiori a 20 mW e frequenze sotto gli 800 MHz non richiedono una licenza. Un vantaggio significativo è l'assenza di ritardi nella trasmissione rispetto alle reti pubbliche.

## 4 Tecnologie Wireless a Corto Raggio

Le reti wireless personali (WPAN) sono emerse con tecnologie come IrDA, Bluetooth e ZigBee. IrDA, progettata per applicazioni informatiche, offre collegamenti 1-1 limitati, mentre Bluetooth (IEEE 802.15.1) consente connessioni a basso consumo su distanze fino a 100 metri. ZigBee, invece, è ottimizzato per applicazioni a basso consumo, con una velocità di trasmissione fino a 250 kbps e la possibilità di supportare una grande densità di nodi in rete.

## 5 La Tecnologia ZigBee

ZigBee, basata sullo standard IEEE 802.15.4, è progettata per reti WPAN a basso consumo e bassa velocità, ideali per applicazioni di monitoraggio industriale e controllo. ZigBee consente un duty cycle estremamente basso (<1%) che permette ai dispositivi di entrare in modalità di risparmio energetico, garantendo una lunga durata della batteria. I suoi principali vantaggi includono bassi consumi, basso costo e la possibilità di supportare reti con un'alta densità di nodi.



ZigBee utilizza una topologia di rete flessibile, che può essere di tipo *star*, *peer-to-peer* o *cluster-tree*. Nella topologia *star*, un dispositivo funge da coordinatore della rete (PAN Coordinator), mentre gli altri nodi agiscono da dispositivi full-function (FFD) o ridotti (RFD). Nella topologia *peer-to-peer*, i nodi possono comunicare direttamente tra loro, mentre nella topologia *cluster-tree*, i nodi formano una rete gerarchica.

A differenza di Bluetooth, ZigBee utilizza un protocollo molto più leggero (lo stack può essere di soli 32 KB) e ha un consumo estremamente ridotto (Sleep Mode con assorbimento di 1  $\mu$ A), rendendolo ideale per applicazioni che richiedono una lunga autonomia senza interventi frequenti.

## 6 Confronto tra ZigBee e Bluetooth

ZigBee e Bluetooth hanno caratteristiche diverse e si adattano a differenti tipi di applicazioni. ZigBee supporta reti con numerosi nodi e ha consumi molto bassi, mentre Bluetooth è ottimizzato per trasmissioni dati più veloci. Bluetooth, con un data rate di 723 kbps, è indicato per applicazioni che richiedono trasferimenti più intensivi, come lo streaming di dati o file. Invece, ZigBee si distingue per il basso consumo energetico e la possibilità di operare per lunghi periodi con batterie di piccole dimensioni.

## 7 Nuove Tecnologie Wireless: BLE e WiFi 6

Bluetooth Low Energy (BLE) è stato progettato per ottimizzare i consumi energetici, con corrente di picco inferiore a 10 mA e una latenza minima di 3 ms. BLE è particolarmente adatto per applicazioni a bassissima potenza, come i sensori indossabili. WiFi 6, invece, offre una maggiore velocità di trasmissione dati e una gestione ottimizzata di più dispositivi simultaneamente, migliorando anche l'efficienza energetica.

## 8 LoRa e LoRaWAN

**LoRa** (Long Range) è una tecnologia di comunicazione wireless che opera su bande di frequenza sub-GHz, come 868 MHz in Europa e 915 MHz in Nord America, ed è progettata per garantire trasmissioni a lungo raggio con un consumo energetico molto basso. È particolarmente adatta per applicazioni di misura distribuite, come il monitoraggio ambientale, dove la trasmissione può avvenire anche su distanze superiori a 10 km in aree rurali. LoRa supporta una modulazione basata su *chirp spread spectrum* (CSS), che consente di mantenere bassi livelli di potenza durante la trasmissione, anche su lunghe distanze.

**LoRaWAN** (LoRa Wide Area Network) è il protocollo di rete che permette la comunicazione tra i nodi LoRa e un'infrastruttura centralizzata. I nodi di misura, alimentati a batteria, comunicano con i *gateway* che, a loro volta, inviano i dati a un *network server* tramite connessioni standard (ad esempio, reti cellulari o cablate). I *gateway* possono gestire la comunicazione con diversi nodi simultaneamente, e i *network servers* si occupano di filtrare i dati, controllare la sicurezza e ridurre il traffico. Grazie alla sua architettura, LoRaWAN è particolarmente efficace in scenari con molti nodi distribuiti su ampie aree geografiche.

LoRa e LoRaWAN garantiscono un'ottima copertura su lunghe distanze con un consumo ridotto, rendendoli ideali per reti di sensori su vasta scala e applicazioni IoT.

## 9 Problematiche di Coesistenza nelle Reti Wireless

Le reti wireless spesso condividono lo spettro di frequenze con altri dispositivi, il che può causare interferenze. Le sovrapposizioni di frequenza più critiche si verificano tra Bluetooth e WLAN, o tra reti cellulari (LTE o 3G) e TV digitale. Per evitare problemi di coesistenza, è importante considerare le frequenze utilizzate e applicare tecniche come lo spread spectrum per mitigare le interferenze.

## 10 Conclusioni

Non esiste una tecnologia wireless ottimale per tutte le applicazioni. La scelta dipende da vari parametri, come area di copertura, prestazioni, costi di installazione, consumi, sicurezza della trasmissione dati e coesistenza con altre reti. ZigBee e LoRa si distinguono per l'efficienza energetica e l'uso in reti con un gran numero di nodi, mentre tecnologie come BLE e WiFi 6 offrono prestazioni superiori per applicazioni più intensive. Il progettista deve valutare attentamente questi aspetti per scegliere la soluzione più adatta.

# Lo Standard IEEE-488

## 1 Bus Standard

Abbiamo accennato che la standardizzazione permette di generalizzare l'interoperabilità tra dispositivi diversi. In particolare i bus standard sono estremamente flessibili e adattabili a diverse esigenze.

## 2 Sistema di Interfaccia

Questo sistema costituisce un canale di comunicazione sul quale trasferire messaggi tra più dispositivi. I principali componenti di un sistema di misura basato su questo standard sono:

- Diversi strumenti
- Interfacce dei vari strumenti
- Bus cable

In particolare possiamo definire:

- Controller: calcolatore dotato di interfaccia che organizza l'accesso al bus
- Periferiche: strumenti autonomi in grado di effettuare misure anche da remoto, ma dotati di interfaccia

### 2.1 Il Ruolo dell'Interfaccia

L'interfaccia si pone logicamente come tramite tra la Periferica e il Bus. Quindi da un lato è in grado di convertire i segnali dall'interno della periferica verso il bus (traducendoli in segnali standardizzati), mentre nel percorso opposto i segnali standardizzati devono essere tradotti in segnali interpretabili dalla circuiteria interna della periferica.

### 2.2 Livello Fisico

Questo livello definisce le specifiche meccaniche, lo sviluppo geometrico, le grandezze elettriche e altre caratteristiche fisiche del dispositivo.

### 2.3 Livello Temporale

Fissa la metodologia di handshake, ovvero quel protocollo di temporizzazione necessario per rendere la comunicazione affidabile e sincrona (nonostante il bus non sia sincrono) tra due o più interlocutori.

### 2.4 Livello di Rete

Questo livello comprende le specifiche riguardanti le tecniche di instradamento attraverso l'indirizzamento, basata sull'assegnazione di un indirizzo ad ogni periferica connessa al bus cable.

## 2.5 Livello di Trasporto

Convertire i dati provenienti dalla circuiteria del dispositivo e (se è un controller) deve essere in grado di sentire alcune variazioni del bus cable che gli servono per il funzionamento del bus.

## 3 Suddivisione

Le interfacce vengono suddivise in base alle funzioni che svolgono:

- Talker
- Listener
- Controller

Tuttavia alle volte uno stesso dispositivo può svolgere tutte queste funzioni.

## 4 Il Bus Cable

Questo bus ha 24 linee suddivise in;

- 8 linee dette "data bus" dove viaggeranno i dati
- 3 linee per il protocollo di handshake
- 5 linee che servono al controller e alle periferiche per sentire quello che vuole il controller
- 8 linee connesse alla massa

### 4.1 Data Bus

Le otto linee del data bus sono usate per:

- Trasmissione di dati
- Indirizzi
- Istruzioni di programma

Ognuno di questi elementi sono codificati in ASCII mediante 7 bit più un ottavo per il controllo di parità. Questi caratteri sono inviati in modalità bit parallelo.

### 4.2 General Interface Management

Le 5 linee hanno funzioni relative al funzionamento delle interfacce degli strumenti e del controller. Queste linee sono:

- 

### 4.3 Data Byte Transfer Control

Queste linee hanno lo scopo di temporizzare le comunicazioni sul bus, garantendo esenzione di errori, nonostante la presenza di più Listener (un solo talker). Questa cosa si ottiene attraverso la soddisfazione di alcuni requisiti:

- Lo strumento abilitato a trasmettere dal controller (il talker) non inizi a "parlare" quando una o più delle periferiche destinate alla ricezione non sono ancora pronte.
- Da parte delle periferiche, esse devono segnalare di aver completato correttamente la ricezione.

Questo protocollo di comunicazione si basa sull'evoluzione temporale di queste tre linee:

- **NRFD** (Not Ready for Data) e **NDAC** (Not Data Accepted) pilotate dalle interfacce che ricevono la comunicazione
- **DAV** (Data Valid) viene pilotata dalla periferica che trasmette

L'insieme delle commutazioni di queste linee costituiscono il protocollo di handshake.

## 4.4 Temporizzazione della Comunicazione

Il protocollo di handshake dovrà essere in grado di adattarsi alla velocità della periferica più lenta, che sarà probabilmente quella che impiegherà più tempo a comunicare la fine di lettura dei dati. Come abbiamo detto questo bus 488 lavora in logica negata.

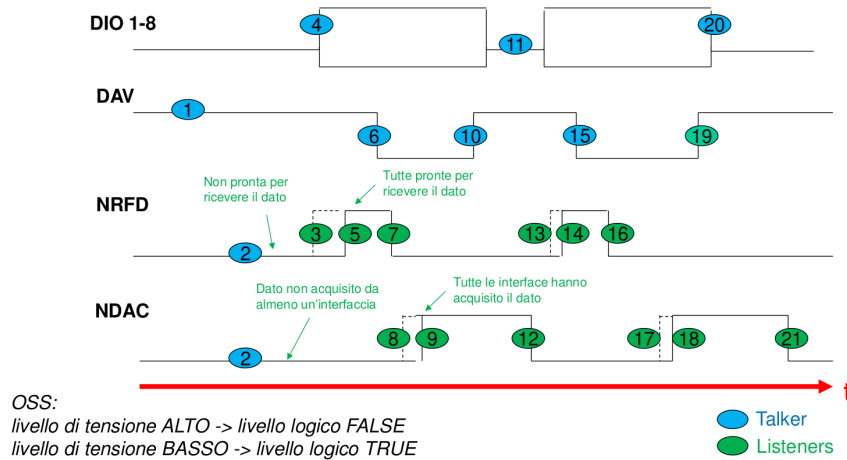


Figura 6.1: a nice plot

- In  $t_1$  il DAV è posto a FALSE
- In  $t_2$  sonda lo stato di NRFD e NDAC, le trova entrambe in livello logico TRUE
- In  $t_3$  la periferica più veloce va ad agire su NRFD ponendola in livello FALSE, tuttavia la linea si alza quando tutt'e le periferiche si alzano al livello logico FALSE ( $t_5$ )
- Nel frattempo, il talker in  $t_4$  ha caricato il bus dati (8 linee) con il carattere. Varia quindi lo stato della linea DAV in  $t_6$  per dire ai Listener di andare a leggere il carattere.
- Per inibire la ricezione di altri dati, la periferica più veloce mette in TRUE la linea NRFD in  $t_7$  (ne basta una sola qui dato che siamo in regime or negato)
- A questo punto le periferiche campionano il dato e comunicano, una volta finito, mettendo a FALSE la linea NDAC, ( $t_8$ ) ma la linea sarà realmente FALSE quando tutte le periferiche alzeranno il livello di tensione in  $t_9$
- A questo punto il talker va a controllare NDAC e se è FALSE, va subito a negare il DAV, per non inviare nuovamente lo stesso dato ( $t_{10}$ )
- A questo punto la periferica più veloce fa ritornare la NDAC a TRUE ( $t_{12}$ )

## Architetture Client-Server

Un'applicazione Client-Server è un'architettura in cui un soggetto detto Client richiede un'azione o un servizio da un fornitore del servizio detto Server.

Un esempio di questa architettura può essere una connessione ad un sito web, in cui il nostro pc è il Client che invia una richiesta ad un Server (sito). A sua volta il Server rimane in attesa di una (o più) potenziale richiesta.

In un'applicazione Client-Server, i vari moduli software non fanno parte dello stesso programma e ogni modulo può essere eseguito anche su diversi hardware.

Supponiamo di avere una configurazione come quella in figura:

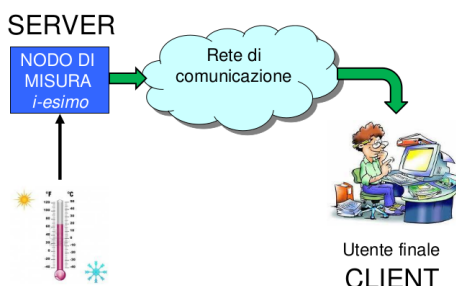


Figura 7.1: a nice plot

Potremo avere tre moduli software separati in questo caso:

- Modulo di acquisizione dati provenienti dal sensore
- Modulo per l'analisi del flusso dati e conversione in gradi
- Modulo per la generazione di un pannello utente di tipo grafico

Una cosa importantissima è che ogni modulo deve fornire i dati in un formato che è comprensibile dal modulo successivo.

La modularizzazione è utile soprattutto per quanto riguarda le modifiche, rendendole più facili e "chirurgiche" da effettuare.

Per quanto riguarda la visualizzazione dei dati possiamo avere due approcci:

- Il server genera una schermata con i dati di misura e la trasmette come immagine al Client
- Il Server invia lo stream dati al client che si occupa di costruire localmente l'interfaccia grafica per la visualizzazione dei dati

Mentre nella prima soluzione il Client visualizza tutto ciò che il Server gli invia, mentre nella seconda situazione eventualmente il Client può decidere di fare altre cose con i dati che gli vengono fornite.

# 1 Considerazioni sul progetto di Client e Server

## 1.1 Considerazione sul progetto del Server

Una delle prime considerazioni risiede nel fatto di domandarsi se il Server in questione deve gestire più Client o uno solo e nel caso in cui debba gestire più Client, in che modo gestirli? Rispetto a che priorità?

Inoltre il Server deve essere in grado di identificare, differenziare e autenticare i servizi di richiesta dei vari Client.

Un'ulteriore cosa che il Server può fare è quella di separare, in maniera modulare, il meccanismo di gestione dei messaggi dalla parte funzionale di elaborazione della richiesta.

## 1.2 Considerazioni sul progetto del Client

- Il Client deve rispettare il protocollo (regole per lo scambio dei dati) utilizzato dal lato Server.
- Deve sapere inviare un messaggio di richiesta. Deve saper interpretare la risposta data dal Server.
- In un'applicazione in cui i Client hanno differenti priorità deve esserci un meccanismo di richiesta che includa alcuni parametri per l'identificazione.

# 2 Richiami sui Protocolli: TCP/IP

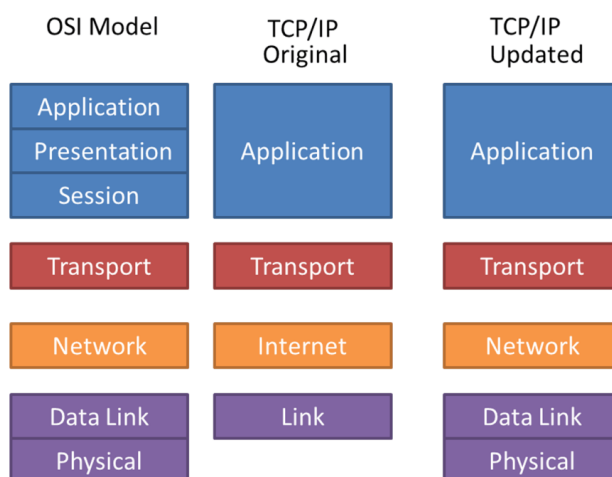


Figura 7.2: a nice plot

Questo è un protocollo orientato alla connessione, quindi l'obiettivo è quello di creare una connessione attraverso la quale scambiare dati in maniera affidabile sul canale.

Tutto ciò si può fare avendo la capacità di creare degli "end points" (detti anche socket) sia in trasmissione che in ricezione. Un socket è formato da tre parametri:

- Indirizzo IP del dispositivo
- Protocollo utilizzato (TCP o UDP)
- Il numero della porta

Il protocollo TCP vede lo stream di dati come una sequenza di byte divisi in vari segmenti, che viaggiano a loro volta sulla rete ognuno in un singolo pacchetto IP. Il controllo di questo flusso è "end-to-end", implementato nel seguente modo:

Se il dispositivo trasmettitore invia dati più velocemente rispetto al tempo necessario al ricevitore per elaborarli, quest'ultimo, invierà un acknowledgement indicante che non c'è più spazio nel buffer. Così il trasmettitore non invierà nuovi dati fino a che il ricevitore non sarà nuovamente pronto.

Tutti questi meccanismi rendono il protocollo TCP abbastanza pesante.

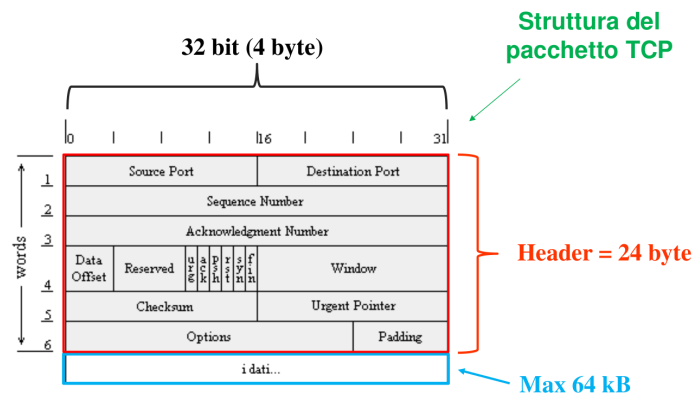
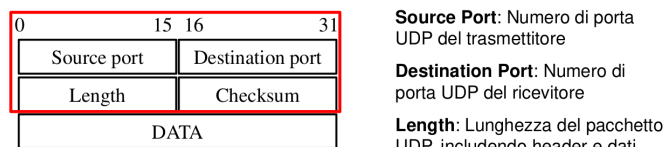


Figura 7.3: a nice plot

### 3 UDP

Rispetto al TCP non prevede la necessità di stabilire una connessione iniziale e non c'è neanche nessun sistema di acknowledgement dei pacchetti persi. Ed è proprio per questo motivo che l'header è più leggero e rende di conseguenza il protocollo UDP più veloce ed efficiente. (esempio video streaming)

Esiste tuttavia un modo per far concordare il sender e il receiver di includere nel pacchetto un controllo dell'affidabilità a livello applicazione.



Header = 8 byte

**UDP Checksum:** E' utilizzato per verificare l'integrità dell'header

Figura 7.4: a nice plot