

CAN: Controller Area Network

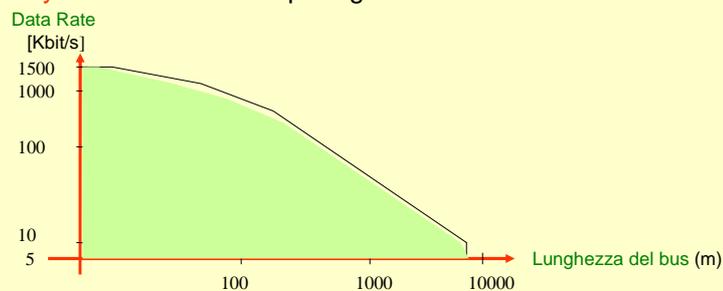
Originariamente concepito per la sostituzione dei cablaggi elettrici relativi a lampade, interruttori, indicatori, ecc. (velocità <10kbps) e per la comunicazione fra le varie Electronic Control Unit (ECU) per il controllo del motore, sospensioni, freni, ecc. (velocità fra 100 kbps e 1Mbps).

Requisiti della rete per applicazioni auto

- Prestazioni **Real-time**
- Velocità di trasmissione **fino a 1Mbps**
- **Flessibilità** di configurazione
- **Sicurezza elevata** in ambiente rumoroso
- Capacità di **connettere qualunque punto** dell'auto, con minima estensione del mezzo fisico
- **Basso costo**

CARATTERISTICHE PRINCIPALI

- **Bus seriale asincrono Multi-master**
- Data bit rate fino a **1 Mbps**
- Lunghezza del bus fino a **40 metri**
- Numero di nodi **illimitato** (almeno in teoria)
- **2032** differenti tipi di messaggi
- da **0 a 8 bytes** di informazione per ogni frame



Lunghezza del Bus e Bit rate

CARATTERISTICHE PRINCIPALI

- **Accesso al bus di tipo CSMA/CD** modificato
 - arbitraggio a livello di bit
 - arbitraggio basato su priorità
 - non distruzione del messaggio a più alta priorità
- **Massimo tempo di latenza garantito** per i messaggi ad alta priorità
- **Elevato livello di sicurezza**
 - rilevazione di errore
 - segnalazione e recupero di errore
 - confinamento di errore
- **Distinzione fra errori temporanei e guasti permanenti**, e disconnessione automatica dei nodi guasti

Il CAN è un sistema Multimaster. Ogni nodo può iniziare a trasmettere appena il Bus è libero.

Il **MESSAGGIO** con la più alta priorità ottiene l'accesso al bus

La priorità è definita dall'identificatore del messaggio.

Il numero di nodi è illimitato (a meno delle limitazioni di ordine fisico) perché non esiste un identificatore di nodo (quindi nuovi nodi possono essere aggiunti senza nessuna modifica Hw/Sw).

Sono i messaggi ad avere dei nomi individuali (fino a 2032 identificatori) che indicano il significato del dato.

Grazie alla comunicazione multicast tutti i nodi possono ricevere ed utilizzare i dati. **Un processo di filtraggio** permette di selezionare i dati di interesse per ogni nodo.

Sistemi CAN indipendenti possono operare a differenti velocità, ma in un sistema CAN tutti i nodi operano alla stessa velocità.

Stati di funzionamento

Un nodo CAN può essere in uno dei seguenti stati:

- Error-Active
- Error-Passive
- Bus-Off

Error-Active è lo stato normale di un nodo funzionante correttamente: un nodo Error-Active partecipa alla comunicazione. Se riscontra un errore, trasmette una **Active Error Frame**.

Un nodo che generi o rilevi molti errori va nello stato Error-Passive: anch'esso può prendere parte alle comunicazioni; in caso di errore trasmette una **Passive Error Frame**.

Se il guasto è di lunga durata, il nodo passa nello stato Bus-Off. Non partecipa attivamente alle comunicazioni ma continua a monitorare il bus: se il guasto cessa, esso torna nello stato Error-Active.

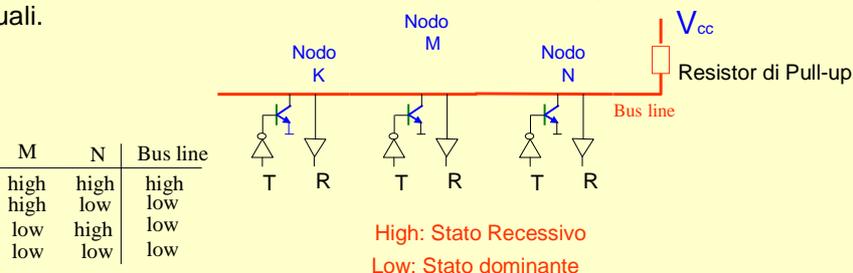
Lo stato di un nodo è definito dal valore di due contatori di errore **TX counter**, **RxCounter** e da un insieme di regole di gestione.

Codifica dei bit e sincronizzazione

I bit sono codificati con una **tecnica on-off** denominata **NRZ**(non-return-to zero).

Poiché il clock non è trasportato col segnale, la sincronizzazione è realizzata sul fronte del bit **"start of frame"**. Per evitare la perdita di sincronizzazione occorrono transizioni periodiche nei livelli del segnale.

Viene utilizzata una tecnica di **bit stuffing**: un bit di segno diverso ogni 5 bit uguali.



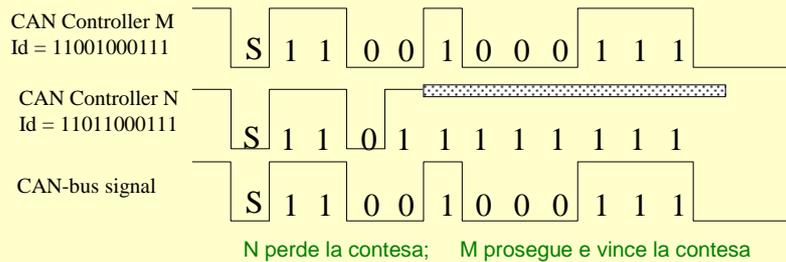
- **Stato Recessivo** del Bus: tutti i trasmettitori inviano un livello alto.
- **Stato dominante**: almeno un trasmettitore invia un bit basso.

Arbitraggio del Bus

L'accesso al Bus è permesso solo se il bus è libero (recessivo).

Ogni accesso inizia col bit Start-of-frame (dominante).

Se due nodi iniziano a trasmettere simultaneamente, il conflitto è risolto mediante un processo di arbitraggio durante la trasmissione dell'**Arbitration Field**.



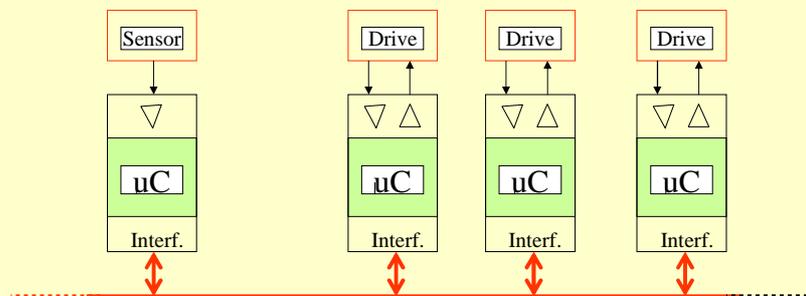
Il processo di arbitraggio non è distruttivo.

La frame a priorità più alta non viene danneggiata.

Configurazione di un sistema CAN

Un tipico sistema CAN comprende diversi micro controllers per l'interfacciamento con i sensori e gli attuatori ed una CPU per il controllo dell'intero sistema.

In alcuni nodi, il microcontrollore è sostituito da una CPU che implementa anche funzioni di controllo.

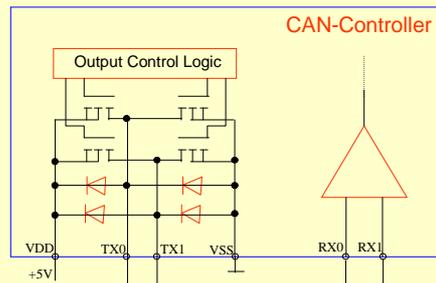


PHYSICAL LAYER

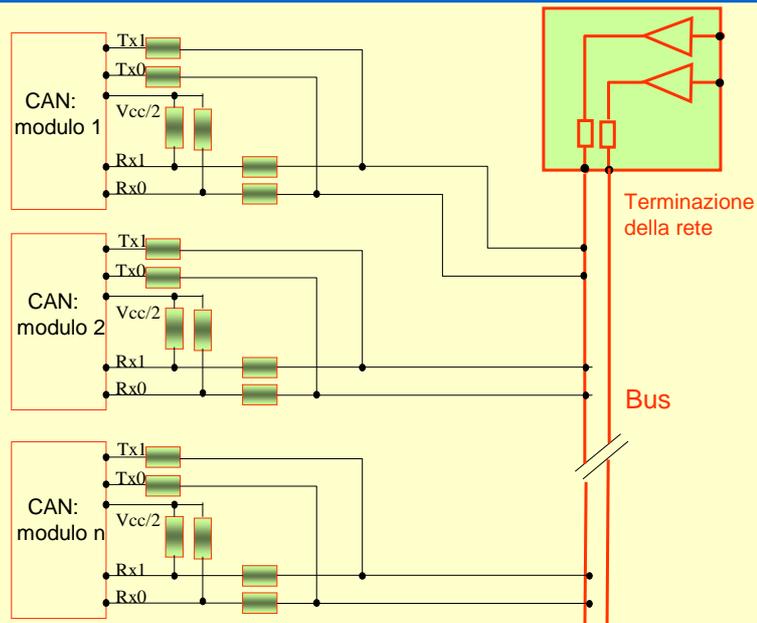
Il Physical layer non è parte dello standard CAN può essere scelto dall'utente, tenendo conto di **immunità ai disturbi**, **sovratensioni**, possibili **rottore del mezzo**, **radiazioni**, **potenza assorbita**, **velocità di trasmissione**, **lunghezza del cavo** e **costi**.

Mezzi consigliati sono: **cavo coassiale**, **doppino intrecciato**, **fibra in vetro** o **plastica**.

I chip CAN attualmente disponibili hanno un **transceiver on chip**, con una **configurazione a ponte di transistor** per la trasmissione ed un **comparatore** per la ricezione dei segnali differenziali.



Realizzazione Elettrica delle rete



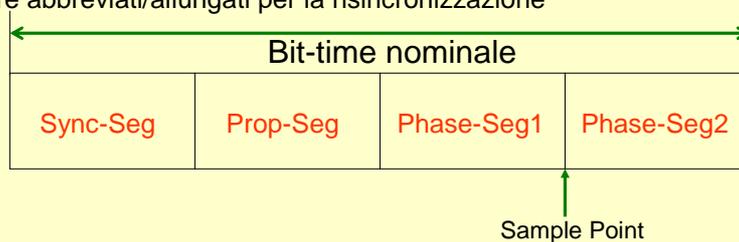
Codifica/decodifica dei Bit

Il Bit-time è definito come la durata di un bit. Esso è diviso in 4 segmenti non sovrapponibili:

Synchronization Segment: è usato per sincronizzare le varie ECUs. La transizione deve avvenire entro questo intervallo.

Propagation Time Segment: è usato per compensare i ritardi nel mezzo fisico.

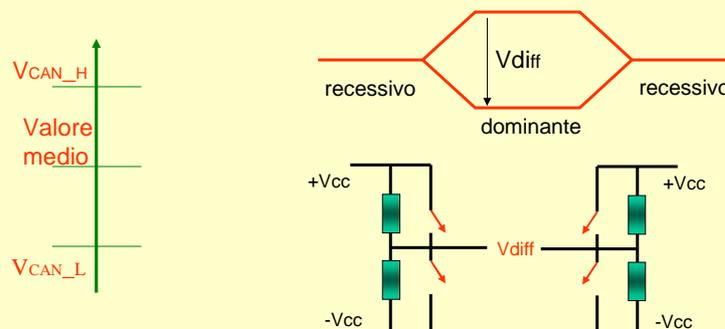
Phase Buffer Segment 1/Phase Buffer Segment2: sono usati per compensare errori di fase dovuti ai fronti di salita/discesa. Possono essere abbreviati/allungati per la risincronizzazione



Livelli del Bus

Il bus utilizza una alimentazione differenziale per il suo funzionamento.

- Nello stato recessivo V_{CAN_H} e V_{CAN_L} sono fissati ad un valore medio della tensione. La V_{diff} è circa zero.
- Nello stato dominante V_{diff} è superiore ad una soglia minima prefissata.



Durante la fase di arbitraggio poiché varie ECUs possono trasmettere simultaneamente un bit dominante, il valore di V_{diff} può variare

Servizi del Physical layer

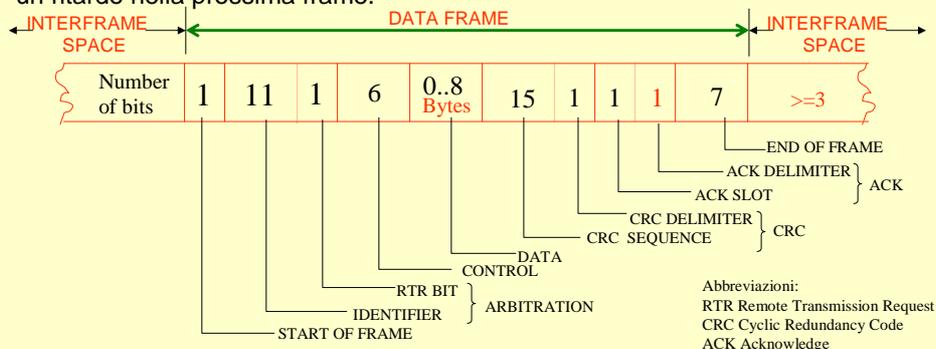
Permettono al MAC locale di scambiare dati col MAC remoto, e controllare le modalità operative del nodo. Sono disponibili i seguenti servizi:

- PLS-data,
- Bus_OFF,
- Bus_OFF_release.
- PLS_data request: richiede la trasmissione di un bit (recessivo/dominante)
- PLS_data indication: permette di passare all'entità MAC il bit appena ricevuto.
- Bus_OFF request: è usata dal Supervisore del nodo per disconnettere il nodo dal Bus.
- Bus_OFF response: passata al supervisore dopo che il nodo si è disconnesso.
- Bus_OFF_Release request: è usata dal Supervisore del nodo per chiedere la riconnessione del nodo al Bus.
- Bus_OFF_Release response: passata al Supervisore dopo che il nodo è passato nello stato operativo "Normal Mode".

Protocollo MAC del CAN

Quattro tipi di frame:

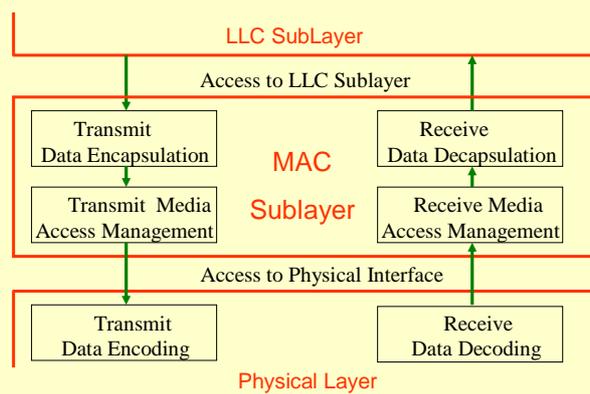
- **Data Frame:** usata per spedire dati (0..8 bytes) da un trasmettitore a tutti i ricevitori. Il campo "Identificatore" ed il campo "Dati" sono passati dall'LLC.
- **Remote Frame:** usata per chiedere dati ad un'altra stazione. Non contiene il campo dati.
- **Error Frame:** usata per informare gli altri nodi che è stato rilevato un errore.
- **Overload Frame:** Segnala che un certo ricevitore è sovraccarico e necessita di un ritardo nella prossima frame.



Architettura del MAC

Le funzioni del MAC sono supportate da 4 moduli base:

- Transmit Data Encapsulation
- Receive Data Decapsulation
- Transmit Media Access Management
- Receive Media Access Management



Primitive di Servizio del MAC

Il MAC Sub-layer fornisce all'LLC tre servizi base:

Acknowledged Data Transfer

MA_DATA.request MA_DATA.indication MA_DATA.confirm

Acknowledged Remote Data Request

MA_REMOTE.request MA_REMOTE.indication MA_REMOTE.confirm

Overload Frame Transfer

MA_OVLD.request MA_OVLD.indication MA_OVLD.confirm

LLC Sublayer

Sono disponibili due tipi di servizi connectionless per l'LLC user:

- data transfer service
- remote data request service

Sono inoltre disponibili i seguenti servizi per la gestione dell'attività

<i>Dall'utente all'LLC</i>	<i>Significato</i>
Reset_Request	riporta il nodo nello stato iniziale

<i>Dall'LLC all'utente</i>	<i>Significato</i>
Reset_Response	Risposta alla Reset_Request
Node_Status	Indica lo stato attuale del nodo, cioè se il nodo è ON od OFF

Servizi dell'LLC

Unacknowledged Data Transfer Service

L_Data.request	Request for data transfer
L_Data.indication	Indication of data transfer
L_Data.confirm	Confirm of data transfer

Unacknowledged Remote Data Request Service

L_Remote.request	Request for remote data transfer
L_Remote.indication	Indication of remote data transfer
L_Remote.confirm	Confirm of remote data transfer

I seguenti parametri sono associati con le primitive di servizio

LLC Service Primitive Parameters

IDENTIFIER	Identifier the data and its priority
DLC	Data Length Code
DATA	data the user wants to transmit
TRANSFER_STATUS	Confirmation parameter

STRUTTURA DELLA FRAME LLC

Una LLC Data frame è composta da tre campi:

- **Identifier field:** lunghezza 11 bit. I 7 bit più significativi non possono essere tutti 1.
- **Data Length Code:** lunghezza 4 bit. Deve contenere valori tra 0..8.
- **Data Field:** da 0..8 bytes.

Una LLC Remote Frame è composta di 2 campi:

- **Identifier Field.**
- **Data Length Code Field.**

Funzioni del sublayer LLC

L'LLC supporta le seguenti funzioni:

Frame Acceptance Filtering

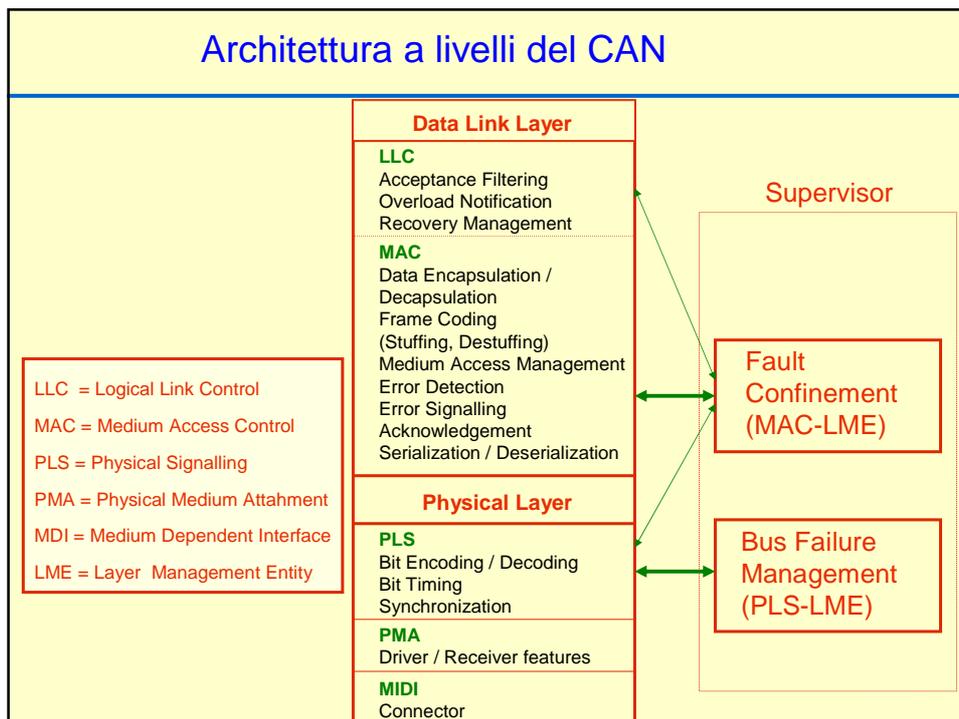
- Il contenuto della frame è individuato dal suo **identificatore**
- l'identificatore non specifica una destinazione ma descrive il significato del dato.
- Ciascun ricevitore decide se la frame è di interesse o meno (**filtraggio** nell'LLC)

Overload notification

- Viene attivata dall'LLC quando un ricevitore non riesce a smaltire le frames ricevute
- Al massimo possono essere generate 2 overload frames consecutive

Recovery Management

- L'LLC provvede alla ritrasmissione automatica delle frames che hanno perso l'arbitraggio o sono state disturbate da errori durante la trasmissione.
- L'utente non riceve una **confirm** finchè la trasmissione non è stata conclusa con successo.



Interframe Space

Le **Data** e/o **Remote Frame** sono separate tra loro e da eventuali **Overload Frame** ed **Error Frame** precedenti, da una sequenza di bit chiamata **Interframe Space**.

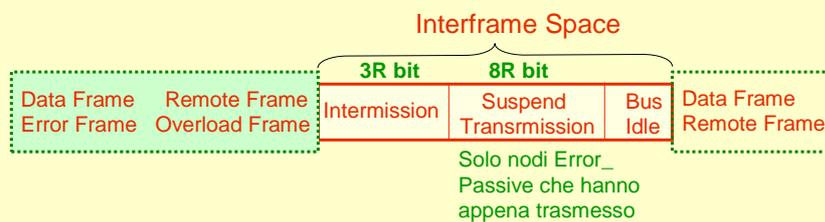
L'Interframe Space è composto da **3 R bit** denominati **Intermission** e, per i nodi Error-Passive che sono stati trasmettitori della frame precedente, da altri **8 R bit** che prendono il nome **Suspend Transmission**.

Questi due campi di bit possono essere seguiti dal campo **Bus Idle**, composto da un numero indefinito di R bit, denotanti la disponibilità del bus.

Interframe Space

L'accesso al bus da parte di un nodo può avvenire :

- durante l'Intermission per inviare una Overload Frame
- subito dopo l'Intermission o durante il Bus Idle se il nodo è Error-Active oppure anche Error-Passive ma non ha trasmesso la frame precedente
- dopo il campo Suspend Transmission se il nodo è Error-Passive e ha trasmesso la frame precedente
- in un qualsiasi istante nel caso in cui venga individuato un errore



Gestione degli Errori

IL MAC si occupa della rilevazione, segnalazione e gestione degli errori

Lo stato di un nodo dipende dal valore dei contatori:

- **Error-Active** se entrambi i contatori sono **minori di 128**
- **Error-Passive** se uno dei due contatori è **maggiore o uguale di 128**
- **Bus-Off** se TxCounter è **maggiore o uguale di 256**

Inizialmente TxC & RxC = 0 e il nodo si trova nello stato Error-Active.

- Se un nodo Error-Active riscontra un errore, trasmette delle Active Error Frame e incrementa TxCounter o RxCounter
- Se non si riscontra alcun errore, il relativo contatore viene decrementato.

Il tasso di incremento dei contatori (normalmente 8) è generalmente maggiore di quello di quello di decremento (solitamente 1)

Gestione degli Errori

Un trasmettitore **Error-Active** affetto da un guasto, si porta velocemente nello stato **Error-Passive**, bloccando in questa fase qualunque altro scambio di messaggi sul bus (a causa dell'invio delle proprie Active Error Frame

Una volta nello stato **Error-Passive**, l'invio di ulteriori **Error Frame** (Passive Error Frame) non è in grado di bloccare il bus.

In **Error-Passive**, un nodo risulta penalizzato in trasmissione dovendo attendere un intervallo di tempo addizionale (il campo **Suspend Transmission dell'Interframe Space** - 8 R bit) prima di trasmettere.

Se il contatore TxCounter raggiunge od oltrepassa il valore 256, il nodo nello stato **Bus-Off** e risulta praticamente sconnesso dalla rete CAN. Esso tuttavia continua a monitorare il bus; se riceve 128 frame corrette, ritorna nello stato Error-Active azzerando TxCounter e RxCounter.

Gestione degli errori

Il CAN può rivelare 5 tipi di errore:

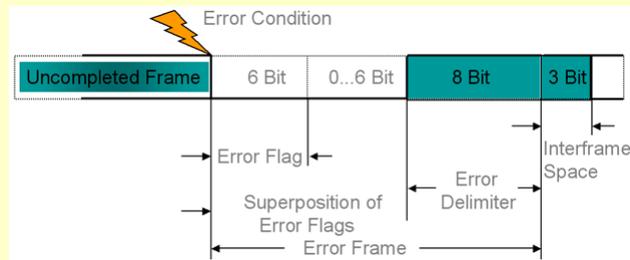
- **Bit Error**: ogni nodo monitora la linea durante la propria trasmissione. Se il bit ricevuto è diverso da quello trasmesso viene segnalato un errore. (eccezione: durante la fase di arbitraggio).
- **Stuff Error**: quando in un messaggio codificato si trovano sei bit dello stesso valore.
- **CRC Error**: dopo il calcolo del CRC.
- **Form Error**: violazione del formato della frame.
- **ACK Error**: è rivelato quando il trasmettitore non monitora un bit dominante durante la finestra di ACK.

Quando il MAC rivela uno di questi errori informa l'LLC e trasmette un error Flag:

- Active Error Flag (6 bit dominanti)
- Passive Error Flag (6bit recessivi)

- L'Error Flag + Error Delimiter costituiscono una Error Frame.

Active Error Frame



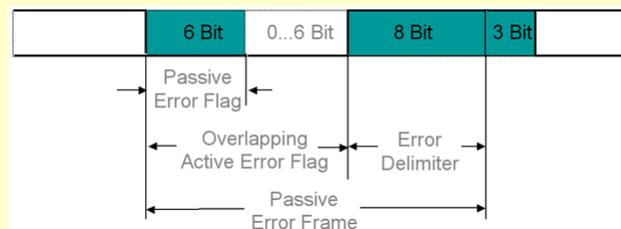
La Error Frame consiste di un Error Flag e di un Error Delimiter Field.

I bit dominanti dell'Error Flag sovrascrivono la frame corrotta e causano una ritrasmissione.

Poichè l'errore può essere rivelato da più nodi contemporaneamente, può verificarsi la sovrapposizione di più Error Flags.

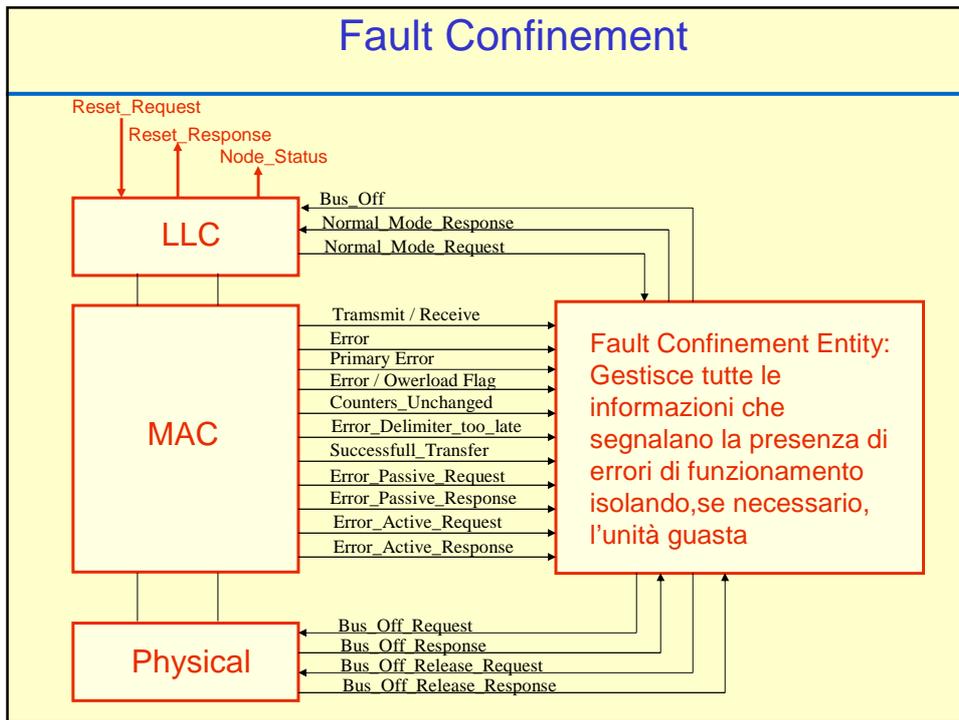
L'Error Delimiter consiste di 8 recessive bits e permette ai nodi del bus di ricominciare una comunicazione corretta dopo l'errore.

Passive Error Frame



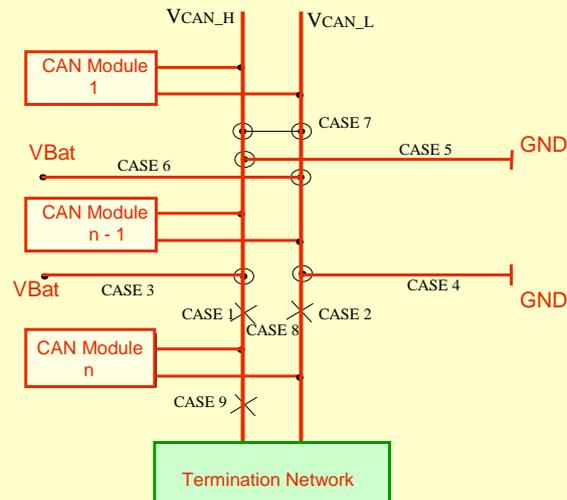
- Nel caso di **nodi Error passive**, (che possono essere nodi faulty o fortemente disturbati) per evitare che il Bus sia bloccato dalle error frames spedite da essi, **la loro capacità di segnalazione è ridotta**.
- Mentre un nodo Error Active può trasmettere un dominant Error Flag, un nodo Error Passive è abilitato a spedire solo Flag recessivi.
- Durante la trasmissione dell'Error Delimiter un nodo può rivelare se esso è il primo della rete a riportare l'errore e quindi bloccare la propria trasmissione.
- I ricevitori Error Passive non possono interrompere una trasmissione in corso poichè un Error Flag recessivo non può influenzare i livelli sul bus.
- Un trasmettitore Error Passive può interrompere il proprio messaggio inviando un passive Error Flag.

Fault Confinement



Bus Failure Management

Durante l'attività della rete possono manifestarsi diversi tipi di guasto che influenzano l'operatività della rete



Bus Failure Management

Description of Bus Failures	Behaviour of the Network	Quality of Specification
One node becomes disconnected from the bus	The remaining nodes continue communicating	Recommended
One node loses power	The remaining nodes continue communicating with reduced S/N	Recommended
One node loses ground	The remaining nodes continue communicating with reduced S/N	Recommended
Loss of the shield connection at any node	All nodes continue communicating	Recommended
<u>Open and Short Failures:</u> (1) CAN-H interrupted (2) CAN_L interrupted (3) CAN_H shorted to battery (4) CAN_L shorted to ground (5) CAN_H shorted to ground (6) CAN_L shorted to battery	All nodes continue communicating with reduced S/N	Recommended
(7) CAN_L wire shorted to CAN_H wire	All nodes continue communicating with reduced S/N	Optional
(8) CAN-H and CAN_L wires are interrupted at the same location	No operation within the complete system: nodes within the resulting subsystem that contains the terminal network, continue communicating	Recommended
(9) Loss of one connection to the termination network	All nodes continue communicating with reduced S/N	Recommended

Meccanismo di Controllo delle Priorità nel CAN

La rete CAN supporta per le comunicazioni in tempo reale un meccanismo basato su uno speciale campo numerico chiamato **campo Identifier (IF)**, che è incluso in ogni frame trasmessa.

L'informazione codificata in questo campo è usata sia dall'**LLC** che dal **MAC**, con diverso significato e scopo:

- ❑ Il contenuto dell'**IF** è interpretato come il livello di priorità della frame corrente quando esso è elaborato dal sottostrato **MAC** ed è usato per risolvere in modo non distruttivo qualsiasi collisione di messaggi sul bus.
- ❑ Il frame *acceptance filtering function* del sottostrato **LLC** elabora invece l'informazione dell'**IF** in modo da identificare i pacchetti trasmessi e spedirli ai programmi applicativi corrispondenti.

CONDIZIONI CRITICHE DI TRAFFICO

Sfortunatamente, l'approccio descritto precedentemente è stato concepito per reti supportanti principalmente traffico periodico. Deve essere inoltre osservato che:

- ❑ Quando la rete diviene **temporaneamente sovraccarica**, il meccanismo base del CAN impedisce la trasmissione di un imprecisato numero di oggetti con alto identificatore numerico.
- ❑ Quando il traffico proposto approssima la larghezza di banda nominale del sistema, c'è un significativo aumento negli **Access Delay** di un certo numero di stazioni.

Meccanismi a priorità dinamica

Il **punto chiave** per garantire un buon comportamento ad una rete CAN è permettere un **riassegnamento dinamico** del valore codificato nell'**IF**.

Cioè dovrebbe essere possibile usare una priorità di **MAC** per la frame trasmessa diversa da quella elaborata dall'**LLC**. Di seguito esamineremo due differenti meccanismi, analoghi al CAN Standard, per riassegnare le priorità dinamicamente, vale a dire:

- ❑ **Coda a Priorità Distribuita**
- ❑ **Promozione di Priorità**

Caratteristiche comuni dei due meccanismi

- ❑ Sia il **DPQ** che il **PP** fanno uso di un **formato esteso per l'identificatore** per mantenere un alto livello di compatibilità con i sistemi CAN convenzionali.
- ❑ I bits più significativi dell'IF codificano la priorità della frame che è utilizzata dal sottostrato **MAC** per imporre una risoluzione deterministica delle contese sul bus.
- ❑ I bits meno significativi dell'IF contengono l'**identificatore effettivo dell'oggetto trasmesso**, che è utilizzato dalla funzione "**frame acceptance filtering**" dell'**LLC** per decidere se le frames ricevute sono relative alla stazione oppure no.
- ❑ Il campo **priority class (PC)** della Frame codifica il **service priority**. Sono definite le seguenti **4 classi di priorità** :
 - **Time Critical**
 - **High**
 - **Low**
 - **Time Available**

DISTRIBUTED PRIORITY QUEUE

Il **meccanismo DPQ**, di assegnazione dinamica della priorità, **si basa sull'utilizzo di una coda FIFO globalmente distribuita**, all'interno della quale tutti i nodi della rete sono virtualmente inseriti.

Ogni nodo non deve memorizzare una copia dell'intera coda, ma **solo la propria posizione**, (che indicheremo con **L.QP**). Tale posizione verrà aggiornata dinamicamente durante l'evoluzione della rete.

La risultante **politica di priorità** è molto semplice:

- ❑ La posizione nella coda distribuita è la priorità corrente della stazione, quindi, più le stazioni sono vicine alla cima della coda, maggiore è la loro priorità nella trasmissione delle frames. Tale meccanismo va ripetuto per tutte e 4 i possibili livelli di priorità

Algoritmo di gestione della Coda Distribuita (2)

2. Se la stazione riesce a trasmettere la frame aggiornerà la propria *Queue Position* **al valore massimo permesso**.
3. Se si ha un errore di trasmissione o la fase di arbitraggio è persa la frame viene ritrasmessa. La *Station Priority* è abbassata solamente quando la frame è effettivamente trasferita a destinazione.
4. Ogni altro nodo della rete legge la frame sul bus e analizza la sua *Priority Class* e la *Queue Position* del trasmettitore.
5. Ogni ricevitore che segue il trasmettitore nella coda aumenta di una posizione in direzione della cima della coda.

Affinché una stazione possa trasmettere non ha bisogno di trovarsi in cima alla coda. Essa è utilizzata solo per risolvere la contesa sul bus in modo deterministico.

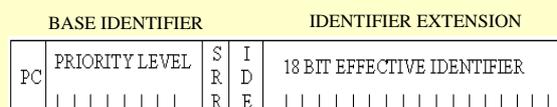
PRIORITY PROMOTION (PP)

L'arbitraggio a Promozione di Priorità è basato sull'idea di assegnare ad ogni stazione **un livello di priorità (*Priority Level*) che è progressivamente incrementato proporzionalmente al tempo trascorso per la stazione prima di poter trasmettere la propria frame sul bus.**

POLITICA DI PRIORITA' NEL PP

- Il *Priority Level* è la priorità corrente delle stazioni (*Station Priority*):
 - ❑ Più tale valore sarà piccolo, maggiore sarà la precedenza della stazione nella trasmissione delle frames.
 - ❑ Si assume che il minimo *Priority Level* corrisponda al valore numerico 300 (minima *Station Priority*).
 - ❑ Può accadere che due o più stazioni abbiano lo stesso *Priority Level*, in questo caso la stazione la *Station Priority* sarà ricavata confrontando pure il valore di un altro campo della frame, chiamato *Effective Identifier*.

FORMATO DI UNA FRAME DEL PP



- ❑ Il Campo **PC** (2 bits) ha la stessa funzione che aveva nel DPQ.
- ❑ Il Campo **Priority Level** (9 bits) codificherà il livello di priorità della stazione per quella particolare classe di priorità. Tale valore sarà utilizzato durante la fase d'arbitraggio.
- ❑ Il Campo **Effective Identifier** (18 bits) viene utilizzato sia durante la fase d' arbitraggio per permettere di risolvere il caso in cui due o più stazioni abbiano la stessa *PC* e lo stesso *Priority Level*, sia dall' LLC per determinare se la frame è destinata alla stazione oppure no.

Algoritmo di gestione dei Priority Level (1)

Supponiamo che F sia una frame e che i suoi campi siano indicati usando una notazione puntata. Per esempio **F.PC** è il campo *Priority Class* della frame, mentre **F.PL** è il *Priority Level* del trasmettitore.

L' **algoritmo di gestione dei Livelli di Priorità** può essere così riassunto:

1. Ogni volta che una frame deve essere spedita sul bus la stazione trasmettente:
 - ❑ Scrive la *Service Priority* **S.PC** nel campo della frame **F.PC**, in questo modo viene definita la *Priority Class* che essa dovrà avere.
 - ❑ Scrive la variabile **S.PL** nel campo **F.PL** , in questo modo la frame conterrà pure il *Priority Level* della stazione.

Algoritmo di gestione dei Priority Level (2)

2. Se la stazione riesce a trasmettere la frame, aggiorna il proprio *Priority Level* settandolo al valore numerico massimo permesso (**S.PL = 300**).
3. Ogni volta che una stazione subisce una collisione con altri nodi che trasmettono frames alla stessa classe di priorità (**S.PC**) e perde la contesa, aumenta il proprio *Priority Level* decrementandolo di una unità (**S.PL[S.PC] = S.PL[S.PC] -1**).
4. Ogni altro nodo della rete legge la frame sul bus e analizza la propria *Priority Class* e il proprio *Priority Level*.

Considerazioni Conclusive sul PP

- Se due o più nodi con la stessa *PC* e lo stesso *Priority Level* iniziano la loro trasmissione nello stesso istante, la contesa è risolta considerando il valore codificato nel campo *Effective Identifier*. **In questo caso il PP mostra un piccolissimo grado di unfairness residuale.**

VALUTAZIONE DELLE PRESTAZIONI

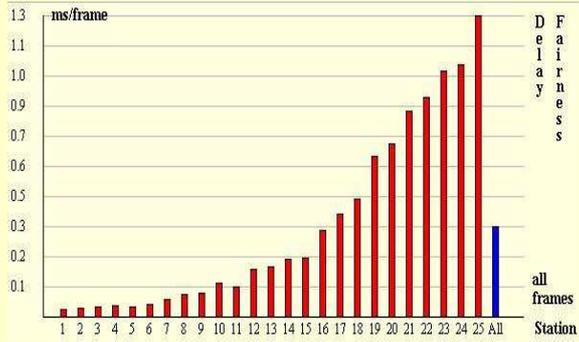
Distribuzione degli Access Delays medi per le singole stazioni.

Esaminiamo una rete costituita da 25 stazioni in condizioni di traffico medio:

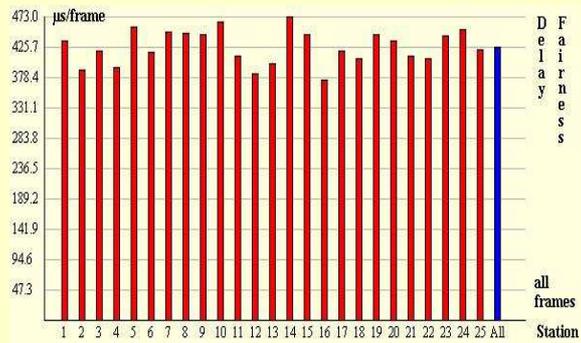
- Si assume un Workload, uguale per tutte le stazioni, di 150 f/s sia per il traffico ciclico che aciclico.
- Tempo di simulazione 2000 secondi.
- Numero di simulazioni 15.
- Lunghezza delle frames 80 bits.

Nel caso di rete CAN Standard e CAN PP si assume che gli Identificatori statici delle variabili sono fissati in ordine crescente.

CAN STANDARD



CAN DPQ



CAN PP



CONSIDERAZIONI

L'utilità di utilizzare i meccanismi a priorità sono evidenti infatti se andiamo a vedere i risultati globali delle tre diverse reti possiamo notare come:

- Pur conducendo ad un certo degrado di prestazioni, i meccanismi a priorità dinamica assicurano che tutte le stazioni ricevano lo stesso trattamento da parte della rete.
- Il meccanismo CAN Standard conduce invece ad una gestione *unfair* della rete che porta, le stazioni con identificatore statico più alto, ad avere degli *Access Delay* nettamente più alti delle altre stazioni.

Diversa Distribuzione delle Collisioni

Come fanno i meccanismi a priorità dinamica a garantire a tutte le stazioni lo stesso trattamento?

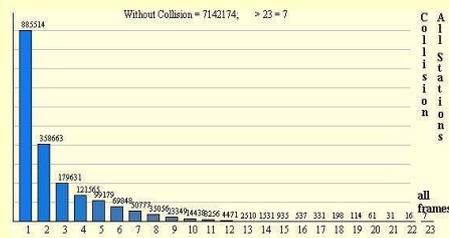
Analizziamo il caso particolare di una rete costituita da 10 stazioni per valori di traffico medio e alto e verificheremo che:

- ❑ Nel CAN Standard non c'è limite al numero di collisioni che una frame può subire prima di essere trasmessa.
- ❑ Nei due meccanismi a priorità dinamica se tutte le stazioni trasmettono frames alla stessa *Priority Class* una frame può subire al massimo un numero di collisioni pari a:

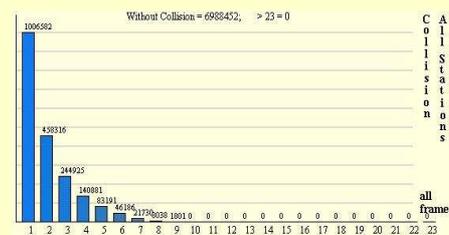
$$\text{Num_Stations} - 1$$

TRAFFICO MEDIO

CAN

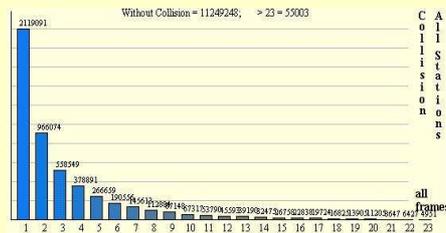


CAN DPQ

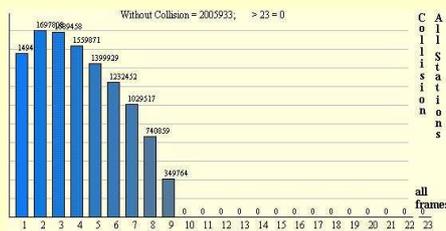


TRAFFICO ALTO

CAN



CAN PP



Application Layer

L'**SDS (Smart Distributed System)** è un Application Layer ideato per applicazioni di automazione industriale che richiedono dispositivi come sensori fotoelettrici, sensori di prossimità, limit switches, valvole elettropneumatiche, relè, controllori, ecc.

I vari dispositivi vengono modellati come degli oggetti (**Object Models**) e di questi vengono specificati:

- **attributi**: ciascuna specifica di attributo include **Attribute_Id**, **Name**, **tipo di dato** rappresentato (Read Only oppure Read/Write), **descrizione**.
- **Eventi**: sono messaggi non sollecitati, generati dal modello del dispositivo; ciascun evento è descritto da **Event_Id**, **Name**, **Type of Data**.
- **Azioni**: ciascuna specifica di azione include **Action_Id**, **Name**, **Input/Output Parameter**, **descrizione**.

Servizi dell'Applicazioni Layer SDS

Vengono usate quattro primitive **request**, **response**, **indication**, **confirm**.

I servizi disponibili sono:

Read: permette di leggere il valore dell'attributo di un dispositivo

Write: permette di modificare il valore dell'attributo di un dispositivo

Event Report: permette di notificare un evento

Action: permette all'AL-service-user di comandare un dispositivo affinché esegua un'azione.

Change of State ON: permette di notificare il cambio di stato a ON di un modello.

Change of State OFF: permette di notificare il cambio di stato OFF di un modello.

Write ON State: permette di settare a ON lo stato di un dispositivo di I/O.

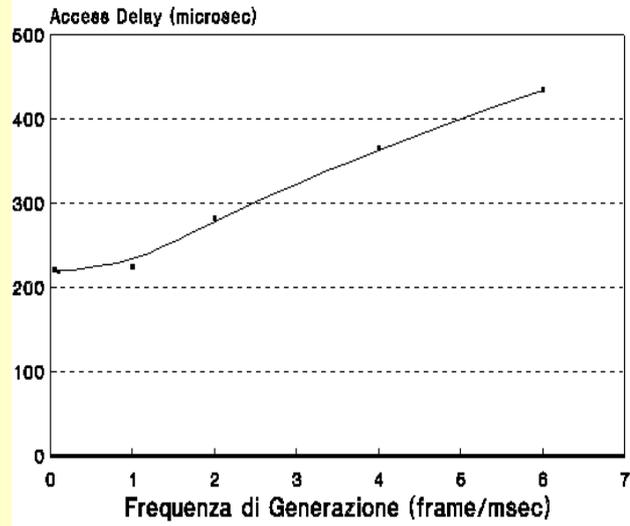
Write OFF State: permette di settare a OFF lo stato di un dispositivo di I/O.

Un esempio: il servizio Read

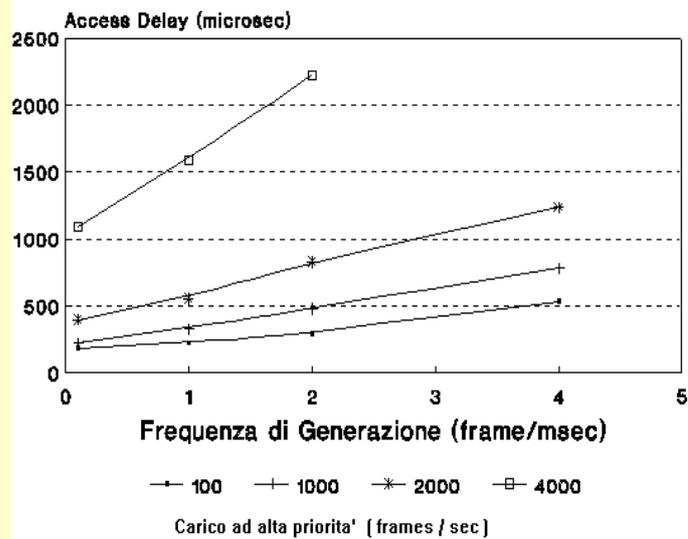
Il servizio **Read** è usato per leggere il valore di un attributo di un dispositivo SDS (ad es. : leggere il valore corrente di un sensore).

PARAMETER	Request	Indication	Response	Confirm	FUNCTION
Address	Mandatory	Come Request	Mandatory	Come Response	...
Logical Device Address	Mandatory	Come Request	Mandatory	Come Response	Definisce il dispositivo dal quale l'attributo deve essere letto
Embedded Device Id	Mandatory	Come Request	Mandatory	Come Response	Specifica quale embedded device deve essere letto.
Attribute Id	Mandatory	Come Request	Mandatory	Come Response	Specifica l'attributo che deve essere letto.
					Il valore del parametro è definito nel modello del dispositivo dell'embedded device.
Result (+) Attribute Id Attribute Value	Selective Mandatory	Come Response	Se la lettura ha avuto successo. Essa ritorna Attribute Id e il valore che è stato letto.
Result (-) Attribute Id Error Code	Selective Mandatory	Come Response	Se la lettura è fallita. Essa ritorna Attribute Id e l'Error Code che specifica il motivo del fallimento.

TRAFFICO AD ALTA PRIORITA'

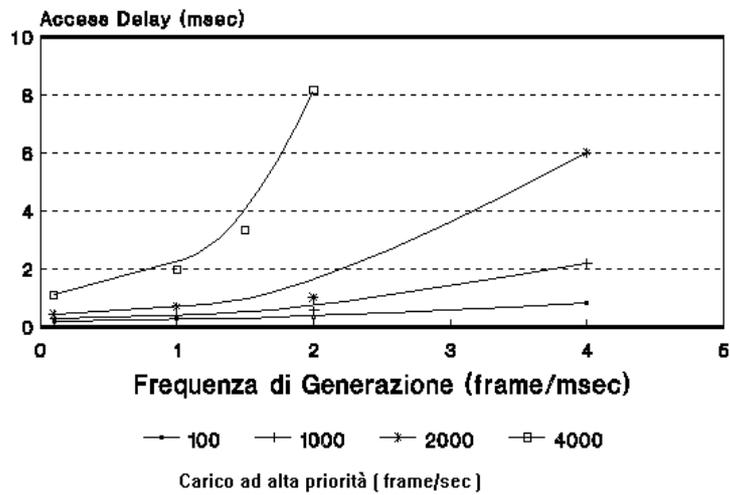


TRAFFICO A MEDIA PRIORITA'



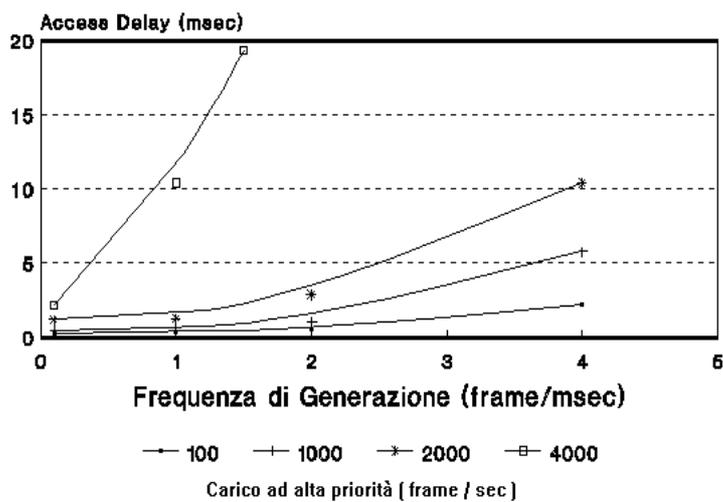
TRAFFICO A BASSA PRIORITA'

carico a media priorità = 100 frame/sec



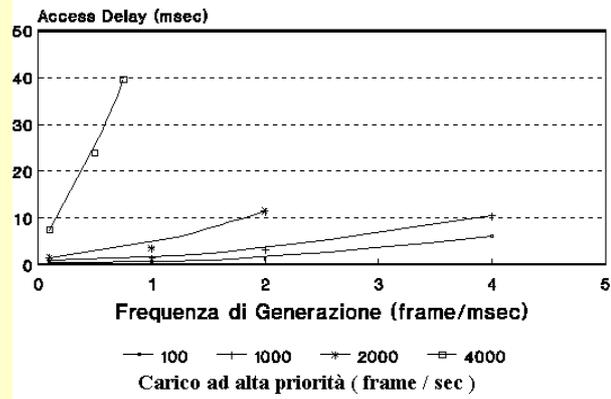
TRAFFICO A BASSA PRIORITA'

carico a media priorità = 1000 frame/sec



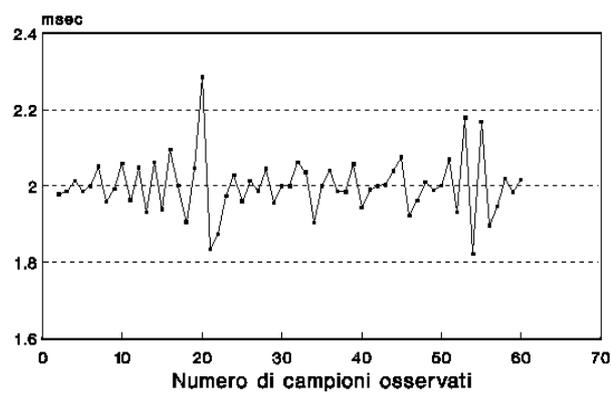
TRAFFICO A BASSA PRIORITA'

carico a media priorità = 2000 frame / sec

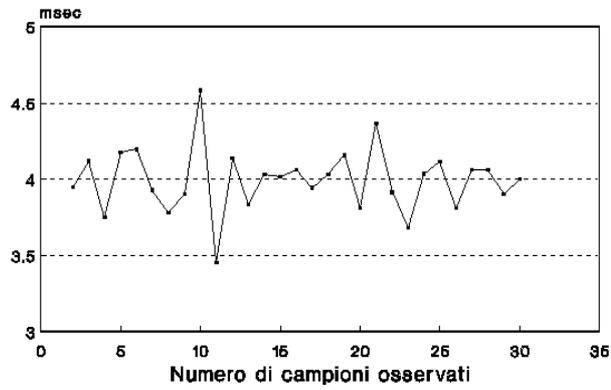


MISURE DI JITTER

traffico ciclico a 2 ms



MISURE DI JITTER traffico ciclico a 4 ms



MISURE DI JITTER traffico ciclico a 6 ms

