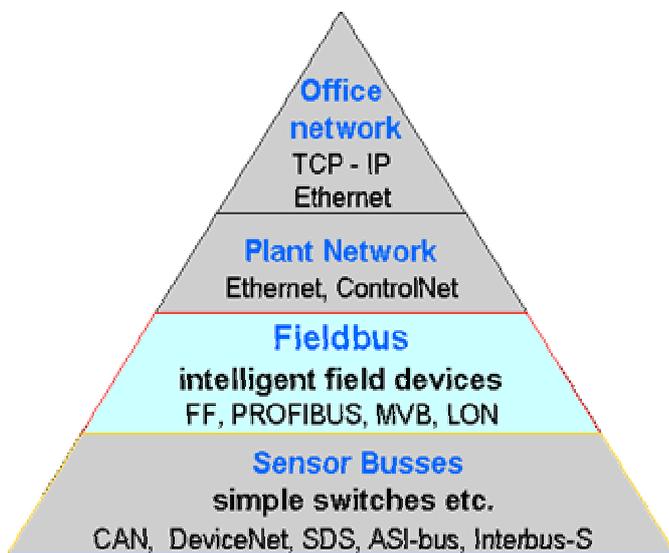


INTERBUS-S

il Bus Sensore/Attuatore

1. Introduzione

Come abbiamo visto durante il corso, attualmente sono presenti sul mercato diversi tipi di Fieldbus con caratteristiche molto diverse, adatti a supportare i diversi tipi di processi industriali. Con riferimento al modello gerarchico usato per rappresentare le interazioni fra le varie attività nel controllo di processo, le diverse esigenze di comunicazione dei vari livelli richiedono reti in grado di supportare Throughput e Tempi di ritardo diversi. Come è possibile vedere nella figura, il più basso livello della gerarchia del controllo è chiamato "Sensor Level" e supporta la comunicazione fra dispositivi semplici che si caratterizzano per lo scambio di informazioni di piccola dimensione (da 1 bit a pochi bytes) che necessitano di brevi tempi di consegna. I diversi tipi di Fieldbus che vengono utilizzati in questo livello si caratterizzano proprio per la loro abilità di operare in maniera efficiente con dati di piccola dimensione. Fra i vari fieldbus l'Interbus-S si distingue per alcune peculiarità che lo rendono molto originale:



- E' tipicamente utilizzato nelle applicazioni di tipo Manufacturing o Batch che sono caratterizzate dalla presenza di molti sensori digitali che scambiano informazioni di piccola dimensione. Non è adatto al controllo di processi continui.
- Controllo Centralizzato non solo per quello che riguarda la comunicazione, ma anche per quanto riguarda la capacità di elaborazione che è concentrata nel Master.
- Struttura Hw che lo rende

paragonabile ad uno Shift Register distribuito in cui i dati fluiscono in modo sequenziale dal Master verso gli Slaves.

- Topologia ad anello (molto rara nei Fieldbus) per cui i dati, dopo un giro rientrano nel Master, in genere modificati dalla presenza di dati spediti dagli Slaves al Master.
- Protocollo di tipo Single Frame, in cui il Master con una sola Frame spedisce i dati a tutti gli Slave e ne riceve indietro le relative risposte (quando necessarie). Questo aspetto del protocollo permette di ottenere elevati livelli di efficienza, soprattutto operando con dati di piccola dimensione.

2. Traffico dati dei sensori/attuatori

Passando da un livello all'altro, nella gerarchia del sistema di controllo di un impianto, il tipo di traffico generato nella rete di comunicazione cambia drasticamente.

Nel livello di processo, PLCs, robot, computers etc. sono interessati allo scambio di programmi e quantità di dati di svariate centinaia di Kbytes, e quindi sono richiesti al protocollo appropriati meccanismi di verifica della correttezza nella trasmissione, dato che si tratta di blocchi di informazioni unici che vengono forniti manualmente dall'operatore una sola volta e in modo asincrono.

Nel livello dei sensori/attuatori diventano fondamentali due requisiti in più della rete che i livelli superiori del sistema prendono in considerazione: garantire il rispetto di vincoli temporali nella trasmissione dati ed avere un protocollo che sfrutti in modo efficace la larghezza di banda nonostante la piccola dimensione dei dati scambiati.

Volendo progettare un protocollo che sia in grado di soddisfare le esigenze di tutti i tipi di dispositivi interconnessi si deve tenere conto tanto dei dispositivi di campo più semplici quanto di quelli "intelligenti". Possiamo pertanto considerare due distinte classi di traffico dati che interessano l'area dei sensori/attuatori:

- **Dati di Processo (dati di I/O)**

Questo tipo di informazione è contraddistinto dall'aver un effetto diretto sul processo fisico, come ad esempio il set-point per un sistema guida. La loro complessità è talmente bassa da richiedere solo pochi bit (anche meno di 8) per dispositivo, ma devono essere trasmessi ciclicamente nella rete, ad intervalli fissi, dettati dalle costanti di tempo dei processi in gioco.

Una stima realistica sul periodo con cui questo tipo di traffico deve essere costantemente generato in un impianto automatizzato moderno varia da 1 a 5 msec, e nel futuro tale intervallo tenderà a stringersi verso il limite inferiore. I vari dati sul processo sono in genere distinti, in modo univoco, da un semplice indirizzo rappresentante il dispositivo di appartenenza, e nessun'altra informazione aggiuntiva è necessaria.

- **Parametri (o messaggi)**

I parametri sono usati per osservare, correggere e programmare il comportamento di dispositivi di campo intelligenti. Il traffico generato da questo tipo di informazione è aciclico, poiché avviene solo su richiesta dell'operatore, senza quindi esigenze temporali stringenti, ma è composto da unità dati la cui dimensione può variare dai 10, 100 bytes, per la parametrizzazione di un dispositivo, alle svariate centinaia di Kbytes, per la sua programmazione. Poiché ogni dispositivo intelligente può scambiare una varietà di dati di tipo diverso ogni messaggio dovrà contenere non solo l'indirizzo del dispositivo ma anche una descrizione del contenuto informativo.

Il sistema di comunicazione deve quindi supportare due diversi tipi di servizi, molto diversi tra loro. Da una parte la rete deve fornire servizi di I/O che gestiscano il traffico con esigenze temporali critiche, dall'altra è auspicabile che ci siano anche dei servizi che permettano la programmazione e la configurazione dei dispositivi intelligenti. Una possibile soluzione per conciliare queste due esigenze, contrastanti per natura, sarebbe utilizzare due reti con due protocolli distinti il che costringerebbe inevitabilmente l'utente a dover

equipaggiare i dispositivi intelligenti con due distinte interfacce di rete : una situazione inaccettabile visto l'elevato numero di dispositivi presenti nelle applicazioni industriali ed il loro costo.

L'Interbus-s affronta il problema cercando di trovare un compromesso fra i due tipi di esigenze (come d'altronde cercano di fare tutti i protocolli dei Fieldbus) riuscendoci abbastanza bene, grazie ad un approccio originale.

3.Confronto tra metodi di trasmissione

Una delle caratteristiche più originali di Interbus-s consiste nell'uso del protocollo basato sull'uso di "one-total-frame".

Gli approcci tradizionali usano un protocollo di tipo "**message-oriented**", basato sulla connessione logica punto a punto tra un master centrale e un certo numero di slaves (sensori/attuatori) che, ciclicamente, il master interroga mandandogli, a turno, un messaggio e attendendo una risposta, prima di passare al successivo. L'efficienza di questo protocollo è molto bassa quando il traffico dati è quello ciclico delle unità dati di processo, perché lo scambio di pochi bit di dati richiederà l'uso di due intere frame, ciascuna contenente informazioni di controllo richieste dal protocollo (indirizzi, campi CRC, comandi etc.). Addirittura, una delle due frames avrà il campo dati assolutamente vuoto, se si tratta di un ack.

Le cose migliorano quando il traffico è quello aciclico dei parametri, ovvero dei messaggi asincroni più lunghi e complessi, che sfruttano meglio la struttura della frame.

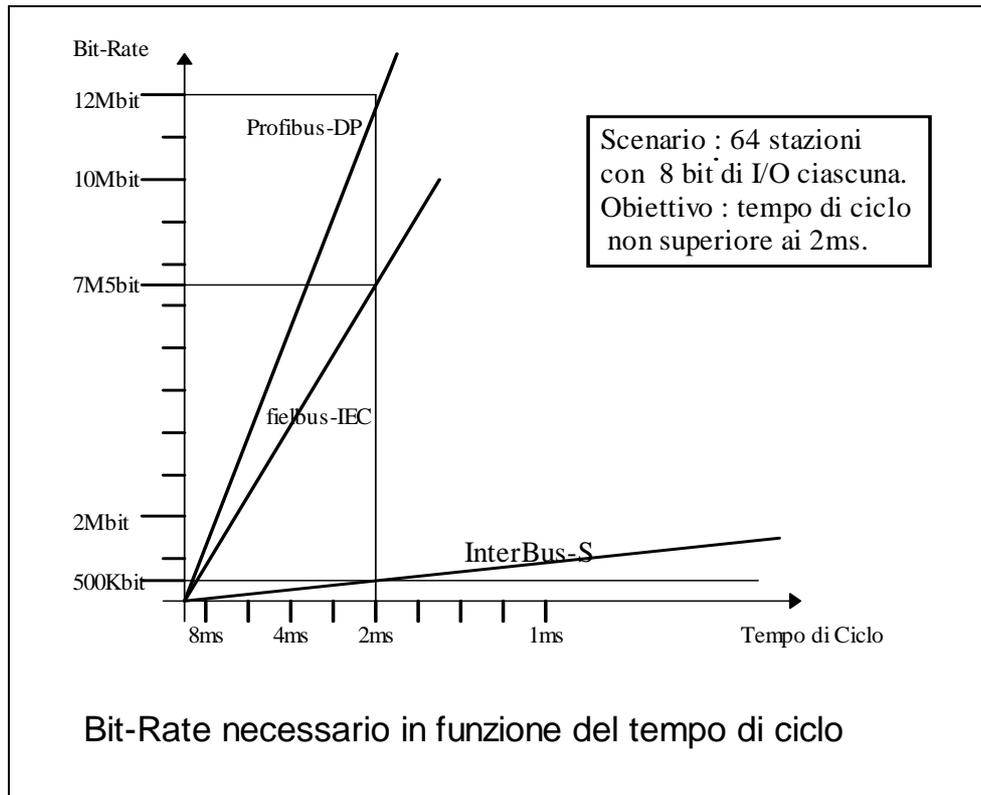
Il protocollo adottato dall'Interbus-s è chiamato "**one-total-frame**" proprio perché combina i dati di tutti i sensori ed attuatori della rete all'interno di un unico messaggio, inviato ciclicamente ; così facendo, il campo di controllo viene aggiunto all'insieme dei dati una volta sola per tutti dispositivi , assicurando sempre un elevato sfruttamento della frame. inoltre, come vedremo, la rigidità della struttura permette di omettere nella frame diverse informazioni, quali l'indirizzo delle stazioni destinazione sostituito dalla loro posizione fisica.

Un uso efficiente della larghezza di banda diventa ancora più significativo quando i dati da scambiare sono parecchi, e le frames non sono più di piccole dimensioni. Il tempo di ciclo può allungarsi troppo, fino a causare pericolosi rallentamenti nel processo di comunicazione, intollerabili quando è in ballo la sicurezza di un impianto.

Consideriamo ad esempio uno scenario in cui esistono 64 dispositivi, ognuno rappresentato da 8 bit di dati di I/O. Se si calcola l'efficienza del protocollo come rapporto tra i soli bit di dati utili e quelli che invece sono realmente trasmessi, otteniamo che il fieldbus PROFIBUS ha un'efficienza pari al 4,6%, il DEVICENET, basato su protocollo CAN, pari all'8% e il fieldbus IEC pari al 7,3% : solo una minima parte della larghezza di banda è usata per informazioni utili !

Il protocollo InterBus-S permette invece di raggiungere una efficienza superiore al 70%. Per ottenere uno stesso tempo di ciclo, a parità di traffico dati, gli altri protocolli dovrebbero avere un bit-rate anche più di 10 volte più grande, ovvero di svariati Mbits (contro i 500Kbps di Interbus-s), con lo svantaggio degli elevati costi realizzativi e dello sfruttamento al limite delle capacità del mezzo trasmissivo (che limiterebbe la possibilità di ulteriori miglioramenti delle prestazioni).

Questi risultati sono sintetizzati nella figura seguente che confronta i bit rate necessari in vari tipi di Fieldbus per ottenere un tempo di ciclo di 2 msec.



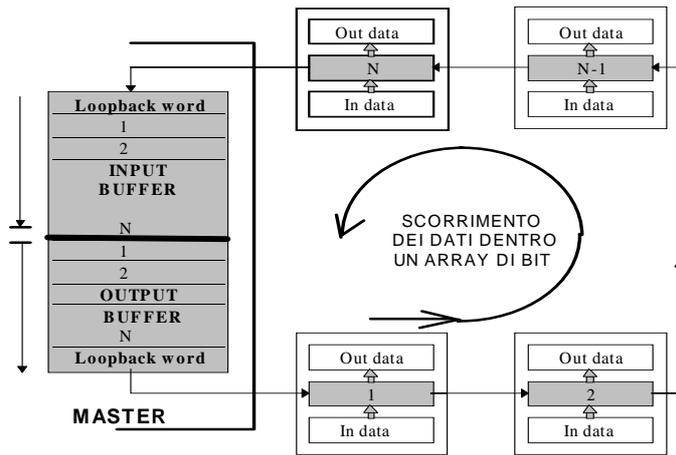
4. InterBus : il bus sensore/attuatore

InterBus-S utilizza una procedura d'accesso **centralizzata di tipo master-slave**, dove il master è anche l'elemento di accoppiamento del sistema con il sovrastante apparato di controllo (un host computer).

La **topologia della struttura è ad anello**: tutti i dispositivi sono nodi attivi di un percorso chiuso, nel quale il protocollo di trasmissione fa scorrere (shifta) la frame con tutti i dati.

Ogni nodo connesso ha un registro a scorrimento di lunghezza finita, legata al numero di unità dati di processo gestite, tipicamente da 1 a 4 ottetti, per cui la loro connessione in serie nell'anello complessivo dà luogo ad un unico lungo array di bit.

Le due estremità di questo array sono connesse rispettivamente all'interfaccia d'ingresso e all'interfaccia di uscita del master, ed ogni ciclo di trasmissione consiste nel fare scorrere fuori da una estremità di questo array fin dentro al master i dati presenti, proprio mentre quest'ultimo va infilando al loro posto altri dati, dall'altra estremità.



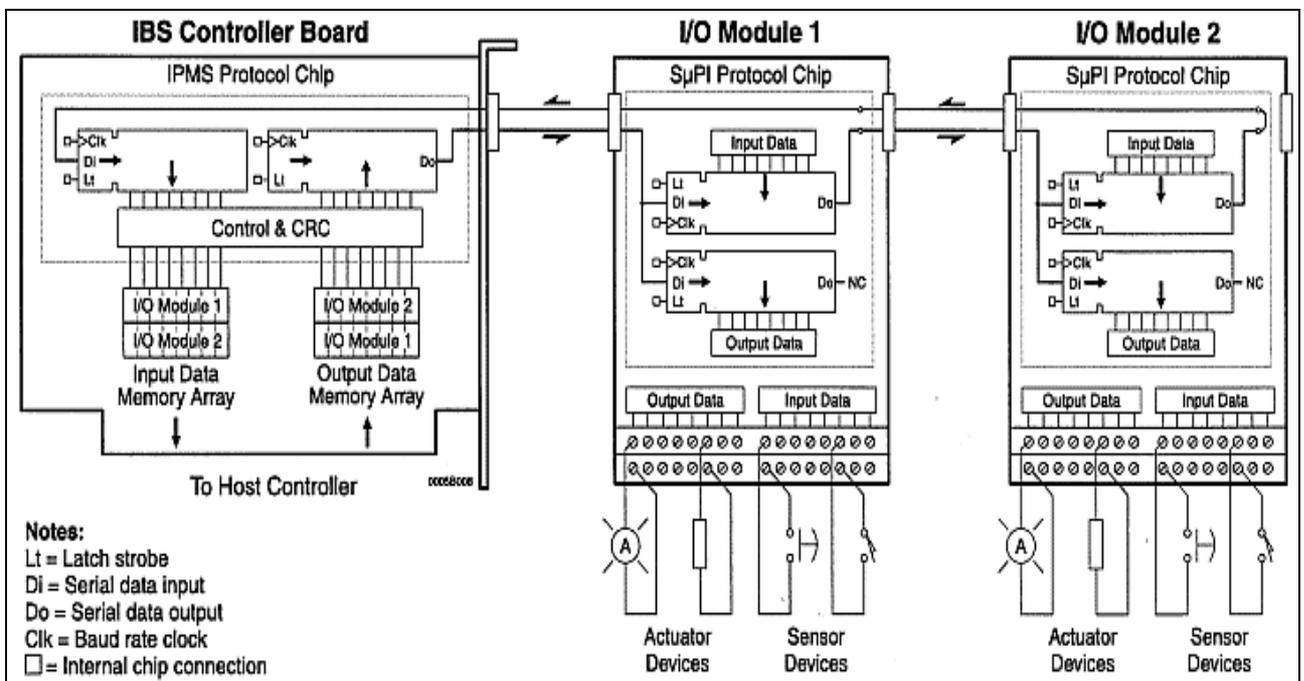
Metodo d'accesso

I dati da trasmettere alle periferiche sono infatti raccolti, nello stesso ordine fisico in cui sono connessi i dispositivi, in un **output-buffer** del master, dal quale vengono prelevati in sequenza per essere trasmessi.

La frame trasmessa dal master inizia con una parola di controllo detta **Loopback word**, ed è composta da tutti i dati di output destinati ai vari nodi connessi all'anello (attuatori, dispositivi programmabili, robot etc...), predisposti nel nell'output buffer.

Al termine dello scorrimento dei dati di tutto l'array di registri di shift (ovvero al termine del ciclo di scansione) ciascun nodo trova nel proprio **out buffer** locale, il dato appropriato inviato dal Master. Continuando il proprio viaggio la frame entrerà nel **buffer di input** del master, e sarà composta da tutti i dati sul processo (l'immagine del processo) inizialmente presenti nell'input buffer locale dei singoli nodi, e sarà conclusa dalla **loopback word**.

Usando questo sistema, affinché il sistema trasmetta 16 bit di output e 16 di input sono necessari solamente 16 impulsi clock di tempo, poiché il protocollo fa in modo che il dato di input occupi nella frame lo spazio lasciato libero dal dato di output. In pratica, la struttura hardware di ogni nodo permette di sostituire i dati ricevuti dal master, con quelli da inviare al master, all'interno dello stesso ciclo di scansione permettendo uno **spatial reuse** del ring che aumenta l'efficienza della trasmissione. In pratica, in un solo ciclo di scansione è possibile contemporaneamente leggere e scrivere dati per lo stesso dispositivo.

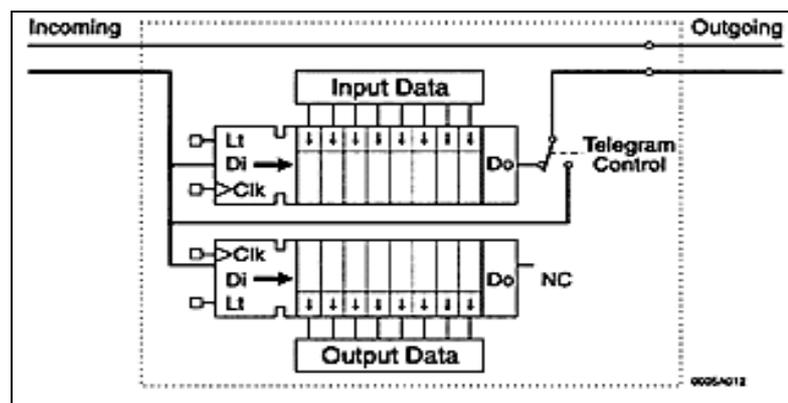


I componenti principali della rete sono sintetizzati in due chip: l'**INTERBUS-S Protocol Master (IPMS)** che si trova nella scheda madre di controllo del sistema, e il chip **Serial Microprocessor Interface (SµPI)** che permette la connessione dei dispositivi di campo al fieldbus.

Il controllore del sistema InterBus-S, cioè il master, elimina la necessità di gestire l'identificazione dei nodi con nomi o indirizzi grazie ad un **ciclo di identificazione (ID Cycle)** che viene eseguito all'atto dell'inizializzazione del sistema, ed in generale ogni volta che si modifica la disposizione dei nodi nell'anello. Il ciclo di identificazione, una volta attivato, dall'utente della stazione di controllo (un host computer) connessa al master, permette di conoscere il tipo e la disposizione dei nodi, ovvero dei registri dati, che compongono il sistema, prima che questo venga posto in funzione. Di fatto è attraverso il ciclo di identificazione che si rende noto al master come organizzare i dati nella frame che deve ciclicamente trasmettere.

Inoltre è fondamentale che tutti i nodi siano sempre perfettamente sincronizzati con un clock comune, poichè lo scorrimento dei dati attraverso i registri, e fin dentro al master, è legato proprio al fatto che le trasmissioni reciproche avvengano simultaneamente. A tale scopo, sia durante il ciclo di inizializzazione che nel funzionamento normale della rete, il chip IPMS del master alterna continuamente la trasmissione, ogni 8 bit di dati shiftati, con una informazione di controllo di 5 bit, detta **messaggio telegram**, inviata in broadcast (contemporaneamente) a tutti gli S_uPI (i nodi).

La trasmissione in broadcast viene realizzata grazie ad una apposita circuiteria (telegram switch) presente negli slaves per permettergli di commutare la loro connessione da seriale a parallela.



Infatti, come si vede in figura, che rappresenta i due shift presenti in ogni nodo, i dati che scorrono lungo l'anello possono entrare nella stazione scorrendo all'interno dei due shift register presenti o possono bypassarla quando il "telegram control switch" è spostato sul conduttore. In questo secondo caso, le informazioni mandate dal Master, giungono contemporaneamente a tutte le stazioni slaves permettendo la sincronizzazione delle attività.

Il messaggio **telegram**, oltre a servire per la sincronizzazione del clock di tutti gli S_uPI contiene informazioni di controllo sul tipo di ciclo attualmente in atto nel bus (Identificazione, scansione Dati o CRC check).

Gli shift register presenti in ogni nodo costituiscono gli elementi essenziali per il funzionamento e meritano qualche approfondimento. Innanzitutto, come si vede si tratta di SR con ingresso seriale/parallelo ed uscita seriale/parallela. L'ingresso/uscita seriale avviene attraverso l'anello. L'ingresso/uscita parallela avviene attraverso i buffer presenti nel nodo e destinati a ricevere dal Master le informazioni (output data) o a spedire al Master informazioni sullo stato del processo (Input data). I due shift ricevono contemporaneamente la stessa informazione attraverso il ring ma, come si vede in figura, lo shift register superiore fa proseguire

l'informazione verso la stazione seguente, mentre quello inferiore ha l'uscita non connessa (in pratica, il flusso di bit che entra nello shift register inferiore, si perde attraverso l'uscita non connessa).

Quando però l'informazione presente negli shift register è esattamente quella destinata ai singoli nodi (ciò avviene contemporaneamente per tutti i nodi) un telegram message dà il comando di trasferire l'informazione dallo shift register inferiore (output data) all'output buffer che memorizza in tal modo il dato destinato alla stazione. Contemporaneamente, il dato prodotto dalla stazione e contenuto nell'input buffer (input data) viene sostituito a quello presente nello shift register superiore e fatto proseguire verso il master.

In tal modo si realizza la lettura del dato destinato ad ogni stazione e la sua sostituzione con quello che la stazione vuole inviare al Master.

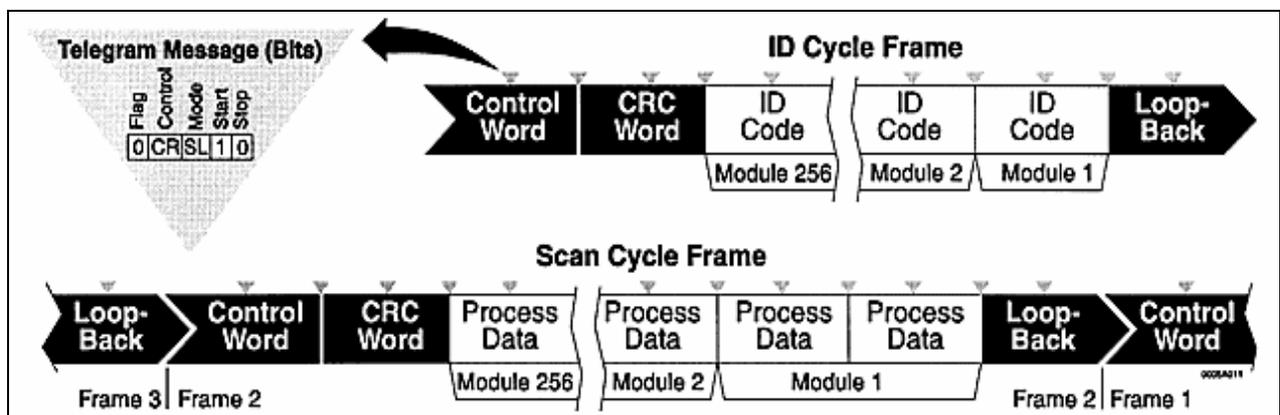
5. Il protocollo di trasmissione InterBus

Il protocollo del sistema InterBus-S è strutturato in una architettura a tre livelli (Physical, Data Link, Application), della quale solo i primi due sono stati standardizzati.

Il MAC del Data Link Layer è un protocollo ciclico basato sul metodo one-total frame: ogni frame contiene i dati di tutti i dispositivi della rete, ma è presente una estensione per il supporto del traffico aciclico dei parametri e programmi : nella frame sono anche presenti degli slot liberi, di 1 byte di capienza, uno per ognuno dei dispositivi che supportano il traffico aciclico.

La frame del **ciclo di identificazione** e quella dei **cicli di scansione dei dati** di processo sono gli unici due tipi di frame esistenti.

La figura che segue mostra i due tipi di frame trasmesse dal protocollo ed i messaggi telegram.



Entrambe le frame vengono trasmesse dal master con la **LoopBack** word di intestazione.

Dopo la LoopBack word seguono i **dati di output** per i dispositivi (ordinati secondo la loro disposizione fisica) e quindi la frame si conclude con 16 bit di **CRC** e 16 bit di **controllo**, usati dagli slaves per segnalare al master eventuali errori rilevati : queste tre sole words costituiscono tutto il sovraccarico aggiunto dal protocollo ai dati utente : da ciò deriva la maggiore efficienza rispetto ad altri bus di campo.

6. Protocollo di trasmissione

Per comprendere il funzionamento del protocollo esaminiamo passo passo come si svolge il ciclo di identificazione (ID Cycle).

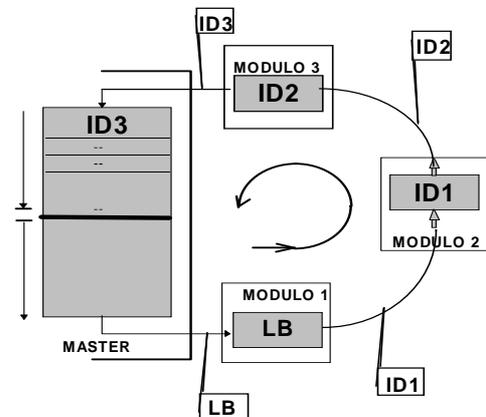
CICLO di IDENTIFICAZIONE

1. il master invia un messaggio telegram col quale mette la rete nello stato di ID cycle ;
2. gli slaves entrano nel modo ID cycle ;
3. il master trasmette la LoopBack word che viaggia attraverso gli shift register dei vari slaves; gli slaves caricano il loro buffer locale con un codice identificativo (ID code) di 16 bit caratterizzante il tipo di dispositivo connesso ed indicante quanti bit deve occupare nella frame il campo del loro dispositivo. Poiché il master non conosce il numero di slaves connessi nel ring, la lunghezza della total frame non è ancora definita.
4. Il master riceve indietro la loopback word e comprende che ha fatto un intero giro. Invia un messaggio telegram che indica agli slaves di caricare nel proprio shift register le informazioni di identificazione della rete. A questo punto la total frame contiene all'interno tutte le informazioni immesse dai vari slaves.
5. La total frame continua a scorrere. Ogni 16 impulsi di clock giunge nel master il codice identificativo di un nodo;

1. La lettura dei codici identificativi continua fino a che il master non riceve la Control Word, che segna la fine del ciclo ;

Dopo che il ciclo di identificazione è completato, il master conosce perfettamente il numero, il tipo e la disposizione dei moduli presenti nella rete.

Quindi ora sa esattamente il numero e la struttura dei campi di dati che devono comporre la frame di trasmissione (informazione disponibile anche al controllore host per una verifica della configurazione di rete).



Ogni 16 bit di trasmissione simultanea un ID code entra nel buffer di input del master

Durante il funzionamento normale, il ciclo di scansione dei dati nel bus è invece riassumibile nei seguenti passi :

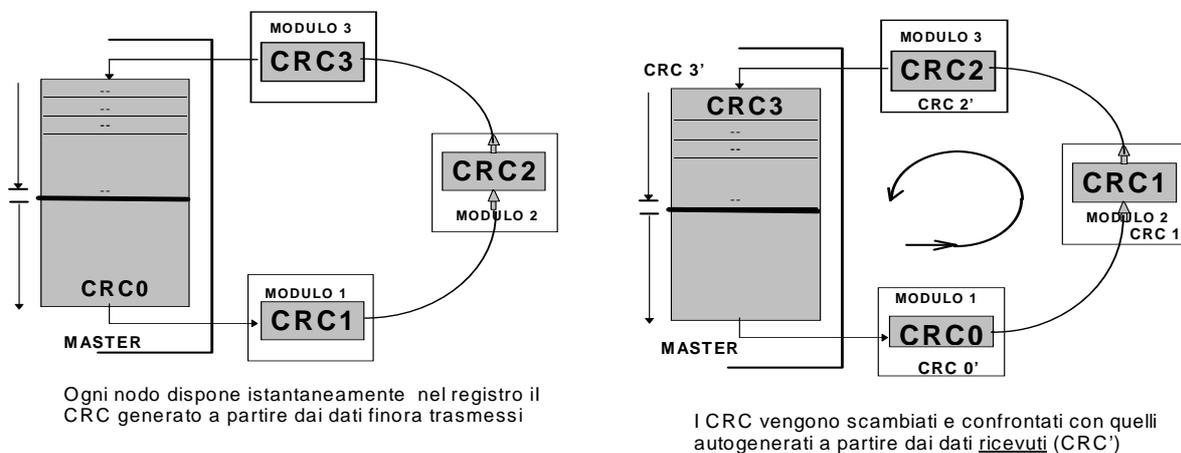
CICLO DI SCANSIONE

1. Il master spedisce un messaggio telegram che mette la rete nello stato operativo di scansione ciclica dei dati ;
2. Tutti gli slaves mettono il dato (1 word, tipicamente) fornito dai sensori nel proprio **input data buffer** locale per inserirlo successivamente nello shift register. Contemporaneamente, il master delinea la dimensione dei dati di output per gli attuatori ed inserisce tali dati nel proprio output buffer.

3. Il master dà il via allo scorrimento della frame nel bus ;
4. La frame scorre completamente attraverso tutti i nodi fino a tornare al master. Il tutto dura un numero di impulsi di clock che il master ha potuto calcolare grazie al ciclo di identificazione. Quando la loopback word è entrata interamente nell'input buffer del Master, i dati si trovano posizionati correttamente nei vari shift register degli slave. E' questo il momento in cui il master può inviare un telegram message che comanda agli slave di registrare i dati ricevuti ed inviare i propri dati.
5. Il controllo della correttezza della frame (campo CRC) consente di verificare che tutti i dati (e quindi anche i propri) sono stati ricevuti correttamente e quindi possono essere utilizzati. Il calcolo del CRC è però particolare nell'Interbus-s poiché ogni slave sostituisce i dati che ha ricevuto dal master con i propri, per cui la frame che giunge al master è diversa da quella che esso ha inviato. Il CRC va quindi ricalcolato continuamente, durante il percorso della one-total frame, e poiché ogni slave modifica un pezzo di tale frame, ogni slave dovrà ricalcolare un nuovo CRC che sostituirà a quello esistente nella frame ricevuta. Ogni volta che una intera frame è transitata attraverso uno slave, questo confronta il campo CRC ricevuto dal suo predecessore con quello autogenerato durante la ricezione della frame. Dopo di che, ricalcola un nuovo valore del CRC che tiene conto delle modifiche che esso ha apportato alla frame stessa e lo sostituisce a quello ricevuto. Dopo aver superato il test del CRC diventano validi e disponibili per l'uso nei dispositivi di campo o nella scheda di controllo del master.

I passi dal 2. al 5. sono ripetuti continuamente finché il sistema IBS non viene arrestato.

Con riferimento al calcolo del CRC va rimarcato come in ogni ciclo ogni nodo, master compreso, ha calcolato due diversi codici CRC di 16 bit, uno per la frame ricevuta ed uno per quella trasmessa, usando il polinomio caratteristico $1+z^{*exp(5)}+z^{*exp(12)}+z^{*exp(16)}$. Il ciclo di trasmissione della frame si conclude proprio con lo scambio dei campi CRC, al fine di convalidare i dati appena ricevuti. La figura sottostante evidenzia il processo di modifica del CRC durante la progressione della one-total frame.



6. Implementazione della rete

Il protocollo seriale InterBus-S è implementato su due tipi di bus : bus **locale** e bus **remoto**.

Entrambi trasportano gli stessi segnali, ma a livelli elettrici differenti.

Il **bus remoto** è usato per il trasferimento dati su lunghe distanze, fino a 400m, ed è l'uscita della scheda del master a costituirne la prima estremità.

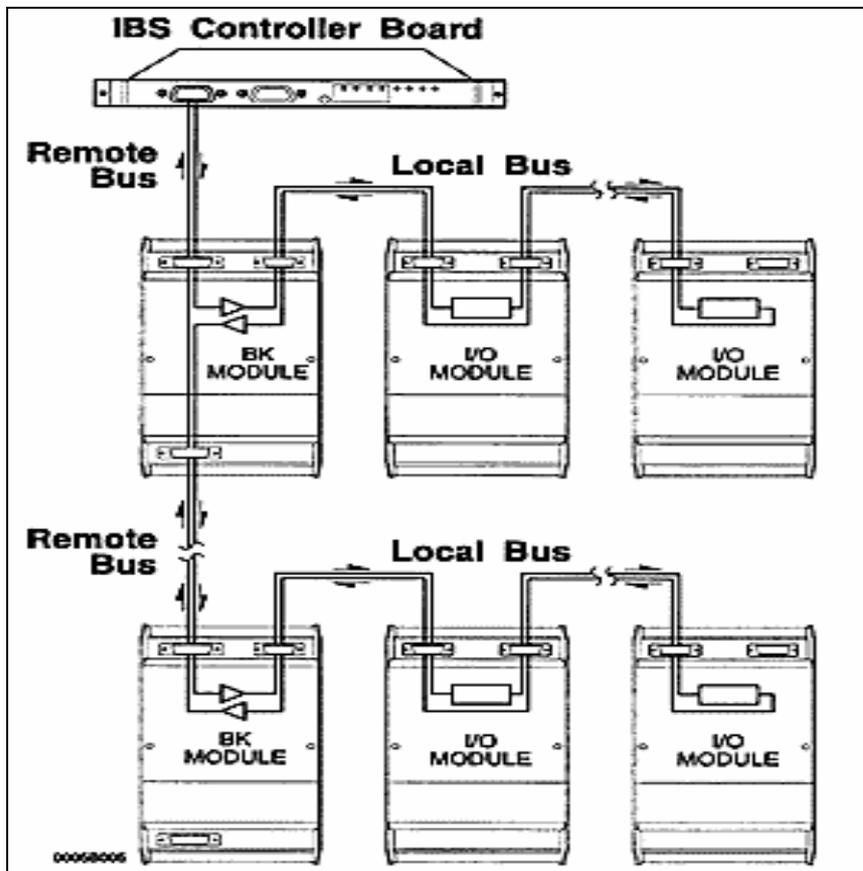
Nei cavi del bus remoto non viene trasportato alcun segnale di alimentazione, ed i livelli elettrici sono quelli dettati dallo standard **RS-485**.

Il bus remoto funziona a **500 k baud**, in **full duplex**.

Il **bus locale** connette i dispositivi di campo (moduli di I/O, etc...) al bus remoto attraverso un **Bus Terminal** (modulo **BK**) che traduce i segnali del bus remoto in quelli del bus locale.

Il modulo BK inoltre fornisce il segnale di alimentazione per il funzionamento dei dispositivi di campo.

Un tratto di bus locale può estendersi per tratti di modesta lunghezza, all'interno di un solo capannone, e può avere caratteristiche indipendenti da quelle del resto del sistema, come lo sfruttare un diverso mezzo fisico di trasmissione.



I livelli elettrici del bus locale, che funziona anch'esso a **500 k baud** in **full duplex**, sono **TTL**.

Nonostante la topologia sia una reale connessione ad anello, poichè sia la linea d'andata che di ritorno del bus sono contenute in un unico cavo che passa per tutti i dispositivi, l'aspetto fisico che si ha è di una struttura lineare o ad albero, quindi estremamente pratica e flessibile.

Per trasmettere i segnali è usato, come scelta di base, il doppino intrecciato :

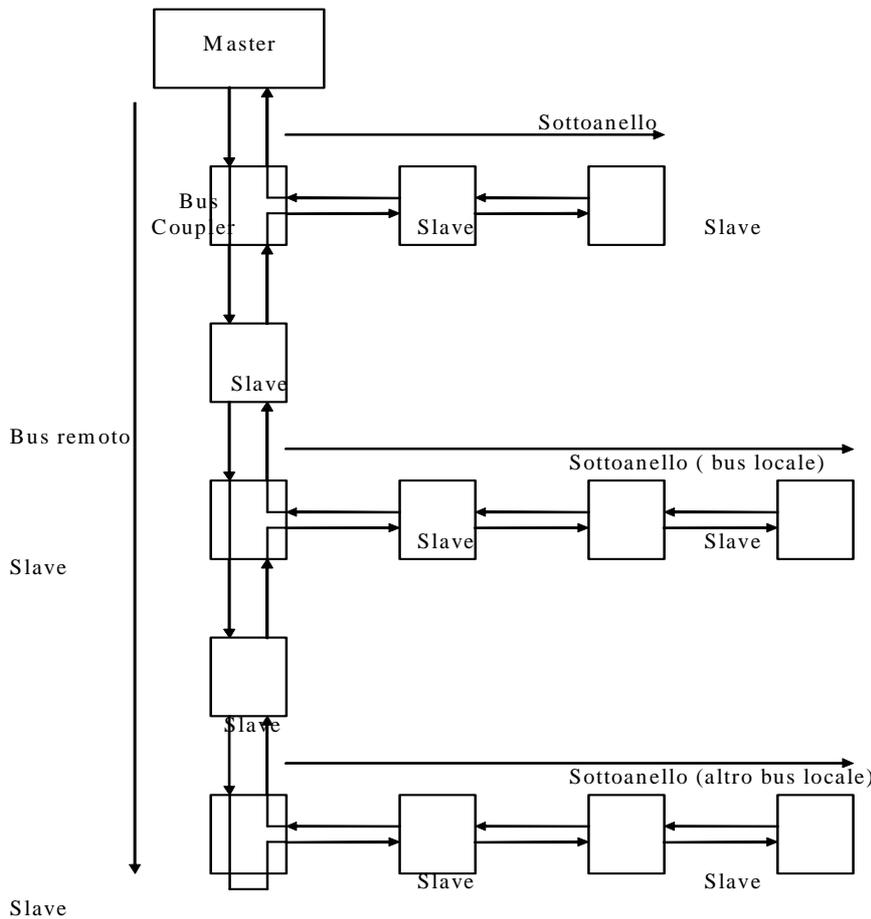
considerando anche il filo di massa sono complessivamente 5 i fili interposti tra i dispositivi.

La funzione di ripetitore, integrata in ogni nodo del bus remoto, permette di estendere il sistema fino a 13Km ; comunque il numero di nodi è limitato ad un max di 512, per semplicità d'uso.

Il bus inoltre può essere equipaggiato in modo da supportare oltre la normale tecnologia a fili di rame anche la fibra ottica, o i raggi infrarossi o qualunque altro mezzo di trasmissione.

La struttura logica ad anello conferisce al sistema due importanti vantaggi rispetto ad un bus lineare : la trasmissione è full-duplex, quindi ogni stazione può contemporaneamente trasmettere e ricevere dati. inoltre,

poichè i vari nodi sono connessi al bus in modo attivo (sono cioè in grado di sconnettersi senza interrompere il bus), la rete può essere segmentata in sottosistemi elettricamente indipendenti. Quindi, il guasto di un nodo o in un punto della linea non causa il blocco dell'intero sistema ma al massimo di un suo segmento, che può essere immediatamente localizzato, anche in modo automatico usando le funzioni diagnostiche della rete (tale funzionalità non sarebbe disponibile se i nodi fossero stati connessi solo passivamente).



In modo del tutto analogo, anche i problemi causati dalle sorgenti di interferenza elettromagnetica o dovuti a connessioni fisiche difettose possono essere localizzati riducendo i tempi morti del sistema: le funzioni diagnostiche dei nodi, attraverso valutazioni statistiche della qualità trasmissiva, permettono di segnalare tempestivamente (e quindi sostituire) i componenti che tendono a guastarsi a causa della normale usura, evitando successivi blocchi dell'impianto produttivo.

La disponibilità di queste funzioni per la gestione dei guasti del sistema rende l'InterBus-S uno dei sistemi più efficienti per le applicazioni di campo.

Un'altra caratteristica interessante, introdotta dall'uso delle interfacce attive intelligenti è che i segmenti di bus locale possono essere connessi/disconnessi all'anello principale (bus remoto) dinamicamente. Ciò permette l'installazione o la rimozione di dispositivi di campo durante il normale funzionamento del sistema, senza alcuna influenza sul resto della rete.

Ancora, grazie al ruolo attivo dei nodi, ogni dispositivo non ha bisogno di indirizzo perché è unicamente individuato dal master attraverso la sua posizione ordinale nell'anello; in tal modo si evitano anche malfunzionamenti legati ad errori dell'operatore nell'indirizzamento dei dispositivi in fase di installazione.

7. Gestione del traffico aciclico

Come sappiamo, il traffico presente nei sistemi di controllo industriale comprende anche una componente asincrona che è normalmente costituita da file di configurazione, software o dati di back up. Il protocollo

InterBus-S supporta il traffico aciclico attraverso quello che, è stato denominato il **Protocollo per le Comunicazioni Periferiche (PCP)**.

I grossi blocchi di dati da trasferire in modo asincrono, vengono disassemblati in frammenti di piccola dimensione (1 ottetto) ed inseriti in sequenza negli appositi slot riservati a questo scopo, insieme al traffico ciclico, per poi essere riassemblati in ricezione, in modo del tutto trasparente per l'utente del sistema.

La durata delle operazioni del protocollo, e quindi i tempi di ciclo per il sistema controllato, restano costanti grazie all'ampiezza fissa delle finestre di trasmissione di tali parametri; così la trasmissione del traffici aciclico non altera la frequenza di aggiornamento ciclico sui dati di processo.

Il canale di comunicazione aciclico fornito da questo protocollo è in genere usato subito dopo la fase di inizializzazione del sistema per programmare ed impostare i parametri di funzionamento di tutti i dispositivi intelligenti, e poi non viene più usato se non sporadicamente (ad esempio per operazioni di back-up).

La comunicazione può avvenire solo tra il master ed uno o più slaves contemporaneamente, in entrambe le direzioni, ma non può avvenire direttamente tra due slaves, a causa dell'architettura centralizzata con cui è strutturato il sistema.

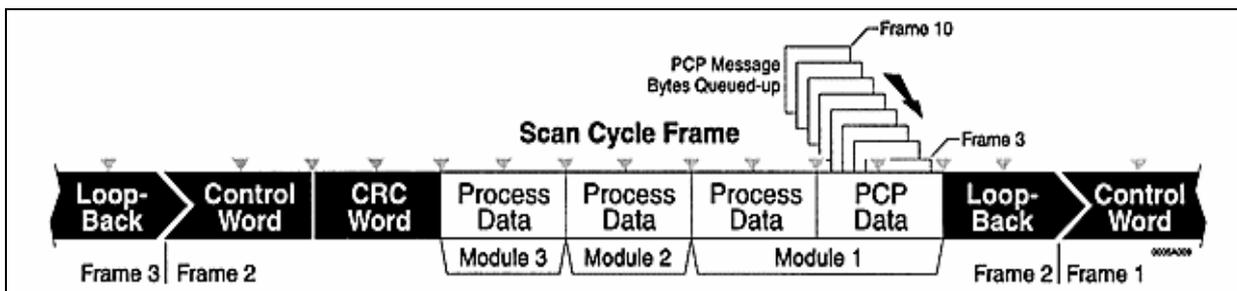
Il protocollo PCP è un protocollo strutturato sul modello client/server, object oriented e connection oriented: dopo aver stabilito una connessione, il client ed il server si scambiano informazioni sul tipo di oggetti disponibili, che possono essere bytes, words, caratteri ASCII, arrays, etc...

Il trasferimento funziona sfruttando lo spazio nella frame dedicato ad uso esclusivo del protocollo PCP, che quindi è disponibile per la comunicazione tra client e server in ogni ciclo di scansione del bus. Ogni dispositivo, nel comunicare al master le dimensioni del suo spazio dati nella frame durante il ciclo di identificazione, specifica anche quanta parte intende riservarne per i dati di processo e quanta è invece allocata per il traffico aciclico.

La trasmissione dei messaggi presenti nella memoria del master avviene quindi spezzettandoli in segmenti inviati durante ogni ciclo di scansione.

Quando il nodo ricevente rileva la fine del messaggio lo riassembla e lo rende disponibile per l'uso.

La figura che segue mostra una frame con un messaggio PCP in coda per la trasmissione.



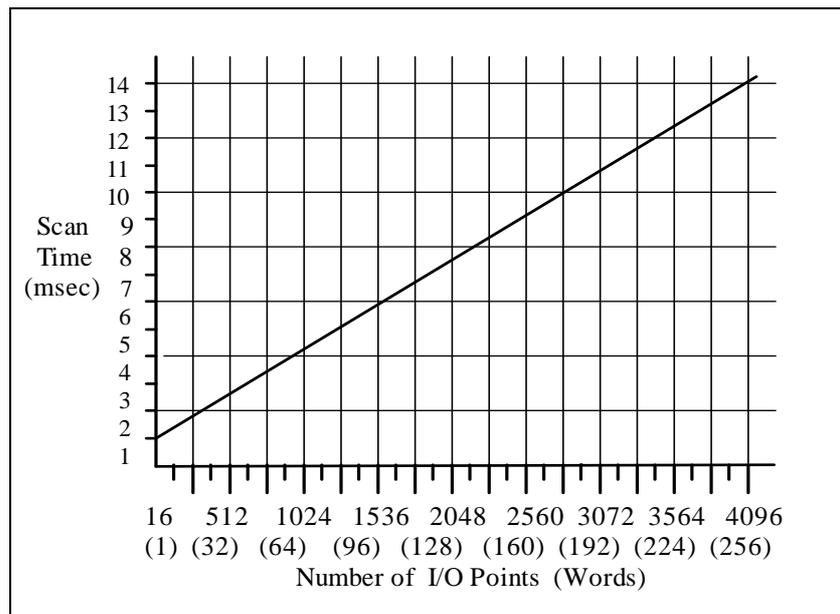
8. Temporizzazioni

I tempi di scansione del sistema InterBus-S sono deterministici e possono essere calcolati al microsecondo da una equazione che tenga conto di tutte le variabili in gioco nella rete.

Il fattore preponderante è il numero di data words presenti nella frame. Grazie al protocollo one-total frame è possibile avere dei tempi di ciclo di pochi msec con bit rate di soli 500Kbit (aumentabile fino a 2.5Mbit): una frame di 256 byte di dati produce un tempo di ciclo inferiore ai 4 msec.

Attualmente la lunghezza della frame è limitata 512 bytes, restrizione legata più alle capacità di memoria del master che al protocollo.

La figura che segue mostra approssimativamente l'andamento della durata del ciclo di scansione in base alla lunghezza della frame (in data words). Si noti come un nodo con 16 bit di output e 16 bit di input data occupa solo 16 bit (e non 32) nella frame, pur avendo la trasmissione in full duplex.



I tempi di ciclo del bus sono ricavabili, trascurando i ritardi nelle linee di trasmissione, attraverso la formula seguente :

$$T_c = (13 \times (6 + n) + 2 \times m) \times T_{bit}$$

dove n = quantità complessiva, in ottetti, di dati utente negli slaves

m = numero di slaves

T_{bit} = bit time

Per comprendere la formula bisogna tener conto che oltre agli n ottetti degli slaves, vanno considerati 6 ottetti dovuti alla loopback word, alla control word e al CRC; inoltre la trasmissione di un ottetto richiede 13 bit (8 per l'ottetto e 5 per la telegram word spedita dopo ogni ottetto). Due bit sono inoltre utilizzati da ogni slave per la sua sincronizzazione.

Il tempo necessario per trasferire invece un messaggio asincrono col protocollo PCP può essere determinato con i seguenti passi :

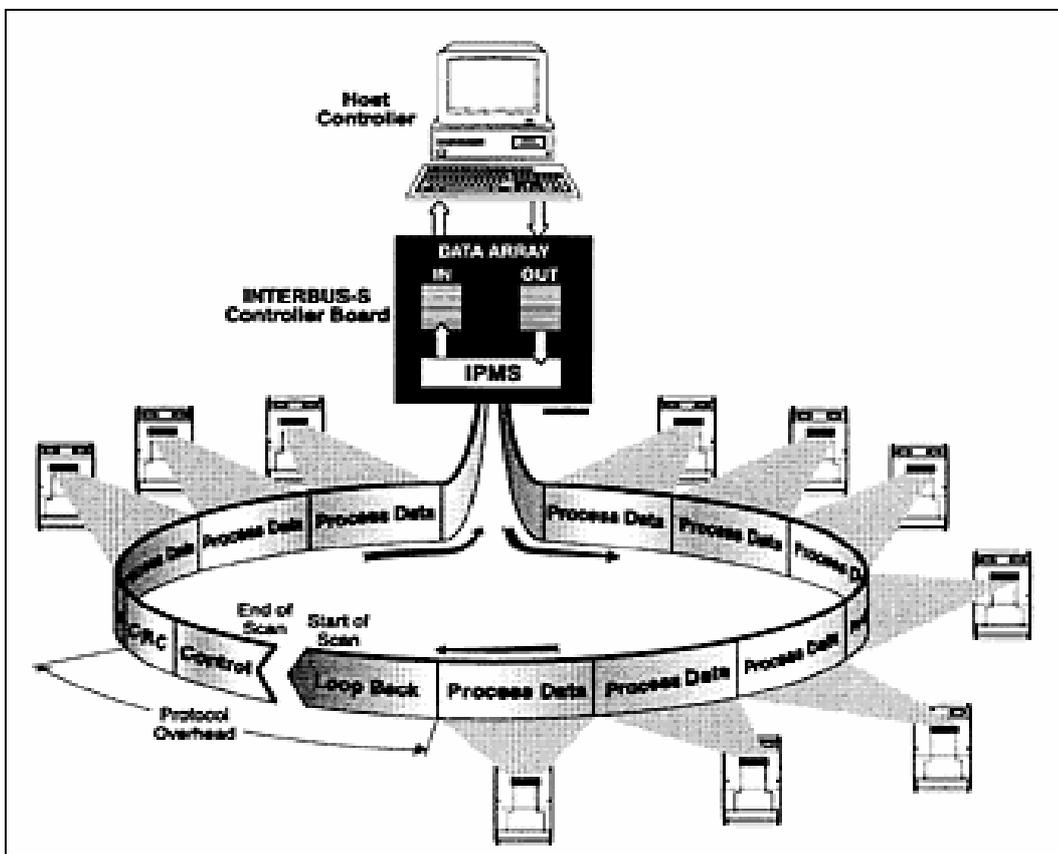
1. si determina la durata di un ciclo di scansione; questo tempo è anche il tempo necessario a trasferire un singolo segmento di un messaggio PCP . L'ampiezza di tale segmento è pari allo spazio m previsto nella frame;

2. In base al tipo di funzione richiesta al server (lettura, scrittura, etc...), un certo numero di informazioni di controllo vengono aggiunte dal protocollo, (tipicamente 12 byte di dati). Questi vanno aggiunti al numero di bytes del messaggio da trasferire determinando così la lunghezza N complessiva del messaggio (in ottetti);

3. il tempo richiesto per il trasferimento si ottiene moltiplicando il risultato della divisione intera $[(N-1) / m] + 1$ per il tempo di durata del ciclo di scansione ; E' importante notare che si possono scambiare messaggi PCP tra più nodi distinti contemporaneamente, senza quindi dover accumulare i tempi dei singoli trasferimenti.

La figura che segue è un esempio concettuale del flusso dei dati in un piccolo sistema IBS.

I dati scorrono attraverso i nodi, per arrestarsi solo quando arrivano al modulo a cui erano destinati, dopo di chè il tutto ricomincia, in un ciclo continuo.



■