

UNIVERSITÀ DEGLI STUDI DI PADOVA  
DIPARTIMENTO DI INGEGNERIA  
DELL'INFORMAZIONE

Tesi di Laurea Specialistica in  
INGEGNERIA DELL'AUTOMAZIONE

**Misure di prestazioni  
della rete Profinet IO**

Relatore  
Prof. Stefano Vitturi

Candidato  
Luca Bellinato

Anno Accademico 2009/2010



# Indice

<b>Introduzione</b>	<b>1</b>
<b>Presentazione Weightpack</b>	<b>5</b>
<b>1 Ethernet ed i bus di campo</b>	<b>7</b>
1.1 I Bus di campo . . . . .	7
1.2 Ethernet . . . . .	10
1.2.1 CSMA/CD . . . . .	11
1.2.2 Verso il determinismo . . . . .	12
1.3 Industrial Ethernet . . . . .	19
1.3.1 Parametri Real-Time . . . . .	20
1.3.2 Performance indicators . . . . .	22
1.3.3 Protocolli Industrial Ethernet . . . . .	23
<b>2 Profinet IO</b>	<b>25</b>
2.1 Introduzione . . . . .	25
2.2 Profinet CBA . . . . .	26
2.3 Profinet IO . . . . .	26
2.4 Classi Real-Time . . . . .	27
2.5 Periodo di trasmissione . . . . .	29
2.6 Le classi dei device . . . . .	31
2.7 Il controllore di rete . . . . .	31

2.8	Modello dei dispositivi . . . . .	33
2.9	Stack di comunicazione . . . . .	34
2.10	Il modello Produttore/Consumatore . . . . .	34
2.11	Scambio di dati ciclico . . . . .	35
2.12	I canali di comunicazione . . . . .	36
2.12.1	Application relationship ASE . . . . .	38
2.12.2	IO Data ASE . . . . .	38
2.12.3	Allarmi Ase . . . . .	39
2.13	Ciclo RT . . . . .	41
2.13.1	Struttura del pacchetto RT . . . . .	41
2.13.2	Ciclo di comunicazione RT . . . . .	43
2.13.3	Stabilità del ciclo . . . . .	45
2.14	Ciclo IRT . . . . .	46
2.14.1	Sincronizzazione e Bypass Clock . . . . .	46
2.14.2	Struttura del pacchetto Profinet IRT . . . . .	48
2.14.3	Canale di comunicazione . . . . .	49
2.14.4	Miglioramento in bus lineari . . . . .	50
<b>3</b>	<b>Strumentazione</b>	<b>53</b>
3.1	IO-Controller . . . . .	53
3.2	Switch . . . . .	55
3.3	IO-Device 1 . . . . .	56
3.4	IO-Device 2 . . . . .	58
3.4.1	Interfaccia di comunicazione seriale (SCI) . . . . .	60
3.4.2	Canale seriale sincrono (SSC) . . . . .	61
3.4.3	Evaluation Board . . . . .	61
3.5	Oscilloscopio . . . . .	63
3.6	Software di progettazione . . . . .	63
3.6.1	Configurazione hardware . . . . .	65
<b>4</b>	<b>Misurazioni</b>	<b>67</b>
4.1	Misura 1 . . . . .	67
4.2	Misura 2 . . . . .	69
4.3	Misura 3 . . . . .	72
4.4	Misura 4 . . . . .	75
4.5	Misura 5 . . . . .	78
4.6	Misura 6 . . . . .	80



---

<b>Conclusioni</b>	<b>83</b>
<b>Bibliografia</b>	<b>85</b>



# Elenco delle figure

1	Vista di una riempitrice prodotta da WeightPack . . . . .	6
1.1	Switch senza e con gestione delle priorità . . . . .	17
1.2	Specifiche dei jobs . . . . .	21
1.3	Protocolli per ethernet industriali . . . . .	24
2.1	Caratteristiche temporali della comunicazione Profinet . . . . .	28
2.2	Periodi di trasmissione . . . . .	29
2.3	Modello IO-Device . . . . .	33
2.4	Stack di comunicazione RT e IRT . . . . .	34
2.5	Scambio di dati ciclico . . . . .	36
2.6	I canali di comunicazione . . . . .	37
2.7	Esempio di CR dell'IO Application Realtionship . . . . .	38
2.8	Servizi dell'IO Data ASE . . . . .	40
2.9	Struttura del frame Profinet RT . . . . .	41
2.10	Ciclo di comunicazione RT . . . . .	44
2.11	Esempio di sovraccarico . . . . .	45
2.12	Esempio di un Control Loop progettato con un PI . . . . .	47
2.13	Boundary Clock e Bypass Clock . . . . .	48
2.14	Struttura del frame Profinet IRT . . . . .	48
2.15	Ciclo di comunicazione IRT . . . . .	49
2.16	Downstream communication . . . . .	50
2.17	Upstream communication . . . . .	51
3.1	CPU 315-2 PN/DP . . . . .	54
3.2	Scalance X204IRT . . . . .	56
3.3	ET 200 M e SM 374 . . . . .	57

---

3.4	Chip Anybus-IC . . . . .	59
3.5	IO Mapping del chip Anybus-IC . . . . .	60
3.6	evaluation board . . . . .	62
3.7	oscilloscopio HP 54504A . . . . .	63
4.1	Topologia della misura 1 . . . . .	68
4.2	Topologia della misura 2 . . . . .	70
4.3	misura 2, 256 ms, unità di misura [ms] . . . . .	71
4.4	misura 2, 64 ms, unità di misura [ms] . . . . .	71
4.5	misura 2, 16 ms, unità di misura [ms] . . . . .	71
4.6	Topologia della misura 3 . . . . .	72
4.7	misura 3, 256 ms, unità di misura [ms] . . . . .	73
4.8	misura 3, 64 ms, unità di misura [ms] . . . . .	73
4.9	misura 3, 16 ms, unità di misura [ms] . . . . .	73
4.10	Topologia della misura 4 . . . . .	75
4.11	misura 4, 256 ms, unità di misura [ms] . . . . .	76
4.12	misura 4, 64 ms, unità di misura [ms] . . . . .	76
4.13	misura 4, 16 ms, unità di misura [ms] . . . . .	77
4.14	misura 4, 4 ms, unità di misura [ms] . . . . .	77
4.15	Topologia della misura 5 . . . . .	78
4.16	ritardi, unità di misura [ms] . . . . .	79
4.17	Topologia della misura 6 . . . . .	80
4.18	OB1, unità di misura [ $\mu s$ ] . . . . .	81
4.19	OB 35 a 16 ms, unità di misura [ms] . . . . .	81
4.20	OB 35 a 4 ms, unità di misura [ms] . . . . .	82

# Introduzione

Le soluzioni di automazione industriale hanno subito in tempi recenti consistenti evoluzioni. La crescita esponenziale di possibilità offerte da forme diverse di comunicazione dati e di elaborazione degli stessi ha investito anche il settore dell'automazione industriale, da sempre alla ricerca di soluzioni migliorative della produttività degli impianti o dei singoli macchinari. Dove per produttività s'intende non solo l'aspetto puramente quantitativo e, quindi, il numero di pezzi prodotti, ma, in senso più generale, la capacità o l'attitudine a produrre di una persona o di un'azienda. Oggi è sempre più normale immaginare una soluzione globale di automazione che, oltre alle tradizionali funzioni di automatizzazione di processo, sia anche in grado di elaborare algoritmi matematici, di creare e gestire in modo semplice ed affidabile archivi di dati di configurazione, di produzione e di diagnostica, di trasferire tali dati ad unità centrali da remoto. I vantaggi di un tale tipo di approccio sono molteplici e possono essere sintetizzati in una riduzione dei costi generali associata alle varie fasi di vita di un impianto: dalla progettazione, all'installazione, al monitoraggio, alla manutenzione, per coinvolgere altri importanti aspetti quali, ad esempio, la flessibilità, la rapidità e la semplicità di modifiche o di riconversione degli impianti. A fronte di tali necessità, le prime soluzioni immaginate, hanno tratto le proprie basi dal mondo dell'Information Technology. Protocolli e tecnologie diffusi e collaudati in questo ambiente quali, ad esempio, HTTP, FTP, SNMP e SMTP, associati alla presenza di un Web Server o alla trasmissione dati via network telefonico, consentono, infatti, una brillante risposta a questo tipo di esigenze. Tutto ciò avvicinava in modo sensibile il mondo dell'automazione industriale a quella che genericamente possiamo definire tecnologia a base Ethernet. La quadratura del cerchio poteva essere considerata l'utilizzo in ambito industriale di

un protocollo a base Ethernet, ma che, laddove necessario, fosse in grado di assicurare determinismo (Real Time o addirittura Isochronous Real Time). Questo approccio consente, infatti, lo sfruttamento dei vantaggi offerti sia da una classica rete Ethernet (elevate quantità di dati trasmissibili, utilizzo di standard di comunicazione già affermati nel mondo office, ecc.) che dai classici bus di campo (determinismo, flessibilità di struttura, diagnostica, cablaggio seriale, ecc.). Usando termini ormai comunemente entrati a far parte del linguaggio tecnico dell'automazione industriale, si trattava di strutturare una soluzione di automazione basata su Ethernet Industriale. L'introduzione di una nuova tecnologia di automazione aveva riaperto le speranze da parte dei tecnici interessati di poter disporre di una tecnologia 'universale', un po' come si era sperato nella fase d'introduzione dei bus di campo. Benché il protocollo di riferimento, Ethernet, possa essere considerato come uno dei pochi protocolli 'universali', in realtà, la necessità di adattamento dello stesso alle esigenze dell'automazione industriale poteva essere soddisfatta mettendo a punto svariate soluzioni tecnologiche e questo, al pari di quanto già successo proprio per i bus di campo, ha portato ad una proliferazione di protocolli tali da rendere ormai utopica la diffusione di un unico protocollo universale per Ethernet Industriale. Tra le varie soluzioni che si stanno imponendo è emersa la tecnologia ProfiNet, sviluppata da PROFIBUS & PROFINET International (PI) principalmente per far interagire segmenti separati di bus di campo attraverso Ethernet e collegarli ai sistemi di gestione dell'azienda secondo una struttura rigorosamente gerarchica. Una delle caratteristiche di Profinet che risulta fondamentale per il raggiungimento dell'integrazione verticale tra livello di campo ed elementi ERP/MES<sup>1</sup> superiori, è la possibilità di utilizzare un unico supporto fisico per la trasmissione di diverse classi di segnale. Mediante l'uso di tecnologia Profinet è, infatti, possibile utilizzare contemporaneamente comunicazione Real-Time e comunicazione basata su TCP/IP su un unico cavo, comunicazione Real-Time scalabile per i controllori, periferia decentrata e Motion Control. Un altro aspetto di fondamentale importanza per il successo della tecnologia Profinet sul mercato è la salvaguardia di precedenti investimenti. In questo caso ci si riferisce alla possibile integrazione dei bus di campo tradizionali (Interbus e Profibus) magari già presenti in campo, con una struttura Profinet IO. Questa possibilità si rivela estremamente efficace per ampliamenti o modifiche di impianti esistenti e gestiti mediante bus di campo, laddove si voglia con queste modifiche sfruttare i vantaggi di una tecnologia a base Ethernet Industriale senza agire sul pregresso, ma anche per la realizzazione di sistemi globali di automazione industriale a tecnologia mista, in cui si vadano a sfruttare al meglio ed in

---

<sup>1</sup>livelli più alti della piramide CIM (Computer Integrated Manufacturing).

modo sinergico i vantaggi di ogni singola tecnologia. Dopo una panoramica sul mondo e sulle caratteristiche dell'ethernet industriale si cercherà di introdurre la soluzione progettata da Siemens (Profinet) specialmente per quanto riguarda il profilo della comunicazione di cella e di campo (Profinet IO). Quest'ultimo tipo di Real-Time ethernet è stata testata nel laboratorio di navigazione autonoma del dipartimento di ingegneria dell'informazione dell'università di Padova; tutto l'occorrente per creare la rete da testare è stato fornito da **WeightPack S.p.A.**, azienda partner di questo progetto, specializzata nella progettazione di macchine per l'imbottigliamento e detentrica di numerosi brevetti. Maggiore attenzione è stata data al chip Anybus-IC di HMS fornito anch'esso da Weightpack ; in particolare l'azienda, che ha in programma un piano di migrazione da Profibus verso Profinet, vuole verificare le effettive prestazioni di tale componente confrontandole con i dati forniti dal costruttore per poi stabilire se tale prodotto sia adatto o meno alle loro esigenze.





# Presentazione WeightPack

WeightPack S.p.A. è stata fondata da Carlo Corniani, conosciuto in tutto il mondo come l'inventore della prima riempitrice a peso netto a controllo completamente elettronico nel 1979. Weightpack è un'azienda italiana che fornisce un'ampia gamma di macchine e servizi per linee da imballaggio a valore aggiunto, con la leadership tecnica in settori come lattiero-caseari, alimenti, bevande, detergenti liquidi, prodotti chimici, per la cura personale, olio da motore ed altre industrie da imballaggio nel mondo. Avendo a che fare con il riempimento di una grande varietà di liquidi non gasati, l'azienda ha messo a punto un gran numero di macchine imbottigliatrici, tutte basate sulla tecnologia di riempimento a peso netto. Inoltre, soprattutto per quanto riguarda i prodotti alimentari, viene data molta importanza all'ambiente stesso di riempimento; si può passare da un ambiente di riempimento classico per prodotti liquidi non sensibili che sono distribuiti a temperatura ambiente fino a prodotti che richiedono un ambiente ad una determinata temperatura, il più possibile sterile usando tecniche di filtraggio del tutto innovative. Dovendo competere a livello internazionale, i loro prodotti sono soggetti a continue evoluzioni tecnologiche per garantire elevate prestazioni nel tempo e soddisfare appieno le richieste del cliente finale. Per questo motivo WeightPack si avvale, oltre che a team di ingegneri interni, di una stretta collaborazione con centri di ricerca universitari, mettendo a disposizione degli studenti risorse e competenze maturate in anni di attività. WeightPack coniuga elevata professionalità e competenze con metodologie di lavoro che prevedono la massima cura dei particolari di progettazione e costruttivi. L'obiettivo è quello di riempire un contenitore nel minor tempo possibile prestando attenzione a parametri rigorosi come può essere la quantità di prodotto: su questo si sono focalizzate le ricerche del team di ingegneri WeightPack. Tra queste

persone indispensabile citare l'ing. Paolo Marogna, che è stato il riferimento di Weightpack per questo lavoro di tesi: ci ha fornito tutto il materiale necessario e ci ha indirizzato sul lavoro da svolgere. La particolarità che distingue il funzionamento delle riempitrici WeightPack da quelle di altri produttori, consiste nella determinazione del livello di riempimento dei contenitori attraverso il monitoraggio del peso. Ogni stazione di riempimento è costituita da una piastra nella quale è installato un sensore di peso. Il contenitore si posiziona sulla piastra e da questo momento ha inizio la fase di riempimento vera e propria. Per prima cosa viene rilevato il peso del contenitore (tara); se il valore rientra nel range stabilito si apre la valvola attraverso la quale scorre il prodotto da imbottigliare. Tutta questa fase viene monitorata campionando il peso fino a che non si raggiunge il valore desiderato.



**Figura 1:** *Vista di una riempitrice prodotta da WeightPack*

Alla chiusura della valvola si procede con un'ulteriore lettura del peso per stabilire con certezza la quantità di prodotto inserito. Successivamente si tappa il contenitore sempre attraverso operazioni meccanizzate. Alla fine del processo avviene la selezione per lo stoccaggio solo se conforme ai parametri della ricetta.

Ogni macchina è progettata per soddisfare le esigenze del cliente e quindi ogni macchina è soggetta a modifiche di progettazione più o meno sostanziali. Molte di queste modifiche sono diventate brevetti industriali WeightPack.

# Capitolo **1**

## Ethernet ed i bus di campo

La comunicazione in ambienti industriali è un argomento particolarmente complesso, inoltre l'attività di standardizzazione in questo settore ha sempre incontrato difficoltà derivanti principalmente dai forti interessi dei produttori. In questo contesto si colloca il protocollo Profinet IO che è il fulcro di questo lavoro di tesi.

### 1.1 I Bus di campo

Nel panorama dei sistemi di controllo il concetto della centralizzazione delle funzioni di controllo è rimasto dominante fino a tutti gli anni '80. In questo modello, tutte le funzioni di controllo sono svolte da un'unità di governo centralizzata, alla quale si collegano i diversi componenti del sistema d'automazione (trasduttori, attuatori, elementi di regolazione), dislocati nelle diverse aree geografiche dell'impianto. Il collegamento tra ciascuno degli elementi periferici e l'unità centrale avviene per mezzo di cablaggi dedicati, di norma con tecnologia analogica (linee di comunicazione punto-punto a 4/20 mA). I limiti di questa soluzione in termini di costi dei cablaggi (e di relativi problemi di disturbi elettromagnetici e schermatura), convivenza nel quadro di controllo di elementi fortemente disomogenei tra loro, inefficienza complessiva, hanno spinto la ricerca verso soluzioni di controllo alternativo. Oggi l'automazione industriale e di processo è ormai operativa verso il controllo distribuito, che impone l'utilizzo di dispositivi dotati di intelligenza integrata, atta a svolgere funzioni di automazione autonoma ausiliarie e su-

bordinate alle funzioni principali del ciclo di lavoro, collegate all'elettronica di controllo macchina mediante sistemi di comunicazione a bus. L'evoluzione dell'uso di questi sistemi distribuiti in un impianto, ha dimostrato da subito i suoi vantaggi, tra cui: la distribuzione intrinseca di alcune applicazioni che richiedono lavorazioni su macchine distanti e separate, affidabilità, scalabilità, modularità, ma anche i difetti reputabili principalmente a una maggiore complessità del software di gestione rispetto a sistemi centralizzati e la necessità di un efficiente sistema di comunicazione in tempo reale. Tutto ciò ha trovato grosso giovamento dai progressi tecnologici nel campo delle comunicazioni digitali (inclusi i protocolli usati in internet) e nel campo dello sviluppo dei microprocessori.

Con il termine *bus di campo* si intende quindi una rete di comunicazione dedicata ai sistemi a basso livello, quali sensori o attuatori, la quale ha sostituito nel campo industriale la tecnologia analogica per collegamenti punto-punto basata su segnali a 4-20 mA. La modalità di trasmissione numerica presenta, infatti, notevoli vantaggi in termini di possibilità di trasmettere più informazioni e migliori potenzialità di interfacciamento. L'architettura a bus, nella quale un unico conduttore viene utilizzato per connettere più sistemi, consente una notevole riduzione dei cablaggi. A livello di comunicazione, rispetto a connessioni tra soli due nodi, il modo di scambiare dati attraverso un bus di campo richiede regole più severe. Bisogna, ad esempio, prevedere l'accesso contemporaneo al mezzo da parte di elementi diversi e regolamentare, nel modo più rigoroso possibile, tutto ciò che serve per poter trasmettere i dati in maniera rapida e affidabile. Tali regole di convivenza vengono comunemente chiamate protocollo. I moderni sistemi di automazione industriale, in seguito all'introduzione di numerosi sensori e attuatori intelligenti, hanno la capacità di trasmettere oltre ai dati anche una serie di informazioni atte al controllo e alla supervisione della rete stessa. Queste informazioni devono essere disponibili in tempi brevissimi e in grado di viaggiare in entrambe le direzioni. I primi bus di campo, utilizzati nei sistemi distribuiti, si basavano sul livello fisico RS485 in una rete di tipo master-slave, dov'era consuetudine l'uso di un protocollo proprietario, comportando un notevole lavoro nell'implementazione del sistema a più alto livello e anche nell'interfacciamento. La gestione di attività con vincoli temporali (anche in tempo reale) è risultata ancora più complessa rispetto ai sistemi centralizzati, in quanto al problema dello scheduling si aggiunge quello della comunicazione e della sincronizzazione tra processi residenti su nodi diversi. In alcuni casi si è pervenuti ad architetture multimaster, consentendo un accesso al bus di tipo deterministico, attraverso una gestione prestabilita del token. Con questa soluzione si riesce a calcolare un tempo massimo entro il quale l'informazione di qualsiasi utente verrà trasmessa, o il sistema reagirà entro un intervallo di tempo prevedibile.

Questo tipo di reti, dovendo agire a livello di dispositivo e cella, devono avere prestazioni ben descritte, soprattutto a livello di:

1. scambio dati ciclico;
2. trattamento traffico urgente aciclico;
3. trattamento messaggistica alto livello.

Le prime due funzioni interessano soprattutto a livello di dispositivo, devono essere implicitamente garantite dalla rete stessa e possono essere implementate come:

- Produttore/Consumatore: in questa implementazione è previsto un nodo arbitro che conosce quali variabili sono necessarie, chi le produce e chi le consuma, chiedendo ciclicamente la produzione di una variabile da rendere disponibile sulla rete.
- Master/Slave: la comunicazione è gestita da un protocollo interno, la gestione di questo è affidata ad un Master.

La seconda funzione che interessa a livello di dispositivo è la gestione del traffico urgente asincrono, tipicamente gli allarmi. Di conseguenza si deve garantire un tempo massimo di trasmissione (il più breve possibile) per file di piccole dimensioni. Infine la messaggistica di alto livello deve essere garantita a livello di cella e deve consentire la trasmissione sia verso i dispositivi sia verso l'impianto di file di grandi dimensioni, FTP, variabili o il caricamento-scaricamento di grandi aree di memoria.

I sistemi di comunicazione a bus di campo sono, oggi, una realtà, vista la disponibilità sul mercato di un'ampia gamma di prodotti commerciali. Alcuni di essi sono relativi a standard proprietari, mentre altri fanno riferimento a standard definiti a livello Europeo. In ogni caso, gli standard di comunicazione per bus di campo attualmente disponibili sono molti e presentano differenze, a volte, notevoli. Il periodo in cui l'attività di standardizzazione è stata avviata era sicuramente adatto per un suo sicuro successo. Verso la metà degli anni 80', infatti, erano presenti solo pochissimi standard di comunicazione per bus di campo, e, dunque, vi era una buona possibilità di poter imporre un unico standard di comunicazione internazionale. Il lavoro di standardizzazione si è però prolungato per troppi anni, a causa di azioni esplicitamente mirate ad ostacolarne la definizione. Solo da poco tempo l'attività di standardizzazione è praticamente conclusa, visto che pochi documenti devono essere ancora definiti. Una considerazione purtroppo da fare è che lo scenario attuale comporterà numerosi problemi all'adozione di una normativa

internazionale. La presenza, come detto, di numerosi e consolidati sistemi di comunicazione a bus di campo (proprietary e non), e la corrispondente chiara definizione delle fette di mercato in cui ciascun sistema di comunicazione ha praticamente il monopolio, ha reso impossibile la loro migrazione verso un'unico standard internazionale.

## 1.2 Ethernet

Il fenomeno Ethernet trainato dal mondo dell'informatica, ha ormai coinvolto molti settori, dalle telecomunicazioni fino al mondo industriale; da una ventina d'anni a questa parte si sta assistendo alla sua progressiva affermazione seguendo la scia della filosofia che vede nella tecnologia Web il nuovo punto di riferimento dei sistemi informatici personali, aziendali e industriali. In azienda Ethernet può essere utilizzata non solo a livello di supervisione e controllo ma anche per le comunicazioni con gli I/O. All'inizio, anche a causa del grande utilizzo già consolidato di trasmissioni seriali di dati tramite i bus di campo che sono riusciti ad imporsi come standard de facto, ma in particolar modo per le caratteristiche intrinseche stesse di Ethernet (protocollo non deterministico), esso non veniva preso in considerazione per lo scambio di dati di campo e per applicazioni remote. L'inizio e l'utilizzo di Ethernet nell'ambito dell'industria era legato quasi esclusivamente a scambi di dati dal quadro elettrico verso i livelli superiori, cioè dai sistemi a logica programmabile verso i sistemi di supervisione e poi da questi sistemi verso i livelli di fabbrica e/o impianto legati alla produzione e all'approvvigionamento fino alla parte finanziaria dell'azienda. Nel corso degli ultimi anni, si è però assistito ad un'evoluzione di Ethernet, attraverso l'utilizzo di particolari protocolli studiati appositamente per sopperire alle mancanze dei requisiti strettamente necessari per la comunicazione di campo si è arrivati alla risoluzione dei problemi di determinismo e real time delle reti. In un futuro non molto lontano (nel giro cioè di qualche anno) si potrà assistere a una situazione in cui i nodi venduti su protocolli Ethernet deterministici supereranno i nodi venduti per i fieldbus tradizionali.

Dal punto di vista tecnico Ethernet non è altro che una raccolta di specifiche relative al tipo di cablaggio, alla codifica dei segnali e alla gestione dei messaggi all'interno di una rete locale. Prima di tutto bisogna ricordare che nella scala ISO/OSI, che descrive l'architettura di qualsiasi rete di telecomunicazione, Ethernet indica solamente i primi due livelli (il mezzo fisico trasmissivo e il Data Link Layer) ed aggiungendo il protocollo TCP/IP si arriva fino a

livello 4 (trasporto dati). Questo è quello che viene generalmente utilizzato nelle familiari architetture tradizionali, basate cioè su IEEE 802.3 per il livello fisico e la parte del meccanismo di controllo di accesso (Medium Access Control - MAC) del livello scambio dati, IEEE 802.2 per la parte LLC (logical link control) del livello scambio dati, IEEE 802.1 per la parte di bridging del livello scambio dati e sui protocolli TCP/UDP/IP. L'introduzione di Ethernet nei sistemi di automazione dà la possibilità di poter utilizzare una singola architettura di rete attraverso tutti i livelli di impresa: dalle celle dei robot, alle linee di trasferimento, ai livelli dell'applicazione commerciale/gestionale. Ethernet ha, inoltre, la sua naturale evoluzione in Internet che consente una diagnostica decentralizzata di semplice implementazione. È per questa ragione che, anche a livello industriale, vanno affermandosi protocolli standard, quali quelli utilizzati dai comuni browser (IP, TCP, HTTP) e cominciano a comparire sul mercato i primi web-sensor, ossia sensori direttamente interfacciati su WEB e consultabili mediante un comune browser, anche se non pienamente compatibili con i protocolli in questione.

### 1.2.1 CSMA/CD

La caratteristica principale di Ethernet è quella di operare a basso livello (livello DataLink) con il protocollo CSMA/CD (Carrier Sense Multiple Access / Collision Detect); tale protocollo è studiato in modo da garantire l'arrivo di un telegramma destinazione attraverso la rete in qualunque modo, ma con una condizione: non è garantito alcun determinismo temporale, in quanto il tempo impiegato dal telegramma per arrivare a destinazione dipende dal traffico sulla rete, da eventuali errori che possono richiedere ritrasmissioni dello stesso, etc. Il funzionamento è semplice: prima di trasmettere su un mezzo dotato di accessi multipli (MA, Multiple Access), ogni nodo verifica, rilevando la presenza della portante, che nessun altro stia utilizzando il canale (CS, Carrier Sense). Affinché tutte le stazioni si rendano conto che il canale è occupato, la lunghezza minima del pacchetto è impostata a 72 Byte (Preambolo e SFD compresi). La trasmissione senza problemi dei primi 64 Byte del pacchetto, implica che il canale è libero e viene assegnato alla stazione trasmittente la quale porta a termine il trasferimento di dati. Una volta iniziata la trasmissione i nodi ascoltano ciò che si trova sul canale per verificare che non vi siano collisioni (CD, Collision Detect), ossia che altri nodi non abbiano iniziato a trasmettere prima di accorgersi che era già in atto una comunicazione. Questo è possibile per effetto del ritardo di propagazione dei segnali nel mezzo fisico, e quando viene individuata una collisione i

responsabili si arrestano subito e emettono un segnale denominato "Collision Presence" avvisando così tutte le altre stazioni dell'avvenuta collisione. A questo punto viene richiamato l'algoritmo di ripristino esponenziale binario (algoritmo di back-off) che posticipa la trasmissione di un intervallo di tempo random. Dopo un massimo di 16 tentativi la trasmissione viene annullata e il tempo di attesa riparte da zero. Una rete basata sul protocollo CSMA/CD è dunque intrinsecamente non deterministica in quanto i ritardi che si possono verificare a causa del ripetersi di collisioni tra pacchetti di dati non sono prevedibili a priori e ciò impedisce, di fatto, il poter calcolare entro quanti millisecondi il dato trasmesso arrivi alla stazione di destinazione. Oggi, con una accurata progettazione della rete e alla sempre maggiore velocità di trasmissione, è possibile fare in modo che le collisioni spariscono completamente e che i ritardi siano del tutto limitati e prevedibili.

### 1.2.2 Verso il determinismo

Come precedentemente descritto, storicamente l'impiego di Ethernet come bus di campo è risultato penalizzato dal non-determinismo e, per quanto concerne la scarsa messaggistica dei sensori e degli attuatori, dalla bassa efficienza, dato che il numero di byte trasmessi è elevato rispetto al numero di byte che effettivamente contengono l'informazione. Queste due grandi problematiche sono state affrontate e superate grazie all'introduzione delle seguenti innovazioni tecnologiche:

#### Fast Ethernet

Fast Ethernet è stato introdotto nel 1995 e rimase la versione più veloce di ethernet per tre anni prima di essere surclassato dalla Gigabit Ethernet; nonostante non sia lo standard ethernet più veloce nel mercato è senza ombra di dubbio quello più utilizzato. Fast Ethernet è un termine collettivo per indicare un numero di standard Ethernet che trasportano il traffico alla velocità di 100 Mbps rispetto alla velocità originale Ethernet di 10 Mbps; sebbene la velocità venga cambiata e i parametri temporali della rete vengano scalati di un fattore dieci (per esempio il *bit-time* passa da 100ns a 10ns), la struttura del frame e il protocollo CSMA/CD vengono mantenuti tali e quali [12]. In realtà molte applicazioni industriali non risentono un sostanziale giovamento nel semplice incremento di *data rate*; ad esempio, una rete a livello di campo è costituita principalmente da sensori e attuatori che solitamente consumano e producono una piccola quantità di dati che si riescono ad incapsulare in un



frame ethernet di 64 byte (la più piccola dimensione di un frame supportata da ethernet). Dunque le performance di questi device sono limitate più dalla velocità limitata del loro microprocessore e firmware piuttosto che dalla velocità della rete in sé. Tuttavia la tendenza di avere traffico eterogeneo su sistemi di comunicazione industriali, con applicazioni che utilizzano molta banda, come applicazioni video o voce altera, sicuramente questo scenario. Un aspetto invece che migliora significativamente è quello della gestione delle collisioni; infatti i tempi di back-off di Fast ethernet sono un decimo di quelli di una rete ethernet a 10 Mbit/s, e questo, in reti particolarmente cariche in cui le collisioni sono un vero problema, porta un netto giovamento. Si noti comunque che l'aumento del *data rate* in sé e per sé non elimina il problema del non determinismo; la soluzione a ciò avviene nel momento in cui si riescono a risolvere le collisioni in modo deterministico o, meglio ancora, eliminandole completamente.

### Switch e micro-segmentazione

Il problema delle collisioni è stato risolto connettendo i vari dispositivi tramite uno switch; esso è un dispositivo hardware che opera a livello Data-Link (livello 2 del modello ISO/OSI) e che inoltra il messaggio solo al destinatario [14]. In questo caso non abbiamo più dispositivi connessi su uno stesso bus ma ogni dispositivo è connesso ad una porta dello switch. Se la stazione A vuole parlare con la stazione B e la stazione C vuole parlare con la stazione D si creano all'interno dello switch dei circuiti per cui le informazioni passano direttamente da C e D e da A a B. Anche nel caso in cui la stazione A stesse parlando con la stazione B e anche D volesse dialogare con B non avremmo collisioni perché lo switch è un elemento con memoria. Può memorizzare quello che la stazione D voleva dire alla stazione B e spedirlo quando la stazione B ha finito di parlare con A. All'interno dello switch ci sono dei buffer che permettono di memorizzare i vari pacchetti nel caso la destinazione sia occupata. Questo permette, fino a quando il buffer non satura, di non perdere nessun pacchetto. Il costo di uno switch è in genere proporzionale alla dimensione di questi buffer e alle molteplici funzionalità aggiuntive di cui possono essere dotati. Dunque questi dispositivi sono in grado di isolare i domini di collisione attraverso la segmentazione della rete poiché ogni porta è configurata come un singolo dominio di collisione [5]. In questo modo si riesce effettivamente a ridurre o perfino ad eliminare il numero di collisioni. Gli switch forniscono una soluzione scalabile e flessibile ai problemi e alle limitazioni inerenti alle reti Shared ethernet (cioè reti a bus o basate sull'impiego di hub dove i nodi competono per l'accesso al mezzo), grazie all'uso

di nuovi meccanismi come la micro-segmentazione e le operazioni in full duplex. Le reti ethernet erano tipicamente basate sugli hub (specifica 10BaseT e 10Base2) o su bus (specifica 10Base5), dove i nodi della rete condividevano lo stesso mezzo fisico; questo significa che solo un singolo nodo alla volta poteva inviare dati. Uno switch è in pratica un hub intelligente che può leggere e processare l'indirizzo di destinazione dei dati entranti e mandarli unicamente alle porte richieste. Mentre in un hub/repeater, i dati inviati da un nodo vengono mandati in broadcast a tutti i nodi della rete, in uno switch, i dati vengono inviati unicamente ai nodi destinazione, il che significa che più nodi possono trasmettere o ricevere contemporaneamente [13]. Anche un bridge riesce a far ciò, ma deficiata di alcune innovazioni tecnologiche introdotte dagli switch come VLAN, flow control etc; inoltre uno switch può avere svariate porte mentre un bridge ne ha tipicamente solo due. Se la segmentazione in uno switch è portata all'estremo, ogni device è isolato nel suo segmento e ha l'intera banda della porta a sua disposizione. La micro-segmentazione rimuove una delle cause primarie di congestione nelle LAN dal momento che ci può essere solo una stazione che invia dati nel segmento ad ogni istante di tempo. In queste condizioni l'onere della risoluzione delle congestioni è affidata allo switch il quale è costruito apposta per riuscire a gestire il carico di traffico di tutti i device a lui collegati. Se molti device provano a comunicare contemporaneamente con un singolo device, allora è possibile avere un problema di congestione nonostante ogni device abbia la sua LAN (la tecnica del flow-control deve essere usata per superare questo ostacolo) [5].

### Funzionalità aggiuntive degli switch

Si elencano brevemente solo alcune tra le più importanti funzionalità dei moderni switch [14]:

- **Flow Control:** è un meccanismo atto a limitare il carico della rete e che può essere cruciale in alcune situazioni per evitare il degrado delle prestazioni delle reti di comunicazione real-time. Esso elimina la possibilità di avere pacchetti persi in porte full-duplex congestionate avvisando le stazioni che stanno sovraccaricando la rete.
- **VLAN:** permette ad uno switch di raggruppare in modo logico più device e di isolare il traffico tra questi gruppi anche se tutti i device condividono lo stesso mezzo trasmissivo.

- **Autonegotiation:** si intende il riconoscimento automatico e la negoziazione della velocità di trasferimento (10-100 Mbit/sec) e del tipo di funzionamento (full-duplex/half-duplex).

Oltre a quelle elencate, gli switch ‘intelligenti’, hanno svariate funzionalità di diagnostica, sicurezza, analisi di rete, ridondanza, port mirroring etc [14].

### Full Duplex

Quando Ethernet fu standardizzata nel 1985, le sue modalità di comunicazione erano half-duplex: in pratica, un nodo poteva solo trasmettere o ricevere nello stesso istante ma non poteva fare le due azioni contemporaneamente. Inoltre, i nodi che condividevano collegamenti half-duplex, appartenevano allo stesso dominio di collisione; cioè questi nodi sono in ‘competizione’ per l’accesso al bus e i loro frame possono collidere con altri frame nella rete. La comunicazione Full-duplex fu standardizzata per Ethernet nel 1997; con essa si richiedono percorsi separati per trasmettere e ricevere dati e si applicano a segmenti che consistono di non più di due device (tipicamente un nodo e uno switch o due switch, ma non hub perchè aumentano la probabilità di collisioni). In questo modo ricezione e trasmissione simultanee sono possibili, inoltre ogni nodo ha un unico dominio di collisione, eliminando il pericolo di collisioni e ignorando quindi il protocollo tradizionale CSMA/CD [5], [13].

### Priorità

Un ulteriore passo nella direzione di un maggior determinismo è stato fatto dall’IEEE con la specifica 802.1p che si occupa della Classe del Servizio (CoS, Class of Service) e che copre l’aspetto della prioritizzazione dei messaggi. La specifica prevede la marcatura del frame Ethernet con un livello di priorità compreso tra 0 e 7 che viene utilizzato dagli switch per stabilire l’importanza nello smistare i messaggi memorizzati nel loro buffer e riducendo ulteriormente l’incidenza del traffico non critico sui dati di maggior importanza. La definizione dei livelli di priorità viene realizzata utilizzando tre bit di una estensione di 4 byte prevista per gli standard 802.1p e 802.1q che modificano la trama originaria di Ethernet portandola ad una lunghezza massima di 1522 byte, preambolo e delimitatore esclusi [5].

## Le limitazioni di Switched Ethernet

Lo switch è un componente fondamentale nella realizzazione di una rete Ethernet, infatti le sue caratteristiche possono influire pesantemente sulle prestazioni finali. La classificazione tradizionale degli switch prevede la divisione in tre grosse classi (Store & Forward, Cut-Through e Fragment-Free) a cui appartengono tutti i componenti attualmente in commercio. La conoscenza delle caratteristiche di ognuno permette la stima dei parametri della rete e dei relativi limiti raggiungibili. Sicuramente lo switch più affidabile è quello denominato Store & Forward, poiché esegue un controllo completo di ogni pacchetto ricevuto prima di inoltrarlo alla destinazione: eventuali pacchetti incompleti, danneggiati o non corretti vengono eliminati liberando così banda per i dati corretti. Il prezzo da pagare è però un ritardo variabile aggiunto ad ogni pacchetto poiché, per eseguire il controllo, tutto il pacchetto deve essere ricevuto e immagazzinato all'interno dello switch. La strategia adottata dagli switch di tipo Cut-Through è invece totalmente differente: vengono ricevuti e memorizzati solo i byte iniziali di un pacchetto Ethernet, i primi 14, necessari a capire la destinazione del pacchetto; subito dopo può iniziare la ritrasmissione, minimizzando così il ritardo. Naturalmente errori di CRC (Cyclic Redundancy Check) o pacchetti incompleti possono passare indisturbati attraverso uno switch di questo tipo, che si occupa più di ridurre i ritardi piuttosto che di garantire una adeguata affidabilità. Per ultimo gli switch di tipo Fragment-Free sono una versione più 'sicura' degli switch Cut-Through, visto che controllano anche che il singolo pacchetto sia più lungo di 64 byte (la minima lunghezza di un pacchetto Ethernet), impedendo di fatto l'inoltro di frame spuri. Come esempio di stima del tempo di propagazione di un pacchetto all'interno di uno switch si possono fare le ipotesi seguenti: in una rete di tipo 100BaseT ( $T_{bit} = 10$  ns) il tempo di propagazione  $T_p$  è in prima approssimazione la somma di due termini: un ritardo interno allo switch  $T_{sid}$ , dovuto ai componenti elettronici utilizzati, alla potenza di calcolo e in ultima analisi all'implementazione hardware e un ritardo  $T_d$  relativo al buffering dei dati prima della ritrasmissione. Solo nel caso dello switch Store e Forward il tempo  $T_d$  è variabile in un range compreso tra  $5.76 \mu s$  e  $122.08 \mu s$  (rispettivamente con frame da 64 byte e da 1518 byte, preambolo e SFD esclusi), mentre negli altri due il ritardo è costante, rispettivamente  $1.12 \mu s$  per il Cut-Through e  $5.76 \mu s$  per il Fragment-Free. L'altro ritardo  $T_{sid}$ , quello interno, è specifico del dispositivo fisico e in generale può essere considerato proporzionale al carico elaborativo, nel senso che più complessa è l'elaborazione da fare sui dati del pacchetto, maggiore sarà il tempo richiesto. Come conseguenza appare evidente che il  $T_{sid}$  di un Cut-Through sarà minore di quello di un Store & Forward. Generalizzando queste considera-

zioni e applicandole alla topologia reale della rete, se ne deduce che il tempo impiegato da un pacchetto per muoversi da una stazione di rete ad un'altra, dipende fortemente dal numero di volte che i frame passano all'interno degli switch [15]. Inoltre se si considera il caso di traffico multiplo il jitter aumenta in modo non deterministico. Questa variabilità imprevedibile è causata dal fatto che il tempo di arrivo di un pacchetto all'ingresso di uno switch e la sua destinazione (ossia la porta verso la quale dovrà essere inoltrato) siano completamente indipendenti da ciò che lo switch sta facendo in quel momento. È infatti possibile che i dati siano destinati ad un canale che in quel momento è già utilizzato e quindi il nuovo pacchetto deve rimanere in una coda in attesa del suo turno di trasmissione: questo introduce un jitter sul tempo di trasmissione.

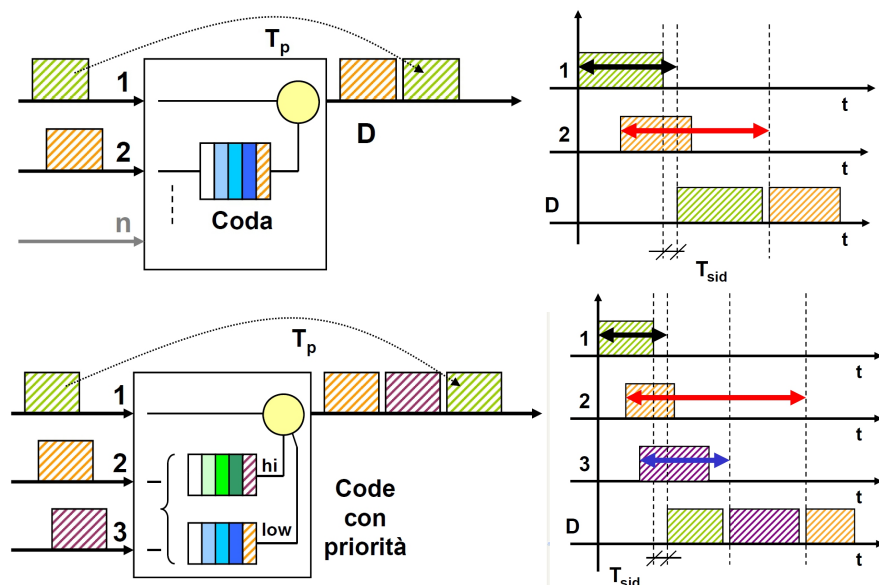


Figura 1.1: Switch senza e con gestione delle priorità

Anche in caso di utilizzo di switch aderenti allo standard IEEE 802.1q/1p, che introduce il concetto di priorità tra i frame, non risolve il problema. Infatti, l'uso dei *priority tag* per rendere un pacchetto più prioritario rispetto ad un altro, garantisce che un frame ad alta priorità sorpassi un frame a bassa priorità nello switch, ma non permette di interrompere un frame già in fase di trasmissione per trasmetterne uno prioritario. In definitiva il problema dei ritardi random non è risolto completamente. Concludendo questa rapida introduzione, si può affermare che il solo utilizzo di Ethernet standard non garantisce il determinismo in caso di reti trafficate [15]. Quindi per garantire applicazioni di Hard Real-Time (HRT) richieste specialmente

nel motion control sono necessarie delle migliorie da apportare all'hardware per ridurre questo jitter (ad esempio il controllore di rete offerto da Siemens per le reti Profinet si basa su un ASIC (Application Specific Integrated Circuit) con elevate prestazioni denominato ERTEC che permette un'accurata sincronizzazione hardware tra i dispositivi).

## TDMA e IEEE 1588

Specialmente in reti di grandi dimensioni con molte catene di switch il jitter nel trasferimento dati real-time può crescere in modo significativo. Questo porta ad una restrizione nell'uso di ethernet in applicazioni isocrone tempo critiche come il motion control. Per sopperire a tale mancanza bisogna introdurre delle estensioni allo standard che spesso sono basate sul metodo TDMA (Time Division Multiple Access); così facendo si preallocano degli slot di tempo per il trasferimento sia di dati real-time che di dati non temporalmente critici. L'applicazione di un tale schema di scheduling ha il vantaggio che i tempi di trasmissione sono calcolati a priori, d'altro canto però l'uso di componenti hardware standard deve essere fortemente limitato.

Uno standard ormai consolidato per la sincronizzazione è il Precision Time Protocol (PTP) detto anche IEEE 1588 che è basato su una procedura master/slave [13], [11]. Prima di tutto il *Best Master Clock Algorithm* (BMC) determina il miglior clock in termini di affidabilità e accuratezza di tutto il sistema e di ogni sottorete; il dispositivo con il miglior clock in assoluto viene chiamato *GrandMaster Clock* (GMD) mentre quello di ogni sottorete viene chiamato *master clock*. Ci può essere un solo GMC per tutta la rete e un solo master clock per ogni sottorete. Lo standard IEEE 1588 definisce due tipi di clock: *Ordinary Clock* impiegato nei normali device di rete e *Boundary Clock* usato in componeti come gli switch o hub dove ci sono più porte e quindi più segmenti di comunicazione. Il processo di sincronizzazione avviene in due fasi. Per prima cosa viene corretto l'offset tra master e slave. Per far ciò il master manda un messaggio di sincronizzazione (*SYNC message*) contenente una stima del tempo in cui il messaggio lascerà fisicamente il master. Parallelamente a questo, viene misurato il tempo in cui il messaggio realmente lascia il master nella maniera più accurata possibile, meglio se con supporto hardware direttamente nel mezzo trasmissivo (si noti come tutte le misure di tempo che richiedono il massimo della precisione, vengano eseguite il più vicino possibile al livello fisico per eliminare i ritardi dello stack di comunicazione, mentre i tempi stimati sono calcolati dall'algoritmo di IEEE 1588 a livello applicazione). A questo punto il master invia agli slave un secondo messaggio (*follow-up message*) contenente il tempo esatto di invio del *sync*

*message*. Questi a loro volta misurano il tempo esatto di ricezione dei messaggi e possono calcolare e correggere i loro offset.

La seconda fase della sincronizzazione, il *delay measurement*, determina il tempo di comunicazione tra master e slave; ciò avviene in maniera simile a prima mediante i messaggi di *Delay-Request* (da parte dello slave) e *Delay Response* (da parte del master), e i clock vengono regolati di conseguenza. Si noti come quest'ultimo è un ritardo simmetrico tra master e slave che è fondamentale per misure di ritardo e per la loro accuratezza, quindi se ci fossero componenti come switch o hub tra i due utenti i ritardi verrebbero falsati. Ecco il motivo per cui viene definito il *Boundary Clock*, il quale viene sincronizzato da un master e a sua volta si comporta come un master in ogni porta dello switch sincronizzando tutti gli slave connessi. In questo modo la sincronizzazione avviene sempre in connessioni punto punto che offrono tempi di esecuzione simmetrici e praticamente assenza di jitter. Questo protocollo ha subito poi successive integrazioni, soprattutto nei sistemi che sfruttano Ethernet Real Time, per via dei ritardi di comunicazione asimmetrici introdotti dagli switch (Profinet risolve il problema, per esempio introducendo il concetto di *Bypass Clock* per il traffico PTP).

Grazie a questo algoritmo e ad un appropriato supporto hardware si possono raggiungere sincronizzazioni inferiori a 1  $\mu s$ .

## 1.3 Industrial Ethernet

Come già detto, da qualche anno a questa parte nel settore dell'automazione industriale si sta cercando di introdurre la rete Ethernet anche a livello di campo. La rete Ethernet è molto diffusa a livello mondiale e l'introduzione di Ethernet a livello industriale può portare diversi vantaggi. Vantaggi economici perchè le cose diffuse costano poco. Un altro vantaggio consiste nella possibilità di utilizzare Internet a livello industriale. Con il termine Industrial Ethernet ci si riferisce in genere a quelle applicazioni che utilizzano Ethernet per le comunicazioni a livello di campo, ossia di PLC e periferia. Nel corso degli ultimi anni, si è assistito ad un'evoluzione di Ethernet, attraverso l'utilizzo di particolari protocolli studiati appositamente per sopperire alle mancanze dei requisiti strettamente necessari per la comunicazione di campo, si è arrivati alla risoluzione dei problemi di determinismo e real-time delle reti. Non essendoci uno standard imposto, ciascun costruttore ha implementato una soluzione consona alle proprie esigenze con la conseguente proliferazione di svariati protocolli applicativi. L'introduzione di Ethernet a livello di cella e

di campo nel panorama industriale garantirebbe più flessibilità, nuovi mezzi trasmissivi (come la tecnologia wireless), la possibilità di usare tutti i protocolli IT e quindi anche il supporto diretto di Internet che consentirebbe ad esempio di creare interfacce remote per i dispositivi usando un normalissimo Web browser. Ma sarebbe inaccettabile se l'adozione della tecnologia Ethernet portasse alla perdita di tutte quelle caratteristiche essenziali nel mondo della comunicazione industriale, ovvero:

- affidabilità;
- robustezza (EMC, robustezza meccanica);
- comunicazione tempo-deterministica;
- azioni tempo-sincronizzate tra periferiche di campo;
- efficiente scambio di dati.

Nei processi a livello di campo la comunicazione deve soddisfare le condizioni appena citate, richiedendo esigenze temporali assai stringenti e spesso molto differenti tra loro.

### 1.3.1 Parametri Real-Time

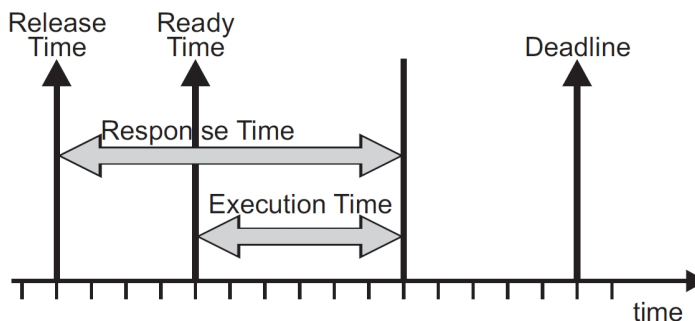
In sostanza, un sistema real-time deve garantire che una elaborazione (o task) termini entro un dato vincolo temporale o scadenza (detta in gergo *deadline*). Per garantire questo è richiesto che la schedulazione delle operazioni sia fattibile. Il concetto di fattibilità di schedulazione è alla base della teoria dei sistemi real-time ed è quello che permette di dire se un insieme di task sia eseguibile o meno in funzione dei vincoli temporali dati. I sistemi RT sono quelli che non dipendono solamente dalla validità dei dati scambiati ma anche dalla loro tempestività. Un corretto sistema RT, dunque, garantirà sia il funzionamento corretto del sistema sia la sua puntuale e deterministica esecuzione. I sistemi real-time generalmente sono suddivisi in due principali sottocategorie [13]:

- **Soft Real-Time (SRT)**: cioè un sistema che se non rispetta la sua scadenza (in gergo si dice sfondare la deadline) provoca un danno non irreparabile;
- **Hard Real-Time (HRT)**: cioè un sistema che nel caso superi temporalmente la sua deadline provoca un danno irreparabile al sistema.



I ‘blocchi’ costituenti i sistemi RealTime sono detti *Jobs*; ogni job RT ha alcune quantità temporali da rispettare [13], (si veda figura 1.2):

- **Release Time:** tempo in cui un job diventa disponibile al sistema;
- **Ready Time:** è il tempo entro il quale il job viene completamente processato;
- **Execution Time:** è l’intervallo tra il Ready Time e il completamento dell’esecuzione;
- **Response Time:** è il tempo più breve in cui un job può iniziare l’esecuzione (mai inferiore al Release Time);
- **Deadline:** è il tempo entro il quale l’esecuzione deve essere terminata, oltre il quale il job è in ritardo. Una deadline può essere hard o soft a seconda delle specifiche temporali del job.



**Figura 1.2:** *Specifiche dei jobs*

Tutti i sistemi Real-Time hanno un certo livello di jitter (variazione sul tempo ideale); quest’ultimo nei sistemi RT deve essere misurabile in modo tale da poter garantire le prestazioni del sistema. Per poter sviluppare un sistema distribuito Real-Time, è di vitale importanza fornire un tipo di comunicazione dotata di protocollo RT, altrimenti le qualità temporali finora citate andrebbero del tutto perse. Le comunicazioni real-time sono come qualsiasi sistema real-time: possono essere hard o soft a seconda dei requisiti e i loro ‘jobs’ includono la trasmissione dei messaggi, la propagazione e la ricezione.

### 1.3.2 Performance indicators

Per definire una comunicazione Real-Time basata su Ethernet (RTE) si sono introdotti alcuni parametri chiamati Performance Indicators riportati di seguito e specificati nella IEC 61784 [8], [30]:

- **Delivery time:** è il tempo necessario per il trasferimento di una *service data unit* (SDU) da un nodo sorgente ad un nodo destinazione. Tale tempo si misura a livello applicazione e il valore massimo che può assumere dipende dai due casi di assenza di errori di trasmissione e di perdita e recupero di un frame. Nel calcolo del delivery time si deve tener conto del tempo di trasmissione e di tutti i tempi di attesa.
- **Numero di nodi:** Ogni CP <sup>1</sup> impone un numero di stazioni massime che possono essere collegate alla rete. Un numero eccessivo di stazioni renderebbe il traffico soggetto a tempi di attesa non compatibili con la gestione real time.
- **Topologie di rete:** le varie topologie di rete influenzano il traffico delle informazioni scambiate attraverso la rete; le topologie base sono: stella, anello e lineare.
- **Numero di switch tra due nodi:** Ogni CP impone un numero massimo di switch tra due stazione RTE; da tale parametro si può definire una possibile conformazione della rete.
- **Throughput RTE:** rappresenta l'insieme di dati APDU (per ottetti) scambiati tramite un collegamento nel tempo di un secondo.
- **Banda non RTE:** indica la percentuale di banda che può essere destinata alla comunicazione non Real-Time in un collegamento.
- **Accuratezza di sincronizzazione:** indica la deviazione temporale massima dei clock appartenenti a due diversi nodi della rete.
- **Tempo di recupero rindondante:** rappresenta il tempo massimo che intercorre tra il fallimento di un singolo evento e il completo ripristino del sistema considerando il caso che il fallimento sia permanente. In pratica si può stabilire che il *redundancy recovery time* sia il delivery time di un messaggio nel caso di un fallimento permanente.

---

<sup>1</sup>Communication Profile, ciascuno dei quali appartiene ad una Communication Profile Family (CPF), definite nello standard IEC 61784. Per esempio Profinet IO corrisponde ai communication profile CP 3/4, CP 3/5 e CP 3/6 dello standard IEC 61784-2; RT Class 2 e 3 coprono il CP 3/6 mentre RT Class 1 corrisponde ai CP 3/4 e 3/5.

### 1.3.3 Protocolli Industrial Ethernet

Dal momento che le reti di comunicazione industriali sono utilizzate in differenti ambienti per differenti scopi, ci sono stati diversi approcci per utilizzare ethernet come rete di campo a seconda anche del tipo di prestazioni richieste. Gli approcci possono essere divisi essenzialmente in tre categorie [8]. Come si può vedere in figura 1.3 dalla categoria 3 alla 1 aumentano le prestazioni del protocollo ma anche le funzionalità che bisogna aggiungere al livello data link. Per ridurre al massimo i tempi è necessario un intervento nella procedura di scheduling a livello MAC di ethernet [9]. Si possono così distinguere tre tipologie di approccio:

- **Incapsulamento:** l'incapsulamento si ha quando un pacchetto dati di alto livello viene inglobato in un pacchetto standard TCP o UDP. In questo caso, ethernet e i livelli TCP/IP dello stack di comunicazione (dal livello 1 al 4 di OSI) vengono lasciati funzionare in modo standard e nuove funzionalità vengono inserite al Livello 7. I dati di campo vengono trasportati dal TCP/IP come 'user data' e l'hardware standard ethernet può essere utilizzato. Questo è il modo più semplice e immediato di usare ethernet a livello industriale. Per stabilire un accettabile livello di determinismo, il controllo è normalmente applicato al processo di comunicazione ethernet dalle funzioni di più alto livello che determinano il flusso di informazioni al livello applicazione. Questo può essere raggiunto sviluppando nuovi protocolli di livello 7, o aggiungendo funzionalità di livello ancora più alto al di fuori del modello OSI, come nel caso di Foundation Fieldbus HSE. Esempi di RTE che usano questo approccio sono Modbus/TCP, Ethernet/IP, P-NET, Vnet/IP, Foundation Fieldbus HSE.
- **Incapsulamento diretto al livello 2:** Alcuni protocolli usano la struttura ethernet standard al livello 1 e il MAC standard a livello 2, ma incapsulano i dati direttamente, non facendo uso quindi del protocollo TCP/IP. Questo diminuisce l'overhead del pacchetto ethernet e inoltre migliora il determinismo. In pratica, questa opzione si trova usualmente in parallelo all'uso di TCP/IP. Applicazioni SRT possono usare l'incapsulamento diretto, ma lo stack TCP/IP permette funzionalità aggiuntive per la gestione della rete usando tool standard. Fortunatamente Ethernet fornisce una schema di prioritizzazione nel quale il traffico SRT viene incapsulato con una più alta priorità rispetto al TCP/IP. Queste realizzazioni RTE non alterano l'hardware di comunicazione ethernet in alcun modo, ma sono realizzate specificando uno speciale tipo di protocollo (Ethertype) nel frame ethernet (Per esempio

l'Ethertype di Profinet è 0x8892). Esempi di RTE che usano questo approccio sono TCnet, EPA, Ethernet Powerlink, Profinet RT (Class 1).

- **Ethernet modificata:** L'unico modo per assicurare la funzionalità isocrona con risposte sotto i millisecondi è cambiare il modo in cui opera Ethernet. In questi casi, la connessione fisica di ethernet è mantenuta (gli stessi cavi e infrastrutture base) ma la funzionalità del protocollo ethernet è direttamente modificata al livello 2 per fornire capacità di comunicazione deterministica. Questo richiede lo sviluppo di interfacce hardware con speciali ASIC o FPGA. La compatibilità con protocollo di livello più alto come HTTP e SNMP può essere raggiunta incapsulando questi dati di alto livello in timeslot definiti. Quindi il determinismo è implementato attraverso una procedura ad istanti fissi di tempo che fornisce cicli di trasmissione prestabiliti e attraverso una sincronizzazione ciclica di tutti i nodi (come lo standard IEEE 1588). Molto spesso la funzionalità di switch è già integrata all'interno del dispositivo di campo. Le modifiche sono obbligatorie per tutti i device all'interno del segmento di rete real-time, ma permettono parimenti la trasmissione di traffico non real-time. Esempi di RTE che usano questo approccio sono: Sercos III, EtherCAT, Profinet IRT (Class 2 e 3).

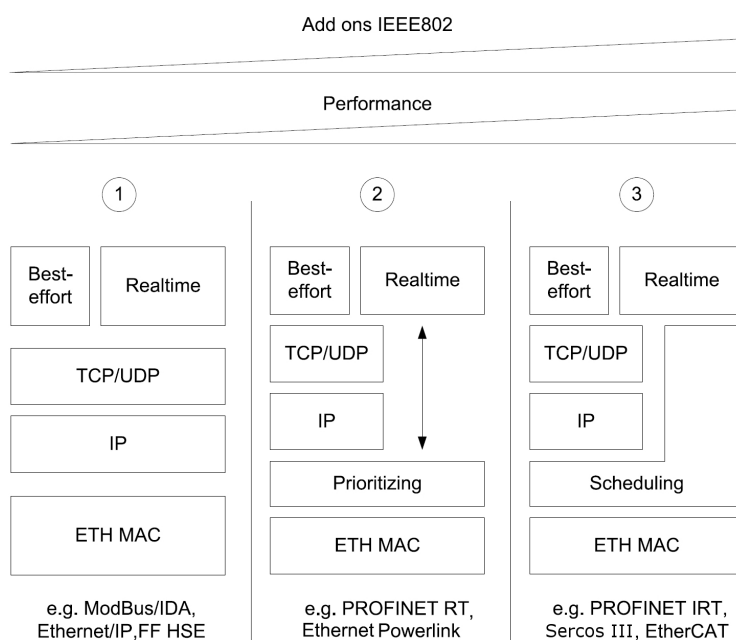


Figura 1.3: *Protocolli per ethernet industriali*

# Capitolo 2

## Profinet IO

### 2.1 Introduzione

Profinet è la soluzione Industrial Ethernet offerta dal *Profibus International User Group*. Oggettivamente, Profinet è più di un protocollo, è un nuovo modo di concepire l'architettura di un sistema di automazione industriale. I suoi obiettivi principali sono la drastica riduzione dei costi di progetto e messa in servizio di un impianto, e l'integrazione tra la nuova automazione di dispositivi basati su Ethernet e quelli tradizionali basati sui bus di campo. Attualmente, per quanto riguarda la sola parte di comunicazione dati, Profinet raggruppa sotto il suo nome due famiglie di protocolli entrambi basate su Ethernet: Profinet CBA (Component Based Automation) e Profinet IO. Il primo, CBA, è dedicato all'integrazione ad alto livello, per esempio tra differenti linee di produzione, e offre il supporto per l'integrazione con OPC, con Internet e il Web, nonché con Profibus e virtualmente con tutti gli altri bus di campo. Il secondo, Profinet IO, è stato invece progettato per dotare i dispositivi di campo di interfaccia Ethernet e garantire delle performance real-time e deterministiche [17].

## 2.2 Profinet CBA

Profinet CBA definisce un ambiente orientato agli oggetti che abbia la fase di engineering molto semplificata, in modo da permettere dei risparmi notevoli. Come esempio di ingegnerizzazione semplificata, basti pensare all'ambiente grafico dove i vari oggetti dell'automazione, rappresentazioni delle funzionalità (es. 'tornio', 'pressa', 'verniciatura', 'imballaggio'), sono collegati fra di loro ad alto livello semplicemente tracciando una linea tra i blocchi. Le applicazioni che usano CBA per definire lo scambio dati non sono critiche dal punto di vista temporale: si tratta di applicazioni a livello di SCADA (Supervisory Control And Data Acquisition) dove l'uso di protocolli standard è fortemente consigliato se non obbligatorio. Come conseguenza Profinet CBA supporta solo la comunicazione Non-RT basata su DCOM <sup>1</sup> (e quindi TCP/IP) e RT Class 1 per applicazioni con esigenze real time poco critiche. Il vantaggio di questo approccio è che l'uso del noto e diffuso DCOM permette un'integrazione trasparente con OPC <sup>2</sup> e con gli SCADA.

## 2.3 Profinet IO

Più recente è l'introduzione del protocollo Profinet IO che permette di soddisfare requisiti temporali più stringenti. Il suo obiettivo primario è l'introduzione della tecnologia Profinet anche al livello dei moduli di I/O che oggi vengono gestiti tramite bus di campo. Profinet IO supporta tutti i protocolli di Profinet anche se per scopi diversi: il protocollo Non-RT, basato su UDP/IP, è infatti usato solo nella fase di configurazione, mentre i protocolli RT Class 1 e Class 2,3 (IRT) sono dedicati all'automazione real-time fino al motion control. Per garantire il real time e contemporaneamente il supporto alle comunicazioni generiche di tipo TCP, viene adottato un modo di accesso al mezzo di tipo TDMA: si stabilisce un ciclo altamente sincronizzato tra tutti i partecipanti al bus e ciascun ciclo è diviso in più fasi, ciascuna delle quali contiene tipi di traffico differenti. In questo contesto la sincronizzazione tra i vari elementi del bus diventa fondamentale e va affidata a speciali meccanismi. Con Profinet IO viene introdotto l'uso dello standard IEEE1588 utilizzato per la sincronizzazione di sistemi distribuiti. Questo però da solo

---

<sup>1</sup>Acronimo usato per Distributed Component Object Model; è una tecnologia informatica di Microsoft. È un'interfaccia per componenti software che permette la comunicazione tra processi e creazione dinamica di oggetti.

<sup>2</sup>Acronimo di OLE for Process Control; è uno standard industriale per il trasferimento dati. Lo standard specifica la comunicazione real-time tra device di diversi produttori.

non è sufficiente perché la IEEE1588 suppone che il ritardo tra due stazioni sia simmetrico, mentre in una rete Ethernet con switch ciò non è vero; inoltre non supporta le topologie ad anello (molto usate a livello industriale) dove, per garantire la ridondanza, un pacchetto può arrivare alla medesima stazione seguendo due percorsi diversi. Queste limitazioni sono state affrontate e risolte da Profinet che ha introdotto un nuovo tipo di switch capace di riconoscere le fasi operative del ciclo Profinet ed agire di conseguenza. In più, questi switch sono in grado di interpretare e ‘correggere’ i messaggi di tipo IEEE1588 per la sincronizzazione del sistema, eliminando di fatto i problemi legati alle asimmetrie. Queste funzionalità sono state integrate all’interno degli ASIC Ertec 200 e Ertec 400 prodotti da Siemens. Da questo momento in poi si farà sempre riferimento al protocollo Profinet IO, con il quale è stata allestita la rete per il lavoro di tesi; verranno spiegate in dettaglio le caratteristiche di tale protocollo (stack di comunicazione, modalità di scambio dati, cicli di comunicazione, etc) e i motivi per i quali è uno dei più performanti protocolli in ambito ethernet industriale.

## 2.4 Classi Real-Time

Come già accennato, il protocollo Profinet IO è in grado di gestire svariati tipi di traffico Real-Time che sono stati suddivisi nelle seguenti classi [30], [20]:

- **Non-RT** viene considerato per la trasmissione di dati di configurazione oppure per implementare cicli che superino le centinaia di millisecondi e che tollerino alta variabilità (più del 100%). Per la comunicazione si fa affidamento sui protocolli standard quali TCP e UDP (rispettivamente con e senza connessione). Vengono supportati anche numerosi servizi IT come per esempio:
  - Gestione della rete (DHCP)
  - Diagnosi di rete (SNMP)
  - Accesso remoto tramite WEB (HTTP)
  - Funzioni di posta elettronica (e-MAIL)
- **RT Class 1** può realizzare cicli di pochi millisecondi (tipicamente 2-16 ms) a variabilità contenuta (15%). Essa si basa sullo standard Ethernet 802.1Q in cui i dati vengono trasmessi tramite telegrammi Ethernet con

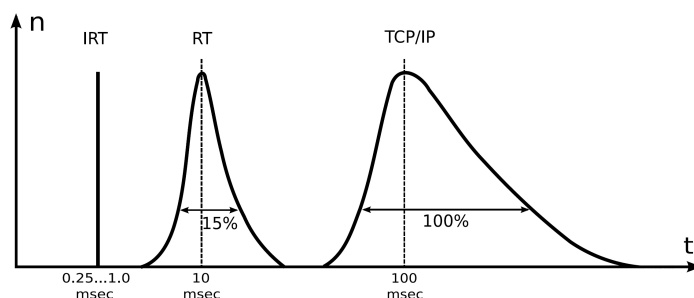
priorità come descritto nel capitolo precedente. Per implementare tale classe è sufficiente hardware ethernet standard.

- **RT Class 2** (denominata IRTflex): i frame in questa fase possono essere trasmessi con o senza sincronizzazione della comunicazione. Nel caso non sincronizzato la RT Class 2 si comporta allo stesso modo della RT Class 1.

La trasmissione sincronizzata consente tempi di ciclo tipici di 1 ms e tutti i telegrammi possono essere inviati e ricevuti in una finestra temporale definita in fase di progetto e riservata al solo traffico IRTflex. Questo permette una garanzia maggiore ma richiede tecniche di sincronizzazione del clock tra le stazioni. Il jitter massimo si attesta intorno ai 10  $\mu s$ . L'instradamento dei messaggi è fatto in base all'indirizzo MAC e serve hardware dedicato per la sincronizzazione (ASIC).

- **RT Class 3** (denominata IRTtop) è il protocollo di comunicazione più performante riservato ad applicazioni Motion Control. Rispetto a IRTflex si ha una maggiore pianificazione temporale del traffico ottenuta tramite la sincronizzazione tra il livello di rete e l'applicazione software che gestisce i dati. Il ciclo di invio è selezionabile da un massimo di 4 ms a un minimo di 250  $\mu s$  con intervalli di 125  $\mu s$  e il jitter è garantito ben al di sotto del  $\mu s$ . Il completo determinismo ottenuto prevede precisioni di sincronismo maggiori gestite direttamente dall'hardware (ASIC). La conoscenza della topologia di rete è fondamentale per la determinazione dei ritardi di trasmissione; questo implica che, in caso di modifiche sulla topologia o sui rapporti di comunicazione, sia necessario riprogettare la configurazione.

In figura 2.1 si possono notare le caratteristiche temporali delle varie classi (RT Class 2 e 3 vengono citate come IRT mentre RT Class 1 come RT).



**Figura 2.1:** *Caratteristiche temporali della comunicazione Profinet*



## 2.5 Periodo di trasmissione

Profinet IO effettua lo scambio di dati su un ciclo altamente ripetibile, il quale è descritto nello standard IEC 61158-5-10 e illustrato in figura 2.2. Un messaggio di sincronizzazione, *Sync Frame*, individua l'inizio del ciclo durante il quale si possono distinguere diverse fasi:

- **RED phase:** In questa fase solo i messaggi RT class 3 possono essere inviati in un fissato momento attraverso un percorso definito (topologia molto rigida). Questo richiede che tutti i dispositivi Profinet IO IRT sappiano precisamente quando e verso quale porta è permesso comunicare o mettersi in ascolto.
- **ORANGE phase:** Solo frame di tipo RT class 2 sono trasmessi durante questa fase. Anche questi messaggi devono essere scambiati con precise scadenze temporali, ma, in questo caso, il percorso fisico non è definito.
- **GREEN phase:** Questa fase è composta da messaggi che utilizzano le priorità Ethernet, scandite dallo standard IEEE 802.1Q. La comunicazione durante la fase verde (GREEN) viene usufruita da dispositivi di classe RT 1, da dispositivi RT Class 2 che hanno frame extra da inviare o che non sono tra loro sincronizzati e dai protocolli TCP/UDP.
- **YELLOW phase:** Questo periodo di transizione è utilizzato dallo stesso tipo di traffico della fase GREEN. Durante questo periodo solo i frame che possono essere trasferiti completamente entro la fine della fase stessa sono effettivamente trasmessi.

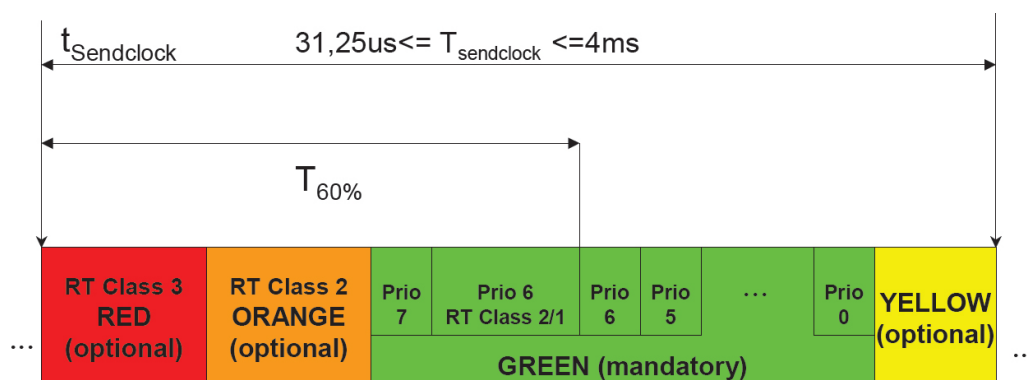


Figura 2.2: *Periodi di trasmissione*

Una porzione rilevante del ciclo (durante la fase verde) è lasciata alla comunicazione non real-time (NRT), come TCP e UDP. Questa tipologia di traffico ha caratteristiche di bassa priorità e tempi di ritardo molto variabili, infatti il traffico di tipo IP è usato per grandi trasferimenti di dati e la sola cosa significativa è la banda. In Profinet IO la fase NRT occupa almeno il 40% della totale larghezza di banda. La RT Class 3 è la più performante: non devono esserci ritardi di trasmissione o ricezione, perché la sequenza di messaggi in ogni ciclo è nota ‘a priori’ e sempre identica. Il tool della rete di configurazione calcola il percorso per ogni frame del ciclo e trasferisce le tempistiche alla rete di infrastruttura, di conseguenza il traffico isocrono deve essere gestito da un’opportuna rete con una rigida topologia, la quale deve essere determinata durante la configurazione (si tiene conto anche dei ritardi di propagazione su cavo). Sui frame non è effettuato alcun controllo sull’indirizzo MAC e il riferimento fondamentale sono le tempistiche. Se un messaggio atteso non arriva a destinazione, viene trasmesso un frame ‘dummy’ contenente una segnalazione d’errore (STATUS=”bad”), mentre se un frame non può essere trasmesso, per esempio se la porta di destinazione è occupata, viene scartato. Proprio per queste motivazioni la durata della RED phase deve essere maggiore della più lunga propagazione di ritardo tra sorgente e destinazione di qualsiasi frame IRT. Inoltre per ogni switch di rete viene costruita una tabella con l’elenco di tutti i messaggi da spedire/ricevere in un ciclo e il relativo tempo di scheduling. Si può anche notare che la RT Class 3 presenta un jitter molto basso (inferiore al microsecondo), dal momento che ogni causa di incertezza è evitata. Anche nella RT Class 2 (sincronizzata) tutti i dispositivi devono essere sincronizzati ma la topologia delle rete può essere cambiata (flessibile); inoltre non è specificata nessuna pianificazione dei messaggi perché quest’ultimi vengono indirizzati attraverso gli indirizzi MAC come di consueto e, se un messaggio non può essere spedito, viene memorizzato momentaneamente in un buffer per essere inviato appena possibile (per esempio nella fase verde).

In RT Class 1 per poter raggiungere un risultato ottimale nello scambio dati e nella gestione dei pacchetti all’interno del ciclo RT, Profinet utilizza la gestione delle priorità in conformità a IEEE 802.1q e, se i pacchetti sono dotati di VLAN Tag (estensione all’Ethernet frame), trasportano anche un valore di priorità codificato in un numero da 0 (senza priorità) a 7 (massima priorità). I pacchetti che fluiscono tra i dispositivi sono gestiti dagli switch in base a questo concetto di priorità: frame con più alto grado di priorità hanno la precedenza durante la fase di accodamento negli switch, rispetto a messaggi a bassa priorità. L’uso della priorità basata su VLAN assicura che i pacchetti real-time siano preferiti durante l’instradamento ai pacchetti standard del traffico TCP/UDP.

## 2.6 Le classi dei device

Con Profinet IO, l'integrazione di dispositivi di campo decentralizzati è implementata direttamente su Ethernet. Per questo scopo, la metodologia di accesso Master-Slave di Profibus DP è stata convertita al modello Provider-Consumer. Profinet prevede quattro tipi di dispositivi [20], [19], [30]:

- **IO-Controller:** dispositivo controllore sul quale gira il programma di automazione; ha il controllo del processo distribuito di uno o più dispositivi di campo ed è responsabile della configurazione e della parametrizzazione dei dispositivi associati ad esso. Riceve dati e allarmi e li processa all'interno del programma utente. Esso rappresenta una stazione centrale intelligente, come un PLC.
- **IO-Device:** dispositivo di bus di campo remoto, che viene assegnato a un IO-Controller. È configurato da un IO-Controller o da un IO-Supervisor e trasmette ciclicamente i suoi dati di processo al controllore. Esso provvede anche ad informare un IO-Controller su condizioni di diagnostica o di allarme.
- **IO-Supervisor:** dispositivo PG/PC di programmazione con funzioni di configurazione e di diagnostica, che può scambiare dati con IO-Controller e IO-Device; ha accesso temporaneo ai dispositivi di campo durante il processo di comunicazione.
- **IO-Parameter-Server:** Dispositivi usati per lo scambio di dati di configurazione rilevanti per le applicazioni con IO-Device.

Il traffico generato da IO-Supervisor e da IO-Parameter-Server avviene tipicamente in fase di offline, quindi non sono necessarie prestazioni real time. Dal punto di vista della comunicazione tutti i dispositivi su Ethernet sono trattati allo stesso modo. Tuttavia durante il processo di configurazione, i diversi componenti sono assegnati ad un controllore centralizzato (IO-Controller).

## 2.7 Il controllore di rete

Ogni dispositivo e ogni switch che fanno parte di una rete Ethernet industriale devono essere in grado di gestire il traffico real-time isocrono in modo da garantire le prestazioni richieste. Ecco perchè, per la creazione di IO-Device Profinet, alcune compagnie offrono pacchetti di sviluppo basati su particolari controllori Ethernet. Ad ogni controllore è associato un stack

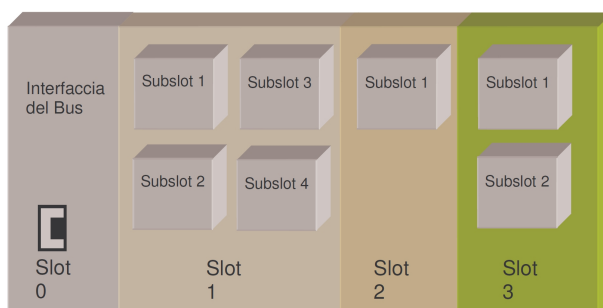
software che funge da interfaccia con il kernel Profinet IO (applicazione). Il controllore di rete offerto da Siemens si basa su un ASIC (Application-Specific Integrated Circuit) con elevate prestazioni denominato ERTEC (Enhanced Real-Time Ethernet Controller). Questi Ethernet Controller, con Real Time Switch integrato e microprocessore a 32 bit, sono stati sviluppati specialmente per l'impiego industriale in modo da implementare in uno spazio molto ridotto connessioni Profinet assai performanti. Facendo riferimento ai dispositivi prodotti da Siemens, esistono due versioni di ASIC: ERTEC 400 e ERTEC 200.

- **ERTEC 400** è un ASIC che permette di supportare comunicazioni ad elevate prestazioni: i sistemi di destinazione di ERTEC 400 sono apparecchiature di automazione come controllori, Motion Controller, sistemi PC-based e componenti di rete. È un dispositivo ottimizzato per la realizzazione sia di un device-IO, sia di un controllore-IO. ERTEC 400 è un Ethernet-Controller con Switch a 4 porte integrato, microprocessore (ARM 946) a 32 bit e interfaccia PCI. Lo Switch a 4 porte integrato consente di realizzare topologie flessibili in strutture a stella, ad albero o lineare senza ulteriori componenti di rete esterni.
- **ERTEC 200** è un ASIC ottimizzato per l'implementazione di device-IO. Rappresenta la soluzione ottimale per apparecchiature da campo. Si tratta di un Ethernet Controller con Switch a 2 porte integrato e microprocessore a 32 bit. Per l'accoppiamento dell'Ethernet Controller alla fisica della rete di comunicazione sono necessari i PHY, connettori che permettono il passaggio dal mondo interno al mondo esterno. Tali apparati sono già integrati all'interno di ERTEC 200 e supportano le funzioni di Auto-negoziazione e di Autocrossover. Grazie al processore ARM 946 integrato, ERTEC 200 può essere impiegato come implementazione "System-on-Chip" per apparecchiature semplici.

Dato che la maggior parte dei dispositivi di campo Profinet necessita del collegamento alla rete tramite switch, l'ASIC Ertec offre costo vantaggioso rispetto ad altri controllori Ethernet, dato che dispone già di 4 porte full-duplex integrate. Non è necessaria quindi alcuna connessione esterna con un ulteriore switch. La famiglia degli ASIC Ertec supporta tutte le classi di comunicazione Profinet IO e le funzionalità di sincronizzazione secondo lo standard IEEE 1588.

## 2.8 Modello dei dispositivi

Ad ogni dispositivo deve essere assegnato un numero di identificazione, *Device Ident Number* (ID), univoco. Questo numero a 32 bit è diviso in due parti: i primi 16-bit rappresentano il numero identificativo del produttore (assegnato dal PNO), i rimanenti 16-bit il numero identificativo del dispositivo (assegnato dal produttore). È stato specificato un modello uniforme per i dispositivi IO-Device Profinet, che permette una configurazione dei singoli moduli del dispositivo. Questo modello è simile alle caratteristiche del modello Profibus DP. Un dispositivo di campo può essere modulare, cioè includere slot fisici per l'inserzione dei moduli, oppure non espandibile e allora gli slot possono essere considerati virtuali. In ogni caso questo tipo di modellizzazione permette di indirizzare i canali di I/O presenti nel sistema in modo univoco. I moduli (slot) configurati contengono uno o più subslot che formano la reale interfaccia verso il processo (input/output) per lo scambio dati. Il contenuto dei dati di un subslot è sempre accompagnato da informazioni di stato dai quali si ricava la validità stessa dei dati. Infine l'*index* specifica i dati dentro uno slot o subslot che possono essere scritti o letti aciclicamente attraverso appositi servizi di scrittura/lettura.



**Figura 2.3:** *Modello IO-Device*

Tale modularità permette che eventuali moduli I/O esistenti su Profibus DP possano essere incorporati in Profinet senza restrizioni e modifiche. Un dispositivo Profinet è integrato nel tool di configurazione allo stesso modo di un dispositivo Profibus DP tramite una descrizione del dispositivo. Le caratteristiche del dispositivo sono descritte in un file GSD (General Station Description), che contiene tutte le informazioni necessarie in formato XML [19], [20].

## 2.9 Stack di comunicazione

In Profinet i dispositivi con funzionalità altamente tempo-critiche possono, quando stabiliscono il collegamento, negoziare protocolli di comunicazione con capacità real-time che riducono il tempo di elaborazione dello stack. In questo modo, Profinet soddisfa le elevate richieste real-time disposte sul sistema di comunicazione.

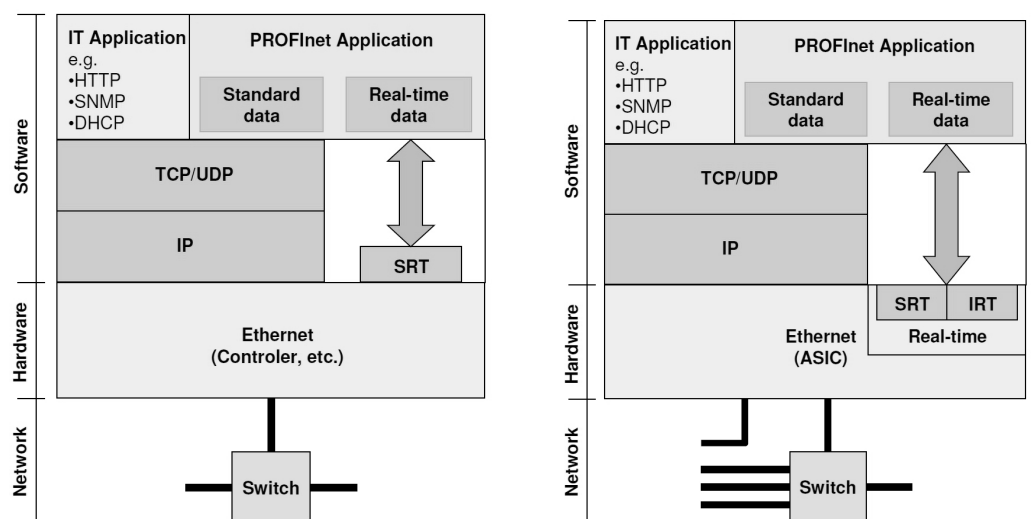


Figura 2.4: Stack di comunicazione RT e IRT

L'eliminazione di alcuni livelli di protocollo riduce la lunghezza di messaggio, il tempo richiesto per la preparazione dei dati da trasmettere e i tempi di procedura dell'applicazione. Un'ulteriore beneficio si ha considerando che la potenza del processore necessario nel dispositivo per la comunicazione è notevolmente ridotto. Nel caso di Profinet RT, avviene un "incapsulamento diretto al livello 2" (si veda il capitolo 1.3.3) quindi le caratteristiche real-time sono implementate in software e non c'è bisogno di hardware dedicato. Profinet IRT, invece, fa parte della categoria "Ethernet modificata" (si veda il capitolo 1.3.3) dove è richiesta una speciale interfaccia hardware (ASIC) per poter fornire una comunicazione deterministica e isocrona di alto livello.

## 2.10 Il modello Produttore/Consumatore

Profinet IO utilizza il modello di comunicazione detto Produttore / Consumatore (Producer / Consumer) e non Master/Slave. I sistemi di tipo Pro-

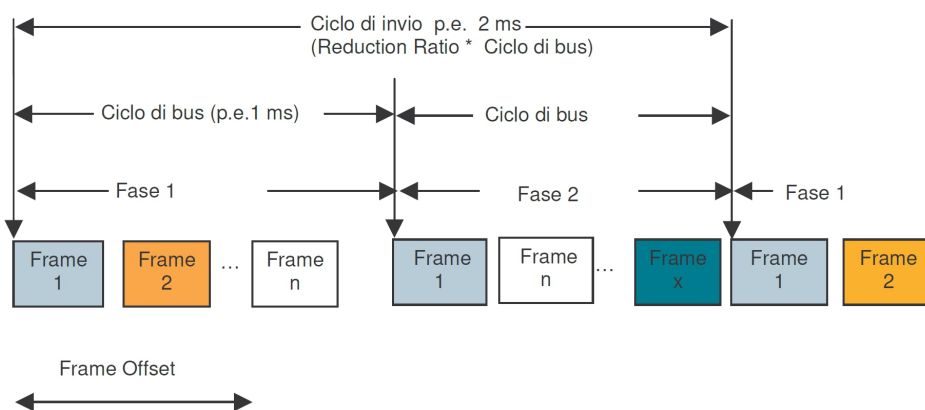
duzione / Consumatore hanno assunto nel corso del tempo una notevole importanza grazie anche allo sviluppo di applicazioni distribuite. Se nel bus uno solo dei nodi svolge la funzione di gestore della rete (controlla e gestisce l'accesso al bus) si parla di protocollo master/slave in cui il predetto nodo funge da master mentre gli altri nodi sono detti slave e sono semplici utilizzatori del canale di comunicazione (possono ricevere o trasmettere informazione ma in funzione del controllo del bus fatto dal master). Una rete master/slave in cui ci sono più dispositivi che fungono da master si dice multi-master. In tal caso è necessario un sistema di arbitraggio per risolvere i conflitti che nascono quando più master richiedono il controllo del canale di comunicazione. Tipicamente in queste reti il nodo che prende il controllo del bus (master), quando inizia una comunicazione, specifica tramite un indirizzo a quale nodo della rete è destinata l'informazione. Una rete in cui non ci sono dispositivi che si dedicano al controllo degli accessi al bus è detta di tipo Producer-Consumer. Qualunque nodo può acquisire momentaneamente il controllo del bus ed iniziare una trasmissione (interfaccia Producer), gli altri nodi in questa fase si attiveranno in ricezione (interfaccia Consumer). Il nodo della rete che trasmette non specifica a quale nodo della rete è destinata l'informazione (ovvero non viene specificato nessun indirizzo) ma specifica da quale interfaccia vengono prodotti i dati. In questo tipo di protocollo ogni stazione deve essere provvista della lista di variabili che può produrre. Durante la normale trasmissione dati, ad ogni stazione può essere richiesto di produrre una variabile o può richiedere essa stessa la produzione di una variabile.

## 2.11 Scambio di dati ciclico

Lo scambio dati ciclico è implementato attraverso la relazione di comunicazione chiamata IO CR (*IO Communication Relationship*). I dati di I/O sono trasmessi a cicli di tempo prefissati la cui frequenza di aggiornamento può essere diversificata per ciascun dispositivo. Inoltre l'intervallo di invio può essere diverso dall'intervallo di ricezione. Per la descrizione di uno scambio di dati ciclico Profinet IO definisce i seguenti termini [17], illustrati anche in figura 2.5 :

- **Ciclo di bus:** La base dei tempi del clock è  $31.25 \mu s$ . Un valore moltiplicativo tipico è 32 per avere un ciclo di bus di 1 ms;
- **Reduction Ratio:** Il rapporto di riduzione indica ogni quanti cicli di bus viene spedito un frame. Il ciclo di invio del frame è calcolato moltiplicando il rapporto di riduzione per la durata del ciclo di bus;

- **Frame Offset:** Il Frame Offset indica l'offset relativo in incrementi di 250 ns dall'inizio del ciclo di bus. Specifica quando un determinato frame verrà spedito all'interno del ciclo di bus;
- **Fase:** La fase indica in quale ciclo di bus il frame corrispondente verrà spedito.



**Figura 2.5:** Scambio di dati ciclico

Non tutti i dati devono essere trasmessi con la stessa frequenza, se ciò avvenisse, la stazione più lenta determinerebbe la velocità di spedizione dei dati. La soluzione a questo problema è fornita dal 'Reduction Ratio' che deve essere considerato come un fattore moltiplicatore del ciclo di invio o di ricezione e determina in quale ciclo di quelli successivi verranno trasmessi i dati. Il rapporto di riduzione è specificato, durante la configurazione, per ogni relazione di comunicazione I/O CR.

## 2.12 I canali di comunicazione

Le varie informazioni possono essere trasferite tra gli IO-Controller e gli IO-Device attraverso i seguenti canali:

- Dati ciclici di IO, trasferiti sul canale real-time;
- Allarmi, trasferiti sul canale real-time;
- Parametrizzazione, configurazione, lettura delle informazioni diagnostiche. Questi dati sono trasferiti attraverso canali standard sulla base di UDP/IP.



Lo standard Profinet IO specifica l'implementazione di routine automatiche attraverso la loro suddivisione in *application process* eseguite da differenti stazioni distribuite sulla rete. Le applicazioni di processo intercomunicano attraverso lo scambio di *application objects*, strutture formali che contengono informazioni riguardo i processi reali della propria stazione. Entrambi necessitano dei servizi forniti da ASE (*Application Service Element*), il concetto base di Profinet a livello applicazione. ASE provvede a fornire una serie di servizi per realizzare la comunicazione tra *application processes* distribuiti e lo scambio di *application objects*. In una tipica '*IO configuration*', alla partenza, vengono stabilite le istanze di relazione (*IO-AR Application Relationship*) tra gli IO-Controller e gli IO-Device attraverso il canale UDP/IP. L'istanza di relazione IO-AR contiene diverse relazioni di comunicazione (*CR Communication Relationships*) attraverso le quali sono trasferite la configurazione, i dati di processo e gli allarmi. Il dispositivo IO-Controller trasferisce i dati di parametrizzazione e di configurazione del dispositivo IO-Device assegnato sulla '*Record data CR*'. La trasmissione ciclica dei dati di ingresso e di uscita è implementata sulla '*IO CR*' e il monitoraggio delle relazioni di comunicazione avviene attraverso la stessa verifica dei messaggi ciclici ricevuti. Per esempio, se un frame atteso in ingresso non si manifesta entro 3 cicli, l'IO Controller determina l'errore o il guasto nel rispettivo IO-Device. Gli eventi aciclici sono trasmessi sulla '*Alarm CR*' verso l'IO-Controller. In Profinet esistono diversi tipi di allarme: unplug, plug in, diagnostici, di stato e allarmi di aggiornamento, inoltre sono comunque possibili allarmi specifici inseriti dal produttore del dispositivo. È possibile assegnare una priorità alta o bassa agli allarmi.

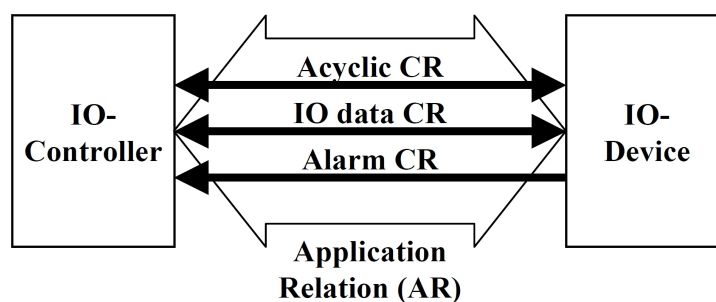


Figura 2.6: I canali di comunicazione

Diversi ASE sono stati definiti in Profinet IO, i più importanti sono *Application Relationship ASE*, *IO Data ASE* e *Alarm ASE* [26].

### 2.12.1 Application relationship ASE

Profinet IO specifica che il trasferimento dati tra tutti i dispositivi deve aver luogo esclusivamente via *Application Relationships* (ARs). Un AR include una o più *Communication Relationships* (CRs) stabilite tra due o più *Communication Relationships End Points* (CREPs), che possono consistere in buffer o code. L'AR ASE definisce tre tipi di AR: l'*IO AR*, il *Supervisor AR* e l'*Implicit AR*. In particolare, l'IO AR viene utilizzato sia da IO Data ASE sia da Alarm ASE per il trasferimento di dati ciclici e aciclici. Nello specifico, lo standard richiede che almeno una connessione unidirezionale CR venga fornita da IO AR per lo scambio di dati di input RT ciclici tra un IO device e un IO controller. Tale CR, che viene mostrato in figura 2.7, come CREPs fa uso di buffer ed è basato su una tecnica Producer/Consumer per lo scambio effettivo dei dati. In alternativa, il suddetto CR può essere sostituito con un altro che realizza la connessione tra un IO device e uno o più IO controller. Anche in questo caso, lo scambio dati avviene in accordo con una tecnica Producer/Consumer. Un CR analogo deve essere fornito dall'IO AR per la trasmissione di dati di output RT ciclici da un IO controller a un IO device. Viceversa, il CR per la trasmissione di allarmi (e degli acknowledgements più rilevanti) tra un IO device e un IO controller, anch'esso mostrato in figura 2.7, è bidirezionale e fa uso di code come CREPs. Lo scambio dati, in questo CR, avviene seguendo un modello Client/Server.

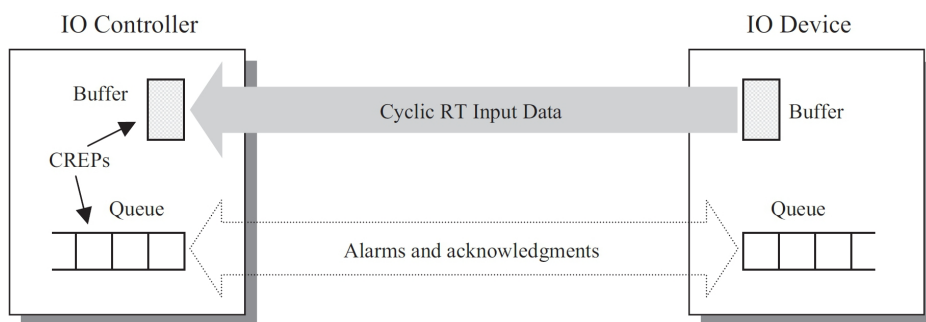


Figura 2.7: Esempio di CR dell'IO Application Relationship

### 2.12.2 IO Data ASE

I dati IO ASE permettono di gestire uno specifico *application object*, cioè i dati IO object, i quali possono essere definiti in modo univoco per ciascun *application process*. Tale oggetto contiene i dati di input/output real time

che devono essere scambiati ciclicamente. In altre parole, i dati IO ASE sono responsabili del trattamento del traffico ciclico RT. La figura 2.8 mostra i servizi più significativi disponibili nei dati IO ASE assieme ad un esempio del loro utilizzo relativo allo scambio dati tra un IO controller e un IO device. Lo standard determina che i dati IO ASE usino un modello Client/Server per trattare la comunicazione, nella quale gli IO controller operano come un client e gli IO device come un server. Tuttavia, il trasferimento dati avviene tramite IO AR e quindi seguendo il modello Producer/Consumer. In particolare, come si può notare in figura 2.8, l'IO controller aggiorna il suo buffer d'uscita con il *Set Output service*; i dati vengono resi disponibili all'IO device tramite gli output CR ciclici RT. L'IO device utilizza *Get Output service* per acquisire i dati di output. Tale servizio è ciclicamente invocato dal protocollo Profinet IO residente nell'IO device, a seconda della durata del ciclo mostrato in figura 2.10 (in alternativa, il servizio opzionale *New Output* può essere utilizzato per generare il *Get Output service*). Sia le richieste di *Set Output* che di *Get Output* possono essere sincrone o asincrone. Nel caso asincrono, se la richiesta di *Set Output* viene rilasciata più velocemente rispetto al *Get Output*, allora non tutti i valori scritti nel buffer di uscita potranno essere acquisiti dall'IO device. Al contrario, se la richiesta di *Set Output* è più lenta rispetto a quella del *Get Output*, allora lo stesso valore di output verrà letto più di una volta dall'IO Device. In caso di richiesta sincrona, i valori di output vengono trasmessi e acquisiti nello stesso momento in cui vengono aggiornati: tale comportamento permette di implementare la classe isocrona di Profinet IO. I dati di input vengono scambiati in modo simile: il servizio *Set Input*, rilasciato dall'IO device scrive nuovi dati nel buffer di input, i quali, a loro volta, vengono trasmessi tramite input CR RT ciclici. Successivamente l'IO controller acquisisce i dati di input ciclicamente invocando il *Get Input service* (o utilizzando il *New Input service* come trigger). Anche in questo caso i dati scambiati potrebbero essere sia asincroni che sincroni.

### 2.12.3 Allarmi Ase

Questo tipo di ASE permette di effettuare il trasferimento di oggetti d'allarme dall'IO device all'IO controller e viceversa. La ricezione di un oggetto allarme deve anche essere esplicitamente confermata. La struttura di un *alarm object* è completamente definita dallo standard Profinet IO e può anche includere specifici dati di fabbrica. L'allarme ASE rende utilizzabili due servizi diversi: *Alarm Notification* e *Alarm Ack*. La trasmissione degli *alarm objects* e dei loro acknowledgements avvengono negli slot RT aciclici tramite il

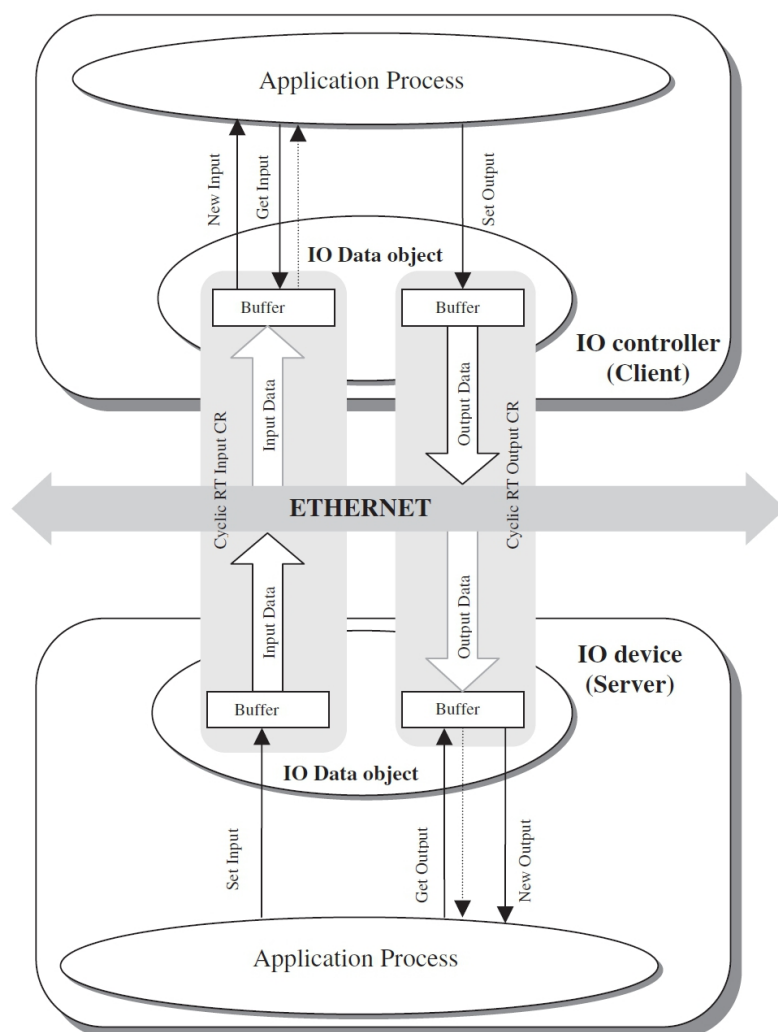


Figura 2.8: Servizi dell'IO Data ASE

CR obbligatorio stabilito dall'IO AR. Tale CR fa uso di code come CREPs, a differenza della trasmissione di dati IO RT ciclici, nei quali vengono usati i buffer. In questo modo, le notifiche successive degli allarmi e/o degli acknowledgments vengono accodate in sequenza e non sovrascrivono quelle precedenti prima che queste ultime vengano acquisite correttamente dagli *application processes*.

## 2.13 Ciclo RT

Il cosiddetto Soft RealTime Channel (SRT Channel), o più semplicemente canale RT, è un primo approccio al problema della gestione della comunicazione realtime ed è in grado di soddisfare processi nei quali:

- risultano accettabili ritardi contenuti entro determinati limiti (circa 10 ms);
- le condizioni imposte sui limiti accettabili per i ritardi debbano essere mantenuti non rigidamente ma attorno a un valore medio.

Si tratta di una soluzione completamente software in grado di soddisfare, per le caratteristiche che presenta, le applicazioni tipiche dell'area della factory automation ove i tempi di risposta del sistema si devono attestare attorno alla decina di millisecondi. In questo caso in Profinet c'è soltanto uno stack di comunicazione minimizzato e ottimizzato che sostituisce i livelli TCP/IP e UDP/IP (vedi sezione 2.10).

### 2.13.1 Struttura del pacchetto RT

In figura 2.9 viene mostrato come è strutturato un pacchetto Profinet RT.

56 Bits	8 Bits	6 Byte	6 Byte	2 byte	2 byte	2 byte	2 byte	40..1440 bytes	2 byte	1 byte	1 byte	4 byte
Preamble	SYNCH	Dest Addr	Src Addr	Tag	Tag Control	Type 8892H	Frame ID	User data	Cycle Counter	Data Status	Transfer Status	FCS
				VLAN Tag					APDU-Status			

**Figura 2.9:** *Struttura del frame Profinet RT*

dove:

- *Preamble* è la sequenza di bit usata per la sincronizzare il ricevitore;
- *SFD*: Start Frame Delimiter è la sequenza di bit che specifica l'inizio del frame;
- *Destination address* è l'indirizzo del nodo destinazione;
- *Source address* è l'indirizzo del nodo sorgente;

- *Type*: 0x8100 esprime il Virtual LAN Header;
- *VLAN* contiene le specifiche della VLAN;
- *Ethertype* per Profinet: 0x8892;
- *Frame ID*: indica il tipo di frame (vedi tabella seguente);
- *Data*: dati RT;
- *Cycle Counter*: contatore di ciclo; in questo campo un bit rappresenta un incremento di tempo pari a 31,25  $\mu s$ ;
- *Data Status* indica lo stato dei dati;
- *Transfer Status* indica se la trasmissione è avvenuta regolarmente;
- *FCS*: Frame Control Sequence è la sequenza che controlla la correttezza del frame.

Frame ID		Significato
Da	a	
0000	00FF	Time Synchronization
0100	7FFF	RT classe 3 Frame ciclico (IRT)
8000	BFFF	RT classe 2 Frame ciclico (RT)
C000	FBFF	RT classe 1 Frame ciclico (RT)
FC00	FCFF	Trasmissione aciclica 'high'
FD00	FDFE	riservato
FE00	FEFC	Trasmissione aciclica 'low'
FEFD	FEFF	DGP: Discovery Configuration Protocol
FF00	FFFF	riservato

**Tabella 2.1:** Valori possibili del campo frame ID

Gli EtherType sono assegnati da IEEE e sono quindi univoci al fine di poter distinguere i vari protocolli. Il FrameID è utilizzato per indirizzare un canale di comunicazione specifico tra due dispositivi. EtherType e FrameID fanno parte dell'header del protocollo RT e favoriscono una classificazione veloce dei pacchetti. Questo permette al dispositivo ricevente di classificare un pacchetto ricevuto velocemente per mezzo di un efficiente algoritmo. I 6-byte dell'indirizzo MAC selezionano il dispositivo. Il dispositivo real-time deve valutare solo l'EtherType e il FrameID per trovare il canale di comunicazione corretto. Nel pacchetto viene anche inserito un contatore di ciclo

Cycle Counter da parte del Provider. Il Provider incrementa il contatore di ciclo ad ogni ciclo e lo inserisce nel pacchetto. Ogni bit di incremento corrisponde ad un intervallo di tempo di  $31,25 \mu s$ . Il Consumer, quando riceve il pacchetto, controlla il valore contenuto nel campo Cycle Counter. Attraverso questo valore può controllare se i dati di processo trasmessi sono quelli attuali. I bit del campo Data Status hanno il seguente significato:

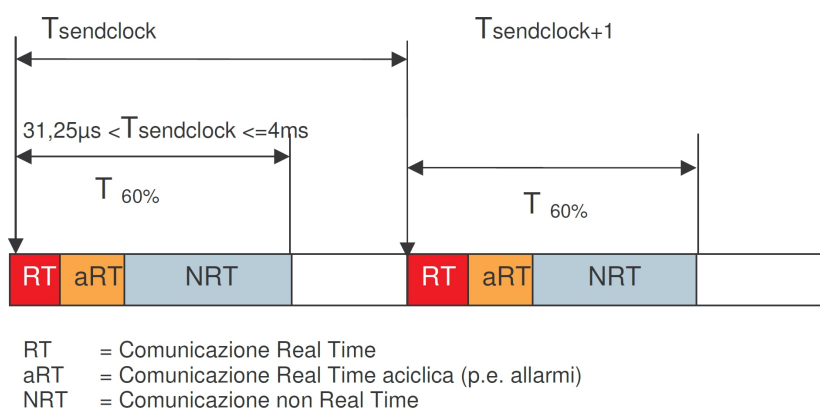
- Bit 0 (Stato) 1 = Primario; determina il canale dominante nel caso di canali multipli (ridondanza);
- Bit 2 (Dati Validi) 1 = Dati validi; il valore 0 è permesso solo durante la fase di avviamento;
- Bit 4 (Stato del Processo) 1 = Il processo che ha generato i dati sta girando;
- Bit 5 (Indicatore di Problema) 1 = Nessun problema; se questo bit è stato settato a 0, l'allarme diagnostico deve essere stato segnalato e le informazioni diagnostiche possono essere recuperate dai corrispondenti record.

Se la trasmissione è avvenuta regolarmente, il byte Transfer Status deve essere uguale a 0. Per potere raggiungere un risultato ottimale, i pacchetti in Profinet possono avvantaggiarsi delle priorità in conformità con IEEE 802.1q; se i pacchetti sono dotati di un VLAN Tag trasportano anche un valore di priorità codificato in un numero da 0 (senza priorità) a 7 (massima priorità). I pacchetti che fluiscono fra i dispositivi sono gestiti dagli switch in base a questo concetto di priorità. Un pacchetto a più alta probabilità ha la precedenza durante la fase di accodamento all'interno dello switch sul pacchetto con priorità più bassa. L'uso della priorità basata su VLAN assicura che i pacchetti real-time siano preferiti, durante l'instradamento tramite switch, ai pacchetti standard del normale traffico IP. Ai pacchetti Profinet real-time viene assegnato il valore di priorità 6 [19].

### 2.13.2 Ciclo di comunicazione RT

Lo scopo della suddivisione del ciclo di comunicazione RT è quello di dividere la banda all'interno del sistema. Per questo motivo non vengono definiti dei parametri di tolleranza per il jitter, parametri invece definiti nella comunicazione IRT. I dispositivi Profinet utilizzano una velocità di trasmissione di 100 Mbit/s in full-duplex e sono collegati tramite switch. In teoria, ciascun

dispositivo potrebbe trasmettere i pacchetti usando tutta la banda massima, ma questa caratteristica porterebbe in breve tempo ad un cattivo funzionamento del sistema; infatti non sarebbe possibile che, verso un singolo nodo, siano diretti più flussi contemporanei tutti alla banda massima (overload) e neppure la trasmissione di pacchetti non RT. Gli switch interessati devono gestire le situazioni di sovraccarico attraverso delle code, oppure in ultima ipotesi gettare via i pacchetti. Per questo motivo Profinet prevede un meccanismo per gestire e limitare l'invio dei pacchetti. Per aumentare l'uso della



**Figura 2.10:** *Ciclo di comunicazione RT*

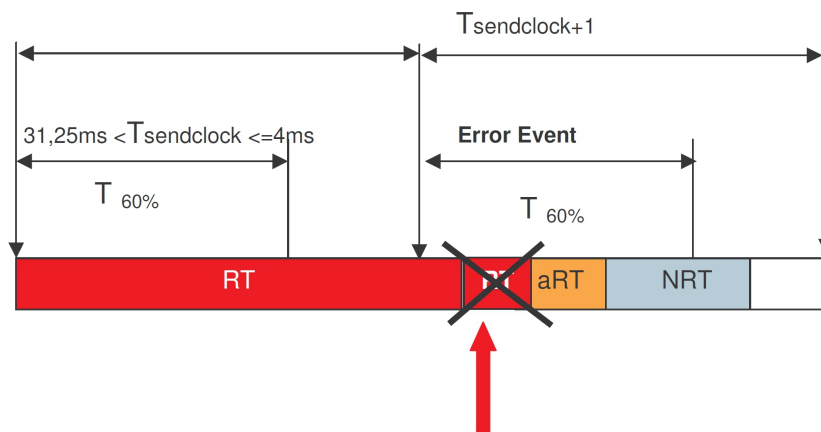
banda e per prevenire sovraccarichi temporanei dovuti a connessioni multiple (ad esempio un Profinet IO-Controller può essere collegato allo stesso tempo con diversi IO-Device) è possibile sincronizzare il ciclo di invio del provider con quello del consumer [17]. Questa sincronizzazione è basata sul valore del contatore di ciclo 'Cycle Counter' con una accuratezza stimata di un ciclo di clock. Come mostrato in figura 2.10 il ciclo di comunicazione è suddiviso in parti differenti. Ogni ciclo è definito da  $T_{sendclock}$  che è compreso tra  $31,25 \mu s$  e  $4 ms$ . All'inizio del ciclo vengono spediti i pacchetti RT relativi al dispositivo (RT). Il tempo destinato ai pacchetti RT non deve superare un certo ammontare (p.e. 50 %) per ciascun ciclo. Successivamente possono essere spediti pacchetti real-time aciclici (aRT). Anche il tempo destinato per questi pacchetti non deve superare un certo ammontare (p.e. 10%). Infine, per il resto del tempo disponibile, verranno spediti i pacchetti non real-time (per esempio pacchetti UDP/TCP) [19]. Attraverso il rapporto di riduzione 'Reduction Ratio' e la fase si possono distribuire i pacchetti RT nel tempo. Il traffico di rete viene così controllato e limitato attraverso questi meccanismi che seguono le seguenti regole:

- il processo che si occupa dell'invio dei pacchetti non deve spedire alcun pacchetto RT se all'inizio del nuovo ciclo esiste una situazione di



sovraccarico per l'interfaccia locale;

- in media, non può essere usato più di un certo ammontare di banda per evitare il sovraccarico del sistema.



**Figura 2.11:** *Esempio di sovraccarico*

La figura 2.11 mostra l'esempio di una situazione di sovraccarico per l'interfaccia locale. Al tempo  $T_{sendclock+1}$  il provider riceve un errore di trasmissione dal MAC dovuto ad una temporanea situazione di sovraccarico. In questo caso il provider salta completamente la spedizione dei pacchetti RT per l'intero ciclo e riprova con il successivo [17], [19]; tuttavia, se possibile, gli altri pacchetti (aRT e NRT) possono essere spediti all'interno del ciclo.

### 2.13.3 Stabilità del ciclo

Per evitare che l'invio di frame NRT finisca dopo l'inizio di un nuovo ciclo (che ritarderebbe la trasmissione del seguente frame RT, causando un jitter nel tempo di ciclo), è stato introdotto un margine di sicurezza durante il quale non è permessa la trasmissione di nessun tipo di frame. Dallo standard si deduce che tutte le attività di un ciclo devono partire entro il 60% della durata del ciclo stesso. Quindi il margine di sicurezza rimanente è il 40% del ciclo. Inoltre lo standard suggerisce, come tecnica per evitare il sovraccarico del sistema, che ogni stazione trasmittente limiti la banda usata al 60% del suo massimo valore. Inoltre, poiché la lunghezza massima di un frame ethernet è di 1526 ottetti, per evitare la presenza di jitter nel tempo di ciclo, si deve assicurare che il margine di sicurezza sia maggiore di 1526

ottetti. Quindi, a 100 Mbit/s, si ha che il minimo margine di sicurezza risulta di 122  $\mu s$ . In particolare, sotto queste ipotesi (raccomandate dallo standard) il minor tempo di ciclo possibile è di  $T_c = 305 \mu s$ , tempi di ciclo inferiori potrebbero essere affetti da jitter. Tuttavia anche se la condizione  $T \geq T_C$  valesse per tutte le stazioni della rete, i periodi di aggiornamento delle variabili di ingresso/uscita possono essere affette da jitter causato dalla possibile mancanza di sincronismo tra le stazioni. Ecco perchè se vengono richieste prestazioni della rete maggiori del semplice Real-Time di classe uno, bisogna ricorrere al funzionamento isocrono [26].

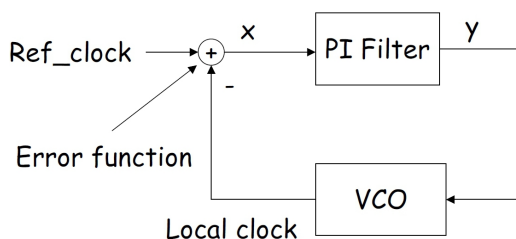
## 2.14 Ciclo IRT

Come comunicazione isocrona si prendono in considerazione le RT Class 2 e 3 definite da Profinet. In un sistema IRT il ciclo di bus deve essere diviso in una parte IRT e una parte non IRT durante la configurazione. Profinet IRT usa gli stessi protocolli visti precedentemente; in aggiunta ha un protocollo di comunicazione per la sincronizzazione e un gestore della comunicazione temporale. Lo scopo principale di IRT è la sincronizzazione della comunicazione attraverso un controllo preciso della comunicazione. Questo può essere implementato attraverso una programmazione della tempistica specifica per ogni singolo nodo inserito nella comunicazione. La programmazione della tempistica di comunicazione definisce esattamente quando un singolo pacchetto arriverà a destinazione e quando dovrà essere di nuovo instradato. Dovranno essere prese anche delle contromisure al fine di evitare che le stazioni senza comunicazioni IRT non interferiscano sul sistema [19].

### 2.14.1 Sincronizzazione e Bypass Clock

La sincronizzazione del clock è disaccoppiata dai meccanismi IRT e usa le sequenze di protocollo secondo la normativa IEEE 1588 (PTP - Precision Time Protocol). Tuttavia questo protocollo di sincronizzazione del tempo non è ottimale se nella rete sono presenti molti switch, perché una catena di switch può introdurre dei ritardi variabili influenzando le procedure di sincronizzazione del PTP, pensate per ritardi costanti. Infatti le derive di clock dei vari dispositivi sono principalmente causate da deviazioni di frequenze tra chi invia il clock e chi lo riceve, dovute soprattutto alla temperatura, all'invecchiamento dei componenti, agli stress meccanici e alle vibrazioni. Quindi

ogni clock locale deve avere un controllo a retroazione al suo interno che compensi tale deriva. Le specifiche del Control Loop (un esempio è riportato



**Figura 2.12:** Esempio di un Control Loop progettato con un PI

in figura 2.12) vanno al di là dello scopo di questa tesi, ma il modo con cui esso viene progettato determina la qualità della sincronizzazione temporale. Si capisce come in topologie lineari, dove sono presenti molti switch in cascata, e dove si fa uso della tecnica del Boundary Clock, il segnale di riferimento per la sincronizzazione dell'oscillatore interno è preso sempre dallo switch/bridge precedente; ciò genera una cascata di controlli a retroazione che porta a deviazioni dei clock distribuiti nella rete non accettabili o addirittura ad instabilità. Essendo la topologia lineare di fondamentale importanza a livello di campo, è necessario introdurre un altro elemento di sincronizzazione accanto a quelli già citati appartenenti allo già standardizzato IEEE 1588. Il nuovo elemento è il cosiddetto *Bypass Clock* (BpC)<sup>3</sup> ed è usato dal protocollo Profinet IO. Come già menzionato, il problema essenziale quando si ha a che fare con gli switch/bridge è il ritardo non costante. Per un'accurata sincronizzazione tra master e slave, il ritardo deve essere noto dal PTP Slave nella maniera più accurata possibile e, se fosse calcolato dagli switch stessi, potrebbe essere compensato.

Il Bypass Clock, infatti, compensa la latenza introdotta dagli switch manipolando gli opportuni telegrammi; aggiungendo il ritardo tra ricezione e invio nel telegramma. Un *sync message* fornirà un modo per considerare gli switch come componenti di rete con ritardo costante. La precisione di questo approccio dipende dal valore dei jitter (cambiamenti casuali dei segnali di clock) di ciascun switch/bridge e dall'effetto della loro messa in cascata, ma non da come viene progettato il Control Loop (si può notare la differenza dei due approcci in figura 2.13). Inoltre, essendo il jitter di natura statistica, può essere trattato facilmente da opportuni filtri [10]. Con questo approccio, dunque, non è più necessario alcun tipo di controllo a retroazione per l'in-

<sup>3</sup>Il Bypass Clock a volte viene anche chiamato *Transparent Clock* e il metodo di sincronizzazione di conseguenza prende il nome di Precision Transparent Clock Protocol (PTCP).

vio di sync message. La suddetta è una procedura speciale sviluppata come una espansione della normativa IEEE 1588 che permette un alto grado di precisione in un sistema anche in presenza di molti switch in cascata.

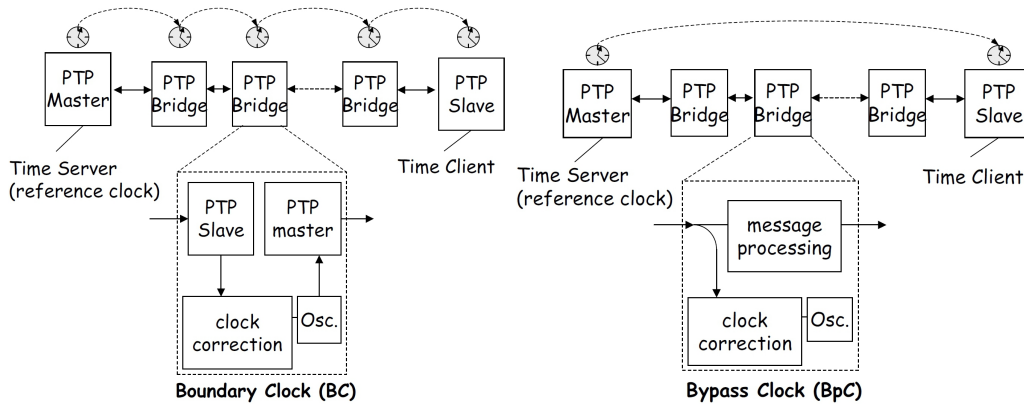


Figura 2.13: Boundary Clock e Bypass Clock

### 2.14.2 Struttura del pacchetto Profinet IRT

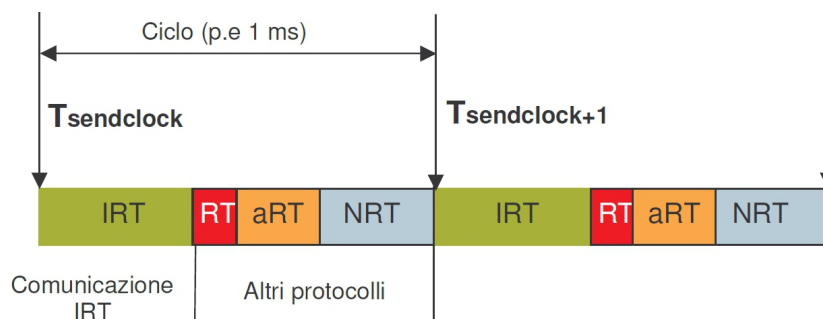
I pacchetti IRT sono soggetti ad una comunicazione guidata dal tempo. Una programmazione delle tempistiche nel dispositivo di campo IRT definisce il tempo esatto di trasferimento dei dati real-time. Gli altri pacchetti possono essere trasferiti solo dopo aver trasmesso i dati real-time. Il pacchetto IRT è specificato dalla sua posizione temporale, dal Frame ID e dall'EtherType 0x8892. Il campo VLAN per la priorità non è necessario e quindi non viene inserito nel pacchetto. In figura 2.14 è mostrata la struttura di un pacchetto IRT.

56 Bits	8 Bits	6 Byte	6 Byte	2 byte	2 byte	36..1490 bytes	4 byte
Preamble	SYNCH	Dest Addr	Src Addr	Ether type	Frame ID	RT.-User data	FCS

Figura 2.14: Struttura del frame Profinet IRT

### 2.14.3 Canale di comunicazione

Utilizzando una rete Ethernet standard dotata di switch, i differenti pacchetti che sono stati spediti sulla stessa porta vengono processati secondo la strategia ‘il primo arrivato, è il primo servito’. Per ottenere il determinismo nel trasferimento di pacchetti Ethernet si deve riservare esplicitamente per i pacchetti IRT una percentuale della banda disponibile, non solo nei dispositivi, ma anche negli switch. La banda disponibile è divisa in due sezioni, banda per la comunicazione IRT e banda per le altre comunicazioni. All’inizio del ciclo è inserita la comunicazione IRT. All’interno del tempo destinato all’IRT possono essere trasmessi solo pacchetti IRT. I pacchetti IRT sono identificati dal loro Frame ID. Nel frattempo gli altri pacchetti sono inseriti in un buffer fino all’inizio della parte non IRT. Il jitter di sincronizzazione di un ciclo IRT deve essere inferiore ad 1 microsecondo. Per ottenere un trasferimento di dati real-time deterministico all’interno di un ciclo IRT il tempo di invio di ogni pacchetto è calcolato esplicitamente ed è fisso per ogni pacchetto. Allo stesso modo vengono definiti i limiti del ciclo IRT. I limiti degli altri pacchetti (RT, aRT, NRT), per la parte del ciclo non IRT, dipendono dal numero di pacchetti da spedire e possono variare nel tempo. I limiti del ciclo IRT (inizio e fine) possono essere utilizzati come indicatori per l’inizio di una applicazione (p.e. motion control)



**Figura 2.15:** *Ciclo di comunicazione IRT*

### 2.14.4 Miglioramento in bus lineari

Per quanto riguarda le prestazioni, Profinet IO si colloca nella fascia più alta dei protocolli industriali basati su ethernet e, inizialmente, è stato ottimizzato per quelle applicazioni che usano una topologia di tipo mista (che è la topologia classica più usata): si ha una backbone e tanti rami che dipartono da questa dorsale per raggiungere i dispositivi di campo remoti. In questi casi, è stato dimostrato da diversi studi che Profinet IO è sempre il più veloce rispetto ad ogni altro tipo di protocollo RTE. Rimangono però quelle applicazioni dove tutti i dispositivi di campo sono pochi e sono tutti connessi in una linea (topologia lineare), cioè sono tutti in cascata uno dopo l'altro. In questo caso Profinet che utilizza un messaggio di input/output differente per ciascun dispositivo risultava leggermente penalizzato rispetto ad altri bus di campo. Con la versione 2.3 questo leggero gap dovuto alla non ottimizzazione della comunicazione è stata completamente risolta. Nella versione 2.3 è stata aggiunta questa nuova modalità di comunicazione che permette di ottimizzare anche le topologie di tipo lineare. Questa ottimizzazione è stata introdotta per sistemi essenzialmente dedicati al motion control dove si ha bisogno di tempi di ciclo ridottissimi e errori (jitter) sui tempi di ciclo inferiori al microsecondo. Viene introdotto questo tipo di approccio [27] (si veda la figura

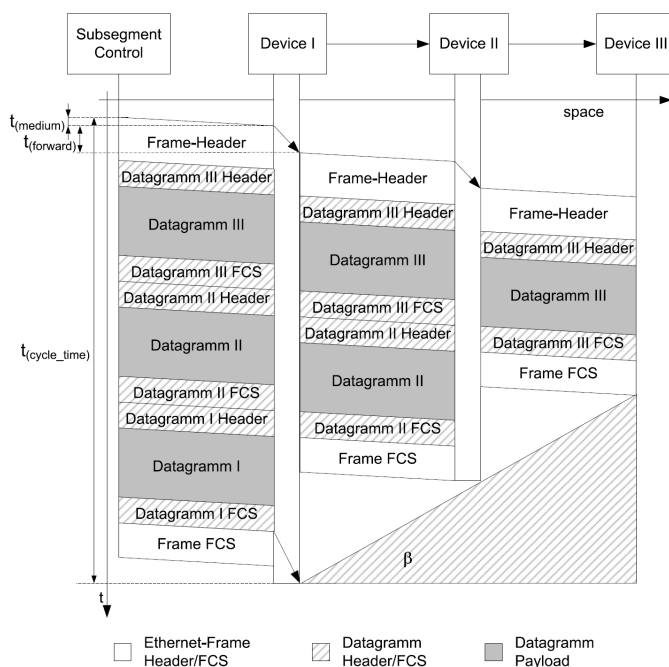
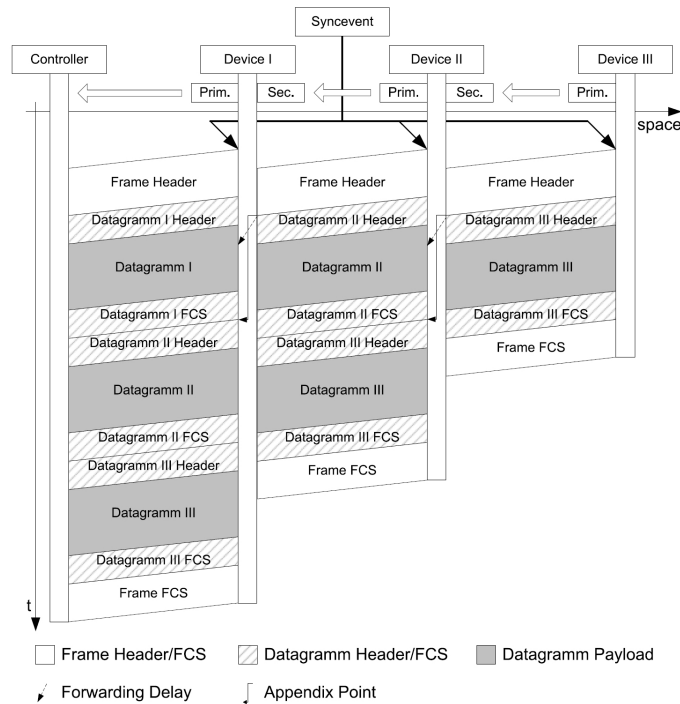


Figura 2.16: *Downstream communication*

2.16): il controllore spedisce un singolo pacchetto che contiene i dati di tutti

i dispositivi che appartengono alla linea. A questo punto il device 1 riceve questo pacchetto e toglie i dati che lo riguardano quindi a valle del primo device il pacchetto che verrà ritrasmesso sarà ridotto; dunque man mano che il pacchetto procede all'interno della rete diventa sempre più corto; questo fa sì che anche l'occupazione della banda continui a diminuire man mano ci si allontana dal controller e all'ultimo device arriveranno solo dati pertinenti ad esso. La stessa filosofia si può applicare anche ai dati che vanno dai dispositivi ai controller [27] (si veda figura 2.17). La trasmissione può essere ancor più ottimizzata se i dispositivi che appartengono a questa topologia sono tra loro sincronizzati, così facendo è possibile scatenare una trasmissione simultanea in tutti i dispositivi (device 1, 2, 3). In questa modalità, il device 2 quando ha finito di trasmettere i suoi dati ha in memoria una parte di dati del device 3 e quindi può iniziare a trasmettere anche quelli accodati ai suoi. Il risultato è che pian piano si va a costruire un frame unico più grande che il device uno manderà al controller contenente i dati di tutta la topologia lineare. Grazie a questa modalità di trasmissione dei dati, aggiunta con la versione 2.3 di Profinet, è possibile ottenere prestazioni che sono in assoluto le più alte rispetto a tutti gli altri protocolli basati su ethernet anche in topologie lineari.



**Figura 2.17:** *Upstream communication*





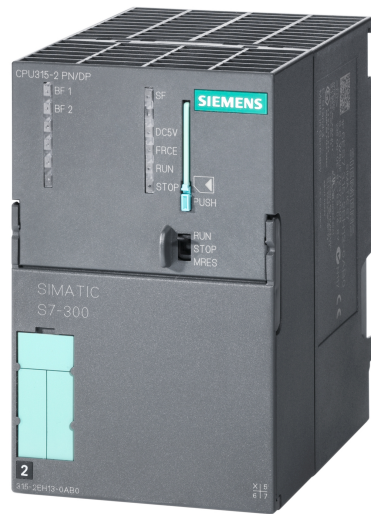
# Capitolo 3

## Strumentazione

### 3.1 IO-Controller

Il dispositivo utilizzato come IO-Controller è la CPU 315-2 PN/DP della famiglia SIMATIC S7-300 [34]. Si tratta di un controllore di media potenza di calcolo adatto a comandare l'automazione di singole macchine o parti di linee di produzione. Il processore installato permette di effettuare operazioni binarie con un tempo di esecuzione attorno ai  $100ns$  mentre impiega circa  $3\mu s$  per operazioni in virgola mobile. Il programma utente viene memorizzato nella memoria RAM interna che ammonta a 128 KB o eventualmente su un supporto memory card MMC (di capacità massima pari a 8 MB) che può essere usata anche per l'archiviazione di dati. Per quanto riguarda la comunicazione, la CPU 315-2 PN/DP integra un'interfaccia combinata MPI/DP per la diagnostica e per il collegamento su rete PROFIBUS. Tramite essa è possibile stabilire fino a 16 collegamenti simultanei con altre stazioni S7-300/400, dispositivi di programmazione Siemens (PG) o PC e scambiare dati con al massimo 32 stazioni collegate al bus di campo. Un collegamento è riservato al PC o PG di configurazione tramite il quale viene programmato. La seconda interfaccia presente sul frontale della CPU ne permette il collegamento su rete Profinet. Tramite questa vengono supportati i seguenti modi di comunicazione:

- Comunicazione S7 riservata per lo scambio di dati tra controllori SIMATIC.



**Figura 3.1:** CPU 315-2 PN/DP

- Real-Time classe 1 con la quale è possibile scambiare dati ciclici con IO-Device.
- Comunicazione PG/PC per la programmazione , lo start-up e la diagnostica tramite programma di configurazione STEP 7 (di cui si parlerà più avanti).
- Comunicazione PG/PC per l'interfacciamento con dispositivi HMI (Human Machine Interface) e sistemi SCADA (Supervisory Control And Data Acquisition).
- Comunicazione Open TCP/IP tramite Profinet per SIMATIC NET OPC server.

Come si nota le classi Real-Time 2 e 3 non sono supportate dalla CPU [37], quindi i test si soffermeranno solamente sulla comunicazione Real-Time classe 1. La CPU integra anche un'interfaccia web server accessibile da un qualsiasi PC che disponga di un browser di rete. La suddivisione in più pagine HTML permette di avere accesso a dati di diagnostica in formato di testo e animazione. Le informazioni accessibili sono:

- Informazioni identificative
- Contenuto del buffer di diagnostica
- Messaggi (senza possibilità di conferma)

- Informazioni su Profinet
- Stato delle variabili
- Tabelle delle variabili
- Pagina iniziale con informazioni generali sulla CPU

L'attivazione del web server avviene in fase di configurazione ed è possibile la selezione di diverse lingue. Per quanto riguarda la comunicazione Profinet IO, tale CPU riesce a supportare intervalli di trasmissione <sup>1</sup> minimi di 1 ms (parametro non impostabile da Step 7); tale valore dipende dalle prestazioni della CPU, dal numero di IO/device e da quanti dati vengono scambiati nella rete.

## 3.2 Switch

Siemens propone una serie di switch ad alte prestazioni adattati per l'utilizzo in ambienti industriali. Il dispositivo utilizzato nel lavoro di tesi è lo SCALANCE X204IRT [36]. Presenta 4 porte con connettore RJ45 con funzionalità autonegotiation e autocrossover. Con Autonegotiation si intende il riconoscimento automatico della funzionalità dell'interfaccia del lato opposto. Con il procedimento Autonegotiation i repeater o i terminali possono determinare la funzionalità della quale dispone l'interfaccia del lato opposto in modo da rendere possibile una configurazione automatica di apparecchi diversi. Permette quindi a due componenti collegati da un segmento, di scambiare tra loro parametri e di impostarsi con l'aiuto di quest'ultimi sui valori marginali della comunicazione supportata. La funzionalità di autocrossover offre i vantaggi di un cablaggio continuo senza che sia necessario un cavo Ethernet esterno incrociato. In questo modo vengono evitati funzionamenti errati nel caso di scambio dei cavi di trasmissione e ricezione. L'installazione per l'utente viene quindi notevolmente semplificata. L'X204IRT assicura massime prestazioni e compatibilità con i più comuni protocolli basati su Ethernet e supporta tutte e tre le classi Real Time previste dallo standard Profinet, combinando il meccanismo di switching 'Cut Through' con quello 'Store & Forward'. Come per la CPU 315-2 PN/DP anche lo switch dispone

---

<sup>1</sup>Periodo di tempo tra due intervalli consecutivi per la comunicazione IRT o RT. L'intervallo di trasmissione è la frequenza di invio minima possibile per lo scambio di dati.



**Figura 3.2:** *Scalance X204IRT*

di un Web Based Management accessibile tramite browser o utilizzando servizi TELNET, FTP o SNMP. Oltre a numerose funzioni di diagnostica e di stato, è possibile:

- impostare le porte per la rindondanza nel caso di collegamento ad anello;
- gestire la funzionalità di e-mail agent per l'invio di e-mail contenenti dati sulla diagnostica;
- configurare e abilitare il Port Mirroring (il traffico di dati viene rispecchiato dalla Mirror Port alla Monitor Port).

### 3.3 IO-Device 1

Si tratta, nel nostro caso, di un'unità di periferia decentrata per la gestione di ingressi e uscite digitali. La particolarità che caratterizza i prodotti Siemens è la loro modularità che permette differenti configurazioni a seconda del tipo di utilizzo. La versione utilizzata all'interno della rete è un ET 200M che prevede un modulo SM323 8DI/8DO a 24V (8 ingressi digitali e 8 uscite digitali) per il collegamento di interruttori normali e di prossimità a due fili,

elettrovalvole, segnaletica luminosa, piccoli motori e qualsiasi altro dispositivo che fornisca o necessiti di un segnale digitale a 24V e una corrente di picco fino a 1A (nominale 500mA) [35]. Il costruttore specifica i tempi di commutazione per le uscite che variano da un minimo di  $100\mu s$  ad un massimo di  $500\mu s$ , mentre i tempi di ritardo sull'ingresso variano da un minimo di  $1,2ms$  a un massimo di  $4,8ms$ . I dati sono validi sia per la commutazione "1" "0" che per quella "0" "1". Durante il lavoro di tesi non si disponeva del modulo SM 323 vero e proprio ma di un'unità simulatore SM 374. Quest'ultima, attraverso una funzione impostabile con un giravite, è in grado di simulare 16 ingressi o 16 uscite o 8 ingressi e 8 uscite (come nel caso in esame). L'unità SM 374 non è contenuta nel catalogo delle unità di Step 7, cioè il numero di ordinazione non viene riconosciuto; è quindi necessario 'simulare' il modulo per eseguire la configurazione della funzione desiderata dell'unità simulatore. Per garantire uniformità di utilizzo, la connessione alla rete industriale del-



**Figura 3.3:** ET 200 M e SM 374

l'ET 200M prevede l'utilizzo di un'interfaccia di rete modulare collegata ad esso tramite il connettore posteriore *backplane* (presente in tutti i dispositivi della famiglia Simatic) attraverso il quale vengono scambiati tutti i dati di IO e di configurazione. Il modulo utilizzato è l'IM 153-4 PN con il quale è possibile gestire fino a 12 unità di IO. Integra uno switch a due porte che permette di inserirlo all'interno di una struttura di rete lineare o in un anello di ridondanza. Supporta i servizi Ethernet quali:

- ping
- arp

- Diagnostica di rete (SNMP): LLDP-MIB/MIB-2
- Resettaggio dei parametri SNMP sulle impostazioni di fabbrica
- Dati I&M

Nonchè funzioni di allarme:

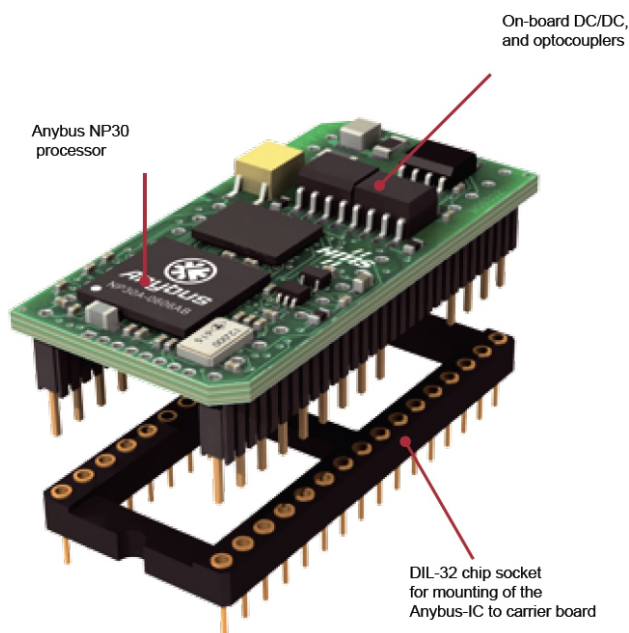
- Allarmi di diagnostica
- Interrupt di processo
- Allarmi di estrazione/inserimento
- Allarmi di manutenzione

Permette la selezione delle porte da attivare e la loro diagnostica. Supporta la comunicazione in tempo reale RT classe 1 e classe 2 [35].

## 3.4 IO-Device 2

Il secondo device inserito nella rete è il chip Anybus-IC di HMS (se dispone di interfaccia Profinet IO, come in questo caso, viene chiamato anche ABIC-PRT). Anybus-IC è una piccola interfaccia di comunicazione single-chip per Profinet IO (ASIC) basata sul nuovo processore NP30 di HMS e certificata dal consorzio Profibus a livello internazionale; fornisce connettività avanzata in una soluzione a ingombro ridotto appositamente progettata per essere integrata in dispositivi di piccole dimensioni che dispongono di uno spazio limitato per alloggiare un'interfaccia di comunicazione. In appena 9  $cm^2$  di spazio, contiene un potente microprocessore, integrato con un controller fast Ethernet e lo stack software completo, per fornire una piena funzionalità come Profinet IO Device. Anybus-IC può essere utilizzato con diversi connettori di rete o con terminali a vite che lo rendono una soluzione ideale per i dispositivi in ambiente industriale. Grazie all'ingombro ridotto Anybus-IC può essere installato in un alloggiamento DIL-32 standard, richiede una alimentazione a 5 V e fornisce un'interfaccia di rete completa con isolamento galvanico. Per ogni rete è disponibile una versione Anybus-IC separata: è disponibile per le Reti fieldbus di tipo Profibus, DeviceNet e CANopen, oltre che per le connessioni Ethernet che supportano i protocolli EtherNet/IP, Modbus-TCP e Profinet. Anybus-IC integra tutto l'hardware digitale e analogico, nonché tutto il software necessario per comunicare con

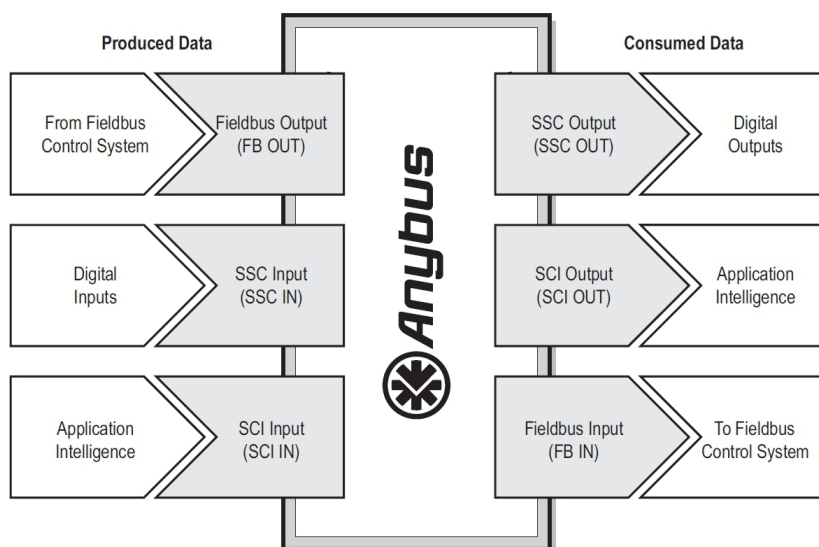
la rete selezionata: in questo modo HMS è in grado di fornire una soluzione certificata per la conformità con Fieldbus/Ethernet. Tra le varie funzionalità offerte dal chip si segnalano: un web server per scaricare qualsiasi tipo di pagina web specifica dell'applicazione, script Java o SSI per creare pagine web con informazioni e funzioni di controllo ed un e-mail client integrato. Nei dispositivi dotati di micro-ctrllore, l'Anybus-IC viene interfacciato



**Figura 3.4:** *Chip Anybus-IC*

tramite la seriale asincrona TTL, mentre negli apparati privi di intelligenza, l'Anybus-IC è in grado di controllare in completa autonomia una serie di shift register (registri a scorrimento) a cui possono essere collegati fino a 128 I/O fisici. In ogni caso, l'Anybus-IC gestisce l'intero protocollo Profinet IO, sgravando totalmente l'applicazione dal compito di dover governare la comunicazione sul lato Fieldbus. Ogni canale di comunicazione (SSC, Fieldbus, SCI) usa due buffer, uno contiene i dati in entrata (prodotti) e l'altro i dati in uscita (consumati). I dati prodotti in un canale possono venir mappati (i.e. copiati in modo automatico) nel buffer di uscita di un altro canale con inoltro da un'interfaccia all'altra (si veda la figura 3.5). In questo modo il modulo supporta la connettività Fieldbus, oltre a fornire I/O interni all'applicazione (attraverso la mappatura di I/O dall'interfaccia SSC all'interfaccia SCI e viceversa). Mappando gli I/O Fieldbus all'interfaccia SSC, gli ingressi e le uscite digitali di tale interfaccia diventano immediatamente disponibili dal fieldbus. Il modo con cui viene effettuata la mappatura viene specificato

dai parametri di I/O. Oltre all'interfaccia dati seriale, il modulo è dotato di una seconda interfaccia (MIF, Monitor Interface) a due fili utilizzata per la configurazione e il monitoraggio attraverso un programma di emulazione del terminale PC standard (p.e. HyperTerminal). L'utilizzo di questa interfaccia è opzionale ma fornisce l'accesso ai parametri del modulo, nonché a funzioni per la configurazione ed il monitoraggio dei dati.



**Figura 3.5:** IO Mapping del chip Anybus-IC

Per quanto riguarda la comunicazione, e quindi il protocollo Profinet IO, il chip supporta solo la comunicazione real-time di classe 1 (non sincronizzata).

### 3.4.1 Interfaccia di comunicazione seriale (SCI)

Dispositivi intelligenti quali encoder incrementali, sensori/attuatori, terminali operatore e unità di controllo motore dispongono generalmente di un proprio microcontrollore. Il modulo Anybus-IC si collega al micro-controllore di un dispositivo intelligente tramite un'interfaccia seriale TTL a due fili (SCI). Questo collegamento consente di accedere ai parametri interni del modulo e allo scambio di dati di I/O tramite il protocollo Modbus RTU integrato. Ciò garantisce l'accesso ai dati di I/O ciclici e ai parametri aciclici della rete. La comunicazione tra Anybus-IC e il microcontroller del dispositivo di automazione è basata sul protocollo Modbus RTU. Attraverso il canale SCI,



Anybus-IC supporta fino a 128 byte di dati di input e 128 byte di dati di output; inoltre tale canale supporta baud rate da 4.8 kbps a 57.6 kbps.

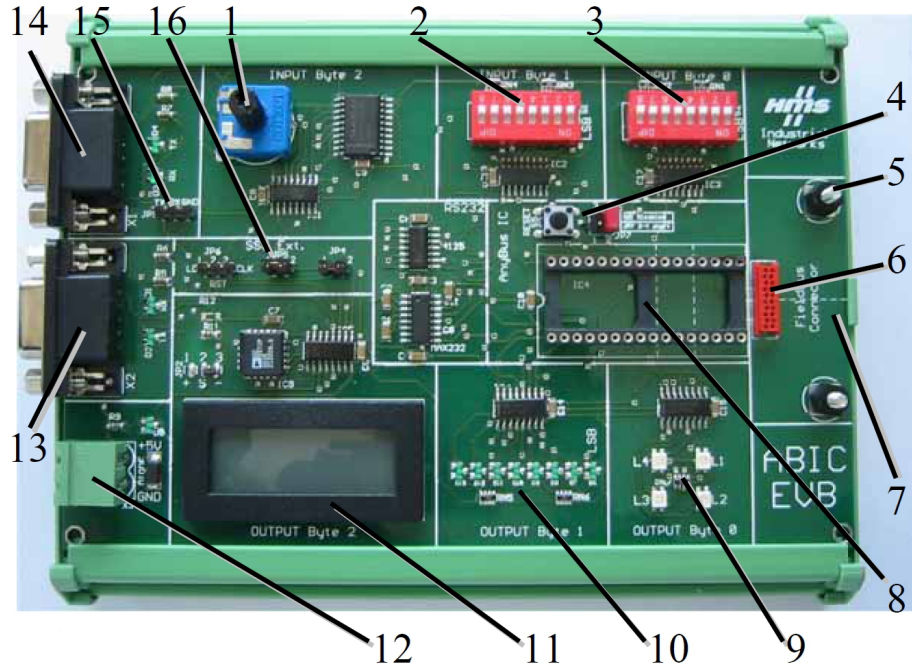
### 3.4.2 Canale seriale sincrono (SSC)

Per i dispositivi non intelligenti, come i connettori per valvole e i dispositivi I/O modulari, Anybus-IC fornisce un'interfaccia di registro con variazione del clock (SSC) per l'accesso diretto ai dati di I/O ciclici della rete senza la necessità di un microprocessore aggiuntivo. Con convertitori A/D o D/A supplementari, i segnali analogici di input o output possono essere integrati in modo semplice nelle Reti fieldbus o Ethernet industriali; inoltre, per espandere l'interfaccia SSC possono essere utilizzati shift register addizionali. L'aggiornamento sul canale SSC avviene mediante una scansione ciclica di tutti i shift register presenti con una temporizzazione di 4 ms.

### 3.4.3 Evaluation Board

La scheda Anybus-IC Evaluation Board (EVB) consente di inizializzare il modulo Anybus-IC e monitorare lo scambio dei dati direttamente da un programma di emulazione terminale sul PC. La scheda EVB è completa di tutti i componenti hardware necessari ad impostare indirizzi, baud rate e monitorare LED di stato del Fieldbus. Oltre all'interfaccia fieldbus (in questo caso Profinet IO), la scheda dispone di due interfacce aggiuntive per lo scambio dei dati (SCI e SSC). Queste interfacce funzionano in modo completamente indipendente l'una dall'altra e possono essere utilizzate contemporaneamente. Per quanto riguarda il canale sincrono, l'EVB mette a disposizione un massimo di 3 byte di ingresso (due dip switch e un potenziometro come ingresso analogico) e 3 byte di uscita (un display digitale e due diverse configurazioni di led). Mentre per quanto riguarda il canale asincrono SCI, oltre che ad una interfaccia seriale TTL per il collegamento al micro-controllore, è presente un'interfaccia RS-232 per un collegamento immediato ad un PC standard. Com'è noto, per pilotare in trasmissione la linea di comunicazione seriale sono necessari dei drivers di linea, per la conversione dei livelli logici TTL forniti dal microcontrollore in livelli di tensione RS-232, mentre per convertire i livelli di tensione RS-232 relativi ai dati ricevuti occorre impiegare dei ricevitori di linea in grado di fornire livelli logici TTL al micro-controllore. Entrambi i dispositivi sono disponibili nel circuito MAXIM 232 a 16 pin già integrato nell'Evaluation Board.

## Vista dettagliata



**Figura 3.6:** *evaluation board*

1. Potenziometro per valori analogici (Input byte #2);
2. Switch digitale per valori di input binari (Input byte #1);
3. Switch digitale per valori di input binari (Input byte #0);
4. Jumper per l'abilitazione del canale SCI / Switch di reset;
5. Viti per il fissaggio dei connettori di campo;
6. Interfaccia di campo;
7. Socket per il connettore di campo;
8. Alloggiamento DIL-32 per l'Anybus IC;
9. Led per l'uscita digitale / Led di diagnostica (Output Byte #0);
10. Led per l'uscita digitale (Output Byte #1);
11. Display digitale per l'uscita analogica (Output Byte #2);

12. Connettore di alimentazione (5V DC, 800 mA);
13. Connettore di interfaccia MIF (RS232);
14. Connettore di interfaccia SCI (RS232);
15. Connettore di interfaccia SCI (5V TTL);
16. Connettori per l'espansione delle SSC.

### 3.5 Oscilloscopio

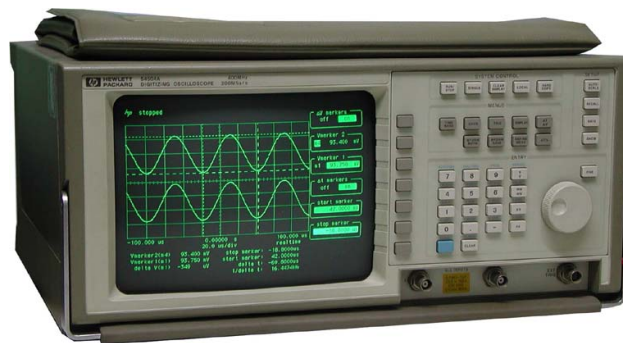


Figura 3.7: oscilloscopio HP 54504A

Per effettuare le misurazioni dagli ingressi o dalle uscite dei vari dispositivi ci si è dotati di un oscilloscopio Hewlett Packard 54504A; esso è un oscilloscopio digitale a due canali con una banda passante di 400 Mhz e con profondità di memoria di 200 MSa/s provvisto di due sonde passive 10:1 da 1 MOhm. È dotato di funzione *autoscale* e di varie funzioni matematiche che riescono a calcolare direttamente le ampiezze dei semiperiodi del segnale, il risetime, il fall-time, il ritardo tra due segnali, etc.

### 3.6 Software di progettazione

Siemens offre un pacchetto software chiamato STEP 7 che permette di gestire e configurare la maggior parte dei sistemi di automazione da lei prodotti. Il software viene fornito per l'installazione su sistemi operativi Windows. Per ogni progetto esso prevede determinati passaggi (step):

- Creazione della struttura del progetto.  
Un progetto assomiglia ad una cartella, nella quale tutti i dati sono organizzati gerarchicamente e sono sempre accessibili. Dopo aver creato un progetto, tutti gli altri compiti vengono eseguiti nel progetto stesso.
- Preparazione della stazione.  
Preparando la stazione si definisce il controllore, che nel nostro caso sarà una stazione SIMATIC 300.
- Configurazione dell'hardware.  
Nella fase di configurazione si definisce in una tabella di configurazione quali unità vengono utilizzate per la soluzione di automazione, e mediante quali indirizzi devono essere indirizzate le unità dall'interno del programma. È inoltre possibile impostare le proprietà delle unità per mezzo di parametri.
- Progettazione di reti e collegamenti di comunicazione.  
La base per la comunicazione è costituita da una rete precedentemente configurata. A tal fine, occorre creare le sotto-reti necessarie per le reti di automazione, definire le proprietà delle sotto-reti, le proprietà del collegamento di rete per le stazioni collegate, ed eventualmente i collegamenti occorrenti per la comunicazione.
- Creazione del programma.  
Con uno dei linguaggi di programmazione disponibili, l'utente crea un programma assegnato a unità o indipendente dalle unità, e lo salva sotto forma di blocchi, sorgenti o schemi. Nel nostro caso il semplicissimo programma è stato scritto in linguaggio AWL.
- Caricamento dei programmi nel sistema di destinazione.  
Dopo aver terminato la configurazione, la parametrizzazione e la creazione del programma, è possibile trasferire il programma utente completo o i suoi singoli blocchi nel sistema di destinazione (PLC).

Per la configurazione di una rete Profinet è necessario inoltre parametrizzare l'interfaccia di accesso alla rete stessa; in pratica viene abilitata la scheda di rete del PC in cui è caricato lo STEP 7. Bisogna tenere conto dell'indirizzo IP e dell'indirizzo di sotto rete dell'interfaccia per evitare malfunzionamenti in fase di download delle impostazioni. Anche ad ogni dispositivo di rete deve essere assegnato un IP e un Subnet Mask oltre che un nome dispositivo univoco.

### 3.6.1 Configurazione hardware

In ‘Configurazione HW’ si possono settare i parametri per ogni unità inserita nel progetto. Si riporteranno di seguito i parametri impostati per necessità in relazione alle misurazioni. Per la CPU315-2 PN/DP (indirizzo IP 192.168.0.1) è stato attivato il web server per poter accedere ai dati di diagnostica in modo rapido ed è stato impostato il campo relativo alla schedulazione dell’orologio (mediante OB35). Quest’ultima permette di interrompere la sequenza ciclica di programma, che viene elaborata nel blocco organizzativo OB1, per eseguire le istruzioni caricate nel blocco organizzativo OB35. Per questa CPU il tempo di schedulazione è impostabile da un minimo di 1ms a un massimo di 60s. Per le misurazioni questo valore è stato impostato a 4, 16, 64, 256 ms per cercare di seguire il ciclo di aggiornamento della comunicazione.

Il set di dati impostati per configurare l’IM153-4 (indirizzo IP 192.168.0.5) prevedono la scelta della modalità di sincronizzazione e il tempo di aggiornamento del ciclo IO. Nella scheda ‘Sincronizzazione’ è possibile impostare il tipo di comunicazione; come già detto l’IM153-4 prevede le classi di comunicazione RT e IRTflex. Visto che la CPU supporta solamente la RT class 1 non si ha possibilità di scelta. Il ciclo di IO si riferisce alla ciclicità temporale con cui avviene lo scambio di dati Real-Time tra l’IO-device (ET200M) e l’IO-Controller (CPU315-2 PN/DP). Il valore impostabile parte da un minimo di 1ms ad un massimo di 512ms. Per le prove da effettuare si sono scelti i valori di 4, 16, 64, 256 ms (come per la schedulazione dell’OB35).

Il chip Anybus-IC (indirizzo IP 192.168.0.4) supporta solamente la RT class 1; anche in questo caso sono stati impostati i soliti tempi di aggiornamento (4, 16, 64, 256 ms). Per la configurazione di altri parametri/funzionalità si è usata l’interfaccia MIF tramite cavo seriale collegato ad un PC e utilizzando il programma di Windows HyperTerminal si è riusciti ad entrare nella memoria del chip e settare ad esempio il numero di byte ingresso/uscita, il funzionamento dei led di diagnostica, nome e indirizzo del dispositivo, etc.

Per quanto riguarda lo switch (indirizzo IP 192.168.0.2) è stato attivato il Web Server e, quando necessario, il Port Mirroring per poter analizzare il traffico entrante/uscente della porta interessata. Un altro parametro fondamentale da tenere in considerazione è il tempo di ciclo di Profinet; purtroppo step 7 non dà la possibilità di impostarlo manualmente ma lo imposta automaticamente a seconda della rete (topologia, quantità e prestazioni dei device, prestazioni dell’IO-Controller e quantità di dati scambiati); nel caso in esame il ciclo Profinet IO è stato impostato ad 1 ms. Per poter monitorare il traffico di pacchetti Ethernet scambiati tra le due unità si è fatto uso di un programma freeware per Windows denominato Wireshark. Si tratta di un

analizzatore di rete software che sfrutta l'accesso a risorse di basso livello del sistema operativo, andando a catturare i pacchetti memorizzati nel buffer di ingresso della scheda di rete. Questo metodo, seppure molto utile a livello di analisi dei pacchetti, non permette una precisione temporale adeguata (a causa del sistema operativo non deterministico). In più non è possibile generare alcun segnale hardware di uscita per segnalare l'arrivo di un pacchetto, pena l'introduzione di ulteriori ritardi. Quindi il tool Wireshark, installato su un PC con scheda di rete Intel PRO/100 S, è stato utilizzato solamente per prendere visione del tipo di pacchetti trasportati dalla rete, il loro ordine di arrivo e la loro frequenza.

Infine sono stati installati tutti gli ultimi aggiornamenti del software di configurazione (STEP 7 V5.4 Service Pack 5) e le ultime versioni dei firmware di tutto l'hardware a disposizione.

# Capitolo 4

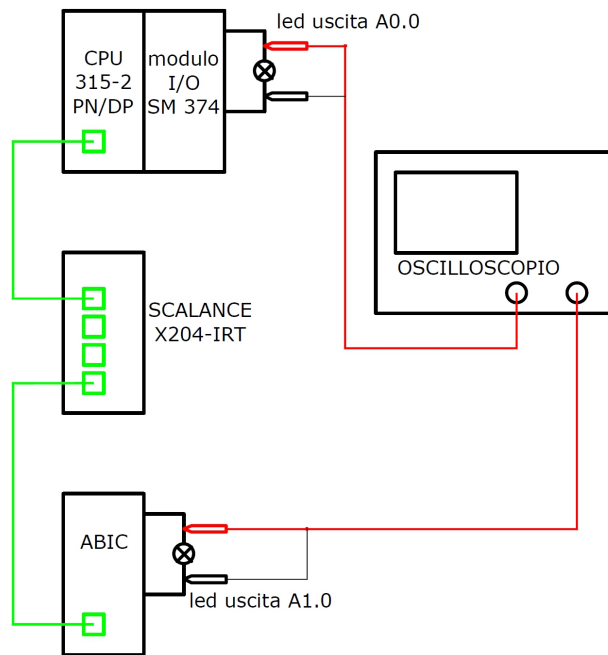
## Misurazioni

Per riuscire a comprendere se il chip Anybus-IC a disposizione fosse o meno preciso nello scambio dati, se rispettasse le caratteristiche fornite dal costruttore e più in generale se fosse un dispositivo Profinet IO adatto alle esigenze industriali di Weightpack, si è deciso di effettuare qualche misurazione dei tempi di trasmissione. Si è inserito il prodotto di HMS in una rete Profinet IO con altri prodotti Siemens: la CPU 315-2 PN/DP, lo switch SCALANCE X204IRT, un blocco I/O ET200M ed un simulatore di ingressi ed uscite digitali SM374. Le misurazioni eseguite necessitano di una descrizione separata per la diversa tipologia e per le differenti topologie di collegamenti adottati in ciascuna di esse.

### 4.1 Misura 1

La prima misura intrapresa, la cui topologia è riportata in figura 4.1, vuole quantificare il lasso temporale tra l'invio di un segnale all'uscita del PLC e l'effettiva commutazione dell'uscita dell'Evaluation Board. L'uscita in questione è rappresentata dal bit 0 del byte 1 dell'EVB descritta nel capitolo precedente. Per poter effettuare questa misura si è collegato il blocco di ingressi ed uscite digitali SM374 direttamente alla CPU tramite connettore posteriore *backplane*, mentre il chip lo si è collegato tramite rete Profinet IO. Si è quindi scritto un semplice programma di prova che è stato caricato nell'OB35 del plc. Il senso di questa misura è appunto determinare il ritardo introdotto dalla comunicazione Profinet IO. Per eseguire questa misura si

sono impostati sia il tempo di schedulazione sia il tempo di aggiornamento dell'ABIC PRT a 4, a 16, a 64 e a 256 ms. La sonda del canale uno dell'oscilloscopio ( $S_1$ ) è stata collegata ad un led del blocco I/O, uscita A0.0, collegato alla cpu. La sonda due ( $S_2$ ) è stata collegata ad un led dell'EVB, uscita A1.0. Il programma caricato in OB35 fornisce in contemporanea il medesimo valore alle uscite, poiché vengono settate nell'immagine di processo delle uscite del plc.



**Figura 4.1:** Topologia della misura 1

Si riassumono, con le seguenti formule, i calcoli delle tempistiche relative ai due canali dell'oscilloscopio:

$$\begin{cases} T_{S1} = T_{CPU} + T_{EL-374} \\ T_{S2} = T_{CPU} + T_{CPU-PNIO} + T_{TX} + T_{PNIO-ABIC} + T_{EL-ABIC} \\ \quad + T_{OFFSET-PNIO} \end{cases}$$

Dove:

- $T_{CPU}$  indica ogni quanto viene eseguito il ciclo macchina, sarà circa uguale al valore di schedulazione dell'orologio impostato per OB35;
- $T_{EL-374}$  è il tempo di reazione impiegato dalla cpu a trasmettere i dati al blocco I/O 374 fino alla commutazione del led d'uscita A0.0;



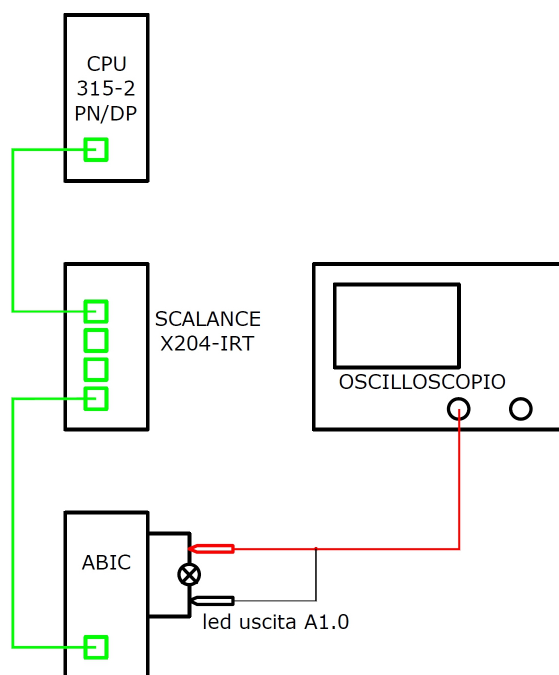
- $T_{CPU-PNIO}$  è il tempo di reazione impiegato dalla cpu a scrivere i dati nel buffer di comunicazione del sistema Profinet;
- $T_{TX}$  è il tempo impiegato dalla rete Profinet a trasferire il dato dal buffer della cpu a quello del device (ABIC-PRT), compreso il passaggio nello switch;
- $T_{PNIO-ABIC}$  è il tempo di reazione impiegato dal dato a passare dal buffer Profinet dell'Anybus-IC al chip stesso;
- $T_{EL-ABIC}$  è il tempo di reazione impiegato dall'ABIC a trasmettere i dati dal chip fino alla commutazione del led d'uscita A1.0;
- $T_{OFFSET-PNIO}$  è lo sfasamento tra quando la CPU entra in funzione ciclica e quando si instaura la comunicazione Profinet tra Cpu e ABIC; è collegato quindi al ciclo di bus Profinet ed è al massimo uguale al tempo di aggiornamento impostato nell'IO Device.

A causa della variabilità del  $T_{OFFSET-PNIO}$  non è possibile rilevare delle misure che caratterizzino correttamente il sistema: si ha infatti un errore di misura dello stesso ordine di grandezza della misura stessa. Quando la cpu viene posizionata in 'Stop' la comunicazione Profinet resta attiva; non appena si passa allo stato 'Run' si avvia il dialogo su Profinet: questo dialogo si aggancia al primo ciclo bus utile disponibile (che nel nostro caso è di 1 ms) fino ad avere un ritardo massimo pari al tempo di aggiornamento del chip Anybus-IC. Tali misure sono quindi direttamente correlate con l'istante in cui l'operatore umano avvia la cpu e di conseguenza il valore misurato non può essere analiticamente rilevante.

## 4.2 Misura 2

La seconda configurazione di misura, rappresentata in figura 4.2, ha lo scopo di misurare la regolarità della comunicazione tra il chip e il plc. Si è caricato nella CPU un programma che commuta un bit d'uscita ad ogni ciclo di schedulazione del blocco organizzativo OB35. A tale proposito si è settato lo stesso valore al tempo di schedulazione e al tempo di aggiornamento dell'ABIC. Sono state effettuate misure a 16, 64 e 256 ms allo scopo di verificare la regolarità e la costanza delle commutazioni; si son raccolti 150 campioni per ogni semiperiodo positivo e altrettanti per quello negativo. Nella fase di acquisizione dati si è tenuto conto dell'impossibilità fisica di sincronizzare i

periodi di OB35 e i cicli di aggiornamento Profinet del chip e quindi si sono scartate le misure palesemente errate ai fini del senso della misurazione. Alla luce dei ritardi introdotti dal canale sincrono della EVB si è deciso di non effettuare misurazioni a 4 ms in quanto non sarebbe stato possibile differenziare i valori errati introdotti dalla perdita di sincronismo tra schedulazione ed aggiornamento da quelli introdotti dal ritardo di elaborazione del dato del canale sincrono SSC: queste misurazioni non sarebbero quindi state sensibili ai fini della caratterizzazione del chip.



**Figura 4.2:** *Topologia della misura 2*

Si riportano i grafici con i valori rilevati in fase di misurazione, da questi si evince la buona precisione del chip nell'intorno del valore atteso del tempo impostato da Step7 come aggiornamento dell'ABIC-PRT. Come possiamo vedere dai grafici il ritardo massimo rilevato è stato intorno ai 4 ms, pari al tempo di aggiornamento del canale sincrono SSC. Le misure sono molto precise, infatti il tempo di refresh del canale sincrono è un sottomultiplo del tempo di aggiornamento del chip, di conseguenza i semiperiodi di accensione e spegnimento del led sono risultati molto stabili. In alcuni rari casi si hanno dei semiperiodi di tempo lunghi 4 ms più del valore medio; ciò accade quando il canale sincrono non riesce ad aggiornare in tempo il led d'uscita e deve aspettare il ciclo di refresh successivo per la commutazione del led.

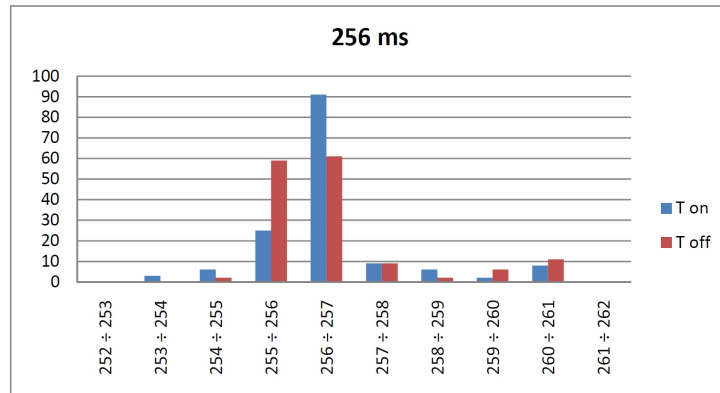


Figura 4.3: misura 2, 256 ms, unità di misura [ms]

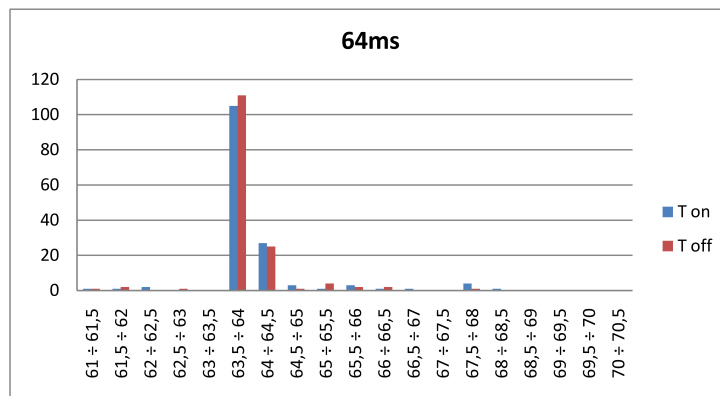


Figura 4.4: misura 2, 64 ms, unità di misura [ms]

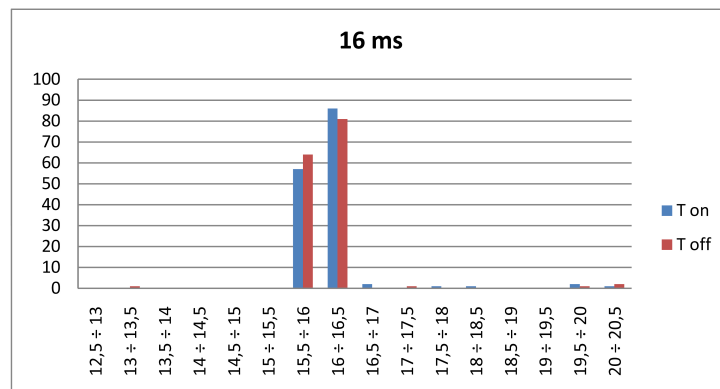
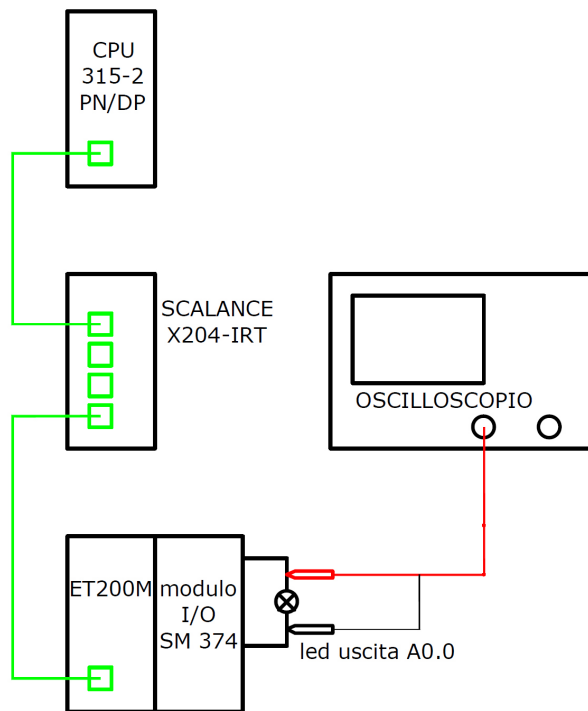


Figura 4.5: misura 2, 16 ms, unità di misura [ms]

### 4.3 Misura 3

La terza configurazione ripropone la stessa tipologia di misura fatta nella sezione precedente. In questo caso si è testato il modulo ET200M anziché l'Anubus IC al fine di poter comparare i due IO device. Anche in questo caso sono state effettuate misure a 16, 64 e 256 ms allo scopo di verificare la regolarità e la costanza delle commutazioni; si sono raccolti 150 campioni per ogni semiperiodo positivo e altrettanti per quello negativo. Come nella misurazione precedente, nella fase di acquisizione dati si è tenuto conto dell'impossibilità fisica di sincronizzare i periodi di OB35 e i cicli di aggiornamento Profinet dell'ET200M e quindi si sono scartate le misure palesemente errate ai fini della misurazione.



**Figura 4.6:** *Topologia della misura 3*

Si riportano i grafici con i valori rilevati in fase di misurazione: si nota come i tempi impiegati dal modulo stiano nell'intorno del valore temporale che ci si poteva aspettare (prossimo al tempo di aggiornamento dell'ET200M impostato da Step7).

Nella tabella seguente si riportano le medie e le deviazioni standard delle misurazioni di entrambi i moduli:

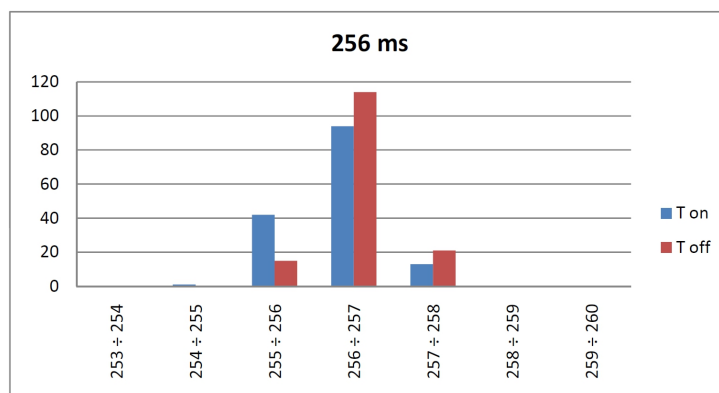


Figura 4.7: misura 3, 256 ms, unità di misura [ms]

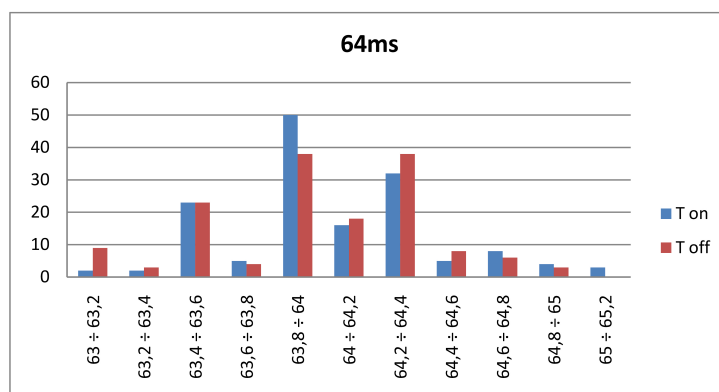


Figura 4.8: misura 3, 64 ms, unità di misura [ms]

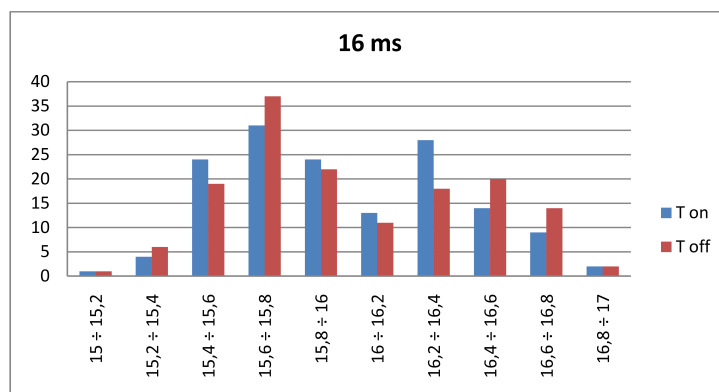


Figura 4.9: misura 3, 16 ms, unità di misura [ms]

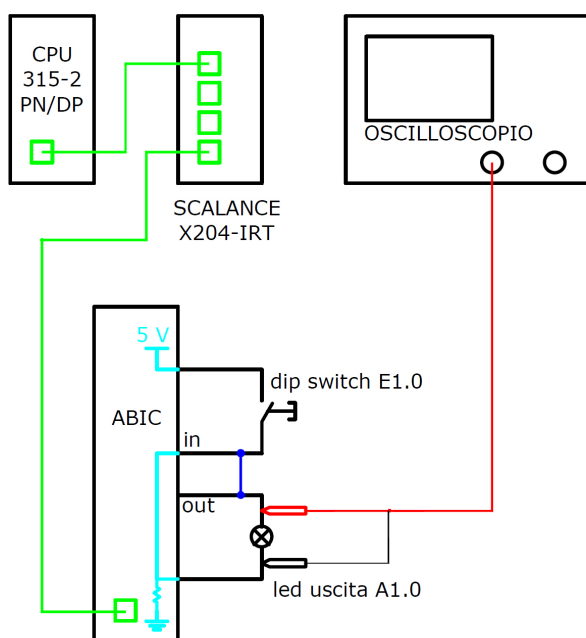
misura 2 - Anybus-IC			
		$\mu$ [ms]	$\sigma$ [ms]
Ton	256ms	256.1	1.30
Toff	256ms	256.1	1.45
Ton	64ms	64.0	0.90
Toff	64ms	63.9	0.63
Ton	16ms	16.1	0.59
Toff	16ms	16.0	0.63

misura 3 - ET200M			
		$\mu$ [ms]	$\sigma$ [ms]
Ton	256ms	255.8	0.60
Toff	256ms	256.0	0.49
Ton	64ms	64.0	0.40
Toff	64ms	63.9	0.41
Ton	16ms	15.9	0.40
Toff	16ms	16.0	0.42

Si nota come le deviazioni standard dell'ET200M variano di poco il loro valore, mentre nel chip Anybus-IC si ha una variazione direttamente proporzionale all'aumentare del tempo di aggiornamento della scheda. Si evince dunque che l'ET200M è un dispositivo più accurato del chip di HMS: questo risultato era facilmente intuibile in quanto l'ET200M è stato progettato per la comunicazione di classe IRT (più precisamente RT Class 2), quindi molto più prestante che non la RT class 1. Si nota inoltre come il valore della deviazione standard misurato sia abbastanza costante nell'ET200M e come tale valore rientri nelle specifiche riportate dal costruttore, che dichiara tempi di commutazioni delle uscite entro i  $500 \mu s$ .

## 4.4 Misura 4

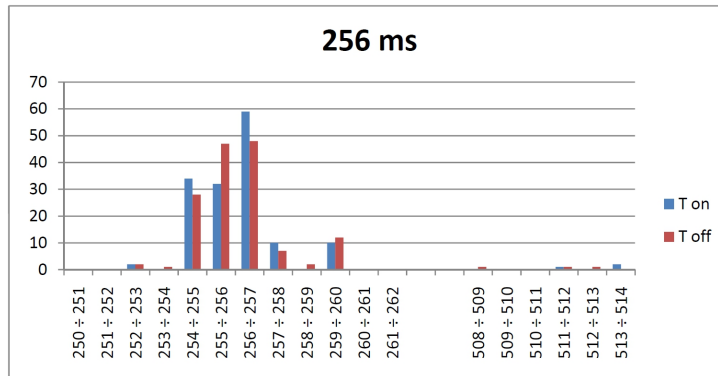
Come quarta misurazione si è voluto testare il tempo complessivo di lettura-scrittura nella Evaluation Board del chip. La topologia è rappresentata in figura 4.10: nell'EVB è stato effettuato un ponte fisico tra il primo bit del byte 1 d'ingresso ed il primo bit del byte 1 d'uscita. In aggiunta si è caricato nella CPU un programma che settasse ad ogni ciclo l'uscita A1.0 come negazione dell'ingresso letto E1.0. Il programma è stato caricato nell'OB1 e non nell'OB35 in quanto il tempo di schedulazione avrebbe introdotto un ulteriore errore causato dalla mancanza di sincronismo.



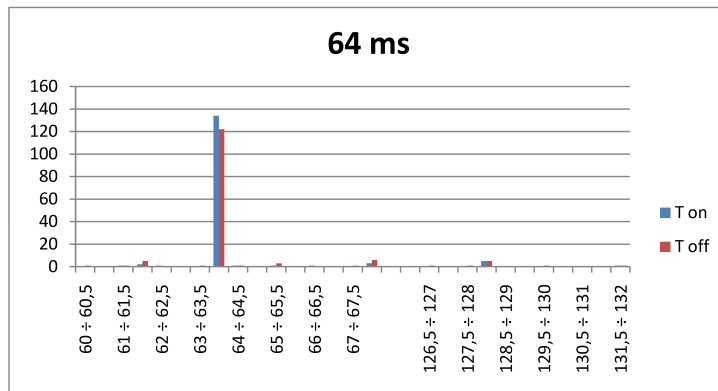
**Figura 4.10:** *Topologia della misura 4*

Così facendo si è potuto osservare una commutazione ciclica nel led d'uscita: il risultato di questo andamento è riportato nei grafici seguenti dove ad ogni istogramma corrisponde un diverso settaggio dei tempi di aggiornamento del chip.

Come si può notare dai grafici, al diminuire del tempo di aggiornamento del Device, aumentano le occorrenze in cui raddoppia il semiperiodo di accensione/spegnimento del led di uscita. Questo comportamento, seppure sporadico a tempi di aggiornamento elevati, è risultato ciclico e di costante ripetizione. Per poterlo spiegare a livello analitico si è dovuti scendere nel dettaglio sul



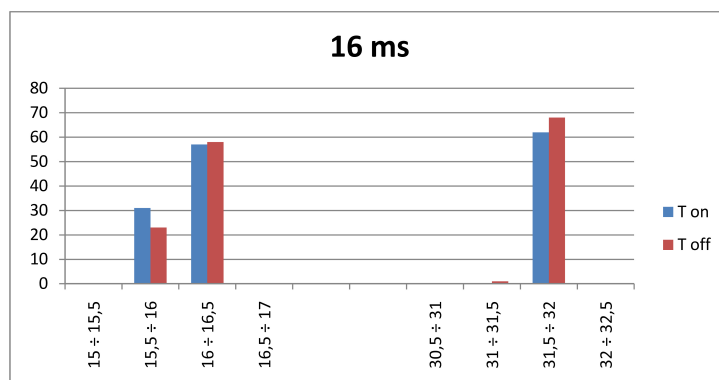
**Figura 4.11:** misura 4, 256 ms, unità di misura [ms]



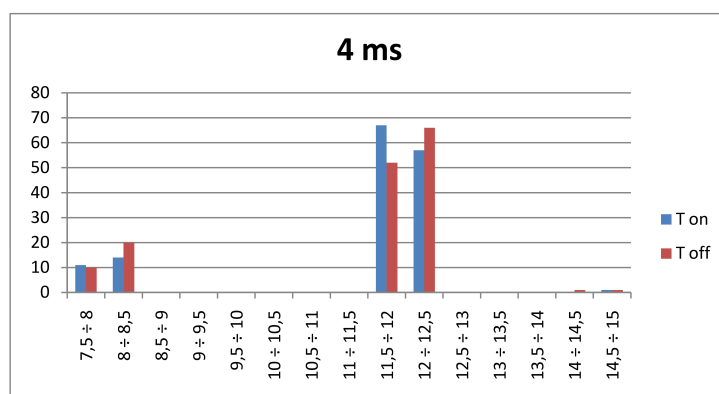
**Figura 4.12:** misura 4, 64 ms, unità di misura [ms]

funzionamento del chip e della scheda EVB in cui è installato. L'Evaluation Board mette a disposizione due tipi diversi di gestione dei dati sul lato seriale: la SSC, seriale sincrona e la SCI asincrona (la sincronizzazione è lasciata in questo caso al microprocessore che andrà a gestire l'ABIC una volta configurato). Nelle nostre misurazioni abbiamo utilizzato solo input ed output appoggiati all'interfaccia SSC dell'Evaluation Board. Questa interfaccia presenta però un funzionamento e delle tempistiche intrinseche non trascurabili ai fini delle nostre misure: il periodo con cui viene eseguito il refresh sulla seriale sincrona è di circa 4 ms e questo dato, fornitoci dal costruttore, va ad incrementare i tempi di risposta che ci si aspetterebbe. La memoria del chip Anybus è gestita in contemporanea sia dal lato fieldbus (Profinet IO nel caso nostro) sia dal lato seriale (canali SSC ed SCI) e viene aggiornata ogni 1 ms (dato fornito dal supporto tecnico di HMS). Nel caso della nostra misurazione l'incidenza di questi ritardi si fa sentire raddoppiata (un ciclo di refresh della memoria per la scrittura dell'uscita ed un altro per la lettura





**Figura 4.13:** misura 4, 16 ms, unità di misura [ms]

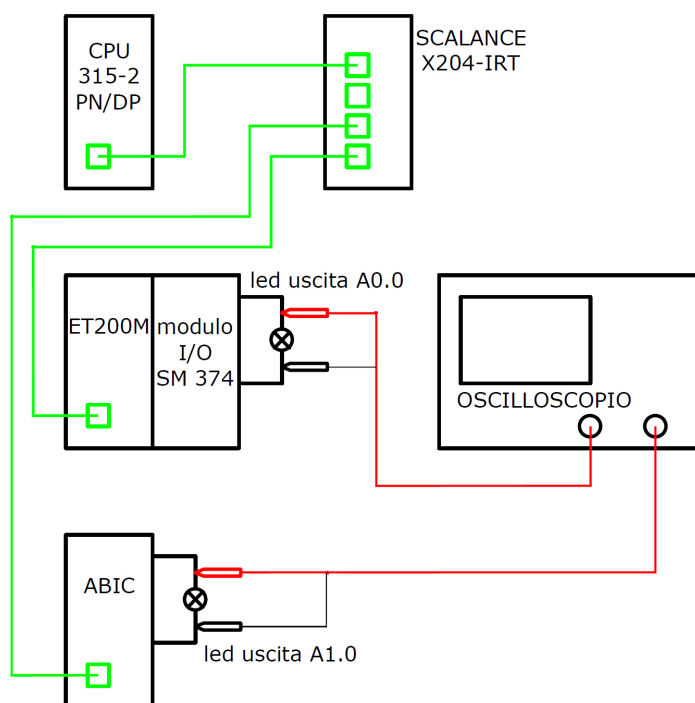


**Figura 4.14:** misura 4, 4 ms, unità di misura [ms]

dell'ingresso): ciò si nota perfettamente nelle misurazioni eseguite con tempo di aggiornamento di 4 ms, in questo caso il ciclo lettura-scrittura dura la maggior parte delle volte ben 12 ms. Guardando i grafici si osserva infatti come le occorrenze, in cui il periodo non è coincidente con il tempo di aggiornamento impostato, crescano al decrescere del tempo di aggiornamento impostato.

## 4.5 Misura 5

In questa misurazione si è voluto stimare la differenza tra i due istanti in cui avviene la commutazione nei due IO device ET200M e Anybus-IC. Per fare ciò abbiamo predisposto la topologia come in figura 4.15: una sonda dell'oscilloscopio collegata ad un'uscita del chip, l'altra ad un'uscita dell'ET200M. Si è misurato il ritardo rispetto all'ET200M utilizzando una funzione di misura dell'oscilloscopio in dotazione; impostando tempi di schedulazione in OB35 e tempi di aggiornamento uguali tra loro, si sono eseguite tre misurazioni, rispettivamente a 16, 64 e 256 ms.



**Figura 4.15:** *Topologia della misura 5*

L'unico grafico riportato in figura 4.16 propone tutti i tempi misurati. Si può notare come la distribuzione sia abbastanza uniforme tra -1 e 5 ms. Il valore negativo viene facilmente spiegato: analizzando il flusso di dati con il programma Wireshark abbiamo visto che il plc comunica con ET200M ed ABIC ad intervallo di un ciclo di bus Profinet IO diverso: in un ciclo di bus viene inviato il frame verso il chip, mentre in quello subito successivo (1 ms dopo) viene inviato all'ET200M.

La modalità con cui avviene la comunicazione è decisa automaticamente da Step7 e quindi tale valore non è modificabile; alla luce di ciò si deve pensare

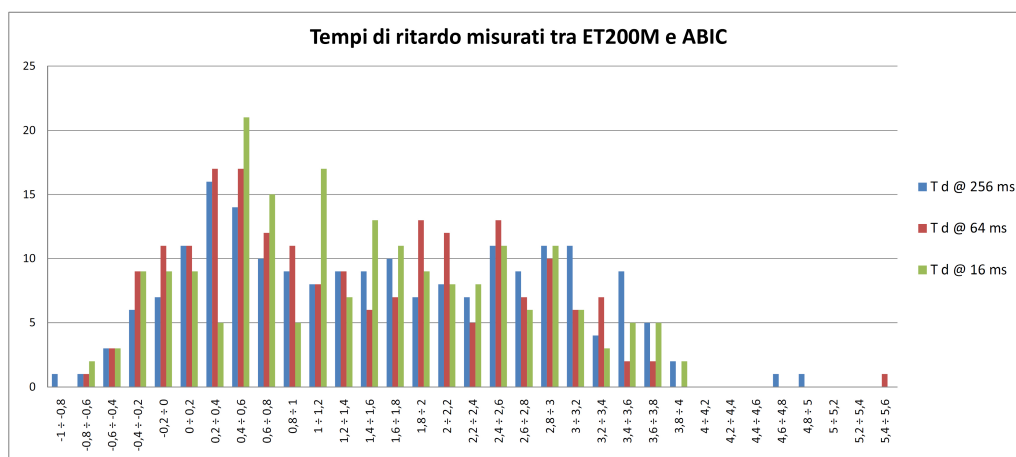


Figura 4.16: ritardi, unità di misura [ms]

che i ritardi dell'Anybus-IC andrebbero tutti traslati di 1 ms portando quindi il ritardo globale tra 0 e 5 ms circa. Si ha infatti che anche in questo caso i cicli di lettura scrittura del canale seriale sincrono dell'Evaluation Board vanno a modificare sensibilmente i tempi di ritardo tra i due IO Device. La misura infatti è eseguita in modo differenziale, cioè:

$$\begin{aligned}
 T_{S-ABIC} &= T_{CPU} + T_{CPU-PNIO} + T_{TX} + T_{PNIO-ABIC} + T_{EL-ABIC} \\
 T_{S-ET200M} &= T_{CPU} + T_{CPU-PNIO} + T_{TX} + T_{PNIO-ET200M} + T_{EL-ET200M} \\
 &\quad + T_{FASE}
 \end{aligned}$$

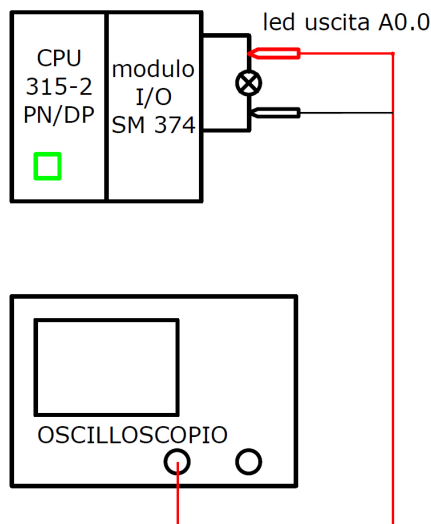
Dove il  $T_{FASE}$  è il tempo di sfasamento introdotto dalla rete Profinet tra i due IO device (nel nostro caso 1 ms) mentre gli altri tempi sono già stati descritti in MISURA 2. Si è quindi attribuito un ritardo massimo tra i due moduli di circa 5 ms. Considerando che l'ABIC ha come tempo minimo di comunicazione fornito dal costruttore proprio 4 ms si può considerare soddisfacente la misurazione ottenuta, anche alla luce delle specifiche tecniche dell'ET200M che è un dispositivo progettato per applicazioni di rete IRT. È doveroso evidenziare come il limite di funzionamento rilevato dalle misurazioni sia da attribuire all'ABIC Evaluation Board, per come gestisce il canale sincrono SSC, e non direttamente al chip ABIC.

## 4.6 Misura 6

In questa misurazione abbiamo voluto testare a quanto ammontasse il tempo di ciclo del PLC (tempo di aggiornamento ‘minimo’ dell’uscita) e quanto fosse corretta la periodicità della schedulazione dell’orologio con il programma utilizzato nelle misure fatte finora. Sono state effettuati tre set di rilievi:

- Programma caricato in OB1 senza schedulazione dell’orologio;
- Programma caricato in OB35 con schedulazione dell’orologio impostata a 4 ms;
- Programma caricato in OB35 con schedulazione dell’orologio impostata a 16 ms.

Per fare ciò abbiamo collegato i dispositivi come in figura 4.17 e si è misurato il periodo di commutazione dell’uscita che veniva negata ad ogni ciclo macchina. In queste misure non si è potuto tenere presente il ritardo del modulo SM374, che abbiamo perciò ritenuto trascurabile relativamente alle tempistiche ottenute.



**Figura 4.17:** *Topologia della misura 6*

I risultati ottenuti sono riportati nei tre grafici seguenti.

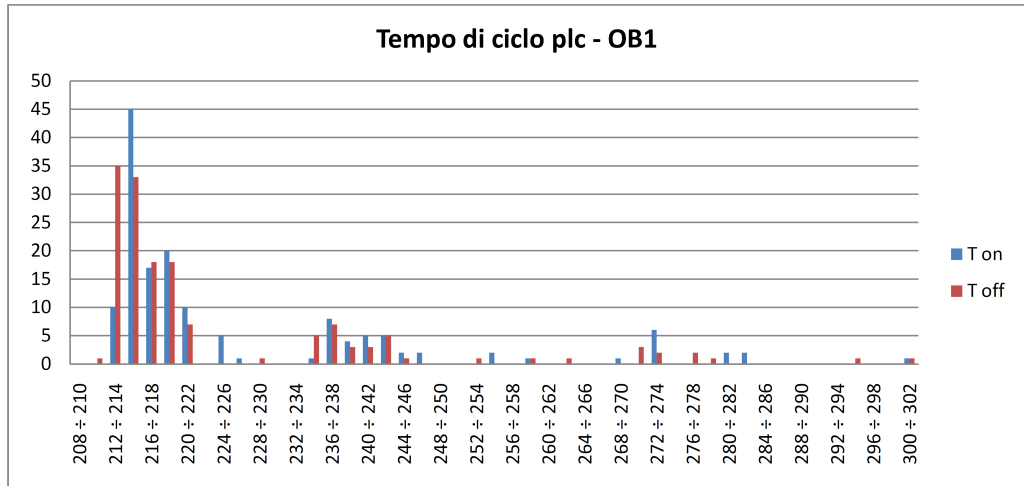


Figura 4.18: OB1, unità di misura [μs]

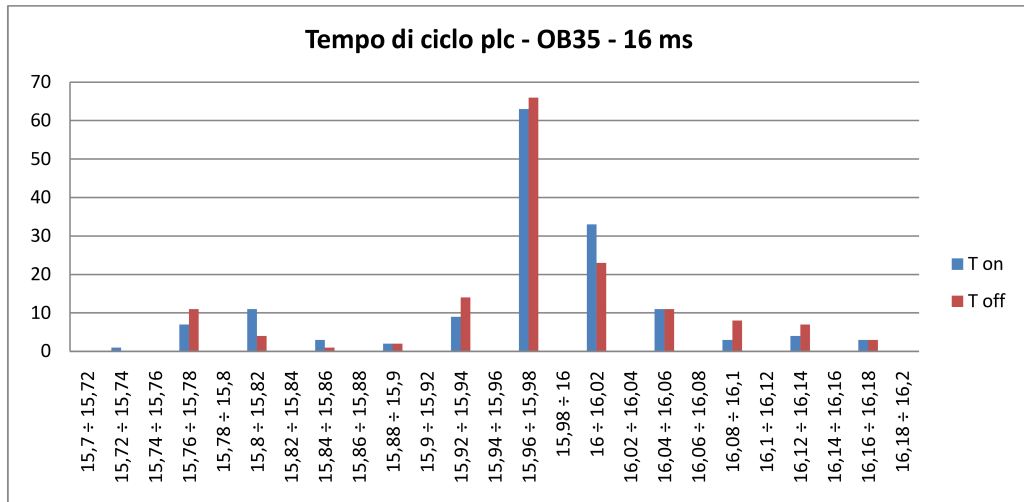


Figura 4.19: OB 35 a 16 ms, unità di misura [ms]

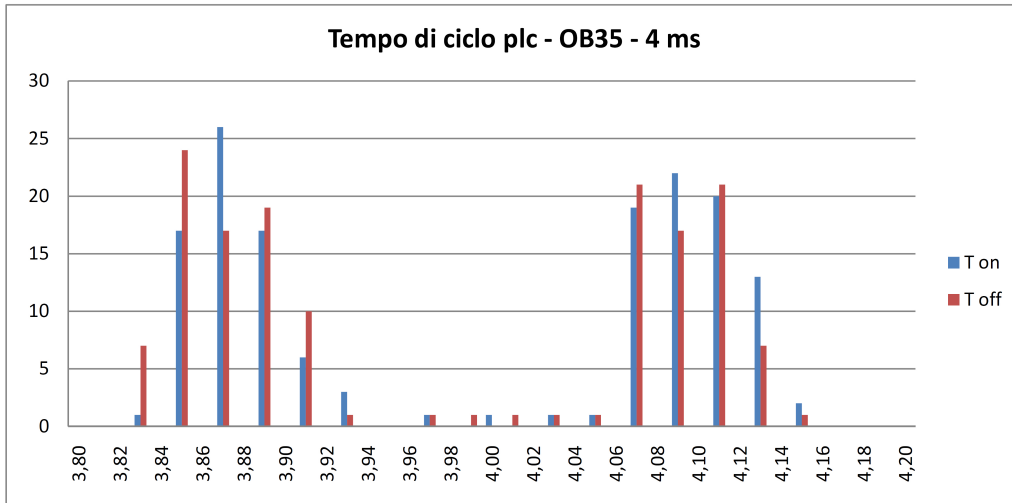


Figura 4.20: OB 35 a 4 ms, unità di misura [ms]

Si noti come i tempi di schedulazione siano abbastanza precisi: la variazione rispetto alla media è di circa  $\pm 150\mu s$ , tale dato rispecchia esattamente quello fornito dal costruttore (Siemens).

Si riportano infine la tabella con medie e deviazioni standard delle ultime misure.

CPU		$\mu$	$\sigma$
Ton	OB1	226 $\mu s$	19 $\mu s$
Toff	OB1	224 $\mu s$	18 $\mu s$
Ton	OB35, 4ms	3.99 ms	0.11 ms
Toff	OB35, 4ms	3.98 ms	0.11 ms
Ton	OB35, 16ms	15.97 ms	0.09 ms
Toff	OB35, 16ms	15.98 ms	0.09 ms

## Conclusioni

Al giorno d'oggi l'applicazione di Ethernet a livello industriale sembra essere il futuro dell'automazione distribuita. Tra l'ampia gamma di protocolli Real-Time Ethernet ci si è soffermati su Profinet, soluzione proposta da *PROFIBUS International User Group* che, per quanto riguarda le prestazioni, si situa nella fascia più alta. L'architettura Profinet dimostra un comportamento eccellente, soprattutto nel protocollo Profinet IO IRT, perchè riesce a coniugare il funzionamento real-time e la comunicazione isocrona senza riguardo alla ulteriore presenza di scambio di dati real-time e di traffico TCP/IP, garantendo comunque l'obiettivo di elevate prestazioni. Anche in fase di progettazione e configurazione della rete in laboratorio ci si è accorti dei vantaggi che questo nuovo bus di campo offre:

- Ha una fase di progettazione e sviluppo molto rapida ed efficace;
- Utilizza al meglio le tecnologie esistenti;
- Evidenzia l'attenzione della comunicazione nelle reti automatiche;
- Lascia una completa interoperabilità anche ad applicazioni molto differenti.

Per quanto riguarda la realtà dei sistemi di automazione proposti da Siemens, Profinet è stato implementato per sostituirsi a Profibus senza sconvolgere le metodologie di progettazione e configurazione di una rete industriale. Con questo non si può affermare con sicurezza che le reti Industrial Ethernet sostituiranno i tradizionali bus di campo, sicuramente però aprirà la strada a nuove applicazioni, supportando la migrazione verso l'intelligenza distribuita nel settore dell'automazione. Osservando i risultati delle misurazioni,

si capisce subito come l'abbinata Anybus-IC - Evaluation Board sia meno performante dell'ET 200M accoppiato con il simulatore SM 374. Questo era prevedibile visto che l'ET 200M è un dispositivo molto performante che supporta anche la comunicazione isocrona mentre il chip Anybus-IC supporta solo la comunicazione RT di classe 1. Tale divario prestazionale (si pensi alla deviazione standard delle misure) tuttavia non è da imputare completamente al chip HMS; infatti il collo di bottiglia del pacchetto HMS è la scheda di valutazione (EVB) fornita per testare il chip stesso. Quest'ultima ha dei tempi di aggiornamento nei canali di ingresso/uscita seriali sincroni di circa 4 ms ; nel caso, dunque, accada una leggera perdita di sincronismo tra tempo di refresh del canale sincrono e tempo di aggiornamento del chip, il semiperiodo di accensione o spegnimento del led dura 4 ms in più. Per questo motivo non si sono potute effettuare misurazioni con tempi di aggiornamento di 4 ms in quanto il ritardo di refresh della scheda sarebbe stato dello stesso ordine di grandezza, rendendo queste misurazioni non sensibili ai fini della caratterizzazione del chip. Tuttavia, anche nell'Anybus-IC, per tempi di aggiornamento maggiori di 16 ms (64 e 256 ms), i ritardi risiedono entro una variazione del 15 % rispetto al valor medio; quindi anche il chip rientra appieno nelle specifiche della classe 1 real-time di Profinet rendendolo un buon prodotto a livello industriale. C'è da dire inoltre che la rete che si è potuta allestire in laboratorio del dipartimento di ingegneria dell'informazione non è paragonabile con una rete industriale sia per dimensioni che per numero di device utilizzati; sarebbe infatti interessante vedere se in reti particolarmente complesse con un carico di dati molto maggiore, le prestazioni vengono mantenute. Un'altra prova interessante sarebbe non utilizzare gli ingressi SSC dell'Evaluation Board e di provare a usare degli ingressi dall'interfaccia Modbus RTU della piastra-peso che Weightpack realmente utilizza per le sue macchine imbottigliatrici.



# Bibliografia

- [1] M. Felser, T. Sauter  
*Standardization of Industrial Ethernet - the Next Battlefield?*  
IEEE, 2004
- [2] J. Jasperneite, P. Neumann  
*Switched Ethernet for Factory Communication*
- [3] Z. Wang, Y. Song, J. Chen, Y. Sun  
*Real Time Characteristics of Ethernet and Its Improvement*  
IEEE, 2001
- [4] P. Neumann  
*Communication in industrial automation - What is going on?*  
Elsevier, 2006
- [5] M. Alves, E. Tovar, F. Vasques  
*Ethernet Goes Real-Time: a Survey on Research and Technological Developments*  
Technical Report, 2000
- [6] S. Vitturi  
*On the use of Ethernet at low level of factory communication systems*  
Elsevier, 2001
- [7] J. Jaspernite, P. Neumann  
*How to guarantee realtime behaviour using ethernet*
- [8] M. Felser  
*Real-Time Ethernet - Industry Prospective*  
IEEE, 2005

- [9] G. Johnson  
*Determinism in industrial ethernet: A technology overview, Part1,2*  
[HTTP://WWW.PROCESSONLINE.COM.AU](http://www.processonline.com.au)
- [10] J. Jaspernite, K. Shehab, K. Weber  
*Enhancements to the Time Synchronization Standard IEEE-1588 for a System of Cascaded Bridges*
- [11] White Paper - Hirschmann  
*Precision Clock Synchronization - The Standard IEEE 1588*
- [12] G. Thomas *Introduction to Fast Ethernet*  
Dal periodico *Extension*, A Technical Supplement to *Control Network*, 2001
- [13] P. Doyle *Introduction to Real-Time Ethernet I-II*  
Dal periodico *Extension*, A Technical Supplement to *Control Network*, 2004
- [14] G. Thomas *Introduction to Switch Technology*  
Dal periodico *Extension*, A Technical Supplement to *Control Network*, 2000
- [15] P. Ferrari, A. Flammini, D. Marioli, A. Taroni  
*"Valutazioni Sperimentali delle Prestazioni di PROFINET"*  
Dal periodico *"Brescia Ricerche"* n.52, 2005
- [16] A. Poschmann, P. Neumann  
*Architecture and Model of Profinet IO*  
IEEE, 2004
- [17] J. Feld  
*PROFINET - Scalable Factory Communciation for all Application*  
IEEE, 2004
- [18] P. Ferrari, A. Flammini, D. Marioli, A. Taroni  
*Experimental evaluation of PROFINET performance*  
IEEE, 2004
- [19] F. Venturini  
*I bus di campo isocroni Profibus DP-V2 e Profinet I/O*
- [20] *PROFINET System Description*  
Manuale di sistema PI

- [21] H. Kleines, J. Sarkadi, F. Suxdorf, K. Zvoll  
*Profinet - An integrated automation concept based on ethernet*  
IEEE, 2003
- [22] H. Kleines, S. Detert, M. Drochner, F. Suxdorf  
*Performace Aspects of PROFINET IO*  
IEEE, 2007
- [23] R. Silvola, J. Hannila, J. Seppala, H. Koivisto  
*Experimental performance evaluation of PROFINET IO Real-Time version*
- [24] P. Ferrari, A. Flammini, D. Marioli, A. Taroni, F. Venturini  
*Experimental analysis to estimate jitter in PROFINET IO Class 1 networks*  
IEEE, 2006
- [25] P. Ferrari, A. Flammini, D. Marioli, A. Taroni  
*An experimental approach to estimate real-time characteristic of PROFINET IO versus PROFIBUS DP V2*  
2006
- [26] P. Ferrari, A. Flammini, S. Vitturi  
*Performace analysis of PROFINET networks*  
ELSEVIER, 2005
- [27] M. Schumacher, J. Jasperneite, K. Weber  
*A New Approach for Increasing the Performance of the Industrial Ethernet System PROFINET*  
IEEE, 2008
- [28] P. Neumann, A. Poschmann  
*Ethernet-based Real-Time Communications with PRONINET IO*
- [29] P. Ferrari, A. Flammini, D. Marioli, A. Taroni  
*Experimental evaluation of PROFINET performance*  
IEEE, 2004
- [30] A. Zuin  
*Caratterizzazione e analisi di una rete Ethernet industriale* Tesi di laurea, 2008
- [31] SIEMENS - SIMATIC  
*PROFINET Descrizione del sistema*  
Manuale di sistema, 2008

- [32] SIEMENS - SIMATIC  
*PROFINET IO Getting Started: Collection*  
Manuale di sistema, 2008
- [33] SIEMENS - SIMATIC  
*Configurazione dell'hardware e progettazione di collegamenti STEP7*  
Manuale, 2006
- [34] SIEMENS - SIMATIC  
*Sistema di automazione S7-300 Getting Started Collection*  
Manuale di sistema, 2008
- [35] SIEMENS - SIMATIC  
*Sistema di automazione S7-300, Caratteristiche delle unità modulari*  
Manuale del prodotto, 2009
- [36] SIEMENS  
*Switch Industrial Ethernet, SCALANCE X-200*  
Istruzioni operative, 2008
- [37] SIEMENS - SIMATIC  
*S7-300, CPU 31xC e CPU 31x: Dati tecnici*  
Manuale del prodotto, 2009
- [38] HMS  
*Anybus-IC PROFINET IO Fieldbus Appendix, Rev 1.02*
- [39] HMS  
*Anybus-IC*  
Design Guide, Rev. 1.59
- [40] Anybus  
*Anybus-IC Evaluation Board*  
User Manual, Rev. 1.01
- [41] HMS  
*Anybus-IC - Single Chip Solutions*  
Brochure