Università degli Studi di Padova

Dipartimento di Ingegneria dell'Informazione

Ottimizzazione del Consumo Energetico per Reti Ethernet Real-time

Laureando
Adriano Salata

Relatore
Prof. Stefano Vitturi

Correlatori
Ing. Lucia Seno
Ing. Federico Tramarin

Corso di Laurea Magistrale in
Ingegneria dell'Automazione

Data Laurea
11 Marzo 2013

Anno Accademico
2012-2013

Indice

Elenco delle figure

Elenco delle tabelle

Abstract

Negli ultimi anni l'efficienza energetica è divenuta un requisito fondamentale di qualsiasi tipo di dispositivo che necessiti di elettricità per funzionare. La forte crisi economica e i problemi legati all'inquinamento ambientale, infatti, hanno reso evidente il bisogno di eliminare il più possibile ogni tipo di spreco ottimizzando i consumi.
L'attenzione si è focalizzata anche sulle reti Ethernet che, dopo la grande diffusione di Internet e dei sistemi di comunicazione, si sono sviluppate con milioni di link in tutto il mondo, che comportano un consumo complessivo tutt'altro che trascurabile. Ciò ha presto portato alla nascita dello standard Energy Efficiency Ethernet (EEE), che ha appunto lo scopo di ridurre lo spreco di energia dei nodi Ethernet durante le fasi di inattività, cioè nei momenti in cui rimangono attivi senza la necessità di compiere trasmissioni.

Un discorso analogo può essere fatto per le reti industriali, alcune delle quali sono basate proprio sullo standard Ethernet e presentato gli stessi problemi dal punto di vista energetico. Anche in questo ambito, quindi, è possibile ridurre notevolmente i consumi dei nodi, implementando ad esempio i meccanismi proposti per le reti Ethernet e ottimizzando l'efficienza energetica.

In questa tesi verranno descritte le caratteristiche fondamentali dello standard EEE e il funzionamento di Ethernet POWERLINK, una delle più comuni reti Real-Time Ethernet (RTE). Si presenterà successivamente come possano essere implementati i meccanismi di risparmio energetico in queste reti, riportando le simulazioni effettuate e le potenzialità di riduzione dei consumi riscontrate.

Introduzione

L'efficienza energetica di un processo è definita come rapporto tra la potenza in uscita e la potenza in entrata. Negli ultimi decenni l'argomento ha acquistato molta importanza, tanto da divenire un requisito fondamentale per qualsiasi tipo di dispositivo tecnologico che necessiti di energia elettrica per funzionare ed ha cominciato ad assumere un significato più generico, infatti può essere espressa come il 'rapporto o altra relazione quantitativa tra i risultati in termini di prestazioni, servizi, beni o energia, e l'immissione di energia' [7]. Grandi sforzi sono stati fatti per ridurre gli sprechi, ad esempio, nel settore residenziale, con l'introduzione degli impianti di illuminazione a fluorescenza o di elettrodomestici e sistemi di riscaldamento efficienti [39].
Anche nel settore industriale sono state adottate numerose innovazioni in grado di ridurre notevolmente i consumi, come l'introduzione di motori elettrici ad alto rendimento.
Lo scopo di questo progetto è quello di focalizzare l'attenzione su un aspetto di questo settore ancora scarsamente tenuto in considerazione: il consumo dei sistemi di comunicazione industriali. Di recente infatti è sempre più diffusa l'adozione di reti all'interno degli impianti industriali, che permettono di creare sistemi distribuiti e che, in molti casi, utilizzano protocolli basati sullo standard Ethernet a cui vengono aggiunti dei meccanismi in grado di assicurare il determinismo richiesto da gran parte delle applicazioni industriali.
Alcuni passi in avanti per la riduzione dei consumi sono già stati compiuti per le comuni reti Ethernet, precisamente è stato definito un nuovo standard, denominato Energy Efficiency Ethernet (EEE), avente lo scopo di limitare il più possibile lo spreco da parte dei link nelle fasi di inattività, ovvero quando non vengono effettuate alcune trasmissioni. Tuttavia questi meccanismi di ottimizzazione non sono ancora stati introdotti nelle reti industriali, dove i nodi consumano costantemente anche quando non devono inviare o ricevere dati. Per questo motivo verrà descritta una delle reti più diffuse in ambito industriale, Ethernet POWERLINK (EPL), e si mostreranno le potenzialità di riduzione dei consumi utilizzando al suo interno gli algoritmi caratteristici dell'EEE.
Come per le comuni reti Ethernet il risparmio ottenuto da un singolo nodo risulta essere ir-

risorio poichè il suo consumo è decisamente ridotto. Osservando il problema tenendo conto del grande numero di link presenti in tutto il mondo, ci si rende conto che le potenzialità di risparmio dal punto di vista economico e di riduzione dell'impatto ambientale conseguenti alla produzione di energia giustifica ampliamente l'interesse su questo argomento. Altro aspetto favorevole è che la creazione di nuove reti industriali in grado di ottimizzare il consumo energetico non comporta un investimento aggiuntivo, ma richiede semplicemente di adottare dei dispositivi che implementano già lo standard EEE. Un discorso analogo può essere fatto per quanto riguarda l'aggiornamento delle reti già presenti, in cui è richiesto di sostituire le schede di rete precedentemente installate che, generalmente, hanno un costo ridotto rispetto a quello degli altri dispositivi utilizzati in questo ambito.

Nel capitolo 1 verrà presentata la struttura ed il funzionamento delle reti Ethernet e la diffusione, in campo industriale, delle reti Real-Time Ethernet (RTE) basate sullo standard IEEE 802.3 e in grado di assicurare trasmissioni deterministiche. Successivamente nel capitolo 2 verrà descritto lo standard EPL. In particolare si presenterà il suo ciclo di funzionamento, le tipologie di noti utilizzati, i frame trasmessi e le tempistiche che lo caratterizzano.
Alcuni algoritmi proposti per ridurre il consumo energetico nelle reti Ethernet vengono presentati nel capitolo 3, in cui si prendono in considerazione due soluzioni che si sono dimostrate inefficaci, ma che hanno portato alla definizione dell'attuale standard EEE, in grado di ridurre sostanzialmente gli sprechi in queste reti. Nel capitolo 4 verrà descritto il possibile utilizzo dello standard EEE nelle reti EPL, prendendo in esame tutti i vari casi che si possono presentare e gli accorgimenti che si possono adottare per massimizzare la riduzione dei consumi. Infine, nel capitolo 5, verrà presentato un modello dei nodi EPL, che si utilizzerà per simulare il consumo energetico di alcune reti POWERLINK e per stimare la potenzialità di risparmio derivante dall'adozione dei meccanismi dello standard EEE.

Capitolo 1

Reti Ethernet in Ambienti Industriali

I sistemi di comunicazione in ambienti industriali sono generalmente costituiti da tre livelli differenti, ognuno associato a fasi diverse della produzione. Come si può vedere in figura 1.1, si distinguono:

livello di dispositivo: rappresenta una piccola unità lavorativa in cui un dispositivo controllore è connesso tramite linee ingresso/uscita ad alcuni dei sensori e degli attuatori dell'impianto. É caratterizzato dallo scambio di segnali fisici e le comunicazioni sono di tipo real-time, cioè impongono deadline stringenti sulle trasmissioni;

livello di cella: rappresenta una unità di lavoro di medie dimensioni con la funzione di coordinazione di più dispositivi. Vengono effettuate trasmissioni di messaggi di dimensione maggiore rispetto al caso precedente ma le velocità sono inferiori e le deadline imposte sono meno stringenti;

livello di impianto: è il livello più esteso, che ha la funzione di programmare e gestire la produzione. Le comunicazioni avvengono grazie a reti di grandi dimensioni in cui i dati scambiati non sono critici ma la loro quantità può essere rilevante.

Le tradizionali architetture di comunicazione che hanno caratterizzato i sistemi industriali fino agli inizi degli anni ottanta erano costituite esclusivamente da collegamenti seriali di tipo punto-punto. Il continuo sviluppo tecnologico e il crescere di necessità come la modularità, la decentralizzazione del controllo, l'utilizzo di una diagnostica integrata, una facile e rapida manutenzione e la diminuzione dei costi [37, 34], hanno presto portato alla nascita delle reti di comunicazione di campo, chiamate anche *fieldbus*, che hanno rivoluzionato lo scambio di informazione soprattutto nei livelli di dispositivo e di cella.

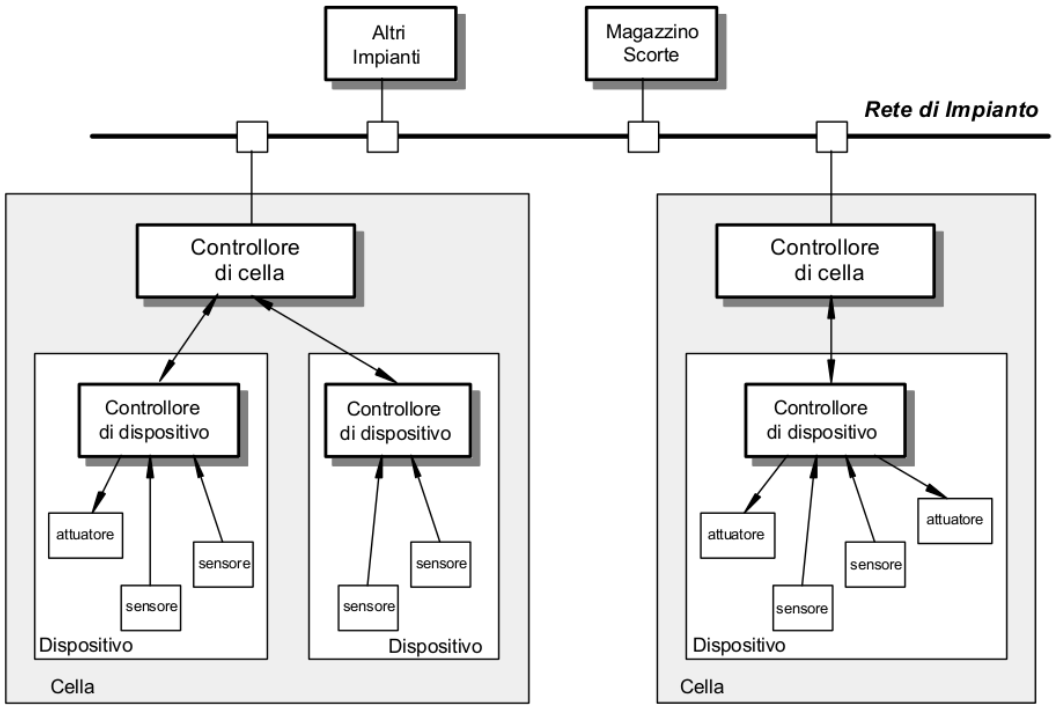


Figura 1.1 – Descrizione della rete di un impiando industriale [44].

Dal punto di vista fisico fanno uso del cavo coassiale, del doppino schermato o della fibra ottica, come avviene nelle comuni Local Area Network (LAN), ma si hanno delle differenze notevoli per quanto riguarda l'architettura di comunicazione e i protocolli utilizzati.

In figura 1.2 si può notare come varia la struttura dei sistemi di comunicazione industriali quando vengono adottati i fieldbus. Per come sono stati definiti i due livelli in cui queste reti vengono utilizzati, risulta chiaro che devono essere in grado di fornire le seguenti funzionalità:

scambio di dati ciclico: è una funzione richiesta dal livello dispositivo che consente al controllore l'acquisizione periodica di dati provenienti dai sensori e l'aggiornamento periodico dei dati da inviare agli attuatori;

traffico urgente asincrono: anche questa funzione è legata al livello dispositivo ed è necessaria a notificare rapidamente situazioni di allarme. Deve quindi essere assicurato il trasferimento di piccole quantità di dati in tempi rapidi ed in corrispondenza di eventi non predicibili;

messaggistica di alto livello: si tratta dello scambio di dati e strutture di dimensioni rilevanti tra i livelli di dispositivo e di cella oppure verso il livello impianto.

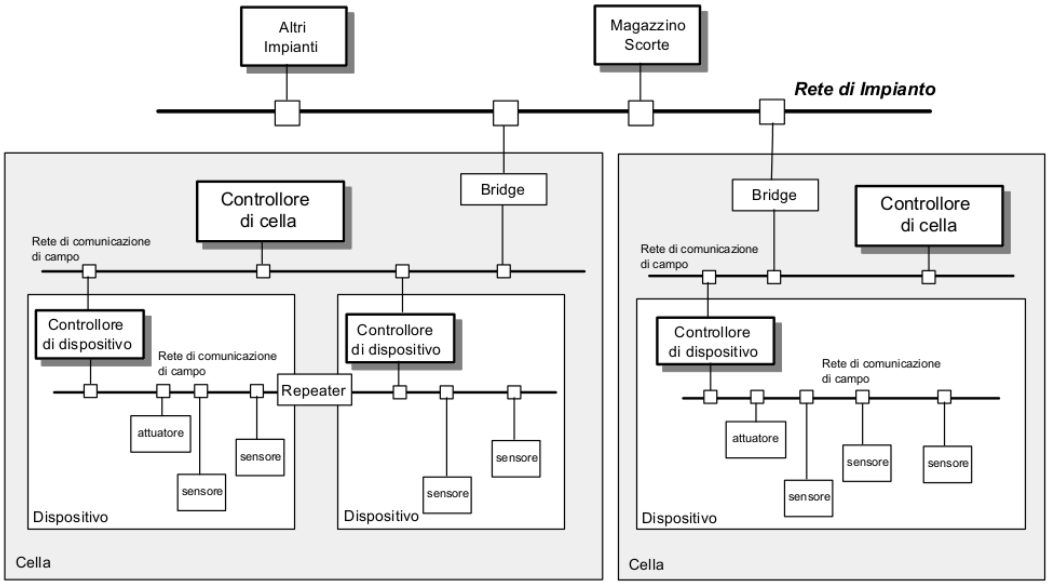


Figura 1.2 – Uso delle reti di comunicazione di campo negli impianti industriali [44].

Alcuni esempi di reti di comunicazione di campo in grado di assicurare performance deterministiche[1] sono WorldFIP [20], Profibus [22], P-Net [32], DeviceNet [12] e CAN [29], molte delle quali sono incluse nello standard internazionale IEC 61158 [30].

In alcune applicazioni, come quelle di motion control [21], le performance fornite da queste reti non sono sempre sufficienti a causa delle ridotta velocità di trasmissione che sono in grado di raggiungere. Per questo motivo, negli anni 90, risultò evidente che i fieldbus fossero troppo limitati, soprattutto se messi a confronto con le reti Ethernet che si stavano sviluppando fortemente in quegli anni al di fuori degli ambienti industriali e che erano in grado di raggiungere velocità decisamente più elevate.

1.1 IEEE 802.3 Ethernet

Lo standard IEEE 802.3, più comunemente chiamato Ethernet, specifica le caratteristiche fondamentali dei livelli Medium Access Control (MAC) e fisico del modello ISO/OSI per le reti locali LAN. Il meccanismo utilizzato nel livello MAC per gestire il canale di comunicazione è il Carrier Sense Multiple Access Collision Detection (CSMA/CD), in cui ogni stazione rimane sempre in ascolto sulla rete a cui è connessa (Carrier Sense significa proprio ascoltare il mezzo di trasmissione), attendendo l'eventuale ricezione di messaggi ad

[1] Con il termine 'deterministico' si intende la capacità di trasmettere un dato da una stazione ad un altra entro un tempo predefinito.

essa indicizzati.
La trasmissione di un messaggio richiede solamente la verifica che non ci sia attività sulla rete ed, essendo la rete ad accesso multiplo (Multiple Access), vi è la possibilità che due o più stazioni trovino la rete libera spedendo contemporaneamente dei dati, provocando una collisione, ed impedendo la corretta ricezione dei messaggi.
La prima stazione che si accorge della collisione (Collision Detection), invia un particolare messaggio a tutte le altre stazioni, bloccando ogni trasmissione fino a quando non saranno ristabilite le normali condizioni della rete. Per impedire che le stazioni che hanno provocato la collisione ritentino la trasmissione contemporaneamente, viene implementato un algoritmo che le costringe ad attendere un tempo casuale prima di ritrasmettere.
Nel livello fisico vengono stabilite le tecniche di trasmissione dei dati sul mezzo e le sue caratteristiche. Lo standard Ethernet prevede che sia presente un dispositivo hardware, detto Medium Attachment Unit (MAU), che permette il collegamento tra la stazione e la rete tramite un cavo denominato Attachment Unit Interface (AUI).
I diversi livelli fisici previsti dallo standard sono rappresentati utilizzando la simbologia seguente:

[velocità][tipo di trasmissione][massima lunghezza del segmento]

in cui la velocità di trasmissione è indicata in Mbps e la lunghezza massima di un segmento che connette due stazioni è espressa in centinaia di metri. La tipologia di trasmissione, infine, può essere in banda base, in cui tutta la banda del mezzo è utilizzata per una sola comunicazione, o in banda passante, in cui più canali di comunicazione possono essere definiti su uno stesso mezzo trasmissivo.
La semplicità degli algoritmi di accesso e di gestione utilizzati da questo standard portano ad avere dei ritardi quasi trascurabili in caso di basso carico e non viene utilizzata banda per gestire l'accesso. Come si vedrà in seguito questa caratteristica rende le reti Ethernet poco adatte alle applicazioni industriali, richiedendo l'implementazione di protocolli aggiuntivi per sottostare ai vincoli più stringenti presenti in questo ambito.

1.1.1 Ethernet con ripetitori

Un altro vantaggio di queste reti è che sono in grado di mettere in comunicazione un numero molto elevato di nodi, ma i limiti imposti sulla lungezza dei cavi rende necessaria l'introduzione di ripetitori per creare sistemi di grosse dimensioni.
La prima tipologia di ripetitore utilizzato è l'*hub*, cioè un dispositivo dotato di un certo numero di porte Ethernet che inoltra i dati in arrivo da una di queste a tutte le altre, quindi in

broadcast. Viene considerato un dispositivo di livello fisico, infatti non si occupa del tipo di dati in transito, ma trasmette solamente una replica dei segnali elettrici ricevuti.
La semplicità con cui viene gestito il traffico porta ad introdurre ritardi nelle trasmissioni spesso trascurabili, che rendono l'hub praticamente trasparente ai nodi che lo utilizzano per effettuare delle comunicazioni.
L'utilizzo degli hub, però, ha lo svantaggio di definire un unico dominio di collisione, cioè se due nodi collegati a due porte diverse effettuano una trasmissione contemporanea, si verifica una collisione ed è necessario inviare nuovamente i frame.
Una seconda tipologia di ripetitore che in parte riesce a risolvere il problema è il *bridge*, un elemento più sofisticato dell'hub che opera sui frame invece che sui segnali elettrici. In questo modo è in grado di estrapolare l'indirizzo di destinazione e di inviare il frame nel segmento corretto, che costituisce quindi un dominio di collisione separato. Questo dispositivo, che viene indicato di livello data link, permette di ridurre notevolmente il numero di collisioni. Ancora più sofisticati sono gli *switch*, che permettono il collegamento diretto dei singoli host, arrivando a ridurre il dominio di collisione ad un solo nodo. Così le uniche collisioni possibili avvengono in corrispondenza di più trasmissioni contemporanee verso uno stesso destinatario, che possono anche essere gestite da dei buffer associati alle porte memorizzando momentaneamente uno dei frame coinvolti nella collisione, per poi inoltrarlo al destinatario non appena il link si è liberato. Gli switch, perciò, permettono di eliminare quasi completamente le collisioni, che si verificano esclusivamente quando i buffer relativi ai link collegati sono pieni.
Il limite sulla dimensione della rete è strettamente legato alla presenza di collisioni, infatti sono imposti esclusivamente per assicurare che ogni nodo sia in grado di rilevarne il verificarsi prima di poter trasmettere nuovamente. Adottando gli switch, si ha la possibilità di togliere questi vincoli e di definire reti di dimensione qualsiasi.

1.2 Reti Ethernet Real-Time

Le reti Ethernet appena descritte non sono adatte ad essere utilizzate in sistemi di tipo real-time in cui è richiesto di poter compiere le trasmissioni entro un tempo prestabilito. Ciò è dovuto a due motivi, il primo riguarda la presenza delle collisioni, che avvengono in maniera non predicibile e che causano la ritrasmissione dei frame, impedendo di determinare entro quanto tempo possano arrivare a destinazione e di assicurare un comportamento deterministico della rete. Il secondo motivo è legato al fatto che non vi è la possibilità di assegnare alcun tipo di priorità alle trasmissioni, pertanto non si è in grado di distinguere

quali siano quelle che richiedono di rispettare delle deadline stringenti da quelle che, invece, possono attendere senza pregiudicare il funzionamento del sistema. L'elevata velocità di trasmissione raggiungibile, la possibilità di collegare un grande numero di nodi e la continua riduzione dei costi, però, hanno presto portato alla nascita di una serie di reti basate sullo standard IEEE 802.3. Queste reti, denominate Real-Time Ethernet (RTE), mantengono gran parte dei vantaggi delle reti Ethernet, ma introducono una serie di modifiche che permettono di assicurare anche il determinismo delle trasmissioni. Alcuni esempi sono ProfiNet [5], EtherNet/IP [9], Ethernet POWERLINK [33] e EtherCAT [18], molti dei quali sono stati introdotti negli standard internazionali IEC 61158 e IEC 61784 [31]. Nel prossimo capitolo verrà presentato il caso di Ethernet POWERLINK, una rete RTE isocrona, basata quindi sullo standard Ethernet, in grado di assicurare trasmissioni deterministiche e caratterizzato da un comportamento ripetitivo nel tempo.

Capitolo 2

Ethernet POWERLINK

EPL è un protocollo di comunicazione per le reti RTE che estende lo standard Ethernet IEEE 802.3 con un meccanismo per trasferire dati con un sincronismo preciso e con temporizzazioni predicibili. Un aspetto molto importante di questo nuovo protocollo è che non modifica le caratteristiche di base dello standard Ethernet, assicurando una completa compatibilità con l'hardware già esistente.
Una sua prima versione è stata sviluppata nel Novembre del 2001 dalla Bernecker & Rainer Industrie-Elektronik (B&R) [33], nel 2002 è stato fondato L'EPL Standardization Group (EPSG) che, nel Novembre del 2003, ha pubblicato una seconda versione [33] basata sui meccanismi definiti in CANopen [19] successivamente entrata a far parte dello standard IEC 61784 come protocollo di comunicazione.

EPL permette di raggiungere i seguenti obbiettivi:

1. rispettare le deadline relative alla trasmissione di dati critici all'interno di cicli isocroni;
2. sincronizzare i nodi della rete con grande precisione;
3. trasmettere dati con deadline meno stringenti o assenti all'interno di cicli asincroni, in cui è possibile utilizzare protocolli di comunicazione come TCP, UDP, HTTP, FTP, etc..

Per evitare che vi possano essere interferenze durante la trasmissione di dati isocroni e asincroni, in EPL vengono stabiliti degli slot temporali ben precisi che vengono dedicati rispettivamente alle due tipologie di trasmissione. Questo meccanismo, rappresentato in figura 2.1, viene chiamato Slot Communication Network Management (SCNM) e viene

attuato da un particolare nodo della rete, detto appunto Managing Node (MN). Tutti gli altri nodi della rete che vengono gestiti dal MN sono invece chiamati Controlled Node (CN).

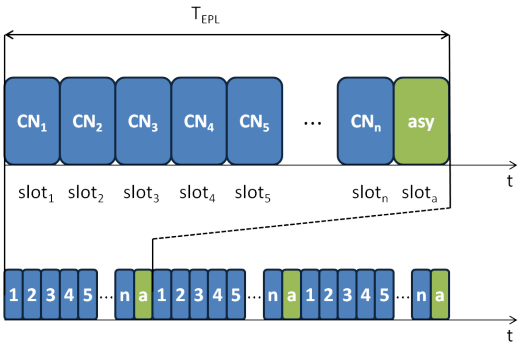


Figura 2.1 – SCNM.

Come si può notare il funzionamento è di tipo ciclico, ogni ciclo viene suddiviso in vari slot temporali che vengono assegnati dal MN ai CN per permettergli l'accesso temporaneo alla rete. Questo meccanismo assicura una gestione molto rigorosa delle trasmissioni, che con un dimensionamento corretto del ciclo e degli slot temporali porta ad avere il comportamento deterministico dello standard.

2.1 Architettura della rete

L'architettura protocollare di un nodo EPL, rappresentata in figura 2.2, si basa sul modello ISO/OSI, in cui la struttura logica è composta da sette livelli che forniscono tutti i servizi necessari. Gran parte delle caratteristiche di questa architettura sono del tutto equivalenti a quella dello standard Ethernet, rendendo le due tipologie di reti perfettamente compatibili.

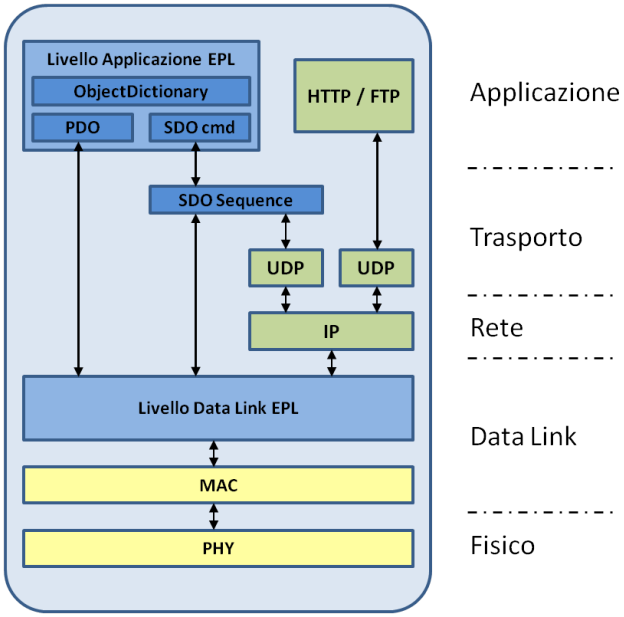


Figura 2.2 – Architettura della rete EPL derivata dal modello ISO/OSI.

2.1.1 Livello Fisico

Il livello fisico delle reti EPL corrisponde esattamente a quello definito per le reti Ethernet, in particolare viene adottata la rete 100BASE-X, con una trasmissione di tipo half-duplex e con i comuni connettori RJ-45 o M12.
I vari nodi della rete possono essere collegati tramite dei comuni hub o switch, anche se è raccomandato di utilizzare i primi [19], che introducono un ritardo inferiore nelle trasmissioni, garantendo di mantenerlo inferiore a 460 ns e di mantenere il jitter[1] dei frame sotto i 70 ns. Lo sviluppo di nuove tecnologie per le reti Ethernet, però, ha fatto rapidamente scomparire dal mercato gli hub, provocando un forte aumento del loro costo e portando, in alcuni casi, a preferire gli switch, che richiedono in fase di configurazione di tenere conto degli ulteriori ritardi introdotti.
Dal momento che POWERLINK non causa collisioni non è necessario applicare il limite Round Trip Time (RTT)[2] delle comuni reti Ethernet ed è possibile adottare topologie lineari con un numero molto elevato di nodi, tipiche delle applicazioni industriali.

Si possono anche adottare topologie a stella, ad albero o ibride, come quella rappresentata in figura 2.3, ma è sempre necessario tenere conto della massima distanza tra il MN e i

[1] Con jitter si intende la variazione temporale rispetto al ciclo precedente
[2] L'RTT è una misura del tempo impiegato da un pacchetto di dimensione trascurabile per viaggiare da un computer della rete ad un altro e tornare indietro

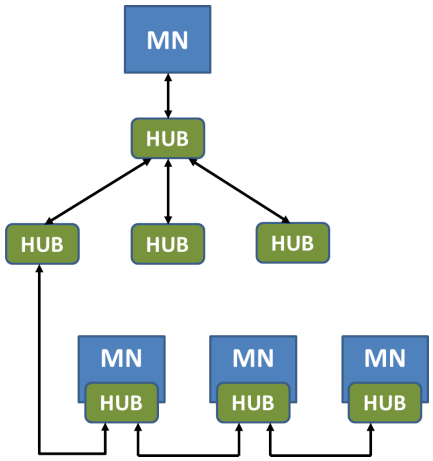


Figura 2.3 – Esempio di una rete EPL con topologia ibrida stella-lineare.

CN per non superare il limite sul jitter. É dunque vantaggioso posizionare il MN al centro della rete per ridurre al minimo tale distanza.

2.1.2 Livello Data Link

Nel livello Data Link di EPL vengono definite due modalità fondamentali di comunicazione:

POWERLINK Mode è l'unica modalità in cui sia possibile trasmettere frame isocroni, cioè i frame necessari ad avere un comportamento real-time; oltre a questi frame è anche possibile trasmettere frame asincroni, utilizzati per comunicazioni che non richiedono comportamenti real-time;

Basic Ethernet Mode è la modalità in cui è possibile trasmettere normali frame Ethernet secondo le regole definite dallo standard IEEE 802.3. Ciò permette di far interagire al meglio dispositivi che implementano lo standard EPL con i normali dispositivi Ethernet, ma è consigliabile non utilizzare questa modalità quando i nodi fanno parte di un sistema di automazione, perchè la comunicazione diviene non deterministica per la possibile presenza di collisioni e non è più assicurato un comportamento di tipo real-time;

Per quanto riguarda la modalità POWERLINK il determinismo è ottenuto grazie ad uno scambio di messaggi pianificato, secondo il quale i messaggi trasmessi tra i vari nodi della rete sono raggruppati in cicli, suddivisi a loro volta in due fasi, una in cui vengono gestite le richieste di tipo real-time, l'altra in cui vengono effettuate le trasmissioni che non richiedono di rispettare deadline stringenti.

Il nodo che gestisce questo comportamento ciclico è il Managing Node (MN), mentre i nodi che partecipano alle fasi di questo ciclo per trasmettere sono i Controlled Node (CN).

POWERLINK MN

Il MN è l'unico nodo attivo della rete, è in grado di compiere trasmissioni indipendentemente dalle richieste degli altri nodi e si occupa di assicurare il determinismo della rete, accedendo in maniera ciclica ai CN e gestendo rigorosamente tutte le trasmissioni.
Esso, infatti, trasmette in maniera unicast dei frame di sincronizzazione ai nodi CN configurati, che a loro volta, possono rispondere trasmettendo i propri dati a tutti i nodi con dei frame multicast. Per garantire che non vi possano essere collisioni è necessario che vi sia un solo MN attivo contemporaneamente.

POWERLINK CN

I CN sono dei nodi passivi, che trasmettono solamente quando gli viene richiesto dal MN. Possono essere distinti in due tipologie differenti:

CN isocroni ognuno di questi CN riceve ciclicamente un frame unicast dal MN, a cui rispondono con una trasmissione multicast, dove vengono inseriti i dati ti tipo real-time. L'accesso a questi nodi avviene ogni n cicli, con $n \geq 1$;

CN solamente asincroni in questo caso l'accesso non avviene in maniera ciclica e possono comunicare con gli altri nodi solo nella fase non deterministica del ciclo.

2.1.3 Livello Applicazione

Per quanto riguarda il livello applicazione, viene adottato lo standard CANopen, per questo POWERLINK viene anche chiamato *CANopen over Ethernet*. Al suo interno vengono definiti due protocolli, il Service Data Object (SDO), utilizzato da un nodo per accedere ad un dispositivo remoto, e il Process Data Object (PDO), necessario a trasmettere dati rilevanti in broadcast.

2.1.4 Modello di comunicazione

Nel modello di comunicazione vengono specificate le differenti tipologie di oggetti e servizi di comunicazione disponibili e le modalità di trigger della trasmissione dei frame.
Le tipologie di relazioni che vi possono essere tra le varie stazioni della rete sono rappresentate nelle figure 2.4, 2.5 e 2.6 , dove si distinguono le comunicazioni Master/Slave, in

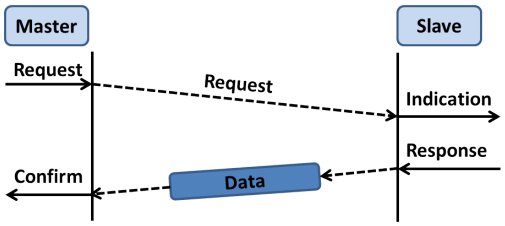


Figura 2.4 – Comunicazione Master/Slave.

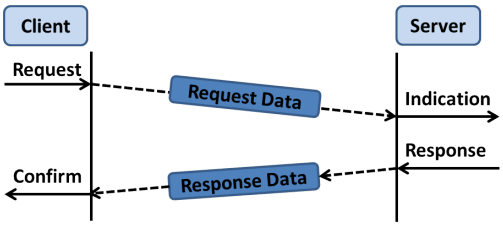


Figura 2.5 – Comunicazione Client/Server.

cui tutti i nodi della rete fanno riferimento ad una singola stazione che fornisce loro i servizi di cui hanno bisogno, le comunicazioni Client/Server, in cui un singolo nodo fa riferimento ad un altro nodo che si occupa di gestirne le richieste, ed, infine, le comunicazioni di tipo Produttore/Consumatore, in cui le stazioni che richiedono un servizio si mettono in attesa della risposta di una unica stazione, detta appunto produttore.

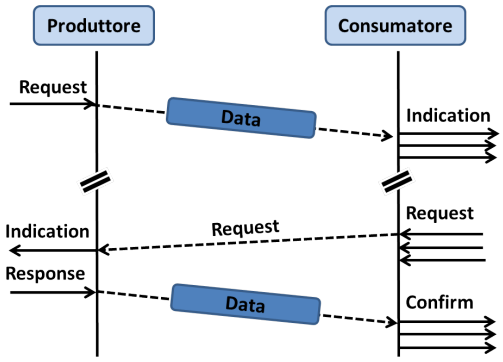


Figura 2.6 – Comunicazione Produttore/Consumatore.

La struttura delle reti industriali, in cui vi è un controllore centrale ed una serie di dispositivi da esso controllati, ed in cui si deve assicurare un comportamento real-time del sistema, porta a dover utilizzare tutte e tre queste tipologie di comunicazione all'interno di un singolo ciclo di lavoro. Ad esempio, durante la configurazione della rete, i nodi trasmettono la richiesta di accedere alla rete ad una stazione centrale, utilizzando quindi una comunicazione di tipo Master/Slave. Nella fase iniziale del ciclo, invece, viene effettuato un polling dei nodi della rete, che richiede una trasmissione confermata dei frame, e che quindi necessita di implementare un modello di tipo Produttore/Consumatore. Infine le richieste di poter trasmettere frame asincroni da parte delle varie stazioni viene gestita con una relazione di tipo Client/Server.

Il modello di una rete EPL comprende anche una astrazione dei dispositivi collegati, che possono essere rappresentati come in figura 2.7, in cui vengono definite le tre parti fondamentali che li compongono:

Communication: provvede a fornire degli oggetti di comunicazione e delle funzionalità necessari a trasmetterli lungo la rete;

Object Dictionary: contiene la lista di tutti gli oggetti utilizzati da una particolare stazione, che vengono definiti PDO (Process Data Object). In particolare vengono memorizzati quelli trasmessi nella rete (TPDO) e quelli ricevuti (RPDO), che, come si vedrà, vengono inseriti in frame denominati PReq e PRes;

Application: contiene le funzionalità del dispositivo per interagire con i vari processi.

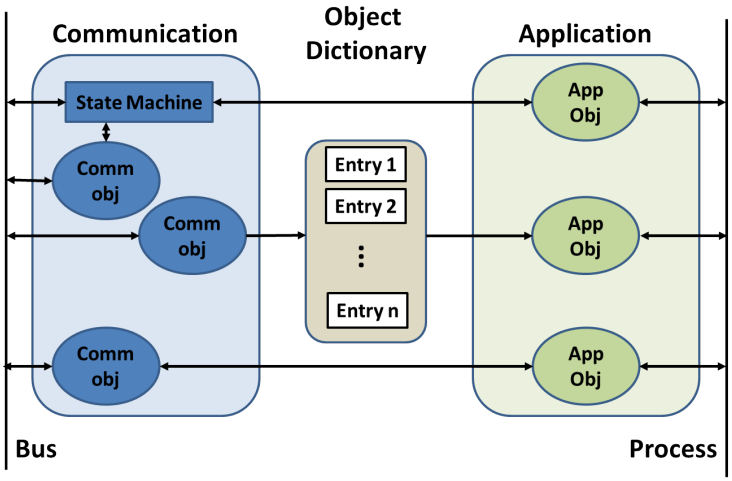


Figura 2.7 – Modello di un dispositivo EPL.

Ad ogni dispositivo EPL collegato alla rete viene assegnato un indirizzo che permette di identificarlo univocamente. Al dispositivo centrale, cioè il MN è assegnato il valore 240; agli altri nodi della rete, cioè i nodi CN possono essere assegnati gli indirizzi tra 1 e 239, mentre gli altri valori disponibili sono utilizzati come riassunto in tabella 2.1.

ID nodo EPL	Descrizione
0	Non valido
1..239	CN
240	MN
241..250	Riservati
251	Utilizzato dal nodo per auto indirizzarsi
252	Nodo EPL fittizio
253	Diagnostica
254	Router
255	Broadcast

Tabella 2.1 – Significato degli indirizzi EPL.

2.1.5 Struttura dei frame EPL

La struttura basilare di un frame EPL è costituita da 5 campi:

- Reserved (1 bit)
- Message Type (7 bit)
- Destination node address (1 byte)
- Source node address (1 byte)
- Payload (n byte)

Come si vede in figura 2.8 il frame viene poi incapsulato all'interno di un normale frame Ethernet, impostando come tipo di frame il numero esadecimale $88AB_h$ ed introducendo altri 7 byte relativi al preambolo, 1 byte di Start Frame Delimiter (SFD), 14 byte relativi all'header e 4 byte finali necessari al controllo degli errori (CRC32).

In tabella 2.2 sono presentati le differenti tipologie di frame EPL che possono essere utilizzate, il tipo di trasmissione corrispondente e il valore da assegnare al campo Message Type.

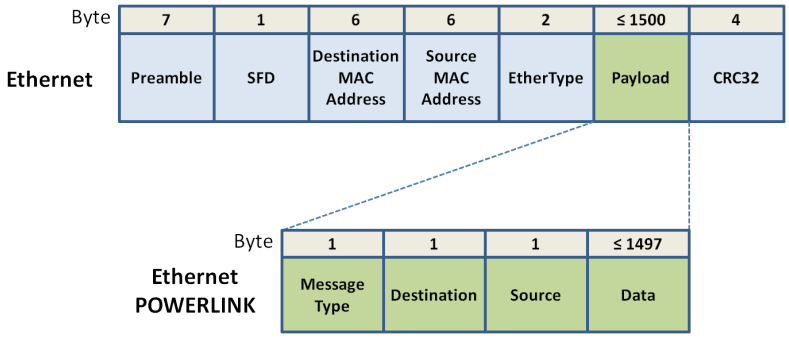


Figura 2.8 – Struttura del frame EPL.

Sempre in figura 2.8 si può notare la struttura di tali frame, in particolare la dimensione dei frame Start of Cycle (SoC) e Start of Asynchronous (SoA) è fissata a 46 byte, mentre i frame Poll Request (PReq) e Poll Response (PRes) hanno una dimensione che può variare tra un minimo di 46 byte ad un massimo di 1500. Per semplicità, di seguito, si farà riferimento esclusivamente a frame della dimensione fissata di 46 byte, a cui vanno poi aggiunti i byte dovuti all'incapsulamento nel frame Ethernet, che porta la dimensione complessiva dei pacchetti che transitano nella rete a 72 byte.

Tipo di messaggio EPL	Tipo di trasmissione MAC	EPL Message Type
Start of Cycle (SoC)	Multicast	01_h
Poll Request (PReq)	Unicast	03_h
Poll Response (PRes)	Multicast	04_h
Start of Asynchronous (SoA)	Multicast	05_h
Asynchronous Send (ASnd)	Multicast	06_h

Tabella 2.2 – Tipologie di messaggi EPL, relativa modalità di trasmissione e valore del campo Message Type.

2.2 Ciclo POWERLINK

Il ciclo POWERLINK viene gestito dal MN e, come già accennato, si divide in tre fasi:

- Fase Isocrona;

- Fase Asincrona;
- Fase di Idle.

É essenziale che l'inizio di tale ciclo sia il più possibile esatto, cioè privo di jitter, pertanto si dovrà dimensionare il suo periodo in modo da assicurare che tutte le trasmissioni delle fasi che lo compongono possano essere effettuate senza il rischio di sforare nel ciclo successivo.

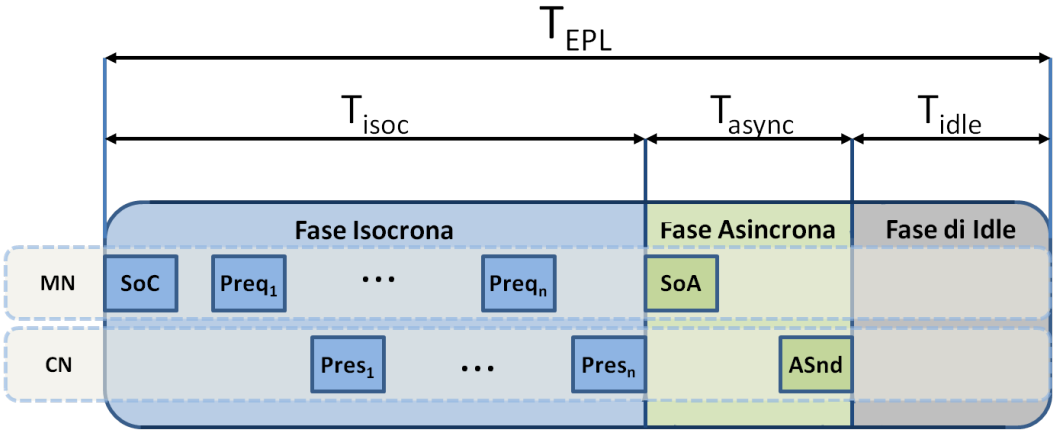


Figura 2.9 – Rappresentazione delle fasi che compongono il ciclo EPL.

Il ciclo POWERLINK è schematizzato in figura 2.9 e nelle sezioni successive verranno descritte in maniera approfondita le tre fasi principali.

2.2.1 Fase Isocrona

Il trasferimento isocrono di dati tra i nodi deve avvenire in maniera ciclica, deve cioè essere ripetuto con un periodo pari o multiplo a quello del ciclo POWERLINK. All'inizio di tale ciclo il MN trasmette a tutti i nodi della rete un frame denominato Start of Cycle (SoC), che è l'unico frame generato in maniera periodica, mentre tutti gli altri devono essere generati da eventi controllati.
Successivamente vengono trasmessi PReq a tutti i nodi attivi e configurati, sotto forma di frame Ethernet unicast, che a loro volta possono rispondere tramite un frame Ethernet multicast PRes. Nella risposta PRes il CN può anche includere la richiesta di una successiva trasmissione asincrona, che verrà gestita opportunamente dal MN secondo il meccanimo descritto in sezione 2.2.2.
Il MN esegue un polling sui CN interrogandoli iterativamente e attendendo la risposta del

nodo corrente prima di procedere con quello successivo. Per evitare che possa rimanere indefinitamente in attesa di questa risposta o che la ricezione avvenga con un ritardo tale da pregiudicare il comportamento real-time del sistema, viene avviato un timer, allo scadere del quale si procede in ogni caso alla trasmissione del prossimo frame PRes.
Una volta che questa operazione è stata svolta per tutti i nodi CN isocroni attivi e configurati, il MN può trasmettere un frame PRes a tutti i nodi, necessario alla trasmissione di dati rilevanti per gruppi di CN.
La durata della fase isocrona può essere calcolata dalla trasmissione del frame SoC all'inizio della successiva fase asincrona ed influenza in maniera significativa la durata dell'intero ciclo POWERLINK. Nella configurazione del suo periodo, infatti, si deve tenere conto del tempo necessario ad accedere ai vari nodi configurati tramite lo scambio di frame PReq e PRes, che portano ad avere una durata della fase isocrona proporzionale al numero di CN attivi.

Multiplexed Timeslots

In POWERLINK vi è la possibilità di scegliere quali CN debbano essere interrogati ad ogni ripetizione della fase isocrona e quali debbano essere interpellati solo in alcune di esse. Per fare ciò vengono definite due differenti classi di comunicazione per i CN:

Continua: i dati continui vengono scambiati ad ogni ciclo POWERLINK;

Multiplata: i dati multiplati non vengono trasmessi ad ogni ciclo, ma solamente per un ciclo multiplato, il cui periodo è multiplo di quello del ciclo POWERLINK.

Questa soluzione ha un grosso vantaggio, infatti, permette di accedere ad un numero molto elevato di CN senza prolungare troppo il ciclo POWERLINK, adattando la frequenza di poll dei singoli CN in base alle necessità.

Ovviamente le due tipologie di accesso possono essere compiute in maniera parallela durante il ciclo, come avviene nell'esempio rappresentato in figura 2.10, in cui il CN numero 3 accede alla fase isocrona ogni due cicli EPL. I nodi che non vengono interrogati ad ogni ciclo, possono comunque monitorare i dati trasmessi, visto che il frame PRes proveniente dagli altri nodi è trasmesso in maniera broadcast.

Per rendere più chiara la necessità di tale meccanismo basti pensare ad una applicazione di motion control in cui i dati multiplati possono essere utilizzati da un numero elevato di assi slave per ricevere le posizioni da degli assi master. Gli assi master vengono configurati per comunicare continuamente, trasmettendo i dati ad ogni ciclo agli assi slave di monitoraggio, che prendono parte alle comunicazione in un ciclo più lento.

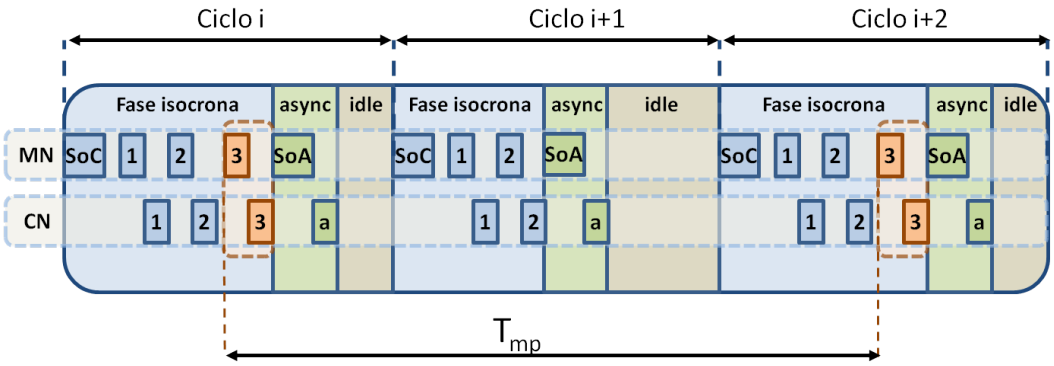


Figura 2.10 – Esempio della coesistenza tra CN continui e multiplati.

La durata delle finestre temporali assegnate alle trasmissioni multiplate viene generalmente scelta come somma dei tempi massimi necessari allo scambio dei pacchetti PReq-PRes tra il MN e i vari CN assegnati alla finestra stessa. Inoltre il MN si occupa di distribuire i nodi nei vari cicli EPL, in modo tale che participino ad essi con la periodicità desiderata, senza però accumularsi in un unico ciclo e, quindi, senza incrementare troppo il carico della rete.

2.2.2 Fase Asincrona

Durante la fase asincrona del ciclo vengono scambiati i pacchetti che non hanno vincoli di tipo real-time. L'accesso alla rete può assere garantito ad un solo CN o al MN per ogni ciclo ed è possibile trasmettere un singolo messaggio asincrono. Esistono due diverse tipologie di messaggi che possono essere trasmessi in questa fase:

- il frame ASnd che utilizza lo schema di indirizzamento di POWERLINK e può essere trasmesso ad un nodo qualsiasi o broadcast;
- un messaggio di tipo *Legacy Ethernet*, cioè appartenente alla categoria 100BASE-T.

Il MN avverte i CN che la fase asincrona sta per iniziare trasmettendo il frame SoA in maniera broadcast ed includendo in esso un campo contenente l'identificativo del CN che ha il permesso di trasmettere un singolo messaggio.
Se non vi è alcuna richiesta pendente per la trasmissione di messaggi asincroni, il MN può inviare il frame SoA senza definire quale nodo possa trasmettere in questa fase, che quindi terminerà immediatamente. Teoricamente la fase asincrona termina in corrispondeza della trasmissione del frame ASnd e, per quanto riguarda il CN designato avviene effettivamente così. Per tutti gli altri nodi della rete, però, l'effettivo istante in cui non si effettuano più

trasmissioni, corrisponde alla ricezione del frame SoA nel caso in cui sia stato richiesto di partecipare alla fase asincrona e alla trasmissione del proprio frame PRes nel caso dei nodi che partecipano solo alla fase isocrona.
Pertanto la fase di idle vera e propria ha inizio dopo la trasmissione del frame ASnd, ma sarà utile tenere conto dell'istante in cui i nodi terminano effettivamente le trasmissioni quando si cercherà di implementare degli algoritmi di risparmio energetico.

Scheduling asincrono

Come già accennato nella sezione 2.2.1, durante la fase isocrona i CN possono inoltrare la richiesta di una trasmissione asincrona, che viene inserita in una coda gestita dal MN. Quest'ultimo si occupa di decidere quale di queste richieste debba essere servita durante il ciclo corrente, assicurandosi che nessuna di esse possa rimanere in attesa indefinitamente, anche nel caso di un carico elevato della rete.
Per decidere quale richiesta debba essere servita vengono gestite quattro code differenti, che sono definite nel modo seguente:

- richieste generiche di trasmissione da parte del MN;
- *identRequest* trasmessi dal MN per identificare i CN;
- *statusRequest* trasmessi dal MN per gestire CN non interrogati ciclicamente nella fase isocrona;
- *transmitRequest* trasmesse dai CN nella fase isocrona per richiedere l'accesso alla fase asincrona.

In ognuna di queste richieste, il MN o i CN possono specificare una priorità scegliendo tra otto livelli differenti, grazie alla quale il MN è in grado di adottare uno scheduling a priorità per decidere l'ordine di trasmissione dei messaggi asincroni.

2.2.3 Fase di Idle

La fase di idle corrisponde all'intervallo temporale tra la fine della fase asincrona e l'inizio del successivo ciclo di POWERLINK.
Come si è già visto, la durata delle due fasi precedenti non è sempre costante, sia perchè alcuni dei nodi potrebbero adottare il multiplexed timeslot, sia perchè non sempre viene effettuata una trasmissione asincrona. Questo, ovviamente, si ripercuote su una variabilità della fase di idle, che non ha sempre la stessa durata in ogni ciclo.

Dal momento che proprio nella fase di idle si cercherà di implementare alcuni algoritmi per ridurre il consumo di energia, sarà necessario tenere conto della sua variabilità, cercando anche di sfruttare il fatto che, l'effettivo istante in cui i nodi terminano di trasmettere e potrebbero iniziare a consumare meno, può essere anticipato rispetto all'inizio della fase di idle.

2.2.4 Temporizzazione del ciclo di POWERLINK

Il periodo del ciclo EPL, che di seguito verrà indicato con T_{EPL}, rappresenta il periodo minimo di campionamento dei dati real-time poichè corrisponde all'intervallo temporale minimo tra due consecutive trasmissioni nelle fasi isocrone di uno stesso CN.
Di conseguenza, la scelta della sua durata, che viene impostata in fase di configurazione della rete, deve essere fatta tenendo conto dei numerosi fattori che potrebbero causare un ritardo nelle trasmissioni e una durata reale del ciclo superiore al periodo con la conseguente impossibilità di mantenere il determinismo richiesto.

Il periodo del ciclo EPL può essere ovviamente suddiviso nella durata delle tre fasi che lo compongono:

$$T_{EPL} = t_{isoc} + t_{async} + t_{idle}.$$

Come si può notare dalla figura 2.11, la durata della fase isocrona è influenzata da vari fattori, in particolare:

t_{frame} tempo necessario alla generazione dei frame trasmessi (SoC, PReq, PRes e SoA);

t_{trasm} tempo di trasmissione dei frame attraverso la rete;

t_{el} tempi di elaborazione introdotti dai nodi della rete;

N_{CN} numero dei CN coinvolti nel ciclo, che determina quante iterazioni siano necessarie a completare il polling.

La durata della fase asincrona dipende essenzialmente dagli stessi fattori, eccezion fatta per il numero di nodi, che non ha effetti su t_{async}, infatti un solo nodo ha accesso alla rete per trasmettere un messaggio asincrono. Infine, la fase di idle viene ottenuta sottraendo al periodo del ciclo EPL, definito a priori, la durata delle due fasi appena descritte.
Per poter dimensionare correttamente le tempistiche del ciclo, quindi, diviene essenziale ottenere una stima il più possibile affidabile del tempo necessario a compiere l'intero ciclo

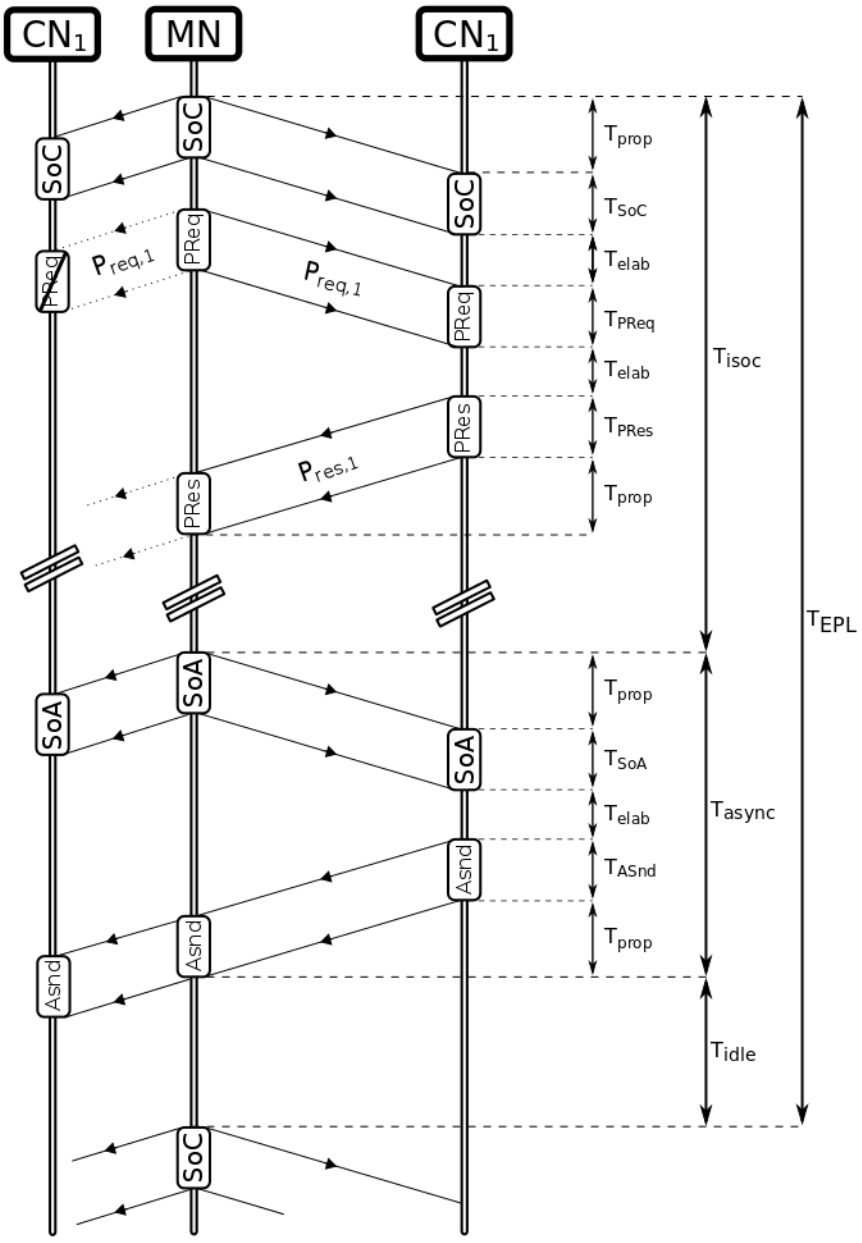


Figura 2.11 – Scambio di frame tra il MN e i CN con relativa temporizzazione.

di polling, che costituisce la fase isocrona e che, all'aumentare del numero di nodi, diviene il fattore dominante dell'intero ciclo.

Come si è visto nella sezione 2.2.1, ad ogni iterazione del ciclo di polling il MN rimane in attesa di un tempo massimo oltre il quale si procede con la iterazione successiva. Questo timeout, che di seguito verrà indicato con $t_{PReq,PRes}$, rappresenta la durata massima della frazione di fase isocrona associata ad ogni CN, pertanto il tempo dedicato al polling non potrà superare il valore:

$$t_{poll} = N_{CN} \cdot t_{PReq,PRes}.$$

Dal momento che il numero di CN è noto, il problema si sposta sul determinare un tempo $t_{PReq,PRes}$ che sia sufficientemente grande da permettere a tutti i CN di ricevere il frame PReq, di preparare il frame di risposta PRes e di trasmetterlo al MN, ma che deve anche essere abbastanza breve da assicurare che il MN non possa rimanere troppo a lungo in attesa. Secondo lo standard [19], tale tempo può essere calcolato come segue:

$$t_{PReq,PRes} = 2 \cdot t_{elab} + 2 \cdot t_{trasm}.$$

Esso è composto dalla somma dal tempo necessario al CN per elaborare il pacchetto PReq ricevuto e a generare il frame PRes corrispondente, più quello necessario alla trasmissione di questi due frame e all'elaborazione del frame PRes da parte del MN. La componente t_{elab} dipende dall'hardware utilizzato, quindi si deve determinarlo tramite prove sperimentali, mentre il tempo relativo alla trasmissione dei frame, che costituisce la componente più influente, può essere stimato con ottima approssimazione, visto che dipende dalla dimensione del frame trasmesso e dalla velocità di trasmissione del link.
Siano v la velocità di trasmissione, espressa in *Mbps*, e d la dimensione del frame che viene trasmesso, espressa in byte. Allora il tempo di trasmissione è dato da:

$$t_{trasm} = \frac{d \cdot 8}{v}.$$

La dimensione dei frame in una rete EPL varia da un minimo di 72 byte ad un massimo di 1526 byte [3], da cui si ricavano i tempi di trasmissione corrispondenti alle varie velocità prese in esame, che sono riassunti in tabella 2.3.

Una volta fissato il timeout si possono presentare tre situazioni differenti durante il polling della fase isocrona:

[3] La dimensione del solo frame EPL varia da 46 byte a 1500 byte, ma è necessario aggiungere 26 byte dovuti all'incapsulamento nel frame Ethernet

Velocità link EPL [Mbps]	Tempo di trasmissione (t_{trasm}) [μs]
	d = 72 byte
100	5.76
1000	0.576
10000	0.0576

Tabella 2.3 – Tempi di trasmissino di un frame EPL di dimensione 72 byte al variare della velocità del link.

- frame PRes ricevuto in tempo: il timeout corrente viene bloccato e si procede con il prossimo CN;
- perdita del frame PRes: è causato da una perdita del link o da errori di trasmissione. Se ciò avviene per un numero di volte consecutive superiore ad un limite prefissato, il CN coinvolto viene considerato inattivo, viene rimosso dalla fase isocrona ed eventualmente gli viene trasmessa una richiesta di reset;
- frame PRes ricevuto in ritardo: avviene quando un frame PRes proveniente da un CN, viene ricevuto in corrispondenza della iterazione relativa ad un altro CN, a causa di ritardi di trasmissione o di un errato dimensionamento del timeout. Il frame viene scartato e il CN interessato viene gestito come nel caso precedente.

Un'altra situazione in cui non vengono rispettati i limiti temporali del ciclo, rappresentata in figura 2.12, avviene in corrispondenza di un ritardo della trasmissione del frame SoC o della sua perdita, causata da collisioni con altri frame Ethernet o con il frame ASnd relativo alla fase asincrona precedente.

La presenza di questo tipo di ritardi viene considerato un errore di configurazione e provoca la sospensione del ciclo corrente. Questa situazione di errore viene notificata e monitorata grazie ad un contatore, che viene incrementato ogni volta che viene saltato un ciclo EPL e che permette di verificare se sia necessaria una nuova configurazione dei suoi parametri temporali.

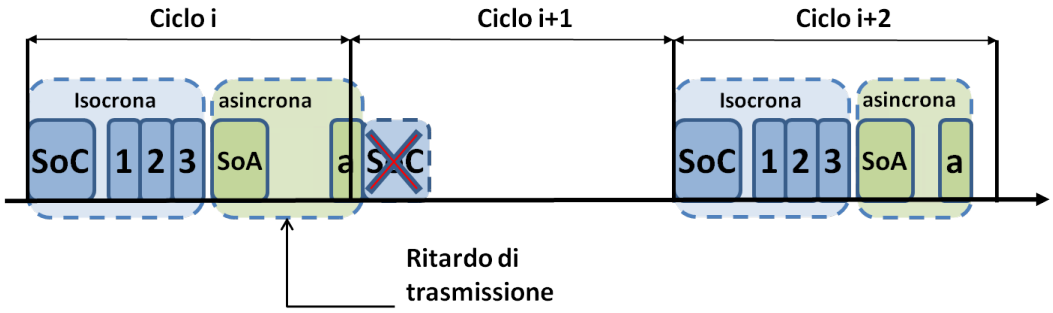


Figura 2.12 – Esempio della sospensione di un ciclo EPL a causa del ritardo di trasmissione del pacchetto ASnd.

Capitolo 3

Algoritmi per il Risparmio Energetico su Reti Ethernet

Il crescente sviluppo dei sistemi di comunicazione ha presto focalizzato l'attenzione su quei dispositivi di trasmissione e ricezione che, presi singolarmente hanno un consumo decisamente ridotto, ma, considerandoli da un punto di vista globale, evidenziano uno spreco di energia tutt'altro che trascurabile.
Uno degli esempi più evidenti è Internet, a cui sono collegati miliardi di nodi in tutto il mondo, portandolo a divenire uno dei maggiori consumatori di energia con un impatto economico ed ambientale piuttosto rilevante.

Gran parte dei nodi connessi ad Internet o collegati all'interno di una rete sfruttano la tecnologia Ethernet come mezzo di comunicazione, pertanto sono state fatte numerose stime sul consumo energetico dei dispositivi dotati di una o più schede Network Interface Controller (NIC) e sulle possibilità di risparmiare energia.
In tabella 3.1 sono riassunti i dati relativi al numero approssimativo dei principali dispositivi che permettono il collegamento di reti Ethernet e del corrispondente consumo energetico, ricavati dalla stima compiuta negli Stati Uniti nel 2000 dalla Annual Electricity Consumption (AEC) [42]. Il costo dell'energia di quegli anni, negli Stati Uniti, era mediamente di 85 centesimi di dollaro per ogni kWh [6], quindi, con un consumo complessivo dell'ordine dei 6 TWh annui, il costo corrispondente raggiungeva i 500 milioni di dollari.

Altro aspetto di cui tenere conto è, ovviamente, l'impatto ambientale; infatti, con una emissione di circa 500 grammi di CO_2 per ogni kWh generato, una produzione di energia di tale entità può portare all'emissione di circa 0.75 milioni di tonnellate di CO_2 [42], senza contare il consumo di materie prime e combustibili. I dati analizzati fanno riferimento alla

situazione negli Stati Uniti di più di dieci anni fa, quando esistevano solamente reti a 10 Mbps e 100 Mbps, mentre le reti moderne a 1 GB erano appena state presentate e non si erano ancora diffuse.
L'aumento della velocità delle trasmissioni ha presto portato ad un incremento del consumo dei singoli dispositivi; infatti un link 10 Mbps o a 100 Mbps consuma circa 500 mW, uno a 1000 Mbps circa 1.4 W e un link a 10 GBps consuma circa 12 W [36, 3].
Inoltre, il costo dei dispositivi in grado connettersi ad una rete Ethernet è diminuito drasticamente, favorendone la diffusione in ambiente domestico e lavorativo, ma anche in ambiente industriale, dove la nascita delle reti RTE ha contribuito ulteriormente ad aumentare il numero di nodi Ethernet, e quindi il loro consumo complessivo.

Una seconda stima, compiuta nel 2005 [26], ha evidenziato che il consumo di questi dispositivi negli USA nell'arco di un anno, era aumentato di ben 1 TWh rispetto a quello calcolato nel 2000.

Dispositivo	Numero di dispositivi	Consumo [TWh]
hub	$93.5 \cdot 10^6$	1.6
Switch LAN	95000	3.2
Router	3257	0.15

Tabella 3.1 – Stima del numero di alcuni dispositivi Ethernet e del loro consumo negli Stati Uniti nel 2000 riferita al AEC [6].

La crescente attenzione su questo argomento ha presto portato alla luce un grande problema delle reti Ethernet, cioè la totale assenza di qualsiasi criterio di efficienza energetica. I trasmettitori ed i ricevitori collegati ad un link Ethernet, infatti, sono stati progettati per funzionare in maniera continua, cioè mantengono praticamente lo stesso stato operativo sia nel caso in cui sia necessario trasmettere delle informazioni lungo il link, sia nel caso in cui non sia presente alcun tipo di traffico [23].
Come si può vedere dalla tabella 3.2 il consumo energetico di un link Ethernet quando è presente traffico in entrambe le direzioni varia di poco rispetto al caso in cui non vi siano trasferimenti [36, 3], cioè quando il link si trova nello stato di idle.

Questo fatto porta ovviamente ad uno spreco di energia, soprattuto considerando che, in alcuni casi, la percentuale di utilizzo di molti dispositivi Ethernet come quelli collegati ai PC domestici si aggira attorno al 5% [15, 38] .
Da ciò si deduce che vi sono grandi opportunità per ridurre sensibilmente il consumo legato

Protocollo	Consumo [W]	
	Attivo	Idle
10BASE-T	0.504	0.334
100BASE-TX	0.388	0.320
1000BASE-T	0.781	0.777
10GBASE-T	8.2	7.9

Tabella 3.2 – Consumo energetico delle reti Ethernet sotto carico e in idle [13].

alle reti Ethernet, cercando di minimizzare il più possibile l'energia consumata quando i nodi si trovano in idle, ovvero quando non è necessario mantenere interamente le funzionalità del ricevitore e del trasmettitore. A partire dalle stime presentate e dalle considerazioni appena fatte sono state proposte numerose soluzioni al problema, alcune delle quali basate sull'idea di modificare la velocità di trasmissione del link durante la fase di idle e altre in cui si cerca di disattivare alcuni componenti dei dispositivi quando non sono strettamente necessari.
Di seguito verrano elencate alcune di queste proposte, che hanno recentemente portato alla definizione di uno standard vero e proprio per le reti Ethernet.

3.1 Cambio di velocità con auto-negoziazione

La auto-negoziazione è una tecnologia introdotta da National Semiconductor nel gruppo di lavoro IEEE 802.3u 100BASE-T con lo scopo di risolvere i problemi di compatibilità derivanti dall'uso di dispositivi in grado di operare a velocità differenti [17]. La semplicità di utilizzo, i costi molto ridotti, la flessibilità e l'adattabilità a future tecnologie hanno presto portato all'ingresso nello standard IEEE 802.3 di questo meccanismo.
L'auto-negoziazione entra in gioco quando viene stabilita una connessione con un dispostivo di rete, si occupa di determinare le velocità e le modalità di trasmissione messe a disposizione dal dispositivo e da ciò che si trova all'altra estremità del cavo Ethernet, detto Link Partner.
Successivamente porta il dispositivo a configurare in maniera automatica la modalità di trasmissione supportata che permetta di avere le performance migliori.

I vantaggi che derivano da questo meccanismo sono molteplici:

Connessione automatica: i dispositivi vengono collegati in maniera automatica assicurando le migliori performance possibili e senza l'ausilio di un utente o di un software di gestione;

Retro compatibilità: nel caso in cui l'auto-negoziazione sia disponibile solamente in uno dei dispositivi, questo si occupa di esaminare il segnale ricevuto dalla rete, di verificare se sia compatibile con una tecnologia supportata dal dispositivo stesso e successivamente avvia l'auto configurazione. Questa funzione di interfacciamento con sistemi privi di auto-negoziazione viene detta Rilevamento Parallelo;

Protezione della Rete: nel caso in cui non vi siano tecnologie comuni tra i dispositivi, l'auto-negoziazione non stabilisce la connessione, preservando l'integrità della rete e permettendo ai dispositivi già connessi di continuare a comunicare;

Interfaccia di Gestione: è disponibile una interfaccia che permette di determinare la presenza di errori, di verificare le capacità della rete e di modificare la velocità di connessione.

Il meccanismo che l'auto-negoziazione utilizza per segnalare le capacità di un dispositivo è la trasmissione di una serie di impulsi che codifica una parola di 16 bit, detta Fast Link Pulses (FLP). Come rappresentato in figura 3.1, questa può essere composta da 17 a 33 impulsi di durata 100 *ns* ciascuno, che sono identici a quelli utilizzati comunemente dalla rete per verificare se una connessione è valida, denominati Normal Link Pulses (NLP).
Ogni FLP ha una durata di 2 *ms* e viene trasmesso ad intervalli di 16 *ms*, con una tolleranza pari a 8 *ms*, come si può osservare in figura 3.1.

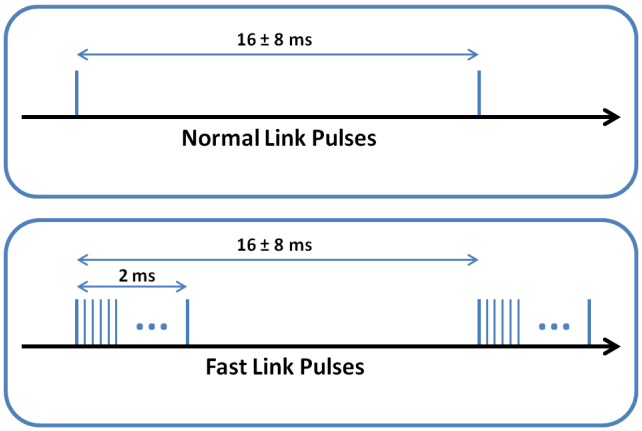


Figura 3.1 – Rappresentazione dei NLP e FLP.

La parola di 16 bit codificata viene definita Link Code Word (LCW) ed è strutturata come si vede in figura 3.2.

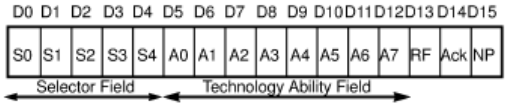


Figura 3.2 – Codifica della LCW [40].

Il Selector Field, $S[0:4]$, viene utilizzato per definire il tipo di messaggio trasmesso dalla auto-negoziazione, si ha la possibilità di utilizzare 32 configurazioni differenti, alcune delle quali rappresentano standard già definiti, come la 802.3, altre sono inutilizzate e messe a disposizioni per eventuali tecnologie future.
Il Technology Ability Field, $S[5:7]$, dipende dal tipo di messaggio definito nel Selector Filed e stabilisce quali modalità di comunicazione sono disponibili. Ad esempio, se nel Selector Field è presente $S[0:4] = 00001$, lo standard selezionato è l'IEEE 802.3 e il significato dei bit del Technology Ability Field è specificato in tabella 3.3.

bit 0	il dispositivo supporta 10BASE-T
bit 1	il dispositivo supporta 10BASE-T in full duplex
bit 2	il dispositivo supporta 100BASE-TX
bit 3	il dispositivo supporta 100BASE-TX in full duplex
bit 4	il dispositivo supporta 100BASE-T4
bit 5	pausa
bit 6	pausa asimmetrica per full duplex
bit 7	riservato

Tabella 3.3 – Significato dei bit del Technology Ability Field

Il bit Remote Fault (RF) permette la trasmissione di semplici informazioni di guasto al Link Partner, il bit Acknowledge (Ack) viene utilizzato per la sincronizzazione dei due nodi, infine il bit Next Page (NP) definisce se sono presenti altre funzioni in grado di trasmettere informazioni aggiuntive riguardanti le configurazioni disponibili.

La sincronizzazione tra il dispositivo e il Link Partner ha inizio quando il dispositivo trasmette il proprio LCW al Link Partner impostando il bit Ack a zero [11]. Una volta che

ha ricevuto tre LCW uguali dal Link Partner, imposta il bit Ack ad uno e trasmette ancora una volta il proprio LCW, informando il Link Partner di aver ricevuto il suo LCW.
A questo punto il dispositivo continua a trasmettere il proprio LCW fino a quando ne riceve dal Link Partner tre uguali con bit Ack ad uno, ha quindi la conferma che quest'ultimo ha effettivamente ricevuto il suo LCW e termina trasmettendolo ancora alcune volte per sicurezza.
Dal momento che entrambi i nodi conoscono le proprie caratteristiche a vicenda, sono in grado di determinare quali sono le tecnologie comuni supportate e sulla base di un sistema di priorità predefinito possono procedere con la auto configurazione, determinando quale tra le combinazioni ammissibili assicuri le performance migliori. La auto-negoziazione risulta essere uno strumento interessante per ridurre il consumo energetico dei link, infatti dà la possibilità di modificare la velocità dei nodi della rete e permette di ridurla nelle fasi di inattività, con un conseguente risparmio energetico. É però necessario determinare quali siano le tempistiche di tale meccanismo, in modo tale da capire se il cambio di velocità venga effettuato in un tempo sufficientemente breve da essere sfruttato nei brevi periodi di idle.

3.2 Misure

Per misurare il tempo necessario ad ottenere il cambio di velocità, si è deciso di collegare direttamente due PC tramite un cavo Ethernet e di prelevare i segnali trasmessi da un PC all'altro e viceversa attraverso due sonde differenziali collegate direttamente al cavo. A questo punto i due segnali ottenuti vengono portati ad un oscilloscopio digitale che permette di rappresentarne visivamente il comportamento e quindi di osservare lo scambio di pacchetti relativi all'auto-negoziazione.
Visto che nel sistema operativo utilizzato vi è la possibilità di impostare il funzionamento della scheda di rete, per ottenere questo scambio di pacchetti è stato sufficiente andare a modificare la velocità di trasferimento in uno dei due elaboratori, portando il secondo dispositivo a doversi adattare alla velocità impostata in maniera automatica. Per prima cosa si è cercato di abbassare la velocità di trasmissione, portandola da 1 Gbps a 10 Mbps e monitorando le informazioni scambiate nell'arco di una decina di secondi da quando le impostazioni scelte vengono effettivamente apportate alla scheda di rete.
I risultati ottenuti, però, sono poco significativi; infatti nella fase iniziale i segnali prelevati sono assimilabili a dei rumori, poichè la frequenza a cui è in grado di lavorare l'oscilloscopio è pari a 350 *Mhz*, nettamente inferiore alla velocità di trasferimento della rete, mentre,

nella fase successiva si notano una serie di impulsi trasmessi tra i due PC, che però non sono compatibili con quelli relativi all'auto-negoziazione. Per capire meglio il comportamento di questo meccanismo, si è quindi scelto di compiere l'operazione inversa, cioè di portare la velocità di uno dei due nodi da 10 Mbps a 1 Gbps.
Si ottiene, ovviamente, un comportamento inverso dei due segnali, ossia si nota una fase iniziale in cui vengono scambiati degli impulsi con una frequenza relativamente bassa e una in cui divengono simili a dei rumori e non è più possibile capire le informazioni trasmesse. In questo caso, come si può notare in figura 3.3, è possibile andare a distinguere in maniera piuttosto chiara gli FLP scambiati; si nota, infatti, un primo pacchetto trasmesso da un dispositivo per richiedere il cambio di velocità (1), una serie di pacchetti trasmessi dal secondo dispositivo con periodo pari a 16 *ms* (2) e le sequenze di tre pacchetti uguali trasmessi dal primo come conferma dell'avvenuta ricezione (3 e 4).

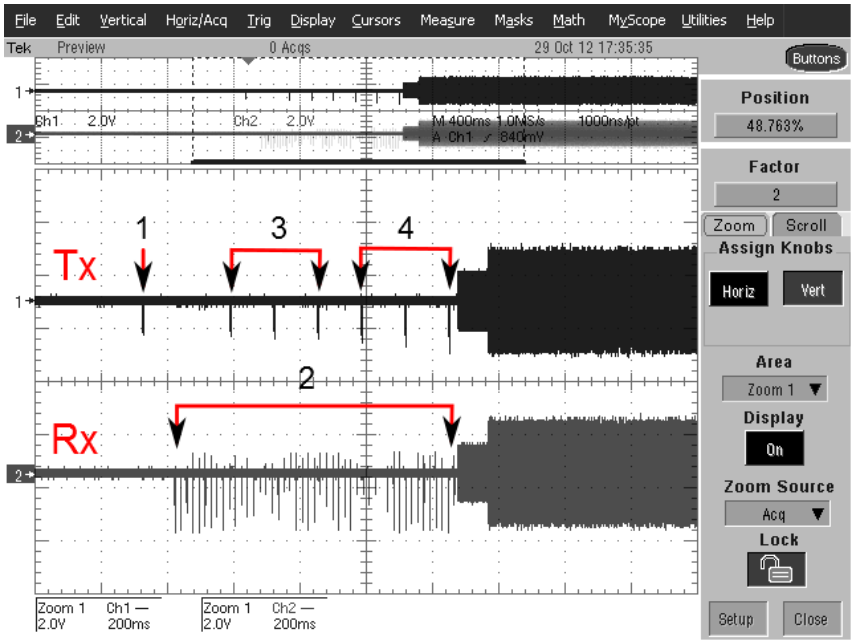


Figura 3.3 – Auto-negoziazione monitorata tramite l'oscilloscopio digitale.

Come si può osservare in figura 3.5, la durata dell'FLP è pari a 2.016 *ms*, perfettamente in linea con quanto specificato dalla teoria, e anche il periodo di trasmissione delle FLP è conforme, infatti misura esattamente 16 *ms*. Le FLP rilevate, tuttavia, non sono esattamente come quelle descritte, presentano, cioè, un numero di impulsi variabile e differente da quello teorico, inoltre la loro ampiezza e l'intervallo temporale che li divide, non è sempre lo stesso.
Ciò è dovuto a una imprecisione nella misura, infatti la loro durata è dell'ordine dei nano-

secondi, mentre l'oscilloscopio ha una risoluzione massima pari ad 1 μ, quindi non sempre è in grado di rilevare il cambiamento di tensione relativo all'impulso.

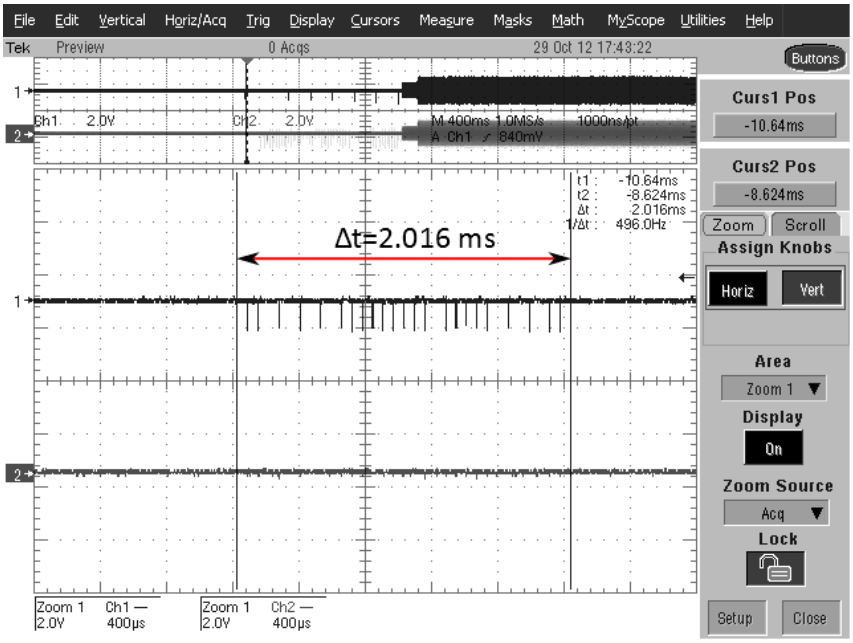


Figura 3.4 – Ingrandimento di un FLP trasmesso dal Link Partner.

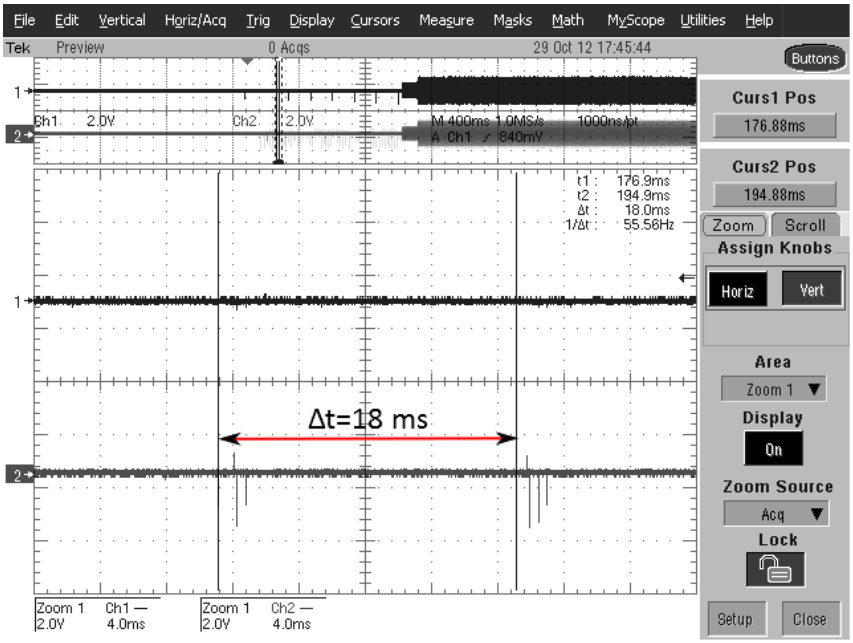


Figura 3.5 – Misura del periodo di trasmissione del link partner.

Sulla base dei risultati ottenuti si può supporre che, nel caso del passaggio da 1 Gbps a 10 Mbps, non si ottengono i risultati sperati perchè lo scambio di pacchetti necessari al cambio di velocità avviene quando i due dispositivi lavorano ancora alla velocità più elevata e l'oscilloscopio non è in grado di rappresentare in maniera chiara e interpretabile l'informazione che attraversa la rete.
Dalla figura 3.3 è anche possibile notare che il tempo necessario ad ottenere l'effettivo cambio di velocità, cioè quando i due segnali rilevati iniziano ad oscillare e a sembrare dei rumori, è dell'ordine di centinaia di millisecondi; in alcuni casi anche superiore al secondo. Questo implica che l'utilizzo dell'auto-negoziazione per implementare algoritmi di efficienza energetica in ambito real-time non è una buona soluzione, visto che si suppone di dover abbassare la velocità di trasferimento anche per brevi periodi di tempo. Potrebbe, tuttavia, essere una buona soluzione per le normali reti Ethernet, dove i periodi di inattività sono generalmente prolungati.
Un notevole vantaggio rispetto ad altre soluzioni proposte sta nel fatto che si utilizza uno strumento già presente nello standard Ethernet, quindi potrebbe essere sfruttato facilmente per ridurre i consumi energetici di una qualsiasi rete che adotta tale standard.

3.3 Adaptive Link Rate

Dalle considerazioni fatte fino ad ora, il tempo richiesto per cambiare la velocità di trasmissione sfruttando il meccanismo dell'auto-negoziazione è decisamente troppo elevato per le applicazioni di tipo real-time.
In [24, 23, 25], viene proposto un metodo alternativo che dovrebbe permettere di compiere l'operazione richiesta in un tempo che si aggira dell'ordine delle centinaia di microsecondi. Questa soluzione, indicata con il nome di Adaptive Link Rate (ALR), viene brevemente sintetizzata in figura 3.6, dove si possono notare due fasi distinte; la prima, in cui due nodi si scambiano una serie di pacchetti per avviare la procedura cambio della velocità e la seconda, in cui i due nodi si sincronizzano alla nuova velocità impostata.

Nella prima fase vengono utilizzati tre diversi frame del livello MAC di Ethernet:

ALR_REQ: rappresenta la richiesta da parte di un nodo di modificare la propria velocità di trasmissione. Nel frame stesso è presente il rate desiderato;

ALR_ACK: indica la conferma da parte del link partner per procedere con il cambio di velocità;

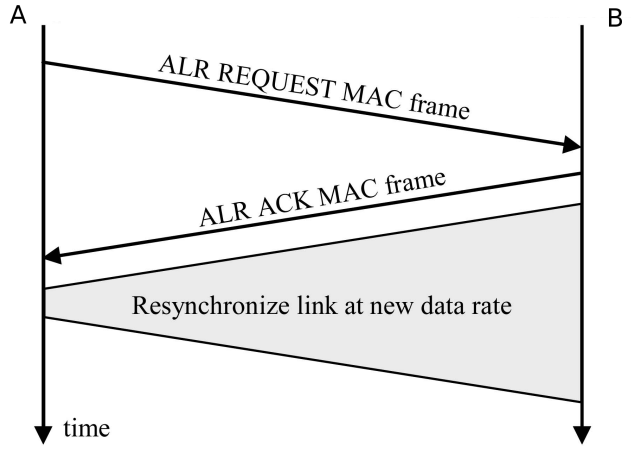


Figura 3.6 – Diagramma temporale raffigurante le fasi di sincronizzazione dell'ALR.

ALR_NACK: indica la presenza di anomalie o problemi che impediscono il cambio della velocità.

Una stazione che desidera cambiare il proprio rate di trasmissione, quindi, invia una richiesta tramite il frame ALR_REQ, per dare al link partner tutte le informazioni necessarie. A questo punto il nodo avvia un timer e si mette in attesa di una risposta, ritrasmettendo nuovamente il frame nel caso in cui il timer scada. Quando il link partner riceve tale frame, verifica se sia effettivamente possibile cambiare la velocità; in caso affermativo trasmette il frame ALR_ACK e avvia la procedura per cambiare la propria velocità di trasmissione, altrimenti informa il nodo che non è possibile compiere tale operazione inviando un frame ALR_NACK. A questo punto, se il nodo iniziale riceve un frame ALR_ACK, avvia la procedura per portare la velocità al valore desiderato, altrimenti continua a trasmettere con il rate precedente.
Una volta che i due nodi hanno effettuato questo scambio di pacchetti, ha inizio la seconda fase dell'ALR, in cui i due si sincronizzano al nuovo rate desiderato. Per far ciò, dal momento che l'auto-negoziazione deve essere disabilitata, si rende necessario configurare direttamente i registri presenti nella scheda NIC e forzare la stessa a sincronizzarsi alla nuova velocità impostata.
Per comprendere meglio come avvenga effettivamente tale sincronizzazione conviene studiare il comportamento della scheda NIC e le operazioni richieste per ottenere il cambio della velocità.
Lo schema in figura 3.7 rappresenta i livelli definiti dallo standard Ethernet [27]. I due componenti che si occupano della verifica dell'integrità del link stabilito tra due stazioni

sono il MAU, nel caso delle reti con velocità di trasmissione non superiore ai 10 Mbps, e il Physical Medium Attachment (PMA), nel caso di reti con velocità superiori a 10 Mbps.
Questi componenti mettono a disposizione una funzione detta *link integrity function*, che stabilisce se il link presente nella rete in questione sia effettivamente funzionante e permetta la comunicazione.

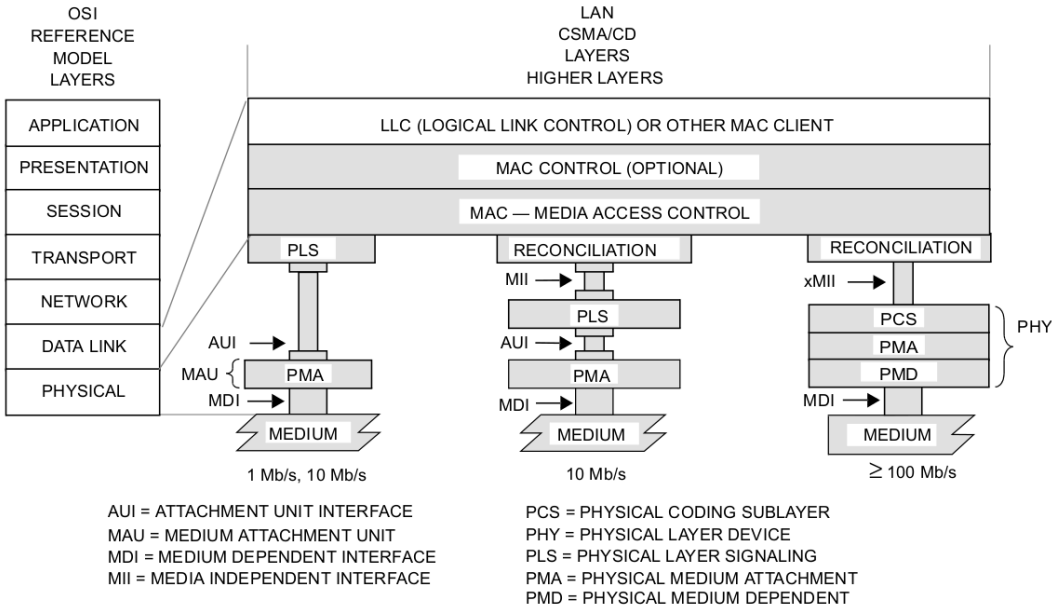


Figura 3.7 – Schema dei livelli definiti nello standard 802.3 relativo alle varie velocità di trasmissione.

La configurazione del link avviene grazie a due registri presenti nella Media Indipendent Interface (MII), cioè una interfaccia che separa il livello MAC da quello fisico che permette l'utilizzo di funzioni come l'auto-negoziazione o la configurazione manuale del link. Tramite il registro di controllo è possibile disabilitare l'autonegoziazione e settare la velocità di trasmissione, mentre tramite il registro di stato è possibile stabilire la modalità di trasmissione (full duplex o half duplex).
Una volta disabilitata la funzione di auto-negoziazione è necessario verificare che le configurazioni impostate nei nodi che si vogliono connettere siano esattamente uguali, altrimenti la comunicazione non sarebbe possibile.
Nel MAU o nel PMA vengono definite due variabili, link_control e link_status, che stabiliscono rispettivamente se la funzione di integrità del link sia abilitata a controllare la comunicazione e se il link monitorato sia stato effettivamente stabilito correttamente.
Quando l'auto-negoziazione è disabilitata, la variabile link_control viene automaticamente

portata al valore ENABLE, quindi il link è continuamente monitorato ed un eventuale variazione della velocità di trasmissione in uno dei due nodi del link viene immediatamente rilevata dalla link integrity function, che porta la variabile link_status ad assumere il valore FAIL, con una conseguente perdita del link.
Un secondo aspetto di cui tenere conto è che la configurazione del link avviene in fase di inizializzazione sulla base delle operazioni risultati dalla auto-negoziazione o sulla base dei registri di configurazione e di stato quando questi sono impostati manualmente. L'eventuale modifica dei bit di questi registri non porta all'immediato aggiornamento della configurazione del link, ma influenzerà la velocità e la modalità di trasmissione solo nel caso in cui il link venga disattivato (*link down*) e successivamente riattivato (*link up*).
Sulla base delle considerazioni appena fatte è possibile dedurre che, nei sistemi che implementano lo standard Ethernet 802.3, per modificare la velocità e la modalità di trasferiemento con un altro nodo è sempre necessario disattivare e successivamente riattivare il link, sia nel caso in cui l'auto-negoziazione è attivata, sia in quello in cui viene adottata la configurazione manuale.
Questa affermazione è stata verificata per sistemi operativi Windows e Linux. Nel primo caso, la configurazione manuale avviene tramite il settaggio di un registro di sistema associato alla scheda Ethernet e grazie alla disattivazione e riattivazione della scheda stessa. Nel secondo caso, invece, è disponibile un tool di gestione (*ethtool*), che permette di modificare i parametri della scheda da linea di comando e compie autonomamente le operazioni di link down e link up.

Dal momento che il link deve essere ristabilito per poter riprendere le trasmissioni, si rende necessario riconfigurare gran parte dei componenti presenti nella scheda, come gli equalizzatori, i soppressori del rumore e i circuiti di temporizzazione, che hanno parametri differenti in base al tipo di trasmissione che si vuole stabilire. Queste operazioni di inizializzazione, che non possono essere evitate nello standard IEEE 802.3, richiedono generalmente dei tempi dell'ordine delle centinaia di millisecondi [45].
Si può quindi concludere che il miglioramento ottenibile rispetto al caso in cui si sfrutta l'auto-negoziazione non è sufficiente a rendere il meccanismo dell'ALR adatto a sistemi real-time, in cui si devono rispettare deadline molto stringenti e non è ammissibile una perdita del link o un rallentamento della velocità di trasmissione per tempi così elevati.

Cambiare la velocità di trasmissione, però, potrebbe essere una buona soluzione per ridurre i consumi energetici nelle comuni reti Ethernet, in cui i tempi di inattività (idle) sono generalmente elevati e non è richiesto di rispettare deadline particolarmente stringenti. Il meccanismo implementato per cambiare la velocità ha il vantaggio di utilizzare funzioni

già definite nello standard Ethernet 802.3 ed è quindi possibile sfruttarlo con una qualsiasi scheda NIC aggiungendo un programma in grado di monitorare le fasi di idle e di cambiare autonomamente la velocità.

3.3.1 Implementazione in linguaggio C

Lo scambio di pacchetti tra i due nodi avviene a livello MAC, si è quindi deciso di creare un programma in C che permetta la comunicazione tra due nodi collegati, specificando esclusivamente l'indirizzo MAC del destinatario del pacchetto di sincronizzazione. Per fare ciò si rende necessario l'utilizzo dei socket, ovvero di una astrazione software di una porta di comunicazione, che permette la trasmissione e la ricezione di pacchetti attraverso la rete, utilizzando delle semplici funzioni della API corrispondente.

I parametri da definire per creare un socket sono

dominio: rappresenta la famiglia a cui appartiene il socket che si vuole utilizzare. Solo socket dello stesso dominio possono comunicare tra loro, quindi, nei due nodi interessati, dovranno essere aperti due socket col medesimo dominio;

tipo: definisce il tipo di comunicazione che caratterizza il socket, in particolare la bidirezionalità e il tipo di dato trasmesso;

protocollo: è il protocollo di comunicazione utilizzato dal socket, che definisce le regole di trasferimento e la configurazione del frame trasmesso.

Dal momento che si vogliono trasmettere pacchetti a basso livello, si è deciso di utilizzare un particolare tipo di socket, denominato RAW, che permette di bypassare la fase di incapsulamento/decapsulamento relativa al protocollo TCP/IP e di trasmettere direttamente il pacchetto al nodo destinatario. Questa modalità ha il vantaggio di rendere la trasmissione del pacchetto il più possibile indipendente dal sistema operativo, impedendo ad altri processi di interferire e riducendo al minimo il tempo di trasferimento necessario.
Una volta creato un RAW socket, si procede con la costruzione del frame da inviare, in cui si deve inserire l'indirizzo MAC del mittente, quello del destinatario e un campo dati, che permette di definire un pacchetto ALR_ACK, ALR_NACK oppure la velocità a cui si vuole passare nel caso di un pacchetto ALR_REQ.
Le funzioni che permettono la trasmissione e a ricezione dei pacchetti sono rispettivamente:

send: richiede un riferimento al socket su cui trasmettere, il frame da trasmettere sotto forma di buffer, la lunghezza del buffer ed un flag per impostazioni aggiuntive non

necessarie per questa applicazione. Restituisce il numero di byte trasmessi o -1 in caso di errore;

recv: richiede un riferimento al socket su cui ricevere, un buffer su cui salvare il frame ricevuto, la sua dimensione ed un flag. Restituisce il numero di byte ricevuti o un codice di errore.

Infine, per ottenere il cambio di velocità, è stata utilizzata una chiamata a sistema che per sistemi Windows si occupa di disattivare l'auto-negoziazione, modificare il registro di sistema relativo alla scheda NIC e di disconnettere e riconnettere il link, mentre, per sistemi LINUX, permette di utilizzare un tool che svolge operazioni analoghe a quelle appena descritte in maniera autonoma.

In figura 5.5 è schematizzata l'implementazione della sincronizzazione. Una volta che il secondo nodo ha trasmesso il pacchetto ALR_ACK, entrambi i nodi avviano la procedura di cambio di velocità. Per misurare quanto tempo sia necessario a modificare il rate di trasmissione, si è poi deciso di far continuare a inviare dei pacchetti al primo nodo, e di rilevare via software il tempo che intercorre tra l'avvio della procedura e la ricezione del primo di questi pacchetti, che non possono giungere a destinazione fino a quando il link non è stato ristabilito con la nuova velocità.

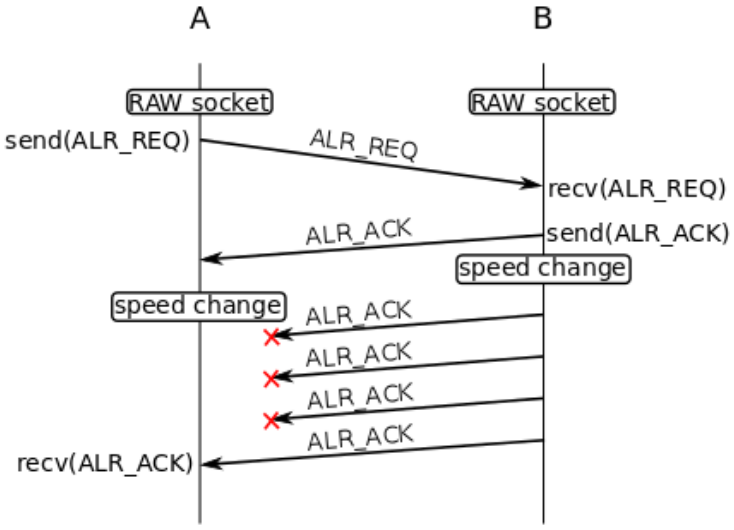


Figura 3.8 – Sincronizzazione dei nodi tramite ALR

Il tempo misurato può essere suddiviso in due parti fondamentali: la prima, indicata con $t_{handshake}$, fa riferimento allo scambio dei pacchetti ALR_REQ e ALR_ACK, mentre la

seconda, denominata t_{change}, rappresenta il tempo necessario riconfigurare il link alla nuova velocità .

$$t_{ALR} = t_{handshake} + t_{change}$$

Per ottenere una misura molto precisa di $t_{handshake}$ si ha la possibilità di utilizzare dei cosiddetti tool di sniffing, cioè dei software in grado di monitorare passivamente i pacchetti che transitano in una rete. Questi tool, però, essendo implementati in maniera software, dipendono dal sistema in cui vengono installati e le misure temporali che sono in grado di fare sui pacchetti in transito sono fortemente influenzate dalle operazioni che sta compiendo in quel momento il processore.
Per evitare questo inconveniente e per ottenere misure più precise, si è deciso di utilizzare un dispositivo hardware, in grado di rilevare pacchetti attraverso la rete senza influenzarla in alcun modo e senza essere a sua volta influenzata dal sistema in cui è installato.

La scheda Endace DAG 3.6E [8] rappresentata in figura 3.9 è una scheda di acquisizione inseribile direttamente in uno degli slot PCI della scheda madre di un PC, che permette di intercettare i pacchetti che transitano lungo una qualsiasi rete Ethernet a cui è collegata.

Figura 3.9 – Scheda Endace DAG 3.6E.

É dotata di due porte per connettori RJ45 non isolate galvanicamente tra loro, quindi non è presente alcun tipo di hub al suo interno.

Come si vede in figura 3.10, queste porte permettono il collegamento in serie con una rete Ethernet a velocità 10 o 100 Mbps full duplex o half duplex e non interferisce in alcun modo con il trasferimento dei pacchetti o con le operazioni di auto-negoziazione, rendendo-si quindi trasparente alla rete stessa. Una terza porta può essere utilizzata per sincronizzare

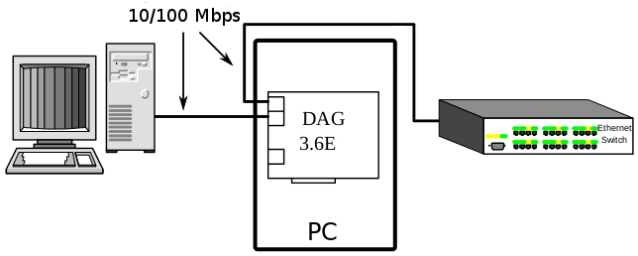


Figura 3.10 – Schema di collegamento della scheda Endace DAG 3.6E.

la scheda con altri strumenti simili, in modo da creare un sistema di misura più complesso. Ogni pacchetto che transita nella rete entra in una delle due porte, viene rilevato da un framer Ethernet ed esce dalla seconda porta senza alcuna modifica. I dati ricevuti, quindi, arrivano ad una Field Programmable Gate Array (FPGA) dotata di un processore e di un sistema di timestamping denominato DAG Universal Clock Kit (DUCK), basato su di un clock che viene inizialmente sincronizzato a quello del PC, ma che poi viene gestito in maniera indipendente da un oscillatore interno alla scheda.
I tempi registrati dalla scheda sono rappresentati in virgola fissa utilizzando 64 bit, i 32 bit più significativi dei quali indicano i secondi trascorsi dalla mezzanotte del primo Gennaio 1970, mentre i 32 bit meno significativi sono utilizzati per rappresentare la parte frazionaria. Sulla base di quanto appena affermato, sarebbe teoricamente possibile ottenere una risoluzione pari a $\frac{1}{2^{32}} \simeq 231.83\ ps$. In realtà la frequenza del clock porta ad avere gli 8 bit meno significativi costantemente a zero, quindi la reale risoluzione temporale della scheda è pari a $\frac{1}{2^{24}} \simeq 59.6\ ns$.
La scheda, una volta installata, può essere comandata da MS-DOS in ambienti Windows, o da Shell in ambienti Linux, da cui è possibile avviare il monitoraggio della rete e salvare i risultati ottenuti in formato .ERF, che, a loro volta, possono essere visualizzati da comuni software di sniffing come Wireshark. Non avendo a disposizione un hub in cui sia in grado di disabilitare l'auto-negoziazione è stato possibile esclusivamente misurare il valore di $t_{handshake}$, mentre, per quanto riguarda t_{change} è stato necessario utilizzare i risultati riportati in [45]. In tabella 3.4 sono riassunti i tempi caratteristici dell'ALR per le reti a 10 Mbps, 100 Mbps e 1000 Mbps. Come si può notare i tempi sono inferiori rispetto al caso del cambio di velocità con auto-negoziazione, ma sono ancora troppo elevati, soprattuto per rendere questo algoritmo adatto a reti real-time in cui i tempi di ciclo sono spesso inferiori al millisecondo.

Velocità iniziale / velocità finale	$t_{handshake}$ $[\mu s]$	t_{change} $[ms]$	t_{ALR} $[ms]$
1000 Mbps / 100 Mbps	2.0	72.333	72.335
100 Mbps / 1000 Mbps	8.4	68.585	68.594
100 Mbps / 10 Mbps	2.0	575,836	575.838
10 Mbps / 100 Mbps	88.6	72,334	72,422
1000 Mbps / 10 Mbps	8.4	575,836	575.844
10 Mbps / 1000 Mbps	88.6	68,586	68,674

Tabella 3.4 – Tempi caratteristici dell'algoritmo ALR.

3.4 Energy Efficiency Ethernet

L'approccio più efficace per risolvere questo problema è quello descritto nello standard IEEE 802.3az [28], contenente una serie di migliorie apportate alle comuni reti Ethernet IEEE 802.3, che permettono di ottenere una rete più efficiente dal punto di vista energetico, definita appunto EEE.
Lo standard è stato sviluppato dalla IEEE 802.3az task force e la sua versione definitiva è stata ufficializzata dalla IEEE nel Settembre del 2010 [41].

Questo standard combina il livello MAC definito da IEEE 802.3 con una serie di differenti livelli fisici, che supportano una nuova modalità di lavoro, denominata Low Power Idle (LPI). Quando questa modalità è abilitata, il sistema può ridurre il consumo in entrambe le direzioni dei link durante le fasi in cui il loro utilizzo è minore. Esiste, inoltre, una generalizzazione di questo standard implementata da alcune compagnie, denominata *Green Ethernet*, in cui il consumo di potenza è regolato non solo in base al carico del link in questione, ma anche in relazione alla sua lunghezza [1, 10].

Le tipologie di livelli fisico supportate dallo standard sono: 100BASE-X, 100BASE-TX, 1000BASE-X, 1000BASE-T, 10GBASE-X, 10GBASE-R, 1000BASE-KX, 10GBASE-KX4 e 10GBASE-KR. Inoltre viene definita la rete 10BASE-Te, cioè una versione efficiente dal punto di vista energetico della comune rete 10BASE-T a 10 Mbps, che necessita di una ampiezza di trasmissione inferiore [28]. Il MAU relativo a questo livello fisico è perfettamente compatibile con quello delle reti 10BASE-T, permettendo di combinare dispositivi che implementano meccanismi di riduzione del consumo con quelli già presenti nelle normali reti Ethernet.

Nello standard sono definite tutte le modalità con cui i nodi si scambiano informazioni per determinare se supportano l'EEE e per determinare la migliore configurazione dei parametri principali che caratterizzano il suo funzionamento, in modo da poter ottenere la massima riduzione possibile dei consumi energetici.

Il concetto fondamentale su cui si basa EEE è la definizione di due differenti modalità in cui possono operare i dispositivi: la prima, denominata ACTIVE, viene assunta nelle fasi di utilizzo del dispositivo stesso, cioè quando è necessario trasmettere informazioni ad altri nodi della rete; mentre la seconda, denominata LPI, rappresenta lo stato di basso consumo energetico del dispositivo, che viene assunto nei momenti in cui l'utilizzo del link è ridotto e si ha uno spreco di energia che può essere evitato.
Quando viene attivata la modalità LPI, si suppone ovviamente che non sia necessario trasmettere dei pacchetti attraverso la rete, quindi è possibile disabilitare gran parte delle funzionalità del trasmettitore, ottenendo il risparmio energetico desiderato. Il concetto è molto simile a quelli descritti negli algoritmi visti in precedenza, ma ha il grande vantaggio di richiedere solo pochi microsecondi per il passaggio tra lo stato ACTIVE e LPI, visto che il link non viene disattivato [28].
Qualora si desideri nuovamente trasmettere lungo la rete, infatti, non è più necessario compiere le operazioni per ristabilire il link, che richiedono alcuni millisecondi, ma è sufficiente riattivare le funzionalità del trasmettitore, riducendo drasticamente il tempo necessario a riprendere le comunicazione. Questa fondamentale caratteristica rende lo standard EEE il miglior candidato per la riduzione dei consumi non solo nelle comuni reti Ethernet, ma anche nelle reti Ethernt industriali, in cui i tempi di trasmissione sono di primaria importanza.

3.5 LPI Client e Reconciliation Sublayer

Come si vede in figura 3.11, rispetto al normale standard Ethernet, viene introdotta una nuova entità, definita LPI Client, che si occupa di gestire i segnali necessari alle transizioni ACTIVE/LPI e che viene collegato al Reconciliation Sublayer (RS). Quest'ultimo ha il compito di tradurre la terminologia adottata dal Media Indipendent Interface (MII) [1], cioè dall'interfaccia di collegamento con il PHY, con quella del MAC e dell'LPI Client.

Il livello fisico descritto dallo standard IEEE 802.3az è dotato di alcune funzioni che permettono di trasmettere e ricevere dei particolari segnali detti di Low Power Idle, necessari a gestire i cambi di stato da parte del dispositivo e a mantenere attivo il link a cui è collegato. Questi segnali permettono al LPI Client di indicare al PHY e al link partner che

[1] In figura è utilizzato il termine xMII per indicare qualsiasi tipo di MII supportato dallo standard Ethernet

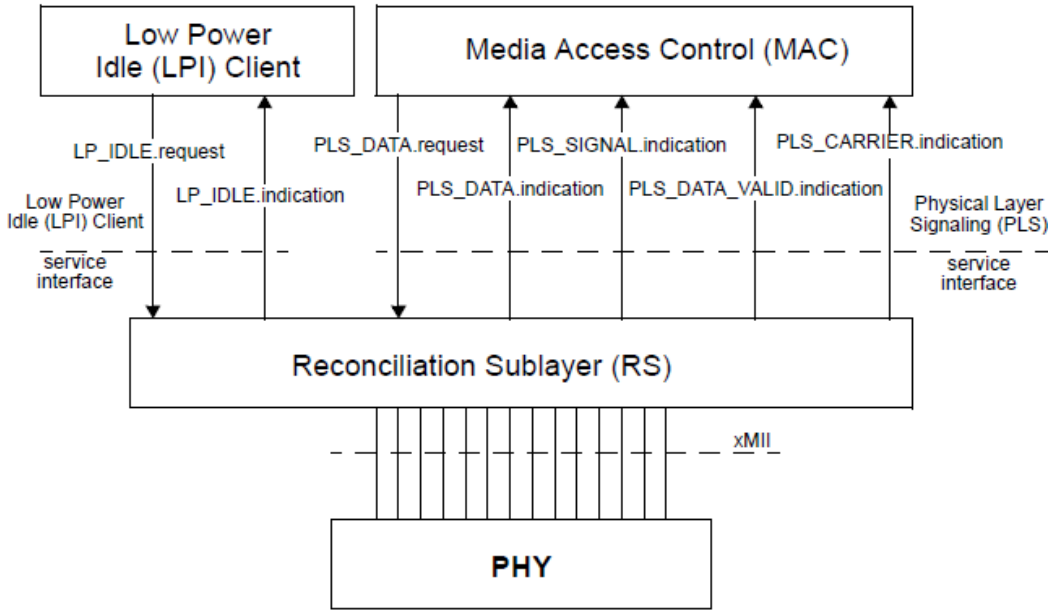


Figura 3.11 – Modulo LPI Client e RS aggiunti allo standard Ethernet.

presto verrà bloccato il flusso di dati e che il trasmettitore locale sta per passare alla modalità LPI. Essi, inoltre, informano il modulo LPI Client dell'eventuale richiesta da parte del link partner di cambiare il proprio stato.

Per generare, ricevere e scambiare questi segnali con l'RS, l'LPI Client sfrutta le seguenti interfacce [35]:

LP_IDLE.request(LPI_REQUEST) viene utilizzato dall'LPI Client per richiedere di iniziare (LPI_REQUEST = ASSERT) o bloccare la trasmissione (LPI_REQUEST = DE-ASSERT) di segnali LPI verso il livello PHY e quindi verso il link partner;

LP_IDLE.indication(LPI_INDICATION) viene utilizzato dall'RS per indicare all'LPI Client che il livello PHY ha ricevuto un segnale LPI dal link partner (LPI_INDICATION = ASSERT) oppure che non vi sono segnali di questo tipo in arrivo (LPI_INDICATION = DE-ASSERT).

Queste stesse interfacce vengono gestite dall'RS tramite due funzioni, la *LPI assert function* e la *LPI detect function*, che si occupano rispettivamente di gestire le richieste di trasmissione da parte dell'LPI Client e di identificare i segnali provenienti dal PHY. In figura 3.12 si possono notare le primitive utilizzate da queste due funzioni per mettere in

comunicazione il PHY con il MAC e con il modulo LPI Client, nel primo caso vengono utilizzate quelle messe a disposizione dall'interfaccia Physical Layer Signaling (PLS), come avviene normalmente nelle reti Ethernet, mentre, nel secondo caso, vengono utilizzate le primitive appena descritte.

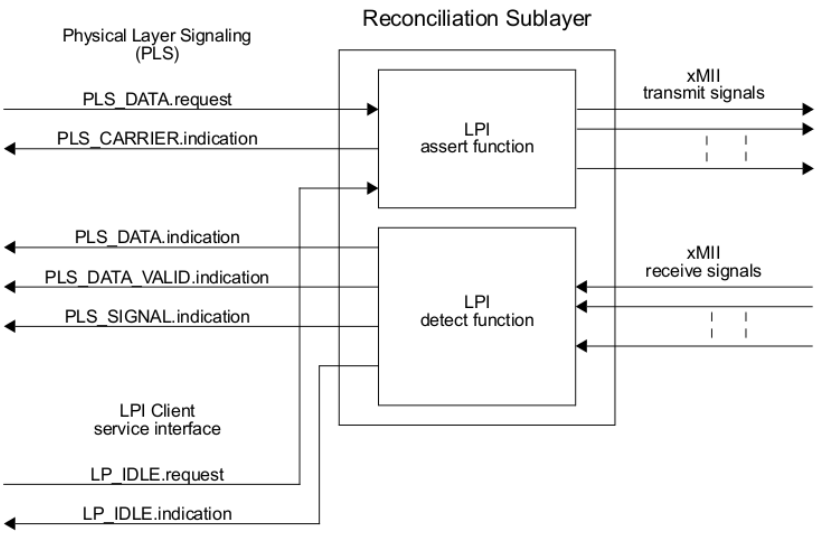


Figura 3.12 *– LPI assert function* e *LPI detect function* presenti nel modulo RS.

Quando non vi sono richieste di passare nello stato LPI, la *LPI assert function* gestisce la comunicazione tra il livello MAC e la xMII mappando l'interaccia di servizio PLS con i segnali trasmessi alla xMII, come avviene nelle normali condizioni di trasmissione. Utilizza, inoltre, le primitive di questo servizio per indicare al MAC che può riprendere a trasmettere.
Quando, invece, viene effettuata una richiesta di passaggio in LPI, corrispondente al settaggio del parametro LPI_REQUEST al valore ASSERT, la *LPI assert function* inizia a trasmettere al xMII la codifica Assert LPI, rappresentata da una sequenza di 4 bit predefinita. Contemporaneamente sfrutta le primitive del PLS per indicare al MAC di interrompere ogni trasmissione.

Per quanto riguarda la *LPI detect function*, essa riceve i segnali provenienti dallo xMII, se questi non codificano la Assert LPI, li mappa nella interfaccia PLS, come avviene normalmente nelle reti Ethernet, e notifica al modulo LPI Client che non ci si trova nello stato LPI, settando il parametro LPI_INDICATION della primitiva LP_IDLE.indication a DEASSERT.
Se, invece, viene rilevata la sequenza che rappresenta la Assert LPI, questa funzione modi-

fica il parametro LPI_INDICATION portandolo al valore ASSERT, in modo da informare il modulo LPI Client che il link partner è passato alla modalità LPI

Riassumendo, si ha che il modulo LPI Client decide quando il segnale LPI debba essere trasmesso al link partner, comunicando con il PHY grazie all'RS. Viceversa le richieste provenienti dal link partner vengono dapprima trasmesse dal PHY all'RS, che successivamente le inoltra all'LPI Client.

3.6 Operazioni compiute dal PHY in trasmissione

Per compiere il passaggio dallo stato ACTIVE a quello LPI, il PHY deve compiere alcune operazioni che variano a seconda della tipologia di rete in questione. In generale, in un dispositivo che si trova nello stato ACTIVE, il modulo LPI Client compie una richiesta per passare nello stato di Low Power Idle, impostando il valore del parametro LPI_REQUEST ad ASSERT all'interno della primitiva LPI_IDLE.request. In questo modo la xMII riceve la codifica Assert LPI e lo notifica al PHY, che trasmette il segnale *sleep signal* al proprio link partner, in modo da informarlo che sta per essere avviata la procedura per passare il trasmettitore locale nella modalità LPI.

Il dispositivo locale, a questo punto, attraversa una prima fase denominata SLEEP, durante la quale vengono compiute le operazioni necessarie a spegnere i componenti del trasmettitore che consumano energia inutilmente.
La fase successiva, denominata QUIET, è quella in cui effettivamente il nodo si trova nella modalità LPI e viene periodicamente interrotta dall'operazione di REFRESH. Durante questa fase periodica, la funzione di trasmissione del PHY locale viene riabilitata momentaneamente, permettendogli di trasmettere periodicamente dei segnali di refresh, che hanno lo scopo di essere sfruttati dal link partner per aggiornare i propri circuiti di temporizzazione. Essi, inoltre, sono necessari a mantenere il link attivo; infatti la funzione di integrità, descritta nella sezione 3.1, monitora continuamente il link e, se non rileva traffico al suo interno per un tempo prefissato, lo disattiva automaticamente supponendo che sia avvenuto qualche errore nelle comunicazioni.
Come si può notare in figura 3.13, la fase di LPI è costituita da un alternarsi periodico delle fasi di QUIET e REFRESH, che si interrompe solo quando viene inoltrata una richiesta di tornare alla modalità ACTIVE, rappresentata dalla ricezione da parte della xMII di una qualsiasi codifica differente da Assert LPI. Il PHY locale, quindi, trasmette un segnale *wake signal* al link partner ed avvia la procedura necessaria alla riattivazione di tutte le funzionalità del trasmettitore, rappresentata dalla transizione nella fase WAKE. Conclusa

questa ultima fase, il nodo locale si trova nuovamente alle normali condizioni di lavoro e può procedere con le trasmissioni.

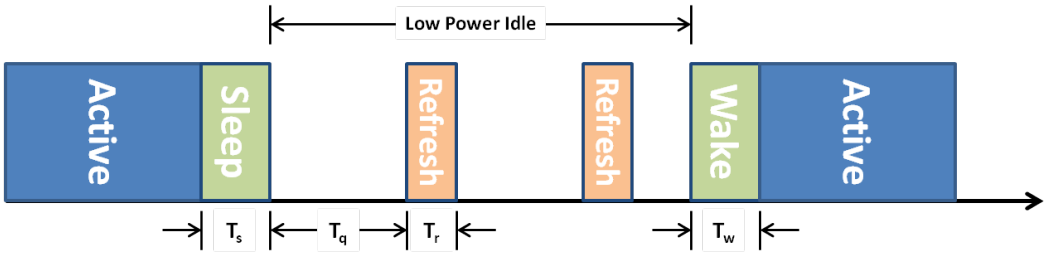


Figura 3.13 – Fasi delle modalità ACTIVE e LPI.

Ognuna di queste fasi è caratterizzata da un tempo ben preciso:

T_s fa riferimento alla fase di SLEEP e rappresenta il tempo necessario a spegnere i componenti del trasmettitore che non richiedono di ristabilire il link per riprendere le comunicazioni;

T_q è un tempo fissato a priori che rappresenta la durata della fase di QUIET, cioè l'intervallo che intercorre tra due fasi di REFRESH consecutive;

T_r è il tempo necessario a trasmettere i segnali di refresh al link partner;

T_w è il tempo impiegato durante la fase WAKE per riattivare le funzionalità del trasmettitore e del ricevitore, provocando il passaggio nella modalità ACTIVE.

La trasmissione periodica dei segnali di refresh non richiede di riattivare completamente il trasmettitore, infatti è sufficiente utilizzare solamente una delle quattro coppie di cavi presenti nel link [28], quindi il consumo e il tempo necessario a trasmettere questi segnali sono ridotti rispetto alle normali trasmissioni del link attivo.

3.6.1 Operazioni compiute dal PHY in ricezione

Nella direzione di ricezione, l'ingresso nella modalità LPI da parte del link partner è rilevata dalla presenza dello sleep signal, dopo il quale vengono interrotte le trasmissioni dal nodo remoto. A questo punto, il PHY locale indica la codifica Assert LPI nello xMII e anche il ricevitore locale può disabilitare alcune delle sue funzionalità, in modo da ridurre ulteriormente il consumo.

Il ricevitore locale acquisisce i segnali di refresh da parte del link partner ed aggiorna i propri componenti.

Quando il nodo remoto decide di uscire dalla modalità LPI e di interrompere il ciclo QUIET-REFRESH, trasmette il wake signal, che viene sfruttato dal ricevitore locale per avviare la procedura di riattivazione delle eventuali funzionalità disabilitate, provocando la transizione nella xMII dalla codifica Assert LPI alla codifica normalmente presente per le tramissioni Ethernet. Passato il tempo necessario a ristabilire il ricevitore, il link può tornare al rate di trasmissione nominale.

Una distinzione va fatta per le reti 1000BASE-T. In questo caso, infatti, il PHY locale si porta nella fase QUIET solo dopo che ha trasmesso lo sleep signal e ha ricevuto lo stesso segnale da parte del PHY remoto. Ciò significa che, se il PHY remoto non dovesse decidere di portarsi nella modalità LPI, allora neanche il PHY locale potrà farlo. Le altre tipologie di reti, invece, possono gestire il passaggio in LPI nelle due direzioni del link in maniera indipendente, cioè si può presentare la situazione in cui il trasmettitore locale ed il ricevitore remoto vengano parzialmente disabilitati, ma non viceversa.
Questa possibilità risulterà essere molto utile nel caso delle reti real-time dal momento che, come si vedrà, talvolta i nodi devono rimanere in ascolto sulla rete per la maggior parte del tempo e non hanno la possibilità di disattivare le funzionalità dal proprio ricevitore, bensì solamente del trasmettitore.

Lo standard descrive anche la gestione delle situazioni in cui si debbano trasmettere o ricevere normali frame durante le fasi di QUIET e REFRESH. Nel caso in cui il PHY riceva la richiesta di trasmettere un frame lungo il link, avvia automaticamente la procedura per la riattivazione di tutte le funzionalità del trasmettitore e del ricevitore, uscendo dalla modalità LPI nel tempo T_w. Dall'altro lato del link, il PHY remoto, è in grado di rilevare la presenza del frame e, di conseguenza, avvia anch'esso la procedura per uscire dal LPI.
Questo introduce un ritardo nella trasmissione del frame, pari alla durata della fase di WAKE, ma, come si può notare nella tabella 3.6, questo ritardo non supera i 20.5 μs ed è generalmente trascurabile rispetto alle altre tempistiche della rete. Nella prossima sezione, inoltre, vedremo che l'adozione di questi meccanismi nei sistemi real-time ha il vantaggio di conoscere con grande precisione gli istanti in cui potrebbe essere necessario trasmettere o ricevere dei frame lungo la rete ed è quindi possibile anticipare il risveglio del link in modo da evitare qualsiasi tipo di ritardo.

Il meccanismo descritto fino ad ora è comune a tutte le tipologie di PHY che supportano l'EEE; si hanno però alcune differenze per quanto riguarda le tempistiche dei vari cambi di stato che caratterizzano questo standard e per quanto riguarda la gestione delle due direzioni del link tra l'LPI Client ed il link partner.
In tabella 3.5 sono riassunti i parametri temporali relativi alla fase di SLEEP, di QUIET e di

REFRESH, la cui durata dipende dalla velocità e dalla tipologia di rete adottata.

Protocollo	$T_s[\mu s]$		$T_q[\mu s]$		$T_r[\mu s]$	
	Min	*Max*	*Min*	*Max*	*Min*	*Max*
100BASE-TX	200	220	20000	22000	200	220
1000BASE-T	182.0	202.0	20000	24000	198.0	218.2
1000BASE-KX	19.9	20.1	2500	2600	19.9	20.1
10GBASE-KX4	19.9	20.1	2500	2600	19.9	20.1
10GBASE-KR	4.9	5.1	1700	1800	16.9	17.5
10GBASE-T	2.88	3.2	39.68	39.68	1.28	1.28

Tabella 3.5 – Sommario dei parametri temporali caratteristici dello standard EEE [28].

Nel caso del tempo T_w, è necessario tenere conto dei tempi di latenza della rete, che hanno una ripercussione sui ritardi di trasmissione e ricezione dei frame.
Nello standard Ethernet vengono stabiliti dei limiti massimi per quanto riguarda i ritardi di propagazione lungo la rete in base al tipo di protocollo adottato, che permettono di determinare il tempo massimo necessario a risvegliare il link, cioè il limite superiore per il tempo T_w, che viene denominato T_{w_phy}. Questo tempo dipende da vari fattori caratteristici del protocollo, tra cui la velocità di trasmissione, la possibilità di gestire separatamente il link nelle due direzioni possibili e la presenza di un controllo degli errori. Pertanto, per alcune delle reti considerate, è necessario distinguere due differenti casistiche, a cui corrispondono due T_{w_phy} diversi.

10GBASE-T per questo protocollo è possibile decidere se il PHY locale debba o meno attendere la ricezione di uno sleep signal da parte del link partner, prima di poter trasmettere il proprio wake signal. Vengono quindi considerati il caso in cui non si attende lo sleep signal (caso 1) e quello in cui è necessario attenderlo (caso 2).

10GBASE-KR in queste reti è possibile applicare un meccanismo di controllo e correzione degli errori denominato Forward Error Correction (FEC). Pertanto vengono considerati il caso in cui non viene implementato il FEC (caso 1) ed il caso in cui questo controllo è presente (caso 2);

In tabella 3.6, sono riassunti i massimi tempi necessari alle varie tipologie di reti per risvegliare il link, distinguendo eventualmente i casi appena descritti.

Protocollo	Caso	$T_{w\_phy}[\mu s]$
100BASE-TX	-	20.5
1000BASE-T	-	16.5
1000BASE-KX	-	11.25
10GBASE-T	caso 1	7.36
	caso 2	4.48
10GBASE-KX4	-	9.25
10GBASE-KR	caso 1	12.25
	caso 2	14.25

Tabella 3.6 – Limite temporale massimo per il parametro T_w [28].

3.6.2 Auto-negoziazione tra LPI Client e link partner

Affinché i due nodi collegati a un link possano conoscere le informazioni da utilizzare per effettuare il passaggio alla modalità di basso consumo, è necessario che in fase di configurazione del link venga compiuta la auto-negoziazione descritta nella sezione 3.1.
In questo modo le due stazioni sono in grado di comunicarsi automaticamente le proprie caratteristiche, permettendo la creazione del link e stabilendo se sia o meno possibile implementare i meccanismi descritti dallo standard EEE. Per far ciò, durante la autonegoziazione, i due nodi possono indicare se sono in grado di supportare tale standard, rendendo possibile il passaggio nello stato LPI solo nel caso in cui in entrambi siano implementate le funzionalità dell'EEE.

3.6.3 Consumo Energetico e Potenziale Risparmio

Come già accennato all'inizio di questo capitolo, il consumo energetico di un link durante il transito di pacchetti, cioè nello stato di ACTIVE, è circa equivalente a quello che si ha durante la fase di IDLE. Adottando lo standard EEE, ovvero portando il link nello stato di

Low Power Idle, si ha una riduzione del consumo di potenza che si può avvicinare al 90% e anche durante i brevi periodi di refresh si ha comunque una riduzione superiore al 10% [14].
In tabella 3.7 sono riassunti i consumi corrispondenti alle tipologie di reti trattate negli stati seguenti

ACTIVE consumo del nodo quando è completamente attivo;

IDLE consumo del nodo quando si trova in idle;

LPI_TX consumo del nodo quando viene disattivato solo il trasmettitore;

LPI_RX consumo del nodo quando viene disattivato solo il ricevitore;

LPI consumo del nodo quando si trova completamente in LPI

Protocollo	Consumo [W]				
	ACTIVE	IDLE	LPI_TX	LPI_RX	LPI
10BASE-T	0.504	0.334	0.252	0.151	0.076
100BASE-TX	0.388	0.320	0.185	0.124	0.06
1000BASE-T[2]	0.781	0.777	-	-	0.117
10GBASE-T	8.2	7.9	4.1	2.46	1.23

Tabella 3.7 – Consumo energetico delle reti Ethernet sotto carico, in idle ed in LPI [36, 4, 13].

Secondo una stima compiuta nel 2007 ancora una volta negli Stati Uniti, la percentuale di tempo durante il quale è possibile portare un link nello stato LPI può raggiungere l'80% con un conseguenti risparmio annuo di circa 7.5 TWh, corrispondeni ad un risparmio economico di ben 410 milioni di Dollari [14] ed una riduzione delle emissioni di CO_2 pari a circa 3.5 milioni di tonnellate [1].
Come si può dedurre da questi valori, il consumo di potenza da parte dei link Ethernet è decisamente aumentato rispetto ai valori stimati negli anni 2000 a causa della grande diffusione della tecnologia Ethernet.
Ciò rende essenziale l'adozione dello standard EEE in ogni link e sta focalizzando l'attenzione anche sui consumi delle reti industriali basate sullo standard Ethernet.

[1] Le reti 1000BASE-T non ammettono il passaggio parziale in LPI, quindi viene indicato solo il consumo del nodo quando si trova totalmente in LPI

Capitolo 4

Risparmio Energetico nelle Reti Powerlink

Il grande sviluppo delle reti Ethernet in ambiente industriale ha presto reso di centrale importanza il problema del consumo energetico dal momento che sono presenti milioni di link in tutto il mondo e, come per le normali reti Ethernet, per una buona parte del tempo in cui sono attivi, non vengono effettivamente utilizzati.
Di seguito verrà ripreso in esame il caso delle reti Ethernet POWERLINK (EPL), in cui la grande rigorosità di gestione del ciclo e la breve durata delle fasi di idle richiedono di adottare dei meccanismi di efficienza energetica che siano in grado di ridurre i consumi senza pregiudicare il determinismo delle trasmissioni. Per questo motivo si descriverà l'utilizzo, nelle reti EPL, dello standard EEE, che permette di portare i nodi in uno stato di basso consumo e successivamente riattivarli in tempi molto brevi, assicurando di rispettare le tempistiche del ciclo e di mantenere il comportamento real-time.

4.1 Reti EPL con EEE

Come descritto nel capitolo precedente, il ciclo EPL si compone di tre fasi fondamentali; la fase isocrona, la fase asincrona e la fase di idle. Quest'ultima, in particolare, costituisce il periodo temporale che interessa maggiormente lo spreco di energia, dal momento che non vi è alcun tipo di trasmissione tra i nodi della rete, mantenendo attivo un numero spesso elevato di link inutilmente. Proprio durante la fase di idle, quindi, si possono implementare i meccanismi presentati nello standard EEE, cioè si portano i nodi che non stanno trasmettendo in uno stato di LPI, riportandoli poi nello stato ACTIVE quando ricomincia la fase

isocrona e devono trasmettere nuovamente. Visti i valori tipici del tempo di ciclo, lo stato di basso consumo non viene imposto per lunghi periodi, come nel caso delle normali reti Ethernet, ma i nodi vengono portati in tale stato per brevi periodi nell'ambito del ciclo di funzionamento.
Dal momento che le reti industriali generalmente operano in maniera continua (o per periodi di tempo elevati), la somma di tutti questi brevi periodi in cui il consumo è ridotto, diviene una percentuale piuttosto rilevante della durata complessiva del periodo di lavoro, quindi è possibile ottenere dei risultati altrettanto soddisfacenti rispetto a quelli ottenuti con le reti Ethernet comuni.

Quando viene implementato lo standard EEE, generalmente, si suppone che siano i dispositivi stessi a determinare in quale momento portarsi nello stato di LPI, altrimenti sarebbe necessario trasmettere ad ognuno di essi un frame per informarli della necessità di cambiare stato, perdendo tempo inutilmente. In base a come è strutturato il ciclo di POWERLINK, però, dopo che è trascorsa la fase di idle, i nodi che si sono portati nello stato di basso consumo dovranno riattivare il link all'inizio del prossimo ciclo. Questa operazione, come si è visto nella sezione 3.4, richiede dalle decine alle centinaia di microsecondi in base alla velocità della rete.
Ciò introduce un ritardo nell'inizio del ciclo, che può portare alle violazioni temporali descritte nella sezione 2.2.4, poichè i nodi richiedono un tempo maggiore per accorgersi dei frame SoC e PReq. Questo ritardo, inoltre, si ripete ad ogni ciclo, di conseguenza si può creare una situazione in cui una o più stazioni non si rendono mai conto dell'inizio di un nuovo ciclo, rimanendo di fatto esclusi dalla rete EPL.
Una possibile soluzione potrebbe essere quella di sommare una stima di tale ritardo ai timer dei CN, in modo da allungare il timeout e permettere ad ogni stazione di ricevere correttamente i frame. Ciò, però, porta ad allungare gli slot associati ad ogni CN con una conseguente riduzione del tempo a disposizione per trasmettere dati.
Una seconda soluzione, che permette di assicurare il determinismo della rete, attuando anche gli algoritmi di riduzione del consumo dello standard EEE, è quella di integrare tali meccanismi con il controllo sul ciclo di POWERLINK effettuato dal MN, imponendo, cioè, che sia questo nodo a richiedere la riattivazione dei link della rete.
Sulla base del tipo di trasmissioni che devono compiere i vari CN è possibile gestire in maniera differente il passaggio nella modalità LPI, in modo tale da massimizzare la percentuale di tempo in cui si ha una riduzione dei consumi energetici. Di seguito verranno descritte le varie casistiche che si possono presentare, con la relativa gestione da parte del MN e con la percentuale di tempo in cui i nodi si possono trovare in LPI.

4.1.1 Caso generale

Il modo più semplice per implementare algoritmi di Energy Efficiency, senza pregiudicare il corretto svolgimento del ciclo, è quello di circoscrivere tutte le operazioni necessarie all'interno della fase di idle, imponendo che il passaggio allo stato di basso consumo avvenga dopo la fine della fase asincrona e che la riattivazione del link avvenga entro l'inizio della successiva fase isocrona.
I vari cambi di stato relativi allo standard EEE, compiuti dai nodi durante il ciclo di POWERLINK, sono rappresentati in figura 4.1, da cui si nota facilmente che le possibilità di ottenere un effettivo risparmio di energia dipendono direttamente dalla durata della fase di idle e dai tempi necessari a passare in LPI e ad attivare il link.

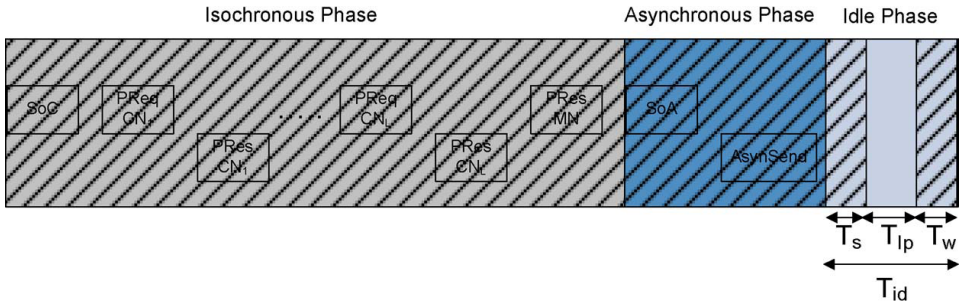


Figura 4.1 – Implementazione dello standard EEE durante la fase di idle.

Siano T_{EPL} la durata complessiva del ciclo, T_s il tempo necessario ai CN per entrare nello stato di Low Power Idle, T_w il tempo necessario a risvegliarsi e T_{idle} la durata dell'idle, che per ora viene considerata costante. Allora la percentuale di tempo durante la quale le stazioni si trovano nello stato di basso consumo energetico, risulta essere pari a

$$\alpha_{gen} = \frac{T_{idle} - T_s - T_w}{T_{EPL}}. \tag{4.1}$$

Ciò significa che il tempo durante il quale si ha un risparmio di energia ad ogni ciclo è pari al tempo di idle meno i tempi necessari a portare nello stato di basso consumo e riattivare il link. Per ottenere dei risultati soddisfacenti si deve avere che $T_{idle} \gg T_s + T_w$, condizione generalmente verificata per cicli di POWERLINK relativamente lunghi.
Normalmente, tuttavia, si cerca di ridurre al minimo i periodi di inattività della rete, pertanto si possono creare situazioni in cui la durata della fase di idle non sia sufficientemente lunga da permettere la disattivazione e successiva riattivazione dei link entro l'inizio del ciclo successivo; In questo caso non sarà possibile implementare i meccanismi dell'EEE.

Nel caso generale appena descritto, è possibile portare il link completamente in LPI,

disattivando sia il ricevitore che il trasmettitore di entrambi i nodi collegati. Infatti, nella fase di idle, non viene effettuata alcun tipo di trasmissione e non è necessario mantenere attivi i ricevitori dei nodi.

Passaggio parziale dei link allo stato LPI

Di seguito si andranno a considerare dei casi in cui i nodi hanno la possibilità di interrompere le trasmissioni e passare nella modalità LPI, prima dell'inizio della fase di idle. Si deve, però, tenere conto del fatto che le trasmissioni effettuate dai CN e dal MN sono di tipo broadcast e quindi, se si effettuasse un passaggio completo alla modalità di basso consumo energetico prima della fine delle trasmissioni, si otterrebbe un immediato risveglio del link, causato, ad esempio, dalla trasmissione di un frame PRes da parte di un altro CN. Per risolvere questo problema e aumentare comunque il tempo di permanenza in LPI è possibile utilizzare uno strumento messo a disposizione dalla standard EEE, il passaggio parziale in LPI. Per alcune tipologie di reti, come le 100BASE-TX e le 10GBASE-T, si ha la possibilità di disattivare le comunicazioni solo in una direzione, mantenendo attive le trasmissioni nell'altra. In questo modo è possibile andare a interrompere le trasmissioni da parte di un nodo, ridurre il consumo energetico del trasmettitore e, contemporaneamente, mantenere attivo il ricevitore fino alla fasi di idle, dove sarà possibile completare il passaggio in LPI. Per quanto riguarda le reti a 1000 Mbps, questa possibilità non è definita dallo standard, quindi non sarà possibile applicare le casistiche che seguono.

4.1.2 Nodi che trasmettono solo dati real-time

Nella sezione precedente si è supposto che la durata della fase di idle sia costante in realtà, come già osservato nella sezione 2.2.3, questa dipende dall'eventuale partecipazione delle stazione alla trasmissione di frame durante la fase asincrona.
La durata della fase asincrona dipende dalla presenza di richieste di trasmissioni asincrone da parte di uno o più CN durante il polling del MN. Essa si considera teoricamente terminata quando queste trasmissioni sono completate, ma, nel caso pratico, per tutti i CN che non sono coinvolti in questa fase, essa termina subito dopo che il frame SoA è stato ricevuto. Per massimizzare il risparmio energetico, quindi, conviene ridurre il più possibile la frequenza di polling dei CN che operano esclusivamente nella modalità asincrona. Come per il periodo del ciclo POWERLINK, la frequenza di polling di questi nodi dipende direttamente dal sistema in questione, dunque è necessario ricavarne il valore minimo sulla base delle specifiche date dal problema.

Un secondo accorgimento che permette di ridurre ulteriormente i consumi è quello di anticipare il passaggio allo stato di basso consumo per tutti i CN che non sono interessati dalle trasmissioni asincrone 4.2, portandoli nello stato LPI subito dopo la trasmissione del proprio frame PRes.

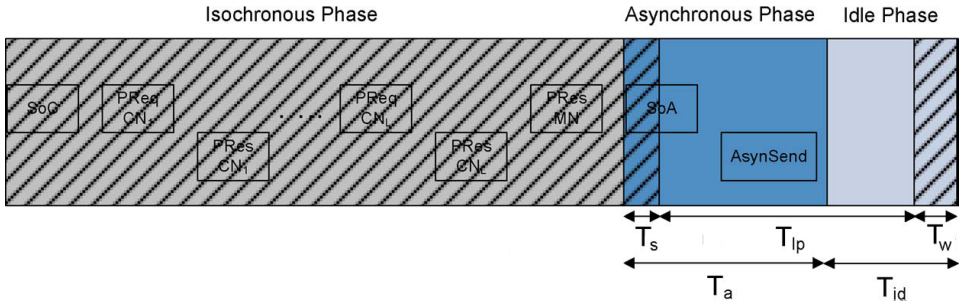


Figura 4.2 – Tempo in LPI per un CN che trasmette solo dati real-time.

Per queste stazioni, la percentuale di tempo durante la quale è possibile rimanere in uno stato di basso consumo è

$$\alpha_{isoc} = \frac{T_{async} + T_{idle} - T_s - T_w}{T_{EPL}}. \tag{4.2}$$

Dove T_{async} è la durata della fase asincrona. La percentuale ottenuta fa riferimento solo al passaggio parziale in LPI; quindi, per tenere conto del passaggio completo durante la fase di idle, basterà aggiungere il valore ottenuto con la formula relativa al caso generale.

4.1.3 CN che lavorano solo in modalità asincrona

Per quanto riguarda i CN che trasmettono esclusivamente frame asincroni, come si vede in figura 4.3, il periodo di tempo in cui è possibile passare nello stato di LPI è dato dall'intervallo tra la fine della fase asincrona del ciclo corrente e l'inizio della fase asincrona del ciclo successivo.

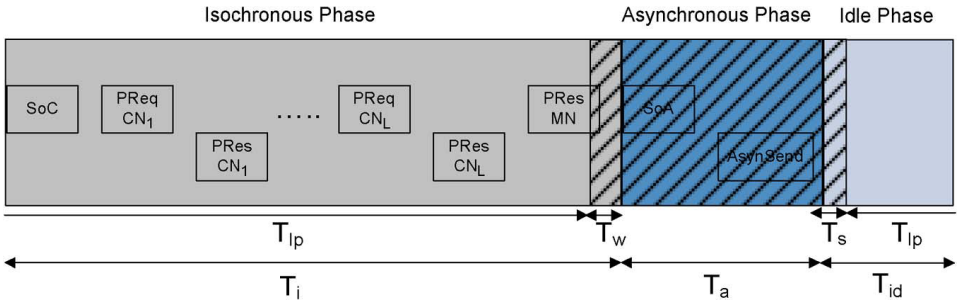


Figura 4.3 – Tempo in LPI per un CN che trasmette solo in modalità asincrona.

Pertanto la percentuale di tempo in cui si ha un risparmio energetico è data da

$$\alpha_{async} = \frac{T_{isoc} + T_{idle} - T_s - T_w}{T_{EPL}}. \tag{4.3}$$

Dove T_{isoc} è pari alla durata della fase isocrona. Anche in questo caso, il valore ottenuto fa riferimento al passaggio parziale in LPI si dovrà poi sfruttare la formula del caso generale per ottenere la percentuale complessiva di permanenza in tale modalità.

4.1.4 CN che lavorano in slot temporali multiplati

Nel caso particolare dei CN che vengono interrogati durante il polling del MN solo in multipli del periodo del ciclo EPL, è possibile estendere il periodo in cui passare nello stato di basso consumo a tutto il tempo che intercorre tra due accessi consecutivi. Sia k la periodicità con cui la stazione viene coinvolta attivamente nel ciclo POWERLINK, allora essa potrà trovarsi nello stato LPI per $k - 1$ cicli, come si può intuire dalla figura 4.4.

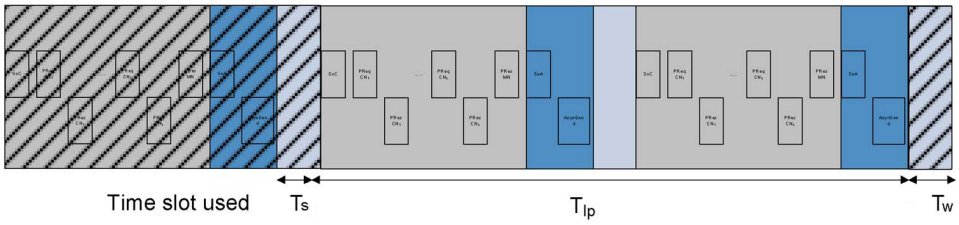


Figura 4.4 – Tempo in LPI per un CN con slot multiplati con k = 3.

Per poter stimare la percentuale di tempo durante la quale questi nodi portano ad un effettivo risparmio di energia, però, non è sufficiente supporre che i CN in questione entrino a far parte del polling effettuato dal MN solo in alcuni cicli, ma è anche necessario tenere conto del fatto che potrebbero voler trasmettere frame asincroni o che potrebbero essere interessati da trasmissioni incrociate tra le stazioni. Questa particolare tipologia di trasmissioni fa riferimento a tutte le possibili comunicazioni dirette che vi possono essere tra le stazioni senza coinvolgere il MN, ad esempio lo scambio di informazioni che non necessitano di essere elaborate dall'unità centrale e che sono direttamente utilizzabili da altre stazioni.

Per come è stato definito lo standard EEE, quando una stazione si trova nello stato di basso consumo e riceve un frame ad essa indirizzato riattiva automaticamente tutte le funzionalità della scheda di rete, uscendo dallo stato LPI e riportandosi allo stato ACTIVE. Non potendo prevedere a priori se dei CN debbano riattivarsi durante cicli di POWERLINK a cui teoricamente non sono assegnati, di seguito sarà ancora una volta necessario supporre

che sia possibile mantenere i nodi in LPI parziale per tutta la durata dei cicli e delle fasi a cui non devono partecipare, per poi completare il passaggio durante le varie fasi di idle.
Supponendo che una di queste stazioni partecipi sia alla fase isocrona sia a quella asincrona del ciclo, sotto le ipotesi appena descritte, la percentuale di tempo in cui le stazioni che lavorano in slot temporali multiplati possono portarsi nello stato di basso consumo è

$$\alpha_{tm} = \frac{(k-1)\cdot T_{EPL} + T_{idle} - T_s - T_w}{k \cdot T_{EPL}}. \tag{4.4}$$

Nel caso in cui la stazione non necessiti di trasmettere frame asincroni neppure nei cicli a cui è assegnata, la percentuale diviene

$$\alpha_{tm,isoc} = \frac{(k-1)\cdot T_{EPL} + T_{async} + T_{idle} - T_s - T_w}{k \cdot T_{EPL}} = \frac{(k-1) + \alpha_{isoc}}{k}. \tag{4.5}$$

Infine, per stazioni che operano esclusivamente nella fase asincrona

$$\alpha_{tm,async} = \frac{(k-1)\cdot T_{EPL} + T_{isoc} + T_{idle} - T_s - T_w}{k \cdot T_{EPL}} = \frac{(k-1) + \alpha_{async}}{k}. \tag{4.6}$$

4.1.5 Riattivazione del link solo durante le trasmissioni

Il passaggio parziale del link nello stato di basso consumo può essere ulteriormente sfruttato per le stazioni che operano normalmente durante le fasi isocrone ed asincrone; infatti, grazie alla conoscenza ben precisa delle tempistiche che caratterizzano un sistema deterministico come la rete POWERLINK, è conveniente portare nello stato di basso consumo i link a cui sono collegate le stazione parzialmente, quando le trasmissioni avvengono in una sola direzione. Ad esempio, ogni link che collega un CN al MN può essere portato nello stato di basso consumo nella porzione che direziona le comunicazione verso il MN, ad eccezione del tempo necessario a trasmettere il frame PRes di risposta al MN e un eventuale frame durante la fase asincrona.
Considerando come esempio un CN che opera esclusivamente nella fase isocrona di ogni ciclo POWERLINK, si calcola la percentuale di tempo in cui il link si trova nello stato di basso consumo nella direzione dal CN al MN.

$$\begin{aligned} \alpha_{isoc}^{CN\rightarrow MN} &= \frac{T_{EPL} - T_s - T_w - T_{trasm}}{T_{EPL}} \\ &= \alpha_{isoc} + \frac{T_{isoc} - T_{trasm}}{T_{EPL}} \end{aligned} \tag{4.7}$$

Dove T_{trasm} rappresenta il tempo necessario a trasmettere il frame di risposta PRes al MN, come rappresentato in figura 4.5.

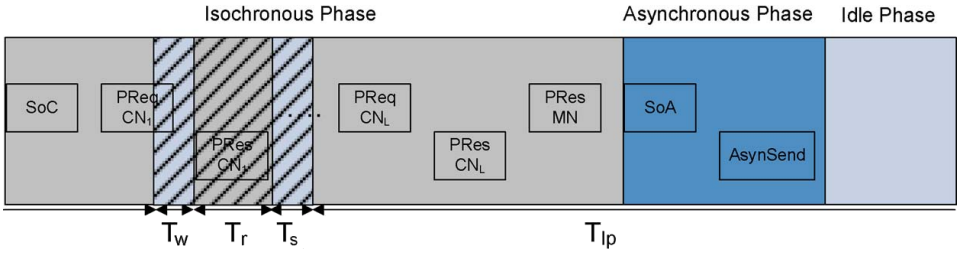


Figura 4.5 – Tempo in LPI per un link nella direzione $CN \rightarrow MN$, con un CN che trasmette solo dati real-time.

Introducendo la possibilità di mantenere il trasmettitore dei CN disabilitato fino a quando è necessario compiere la trasmissione del frame PRes o ASnd, si ottengono i risultati riassunti in tabella 4.1, in cui vengono indicate le percentuali di tempo in LPI parziale per i casi presi in considerazione in precedenza.

Tipo di CN	$CN \rightarrow MN$
Generale[1]	$\alpha_{gen}^{CN \rightarrow MN} = \alpha_{gen} + \frac{T_{isoc}+T_{async}-T_{trasm}}{T_{EPL}}$
Solo isocrono	$\alpha_{isoc}^{CN \rightarrow MN} = \alpha_{isoc} + \frac{T_{isoc}-T_{trasm}}{T_{EPL}}$
Solo asincrono[2]	$\alpha_{async}^{CN \rightarrow MN} = \alpha_{async} + \frac{T_{async}-T_{trasm}}{T_{EPL}}$
Slot multiplati	$\alpha_{tm}^{CN \rightarrow MN} = \alpha_{tm} + \frac{T_{isoc}+T_{async}-T_{trasm}}{k \cdot T_{EPL}}$

Tabella 4.1 – Percentuali di tempo nello stato di basso consumo nei vari casi presi in considerazione.

Nel caso delle rete 1000BASE-T non è possibile sfruttare questa caratteristica, riducendo in maniera significativa il risparmio ottenibile, dal momento che è possibile portare un link nello stato LPI solo quando non è necessario trasmettere in entrambe le direzioni, cioè durante la fase di idle. Si potrà, quindi, applicare solamente il caso generale.

[2]In questo caso T_{trasm} costituisce il tempo necessario a trasmettere il frame PReq e l'eventuale frame asincrono

[3]In questo caso T_{trasm} costituisce il tempo necessario a trasmettere il frame asincrono

4.1.6 Risparmio energetico

Una volta determinata la percentuale di tempo in cui i vari nodi della rete si possono trovare nello stato di basso consumo energetico, è possibile procedere calcolando a quanto corrisponda l'effettivo risparmio derivante dall'adozione dello standard EEE in una rete POWERLINK.
Siano P_{EPL} la potenza consumata da una normale rete POWERLINK e P_{EEE} la potenza consumata dalla stessa rete, in cui però si adottano i meccanismi descritti per consumare meno energia. Indicando genericamente con la lettera α, la percentuale di tempo in LPI, allora la percentuale di energia risparmiata è pari a:

$$S = 1 - \frac{\alpha \cdot P_{EEE} + (1 - \alpha) \cdot P_{EPL}}{P_{EPL}} \cdot 100 = \alpha \cdot \left(1 - \frac{P_{EEE}}{P_{EPL}}\right) \cdot 100 \tag{4.8}$$

Dal momento che, in molti casi, si compie il passaggio parziale in LPI durante il ciclo per poi completarlo nella fase di idle, è necessario calcolare il risparmio ottenuto dal passaggio parziale e, successivamente, aggiungere quello derivante dalla disattivazione del ricevitore, a cui corrisponde un valore di α ottenibile con la formula del caso generale. Il valore assunto dal parametro α, che dipende dal tipo di CN presenti nella rete, risulta essere fondamentale per determinare quale sia la configurazione migliore per ottenere il risparmio massimo.
Dal momento che gli accorgimenti da adottare per aumentare il più possibile tale parametro fanno riferimento alle caratteristiche dei singoli CN, è possibile gestire in maniera differente i vari nodi di una stessa rete, così da ottenere il massimo contributo da ognuno di essi.

Capitolo 5

Simulazioni in OMNeT++

OMNeT++ è un framework basato sulla programmazione orientata ad oggetti in C++ [43], è dotato di un aspetto modulare ed è in grado di simulare qualsiasi dipo di rete ad eventi discreti.
Ha un'architettura molto generica, quindi può essere sfruttato per trattare numerose problematiche

- modellare reti wireless e cablate;
- modellare protocolli;
- modellare reti di code;
- modellare sistemi hardware distribuiti come i multiprocessori;
- testare architetture hardware;
- testare le performance di sistemi software;
- modellare e simulare qualsiasi tipo di sistema ad eventi discreti.

Le varie entità del sistema che si desidera simulare vengono definite sottoforma di moduli, che possono essere combinati tra loro e messi in comunicazione. Non vi è limite alla profondità a cui si può arrivare nella modellizzazione, dal momento che è possibile gestire qualsiasi aspetto di interesse del sistema, tralasciando eventualmente le caratteristiche o gli eventi meno rilevanti per la propria simulazione.

Seguendo il procedimento adottato fino ad ora nella trattazione teorica del problema, si è deciso di partire da una implementazione dello standard Ethernet, e si è proseguito estendendolo ed introducendo i vari meccanismi caratteristici delle rete Ethernet POWERLINK (EPL) e dello standard Energy Efficiency Ethernet (EEE).
Per far ciò si è utilizzato il framework INET 2.0.0 [2], cioè un package open source, contenente i modelli relativi a gran parte dei protocolli di rete wireless e cablate.
In particolare le componenti di INET utilizzate per le simulazioni sono:

- livelli Data Link e MAC di Ethernet;
- modello degli hub Ethernet;
- modello dei nodi Ethernet;

Le numerose interfacce messe a disposizioni da OMNeT++ permettono di definire agevolmente una rete, come quella molto semplice rappresentata in figura 5.1 , all'interno della quale vengono inseriti i modelli necessari, come un hub o le stazioni Ethernet, a loro volta modellate come rappresentato in figura 5.2.

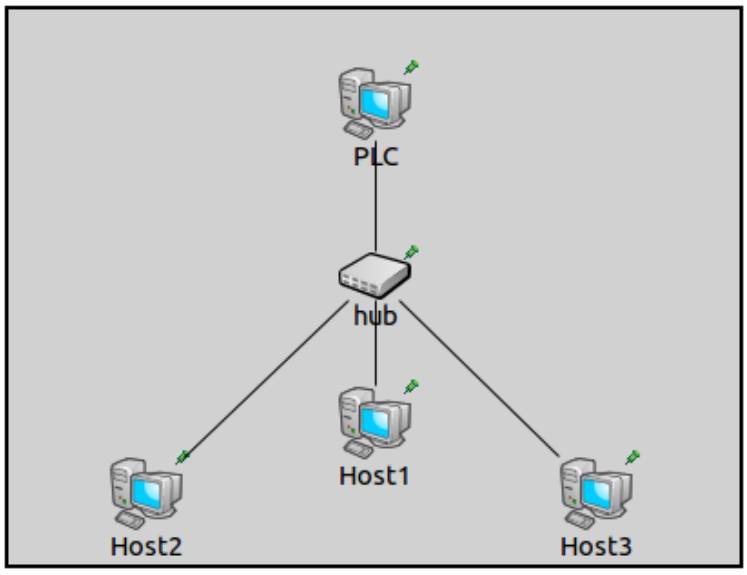


Figura 5.1 – Esempio di una semplice rete Ethernet simulata con OMNeT++.

All'interno di ogni stazione vengono definiti i due livelli che compongono il Data Link, cioè il livello MAC collegato al PHY, il Logical Link Control (LLC) e due ulteriori moduli, il *cli*, che si occupa di gestire la generazione di pacchetti da parte del nodo, e il *srv*, che ha il compito di gestire la ricezione di pacchetti da altri nodi ed eventualmente di preparare e

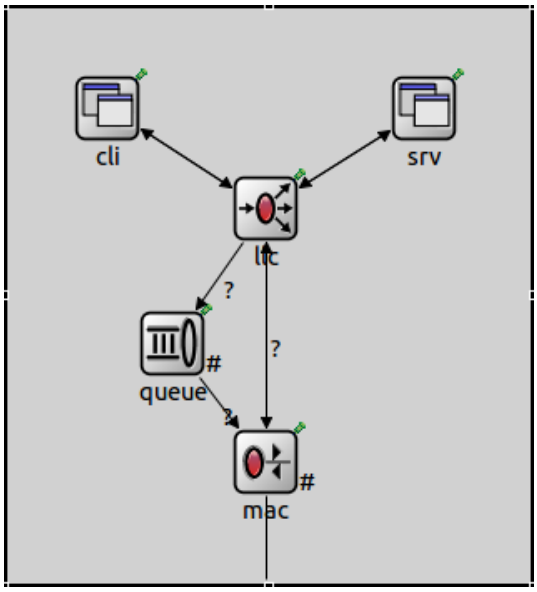


Figura 5.2 – Modello di un nodo Ethernet del framework INET.

trasmettere un frame di risposta.
Come si può notare, non vengono modellizzati i livelli superiori, che sono però disponibili nel framework. Ciò è dovuto al fatto che i vari meccanismi adottati nello standard EEE e nelle reti EPL vanno ad intervenire soprattuto nei due livelli rappresentati e sono sufficienti i due moduli *cli* e *srv* per simulare a generazione di pacchetti con tempistiche ben precise.
Per quanto riguarda i canali di comunicazione è sufficiente definire la velocità ed eventualmente un ritardo di propagazione sarà poi il simulatore stesso a gestire il transito dei pacchetti, i tempi necessari ad arrivare a destinazione e ad indicare la presenza di eventuali collisioni.

5.1 Simulazione di una Rete Powerlink

Per creare il modello di una rete EPL è anzitutto necessario definire i due nodi fondamentali, ovvero il Managing Node (MN) ed il Controlled Node (CN). Entrambi sono ottenuti estendendo i comuni nodi Ethernet precedentemente descritti ed implementando i meccanismi che assicurano un comportamento real-time del sistema.
Per definire le azioni compiute dai vari moduli di OMNeT++, è necessario associare al modulo stesso un programma scritto in C++, in cui è possibile definire i parametri principali dell'entità in questione e tutte le attività da essa svolte, sia autonomamente sia in risposta ad un evento esterno.

Il modo di programmare questi moduli ricalca l'aspetto ad eventi discreti del sistema; infatti, per ciascuna operazione da compiere, è possibile definire una condizione di avanzamento ben precisa, in corrispondenza della quale vengono eseguite tutte le istruizioni che descrivono il comportamento che il modulo in questione deve adottare. Sulla base delle considerazioni appena fatte, è chiaro che, prima di procedere con la programmazione vera e propria, risulta conveniente dare una rappresentazione il più schematica possibile del susseguirsi di tutte le possibili condizioni ed azioni corrispondenti che coinvolgono i nodi della rete.
Lo strumento scelto per far ciò sono le reti di Petri [16], in cui è possibile descrivere un sistema ad eventi discreti utilizzando due elementi, i posti e le transizioni. I posti corrispondono allo stato in cui si trova il sistema in questo momento, ad esempio quando un nodo sta compiendo una certa operazione o quando si trova in attesa di un qualche evento. Questi sono collegati a delle transizioni, che corrispondono alle condizioni che devono verificarsi per poter compiere una azione ben precisa, cioè per potersi spostare dal posto corrente a quello successivo.
Ad ogni posto viene poi associato un numero intero, definito marcatura, che in questo caso specifico, viene utilizzato esclusivamente per definire in quale degli stati si trova il sistema, ovvero assume il valore uno se il sistema si trova nel posto corrispondente alla marcatura, altrimenti assume il valore zero.
Sfruttando le reti di Petri, si procede con la rappresentazione dei due nodi della rete EPL: in particolare, per quanto riguarda il MN, si rappresenta il funzionamento del suo modulo *cli*, che gestisce la trasmissione dei frame di sincronizzazione, mentre per il CN, si va a rappresentare il modulo *srv*, dal momento che è un nodo passivo e deve generare pacchetti esclusivamente in risposta ad una richiesta del MN.

5.1.1 Rete di Petri del Managing Node

Come descritto nella sezione 2.1.2, il MN si occupa di gestire gli altri nodi della rete, trasmettendo i frame di sincronizzazione e compiendo il polling durante la fase isocrona. La sequenza di operazioni compiute dal MN è descritta dalla rete di Petri in figura 5.3, in cui si nota un posto iniziale, denominato *p-Idle*, dove il nodo rimane in attesa di poter iniziare un ciclo EPL.
Quando la condizione della transizione T_{EPL} viene verificata, cioè quando scatta un timer di valore pari al periodo, è possibile procedere nel posto *p-SoC*, in cui si da inizio al ciclo con la preparazione e la trasmissione broadcast del frame SoC. La transizione successiva, cioè *t-SoC*, rappresenta l'attesa del tempo di propagazione necessario al completamento di tale trasmissione.

Si procede, a questo punto, con il polling isocrono, corrispondente ad un ciclo ripetuto per il numero di CN della rete, in cui il posto *p-PReq* viene attraversato ciclicamente quando avviene la trasmissione di un frame PReq. Si ripete il passaggio in questo posto in corrispondenza dello scatto della transizione *t-PRes*, ovvero nel caso in cui venga ricevuto il frame PRes o allo scadere del timeout, mentre si ha l'uscita dal ciclo quando viene interrogato l'ultimo nodo della rete e scatta la transizione *Ncn*.
Questa transizione porta alla conclusione della fase isocrona e, quindi, all'inizio della fase asincrona, che combacia con la trasmissione del frame SoA e l'attesa della sua ricezione da parte dei CN, rappresentate rispettivamente dal posto *p-SoA* e dalla transizione *t-SoA*. Una volta scattata quest'ultima transizione, il MN si porta nel posto *p-ASnd* rimanendo in ascolto nella rete fino a quando avviene la ricezione del frame asincrono, che causa l'avanzamento nella transizione *t-ASnd* e il ritorno al posto iniziale.

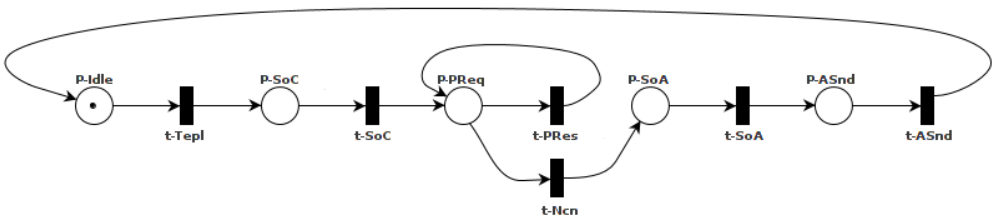


Figura 5.3 – Rete di Petri del modulo *cli* del MN.

5.1.2 Rete di Petri del Controlled Node

Anche nel caso del CN, si definisce un programma sequenziale, rappresentato dalla rete di Petri in figura 5.4.
I vari CN rimangono inizialmente in attesa dell'inzio del ciclo EPL a cui devono partecipare, rimanendo nel posto *p-Idle* fino a allo scatto della transizione *t-isoc* o della transizione *t-Async*, che rappresentano rispettivamente l'avanzamento, dopo la ricezione del frame SoC, alla fase isocrona per i nodi che devono parteciparvi o a quella asincrona per i nodi che non effettuano trasmissioni real-time. Nel caso in cui si debba effettuare una trasmissione isocrona il nodo passa nel posto *p-PReq*, dove rimane in attesa della ricezione del proprio frame PReq, successivamente può procedere oltre la transizione *t-PReq* e portarsi nel posto *p-PRes*, corrispondente alla fase di elaborazione e trasmissione del frame di risposta PRes. La transizione successiva *t-PRes* rappresenta il tempo necessario ad effettuare la trasmissio-

ne, che è seguita dall'inizio della fase asincrona. Tutti i CN che hanno effettuato trasmissioni nella fase isocrona o che sono passati direttamente alla fase successiva, si trovano ora nel posto *p-SoA* e, una volta ricevuto il frame SoA, possono percorrere due strade differenti. La prima viene seguita nel caso in cui il nodo non sia stato selezionato per la trasmissione asincrona o non abbia richiesto di effettuarla e corrisponde a tornare immediatamente nel posto iniziale con lo scatto della transizione *t-SoAi* mentre, la seconda viene seguita quando il nodo è stato scelto dal MN e corrisponde allo scatto della transizione *t-SoAa* con il conseguente passaggio nel posto *p-ASnd*. A questo punto il nodo selezionato si prepara alla trasmissione del frame ASnd, attende lo scatto della transizione *t-ASnd*, rappresentata dal tempo di propagazione, e si riporta anch'esso nel posto iniziale, corrispondente alla fase di idle.

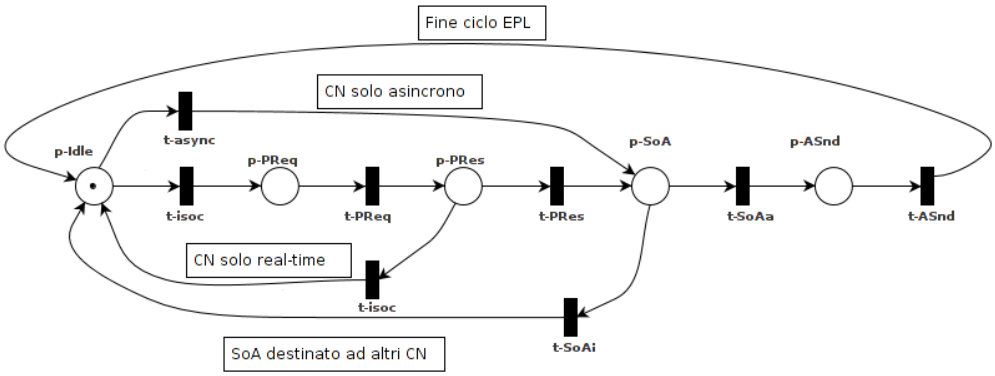


Figura 5.4 – Rete di Petri del modulo *srv* del CN.

5.1.3 Implementazione e Verifica del Modello

Una volta definiti i comportamenti assunti dai vari nodi della rete, è possibile programmare i moduli dei nodi e creare una rete di prova per verificare l'effettivo funzionamento dei meccanismi e per determinare un valore del periodo T_{EPL} che sia sufficientemente grande da assicurare il polling di tutti i nodi e la trasmissione asincrona.
Negli algoritmi 1 e 2, che seguono, si possono vedere le strutture del modulo *cli* per quanto riguarda il MN e *srv* per i nodi CN. Le funzioni *selfMsg('nome', t)* e *send('nome')* permettono rispettivamente di trasmettere dei messaggi denominati *self message* dopo un tempo *t* e di trasmettere un frame broadcast, mentre le funzioni *isSelfMsg(msg)*, *isPRes(msg)*, *isPReq(msg)* e *isSoA(msg)* permettono di terminare il tipo di messaggio ricevuto dal nodo in questione. Infine la funzione *getSoA(msg)* permette di estrapolare dal frame SoA il codi-

ce identificativo del CN selezionato per la fase isocrona, mentre la funzione *isFrame(msg)* permette di verificare se il messaggio ricevuto è proveniente da un nodo esterno.

Algoritmo 1 Gestione ciclica dei messaggi (*msg*) ricevuti dal modulo *cli* del MN

```
n ← 0
selfMsg('startEPL', 0)
name ← getName(msg)
if isSelfMsg(msg) then
    if name = 'startEPL' then // inizio del ciclo
        send('SoC')
        selfMsg('startEPL', T_EPL)
        selfMsg('PReq', 0)
    else if name = 'PReq' then
        n ← n + 1 // passo al nodo successivo
        if n = N_CN then
            Scheduling CN i-esimo
            send(SoA_i)
        else
            send('Pres_n')
            selfMsg('waitPRes', timeout)
        end if
    else if name = 'waitPRes' then
        selfMsg('PReq', 0)
    end if
else
    if isPRes(msg) then
        selfMsg('waitPRes', 0)
    end if
end if
```

Questi algoritmi sono costituiti da una sequenza di costrutti *if*, le cui condizioni rappresentano le transizioni delle reti di Petri precedentemente illustrate, mentre l'insieme di istruzioni che seguono a queste condizioni rappresentano le operazioni relative ai posti in cui procede il sistema in seguito a queste transizioni.
Per la generazione dei timer che permettono l'avanzamento nelle varie transizioni, si sfrutta il meccanismo dei *self message* messo a disposizione da OMNeT++. Si tratta di messaggi temporizzati che un modulo è in grado di trasmettere a se stesso per notificare un evento, ad esempio che è passato il tempo necessario alla trasmissione di un frame o che è avvenuta la ricezione di un frame dall'esterno che potrebbe aver sbloccato una delle condizioni pendenti dei vari *if*.

Algoritmo 2 Gestione dei messaggi (*msg*) ricevuti dal modulo *srv* del CN i-esimo

```
name ← getName(msg)
if isSelfMsg(msg) then
    if name = 'PRes' then // fase isocrona
        send('PRes')
    else if name = 'ASnd' then // fase asincrona
        send('ASnd')
    end if
else
    if isPReq(msg) then
        selfMsg('PRes',0)
    else if isSoA(msg) and getSoA(msg) = i then
        selfMsg('ASnd',0)
    end if
end if
```

Tenendo presente tutte le condizioni di attesa in cui un nodo potrebbe trovarsi e gestendo correttamente le operazioni che ne conseguono, è quindi possibile andare a costruire una rappresentazione realistica delle due tipologie di stazioni EPL, in grado di gestire autonomamente la ricezione, la generazione e la trasmissione di frame attraverso la rete e assicurando il determinismo dell'intero sistema.
La rete scelta è quella già rappresentata in figura 5.1, in cui al nodo PLC viene assegnato il ruolo di MN, mentre i vari host assumono il ruolo di CN. Si continua ad utilizzare un comune dispositivo hub Ethernet; infatti nello standard EPL si consiglia di adoperare questi dispositivi, che introducono dei ritardi praticamente trascurabili rispetto a quelli introdotti da uno Switch. Per verificare se tutti i frame caratteristici delle reti EPL vengono scambiati nel modo corretto ed entro i tempi stabiliti, è possibile utilizzare una della tante statistiche messe a disposizione da OMNeT++, cioè il Sequence Chart, in cui è tenuto traccia di tutti i messaggi scambiati tra i vari nodi e degli istanti temporali in cui questi trasferimenti avvengono.

Come si può notare dalla figura 5.5, tutti i frame di sincronizzazione vengono trasmessi broadcast ai nodi CN, inoltre anche le trasmissioni dei frame PReq e PRes avvengono correttamente ed il frame ASnd viene trasmesso dal nodo CN designato per la fase asincrona, senza che si vada a sforare nel ciclo successivo.

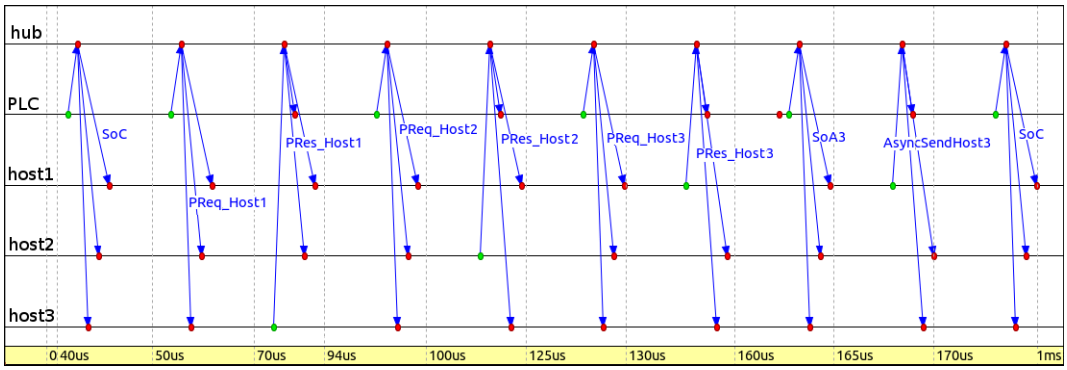


Figura 5.5 – Sequence Chart relativo allo scambio di pacchetti in una rete EPL

5.2 Simulazione di un Nodo con EEE

Prima di procedere con l'implementazione di algoritmi per il risparmio energetico nelle reti EPL, è conveniente adottarli all'interno di una comune rete Ethernet, per poi introdurre i meccanismi descritti nella sezione precedente.
La rete di Petri in figura 5.6 rappresenta la sequenza di operazioni che un nodo EEE deve compiere per potersi portare nello stato LPI e successivamente per tornare nella modalità ACTIVE.

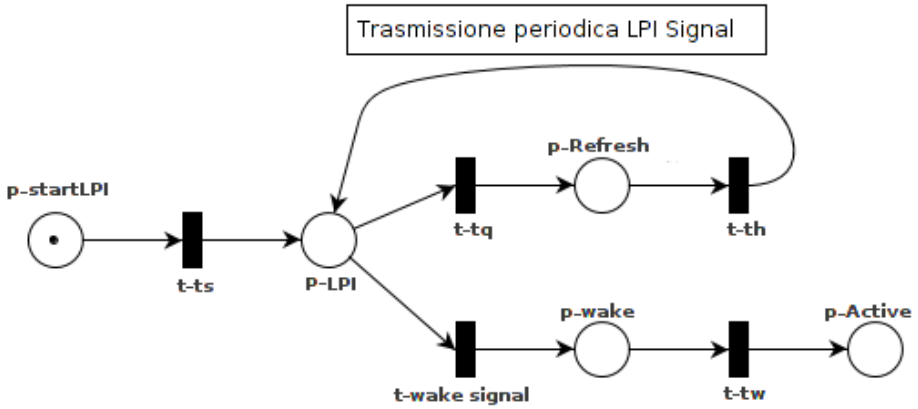


Figura 5.6 – Rete di Petri di un nodo che implementa lo standard EEE.

L'evento che da l'avvio alla procedura per passare alla modalità LPI è la trasmissione di un *self message* denominato *sleep signal*, che viene rappresentata dal posto *p-startLPI*. La prima transizione, denominata *t-ts*, corrisponde all'attesa del tempo necessario al nodo per spegnere la porzione del trasmettitore non necessaria e per portarsi effettivamente in

uno stato di basso consumo. Trascorso questo tempo il nodo inizia la trasmissione ciclica degli LPI signal. Esso, infatti, si porta nel posto *p-LPI*, attende il periodo prefissato t_q corrispondente alla transizione *t-tq* e procede con l'operazione di refresh, portandosi nel posto *p-refresh*. Dopo un tempo t_h, necessario ad accendere temporaneamente alcune funzionalità del trasmettitore, si ha lo scatto della transizione *t-th*, che riporta il nodo nel posto relativo alla trasmissione degli LPI signal. Questo comportamento ciclico continua fino a quando viene generato il segnale *wake signal*, che può essere dovuto ad una decisione autonoma da parte del nodo in questione o alla ricezione di un qualsiasi frame da parte di un altro nodo della rete e che causa lo scatto della transizione *t-wake signal*.
Una volta generato questo segnale, ha inizio la procedura di riattivazione del trasmettitore, rappresentata dall'attesa nel posto *p-wake* per un tempo t_w, dopo il quale si ha l'avanzamento nella transizione *t-tw* e il passaggio nel posto *p-Active*, con il conseguente ritorno nello stato ACTIVE da parte del nodo.

5.2.1 Implementazione e Verifica del Modello

In questo caso particolare sarebbe necessario implementare la parte di operazioni necessarie ad entrare nella modalità LPI, nel modulo *cli*, e la parte relativa alla ricezione di eventuali frame esterni e alla conseguente riattivazione del link, nel modulo *srv*. Per semplicità si è deciso di compiere tutte le operazioni inerenti allo standard EEE nel modulo *cli* dei nodi, permettendo al modulo *srv* di gestire il normale traffico nella rete e, se il nodo si trova in LPI, di notificare al modulo *cli* la ricezione di un frame.
Nell'algoritmo 3, si vedono le istruzioni implementate in questo modulo, dove il nodo in questione, decide di portarsi in LPI trasmettendo il self message *startLPI*, attende un tempo t_s prima di avviare la trasmissione degli LPI signal separati tra loro dal tempo t_q e avvia la riattivazione quando è verificata una condizione particolare o quando viene ricevuto un frame dall'esterno.

Algoritmo 3 Gestione ciclica dei messaggi (*msg*) ricevuti dal modulo *srv* di un nodo EEE

```
selfMsg('startLPI', 0)
if isSelfMsg(msg) then
    if msg = 'startLPI' then // passaggio in LPI
        disattivazione trasmettitore
        selfMsg('sendLPI', t_s)
    else if msg = 'sendLPI' then
        send('LPISignal')
        selfMsg('sendLPI', t_q)
    else if msg = 'wakeUp' or < condizione riattivazione > then
        riattivazione trasmettitore
    end if
else
    if isFrame(msg) then // risveglio per ricezione di un frame
        selfMsg('wakeUp', 0)
    end if
end if
```

5.3 Simulazione di una Rete EPL con Standard EEE

La fase conclusiva di questo progetto consiste nell'unire i due modelli appena creati in modo tale da ottenere la simulazione di una rete EPL che implementi lo standard EEE.
In particolare, il MN e i CN compieranno esattamente le stesse operazioni descritte nella sezione 5.1, sfruttando anche i meccanismi dell'EEE per portarsi in LPI durante i periodi di idle.
Si tratta, quindi, di introdurre i *self message* e le condizioni presenti nei nodi EEE all'interno dei CN di una rete EPL, facendo in modo che avviino la procedura per entrare in LPI non appena si conclude l'ultima fase del ciclo a cui devono partecipare ed inizia la fase di idle.
Il tipo di programmazione adottata rende molto semplice l'unione di questi due standard; infatti le porzioni di codice eseguiti al verificarsi delle varie condizioni di attesa generano automaticamente dei *self message*, che vengono utilizzati dal modulo stesso per capire quale sia la prossima sequenza di istruzioni da compiere per avanzare nel ciclo.
Pertanto, è sufficiente riportare il codice presente nel modulo *cli* dei nodi EEE in coda all'algoritmo utilizzato dal modulo *srv* dei CN, gestendo diversamente l'utilizzo dei *self message* 'startLPI' e 'wakeUp'. Il primo dovrà essere inviato negli istanti in cui il nodo può portarsi in LPI, ad esempio dopo la trasmissione del frame PRes o del frame ASnd; mentre il secondo dovrà essere inviato in corrispondenza dell'inizio di un nuovo ciclo EPL, cioè subito prima della ricezione del frame SoC.

L'algoritmo associato al MN, invece, rimane pressochè invariato, visto che si suppone che questo nodo rimanga sempre attivo e si limiti a trasmettere un frame a tutti gli altri nodi della rete poco prima dell'inizio di un nuovo ciclo, in modo da riattivarli prima della trasmissione del frame SoC.

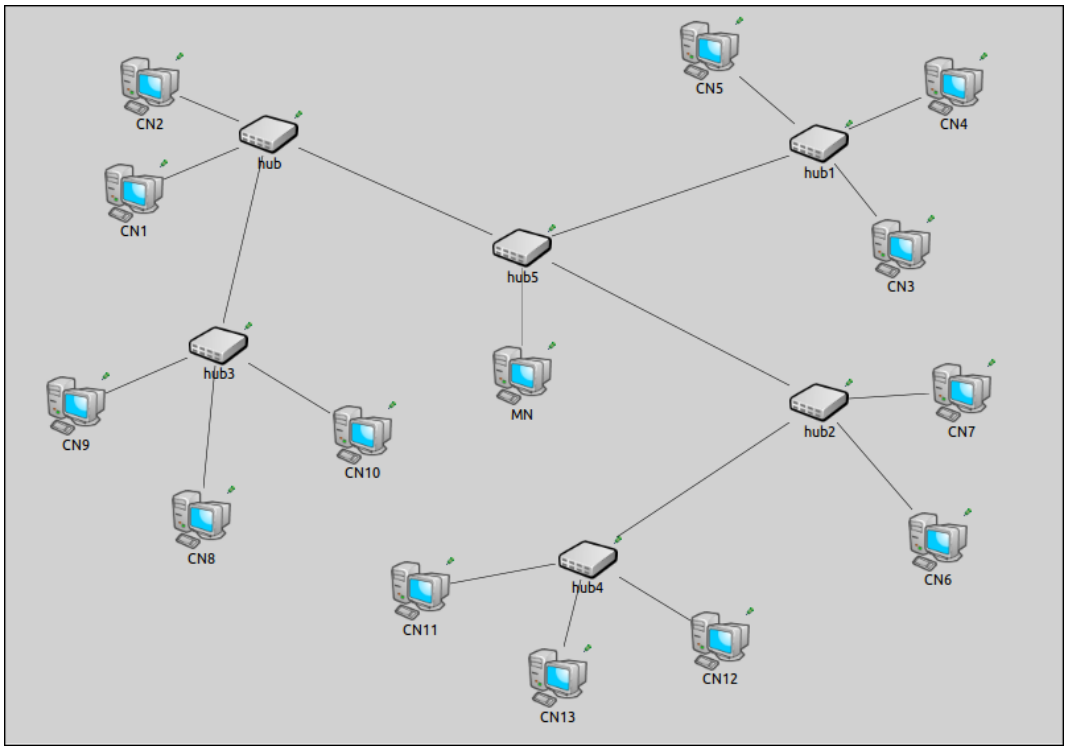


Figura 5.7 – Rete EPL con topologia ad albero.

Per ottenere delle simulazioni più significative, si è deciso di creare una rete più estesa e complessa, in modo da poter vedere le potenziali riduzioni del consumo energetico in una rete che effettivamente potrebbe essere adottata nella realtà.
La rete utilizzata è quella rappresentata in figura 5.7, in cui è presente un MN e 13 CN, collegati tra loro tramite degli hub Ethernet. Per poter studiare l'effetto che possono avere più livelli di hub sul ciclo EPL, si è scelto di adottare una topologia ad albero, in cui è necessario tenere conto del fatto che, per quanto riguarda i link di collegamento tra due hub, non è possibile il passaggio alla modalità LPI, dal momento che questi dispositivi non implementano lo standard EEE.

Il primo passo da compiere per poter procedere con le simulazioni è quello di determinare un periodo di ciclo T_{EPL} adeguato, sulla base del numero di nodi considerati, della velocità dei link e della dimensione dei pacchetti scambiati.

Come già accennato nella sezione 2.1.4, verranno utilizzati frame EPL della dimensione di 45 byte, incapsulati in frame Ethernet di dimensione complessiva pari a 72 byte. Questo valore corrisponde al numero di byte di un frame di sincronizzazione di POWERLINK e si è scelto, per semplicità, di adeguare a tale dimensione anche i frame del polling e della fase asincrona, che generalmente hanno una lunghezza massima pari a 1526 byte. Potendo stimare solamente i tempi di trasmissione dei frame, che vengono nuovamente riportati in tabella 5.1, e non avendo a disposizione una stima accurata di quelli necessari alla loro elaborazione, si è deciso di adottare il valore proposti in [36] come tempo che intercorre tra la trasmissione di un frame PReq e la ricezione del corrispondente PRes. Sulla base di questi tempi e delle equazioni descritte nella sezione 2.2.4 è possibile dimensionare agevolmente il periodo del ciclo EPL.

Di seguito sono presentati i calcoli approssimativi per una rete a 100 Mbps, mentre in tabella 5.1 vengono riassunti i risultati ottenuti in maniera analoga per le altre due casistiche.

$$
\begin{aligned}
T_{EPL} &= t_{isoc} + t_{async} + t_{idle} \qquad (5.1)\\
t_{PRes,PReq} &= 2 \cdot t_{elab} + 2 \cdot t_{trasm} \simeq 40\mu s\\
t_{poll} &= N_{CN} \cdot t_{PRes,PReq} \simeq 15 \cdot 40 \cdot 10^{-6} = 600\mu s\\
t_{isoc} &= t_{poll} + t_{SoC} \simeq t_{poll} + t_{elab} + t_{trasm} = 620\mu s\\
t_{async} &= t_{SoA} + t_{ASnd} \simeq 2 \cdot t_{elab} + 2 \cdot t_{trasm} = 40\mu s\\
t_{isoc} + t_{async} &\simeq 660\mu s
\end{aligned}
$$

Ovviamente, bisogna tenere conto di eventuali ritardi nelle trasmissioni e del fatto che anche la lunghezza dei cavi RJ-45 utilizzati ha influenza sul tempo di propagazione dei frame. Altro fattore da prendere in considerazione è il tempo necessario a portare i nodi in LPI, che in generale occupa una parte rilevante del periodo di idle. Pertanto si tratta di dimensionare il ciclo POWERLINK in modo tale che sia possibile compiere tutte le sue fasi senza rischiare di incorrere nella ricezione ritardata o nella perdita di frame, che potrebbero impedire di rispettare le deadline imposte dalle trasmissioni. Una volta scelto questo periodo si potrà verificare se la durata effettiva della fase di idle sia sufficientemente grande da permettere ai nodi di portarsi in LPI e risvegliarsi prima dell'inizio di un nuovo ciclo.
Sulla base delle considerazioni fatte, si è deciso di adottare un timeout di attesa per le trasmissioni del polling pari a 60 μs, che deve essere più grande del tempo stimato per trasmettere e ricevere i due frame PReq e PRes. Il periodo di ciclo EPL è stato fissato a

1.2 *ms*, che risulta essere sufficientemente grande da permettere lo scadere di tutti questi timeout senza che il ciclo corrente sfori in quello successivo e che è in linea con gran parte dei periodi utilizzati nelle reti industriali a 100 Mbps reali [19].

Velocità	$T_{EPL}[\mu s]$	$Timeout[\mu s]$	$t_{isoc}[\mu s]$	$t_{async}[\mu s]$	$t_{idle}[\mu s]$
100 Mbps	1200	60	458.88	13.76	727.36
1000 Mbps	130	7.5	52.888	1.376	75.736
10 Gbps	12	0.7	3.3888	0.1376	8.4736

Tabella 5.1 – Periodi del ciclo EPL considerati e corrispondenti timeout di attesa durante il polling.

In questo modo si ottiene una fase di idle sufficientemente lunga da permettere ai nodi di portarsi in LPI, ma si ha anche un ciclo EPL sufficientemente breve da assicurare il corretto funzionamento della rete. Nel caso in cui le specifiche del problema siano più stringenti, sarà necessario selezionare un periodo di ciclo minore, portando però ad una riduzione della percentuale di permanenza in LPI. I parametri che devono essere impostati nei modelli dei nodi MN e CN da utilizzare durante la simulazione sono:

T_{EPL} periodo del ciclo EPL;

t_{trasm} tempo di trasmissione dei frame;

$t_{PReq,PRes_{CN}}$ tempo stimato tra la trasmissione del frame PReq alla ricezione del corrispondente PRes;

$t_{wait,PRes}$ timeout di attesa durante il polling

$$t_{wait,PRes} > t_{PReq,PRes_{CN}};$$

t_s tempo necessario a portare il link in LPI;

t_w tempo necessario a risvegliare il link;

t_q periodo di refresh;

T_{CN} tempo tra la trasmissione del *wake signal* e la trasmissione del frame SoC, necessario ai CN per ricevere il primo ed essere attivi prima di dover ricevere il secondo.

Di seguito verranno presentate le simulazioni compiute su delle reti EPL a 100 Mbps, in cui vengono implementati i meccanismi dell'EEE, mentre nelle appendici A e B si potranno trovare rispettivamente le simulazioni relative alle reti EPL a 1000 Mbps e a 10 Gbps. Per prima cosa si andranno a considerare i vari casi presentati nella sezione 4.1, in cui il MN gestisce in maniera differente il passaggio in LPI da parte dei CN sulla base del tipo di trasmissioni che devono compiere.
Per semplicità verrà supposto che i ritardi di trasmissione introdotti dai cavi siano uguali per tutti i nodi e che quelli introdotti dagli hub siano trascurabili.

5.4 Passaggio in LPI dopo la fase isocrona

Per prima cosa viene considerato il caso in cui tutti i nodi si portino in LPI dopo la fase isocrona, quindi in corrispondenza della ricezione del frame SoA. Si suppone anche che alcuni di essi compiano una richiesta di trasmissione durante la fase asincrona, perciò, ad ogni ripetizione del ciclo EPL, il CN designato dal MN dovrà attendere la trasmissione del frame ASnd prima di potersi portare nello stato di basso consumo energetico, mentre gli altri in coda potranno farlo dopo la ricezione del frame SoA.

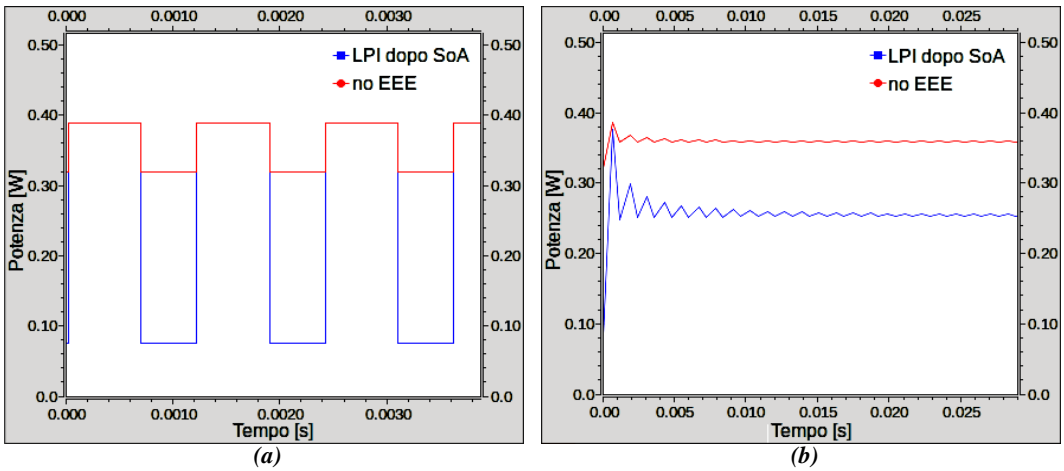


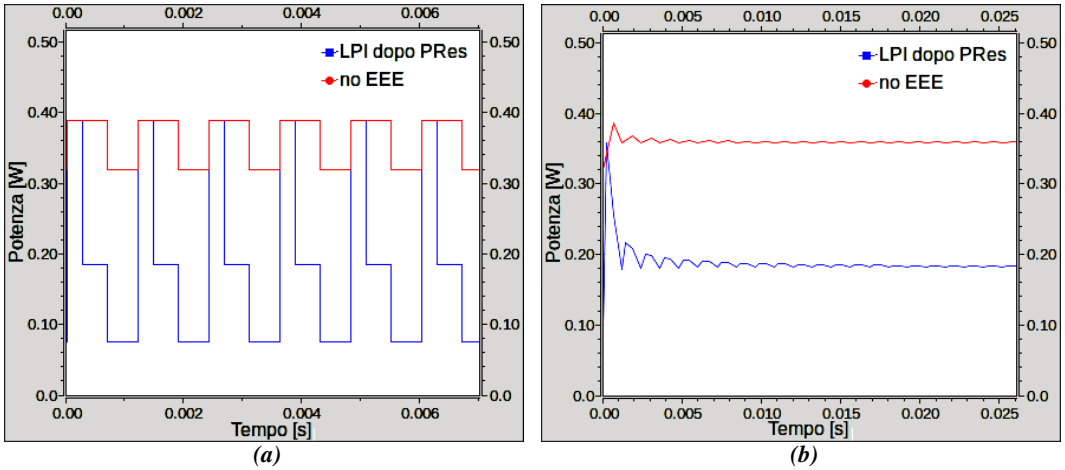
Figura 5.8 – Confronto tra i consumi assoluti (a) e medi (b) di un CN con passaggio in LPI dopo il frame SoA.

In figura 5.8 è possibile vedere il confronto tra il consumo di energia da parte di un singolo nodo CN di una rete EPL in cui non è implementato l'EEE e lo stesso nodo di una rete in cui questo standard è presente. Come si può notare, il passaggio nella modalità LPI

può avvenire in maniera completa non appena si è conclusa la fase asincrona, ottenendo una riduzione media dei consumi di un singolo nodo vicina al 30%.

5.5 Passaggio in LPI dopo il frame PRes

Per aumentare il tempo di permanenza nella modalità LPI da parte dei nodi della rete, risulta essere conveniente gestire singolarmente quei nodi che non devono partecipare alla fase asincrona corrente e che non hanno effettuato alcuna richiesta in tal senso nei cicli precedenti. In questo caso, infatti, è possibile anticipare il passaggio nello stato di basso consumo, attendendo esclusivamente l'iterazione del polling, durante la fase isocrona, che coinvolge questi nodi ed effettuando il cambio di modalità immediatamente dopo la trasmissione del corrispondente frame PRes.
In questo modo, soprattutto per quanto riguarda i primi nodi che partecipano solo alle trasmissioni deterministiche, si ha la possibilità di incrementare notevolmente la permanenza in LPI, ottenendo un beneficio sul risparmio medio dell'intera rete.



(a) *(b)*

Figura 5.9 – Confronto tra i consumi assoluti (a) e medi (b) di un CN con passaggio in LPI dopo il frame PRes.

In figura 5.9 è rappresentato il confronto tra il consumo di uno di questi CN che non implementa lo standard EEE con uno che invece adotta i meccanismi di risparmio energetico. Come si può facilmente verificare, subito dopo la trasmissione del proprio frame PRes, il nodo effettua un passaggio parziale nella modalità LPI, riducendo in maniera consistente il proprio consumo, per poi compiere il passaggio completo durante la fase di idle.

5.6 Consumo della rete complessiva

Fino ad ora si è valutato il risparmio energetico che si può ottenere adottando lo standard EEE su un singolo nodo CN. Per poter dare un'idea dell'effettiva efficacia di questi meccanismi, però, è necessario determinare il comportamento globale della rete; tenendo conto di tutti i nodi presenti e degli eventuali hub. Si deve quindi supporre che anche questi ultimi implementino lo standard EEE, altrimenti non sarebbe possibile portare in LPI i link ad essi collegati.

Come già accennato in precedenza, non è possibile portare completamente in LPI i link della rete, infatti è necessario mantenere sempre attivo il collegamente dal MN ai vari CN. Pertanto, ogni nodo che desideri disattivare il proprio trasmettitore portandosi parzialmente in LPI dovrà prima notificarlo inviando all' hub a cui sono collegati un segnale LPI. Ricevuto questo segnale, gli hub potranno disattivare il ricevitore corrispondente per poi portarsi completamente in LPI durante la fase di idle.

Una volta introdotto il consumo energetico da parte degli hub, è possibile riprendere in considerazione i due casi visti in precedenza, questa volta determinando la riduzione del consumo energetico relativo a tutta la rete.
In figura 5.10 è presentato il confronto tra tre reti topologicamente identitiche in cui, nella prima non viene implementato lo standard EEE, nella seconda viene implementato questo standard, imponendo il passaggio dei nodi in LPI dopo la fase asincrona, mentre nella terza il passaggio avviene durante la fase isocrona, quindi subito dopo la trasmissione del proprio frame PRes.

Come si può notare, il passaggio alla modalità ACTIVE avviene in tutti e tre i casi in corrispondenza dell'inizio del ciclo EPL; inoltre, l'ipotesi di trascurabilità dei ritardi di trasmissione porta ad avere un cambiamento di stato pressochè contemporaneo di tutti i nodi che, altrimenti, risulterebbe leggermente ritardato man mano che i nodi sono più distanti dal MN.
Il passaggio in LPI dopo la fase asincrona avviene per quasi tutti i nodi in corrispondenza della ricezione del frame SoA; infatti, solamente il nodo designato per questa fase deve attendere la trasmissione del frame ASnd prima di poter disattivare il proprio trasmettitore. Inoltre il passaggio avviene in maniera completa, poichè durante questa fase non è necessario mantenere un collegamento con il MN. Nell'ingrandimento in figura 5.10(c), si vede proprio il comportamento di questo nodo che, rispetto agli altri nodi della rete, deve ritardare leggermente il passaggio in LPI.
Nel caso in cui la disattivazione parziale del link possa avvenire durante la fase isocrona, si ha che i nodi che non hanno richiesto di partecipare alla fase asincrona del ciclo corrente,

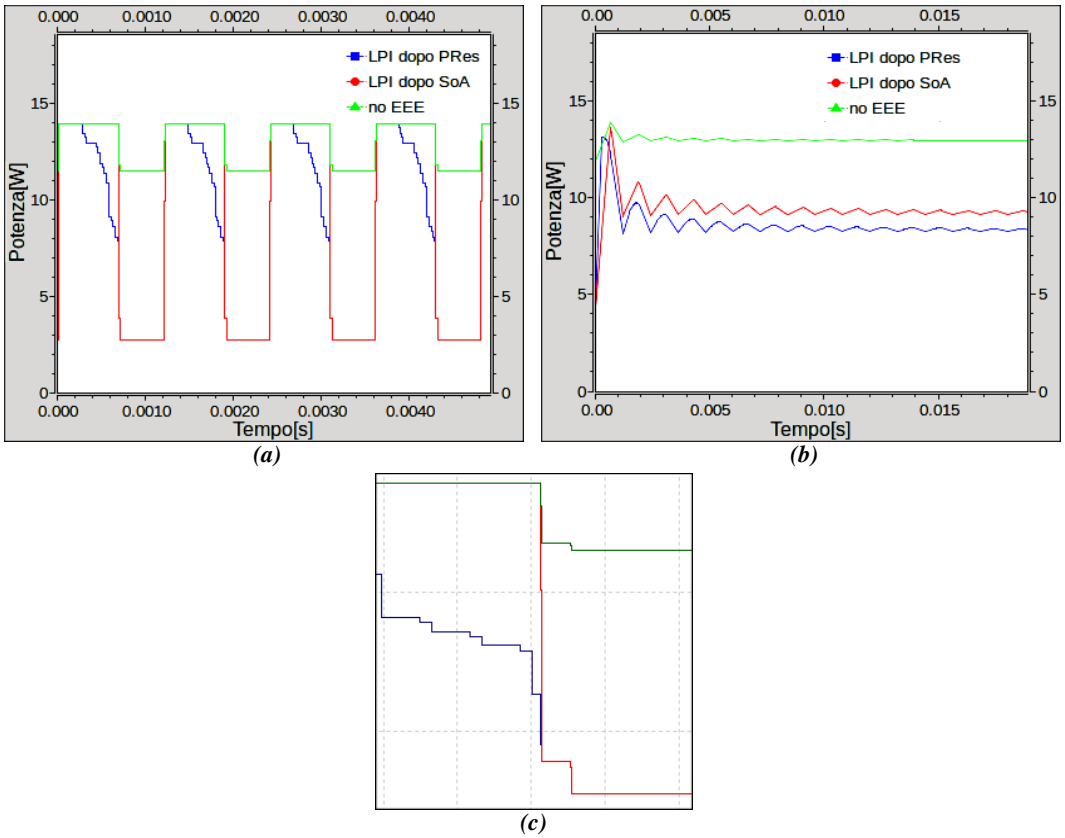


Figura 5.10 – Confronto tra i consumi assoluti (a) e medi (b) di reti EPL senza EEE, con passaggio in LPI dopo il frame PRes oppure dopo il frame SoA. Ingrandimento del passaggio in LPI del CN che effettua la trasmissione asincrona (c).

hanno la possibilità di anticipare il passaggio parziale in LPI per poi ridurre ulteriormente il consumo in corrispondenza della fase di idle con il passaggio completo a tale modalità. Osservando il confronto tra i consumi medi delle tre reti appena descritte, si vede che, introducendo il passaggio in LPI durante la fase di idle, si ottiene una riduzione dei consumi dell'intera rete vicina al 30%, mentre, permettendo il passaggio anche durante la fase isocrona, si raggiunge una riduzione superiore al 40%.

5.7 Rete con Multiplexed Timeslot

Mantenendo le ipotesi di adottare hub in grado di portare le proprie porte Ethernet in LPI, si procede supponendo che vi siano alcuni dei nodi che partecipano alle trasmissioni con una periodicità multipla del periodo T_{EPL}. Anche questi nodi possono essere gestiti in maniera differente in base al tipo di trasmissione che devono compiere, ad esempio, se devono partecipare solamente alla fase isocrona del ciclo a cui sono stati assegnati, è possibile compiere il passaggio in LPI subito dopo la trasmissione del proprio frame PRes, mantenendoli in questa modalità per il tempo restante del ciclo corrente e per tutta la durata dei cicli a cui non devono partecipare. Se, invece, devono anche compiere trasmissioni asincrone, sarà necessario attendere la fase di idle. La necessità di mantenere i nodi CN in ascolto durante le fasi isocrone e asincrone, vincola, anche in questo caso, a compiere esclusivamente il passaggio parziale in LPI dei link, mantenendo i ricevitori attivi durante tutti i periodi EPL, compresi quelli a cui non sono stati assegnati e completando il passaggio nello stato di basso consumo energetico solo durante le fasi di idle.

Per analizzare questa casistica si è deciso di mantenere il periodo di ciclo utilizzato fino ad ora ma di adottare una rete con topologia differente da quella vista fino ad ora, in modo da rendere più evidente la distinzione tra i nodi che partecipano ad ogni ciclo POWERLINK da quelli che, invece, prendono parte solo ad alcuni di essi. In particolare, la rete utilizzata è quella rappresentata in figura 5.11, in cui i nodi collegati al MN dall'*hub1* hanno una periodicità pari a T_{EPL}, mentre i nodi collegati tramite l'*hub2* e l'*hub3* compiono trasmissioni solamente ogni due periodi del ciclo. Si suppone che tutti i nodi che non devono effettuare trasmissioni durante la fase asincrona passino in LPI dopo la trasmissione del prorpio PRes, che alcuni dei nodi richiedono di partecipare a tale fase e attendono la ricezione del frame SoA e che solamente uno di essi trasmetta il frame ASnd prima di portarsi nello stato di basso consumo.

In figura 5.12 si possono vedere le simulazioni compiute sulla rete appena descritta, supponendo che la velocità di trasmissione sia pari a 100 Mbps. Per rendere evidente la

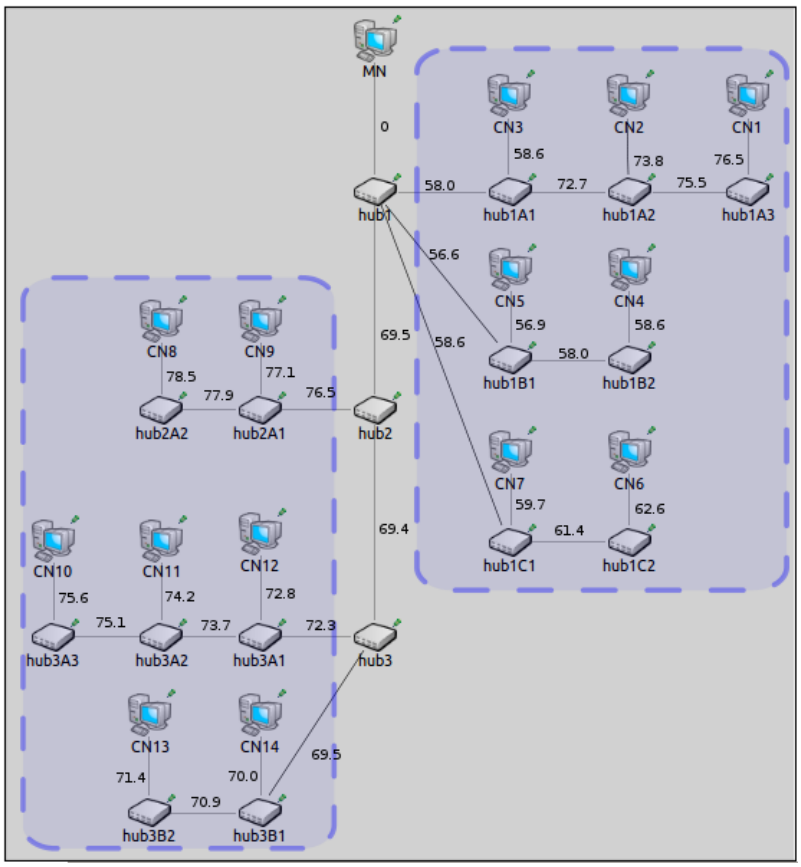


Figura 5.11 – Rete EPL con multiplexed timeslot.

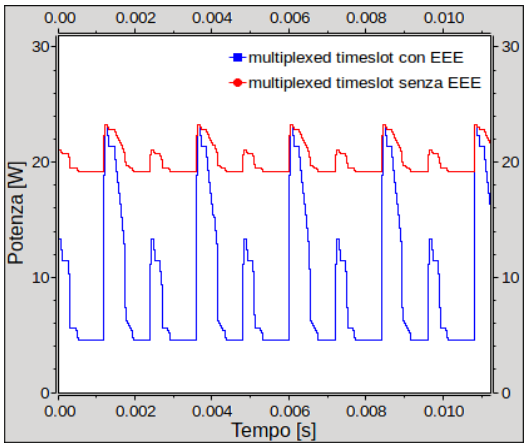
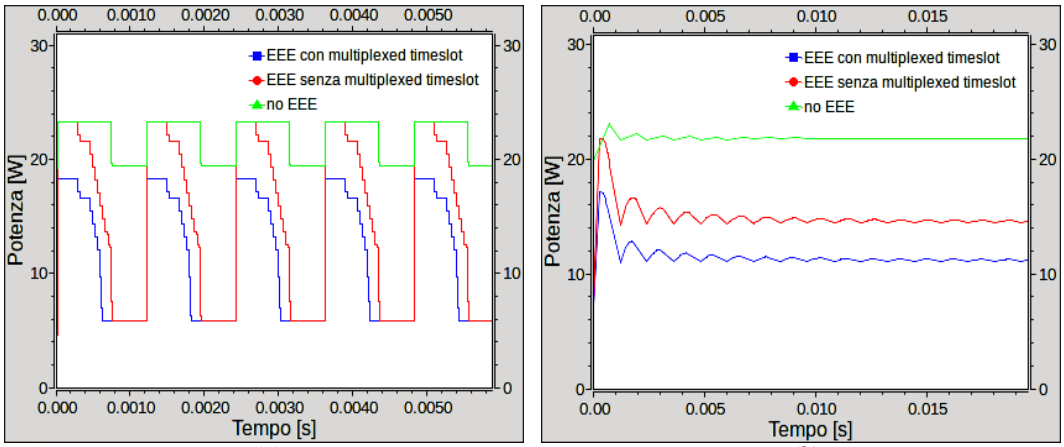


Figura 5.12 – Consumo di potenza di una rete EPL con multiplexed timeslot in un unico ciclo.

riduzione dei consumi negli istanti in cui solo una parte dei nodi partecipa attivamente ad ogni ciclo POWERLINK, si è inizialmente deciso di considerare il caso in cui i nodi con multiplexed timeslot compiano le trasmissioni durante uno stesso ciclo. Si può notare che, durante questo ciclo, si ha un picco del consumo energetico, che risulta essere all'incirca il doppio di quello corrispondente al ciclo in cui solo una parte dei nodi è coinvolta, proprio perchè, circa la metà dei nodi, adotta il multiplexed timeslot.
Nello standard EPL, è però indicato di implementare il MN in modo tale che sia in grado di bilanciare il carico di lavoro nella rete. Come si può notare dalle seconde simulazioni in figura 5.13, infatti, non vi è una alternanza di cicli in cui il consumo è notevolmente ridotto e altri in cui questo aumenta per la partecipazione di tutti i nodi. Ciò avviene perchè il MN distribuisce equamente i nodi, in modo tale che questi non si accumulino tutti in un unico ciclo e assicurando un carico circa costante della rete. Vengono messi a confronto



(a) *(b)*

Figura 5.13 – Consumo di potenza assoluto (a) e medio (b) di una rete EPL con multiplexed timeslot distribuito.

i casi in cui non venga implementato lo standard EEE con quello in cui questo standard è implementato ma i nodi participano a tutti i cicli EPL e quello in cui adottano il multiplexed timeslot. Si nota facilmente che il consumo medio della rete risulta essere ulteriormente ridotto se vi sono dei nodi che partecipano solo ad alcuni dei cicli, dal momento che questi possono portarsi in LPI per un periodo più prolungato.
Sempre in figura 5.11 sono anche state inserite le percentuali di permanenza in LPI dei link della rete, da cui si evince ancora una volta che i primi nodi che partecipano alla fase isocrona sono quelli che massimizzano tale valore, mentre quelli che devono attendere la fase asincrona hanno percentuali inferiori. Un aumento del tempo in LPI si nota anche in corrispondenza dei nodi che non partecipano a tutti i cicli POWERLINK, ancora una volta

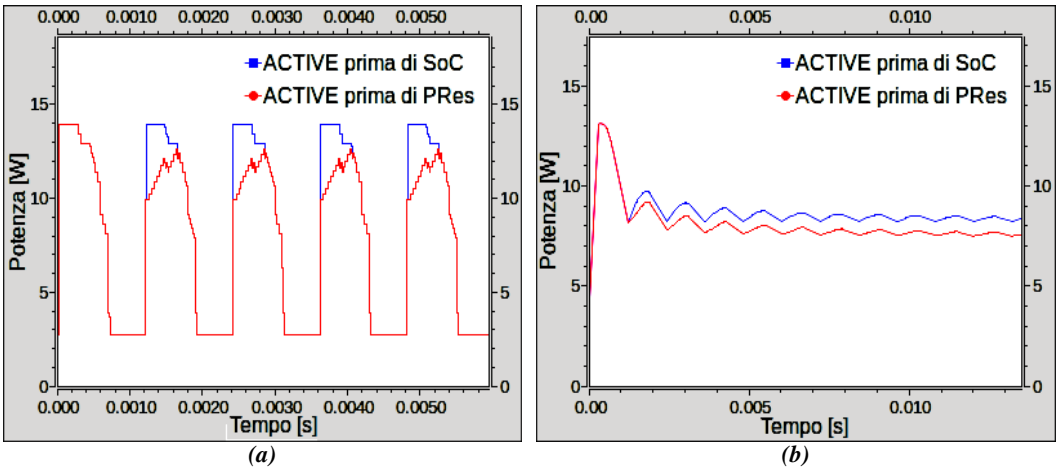
per la possibilità di mantenere il trasmettitore disattivato per un periodo più prolungato.

5.8 Attivazione del link prima della trasmissione dei frame

La grande rigorosità con cui viene gestito il ciclo all'interno di una rete EPL permette di incrementare ulteriormente la percentuale di tempo in cui i nodi si trovano in LPI stimando con buona precisione il momento in cui questi debbano effettuare delle trasmissioni e riattivandoli esclusivamente per l'intervallo temporale necessario a compierle. In tal caso, è essenziale tenere conto del tempo necessario a riportare i nodi nello stato ACTIVE, in modo da assicurare che questi siano pronti a trasmettere nell'istante esatto, evitando di causare ritardi che potrebbero pregiudicare il determinismo della rete.
Ad esempio, prendendo in considerazione un CN che deve partecipare solamente alla fase isocrona corrente, è possibile mantenere il link parzialmente in LPI, nella direzione del MN, per tutta la durata del ciclo EPL, attivandone il trasmettitore con un anticipo pari a T_w rispetto all'inizio della trasmissione del frame PRes e disattivandolo nuovamente una volta che questo è stato trasmesso. Un ragionamento analogo, come già anticipato nella sezione 4.1.5, può essere esteso anche ai CN che necessitano di compiere trasmissioni asincrone o che adottano il multiplexed timeslot, prolungando in entrambi i casi la permanenza dei nodi in LPI. Per verificare quanto detto, si è deciso di riprendere in esame il caso trattato nella sezione 5.6 rappresentato in figura 5.7 imponendo che tutti i nodi vengano riattivati solamente in corrispondenza della propria trasmissione isocrona o asincrona e supponendo che non venga adottato il multiplexed timeslot.

Come è possibile osservare in figura 5.14, il comportamento dei nodi è identico durante il primo ciclo EPL; infatti si suppone che ognuno di essi non conosca inizialmente il momento in cui verrà interrogato dal MN. Una volta avvenuta la prima iterazione, i CN memorizzano questo istante temporale ed aggiungono la durata del periodo per stimare il momento in cui dovranno essere attivi nel ciclo successivo. Tenuto conto del tempo necessario a riattivare il trasmettitore, i nodi possono posticipare notevolmente il momento in cui riattivarsi, per poi riportarsi in LPI esattamente nello stesso istante dell'altro caso considerato.

Adottando questa tecnica, si ottimizza l'utilizzo del link e il conseguente consumo energetico, dal momento che esso rimane attivo solo per il tempo strettamente necessario, portandosi nello stato di basso consumo per il resto del ciclo.



(a) *(b)*

Figura 5.14 – Consumo di potenza assoluto (a) e medio (b) di una rete EPL con passaggio in ACTIVE prima della trasmissione del frame PRes.

5.9 Effetto della topologia sul risparmio energetico

Un'ultima considerazione da fare per quanto riguarda il consumo energetico all'interno di una rete EPL, riguarda il tipo di topologia utilizzata. Per capire quanto possano essere influenti i vari livelli di hub si è deciso di mettere a confronto la rete con topologia ad albero descritta nella sezione 5.3, con una contenente lo stesso numero di nodi, ma con topologia a stella. In entrambi i casi si è deciso di compiere il passaggio in LPI subito dopo la trasmissione del frame PRes per i nodi che partecipano solamente alla fase isocrona, mentre i nodi che hanno effettuato una richiesta da trasmissione asincrona (i nodi CN3, CN4 e CN5), rimangono in attesa della ricezione del frame SoA ed, eventualmente, della trasmissione del frame ASnd.

Nelle figure 5.15 e 5.16 sono presentate le due reti, in cui sono state inserite le percentuali di tempo in LPI dei vari link. Risulta subito evidente che, i nodi che partecipano per primi alla fase isocrona, sono quelli che possono rimanere in tale modalità per un tempo più prolungato, ad esempio, il nodo CN1, cioè il primo a partecipare a questa fase, può disattivare il proprio link per circa il 70% dell'intero ciclo EPL. Mano a mano che ci si avvicina alla fase asincrona, questa percentuale cala, fino a raggiungere valori vicini al 40% per i nodi che hanno richiesto di trasmettere frame asincroni.
Per quanto riguarda la rete con topologia ad albero, inoltre, è possibile notare il comportamento dei link che collegano due hub. Questi, infatti, hanno una percentuale di permanenza in LPI minore o uguale alla più piccola associata ai link ad esso collegati. Ciò si verifica

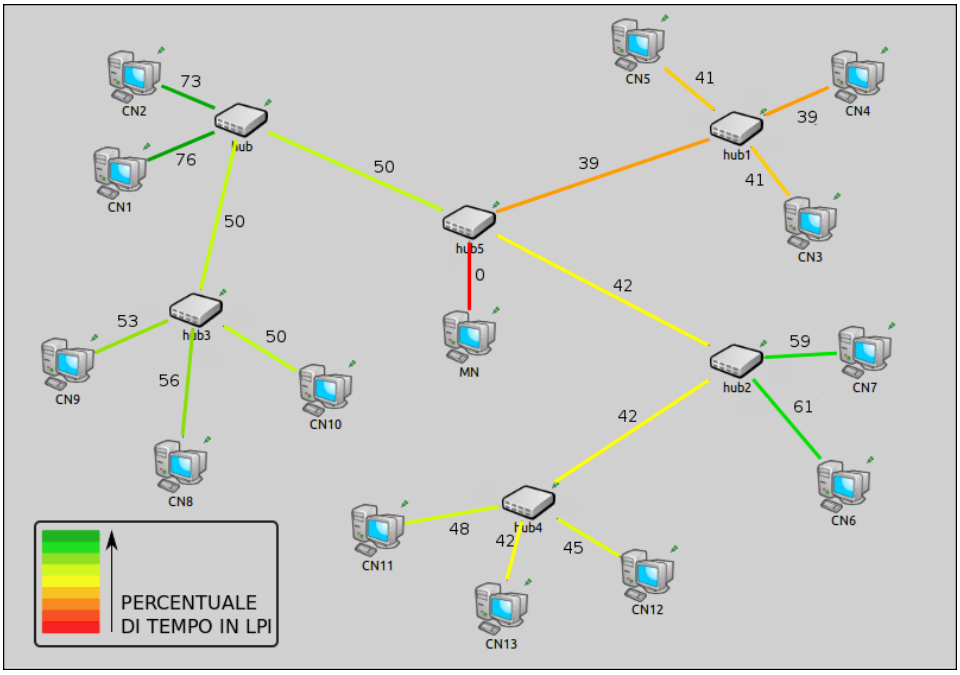


Figura 5.15 – Percentuali di permanenza in LPI nella reta con topologia ad albero.

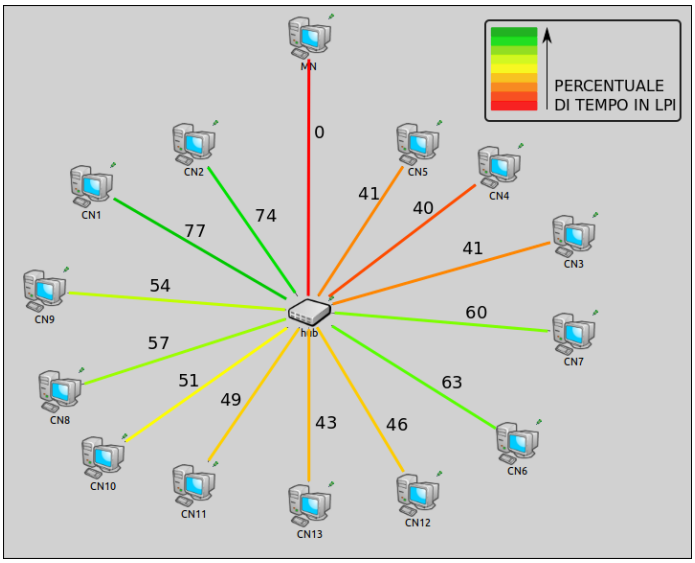


Figura 5.16 – Percentuali di permanenza in LPI nella rete con topologia a stella.

perchè, questi hub, prima di poter disattivare parzialmente questi link, devono attendere il completamento di tutte le trasmissioni compiute dai nodi a cui sono collegati, in modo tale da impedire situazioni di perdita dei frame.
Per implementare questo comportamento, si è semplicemente imposto che ogni hub tenga traccia dei nodi da cui ha ricevuto uno sleep signal, portando in LPI il link di collegamento con altri hub, solamente quando questo è l'unico a rimanere ancora attivo.
La misura del consumo medio delle due reti è presentato in figura 5.17, da cui si evince facilmente che la rete con topologia a stella consuma nettamente meno rispetto a quella con topologia ad albero.

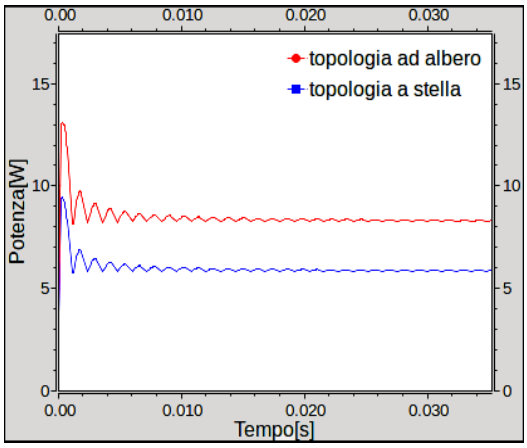


Figura 5.17 – Confronto tra i consumi medi di due rete EPL con topologie ad albero e a stella.

Ciò deriva dal fatto che, nel primo caso, il numero di hub, e quindi di link attivi, è decisamente inferiore, perciò anche il consumo risulta essere ridotto. Ovviamente, in molti casi pratici non è possibile adottare una configurazione di questo tipo, ma è sempre conveniente dal punto di vista energetico cercare di ridurre al massimo il numero di livelli di hub, creando dei gruppi di CN collegati ad uno stesso hub. Questi dispositivi, inoltre, introducono un leggero ritardo nelle trasmissioni e riducendone il numero, si ha che la durata delle fasi isocrone ed asincrone risulta minimizzata, aumentando quindi la possibilità di portare completamente i nodi in LPI.

Conclusioni

In questa tesi sono state analizzate le potenzialità dello standard EEE nelle reti Ethernet industriali, studiando in particolare il caso delle reti EPL. Si sono presi in considerazione i vari casi che si possono presentare in queste reti, proponendo delle soluzioni in grado di massimizzare la riduzione dei consumi energetici senza pregiudicarne l'aspetto deterministico.
Sulla base della percentuale di tempo in cui i nodi possono portarsi in uno stato di basso consumo durante il ciclo di funzionamento, è stato possibile determinare l'effetto dell'adozione dei meccanismi di efficienza energetica nei casi presi in esame. I risultati ottenuti sono riassunti in tabella 5.2, dove è possibile notare che, nel caso più generale in cui i nodi vengono portati in LPI solo nella fase di inattività dell'intera rete, si può ottenere una riduzione dei consumi vicina al 30% rispetto ad una rete priva di EEE, nel caso in cui i nodi possano effettuare il passaggio dopo la trasmissione del frame PRes si ottiene una riduzione attorno al 40%, se si introduce la possibilità di risvegliare i link subito prima della trasmissione di tale frame si ottiene una ulteriore riduzione del 5% e se vi sono dei nodi che partecipano solamente ad alcuni dei cicli EPL è possibile ridurre il consumo di oltre il 50%. Dal momento che questi valori dipendono dalla durata della fase di idle e, quindi, dal periodo del ciclo, non è possibile mostrare un risultato generico per tutte le reti, ma vengono fornite delle simulazioni su modelli di reti industriali in linea con quelle effettivamente esistenti nella realtà.

Uno sviluppo futuro potrebbe essere quello di implementare questi meccanismi in un caso pratico, in modo da verificare se i risultati ottenuti abbiano effettivamente un riscontro con una rete vera e propria. Ad esempio, sarebbe possibile creare una rete di PC dotati di schede Ethernet con EEE, che sono già in commercio, per poi imporre ai vari nodi di adottare un comportamento il più possibile simile a quello dei nodi EPL e misurare la reale

Caso considerato	Percentuale di energia risparmiata		
	100 Mbps	1000 Mbps	10 Gbps
LPI dopo SoA	28.9	35.3	36.6
LPI dopo PRes	36.1	-	41.7
ACTIVE prima di PRes	42.3	-	46.8
Multiplexed timeslot	48.8	-	53.1

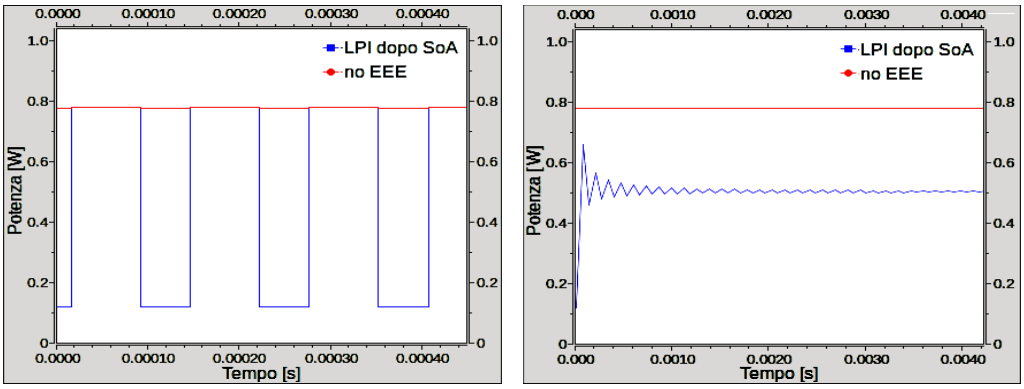
Tabella 5.2 – Percentuali di energia che può essere risparmiata adottando lo standard EEE nei vari casi considerati.

riduzione dei consumi.
Potrebbe essere molto interessante anche estendere le considerazioni fatte ad altre reti industriali con lo scopo di creare degli algoritmi compatibili con standard differenti. Ad esempio si potrebbe prendere in considerazione il caso delle reti PROFINET, in cui esistono delle modalità operative di tipo real-time. Alcune di queste sono organizzate in maniera ciclica ed è possibile compiere dei ragionamenti analoghi a quelli fatti nelle reti EPL, in altri invece il funzionamento non è organizzato in cicli sincronizzati, ma è comunque possibile adottare soluzioni già presenti nelle comuni reti Ethernet, come mantenere i nodi in LPI fino alla ricezione di un frame. In questo modo si va ad introdurre un leggero ritardo corrispondente al tempo necessario alla riattivazione del link che può spesso essere trascurato se le deadline associate alle trasmissioni non sono particolarmente stringenti.

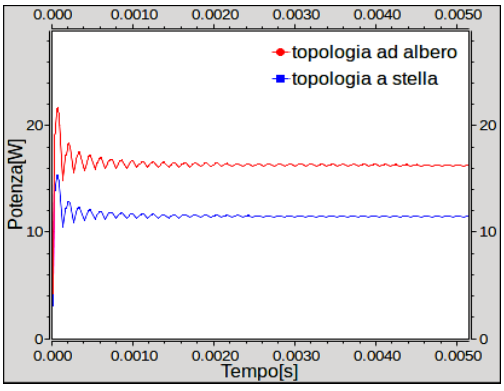
Appendice A

Simulazioni rete 1000 Mbps

Nel caso delle reti a 1000 Mbps non è possibile il passaggio parziale in LPI, quindi si considerano i casi di passaggio dopo il frame SoA nelle reti di figura 5.15 e 5.16.



(a) *(b)*



(c)

Figura A.1 – Confronto tra i consumi assoluti (a) e medi (b) di un CN con passaggio in LPI dopo il frame SoA. Confronto tra i consumi medi di due reti EPL con topologie ad albero e a stella (c).

Appendice B

Simulazioni rete 10 Gbps

Per le reti a 10 Gbps è possibile il passaggio parziale in LPI, quindi verranno considerati tutti i casi già visti per le reti a 100 Mbps.

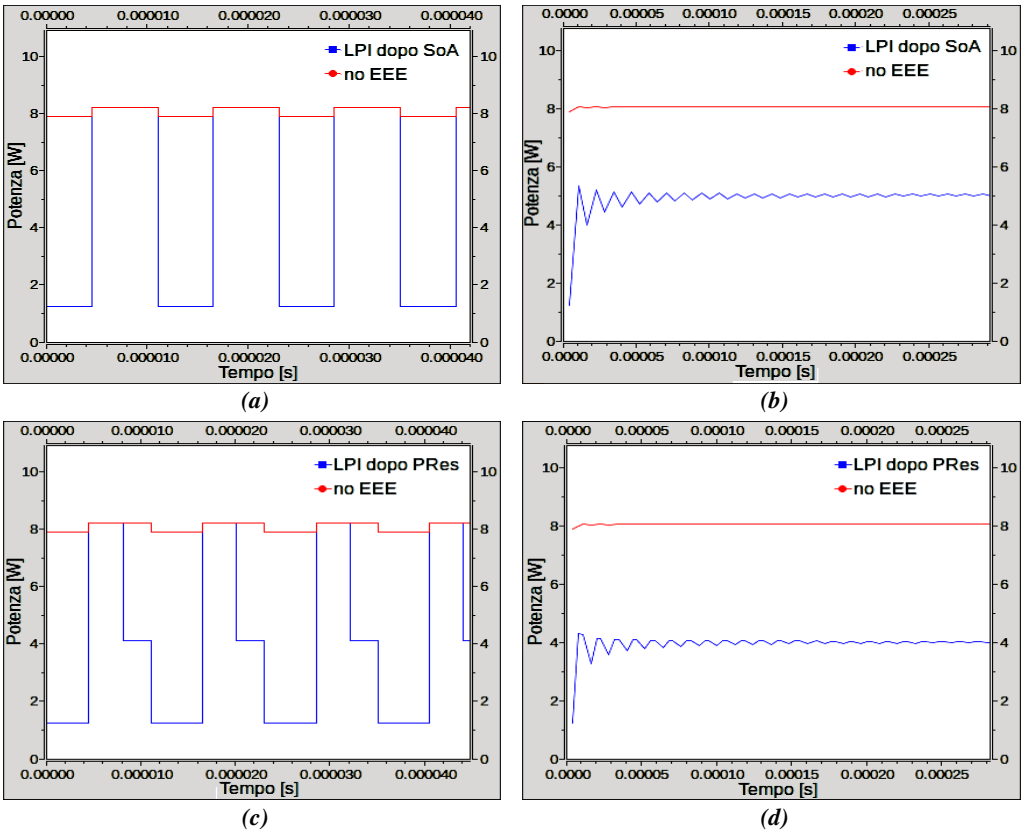


Figura B.1 – Confronto tra i consumi assoluti e medi di un CN con passaggio in LPI dopo il frame SoA (a) - (b) e dopo il frame PRes (c) - (d) nelle reti di figura 5.7 .

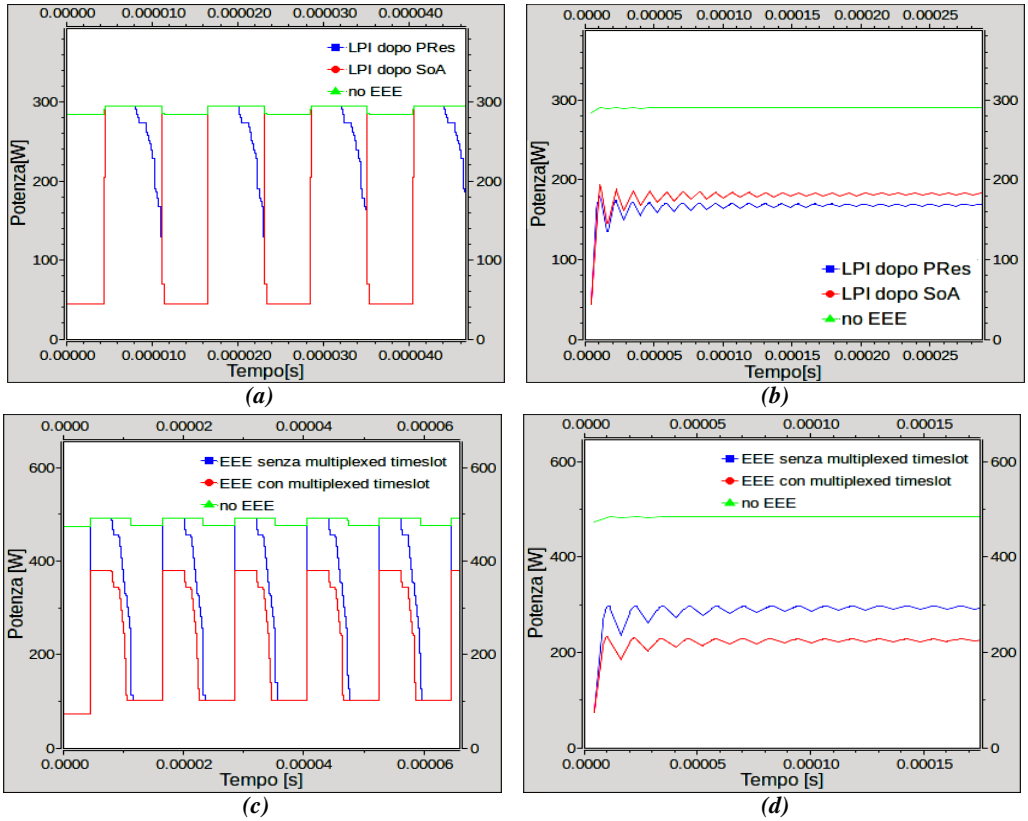


Figura B.2 – Confronto tra i consumi assoluti e medi di reti EPL senza EEE, con passaggio in LPI dopo il frame PRes oppure dopo il frame SoA (a) - (b) con topologia rappresentata in figura 5.7. Consumo di potenza assoluto e medio della rete EPL di figura 5.11 con multiplexed timeslot distribuito (c) - (d).

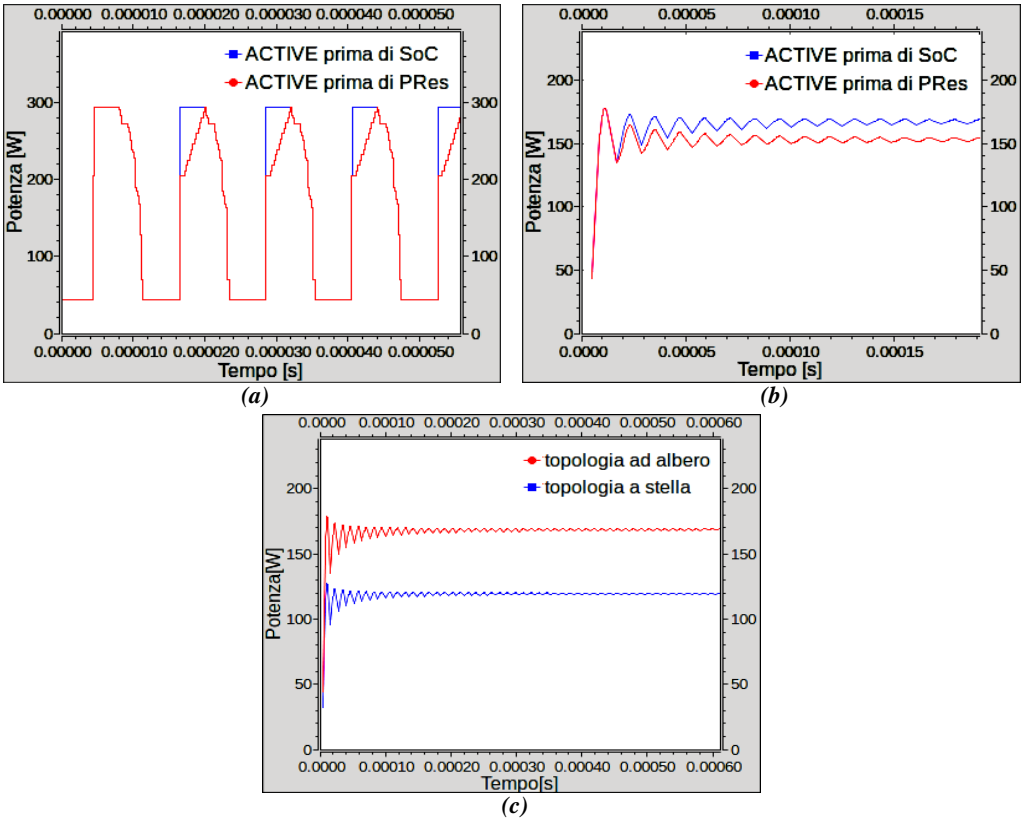


Figura B.3 – Consumo di potenza assoluto e medio di una rete EPL con passaggio in ACTIVE prima della trasmissione del frame PRes (a) - (b) nella rete di figura 5.11. Confronto tra i consumi medi di due rete EPL con topologie ad albero e a stella (c) con topologie rappresentate in figura 5.15 e 5.16.

Ringraziamenti

Ringrazio la mia famiglia, che mi ha sempre supportato e mi ha dato la possibilità di concludere con grande soddisfazione il mio percorso universitario. Ringrazio poi il Professor Stefano Vitturi e gli Ingegneri Federico Tramarin e Lucia Seno, che hanno sempre dimostrato una grande disponibilità nei miei confronti seguendomi costantemente durante tutto lo svolgimento della tesi. Un ringraziamento particolare va anche ai miei compagni di università che hanno reso questi cinque anni indimenticabili. Infine, desidero ringraziare la mia fidanzata Nicoletta, senza la quale non sarei arrivato fin qui.

Bibliografia

[1] *Broadcom BCM5421xE Family datasheet*. [Online]. Available: http://www.broadcom.com/products/features/energy_efficient_network.php

[2] *INET Framework for OMNeT++*. [Online]. Available: http://inet.omnetpp.org/doc/INET/inet-manual-draft.pdf

[3] *Intel 82573L Gigabit Ethernet Controller Datasheet*. [Online]. Available: http://www.intel.com/products/ethernet/resource.htm#s1=all&s2=all&s3=all

[4] *Intel 82579 Gigabit Ethernet Controller Datasheet*. [Online]. Available: http://www.intel.com/products/ethernet/resource.htm#s1=all&s2=all&s3=all

[5] *Profibus International, PROFINET Application Layer Service Definition , Version 1.95*.

[6] *Survey form EIA-861 – Annual Electric Power Industry Report*. [Online]. Available: http://www.eia.gov/electricity/data/eia861/index.html

[7] *UNI CEI EN ISO 50001:2011 Sistemi di gestione dell'energia - Requisiti e linee guida per l'uso*.

[8] *User Guide for Endace DAG 3.6E*.

[9] *Ethernet/ip specification*, 2001. [Online]. Available: http://www.odva.org

[10] "Green computing and d-link: The green movement and specific d-link solution reduce detrimental impacts on the envirorment," July 2009. [Online]. Available: ftp://ftp10.dlink.com/pdfs/products/D-Link_Green_whitepaper.pdf

[11] B. Brunch, "An introduction to auto-negotiation," National Semiconductor, Tech. Rep., 1995.

[12] *EN 50325-2: Industrial Communication Subsystem based on ISO 11898 (CAN) for controller-device interface - part 2: DeviceNet*, cenelec Std.

[13] J. Chou, A. Kuo, D. Tseng, J. Lin, and K. Huang, "roposal of low-power idle: 100base-tx," 2008, iEEE 802.3 Meeting.

[14] K. Christensen, P. Reviriego, B. Nordman, M. Bennett, M. Mostowfi, and J. Maestro, "Ieee 802.3az: the road to energy efficient ethernet," *Communications Magazine, IEEE*, vol. 48, no. 11, pp. 50–56, november 2010.

[15] K. J. Christensen, C. Gunaratne, B. Nordman, and A. D. George, "The next frontier for communications networks: power management," *Computer Communications*, vol. 27, no. 18, pp. 1758 – 1770, 2004, ¡ce:title¿Performance and Control of Next Generation Communications Networks¡/ce:title¿. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0140366404002427

[16] G. Clemente, F. Filira, and M. Moro, *Sistemi operativi. Architettura e programmazione concorrente*. Libreria Progetto, 2003. [Online]. Available: http://books.google.it/books?id=twOMAAAACAAJ

[17] J. Eggers and S. Hodnett, "Ethernet autonegotiation best practices," in *Sun BluePrints OnLine*, 2004.

[18] *EtherCAT: Ethernet for Control Automation Technology.*, EtherCAT Technology Group Std., 2003. [Online]. Available: http://www.ethercat.org

[19] *EPSG Draft Standard 301. Ethernet POWERLINK Communication Profile Specification Version 1.1.0*, Ethernet POWERLINK Standardisation Group Std., 2008.

[20] *EN50170: Worldfip Protocol*, European Fieldbus Standard Std. [Online]. Available: http://cern-worldfip.web.cern.ch/cern-worldfip/pdffiles/WDPROTOC.pdf

[21] M. Felser, "Real-time ethernet - industry prospective," *Proceedings of the IEEE*, vol. 93, no. 6, pp. 1118 –1129, june 2005.

[22] *DIN 19245 : Profibus Standard*, German National Standard Std.

[23] C. Gunaratne, K. Christensen, B. Nordman, and S. Suen, "Reducing the energy consumption of ethernet with adaptive link rate (alr)," *Computers, IEEE Transactions on*, vol. 57, no. 4, pp. 448 –461, april 2008.

[24] C. Gunaratne and K. Christensen, "Ethernet adaptive link rate (alr): Analysis of a buffer threshold policy," in *Proceedings of IEEE GLOBECOM*, 2006.

[25] ——, "Ethernet adaptive link rate: System design and performance evaluation," in *Proceedings of the IEEE Conference on Local Computer Networks*, 2006, pp. 28–35.

[26] M. Gupta and S. Singh, "Greening of the internet," in *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, ser. SIGCOMM '03. New York, NY, USA: ACM, 2003, pp. 19–26. [Online]. Available: http://doi.acm.org/10.1145/863955.863959

[27] *IEEE 802.3 - 2008 - IEEE Standard for Information technology-Specific requirements - Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications*, IEEE Std. [Online]. Available: http://standards.ieee.org/about/get/802/802.3.html

[28] *IEEE 802.3az-2010 : Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications*, IEEE Std.

[29] C. in Automation (CiA) International Users and M. Group, *CANopen: Application Layer and Communication Profile ; CiA Draft Standard 301 ; Version 4.0*. CiA, 1999. [Online]. Available: http://books.google.it/books?id=qkm_tgAACAAJ

[30] *IEC 61158: Digital data communications for measurement and control - fieldbus for use in industrial control systems - part 2 and 6.*, International Electrotechnical Commission Std.

[31] *IEC 61784: Digital data communications for measurement and control. Part 1: Profile sets for continuous and discrete manufactoring relative for iso/iec8802-3 based communication networks in real-time applications.*, International Electrotechnical Commission Std.

[32] *IEC 61158 : P-Net Standard*, International Fieldbus Standard Std. [Online]. Available: http://www.p-net.dek

[33] I. IXXAT, "Introduction to ethernet powerlink (epl 2.0)," Bedford Center Rd., Tech. Rep., 2005.

[34] F. Lian, J. R. Moyne, and D. M. Tilbury, "Performance evaluation of control networks for manufacturing systems," in *In Proceedings of the ASME International Mechanical Engineering Congress and Exposition (Dynamic Systems and Control Division)*, 1999, pp. 6–7.

[35] R. Lu and C. Min, "Energy efficient ethernet technology based on ieee 802.3az," iTU-T Wuhan Research Institute of Posts and Telecommunications.

[36] J. Maestro and P. Reviriego, "Energy efficiency in industrial ethernet: The case of powerlink," *Industrial Electronics, IEEE Transactions on*, vol. 57, no. 8, pp. 2896 –2903, aug. 2010.

[37] J. Moyne and D. Tilbury, "The emergence of industrial control networks for manufacturing control, diagnostics, and safety data," *Proceedings of the IEEE*, vol. 95, pp. 29 –47, 2007.

[38] A. Odlyzko, "Data networks are lightly utilized, and will stay that way," *Review of Network Economics*, vol. 2, no. 3, pp. 210–237, September 2003. [Online]. Available: http://ideas.repec.org/a/rne/rneart/v2y2003i3p210-237.html

[39] PAE, "Piano d'azione italiano per l'efficienza energetica," 2011.

[40] N. Patric and W. David, "Power savings in multiple technology physical layer devices supporting autonegotiation," Unites States Patent 5 907 553, 1999.

[41] K. Roebuck, *Energy Efficient Ethernet (802.3az): High-Impact Technology - What You Need to Know: Definitions, Adoptions, Impact, Benefits, Maturity, Vendors*. Emereo Pty Limited, 2011. [Online]. Available: http://books.google.it/books?id=XhWNZwEACAAJ

[42] K. Roth, F. Goldstein, and J. Kleinman, "Energy consumption by office and telecommunications equipment in commercial buildings– volume i: Energy consumption baseline," U.S. Department of Energy, Tech. Rep., 2002.

[43] A. Varga, *OMNeT++: User Manual (Version 4.2.2)*. [Online]. Available: http://www.omnetpp.org/doc/omnetpp/Manual.pdf

[44] S. Vitturi, "Dispensa del corso: Laboratorio di automazione industriale."

[45] B. Zhang, K. Sabhanatarajan, A. Gordon-Ross, and A. George, "Real-time performance analysis of adaptive link rate," in *Local Computer Networks, 2008. LCN 2008. 33rd IEEE Conference on*, oct. 2008, pp. 282 –288.