

Reti Ethernet a efficienza energetica per applicazioni industriali.

Parte B: le misure in laboratorio

Perazzolo Mirko

26 settembre 2013

Indice

1	Introduzione	6
2	Misure Software	8
2.1	Dispositivi e connessioni	8
2.1.1	ping	9
2.1.2	wireshark	9
2.1.3	ethtool	9
2.2	Misure da terminale	10
2.2.1	ping da PC2 a PC1	10
2.2.2	ping da PC1 a PC2	13
2.3	Misure con generatore esterno	15
2.4	Conclusioni delle misure via software	15
3	Misure sulla linea con l'oscilloscopio	15
3.1	Dispositivi e connessioni	16
3.2	Confronti dei segnali sul cavo	17
3.2.1	Energy Efficiency OFF	17
3.2.2	Energy Efficiency ON	17
3.3	Le esperienze eseguite in laboratorio	18
3.3.1	Misure sul tempo di quiet T_q	18
3.3.2	Misure sul tempo di refresh T_r	18
3.3.3	Misure con generatore di pacchetti esterno 1	20
3.3.4	Misure di corrente sulla scheda	21
3.3.5	Misure con generatore di pacchetti esterno 2	23
3.4	Conclusioni delle misure via hardware	25
4	Misure a 1Gbit/s	27
4.1	Misure sul tempo di quiet T_q	27
4.2	Misure sul tempo di refresh T_r	27
4.3	Misure con generatore di pacchetti esterno	29
5	Conclusioni	31

Pagina lasciata intenzionalmente bianca

Elenco delle figure

1	Modello a stati di EEE	6
2	Transizioni nel ciclo di EEE	7
3	Switch	8
4	Wireshark	9
5	ping	10
6	Tempi di ping da TERMINALE	11
7	Tempi di ping da WIRESHARK	11
8	Tempi di ping da TERMINALE	12
9	Tempi di ping da WIRESHARK	12
10	Tempi di ping da TERMINALE	13
11	Tempi di ping da WIRESHARK	13
12	Tempi di ping da TERMINALE	14
13	Tempi di ping da WIRESHARK	14
14	Struttura di un pacchetto ethernet	15
15	Circuito per misure su cavo ethernet	16
16	Prolunga PCI 1x per misure amperometriche sulla scheda	16
17	Rete utilizzata per le misure hardware	17
18	Risultato dell'oscilloscopio con l'EEE off	18
19	Risultato dell'oscilloscopio con l'EEE on	18
20	Risultato oscilloscopio misura T_q	19
21	Risultato oscilloscopio misura T_r	19
22	Risultato oscilloscopio sequenza pacchetti	20
23	Sonda amperometrica "Agilent 1147A 50 MHz"	21
24	Collegamento della scheda al computer per mezzo della prolunga modificata	21
25	Misura pacchetto con segnale di trigger	23
26	Grafico riassuntivo tempi al variare della dimensione dei pacchetti	25
27	Risultato oscilloscopio sequenza pacchetti a 1Gibt/s	27
28	Risultato oscilloscopio misura T_q	28
29	Risultato oscilloscopio misura T_r	28
30	Misura pacchetto con segnale di trigger	29
31	Grafico riassuntivo tempi al variare della dimensione dei pacchetti	31

Elenco delle tabelle

1	Tempistiche dell'EEE	7
2	Indirizzi IP	8
3	Misure ping da PC2 a PC1	11
4	Misure ping da PC2 a PC1	12
5	Misure ping da PC2 a PC1	13
6	Misure ping da PC2 a PC1	14
7	Caratteristiche elettriche scheda ethernet Intel I210	22
8	Misure "Inizio Trigger"- "Inizio Pacchetto"	24
9	Misure "Inizio Pacchetto"- "Fine Trigger"	24
10	Misure "Fine Trigger"- "Fine Pacchetto"	24
11	Misure "Inizio Trigger"- "Fine Trigger"	24
12	Misure "Inizio Pacchetto"- "Fine Pacchetto"	24
13	Tempi di trasmissione ethernet	25
14	Misure "Inizio Trigger"- "Inizio Impulso"	29
15	Misure "Inizio Impulso"- "Fine Trigger"	29

16	Misure “Fine Trigger”-“Fine Pacchetto”	30
17	Misure “Inizio Impulso”-“Inizio Pacchetto”	30
18	Misure “Inizio Trigger”-“Fine Trigger”	30
19	Misure “Inizio Pacchetto”-“Fine Pacchetto”	30
20	Tempi di trasmissione ethernet	30

Pagina lasciata intenzionalmente bianca

Sommario

Nel 2010 è stato pubblicato da parte dell'*Institute of Electrical and Electronics Engineers* (IEEE) l'emendamento 802.3az, volto ad aumentare l'efficienza energetica delle reti Ethernet. Tale standard, chiamato *Energy Efficient Ethernet* (EEE) introduce una nuova modalità operativa, denominata *Low Power Idle* (LPI), che riduce notevolmente il consumo energetico della scheda Ethernet.

Nella parte A di questa tesi viene esposto lo standard Ethernet nel suo aspetto teorico, con particolare attenzione per le reti *Real Time Ethernet* (RTE), reti di comunicazione caratterizzate da tempistiche molto ristrette, di interesse per l'automazione industriale. Successivamente vengono spiegate alcune possibili misurazioni effettuabili su una scheda che implementa l'EEE per verificare che rispetti lo standard.

Nella parte B della tesi sono illustrate le suddette misure, eseguite sulla scheda Ethernet *Intel I210-T1*, focalizzando l'attenzione sulla descrizione dell'apparato di misura, i risultati ottenuti e le osservazioni relative ad essi.

Il lavoro di misura, è stato svolto presso il laboratorio di *Compatibilità Elettromagnetica e Misure Elettroniche* del dipartimento di Ingegneria dell'Informazione DEI.

1 Introduzione

Lo standard IEEE 802.3az definisce le caratteristiche di *Energy Efficient Ethernet*. L'emendamento introduce una nuova modalità operativa denominata *Low Power Idle* (LPI). Quando tale modalità è adottata qualsiasi partner di un link Ethernet può entrare in uno stato denominato "*Quiet*" e caratterizzato da basso consumo energetico.

Il funzionamento di una scheda che implementa l'EEE si può modellizzare con una macchina a stati (Fig.1) in cui si possono distinguere due stati principali *Active* e *Quiet* che sono rispettivamente lo stato di attività, durante il quale c'è trasmissione di dati, e lo stato di quiete.

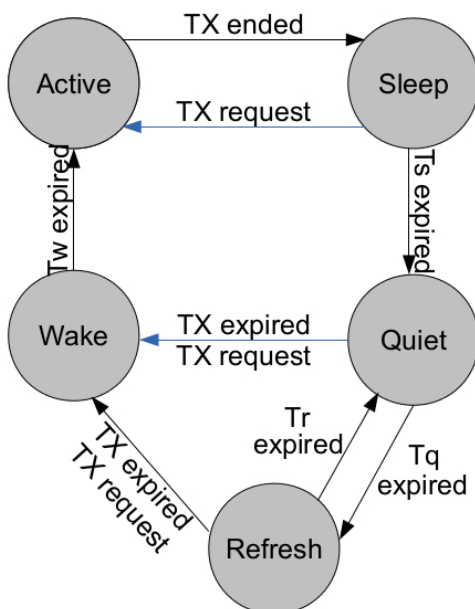


Figura 1: Modello a stati di EEE

Oltre a questi vi sono altri tre stati:

- *Sleep*: fase di transizione durante la quale avvengono le procedure per portare il link nello stato quiet;
- *Wake*: fase di transizione durante la quale avvengono le procedure per riattivare il link;
- *Refresh*: fase necessaria per mantenere integro il link mediante l'invio di un segnale puramente elettrico.

Ovviamente la presenza di questi stati comporta dei tempi non nulli per passare dall'uno all'altro. Lo standard a riguardo ha imposto dei vincoli che ogni scheda deve rispettare per implementare correttamente l'EEE.

I tempi principali su cui pone l'attenzione lo standard sono:

- T_s : tempo necessario per passare dallo stato *Active* allo stato *Quiet*;
- T_q : tempo di permanenza nello stato di *Quiet* trascorso il quale, senza attività, avviene una procedura di Refresh per mantenere attivo il link;
- T_r : tempo necessario per eseguire una procedura di Refresh;
- T_w : tempo necessario per passare dallo stato *Quiet* allo stato *Active*.

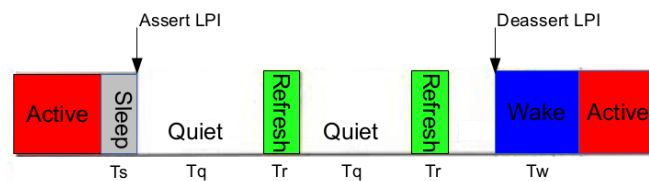


Figura 2: Transizioni nel ciclo di EEE

L'**obbiettivo** di questa parte di tesi è la rilevazione dei tempi dettati dallo standard su una scheda campione e verificare che questi siano entro i limiti imposti dallo standard (Tabella 1).

PHY	T_s^{min}	T_s^{max}	T_q^{min}	T_q^{max}	T_r^{min}	T_r^{max}	T_w^{max}
100Mbit/s	200	220	22000	24000	200	220	20.5
1Gbit/s	182	202	20000	24000	198	218.2	16.5
10Gbit/s	2.88	3.2	39.68	39.68	1.28	1.28	4.48

Tabella 1: Tempistiche dell'EEE

Le prove sono state così strutturate:

1. Misure via software:

- Pacchetti generati con comando `ping` da terminale
- Pacchetti ricevuti da generatore esterno

2. Misure sul cavo con l'oscilloscopio:

- Confronti del segnale in caso di *EEE on* e *EEE off*
- Pacchetti ricevuti da generatore esterno

2 Misure Software

Le misure software sono state effettuate con programmi avviati nel PC che monta la scheda ethernet di interesse oppure su un secondo PC comunque collegato alla stessa rete del primo.

2.1 Dispositivi e connessioni

La scheda ethernet presa in esame è una "Intel® Ethernet Controller I210-T1", ed è stata montata su un computer messo a disposizione dal laboratorio di *Compatibilità Elettromagnetica e Misure Elettroniche* presso il dipartimento di Ingegneria dell'Informazione DEI.

Altri dispositivi sono:

- lo switch "D-Link® DGS-1100-16 EasySmart" che implementa l'EEE;
- un pc portatile;
- un microcontrollore "Arduino Ethernet" basato sull'ATmega328, utilizzato come generatore di pacchetti esterno.

Per le misure via software è stata montata una piccola rete ethernet, isolata dalla rete del laboratorio, collegando via cavo i computer e il microcontrollore allo switch. Inoltre sono stati attribuiti degli IP statici ad ognuno dei quattro dispositivi (vedere la Tabella 2).

	IP
Switch	10.90.90.90
PC1	10.90.90.91
PC2	10.90.90.92
Arduino Ethernet	10.90.90.93

Tabella 2: Indirizzi IP



Figura 3: Switch

I computer montano il sistema operativo Ubuntu 13.04, il quale mette a disposizione i seguenti strumenti:

- **ping** : comando per il terminale che genera dei pacchetti e li invia ad un indirizzo IP indicato misurando il tempo che intercorre tra l'invio del pacchetto e l'arrivo dell'ACK/NACK;
- **wireshark** : utility di sistema che permette di controllare tutto il traffico dati che transita sulla scheda ethernet.

- **ethtool** : comando per il terminale che permette la gestione di alcune funzionalità della scheda ethernet, quali abilitazione EEE oppure limitazione del bit-rate.

2.1.1 ping

Il comando **ping**: è disponibile sui principali sistemi operativi e misura il tempo, espresso in millisecondi, impiegato da uno o più pacchetti ICMP (*Internet Control Message Protocol*) a raggiungere un altro dispositivo di rete (attraverso una qualsiasi rete informatica basata su IP) e a ritornare indietro all'origine. Per le misure che verranno presentate è stato usato questo comando perché fornisce la possibilità di generare ed inviare pacchetti ad intervalli regolari ed inoltre dà una misura approssimativa dei tempi.

Un esempio di comando è:

```
ping -i 0.02 -c 100 10.90.90.92
```

questo comando genera ed invia all'indirizzo IP 10.90.90.92 fino a 100 pacchetti con un intervallo di 20ms tra un pacchetto e l'altro e poi stampa il tempo di ogni pacchetto.

2.1.2 wireshark

Wireshark[®] è un analizzatore di protocollo di rete. Permette all'utente di osservare tutto il traffico presente sulla rete utilizzando la modalità promiscua dell'adattatore di rete.

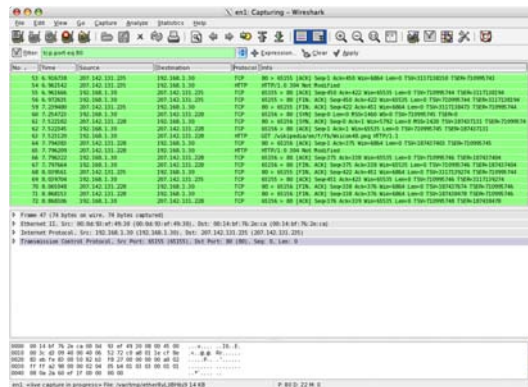


Figura 4: Wireshark

Con questo programma si ha una misurazione più accurata del tempo di ping ed inoltre si hanno molte altre informazioni, in quanto rileva con un tempo molto accurato tutti i pacchetti dati entranti ed uscenti dalla scheda Ethernet verso la rete.

2.1.3 ethtool

ethtool è un comando da terminale per Linux per gestire i parametri della scheda ethernet. Questa utility mette a disposizione svariate possibilità, ma di particolare interesse per noi sono i parametri con i quali possiamo controllare:

- abilitazione/disabilitazione dell'opzione Energy Efficiency
- regolazione del bit-rate, nel nostro caso 100Mbit/s o 1Gbit/s
- stato della scheda

Nel dettaglio, i comandi sono:

- `ethtool --show-eee eth1` restituisce tutte le informazioni sullo stato della scheda riguardanti l'EEE, ovvero se è abilitata o meno la possibilità di andare in LPI, verificare lo stato di LPI e il bit-rate massimo al quale è autorizzata a comunicare la scheda.
- `ethtool --set-eee eth1 eee on` abilita l'EEE, va osservato che tale comando dà alla scheda la possibilità di passare allo stato di LPI, ma non implica che la scheda sia in LPI.
- `ethtool --set-eee eth1 eee off` disabilita l'EEE, al contrario del precedente questo impedisce alla scheda di passare allo stato di LPI. Con questo comando si porta la scheda a lavorare come le schede di più vecchia generazione, che non implementano l'Energy Efficiency.

2.2 Misure da terminale

Lo standard esprime chiaramente che l'utilizzo dell'Energy Efficiency comporta l'inserzione di ulteriori tempi, in particolare i tempi di transizione dallo stato di *Quiet* a quello di *Active*. Viene quindi naturale pensare che un pacchetto impiegherà meno tempo ad essere inviato da una scheda che non implementa l'EEE piuttosto che da una scheda che lo implementa. La prima prova è finalizzata a verificare tale affermazione. Essa consiste nell'effettuare un ping da un computer all'indirizzo dell'altro e leggerne i tempi. Per avere maggior significato vengono effettuate varie prove sia con l'EEE della scheda Intel abilitato che disabilitato.

La sensibilità dei tempi riportati dal comando ping è dell'ordine di $1\mu s$, sufficiente per avere una buona approssimazione dei tempi che si vogliono rilevare. Per un'ulteriore verifica e per scongiurare l'aggiungersi di tempi di elaborazione dovuti al clock del processore del computer utilizziamo anche il programma `wireshark`, anch'esso con una sensibilità dell'ordine di $1\mu s$.

Le prove vengono così eseguite:

1. Si verifica lo stato EEE della scheda ed eventualmente si imposta l'EEE off col relativo comando `ethtool`
2. Si avvia `wireshark` su uno dei due computer, a seconda della misura che si sta effettuando
3. Si esegue il comando ping con la seguente sintassi

```
ping -i 0.001 -c 1000  $\alpha$ 
```

con α indirizzo IP del computer di destinazione, così da avere 1000 pacchetti di ping, distanti $1ms$ l'uno dall'altro.
4. Si rilevano ed elaborano i tempi dei pacchetti mostrati da `wireshark` e dal terminale
5. Si imposta l'EEE on della scheda col relativo comando `ethtool` e si ripetono le misure

Di seguito vengono riportate le varie misure che sono state effettuate.

2.2.1 ping da PC2 a PC1



Figura 5: ping

Durante queste prove è stato inviato il comando di ping dal portatile con indirizzo 10.90.90.92 al computer con la scheda in esame di indirizzo 10.90.90.91.

Nella Tabella 3 sono riportati i risultati delle misure:

ping da PC2 a PC1		
EEE PC1	OFF	ON
TERMINALE		
min	0.151 ms	0.202 ms
media	0.211 ms	0.251 ms
max	0.540 ms	0.799 ms
mdev	0.070 ms	0.104 ms
WIRESHARK avviato su PC2		
min	0.113 ms	0.157 ms
media	0.153 ms	0.190 ms
max	0.268 ms	0.442 ms
mdev	0.021 ms	0.023 ms

Tabella 3: Misure ping da PC2 a PC1

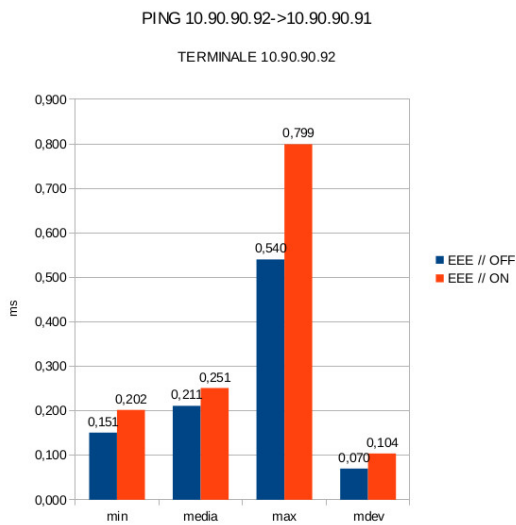


Figura 6: Tempi di ping da TERMINALE

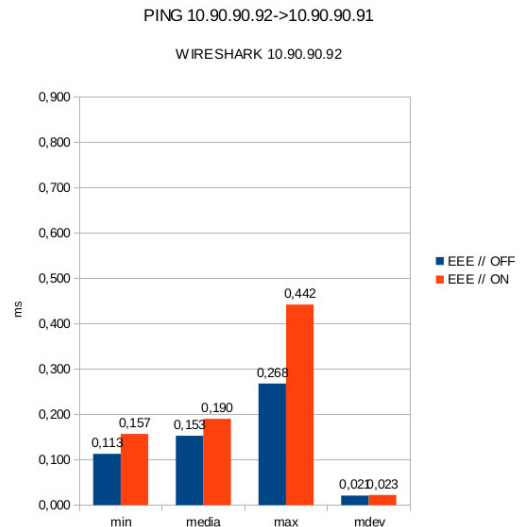


Figura 7: Tempi di ping da WIRESHARK

Da notare che i tempi letti dal terminale fanno sempre riferimento al transito dei dati sulla scheda del computer dal quale è stato inviato il comando di ping. Quindi i tempi misurati con wireshark e quelli misurati col terminale saranno simili solo quando wireshark è in esecuzione sullo stesso computer che invia il comando di ping.

La colonna di maggior interesse nelle Figure 6 e 7 è quella della **media**. Già da questi primi risultati possiamo osservare uno scarto di circa $40\mu s$ tra le misure con EEE off e le misure con EEE on.

Si può rilevare inoltre che tra le misure col terminale e quelle con wireshark c'è un discostamento in ritardo del primo rispetto al secondo di circa $60\mu s$ sia con l'EEE off, che con l'EEE on. È ammissibile supporre che questi siano i tempi di elaborazione del processore menzionati in precedenza, in quanto il terminale deve leggere il pacchetto e poi elaborare l'informazione, mentre wireshark indica immediatamente il tempo di transizione del pacchetto sulla scheda.

ping da PC2 a PC1		
EEE PC1	OFF	ON
TERMINALE		
min	0.191 ms	0.234 ms
media	0.259 ms	0.336 ms
max	0.700 ms	0.858 ms
mdev	0.099 ms	0.137 ms
WIRESHARK avviato su PC1		
min	0.024 ms	0.022 ms
media	0.032 ms	0.032 ms
max	0.036 ms	0.036 ms
mdev	0.001 ms	0.001 ms

Tabella 4: Misure ping da PC2 a PC1

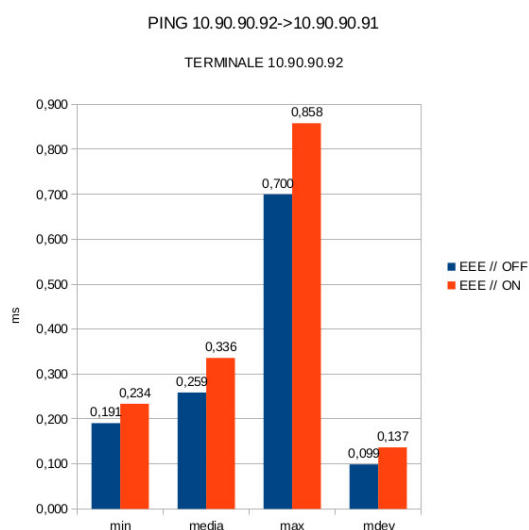


Figura 8: Tempi di ping da TERMINALE

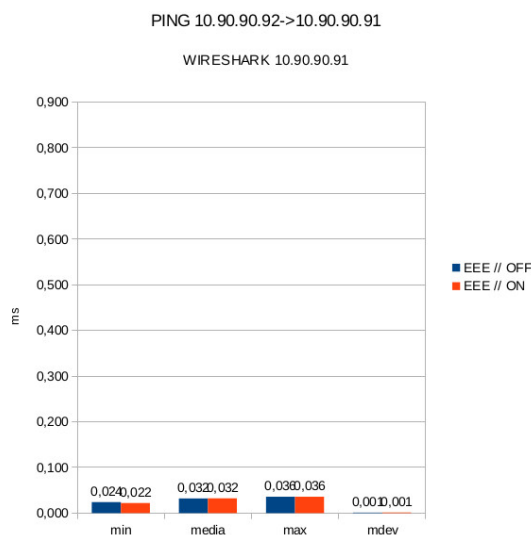
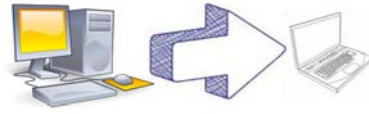


Figura 9: Tempi di ping da WIRESHARK

Nelle misure in Tabella 4, i risultati letti al terminale rispecchiano ancora quanto visto per le precedenti, ovvero con l'EEE off la trasmissione risulta essere più veloce rispetto all'EEE on, ma non si può dire lo stesso delle misure rilevate con wireshark. Si nota subito che i tempi di EEE off ed EEE on sono quasi coincidenti ed in generale sono decisamente inferiori rispetto alla rilevazione a terminale. Questo si può motivare molto semplicemente osservando che il programma wireshark è in esecuzione sul computer che riceve il ping che in questo caso è quello che monta la scheda che si sta esaminando. Quindi i tempi misurati sono esclusivamente i tempi di elaborazione del processore che deve leggere il pacchetto e generare l'ACK/NACK da ritornare. Inoltre non si osservano differenze a seconda dello stato dell'EEE in quanto wireshark rileva il pacchetto quando è già arrivato alla scheda e quando questa ha già finito la fase di *wake*. Dati i tempi di elaborazione del processore, circa $30\mu s$, la scheda non riesce a tornare allo stato di *sleep* prima di dover essere pronta ad inviare l'ACK/NACK, quindi si può affermare che nel tempo misurato da wireshark la scheda si comporta allo stesso modo, che l'EEE sia abilitata o meno.

2.2.2 ping da PC1 a PC2



Nella Tabella 5 vengono proposti i risultati delle misure effettuate come le precedenti salvo per il fatto che il comando di ping che è stato inviato dal computer con la scheda in esame di indirizzo 10.90.90.91 al portatile con indirizzo 10.90.90.92.

ping da PC1 a PC2		
EEE PC1	OFF	ON
TERMINALE		
min	0.169 ms	0.202 ms
media	0.215 ms	0.255 ms
max	0.370 ms	0.635 ms
mdev	0.035 ms	0.073 ms
WIRESHARK avviato su PC2		
min	0.039 ms	0.042 ms
media	0.058 ms	0.056 ms
max	0.071 ms	0.066 ms
mdev	0.008 ms	0.007 ms

Tabella 5: Misure ping da PC2 a PC1

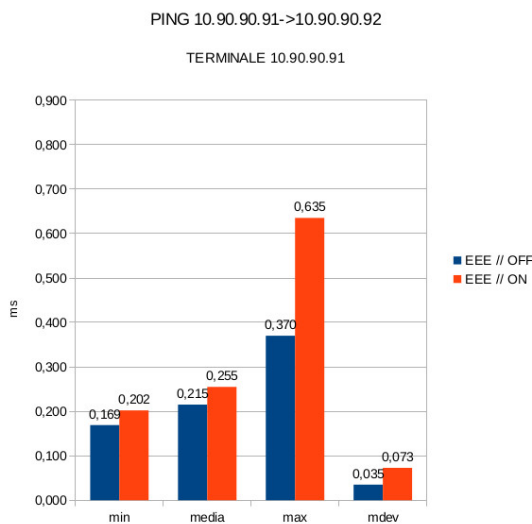


Figura 10: Tempi di ping da TERMINALE

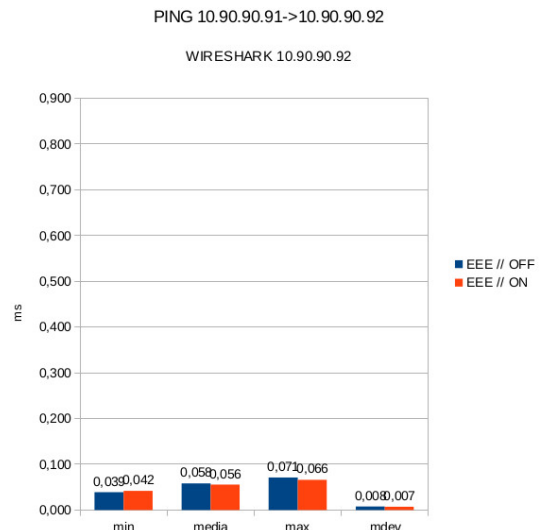


Figura 11: Tempi di ping da WIRESHARK

Come ci si aspettava, in queste misure abbiamo un ripetersi delle situazioni viste in precedenza. Nei risultati da terminale si nota che con l'EEE off la scheda è più rapida rispetto l'EEE on di qualche decina di μs . Nei risultati da wireshark ritroviamo il caso in cui i tempi misurati sono decisamente inferiori rispetto ai tempi del terminale, e come prima si osserva che questo è dovuto al fatto che il programma è eseguito sul computer che riceve il ping e non deve far altro che ritornare

un ACK/NACK. In questo caso non c'è il dubbio se interviene Energy Efficiency in quanto wireshark è in esecuzione su un computer dove non è stato abilitato l'EEE per non inserire ulteriori ritardi.

ping da PC1 a PC2		
EEE PC1	OFF	ON
TERMINALE		
min	0.186 ms	0.213 ms
media	0.234 ms	0.258 ms
max	0.617 ms	0.638 ms
mdev	0.075 ms	0.073 ms
WIRESHARK avviato su PC1		
min	0.170 ms	0.201 ms
media	0.205 ms	0.229 ms
max	0.394 ms	0.372 ms
mdev	0.024 ms	0.017 ms

Tabella 6: Misure ping da PC2 a PC1

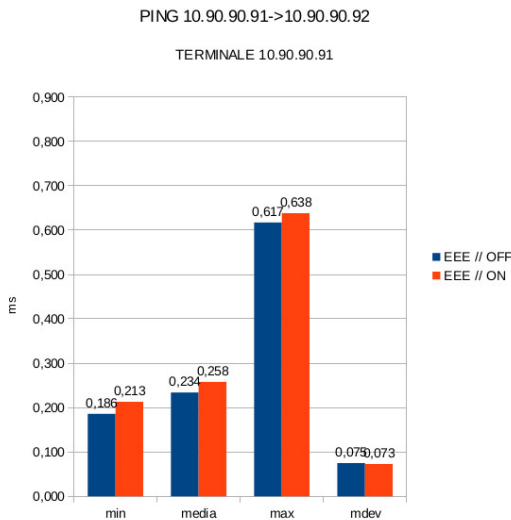


Figura 12: Tempi di ping da TERMINALE

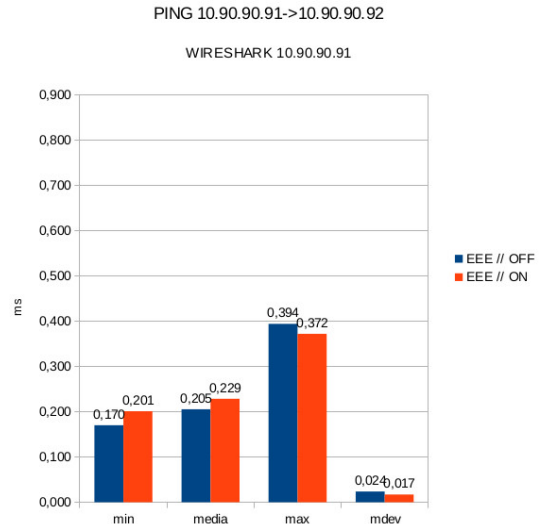


Figura 13: Tempi di ping da WIRESHARK

Per finire, anche in queste misure si può osservare come la scheda risulti più veloce nella comunicazione quando l'EEE è disabilitato rispetto a quando è abilitato, sempre con uno scarto di qualche decina di μs .

Si può notare che i valori medi delle misure "da PC1 a PC2" sono più elevati rispetto quelli delle misure "da PC2 a PC1". Questo perché EEE è su PC1 (che invia) e quindi bisogna attivare la scheda. Da PC2 a PC1 EEE è sempre su PC1 (che riceve), in questo caso, però, lo Switch comincia ad attivare il link prima di trasmettere verso PC1.

Si può anche notare che l'incremento da NO EEE a EEE è minore del caso "da PC2 a PC1".

Per concludere possiamo affermare che l'Energy Efficiency introduce dei ritardi nella trasmissione dati. Purtroppo per quanto riguarda l'entità di questi ritardi possiamo solo dire che sono dell'ordine di decine di μs , altro non sappiamo in quanto non si riesce a quantizzare l'influenza del sistema operativo nelle misure, oltre alla sensibilità e affidabilità non sufficienti degli strumenti utilizzati.

2.3 Misure con generatore esterno

Fatta questa prima verifica, ora bisogna riuscire a determinare con una maggiore certezza questi tempi.

Diventa importante conoscere con la maggior precisione possibile il contenuto dei pacchetti che vengono trasmessi.

Per fare ciò torna molto utile il microcontrollore Arduino Ethernet. Esso infatti è munito di un modulo ethernet attraverso il quale può comunicare con la rete in cui è inserito. Questo microcontrollore è in grado di generare dei pacchetti ethernet in cui inserisce autonomamente il preambolo, l'header e il trailer, di cui però conosciamo esattamente le dimensioni in bit, ma molto più utile è la possibilità di decidere cosa inserire nel payload senza temere che il sistema inserisca altri dati.

Arduino dà anche la possibilità di inviare gruppi di pacchetti con un intervallo di tempo a nostra

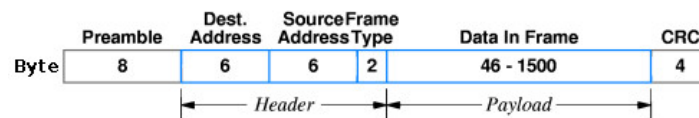


Figura 14: Struttura di un pacchetto ethernet

scelta tra un pacchetto e l'altro, inoltre non richiede obbligatoriamente di ricevere un ACK/NACK, così quando andremo a vedere il segnale, ci sarà un minor traffico sulla rete.

2.4 Conclusioni delle misure via software

L'approccio software può essere valido per avere un primo riscontro delle funzionalità dell'Energy Efficiency, come per avere un'idea di cosa succede, ma non può spingersi oltre. Un esempio rilevante è che con gli strumenti visti fin ora non siamo riusciti a rilevare la presenza del segnale di *Refresh* che sappiamo essere un segnale puramente elettrico per mantenere in essere il canale. Infatti il software è in grado di elaborare solamente informazioni che il sistema operativo sa interpretare, mentre un segnale elettrico, non in formato digitale riconoscibile ma che va ad agire solo in forma "meccanica" su una componente del computer, va oltre le capacità di gestione del sistema operativo.

Prima di concludere è utile osservare di come varino le misure effettuate con wireshark quando a turno viene eseguito il programma sul computer che è ricevente del ping, e quindi questo misura solo il tempo di risposta del processore. Nel caso in cui il ricevente sia il computer di indirizzo 10.90.90.91 abbiamo un tempo medio di $32\mu s$, mentre nel caso in cui il ricevente sia il computer di indirizzo 10.90.90.92 abbiamo un tempo medio di $56\mu s$. Questo ci avverte dell'influenza non trascurabile del sistema operativo sulle misure di tempo, la quale dipende da vari fattori, come la velocità del processore o i processi attivi che devono essere gestiti.

3 Misure sulla linea con l'oscilloscopio

Abbiamo visto come l'approccio software non dia informazioni su tutto ciò che avviene effettivamente sul canale e di come il sistema operativo influenzi in modo non trascurabile le misure di tempo.

Per poter avere delle rilevazioni di tempo affidabili, bisogna quindi appoggiarsi ad uno strumento di misura con una sensibilità ed una affidabilità maggiori. Inoltre ci interessa uno strumento che riesca a fornirci informazioni riguardo cosa sta avvenendo istante per istante sul cavo ethernet. Queste due caratteristiche ci portano a optare per l'utilizzo di un oscilloscopio, il quale è in grado di fornirci entrambe le informazioni di nostro interesse combinandole tra loro.

Le misure presentate di seguito fanno riferimento alla scheda con bit-rate di $100Mbit/s$. Successivamente si daranno i risultati delle misure effettuate sulla scheda con bit-rate di $1Gbit/s$ le quali sono eseguite nello stesso modo delle prime.

3.1 Dispositivi e connessioni

Oltre ai dispositivi utilizzati per le misure via software, sono stati aggiunti i seguenti strumenti:

- L'oscilloscopio "Agilent InfiniiVision DSO-X 3024A" con Bandwidth di $500MHz$ e Sample Rate di $4GSa/s$
- Un piccolo circuito autocostruito al fine di rendere più pratiche le misure sul cavo ethernet, visibile alla Figura 15

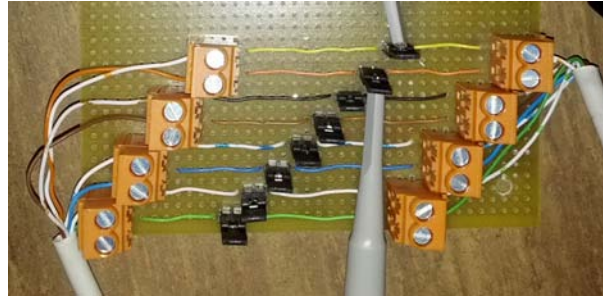


Figura 15: Circuito per misure su cavo ethernet

- Una prolunga PCI 1x per rendere più agevole l'accesso alla scheda ethernet. È stata modificata al fine di poter effettuare alcune misure amperometriche direttamente sull'alimentazione della scheda stessa (Figura 16)

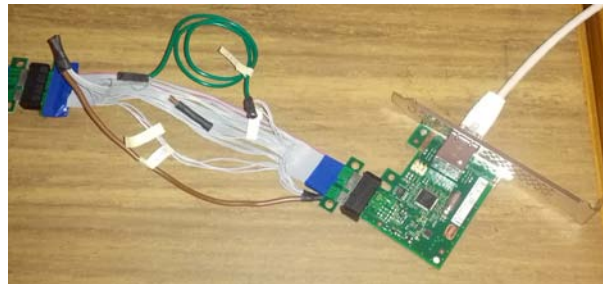


Figura 16: Prolunga PCI 1x per misure amperometriche sulla scheda

- Sonda amperometrica "Agilent 1147A 50 MHz" con Bandwidth della sonda di $50MHz$ e corrente massima di $15A$

Con l'inserimento di questi nuovi dispositivi la rete viene così modificata:

1. La scheda ethernet in esame viene collegata al computer per mezzo della prolunga che è stata modificata
2. Il cavo ethernet che collega la scheda allo switch è stato tagliato e collegato al circuito per le misure
3. Vengono collegati i puntali delle sonde dell'oscilloscopio ai jumper appositamente posti nel circuito
4. Viene collegata una sonda dell'oscilloscopio ad Arduino per avere un segnale di Trigger (quando serve)

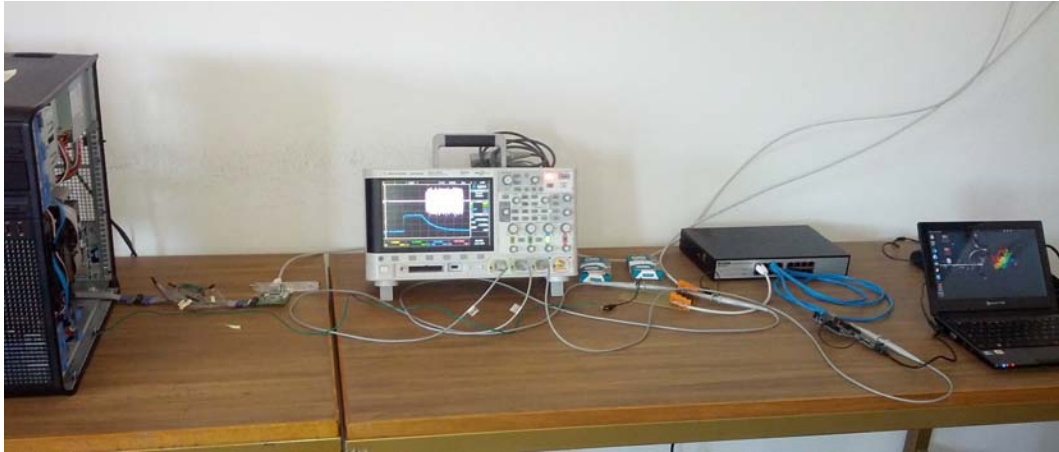


Figura 17: Rete utilizzata per le misure hardware

3.2 Confronti dei segnali sul cavo

Prima di vedere le misure effettuate, è utile osservare com'è il segnale sul cavo ethernet e fissare bene le differenze tra il segnale con l'Energy Efficiency attivo ed il segnale con l'Energy Efficiency disattivo. Le misure vengono così effettuate:

1. I cavi ethernet sono composti da quattro coppie di fili, prendiamo in esame una coppia
2. Colleghiamo il puntale di una sonda dell'oscilloscopio ad uno dei due fili e colleghiamo il riferimento alla carcassa del computer
3. Colleghiamo il puntale di una seconda sonda all'altro filo della coppia e colleghiamo il riferimento alla carcassa del computer
4. Effettuiamo una misura differenziale sui due segnali letti dalle due sonde

3.2.1 Energy Efficiency OFF

Vediamo ora il segnale che viene restituito dall'oscilloscopio nel caso non ci siano pacchetti dell'utente in transito.

Dalla Figura 18, si vede subito come, nonostante non ci sia un traffico di dati, sul cavo ci siano continuamente segnali in transito. Questo implica che c'è un dispendio di energia superiore del necessario.

3.2.2 Energy Efficiency ON

Ora ripetiamo la misura sempre nelle stesse condizioni (assenza di traffico dati), però abilitando l'EEE.

Nella Figura 19 la differenza con la precedente è notevole. Si può cogliere al volo come la scheda riduca considerevolmente l'ampiezza del segnale sul cavo ethernet. Livelli inferiori del segnale implicano un minor consumo di potenza, come previsto dal protocollo 802.3az.

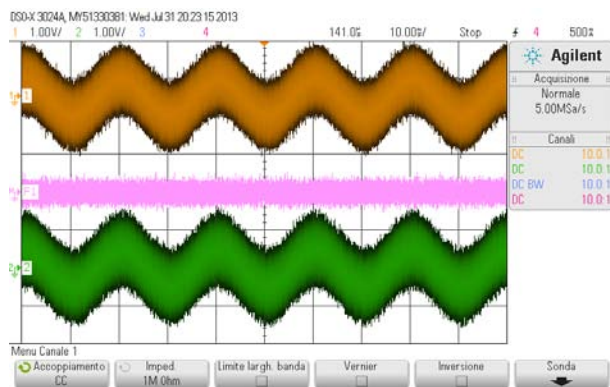


Figura 18: Risultato dell'oscilloscopio con l'EEE off

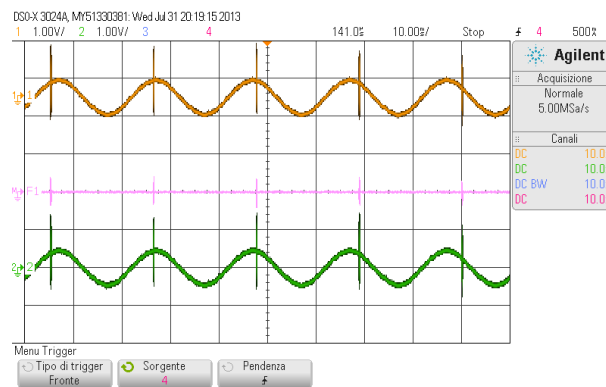


Figura 19: Risultato dell'oscilloscopio con l'EEE on

3.3 Le esperienze eseguite in laboratorio

Per riuscire a rilevare i vari tempi di nostro interesse, bisogna effettuare diverse misure. Le esperienze saranno presentate nella sequenza con cui sono state eseguite, in modo da riproporre il percorso che ha portato ai risultati finali.

Le prove effettuate sono state, nell'ordine:

1. Misure sul tempo di quiet T_q
2. Misure sul tempo di refresh T_r
3. Misure con generatore di pacchetti esterno 1
4. Misure di corrente sulla scheda
5. Misure con generatore di pacchetti esterno 2

3.3.1 Misure sul tempo di quiet T_q

Il primo tempo che andiamo a rilevare è quello di quiet T_q . Infatti per riuscire ad avere il passaggio tra gli stati di *Quiet* e di *Refresh* basta lasciare il link in una situazione di "riposo" e autonomamente passerà dallo stato di *Quiet* a quello di *Refresh*. Le varie prove effettuate hanno portato a dei risultati sempre riconducibili alla Figura 20. Nel segnale differenziale si possono osservare numerosi picchi. Con un'analisi sufficientemente attenta si può osservare che tali picchi sono tutti equidistanti tra loro con una distanza tra la fine di un picco e l'inizio del successivo di circa 24 ms.

Questo è un primo importante risultato per le misure che vogliamo effettuare, in quanto rispecchia a pieno quanto imposto dallo standard, $T_q^{min}=22$ ms e $T_q^{max}=24$ ms.

3.3.2 Misure sul tempo di refresh T_r

Le prove effettuate per la rilevazione del tempo di refresh T_r non sono altro che un approfondimento delle misure appena viste per il tempo di quiet. Infatti il tempo di refresh non è altro che la durata del picco.

La Figura 21 mostra nel dettaglio uno dei picchi che si potevano vedere nel segnale differenziale della Figura 20. Il segnale al centro è quello di nostro interesse, mentre quello superiore è il segnale di una sonda di corrente.

Si può vedere nella Figura 21 che la durata di un segnale di refresh è di circa $220\mu s$.

Come in precedenza anche questo è un risultato importante, infatti anche questo è in pieno accordo con quanto dettato dallo standard, $T_r^{min}=200\mu s$ e $T_r^{max}=220\mu s$.

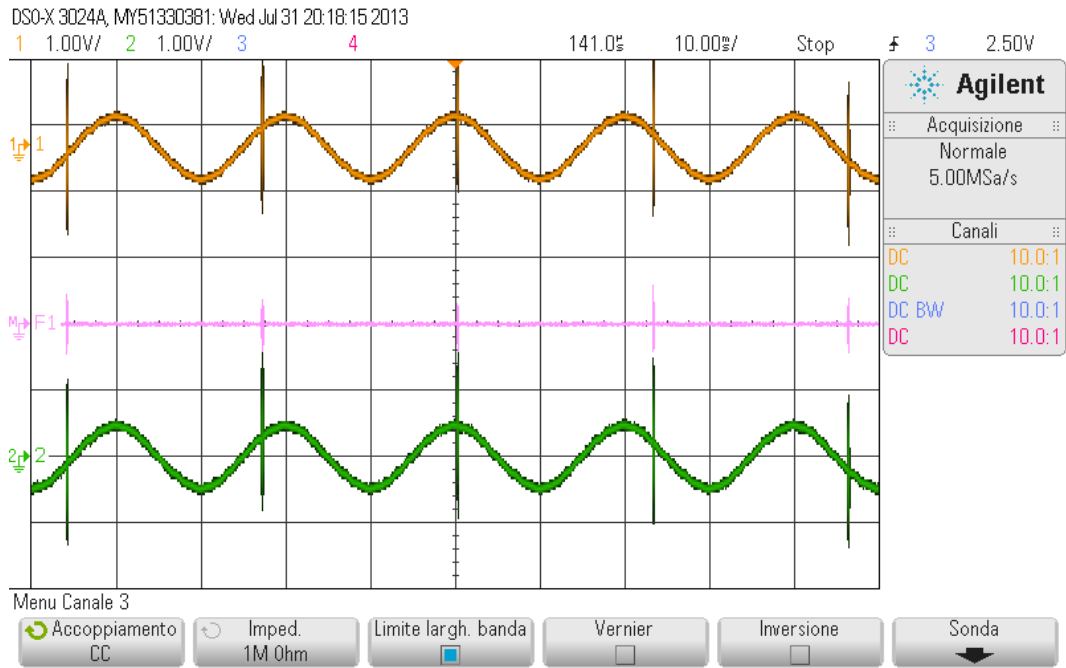


Figura 20: Risultato oscilloscopio misura T_q



Figura 21: Risultato oscilloscopio misura T_r

3.3.3 Misure con generatore di pacchetti esterno 1

Rilevati i primi due tempi rilevabili con il sistema a “riposo”, passiamo agli altri due tempi: tempo di sleep T_s e tempo di wake T_w . Questi due tempi si possono osservare quando la scheda compie le transizioni dallo stato attivo a quello di LPI e viceversa. Per ottenere tali transizioni bisogna stimolare il sistema imponendo delle trasmissioni dati. Ovviamente le misure risultano più semplici ed accurate se si possono controllare i pacchetti che vengono trasmessi. A questo scopo torna utile Arduino Ethernet, già introdotto in precedenza, come generatore di pacchetti.

Le misure sono state così effettuate:

1. È stato programmato il microcontrollore in modo tale da generare dei pacchetti ethernet ed inviarli al computer con la scheda in esame ad intervalli prefissati.
2. Si effettuano le misure con l'oscilloscopio come fatto per le misure precedenti
3. Si registrano ed elaborano in un foglio di calcolo i risultati

Il programma caricato in Arduino genera un pacchetto ethernet che viene trasmesso con le seguenti tempistiche:

- invia quattro pacchetti col minor intervallo di tempo possibile per Arduino
- invia due pacchetti con $2ms$ di ritardo rispetto il precedente
- invia due pacchetti con $4ms$ di ritardo rispetto il precedente
- invia due pacchetti con $6ms$ di ritardo rispetto il precedente
- invia due pacchetti con $8ms$ di ritardo rispetto il precedente
- invia due pacchetti con $10ms$ di ritardo rispetto il precedente
- invia due pacchetti con $23ms$ di ritardo rispetto il precedente così da anticipare di poco il refresh

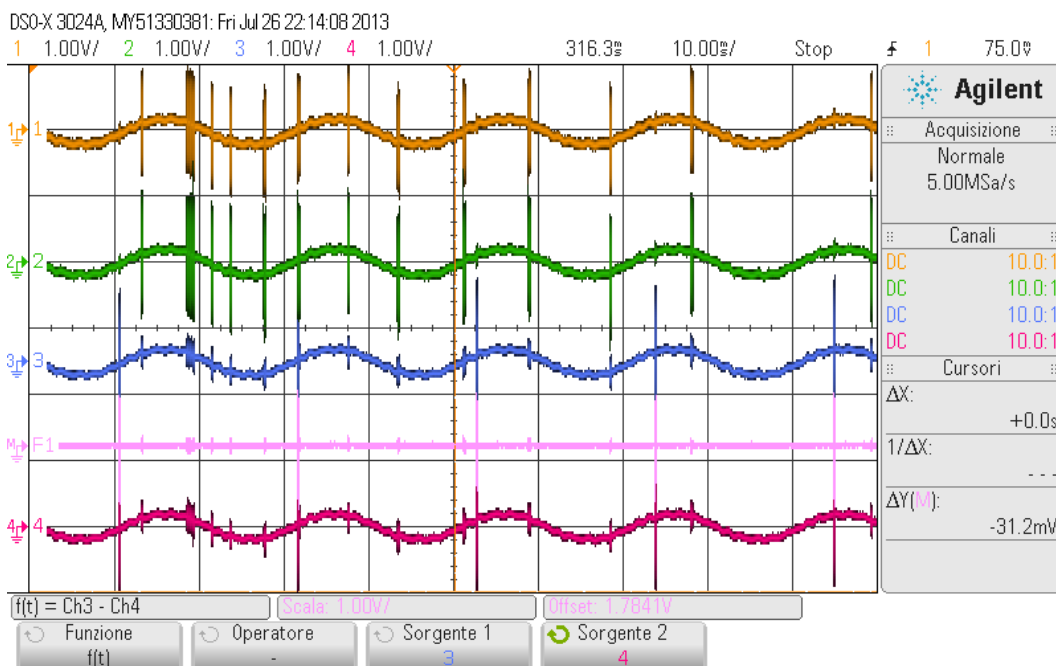


Figura 22: Risultato oscilloscopio sequenza pacchetti

Come si può notare alla Figura 22, questa volta le misure sono state effettuate su due coppie di fili, in particolare essendo impostato il bit-rate a 100Mbit/s sono le due coppie impegnate nello scambio di dati, una per la ricezione ed una per la trasmissione. Il segnale differenziale è stato calcolato dalla coppia di ricezione in quanto stiamo misurando i tempi dei pacchetti in arrivo dal microcontrollore. Sull'altra coppia di fili possiamo trovare dei picchi più elevati indipendenti da quelli della prima coppia, mentre quelli più piccoli risultano essere disturbi provenienti dal segnale sulla prima coppia. Si può notare come tra i picchi principali della seconda coppia vi sia un intervallo di circa 24ms . Infatti questi picchi risultano essere dei segnali di refresh sulla coppia di trasmissione. Questo è possibile in quanto le schede ethernet a 100Mbit/s utilizzano due canali indipendenti e a tale velocità di trasmissione l'Energy Efficiency è stato implementato in modo che anche solo uno dei due canali possa andare in LPI, mentre il secondo può rimanere comunque attivo. Purtroppo queste misure non hanno portato a dei risultati direttamente utili a giungere al nostro obiettivo di rilevare i tempi di sleep e di wake, ma possono essere un buon punto di partenza per le misure successive.

3.3.4 Misure di corrente sulla scheda

Siccome con l'ultima misura, non siamo riusciti a rilevare i tempi che ci interessano, proviamo allora a cambiare approccio. Inseriamo nel sistema anche la sonda di corrente. La sonda occupa un ingresso dell'oscilloscopio e misura la corrente in modo indiretto. Essa infatti funziona come una morsa amperometrica ovvero misura il campo magnetico generato dalla corrente in transito nel cavo di interesse che viene fatto passare all'interno della morsa. Una delle comodità di questo strumento risiede nel fatto che non è parte integrante del circuito, in quanto non occorre modificare il sistema per poter inserirlo.

Proprio per questa peculiarità, si riesce ad utilizzare la sonda sulla prolunga PCI 1x. La prolunga è stata opportunamente modificata, dopo un'attento studio del datasheet del connettore PCI 1x, in modo da unire in un unico cavo tutti i fili tra loro in parallelo, andando così a formare il cavo verde per l'ingresso a $+3.3\text{V}$ e il cavo marrone per il GND .



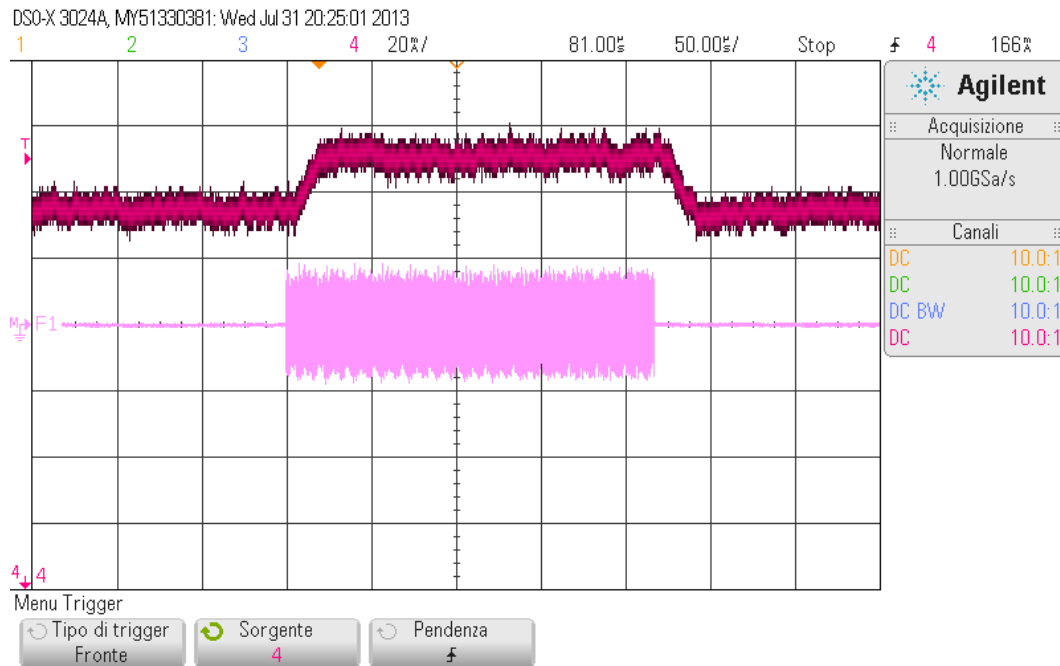
Figura 23: Sonda amperometrica "Agilent 1147A 50 MHz"



Figura 24: Collegamento della scheda al computer per mezzo della prolunga modificata

Con la prolunga così modificata per effettuare delle misure amperometriche sulla scheda è sufficiente applicare la sonda al cavo verde.

Già dalle prime misure si è notato che le correnti in gioco sono eccessivamente ridotte per la sonda e quindi sono state create quattro spire col cavo verde tutte passanti all'interno della morsa della sonda. Così facendo la corrente attraversa la morsa della sonda una volta per ogni spira ottenendo che la sonda misura una corrente quattro volte la corrente effettivamente assorbita dalla scheda. Per le immagini riportate in questo elaborato sono state effettuate misure senza spire, quindi la corrente viene conteggiata una sola volta.



I datasheet della scheda forniti dal costruttore Intel danno le seguenti informazioni riguardo le caratteristiche elettriche:

Condizione	Velocità [Mbit/s]	+3.3V Current-External [mA]
Active link		75.5
Idle link-EEE disabilitato	100	54.5
Idle link-EEE abilitato		33.6

Tabella 7: Caratteristiche elettriche scheda ethernet Intel I210

Osserviamo che le ampiezze delle correnti misurate si discostano di un valore rilevante rispetto ai valori del datasheet.

Queste discrepanze si possono attribuire allo strumento non adatto alle misure in questione, infatti il fondoscala dello strumento è 15A mentre le misure di nostro interesse sono di circa 30mA ovvero lo 0.2% del valore massimo che lo strumento riesce a leggere. Il costruttore garantisce una sensibilità del $\pm 1\%$ quindi anche utilizzando lo stratagemma delle spire, la corrente misurata dalla sonda rimane insufficiente per ottenere delle misure affidabili.

Dal punto di vista dei tempi anche questo approccio non ci fornisce ulteriori informazioni se non che la scheda ha un assorbimento di energia strettamente legato al suo fattore di attività. Essa accresce il suo assorbimento solamente quando sul cavo risulta esservi segnale, con un leggero ritardo, ma di medesima estensione.

3.3.5 Misure con generatore di pacchetti esterno 2

Dal momento che con le misure di corrente non abbiamo ottenuto i risultati sperati, ritorniamo al nostro microcontrollore, utilizzato come generatore di pacchetti. Questa volta invece di sfruttare la possibilità di variare l'intervallo di tempo tra un pacchetto e l'altro, utilizziamo la capacità di Arduino di creare pacchetti di dimensioni specificate.

Le dimensioni minime di un pacchetto ethernet sono

$$8\text{byte}[\text{preamble}] + 14\text{byte}[\text{header}] + 46\text{byte}[\text{payload}] + 4\text{byte}[\text{trailer}] = 72\text{byte} \quad (1)$$

ovvero $72 \times 8 = 576\text{bit}$.

Se proviamo ad imporre ad Arduino di generare un pacchetto con un payload inferiore ai 46 byte, il microcontrollore è obbligato ad inserire dei bit fino a raggiungere la dimensione minima. Questo è importante ricordarlo in quanto delle misure effettuate tentando di far generare dei pacchetti con payload di dimensione inferiore al valore minimo produrrebbero dei risultati errati.

Inoltre Arduino è stato programmato in modo tale che esegua le tre operazioni seguenti per ogni pacchetto che invia:

1. Alza al livello logico alto "1" un bit di uscita
2. Inizia la trasmissione del pacchetto
3. Riporta al livello logico basso "0" il bit alzato prima

Con questa sequenza di comandi si riesce ad ottenere un segnale di trigger che inizia un attimo prima dell'inizio della trasmissione e finisce un attimo dopo che è iniziato l'invio del pacchetto. Con questo speriamo di rilevare con maggior precisione i tempi di nostro interesse o per lo meno, impostato l'oscilloscopio in modo da leggerlo come segnale di trigger, rende più agevoli la lettura delle misure.

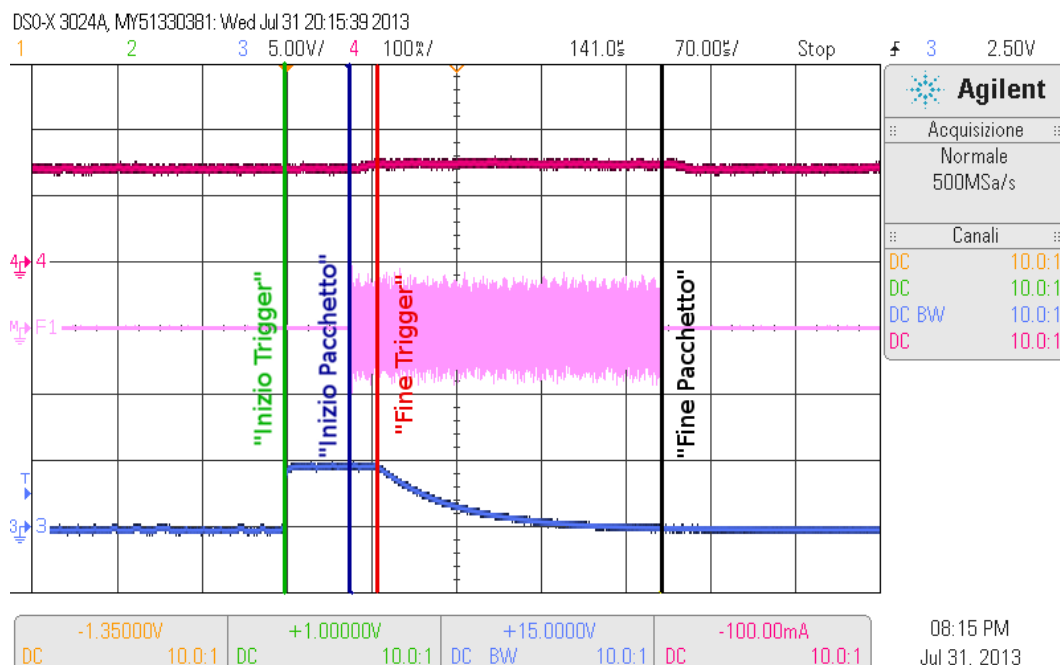


Figura 25: Misura pacchetto con segnale di trigger

Nella misura in Figura 25 sono stati messi in evidenza i punti di riferimento utilizzati per la rilevazione dei tempi.

Nelle Tabelle 8, 9 e 10 riportiamo i valori medi delle misure effettuate:

Payload		Valori medi [μs]
Byte	bit	
125	1000	51.4
250	2000	72.4
375	3000	94.2
500	4000	115
625	5000	135

Tabella 8: Misure **“Inizio Trigger”-“Inizio Pacchetto”**

Payload		Valori medi [μs]
Byte	bit	
125	1000	26.2
250	2000	5.57
375	3000	19.9
500	4000	-1.91
625	5000	-22.8

Tabella 9: Misure **“Inizio Pacchetto”-“Fine Trigger”**

Payload		Valori medi [μs]
Byte	bit	
125	1000	232
250	2000	263
375	3000	258
500	4000	290
625	5000	321

Tabella 10: Misure **“Fine Trigger”-“Fine Pacchetto”**

Dalle misure appena mostrate si ricavano direttamente le estensioni di tempo del pacchetto e del trigger (Tabelle 11 e 12):

Payload		Valori medi [μs]
Byte	bit	
125	1000	77.5
250	2000	78.0
375	3000	114
500	4000	113
625	5000	113

Tabella 11: Misure **“Inizio Trigger”-“Fine Trigger”**

Payload		Valori medi [μs]
Byte	bit	
125	1000	258
250	2000	269
375	3000	278
500	4000	289
625	5000	298

Tabella 12: Misure **“Inizio Pacchetto”-“Fine Pacchetto”**

Bisogna specificare che il tempo del pacchetto misurato è così composto:

$$Tempo\ pacchetto\ misurato = Tempo\ wake + Tempo\ pacchetto\ ethernet + Tempo\ sleep \quad (2)$$

Però, come già menzionato, noi siamo in grado di dire esattamente quanto vale il *Tempo pacchetto ethernet*. Sapendo che la trasmissione avviene a 100Mbit/s , per calcolare il tempo T_e necessario per trasmettere il pacchetto ethernet basta utilizzare la formula

$$T_e = \frac{x}{100 \cdot 10^6} \quad (3)$$

dove x è la dimensione del pacchetto ethernet completo di preamble, header e trailer.

Possiamo quindi per ogni dimensione conoscerne il tempo necessario per trasmettere un pacchetto:

Osserviamo ora la figura 26, ove sono stati inseriti i grafici delle misure finora presentate.

È interessante osservare come il tempo per trasmettere il pacchetto cresca in modo lineare con il crescere della dimensione del pacchetto stesso, mentre la durata del trigger ha un comportamento che non ci si aspettava, infatti risulta essere una spezzata, quando invece ci si aspettava un comportamento che seguisse quello del pacchetto. Il motivo di questo è dovuto con alta probabilità ad una questione di ciclo e di clock del processore di Arduino.

Dimensione Payload Byte	Dimensione Pacchetto		Tempo Trasmissione Ethernet
	Byte	bit	μs
125	151	1208	12.1
250	276	2208	22.1
375	401	3208	32.1
500	526	4208	42.1
625	651	5208	52.1

Tabella 13: Tempi di trasmissione ethernet

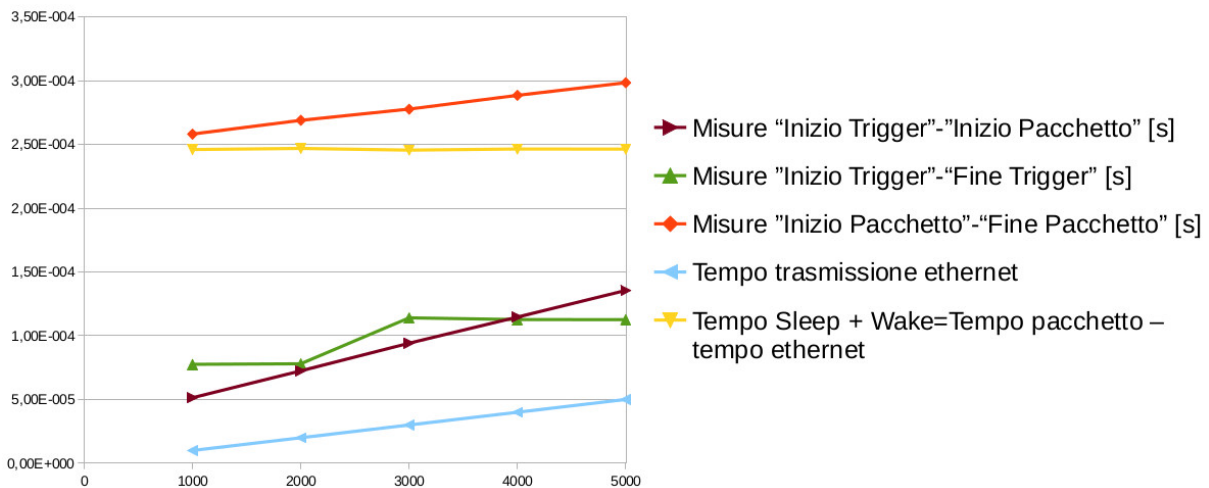


Figura 26: Grafico riassuntivo tempi al variare della dimensione dei pacchetti

Oltre ai grafici delle misure finora presentate, è stata aggiunto il grafico del “Tempo Sleep + Wake” che si ottiene semplicemente con:

$$T_{Sleep + Wake} = T_{pacchetto\ misurato} - T_{trasmissione\ ethernet} \quad (4)$$

La cosa da notare è il valore all’incirca costante a $246\mu s$ che assume questa misura, a prescindere dalla dimensione del pacchetto. Questo valore è proprio quello che ci si aspettava. Infatti lo standard afferma che il *Tempo di Sleep* può valere fino a $220\mu s$ e il *Tempo di Wake* può valere fino a $20.5\mu s$, che sommati danno $240.5\mu s$. Il nostro risultato si distanzia di circa il 2.2% dal valore dello standard. Come risultato è accettabile tenendo conto di eventuali errori che, per quanto accurata possa essere la misura, sempre incorrono, per disturbi o per approssimazioni.

3.4 Conclusioni delle misure via hardware

Le misure via hardware hanno dato risultati molto interessanti. Siamo riusciti a rilevare:

- il tempo di *Quiet* di $24000\mu s$
- il tempo di *Refresh* di $220\mu s$
- la somma dei tempi di *Sleep+ Wake* di $246\mu s$

Tutte le misure rilevate sono concordi con i valori dello standard.

Con gli strumenti a nostra disposizione non possiamo spingerci oltre e non è possibile scindere i tempi di *Sleep* e di *Wake*. Per poter ottenere questi valori bisognerebbe avere a disposizione un oscilloscopio in grado di riconoscere il frame ethernet, cosicché si possa rilevare la posizione esatta del pacchetto contenente l’informazione all’interno del pacchetto misurato fino ad ora. Una volta

riconosciuto il frame, misurare il tempo di *Wake* diventa semplice in quanto basta osservare il tempo di attività sul cavo ethernet prima del frame, e stessa cosa per il tempo di *Sleep* osservando però dopo il frame, invece che prima.

Possiamo comunque avere un'idea generale delle dimensioni dei due, interpolando le misure software con quelle hardware. Infatti con le misure software siamo giunti a dire che il pacchetto impiega un tempo dell'ordine di $20\mu s$ in più quando l'EEE è attivo. Dal momento che la scheda, quando è in trasmissione/ricezione, si comporta allo stesso modo che l'EEE sia attivo o meno, si può dedurre che il tempo così rilevato sia il tempo di wake più ritardi di varia origine. Quindi possiamo dire che il tempo di wake è dell'ordine di $20\mu s$ e il tempo di sleep si aggira attorno alle $200\mu s$, deducendolo dal valore della somma che abbiamo misurato con l'oscilloscopio.

4 Misure a 1Gbit/s

Le misure con bit-rate 1Gbit/s sono state eseguite nello stesso modo con cui sono state eseguite quelle a 100Mbit/s, quindi di seguito non verranno approfondite le misure, ma verranno presentati solo i risultati più rilevanti.

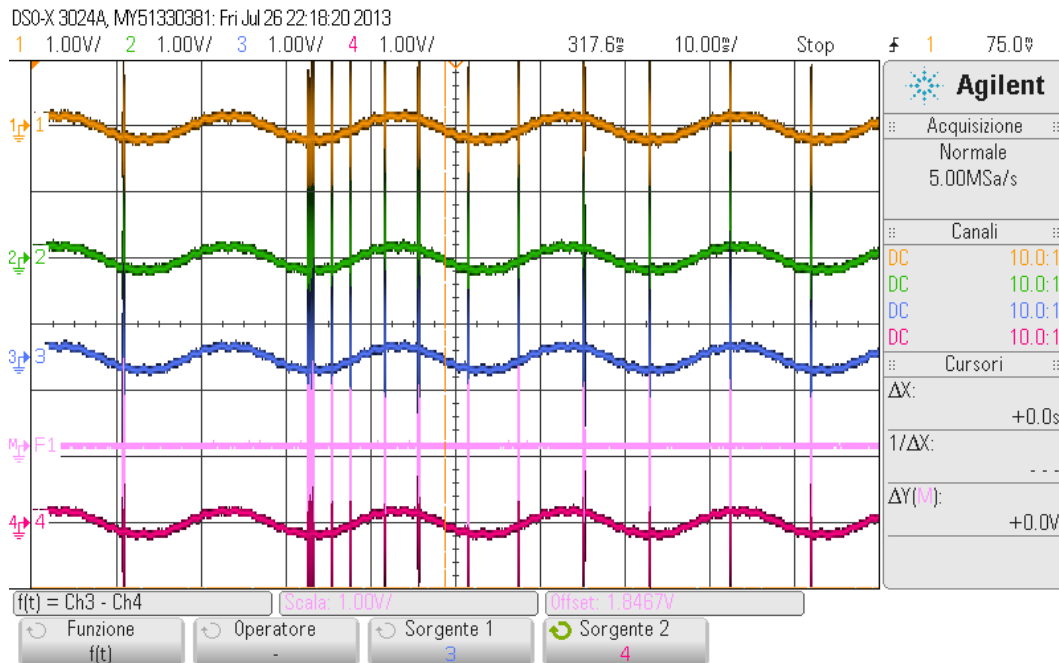


Figura 27: Risultato oscilloscopio sequenza pacchetti a 1Gbit/s

Con la figura 27 si vuole far notare una forte differenza rispetto al bit-rate a 100Mbit/s. Mentre prima si avevano solo due canali attivi su quattro e indipendenti l'uno dall'altro, adesso sono attivi tutti e quattro i canali ed il segnale viene inviato su tutti contemporaneamente.

Fatta questa osservazione si può passare alle misure:

1. Misure sul tempo di quiet T_q
2. Misure sul tempo di refresh T_r
3. Misure con generatore di pacchetti esterno

4.1 Misure sul tempo di quiet T_q

Come fatto in precedenza, il primo tempo che andiamo a rilevare è quello di quiet T_q . Le varie prove effettuate hanno portato a dei risultati sempre riconducibili alla Figura 28. Anche in questo caso si riesce a rilevare che la distanza tra la fine di un picco e l'inizio del successivo di circa 24ms, come riferito dallo standard.

4.2 Misure sul tempo di refresh T_r

Esaminiamo ora uno dei picchi per la lettura del tempo di refresh.

Nella Figura 29 si può notare all'inizio del pacchetto un picco che anticipa il resto del pacchetto per un tempo di circa 2μs. Dal momento che lo si può ritrovare in qualsiasi segnale che viene trasmesso sul cavo ethernet sempre della stessa "forma", che sia un pacchetto dati o un semplice segnale di refresh, è lecito supporre che sia un segnale di sincronismo per le schede ed i segnali sui quattro canali.

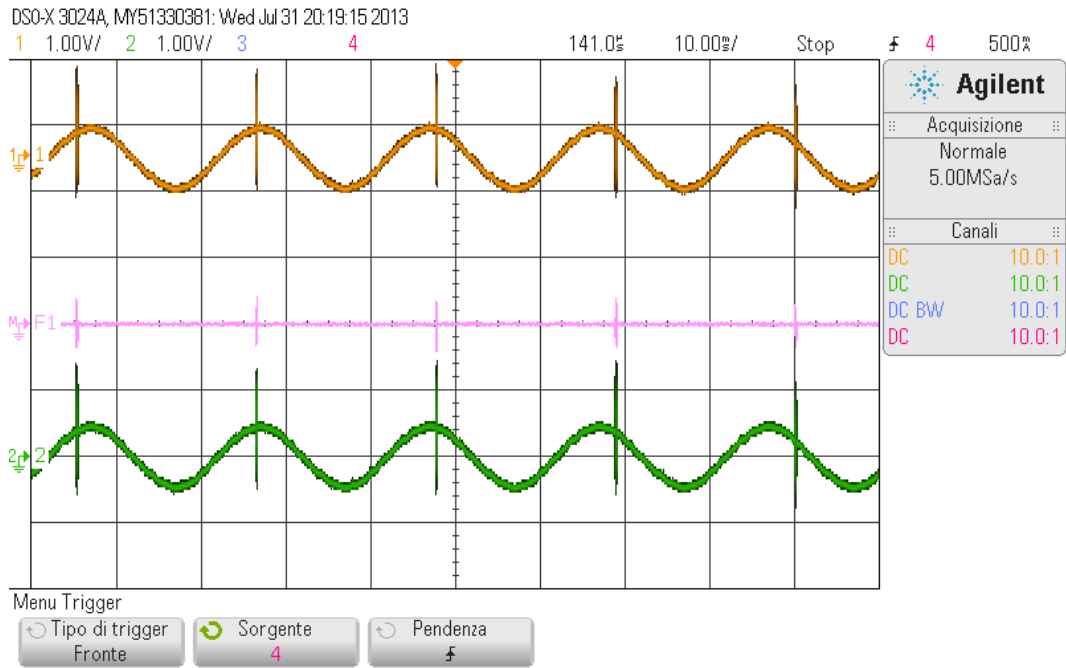


Figura 28: Risultato oscilloscopio misura T_q

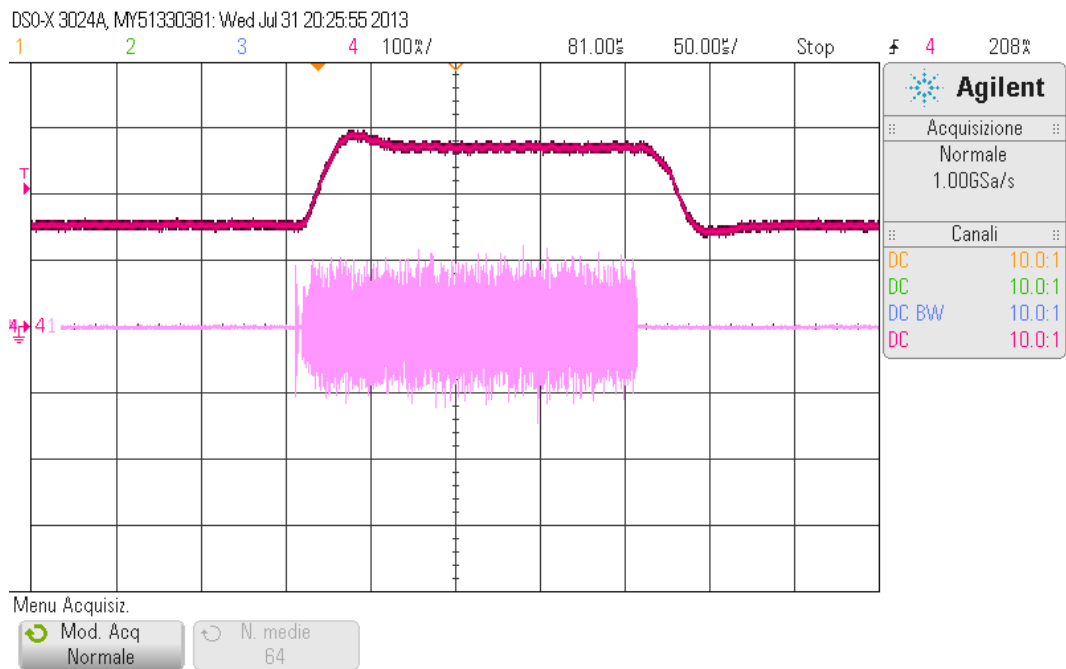


Figura 29: Risultato oscilloscopio misura T_r

Nonostante questa differenza rispetto al bit-rate di 100Mbit/s , possiamo osservare di come il tempo di refresh, tenendo conto anche del picco iniziale, risulti essere all'incirca $T_r = 202\mu\text{s}$, anch'esso rientrante nei limiti imposti dallo standard.

4.3 Misure con generatore di pacchetti esterno

Come fatto in precedenza, anche in questo caso programmiamo Arduino in modo da generare i pacchetti delle varie dimensioni con il corrispondente trigger.

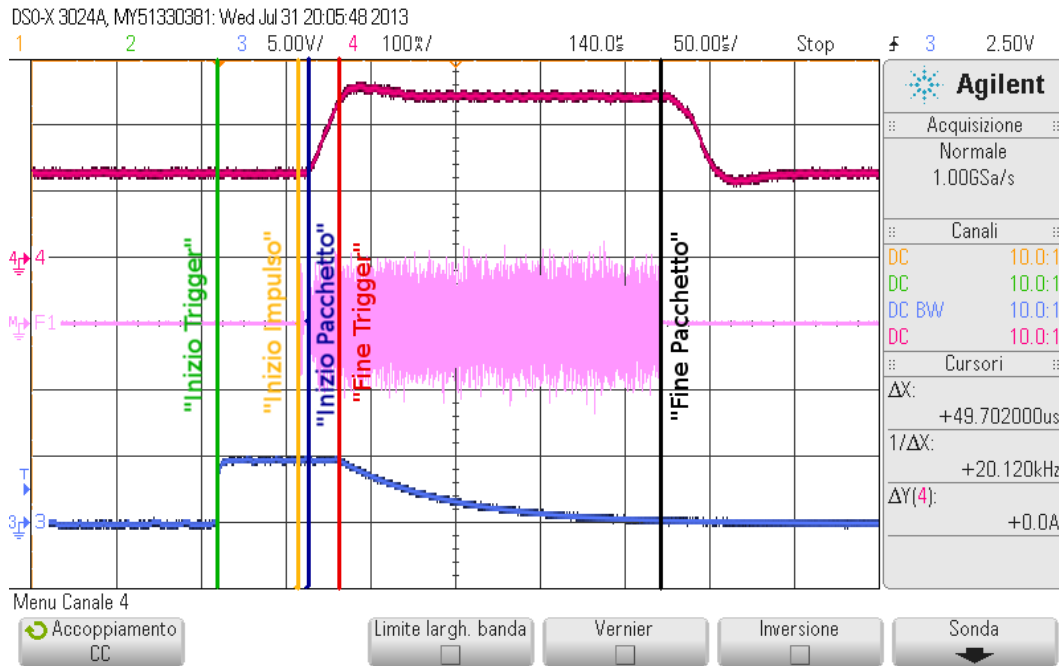


Figura 30: Misura pacchetto con segnale di trigger

Nella misura in Figura 30 sono stati messi in evidenza i punti di riferimento utilizzati per la rilevazione dei tempi. Rispetto alla medesima misura effettuata a 100Mbit/s , in questa è stato aggiunto il riferimento al “Inizio Impulso”.

Nelle Tabelle 14, 15, 16 e 17 riportiamo i valori medi delle misure effettuate:

Byte	Payload bit	Valori medi [μs]
125	1000	49.6
250	2000	70.6
375	3000	91.5
500	4000	113
625	5000	133

Tabella 14: Misure “Inizio Trigger”- “Inizio Impulso”

Byte	Payload bit	Valori medi [μs]
125	1000	20.1
250	2000	-0.85
375	3000	12.5
500	4000	-8.28
625	5000	-28.7

Tabella 15: Misure “Inizio Impulso”- “Fine Trigger”

Payload		Valori medi [μs]
Byte	bit	
125	1000	193
250	2000	212
375	3000	201
500	4000	222
625	5000	244

Tabella 16: Misure **“Fine Trigger”-“Fine Pacchetto”**

Payload		Valori medi [μs]
Byte	bit	
125	1000	4.48
250	2000	4.54
375	3000	4.50
500	4000	4.49
625	5000	4.47

Tabella 17: Misure **“Inizio Impulso”-“Inizio Pacchetto”**

Dalle misure appena mostrate si ricavano direttamente le estensioni di tempo del trigger e del pacchetto (Tabelle 18 e 19):

Payload		Valori medi [μs]
Byte	bit	
125	1000	69.7
250	2000	69.7
375	3000	104
500	4000	104
625	5000	105

Tabella 18: Misure **“Inizio Trigger”-“Fine Trigger”**

Payload		Valori medi [μs]
Byte	bit	
125	1000	213
250	2000	211
375	3000	213
500	4000	214
625	5000	215

Tabella 19: Misure **“Inizio Pacchetto”-“Fine Pacchetto”**

Possiamo utilizzare la stessa formula usata in precedenza, adeguandola alla velocità di trasmissione:

$$T_e = \frac{x}{1 \cdot 10^9} \quad (5)$$

Il tempo necessario per trasmettere un pacchetto, per ogni dimensione, è:

Dimensione Payload	Dimensione Pacchetto		Tempo Trasmissione Ethernet
	Byte	bit	
125	151	1208	1.21
250	276	2208	2.21
375	401	3208	3.21
500	526	4206	4.21
625	651	5206	5.21

Tabella 20: Tempi di trasmissione ethernet

Ed ora presentiamo il grafico delle misure e dei tempi indicati, in funzione della dimensione del payload:

Nella Figura 31 possiamo osservare dei grafici molto simili a quelli visti per il $100Mbit/s$. Si può notare anche in questo caso la spezzata del tempo di trigger, a rafforzare l'ipotesi fatta in precedenza. Anche in questa figura c'è il grafico del **“Tempo Sleep + Wake”** che questa volta si ottiene con:

$$T_{Sleep + Wake} = T_{pacchetto\ misurato} - T_{trasmissione\ ethernet} - T_{[“Inizio Impulso”-“Inizio Pacchetto”]} \quad (6)$$

La differenza sta nel fatto che è stato sottratto anche il tempo tra l'inizio dell'impulso e l'inizio del pacchetto che nelle misure a $100Mbit/s$ non era presente. Come in precedenza anche questa misura assume un valore all'incirca costante a $206\mu s$ a prescindere dalla dimensione del pacchetto. Questo valore è proprio quello che ci si aspettava. Infatti lo standard afferma che il *Tempo di Sleep* può

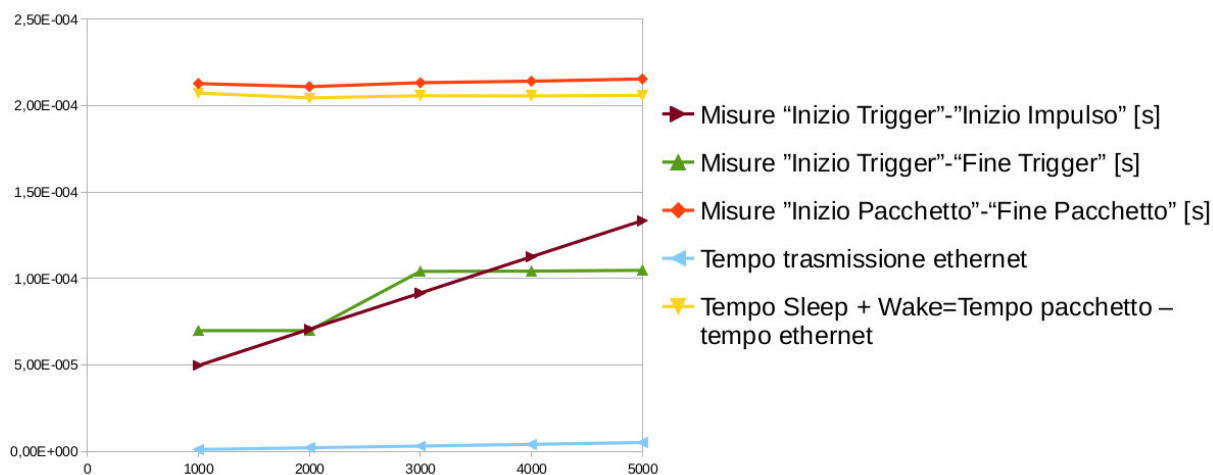


Figura 31: Grafico riassuntivo tempi al variare della dimensione dei pacchetti

valere da $182\mu s$ fino a $202\mu s$ e il *Tempo di Wake* può valere fino a $16.5\mu s$, il che, sommati, danno un valore da $198.5\mu s$ fino a $218.5\mu s$. Il risultato da noi ricavato è in pieno accordo con quanto imposto dallo standard.

5 Conclusioni

Con le misure mostrate siamo riusciti a verificare che la scheda risponde ai vincoli imposti dallo standard per quanto riguarda il tempo di refresh e per il tempo di quiet, sia a 100Mbit/s che a 1Gbit/s. Per il tempo di sleep e il tempo di wake siamo riusciti a determinarne solo la somma, ma come già detto in precedenza, con gli strumenti a nostra disposizione non siamo in grado di approfondire ulteriormente lo studio. Per ottenere il valore dei due tempi separati bisognerebbe usare un oscilloscopio in grado di riconoscere i frame ethernet all'interno del segnale letto sul cavo. Una volta riconosciuto il frame, il tempo di wake è il tempo di attività sul cavo che ne precede l'inizio, mentre il tempo di sleep è il tempo di attività sul cavo che ne segue la fine.

Riferimenti bibliografici

- [1] Energy Efficient Ethernet for the Industrial Communication Scenario, Federico Tramarin and Stefano Vitturi, CNR-IEIIT, Padova
- [2] Energy Efficient Ethernet for Real-Time Industrial Networks, S. Vitturi, Member, IEEE, and F. Tramarin, Member, IEEE, Padova
- [3] IEEE Standard 802.3azTM-2010
- [4] Datasheet “Intel®Ethernet Controller I210”
- [5] Datasheet “D-Link® DGS-1100 Series EasySmart Switches”
- [6] Datasheet “Agilent InfiniiVision 3000 X-Series Oscilloscopes”
- [7] Datasheet “Agilent 1147A 50 MHz Current Probe”