

ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

SCUOLA DI SCIENZE
Corso di Laurea Magistrale in Informatica

Time Sensitive Networking

Sviluppo di un ambiente di test

Relatore:
Chiar.mo Prof.
Marco Prandini

Presentata da:
Marco Negrini

Correlatori:
Chiar.mo Prof. Renzo Davoli
Ph.D. Andrea Melis
Dott. Mag. Guido Munari
Dott. Mag. Davide Berardi

II sessione - secondo appello
2020/2021

*La più veloce delle macchine
sarà sempre lenta,
se ciò che cerchi
è a portata di mano.*

Abstract

In ambito industriale si hanno processi real-time, le cui azioni devono avvenire entro un intervallo di tempo. Non rispettare una scadenza può avere conseguenze disastrose. Questi processi possono essere portati avanti da più dispositivi, i quali necessitano di comunicazioni deterministiche per parlare tra di loro. Ethernet offre ottime prestazioni e ha dimostrato di poter stare al passo delle innovazioni tecnologiche, ma sono necessarie modifiche per potere garantire una comunicazione deterministica. Gli standard per le Time Sensitive Network(TSN) permettono di utilizzare Ethernet come protocollo real-time. È stato definito un ambiente di test per reti TSN, atto a replicare utilizzi industriali, per verificare il funzionamento degli standard TSN.

Introduzione

Durante questo lavoro di tesi si è implementato un ambiente di test per reti TSN (Time Sensitive Network). Gli standard TSN, definiti dal IEEE, apportano modifiche allo standard Ethernet, IEEE 802.3, tali da poterlo utilizzare come comunicazione real-time. Questo tipo di comunicazioni vengono utilizzate in ambito industriale, dove è diffuso l'interesse per le comunicazioni Ethernet.

Ethernet offre prestazioni molto maggiori dei bus di campo, ed ha mostrato di sapersi adattare alle innovazioni tecnologiche, essendo separato dal livello fisico utilizzato. Si è infatti evoluto da protocolli da 10Mbps a 100Mbps, successivamente 1Gpbs, e promette di andare oltre i 10Gpbs. Non è però adatto a dare le garanzie necessarie alla comunicazione di processi real-time. Nel capitolo 1 vengono introdotte le soluzioni utilizzate per bus di campo ed Ethernet real-time e l'implementazione fisica del MAC e dello switch.

Nel capitolo 2 viene descritto un ambiente di test, atto a verificare la funzionalità e le prestazioni di una rete TSN. Vengono illustrate le metodologie per misurare la latenza di comunicazione. La latenza è un intervallo di tempo tra due eventi che avvengono durante la comunicazione tra i vari dispositivi. Questi eventi devono poter essere rilevati e l'intervallo che intercorre deve essere misurato rispetto ad uno stesso clock. Tra le tecniche utilizzate per garantire una comunicazione real-time si sono prese in considerazione la priorità dei frame, Time Aware Shaper e la Frame Preemption. Per ciascuna di esse si sono descritte le performance attese.

Infine si è costruita una rete TSN su cui applicare i test descritti. I

risultati sono descritti nel capitolo 3.

Indice

Introduzione	iii
1 Stato dell'arte	1
1.1 Ethernet in ambito industriale	3
1.1.1 Evoluzione del protocollo Ethernet	4
1.1.2 Bus di campo	11
1.1.3 Ethernet real-time	12
1.2 Protocolli TSN	14
1.2.1 IEEE 802.1AS	16
1.2.2 IEEE 802.1Qbv	18
1.2.3 IEEE 802.1Qbu	23
2 Analisi progettuale	27
2.1 Ambiente di test	27
2.1.1 Topologia di test	27
2.2 Misura delle prestazioni della rete	29
2.2.1 Misura della latenza	31
2.3 Utilizzo dell'ambiente di test	35
3 Realizzazione test	39
3.1 Risultati	41
3.1.1 Test con priorità	41
3.1.2 Test frame preemption	43
3.1.3 Test Time aware shaper	46

Conclusioni	47
A OpenTSN	49
Bibliografia	51

Elenco delle figure

1.1	Standard IEEE 802.3 rapportato ai livelli del modello OSI . . .	2
1.2	Caratteristiche di una comunicazione deterministica	5
1.3	Esempio di struttura di un MAC 802.3 standard	6
1.4	Confronto tra hub e switch[12]	7
1.5	schema a blocchi di uno switch	8
1.6	Differenti modalità operative di uno switch	9
1.7	Tipologie di protocolli Ethernet real-time	12
1.8	frame IEEE 802.1Q	16
1.9	Processo di peer-delay-measurement	17
1.10	Gerarchia Grandmaster PTP	20
1.11	Finestre Time Aware Shaper, con guardband	20
1.12	schema di un device TSN compatibile con TAS	22
1.13	Divisione di un frame in frammenti	23
1.14	schema di un device TSN compatibile con FPE	25
2.1	Topologie tipicamente utilizzate in ambito industriale	28
2.2	Topologia di test	29
2.3	Ritardi di comunicazione in una rete, con lo switch utilizzato in modalità cut-through	31
2.4	Rappresentazione schematica del passaggio di un frame	32
2.5	Configurazioni per la misura della latenza	35
3.1	Risultati osservati sull'oscilloscopio	42
3.2	Visualizzazione grafica dei risultati con IEEE 802.1P	44

3.3	Visualizzazione grafica dei risultati con frame preemption . . .	45
A.1	struttura di OpenTSN riportata nel paper	50

Elenco delle tabelle

1.1	Priorità IEEE 802.1P[29]	19
3.1	Test effettuati utilizzando la sola priorità 802.1P	44
3.2	Test per valutare le prestazioni della FPE	45
3.3	Test effettuati utilizzando il TAS	46

Elenco delle sigle

CAN	Controller Area Network
FIFO	First In First Out
FPE	Frame Preemption
IEEE	Institute of Electrical and Electronics Engineers
MAC	Medium Access Control
TAS	Time Aware Shaper
TSN	Time Sensitive Networking

Capitolo 1

Stato dell'arte

In questo capitolo viene illustrata la storia dei protocolli time-sensitive: i casi d'uso e l'evoluzione delle soluzioni utilizzate, seguendo i bisogni tecnologici. I protocolli TSN modificano il protocollo Ethernet (IEEE 802.3), in modo da garantire funzionalità real-time. Verranno illustrate le modifiche apportate da alcuni standard esaminati: la priorità dei frame, il Time Aware Shaper, e la Frame Preemption.

Spesso si farà riferimento al modello ISO/OSI, in particolare per la distinzione tra i livelli fisico, collegamento, e applicativo, viene quindi ora introdotto.

Il modello OSI La comunicazione tra device comprende numerosi aspetti di cui bisogna tenere conto. L'International Organization for Standardization ha definito un modello per le reti di calcolatori, separando su sette differenti livelli le funzionalità che possono essere implementate. Ciascun livello definisce un insieme di funzionalità da implementare facendo uso di quelle fornite dal livello inferiore. Il modello nasce con l'obiettivo di semplificare lo sviluppo di nuovi protocolli, aumentando l'interoperabilità con quelli esistenti[8].

In immagine 1.1 si possono vedere i livelli del modello osi rapportati con lo standard IEEE 802.3, comunemente chiamato Ethernet. All'interno

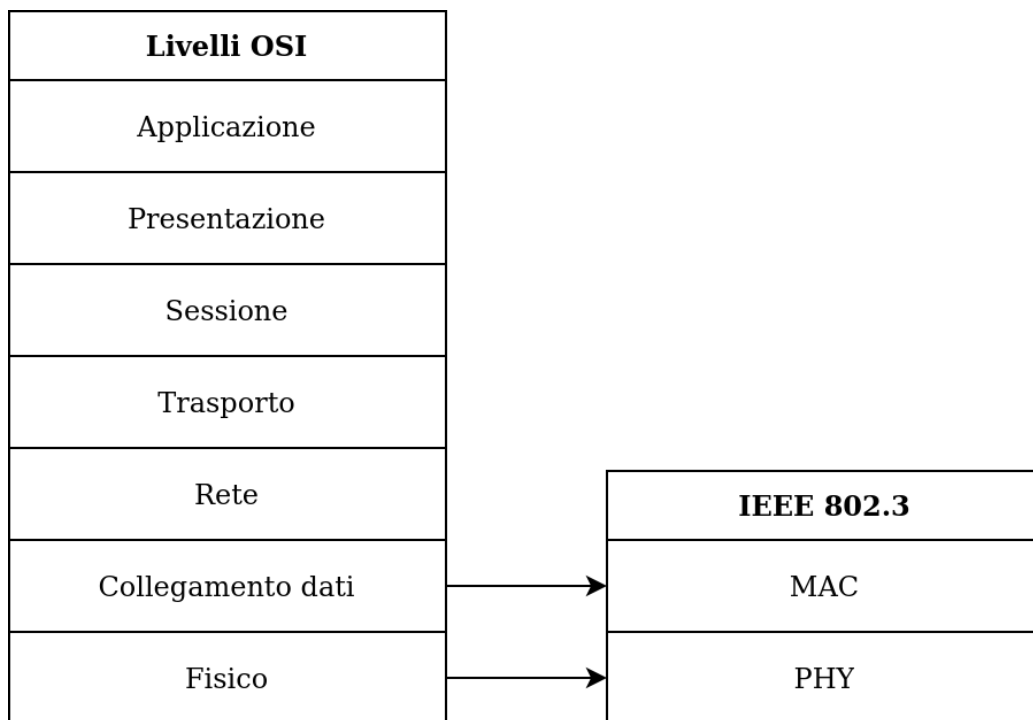


Figura 1.1: Standard IEEE 802.3 rapportato ai livelli del modello OSI

dello standard Ethernet vengono implementati il livello fisico e il livello di collegamento dati.

Il livello fisico si occupa di inviare e ricevere dati, dando un significato digitale agli aspetti analogici del medium di comunicazione. Il medium può essere utilizzato per fornire comunicazioni half-duplex o full-duplex. Nel caso specifico delle comunicazioni IEEE 802.3 il livello fisico è gestito da componenti chiamati PHY (dall'inglese physical layer).

Il livello di collegamento dati controlla l'utilizzo del livello fisico e definisce un pacchetto (o frame, nel caso di Ethernet) che contiene i dati. Nel frame Ethernet i dati vengono trasmessi all'interno di frame, che associano ai dati una lista di informazioni:

- MAC Address del mittente,
- MAC Address del destinatario,
- Ethernet Type, indica il tipo di contenuto,
- Cyclic redundancy check, che permette di verificare la correttezza del contenuto del frame, e di correggere al più un errore.

Ciascun device ha un indirizzo unico nella rete, il MAC Address.

I PHY e i MAC comunicano tramite un'interfaccia chiamata Media Independent Interface (MII). Dell'interfaccia MII esistono numerose varianti, che differiscono per velocità e numero di segnali.

1.1 Ethernet in ambito industriale

In ambito industriale si sono per lungo tempo utilizzati protocolli di comunicazione basati su bus di campo[5]. Si è solitamente utilizzato un protocollo seriale, in cui tutti i dispositivi comunicano contendendo lo stesso medium, chiamato bus.

Lo standard Ethernet offre prestazioni maggiori dei comuni bus seriali utilizzati in ambito industriale[26], ma non è possibile utilizzarlo per implementare comunicazioni real-time.

Comunicazioni real-time I sistemi real time sono sistemi che devono soddisfare requisiti real-time, ovvero, in cui le azioni devono venir completate entro un determinato intervallo di tempo.

La fine di questo intervallo di tempo è anche chiamata deadline. Le comunicazioni, per esempio l'invio di un comando, o la lettura di un sensore, devono anch'esse terminare entro un intervallo di tempo finito e calcolabile.

I sistemi real time si possono dividere tra hard-real-time e soft-real-time. Sforare una scadenza in un sistema soft-real-time può avere ripercussioni sulla performance, mentre in un sistema hard-real-time può avere conseguenze disastrose, per esempio danneggiare il sistema, o nel caso peggiore, una persona[27]. In immagine 1.2 si possono vedere le differenze tra una comunicazione non-deterministica e una deterministica. Si può notare che una comunicazione deterministica garantisce sempre la consegna di un pacchetto entro una determinata deadline. La deadline dipende dalle prestazioni del livello fisico che si utilizza.

Le comunicazioni possono essere periodiche o occasionali. Per le comunicazioni periodiche si può parlare di frequenza, o periodo, di aggiornamento.

1.1.1 Evoluzione del protocollo Ethernet

Quando Ethernet è stato standardizzato nel 1985 tutta la comunicazione era half-duplex e basata su un singolo bus condiviso tra tutti i dispositivi della rete. Per gestire l'accesso multiplo si utilizzava un algoritmo di contesa distribuito chiamato Carrier Sense Multiple Access with Collision Avoidance (CSMA/CD). Un nodo che ha un frame da trasmettere ascolta lo stato del cavo (Carrier Sense), e se lo sente libero inizia la trasmissione. Per via dei ritardi di propagazione due o più nodi possono iniziare la trasmissione contemporaneamente, e generare una collisione. Tutti i nodi coinvolti si interrompono, e aspettano un intervallo casuale prima di riprovare. Nel caso la collisione si ripeta l'intervallo di attesa viene aumentato. Dopo un massimo numero di tentativi il frame viene scartato. Tutti i device che hanno accesso

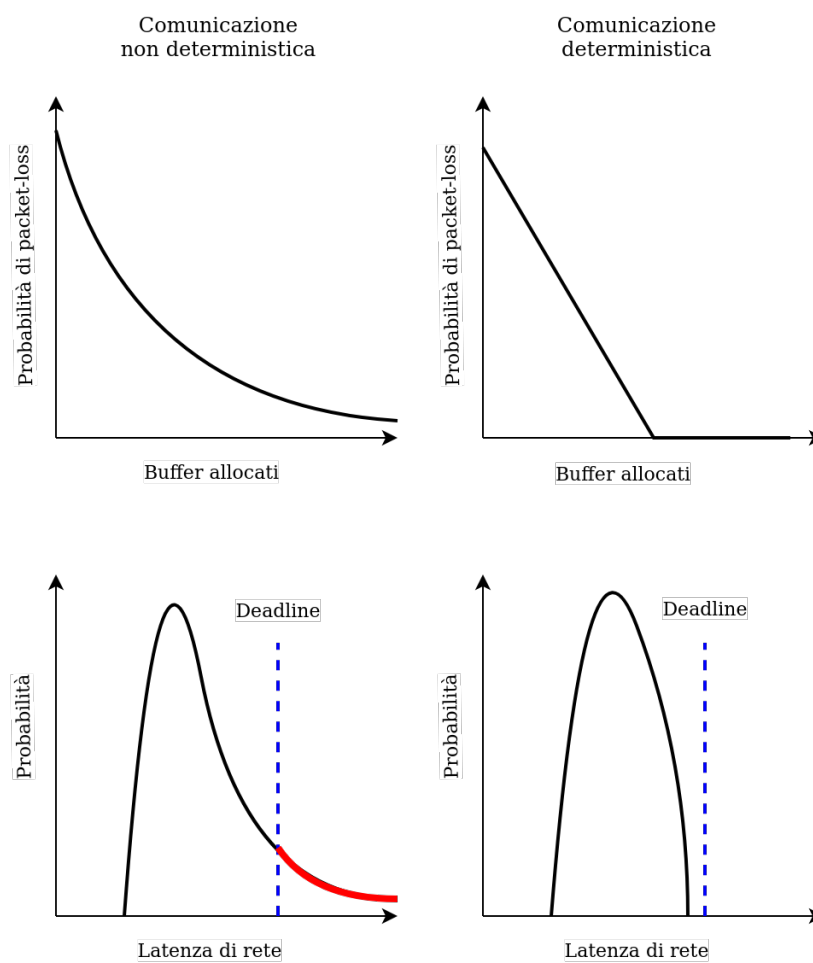


Figura 1.2: Caratteristiche di una comunicazione deterministica

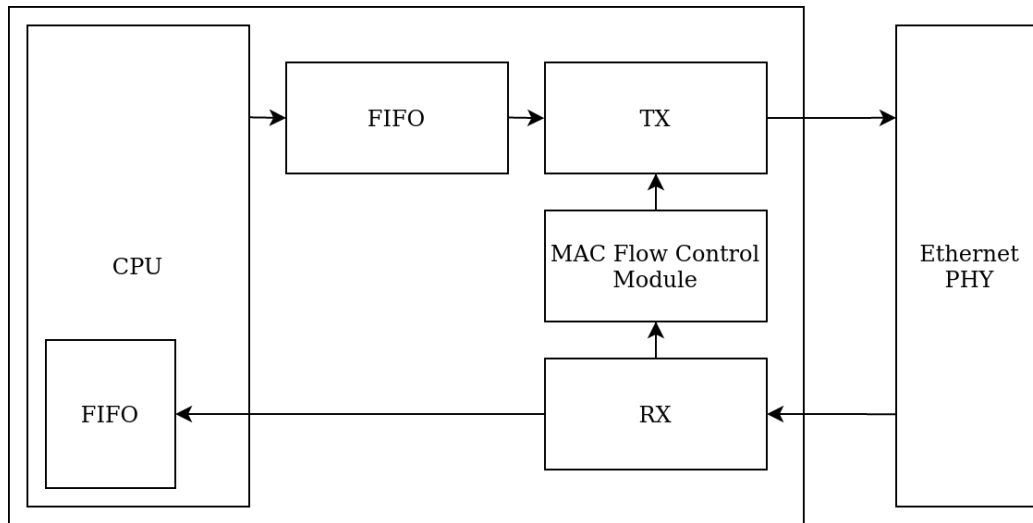


Figura 1.3: Esempio di struttura di un MAC 802.3 standard

allo stesso bus di comunicazione si dice che fanno parte dello stesso dominio di collisione.

La possibilità di collisioni implica inevitabilmente l'impossibilità di determinare un limite massimo del tempo di comunicazione.

Il MAC si occupa di inviare e ricevere frame Ethernet. In entrambe le direzioni è necessaria un'area di memoria in cui memorizzare temporaneamente i frame. Queste aree di memoria possono essere delle semplici code. Nel caso in cui il buffer di destinazione sia pieno il frame viene eliminato. Questo può avvenire se i frame ricevuti non vengono prelevati abbastanza velocemente o se si cerca di inviare più frame di quelli che il medium è capace di trasmettere.

Esempio implementazione MAC 802.3 In immagine 1.3 è possibile vedere un esempio di struttura di un componente MAC IEEE 802.3[22]. Sono stati riportati i componenti principali:

- la CPU accoda i frame da inviare all'interno di una coda FIFO
- il componente TX preleva i frame dalla coda e li trasmette verso il PHY tramite l'interfaccia MII, e applica CSMA/CD

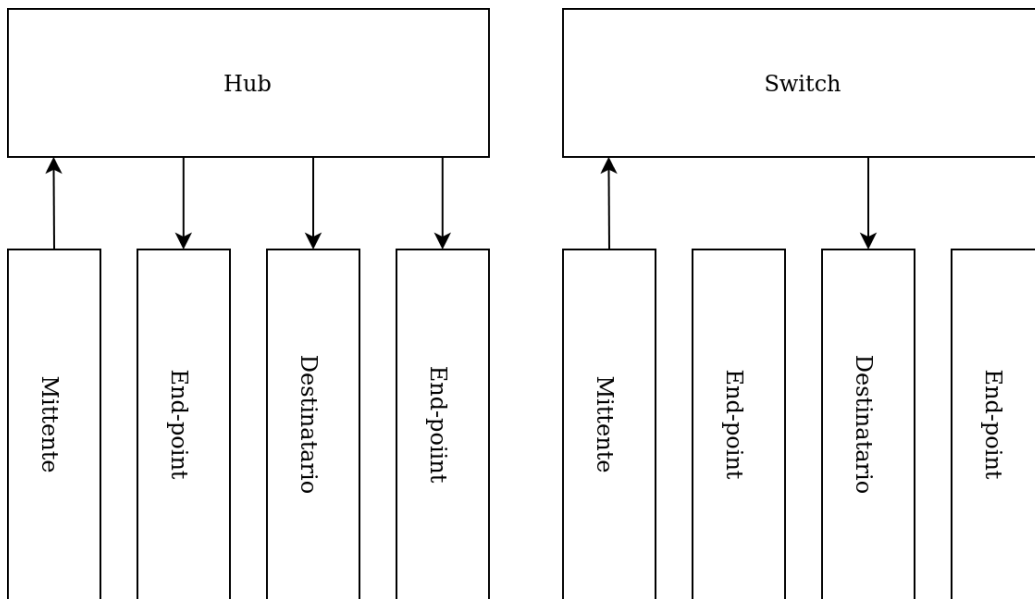


Figura 1.4: Confronto tra hub e switch[12]

- i frame vengono ricevuti dal RX che li accoda in una FIFO da cui la CPU li può leggere.

Confronto tra switch e hub Con il passare del tempo si è passati verso reti composte da connessioni punto-punto e full-duplex. Ogni connessione tra due dispositivi diventa quindi un dominio di collisione separato dagli altri[5].

La figura 1.4 mostra la differenza tra un utilizzo di Ethernet come bus e utilizzando uno switch per mediare la comunicazione tra i device. Un hub è un bus, a cui molteplici end-point possono essere collegati, ha quindi un unico dominio di collisione.

Lo switch, invece, riceve i pacchetti e li reindirizza verso il destinatario. Quando riceve un frame memorizza in una forward lookup table il MAC Address del mittente e lo associa alla porta di ricezione. Quando sarà necessario reindirizzare un frame verso quel MAC Address verrà utilizzata quella stessa porta.

In immagine 1.5 si illustra la struttura appena descritta di uno switch IEEE 802.3. L'unico tempo di accodamento aggiunto dallo switch è speso

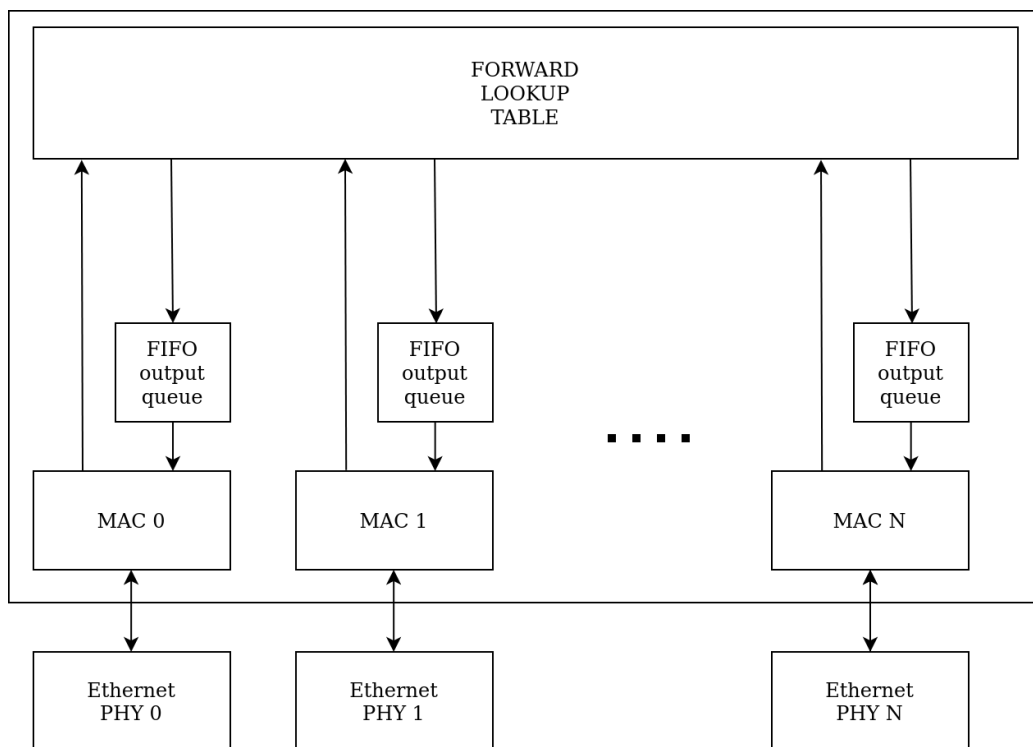


Figura 1.5: schema a blocchi di uno switch

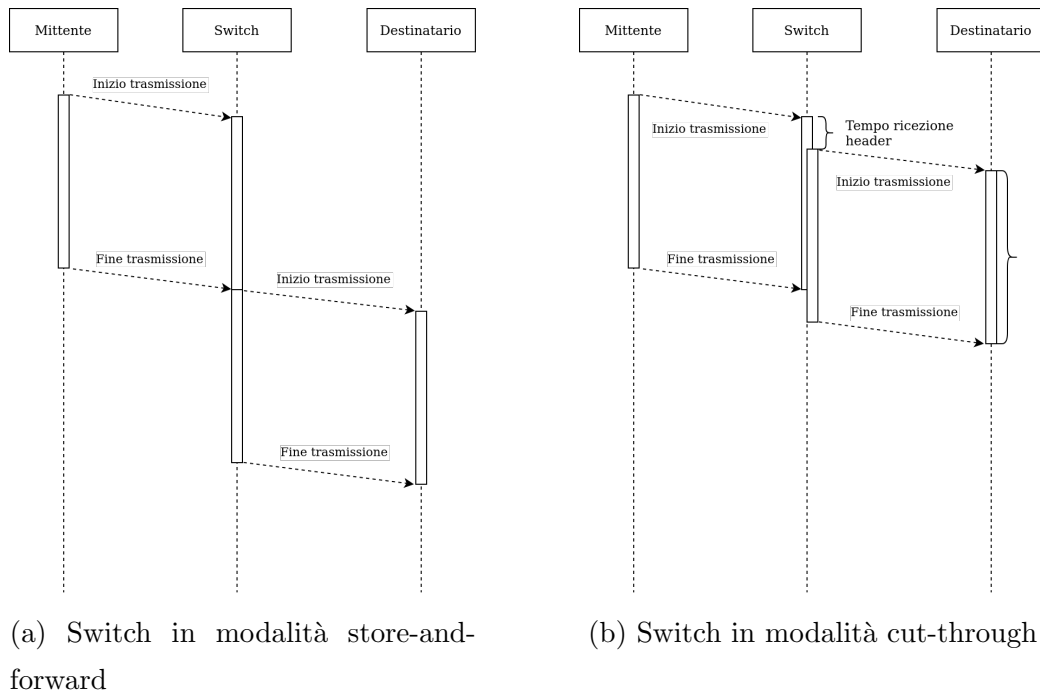


Figura 1.6: Differenti modalità operative di uno switch

nella coda di uscita associata al MAC.

Le collisioni non possono più avvenire, ma il frame può venir perso per via di una coda saturata all'interno di uno switch. Inoltre la latenza di comunicazione è fortemente dipendente dall'ordine di arrivo dei frame, dato che vengono ritrasmessi seguendo l'ordine in cui sono arrivati[17].

Cut-through e store-and-forward Negli switch i frame vengono reindirizzati in base al MAC Address di destinazione. La procedura standard è:

- ricevere l'header,
- ricevere i dati,
- ricevere il CRC e validare il frame,
- ritrasmettere il frame.

Questa modalità operativa viene chiamata store-and-forward. In uno switch[2] (ma anche in un router[6]) si può applicare una procedura diversa, chiamata cut-through, che prevede la ritrasmissione del frame (o del pacchetto) appena ricevuto il campo di destinazione. La figura 1.6 mostra la differenza tra le due procedure.

Traffic Shaper I pacchetti vengono persi quando avviene un congestionamento, che porta uno switch a riempire le code interne. I picchi di traffico inaspettato possono causare congestioni, e sono stati ricercati meccanismi per risolverli o prevenirli.

Numerose ricerche[33][30][11][20] nate con l'obiettivo di implementare tecniche di controllo del flusso di traffico hanno portato soluzioni operanti sopra il livello dell'Internet Protocol.

Un MAC che si ritrova sovraccarico di traffico, e quindi con i buffer interni occupati oltre una certa soglia, può richiedere a chi sta trasmettendo di aspettare prima di mandarne altri. Questo meccanismo di controllo di flusso è descritto nello standard IEEE 802.3x, ed è stato mantenuto all'interno degli standard TSN con l'IEEE 802.1Qbb - Priority Based Flow Control, che permette di controllare il flusso di traffico in base alla priorità IEEE 802.1P[19].

In alternativa i picchi di traffico possono essere prevenuti distribuendo il traffico su una finestra di tempo più ampia. I traffic shaper regolano l'invio di traffico in modo da mantenere un'equa distribuzione delle risorse di rete tra i diversi flussi di traffico. Si utilizzano algoritmi che limitano il consumo della banda utilizzata:

- alcuni di questi, come ad esempio il Credit Based Shaper e l'Asynchronous Traffic Shaper, funzionano dando un costo all'utilizzo della porta in uscita e limitando la banda utilizzabile[23].
- altri, come ad esempio il Time Aware Shaper, definiscono finestre di tempo associate a tipi di traffico, garantendo a ciascun tipo una banda minima.

Queste limitazioni vengono applicate al traffico in uscita su ciascuna porta di uno switch o di un end-point.

1.1.2 Bus di campo

Vengono illustrati i meccanismi utilizzati da altri protocolli per garantire un tempo di consegna deterministico.

Connessioni seriali Storicamente, in ambito industriale sono stati utilizzati protocolli seriali, half-duplex[9]. Nella maggior parte di questi protocolli la contesa per l'accesso al bus di comunicazione è controllato da un master. In questi si possono notare due approcci:

- master centralizzato,
- token ring, in cui i device si passano il token tra di loro.

Si noti che la presenza di un master è un tipo triviale di priorità binaria.

Nei bus di campo in cui il controllo è distribuito si è utilizzata una tecnica di accesso chiama Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). CSMA/CA richiede ai dispositivi che desiderano utilizzare il bus di annunciare la loro priorità prima di inviare il messaggio. Essendo il bus sincronizzato su un unico clock se più dispositivi si stanno annunciando faranno interferenza. In particolare, dato che le trasmissioni sono sincronizzate, faranno interferenza sul primo bit differente, e tutti i device con priorità minore possono interrompere la trasmissione per evitare la collisione.

CAN bus Il Controller Area Network (CAN) è un esempio di protocollo che fa uso di CSMA/CA ancora in uso ad oggi. L. Jianxin et al.[13] hanno analizzato il comportamento del bus CAN all'aumentare del carico. La loro ricerca mostra che il traffico ad alta priorità viene sempre trasmesso entro tre millisecondi, a prescindere dall'utilizzo del bus. Diversamente, il traffico a bassa priorità subisce ritardi sempre maggiori tanto più si satura il canale di comunicazione. Obiettivo della ricerca è proporre un meccanismo di Time

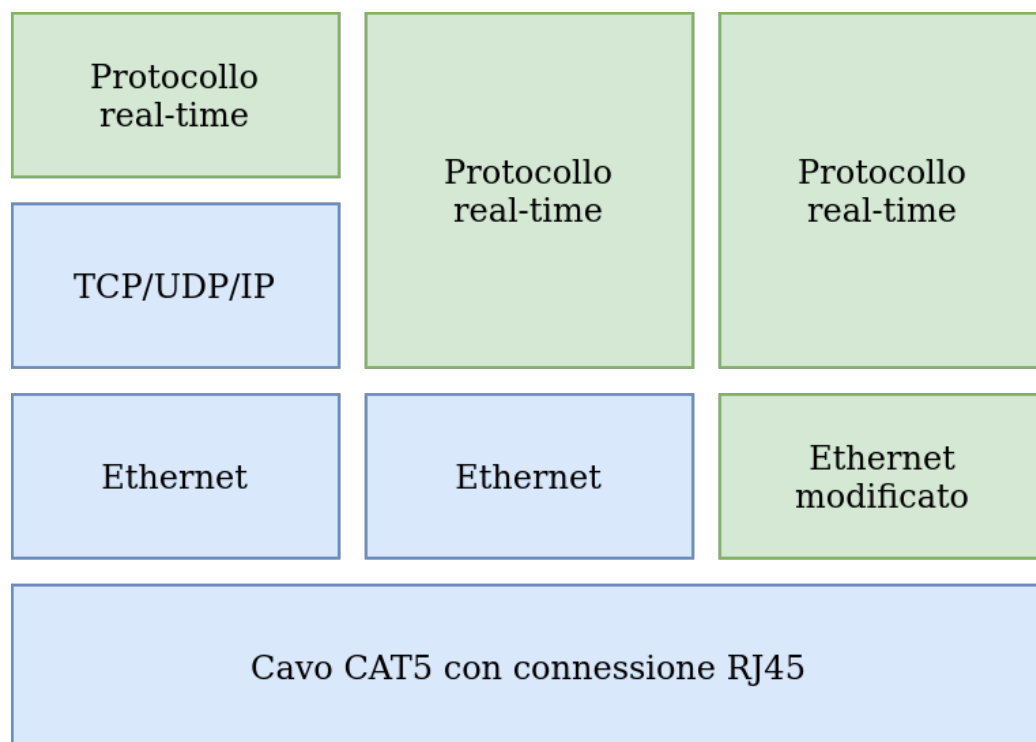


Figura 1.7: Tipologie di protocolli Ethernet real-time

Division Multiple Access (TDMA) e mostrano che anche il traffico a bassa priorità viene sempre trasmesso entro un determinato tempo.

1.1.3 Ethernet real-time

Ethernet offre numerosi vantaggi, maggiori prestazioni, hardware più comune e meno costoso, e permette di unificare il traffico real-time con traffico di diagnostica, riducendo i cablaggi necessari per un'installazione. Manca però di quelle caratteristiche necessarie a una comunicazione real-time. Osservando quanto descritto in[7] si può notare un interesse condiviso nell'ambito industriale nello sviluppare protocolli real-time basati su Ethernet.

Come mostrato in immagine 1.7, presa da[7], si può lavorare a diversi livelli, e sviluppare un protocollo che:

- utilizza TCP/UDP/IP,
- si interfaccia direttamente con Ethernet,
- apporta modifiche ad Ethernet.

Le maggiori prestazioni si ottengono modificando Ethernet.

TCP e UDP non sono real-time, e i protocolli che ne fanno uso applicano strategie per migliorare le prestazioni:

- EthernetIP utilizza le priorità VLAN, mostrate in tabella 1.1,
- CIPsync sincronizza i device utilizzando IEEE 1588, dopodiché permette di eseguire eventi pianificati,
- altri protocolli, MODBUS e Vnet/IP, non hanno nessun meccanismo per garantire determinismo,

e arrivano a garantire periodi di aggiornamento di 100 millisecondi, o al più 10 millisecondi quando richiedono di utilizzare Ethernet full-duplex.

I protocolli che si interfacciano con Ethernet (Ethernet Powerlink, TCnet, Ethernet for Plant Automation, Profinet CBA) fanno tutti uso di TDMA per garantire tempi di ciclo di 10 millisecondi. Il meccanismo di TDMA richiede una sincronizzazione tra i device per definire i momenti di inizio e fine delle finestre. Questa sincronizzazione può essere implementata in maniera distribuita o centralizzata, ma tutti questi protocolli richiedono di utilizzare solamente dispositivi compatibili.

Tra i protocolli che modificano Ethernet sono di interesse EtherCAT e Profinet IO. EtherCAT prevede una frame che attraversa tutti i device della rete, lungo un anello. Il frame viene generato da un master, e contiene una sezione per ciascuno slave, nella quale lo slave può leggere e scrivere le informazioni richieste. La comunicazione avviene in modalità cut-through, ovvero, la ritrasmissione del frame inizia appena viene ricevuto. Ne consegue che gli slave devono modificare il frame “on-the-fly”.

Profinet IO permette di configurare diverse classi di traffico. Ciascuna classe riesce a garantire deadline più o meno stringenti[24]. Per gestire le diverse classi definisce un modulo dedicato a interfacciarsi con Ethernet. Questo modulo si occupa di gestire una TDMA con finestre per il traffico real-time ciclico, real-time aciclico e non real-time. Inoltre assegna una priorità IEEE 802.1P ai frame in uscita. Profinet IO richiede di comporre la rete con soli device compatibili per garantire le deadline più stringenti.

Protocolli Ethernet real-time, come Ethercat o Sercos III, mirano ad utilizzare i vantaggi del cut-through. Si può notare una somiglianza tra il funzionamento di un bus e una rete di switch che reindirizzano frame in modalità cut-through, a patto che gli switch inseriscano ritardi di accodamento minimi. I ritardi di accodamento avvengono a causa di:

- le code FIFO contenute negli switch,
- un eventuale frame la cui trasmissione è già iniziata.

Le soluzioni proposte ai problemi appena elencati e si possono dividere in:

- permettere solo frame di traffico real-time all'interno della rete,
- ridurre la dimensione massima dei frame non real-time
- imporre una topologia in cui i device sono collegati in serie, e tutti i frame viaggiano nella stessa direzione.

1.2 Protocolli TSN

Nella sezione precedente si è mostrata l'evoluzione dei protocolli industriali, si è motivato l'interesse nell'utilizzo di Ethernet e si sono elencate le tecniche utilizzate per superare le mancanze del protocollo. I protocolli TSN utilizzano tecniche già utilizzate da bus di campo e protocolli Ethernet real-time per garantire limiti alla latenza.

In primis vengono definite diverse priorità del traffico. Nei protocolli half-duplex utilizzati nei bus di campo le priorità vengono utilizzate per gestire l'accesso multiplo alla comunicazione. Nei protocolli Ethernet real-time (ad esempio EtherNet-IP) viene utilizzata la priorità definita dallo standard IEEE 802.1P. Lo standard IEEE 802.1Q prevede un frame Ethernet esteso con un campo dedicato a indicarne la priorità.

Numerosi protocolli Ethernet real-time sviluppati prima degli standard TSN fanno utilizzo di TDMA. Negli standard TSN è presente l'IEEE 802.1Qbv che permette di implementare TDMA in un traffic shaper a livello di interfaccia di rete.

Alcuni protocolli Ethernet real-time cercano di sfruttare i vantaggi della modalità cut-through negli switch. La modalità cut-through, illustrata in sezione 1.1.1, permette di iniziare la ritrasmissione di un frame prima di averlo ricevuto completamente. Se però è già in transito un altro frame è necessario attendere che venga ultimata la trasmissione. Negli standard TSN è presente l'IEEE 802.1Qbu, che prevede l'interruzione di un frame in trasmissione.

PTP (IEEE 1588) è un protocollo comune per sincronizzare dispositivi, anche tra i protocolli Ethernet real-time, lo standard IEEE 802.1AS definisce un meccanismo di sincronizzazione adeguato ai requisiti industriali[31].

Bisogna però notare che gli standard TSN non mirano a sostituirsi completamente ai protocolli Ethernet real-time attualmente utilizzati in industria. Si prenda in considerazione Profinet.

Profinet non è solamente un standard di comunicazione. Descrive la comunicazione tra sensori e attuatori, le azioni che possono venire effettuate su ciascun device, e altri aspetti di livello principalmente applicativo. La parte di Profinet che descrive la comunicazione è indipendente dal resto. Sono state definite diverse classi di conformità (da classe A a classe D) con prestazioni via via crescenti. La classe di conformità D, in particolare, prevede l'utilizzo degli standard TSN descritti.

Preambolo							SFD	MAC Address del destinatario						MAC Address del mittente						Ethernet Type		Contenuto				CRC							
1	2	3	4	5	6	7	1	1	2	3	4	5	6	1	2	3	4	5	6	1	2	1	...	46-1500	1	2	3	4					
Preambolo							SFD	MAC Address del destinatario						MAC Address del mittente						802.1Q Header				Ethernet Type		Contenuto				CRC			
1	2	3	4	5	6	7	1	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	1	2	1	...	46-1500	1	2	3	4	

Figura 1.8: frame IEEE 802.1Q

Il frame IEEE 802.1Q Viene definito un header alternativo al classico 802.3, con un campo aggiuntivo, utilizzato per indicare VLAN e priorità. Su questo frame si basano i protocolli TSN, che usano spesso il campo priorità per dividere il traffico in diverse classi.

Come è possibile vedere in immagine 1.8 nel frame 802.1Q sono presenti quattro ottetti extra nell'header. I primi due ottetti vanno a sovrapporsi all'Ethernet type, e hanno un valore predefinito. È così possibile riconoscere se il resto dell'header segue il formato IEEE 802.3 o IEEE 802.1Q osservando il valore dell'Ethernet type. Nei due ottetti successivi sono contenuti i campi PCP, per indicare la priorità, e il campo VID per indicare la VLAN di appartenenza del frame, come mostrato in tabella 1.1.

Lo switch che riceve un frame IEEE 802.1Q deve prevedere molteplici code di uscita in ciascun MAC. In base al valore del campo PCP il frame viene inserito in una determinata coda. In tabella sono mostrati gli assegnamenti tra code e priorità quando sono disponibili 8 code. Se le code disponibili sono meno, classi diverse vengono raggruppate sulla stessa coda[29].

1.2.1 IEEE 802.1AS

Lo standard IEEE 802.1AS è lo standard TSN che descrive la sincronizzazione tra device all'interno di una stessa rete[10]. Il protocollo è anche chiamato Generalized Precision Time protocol (gPTP), ed è un profilo del IEEE 1588 Precision Time Protocol (PTP).

Ciascuna porta full-duplex periodicamente inizia una peer-delay-measurement, illustrata in figura 1.9. Assumendo che il ritardo di comunicazione è simmetrico tra le due direzioni, può essere calcolato[10].

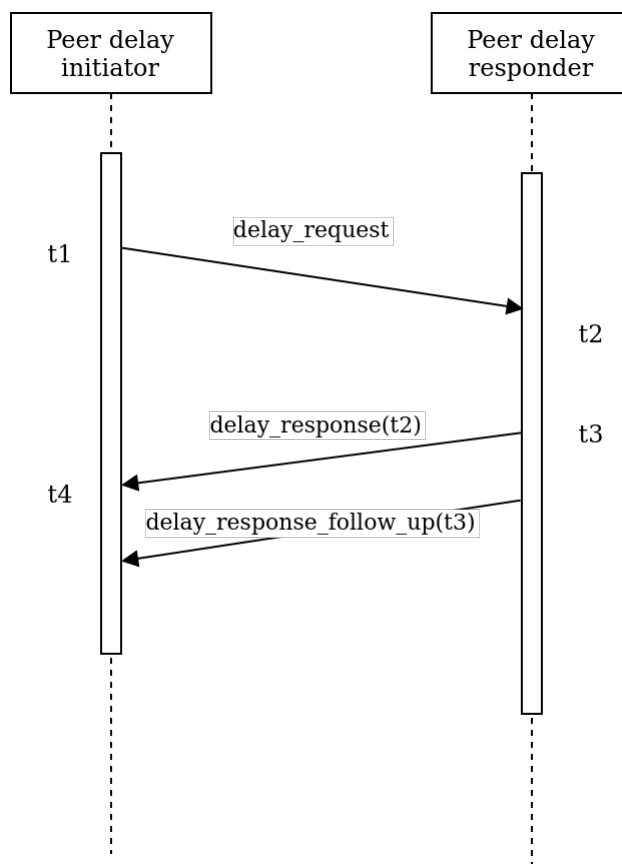


Figura 1.9: Processo di peer-delay-measurement

Campo	Lunghezza (bit)	funzione
TPID	16	Si trova nella stessa posizione originale dell'ethertype. È utilizzato per riconoscere un frame 802.1Q
PCP	3	priorità
DEI	1	indica se il frame è scartabile in caso di congestione
VID	12	indica a quale VLAN appartiene il frame

Viene eletta un'istanza Grandmaster all'interno della rete. Il Grandmaster sarà la radice di un albero, come mostrato in figura 1.10. Dopodiché il Grandmaster invia periodicamente una sincronizzazione ai device connessi, indicando il tempo attuale. Ciascun device corregge con il ritardo di comunicazione calcolato precedentemente e ritrasmette il tempo corretto ai device a se connessi[10].

A differenza di PTP, IEEE 802.1AS assume che tutte le comunicazioni tra istanze PTP avvengono solo al livello MAC, mentre IEEE Std 1588 supporta anche comunicazioni a livelli superiori.

1.2.2 IEEE 802.1Qbv

Lo standard IEEE 802.1Qbv permette di definire un meccanismo di Time Division Multiple Access (TDMA). Nello standard vengono riprese le classi di priorità già definite nel IEEE 802.1P (elencate in tabella 1.1) e viene descritto uno shaper del traffico.

Vengono configurate delle finestre temporali, con associate delle priorità, e durante ciascuna finestra lo shaper consente l'uscita del traffico di quelle specifiche priorità. Lo shaper utilizza un meccanismo basato su porte (o

Valore campo PCP	Priorità	Tipo di traffico	Descrizione
0	1	Best-Effort	Valore di default per frame senza campo PCP
1	0	Background	Attività che non influenzano l'utilizzo della rete
2	2	Spare	
3	3	Excellent Effort	Traffico Best-effort prioritario
4	4	Controlled Load	Traffico con banda minima garantita
5	5	Video	Traffico con delay minore di 100 mS
6	6	Voice	Traffico con delay minore di 10 mS
7	7	Network Control	Traffico con consegna garantita

Tabella 1.1: Priorità IEEE 802.1P[29]

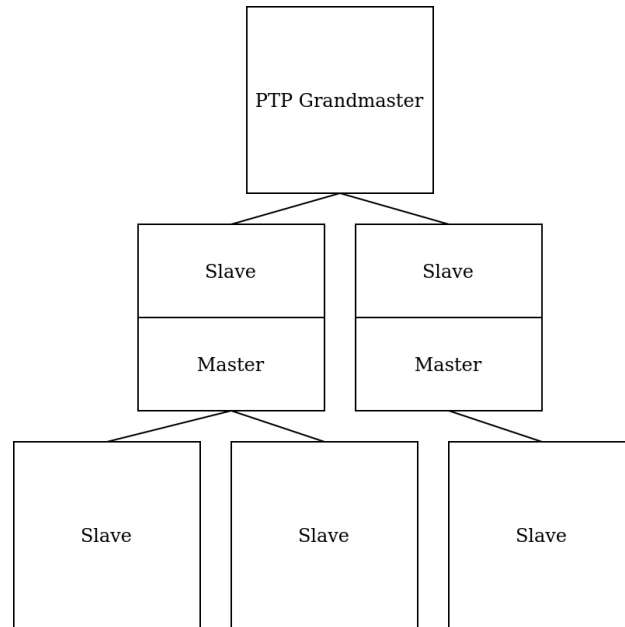


Figura 1.10: Gerarchia Grandmaster PTP

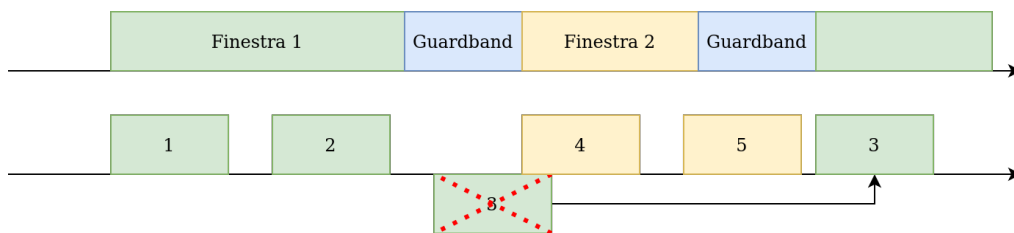


Figura 1.11: Finestre Time Aware Shaper, con guardband

cancelli, dall'inglese gate) che può aprire o chiudere. Per ciascuna finestra viene definita un Gate Contro Entry (GCE), e l'insieme di tutte le GCE forma la Gate Control List (GCL).

La trasmissione di un frame avviene se:

- una coda il cui cancello è aperto contiene almeno un frame (nel caso siano disponibili frame da più code ne viene selezionato uno dalla coda con priorità maggiore),
- la trasmissione del frame può essere completata prima che il cancello si chiuda

Per garantire di poter completare la trasmissione del frame viene definita una guardband, la cui durata corrisponde al tempo necessario all'invio del frame di dimensione massima. I gate vengono chiusi in anticipo all'inizio della guardband[14]. Le finestre possono essere coordinate tra i diversi dispositivi nella rete utilizzando protocolli di sincronizzazione come per esempio PTP.

La figura 1.11 mostra un esempio di funzionamento del TAS:

- vengono definiti due finestre che si alternano, ciascuno associato a diverse priorità di traffico,
- durante ciascuna finestra è possibile inviare solamente il traffico delle priorità specificate (frame numero 1, 2, 4 e 5),
- la trasmissione del frame 3 viene impedita dalla guardband.

Siccome lo shaper lavora in base alla priorità del frame inviato non è possibile gestire le finestre in funzione dei singoli flussi di traffico. È necessario fare utilizzo di standard aggiuntivi, come l'IEEE 802.1CB (Frame Replication and Elimination for Reliability)[4].

La figura 1.12 illustra un componente MAC con i componenti necessari per essere compatibile con IEEE 802.1Qbv. In particolare si noti la presenza del Time Aware Shaper, con i gates gestiti in funzione del clock.

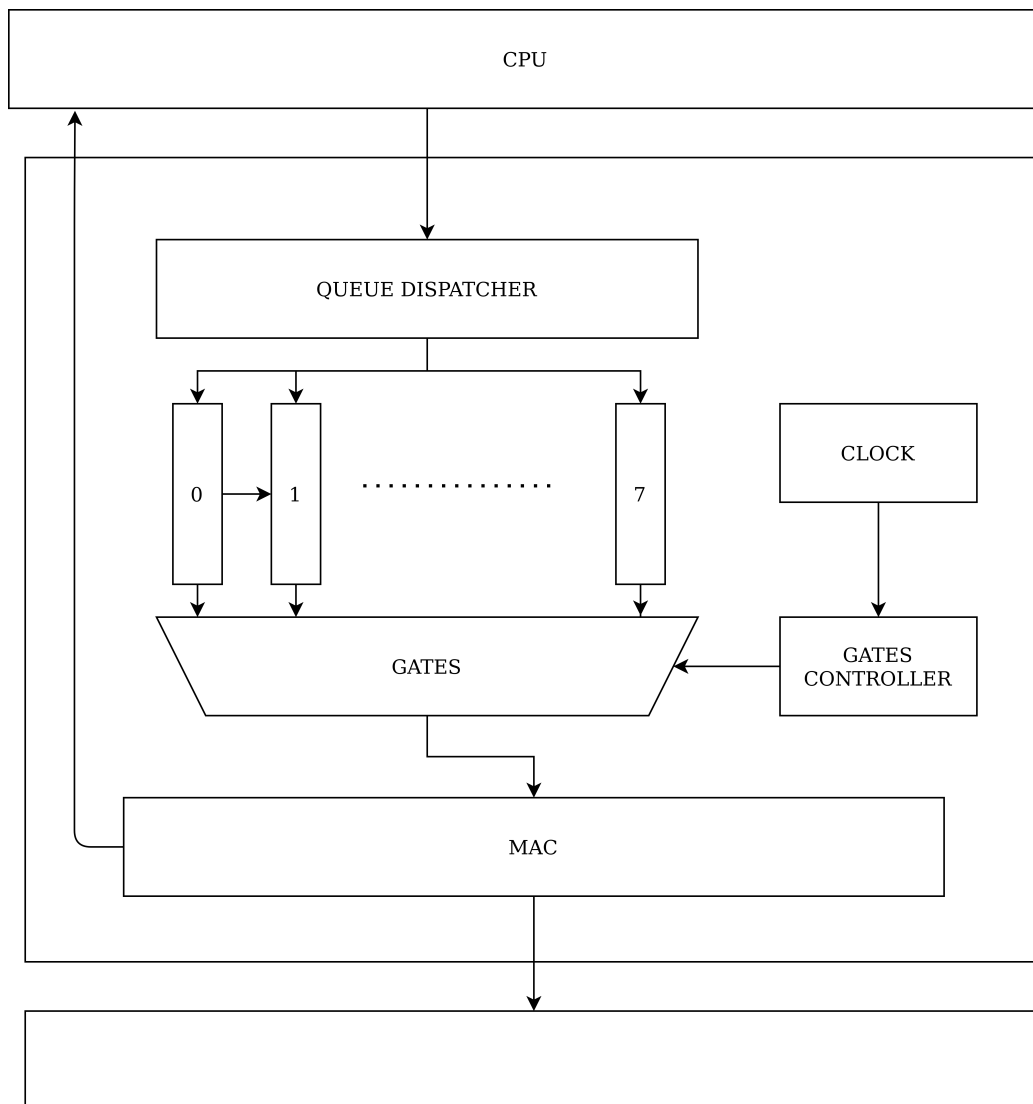


Figura 1.12: schema di un device TSN compatibile con TAS

Frame standard

Preambolo							SFD	MAC destinatario						MAC mittente						Ethernet Type		Contenuto			CRC			
1	2	3	4	5	6	7	1	1	2	3	4	5	6	1	2	3	4	5	6	1	2	1	...	46-1500	1	2	3	4

Primo frammento

Preambolo							SMD-Sx	MAC destinatario						MAC mittente						Ethernet Type		Contenuto			mCRC			
1	2	3	4	5	6	7	1	1	2	3	4	5	6	1	2	3	4	5	6	1	2	1	...	46-1500	1	2	3	4

Frammenti intermedi

Preambolo						SMD-Cx	Contatore frammenti						Contenuto			mCRC			
1	2	3	4	5	6	1	1						1	...	46-1500	1	2	3	4

Ultimo frammento

Preambolo						SMD-Cx	Contatore frammenti						Contenuto			CRC			
1	2	3	4	5	6	1	1						1	...	46-1500	1	2	3	4

Figura 1.13: Divisione di un frame in frammenti

1.2.3 IEEE 802.1Qbu

La Frame Preemption (FPE) è la possibilità di interrompere un frame a durante l'invio, quando è necessario doverne mandare uno a priorità maggiore. Viene inizialmente introdotta tra gli standard TSN per mitigare gli effetti negativi della guard-band del TAS[16].

Si definiscono due tipi di frame Ethernet: Express e Preemptable (interrompibile). I frame interrompibili sono anche chiamati mPacket. Vengono apportate modifiche al frame Ethernet per renderlo interrompibile.

In particolare si noti che:

- un frame interrompibile ha uno SFD differente, chiamato SMD(Start mPacket Delimiter),
- un frame interrompibile può venire terminato da un CRC parziale, utilizzato per indicare che è stato interrotto,
- in ciascun frammento, tranne l'ultimo, il CRC è sostituito da un mCRC, relativo al singolo frammento e avente hanno gli ultimi 16 bit invertiti. Questo permette di riconoscere che il frame è in realtà un frammento.
- i frammenti di frame successivi al primo hanno un preambolo diverso.

Un frammento ha una dimensione minima di 64 ottetti, da ciò ne consegue che è interrompibile in un determinato momento se sono mantenute le seguenti condizioni:

- sono stati inviati almeno 64 ottetti,
- rimangono almeno altri 64 ottetti da inviare.

Ne consegue che bisogna considerare i frammenti aventi una dimensione minima nel caso peggiore di 127 ottetti[1].

Dato che un frame interrompibile ha uno SFD differente, ne consegue che un componente MAC non compatibile con la FPE non è in grado di riconoscere frame interrompibili.

Si deve aggiungere inoltre che l'interruzione dei frame è localizzata al collegamento tra due nodi. Il ricevente attende che venga completata la trasmissione del frame interrotto prima di ritrasmetterlo.

Interazione con il TAS La FPE ha tra i suoi obiettivi iniziali migliorare le prestazioni del TAS. Il TAS, ricordiamo, definisce una lista di GCE, ciascuna con un intervallo di tempo associato. Al termine di ciascun intervallo è necessario cambiare lo stato dei gate, normalmente viene eseguita l'operazione Set-Gate-States, che apre e chiude i gate in funzione della successiva GCE. Con la FPE vengono definite due nuove operazioni possibili: Set-And-Hold-MAC e Set-And-Release-MAC. Queste operazioni estendono Set-Gate-States, specificando come deve comportarsi il componente MAC nei confronti dei frame interrompibili. In particolare Set-And-Release-MAC indica che durante la seguente CGE è possibile inviare frame interrompibili, viceversa Set-And-Hold-MAC indica che non è possibile.

Nell'immagine 1.14 è possibile vedere la struttura indicata nell'IEEE 802.3br. Si utilizzano due componenti MAC, dedicati ai due diversi tipi di traffico, che prendono il nome di eMAC (per il traffico Express) e pMAC (per il traffico Preemptable) I due MAC sono coordinati da un MAC Merge Sublayer (MMS) che seleziona quale traffico, sia in entrata che in uscita, utilizzare.

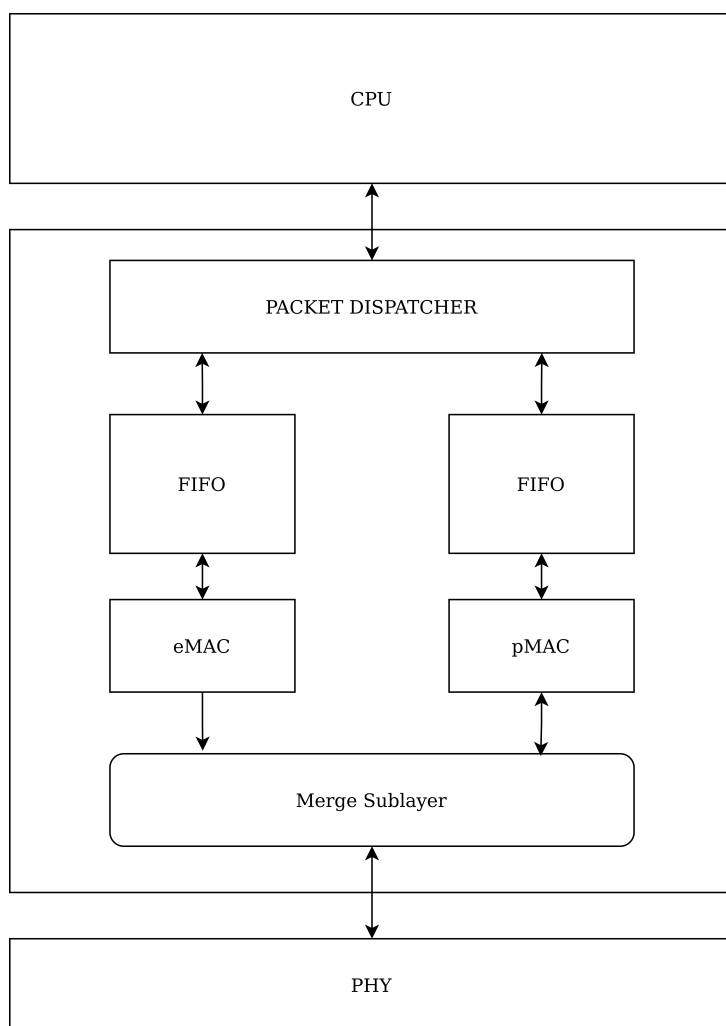


Figura 1.14: schema di un device TSN compatibile con FPE

Capitolo 2

Analisi progettuale

Nella sezione precedente è stato illustrato il caso d'uso per reti Ethernet real-time. Sono state illustrate le modifiche previste al protocollo IEEE 802.3 all'interno dei protocolli della famiglia TSN, che mirano a costruire una comunicazione deterministica.

Successivamente vengono presentati gli aspetti a cui prestare attenzione quando si testa una rete TSN: topologia e misura della latenza di rete.

2.1 Ambiente di test

Verranno discussi ora i test proposti per la validazione dell'implementazione di uno switch TSN. I test saranno condotti utilizzando un certo numero di device, tra switch ed end-point, connessi tra loro seguendo una topologia che verrà illustrata in questa sezione.

2.1.1 Topologia di test

Come indicato da[7] le topologie utilizzate in protocolli Ethernet real-time possono essere:

- albero,
- in serie, che può eventualmente essere chiusa diventare ad anello

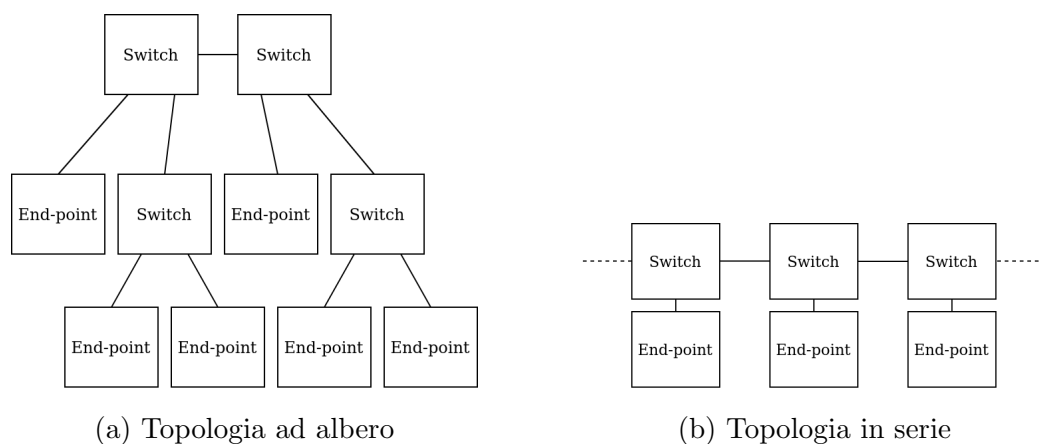


Figura 2.1: Topologie tipicamente utilizzate in ambito industriale

Le topologie sono mostrate rispettivamente in figura 2.1a e 2.1b. Nella topologia in serie end-point e switch possono essere uniti in un unico dispositivo.

Osservando entrambi i casi si è notato un elemento base comune: la connessione tra due switch, i quali trasportano traffico verso diversi end-point.

Si è osservato in precedenza (capitolo 1.1.3), che alcuni protocolli Ethernet real-time cercano di utilizzare gli switch in modalità cut-through per raggiungere prestazioni più elevate. Devono però evitare i ritardi di accodamento inevitabili che avvengono se la trasmissione di un frame di minore priorità è già iniziata. L'utilizzo di una topologia in serie, come quella mostrata in figura 2.1b, è una possibile soluzione utilizzata. La FPE prevede un meccanismo utilizzabile per ridurre questo ritardo di accodamento. E le prestazioni di questa soluzione sono verificabili sulla singola comunicazione tra due switch TSN. È necessario collegare due end-point al primo switch, per rendere necessario l'utilizzo della FPE.

Il TAS, come ogni traffic shaper, viene applicato sul traffico in uscita da un MAC. La singola connessione tra due switch è quindi sufficiente per verificarne le funzionalità.

Si propone quindi la topologia mostrata in figura 2.2, che permette di effettuare tutti i test senza utilizzare troppe risorse, in termini di quantità di

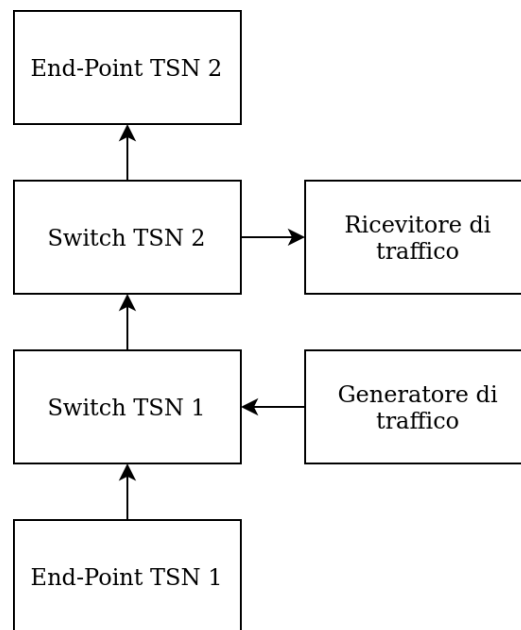


Figura 2.2: Topologia di test

device necessari. Il traffico sarà trasmesso seguendo le direzioni indicate in figura, si noti il che lo switch TSN 1 ha i due tipi di traffico in ingresso. Lo switch TSN 2 agisce solamente come divisore dei flussi di traffico, evitando così di sovraccaricare l'interfaccia di un unico end-point ricevente.

Si pone ora il problema di misurare le prestazioni della comunicazione. Nella sezione successiva è stata descritta l'architettura standard utilizzata per organizzare questi componenti.

2.2 Misura delle prestazioni della rete

Si considera ora il problema tecnico della misurazione delle prestazioni della rete. In generale le prestazioni sono misurabili sotto diversi aspetti:

- packet-loss (perdita di frame)
- tempo di consegna
- banda disponibile

Di particolare interesse per le reti real-time sono i primi due parametri.

Packet-loss Un frame può venire perso a causa di un'interferenza o di una congestione. L'interferenza viene già considerata dal campo CRC del frame Ethernet, che permette di correggere un errore nel caso avvenga[3]. La congestione, invece, è dovuta ad accumulo di frame nella coda di un device, fino a saturarla.

Per misurare la packet-loss viene inserito a livello applicativo un packet counter nel contenuto del frame. Il ricevente si aspetta di ricevere frame contenenti packet counter consecutivi in ordine crescente. Data la topologia utilizzata nei test non si possono ricevere frame in ordine invertito. Sarebbe possibile se, ad esempio, si utilizzasse una topologia ad anello e protocolli che inviano frame ridondanti come l'IEEE 802.1QCB Frame Replication and Elimination for Reliability.

Tempo di latenza La latenza di consegna è il tempo necessario ad attraversare l'intera rete presente tra mittente e destinatario. Siccome si stanno considerando reti sensibili al tempo la latenza diventa la caratteristica principale, a discapito della capacità di utilizzo della rete.

In figura 2.3 Il tempo di consegna, la latenza, è divisibile in più intervalli:

- ritardo di trasmissione, tempo necessario a serializzare l'intero frame, dipende dalla velocità di trasmissione
- ritardo di accodamento, tempo speso all'interno dello switch, diviso in tempo necessario a processare il frame e tempo speso nelle code di attesa
- ritardo di propagazione, tempo necessario al segnale elettrico di propagarsi

Di particolare interesse è il ritardo di accodamento, dato che gli standard TSN mirano ad abbassare il suo limite massimo. Il ritardo di propagazione dipende dalla velocità di propagazione del segnale elettrico sul cavo. Lo

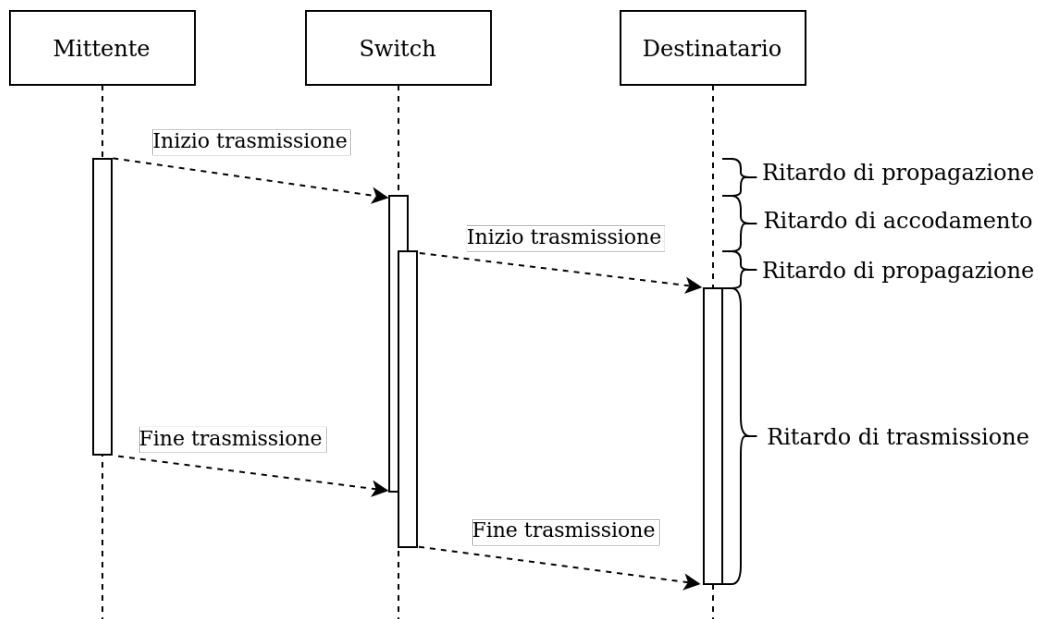


Figura 2.3: Ritardi di comunicazione in una rete, con lo switch utilizzato in modalità cut-through

standard IEEE 802.3-2018 richiede l'utilizzo di cavi con velocità di propagazione di almeno $0.67c$ [18]. Ricerca esistente[25] mostra che un cavo CAT5-e ha ritardi di propagazione inferiori a 400 nanosecondi per 80 metri di cavo. Considerando che nell'ambiente di test costruito sono stati utilizzati pochi metri di cavo il ritardo di propagazione è dell'ordine delle decine di nano secondi, e quindi trascurabile rispetto al ritardo totale, che si mostrerà essere dell'ordine delle unità di microsecondi. In installazioni complesse, con chilometri di cavo, il ritardo arriva all'ordine dei microsecondi, andando a equipararsi, come ordine di grandezza, con il ritardo di accodamento. In questo lavoro di tesi si vuole concentrare tutta l'attenzione sul livello data-link, e quindi si sono utilizzati cavi corti e si trascura il ritardo di propagazione.

2.2.1 Misura della latenza

L'immagine 2.4 mostra i componenti attraversati dal frame. La latenza di comunicazione può essere misurata come intervallo tra due eventi, e durante

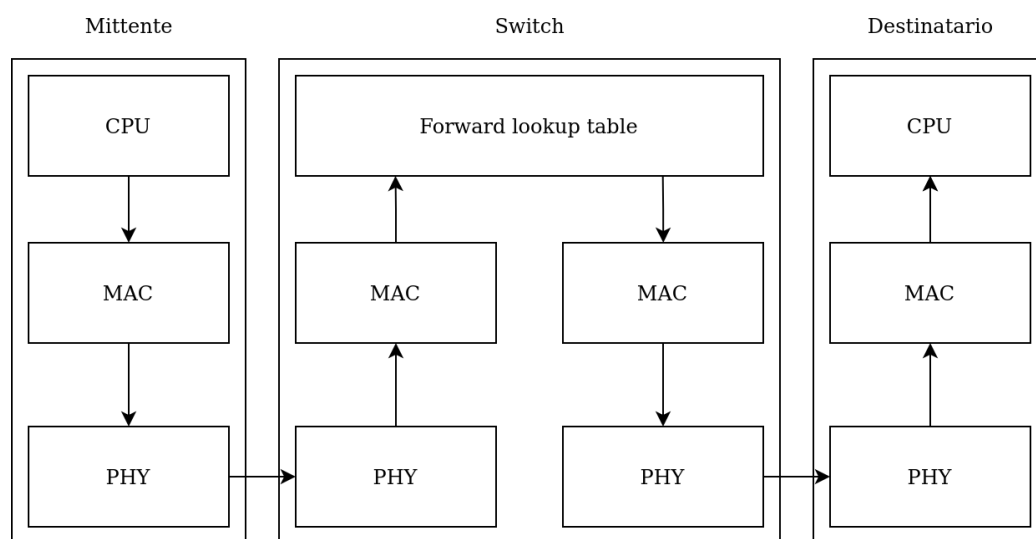


Figura 2.4: Rappresentazione schematica del passaggio di un frame

l'analisi bisogna tenere conto di tutti i componenti che hanno aggiunto ritardo tra i due eventi in questione.

Gli switch si suppone siano sviluppati in hardware, eventualmente programmabile. Hanno molteplici componenti MAC che ricevono e trasmettono frame, introducendo ritardi per via delle code in uscita, e una forward lookup table che reindirizza i frame sulla porta (o sulle porte) corretta.

Gli end-point possono essere sviluppati sia su hardware programmabili che su processori collegati a una periferica MAC TSN. Ad esempio, durante i test descritti in questa tesi sono stati utilizzati dei processori Intel Cyclone-V, e sono composti da un cortex-A9 affiancato da una parte di hardware programmabile, in cui viene implementato il componente MAC TSN compatibile. Siccome si fa uso di un processore, che esegue istruzioni in sequenza, è necessario tenere conto dei ritardi che ne conseguono.

All'interno di ogni end-point si ha un applicativo che richiede l'invio del messaggio, dopodiché si hanno i seguenti eventi

- inizio comunicazione tra CPU e periferica MAC,
- inizio trasmissione del frame verso il PHY,

- completata trasmissione verso il PHY,

All'interno di ciascuno switch si possono misurare i seguenti eventi:

- inizio ricezione frame, tra PHY e MAC,
- completata ricezione dell'header,
- inizio accodamento nella coda del MAC di destinazione,
- inizio trasmissione,
- fine trasmissione

Nell'end-point ricevente si hanno gli stessi eventi descritti per l'end-point che trasmette, ma notificare alla CPU il frame ricevuto può richiedere un tempo variabile.

Da questi eventi è possibile misurare il tempo di latenza tra due end-point, misurando il tempo tra l'evento di inizio trasmissione del frame dal MAC verso il PHY sul mittente, e l'evento di inizio ricezione da PHY a MAC sul ricevente. È possibile anche misurare il tempo speso all'interno di un singolo switch, ad esempio misurando il tempo tra l'inizio ricezione da PHY a MAC su una porta e l'inizio trasmissione da MAC a PHY su un'altra porta dello stesso switch.

Per avere la massima precisione di misura è opportuno

Rilevamento degli eventi Si deve quindi misurare l'intervallo di tempo che trascorre tra due degli eventi precedentemente descritti. È necessario avere un riferimento a un clock di tempo unico su cui collocare gli eventi:

- si può utilizzare un singolo dispositivo a cui notificare tutti gli eventi. Questo dispositivo può essere uno dei dispositivi della rete, o uno esterno,
- si possono sincronizzare gli end-point utilizzando PTP.

Per gli eventi che riguardano la CPU è possibile eseguire azioni immediatamente dopo un evento per registrare il momento in cui avviene. Nel caso è opportuno tenere conto del tempo necessario alla CPU reagire all'evento appena accaduto ed eseguire le operazioni indicate.

Per gli eventi che riguardano il MAC si può, ipotizzando che il componente MAC sia implementato su hardware programmabile, aggiungere un modulo extra che comunichi verso l'esterno l'avvenimento dell'evento.

D. Pedretti et al.[25] mostrano che è possibile utilizzare PTP per sincronizzare due dispositivi con la precisione del nanosecondo. Per riferimento si consideri che un modulo MAC Gigabit utilizza un clock a 125MHz[15], da cui consegue che l'intero ciclo di clock dura 8 nanosecondi.

Hyung-Taek et al.[21] mostrano invece che le performance di gPTP non sono influenzate da reti sovraccaricate, in quanto è sufficiente effettuare la procedura di sincronizzazione con una frequenza adeguata.

Utilizzare gPTP per sincronizzare i device permette di utilizzare dei timestamp come misura del tempo. I timestamp danno il vantaggio di poter misurare il passare del tempo utilizzando risorse locali: ciascun device prende come riferimento il proprio timestamp quando rileva l'evento. Lato trasmissione si inserisce il timestamp nel frame inviato, lato ricezione si calcola la differenza tra il tempo ricevuto e il tempo attuale. Dopo aver calcolato la latenza di comunicazione il dato può essere comunicato e memorizzato senza requisiti stringenti di tempo.

È, in alternativa, possibile utilizzare il fronte di salita di un contatto. Ha il vantaggio di essere una soluzione più semplice da implementare. Ma ha lo svantaggio di richiedere un contatto per ogni diverso evento che si vuole misurare. Tutti i contatti devono essere collegati a un unico device, al cui clock faranno riferimento tutte le misure. Questo device può anche essere un device della rete.

Riassumendo, l'analisi proposta mostra le possibili tecniche di misura (anche mostrate in immagine 2.5):

- utilizzare due device, raccogliendo misure su entrambi, sincronizzati

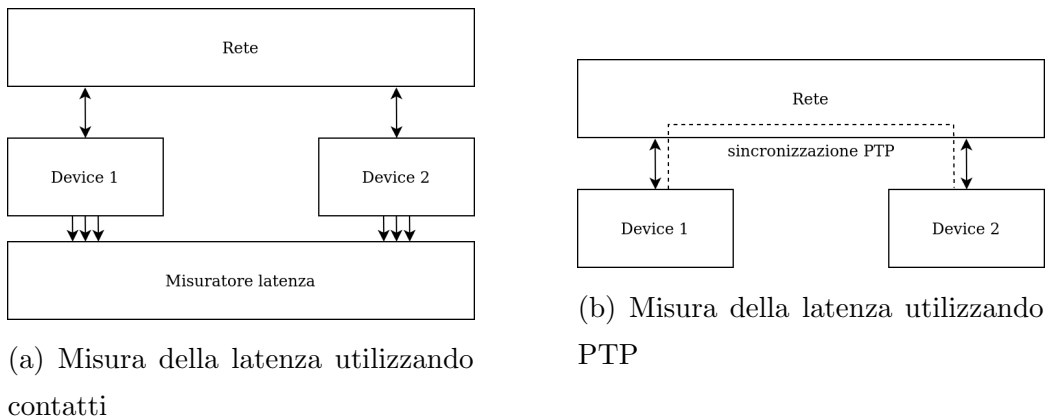


Figura 2.5: Configurazioni per la misura della latenza

tramite PTP, immagine 2.5b

- utilizzare tre device, di cui uno

La memorizzazione può essere gestita diversamente, è sufficiente comunicare le misurazioni a un device capace di memorizzarle.

2.3 Utilizzo dell'ambiente di test

Nelle sezioni precedenti sono stati descritti gli aspetti di un ambiente di test per reti TSN. È stata proposta una topologia che permette di verificare il funzionamento degli standard IEEE 802.1P, 802.1Qbu, 802.1Qbu e di verificare se il 802.1AS influisce sulle prestazioni della rete.

La packet-loss è misurabile a livello applicativo, inserendo un packet-counter nel frame che viene inviato. Sia l'inserimento del packet counter, lato mittente, che la verifica, lato ricevente può avvenire a livello applicativo, in quanto non aggiunge ritardi di trasmissione.

La latenza può essere misurata sia utilizzando PTP che utilizzando dei contatti. In ogni caso è necessario sviluppare un modulo hardware per avere la massima precisione possibile.

Utilizzando la topologia descritta in 2.1.1, la procedura di test è la seguente.

- Utilizzare il “generatore di traffico” per produrre traffico di background. Utilizzando tutta la banda disponibile in uscita dal generatore.
- Utilizzare il “End-point TSN 1” per generare il traffico di test. Si vuole misurare il ritardo causato dall’interazione tra il traffico di test e di background. È quindi opportuno generare i frame di test con una frequenza sufficientemente bassa, che consenta allo switch di ritrasmettere un frame prima dell’arrivo del successivo, in modo da non creare accodamenti tra di essi.
- Controllare sul “ricevitore di traffico” che entrambi gli switch sono configurati correttamente e che quindi solo il traffico di background arriva a quel nodo.
- Controllare sul “End-point TSN 2” che solo i frame di test arrivano a quel nodo.
- Iniziare a raccogliere dati su latenza e packet-loss. È opportuno rendere indipendente la trasmissione del traffico dalla raccolta dei dati di test.

Test latenza di base Si vuole misurare la latenza di rete, ovvero il ritardo aggiunto dagli switch, per ricevere, accodare, e ritrasmettere i frame di test.

Gli eventi rilevati su trasmittente e ricevente, possono avere dei ritardi compresi tra gli eventi rilevati. Per calcolare questo ritardo di base si collegano gli end-point direttamente tra di loro e si misura la latenza di comunicazione.

Test 802.3 Per poter comparare i miglioramenti apportati dagli standard TSN si confronta con una situazione base. Non è necessaria nessuna configurazione per questo test, è sufficiente avere switch ed end-point funzionanti.

Siccome il traffico di background satura il canale di comunicazione, si avrà della packet-loss. Ci si aspetta che i frame vengano persi in egual misura tra traffico di background e traffico di test.

La latenza, per i frame di test che riescono a passare, dipende dall'implementazione degli switch. I frame vengono accodati nella stessa coda, ed è un frame di test viene trasmesso dopo tutti gli altri frame in coda.

Test 802.1P Per poter testare le priorità è necessario configurare l'endpoint trasmittente ad inviare traffico a priorità maggiore del traffico di background. Ed è necessario configurare gli switch in modo da utilizzare due code diverse per le due diverse priorità di traffico.

Ci si aspetta di non avere packet-loss. I frame di test hanno una coda a loro dedicata, prioritaria rispetto alla coda del traffico di background.

La latenza di comunicazione può variare in base allo stato del MAC che trasmette verso lo switch 2:

- quando arriva il frame di test il MAC sta trasmettendo un frame di background,
- nel caso migliore viene completata la trasmissione immediatamente dopo l'arrivo, e il frame di test può venire ritrasmesso immediatamente,
- nel caso peggiore la trasmissione è appena iniziata, ed è necessario attendere che venga inviato l'intero frame di background prima di inviare il frame di test.

Sapendo la dimensione del frame di background e la velocità del canale, è possibile calcolare la latenza massima teorica.

Test 802.1Qbu Per poter testare la FPE è necessario configurare lo switch 1 a inviare traffico IEEE 802.1Qbu sulla porta in uscita. È necessario configurare il traffico di test come express e il traffico di background come preemptable. A seconda delle configurazioni offerte dallo switch può essere necessario dover configurare anche le porte in ingresso, e i generatori di traffico.

Ci si aspetta di non avere packet loss, come nel caso del IEEE 802.1P si hanno code separate per i due tipi di traffico.

La latenza dipende ancora dallo stato di trasmissione del frame di background. Con la FPE è possibile interrompere la trasmissione in corso per iniziarne una con maggiore priorità. Si ha però una dimensione minima del frammento, sia attuale, che successivo, di 64 byte. Nel caso peggiore, in cui si devono inviare altri 64 byte per completare il frammento corrente, e poi altri 63 perché non sufficienti per creare il frammento successivo, si inviano un totale di 127 byte. Sapendo la velocità del canale, è possibile calcolare la latenza massima teorica.

Test 802.1Qbv Per poter testare il TAS è necessario configurare due finestre, per le due diverse priorità di traffico. Ciascuna finestra deve avere una durata sufficiente per permettere l'invio di un frame. È calcolabile sapendo le dimensioni di entrambi i tipi di frame e la velocità di trasmissione.

Se il modulo hardware (implementato per misurare la latenza) rileva l'evento di invio dopo le code interne al MAC del trasmittente è possibile misurare due tipi di latenza:

- latenza del traffico “time triggered”: configurando il TAS anche sull'endpoint trasmittente il frame viene trasmesso all'interno della finestra dedicata, e viene immediatamente ritrasmesso dallo switch.
- latenza del traffico “event triggered”: non configurando il TAS sull'endpoint trasmittente il frame viene trasmesso appena possibile, e subisce un ritardo di sincronizzazione all'interno dello switch.

Se il modulo hardware rileva l'evento di invio prima del MAC, il ritardo di sincronizzazione con la finestra del TAS avviene in ogni caso.

Ci si aspetta di non avere packet loss, dato che si hanno code separate per i due tipi di traffico.

Il traffico di test “time triggered” non subisce nessuna interazione con il traffico di background, e ci si aspetta latenze simili alle latenze minime misurate durante gli altri test.

Il traffico di test “event triggered” avrà una latenza massima equivalente alla durata della finestra dedicata al traffico di background.

Capitolo 3

Realizzazione test

Questa sezione descrive il processo di test della rete TSN. Nel capitolo precedente si sono analizzati gli aspetti chiave di un test per reti TSN, la scelta della topologia, e la misura delle prestazioni, ovvero latenza e packet-loss.

Per alcuni aspetti, come la topologia, si è proposta un'unica opzione, non per mancanza di alternative. Per altri aspetti, come la misura della latenza, si sono analizzate diverse opzioni, dato che dipendono da quali device vengono scelti per costruire la rete di test.

Per i device della rete si è valutato l'utilizzo di progetti open-source. Dalle ricerche effettuate si è potuto trovare solo OpenTSN: un progetto open-source descritto in un paper[28]. Dopo alcune ricerche, condotte sul software scaricabile, descritte in appendice A, si è concluso che il progetto non è maturo, in quanto di difficile installazione e mancante di funzionalità. Si sono quindi utilizzate implementazioni proprietarie di switch ed end-point.

Per ragioni tecniche non si aveva modo di modificare gli switch. Si è quindi deciso di misurare la latenza di comunicazione tra end-point. Si sono utilizzati dei contatti sugli end-point su cui generare fronti di salita per notificare gli eventi. Per misurare il tempo che intercorre tra i due eventi si è utilizzato un oscilloscopio. Non è stato quindi possibile generare dei log delle misurazioni, che sono avvenute leggendo a schermo i dati riportati.

La precisione delle misurazioni è più o meno 10 nanosecondi, ovvero 0,01 microsecondi.

Si è implementato un modulo hardware, connesso al MAC dell'end-point, e ad un contatto esterno. Questo contatto viene alzato:

- lato trasmissione: quando il MAC inizia la trasmissione del frame al PHY.
- lato ricezione: quando il MAC inizia la ricezione dei dati dal PHY, e inizia ad accodare internamente il frame in ricezione.

Viene abbassato:

- lato trasmissione: quando il MAC inizia termina ricezione del frame dalla CPU.
- lato ricezione: quando il MAC termina la ricezione dei dati dal PHY.

Per generare traffico si è implementata una semplice routine, che inserisce un packet counter in ciascun frame e permette di configurare:

- l'indirizzo e la porta di destinazione
- la priorità
- l'intervallo tra un frame e l'altro
- la dimensione del frame

Si hanno due modalità di utilizzo di questo generatore di frame, può essere utilizzato sia per il traffico trigger che di background.

Per misurare la packet loss si è utilizzato un programma in user space, questa routine si coordina con il generatore in modo che:

- il generatore mandi frame in continuazione (periodicamente), senza curarsi della ricezione,

- il ricevente venga avviato manualmente sull'altro end-point dopo un certo tempo sufficiente a stabilizzare il sistema, ed analizza i primi N frame che riceve.

In questo modo non si deve coordinare l'avvio dei due programmi, e la parte di analisi statistica è stata tenuta solo lato ricevente. Il mittente non ha idea della quantità di frame contenuta in un test, quindi non è nemmeno necessario riavviarlo per ogni test, può essere lasciato operare in continuazione.

3.1 Risultati

Utilizzando fronti di salita su contatti, si è potuto osservare quanto mostrato in figura 3.1. I due fronti di salita sono intervallati dal ritardo di accodamento all'interno dello switch. Il fronte di inizio ricezione avviene all'interno di una finestra temporale, e definisce così la latenza minima, e la latenza massima. La latenza minima si ha quando il frame arriva e il canale di uscita è immediatamente pronto per ritrasmetterlo. La differenza tra le due latenze è il ritardo di accodamento.

3.1.1 Test con priorità

Si è configurato la sorgente per generare di traffico di test a priorità 7, ed il traffico di background a priorità 0. Si è configurato lo switch per utilizzare due code diverse per i due tipi di traffico e in modalità cut-through.

Si è misurata la latenza come il tempo che intercorre tra l'uscita dal MAC della sorgente e l'entrata al MAC di destinazione. Questa latenza è la somma dei ritardi per gli accodamenti e dei ritardi di propagazione.

I test, numerati, sono riportati nelle tabelle 3.1, 3.2 e 3.3:

1. test con i due soli endpoint, PTP abilitato;
2. test con i due soli endpoint, PTP disabilitato;

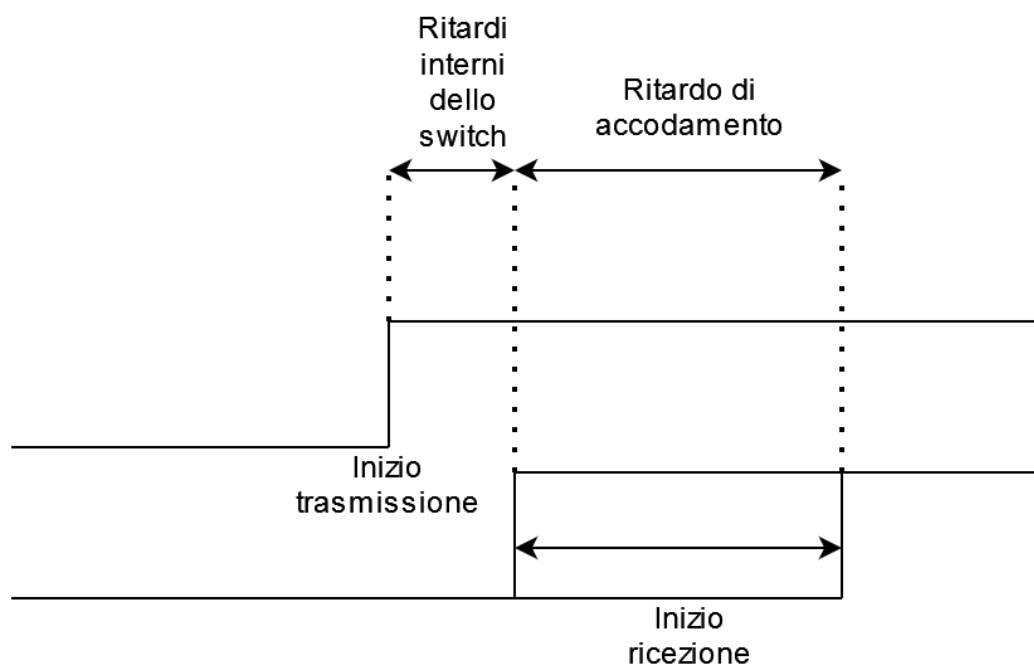


Figura 3.1: Risultati osservati sull'oscilloscopio

3. test con i due switch, nessun traffico di background, priorità configurate, PTP abilitato;
4. test con i due switch, nessun traffico di background, priorità configurate, PTP disabilitato;
5. test con i due switch, traffico di background, priorità configurate, PTP abilitato;
6. test con i due switch, traffico di background, priorità configurate, PTP disabilitato;
7. test con i due switch, traffico di background, FPE abilitata, PTP abilitato;
8. test con i due switch, traffico di background, TAS abilitato sui soli switch, PTP abilitato;

9. test con i due switch, traffico di background, TAS abilitato su tutti i device, PTP abilitato;

Ci si riferisce al delay introdotto dagli switch come “latenza di rete”, ed è calcolato per ciascun test come la differenza con la latenza misurata nei test 1 e 2, dove gli switch non sono presenti.

È possibile notare che:

- Si hanno $1,07\mu s$ di ritardo tra gli eventi rilevati dal test, questi dovranno essere sottratti a tutte le misure perché non fanno parte del ritardo introdotto dagli switch.
- La presenza del PTP non mostra influenze sui test.
- Ogni switch causa un ritardo di $1,58\mu s$ per ritrasmettere il frame. Questo ritardo è dovuto a operazioni interne, come per esempio la consultazione della forward lookup table.
- Aggiungendo il traffico di background la latenza massima aumenta di $9,11\mu s$, compatibile con il tempo necessario a trasmettere un intero frame di background (1045 byte di dati, e 30 di header, preambolo, e CRC).

I risultati si possono anche osservare in figura 3.3

3.1.2 Test frame preemption

Si può confrontare il test 7 ed i test 5 e 6, e per vedere gli effetti introdotti dalla FPE. La latenza di rete massima cala da $12,03\mu s$ a $3,95\mu s$, indicando un tempo di $0,77\mu s$ per inviare un frammento nel caso peggiore. Analiticamente il caso peggiore dovrebbe essere 127 ottetti a 1Gbps, ovvero un $1\mu s$ di ritardo extra. È coerente con quanto misurato. I risultati si possono osservare in figura 3.3.

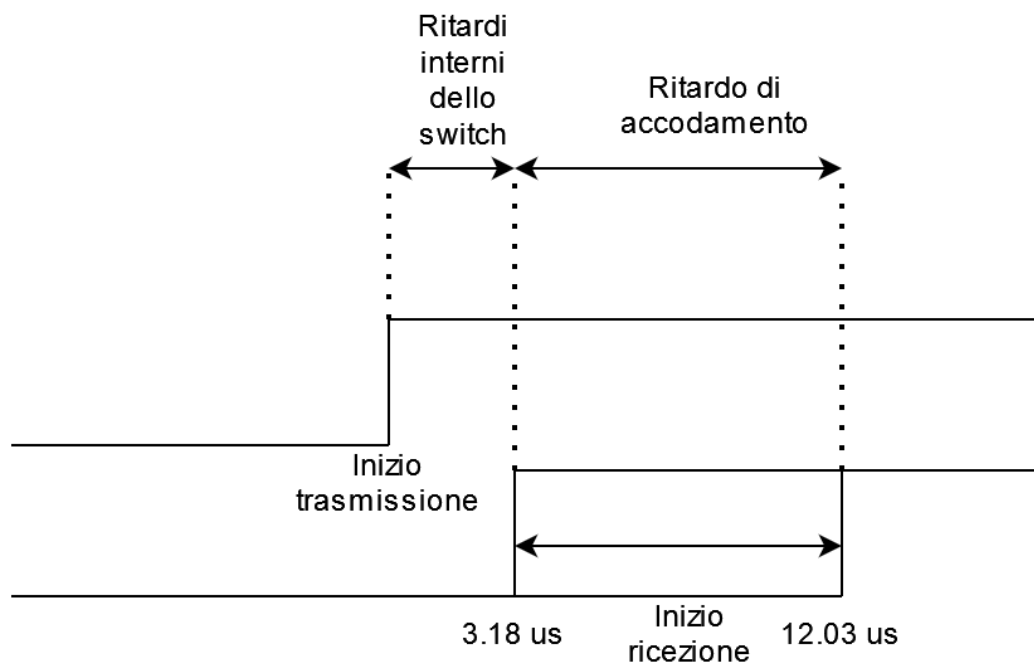


Figura 3.2: Visualizzazione grafica dei risultati con IEEE 802.1P

Numero test	PTP	Latenza minima (μs)	Latenza massima (μs)	Latenza di rete minima (μs)	Latenza di rete massima (μs)	Packet-loss
1	Si	1,07	1,08	0	0	0
2	No	1,07	1,08	0	0	0
3	Si	4,25	4,26	3,18	3,19	0
4	No	4,25	4,26	3,18	3,19	0
5	Si	4,25	13,10	3,18	12,03	0
6	No	4,25	13,10	3,18	12,03	0

Tabella 3.1: Test effettuati utilizzando la sola priorità 802.1P

Numero test	Latenza di rete minima (μs)	Latenza di rete massima (μs)	Packet-loss
7	3,18	3,95	0

Tabella 3.2: Test per valutare le prestazioni della FPE

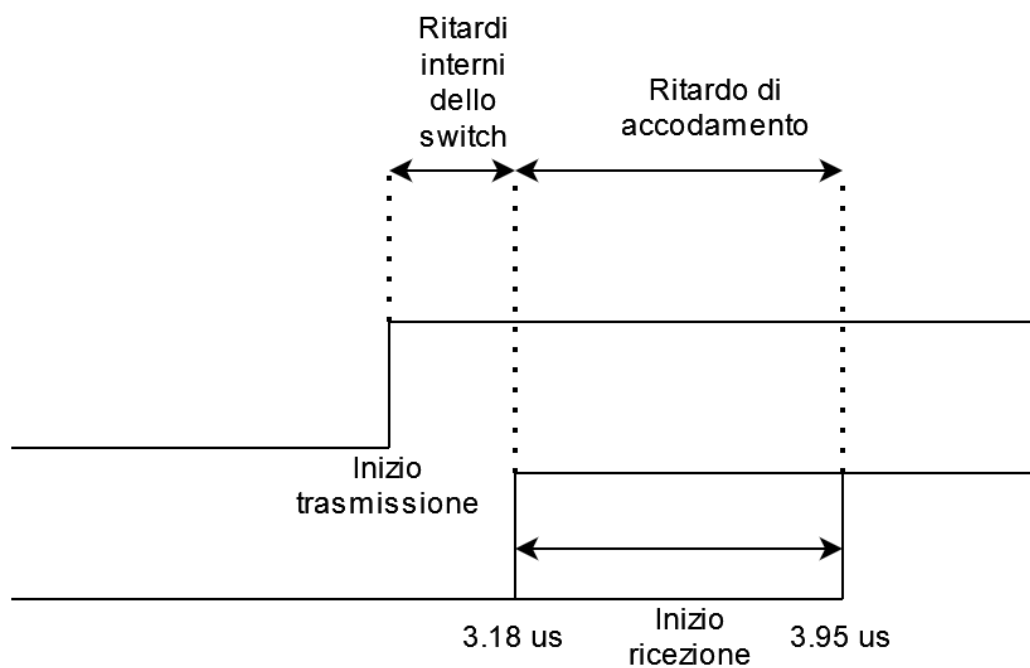


Figura 3.3: Visualizzazione grafica dei risultati con frame preemption

Numero test	Latenza di rete minima (μs)	Latenza di rete massima (μs)	Packet-loss
8	3,18	23,18	0
9	3,18	3,19	0

Tabella 3.3: Test effettuati utilizzando il TAS

3.1.3 Test Time aware shaper

È stato configurato il TAS sugli switch. Sono state configurate due finestre: una di $30\mu s$ per il traffico ad alta priorità ed una di $20\mu s$ per il restante traffico, che nella rete di test corrisponde al solo traffico di background. Si è simulato sia il traffico “event triggered” (test 8) che quello “time triggered” (test 9).

Si è potuto verificare che il traffico “time triggered” ha latenze paragonabili al test 3, dove non si aveva traffico di background. E il traffico “event triggered” ha una latenza aggiuntiva di $20\mu s$, durata della finestra dedicata al traffico di background.

Conclusioni

L'ambiente di test per reti TSN è stato descritto e implementato. Ha mostrato i risultati aspettati durante la verifica su una rete TSN.

I protocolli TSN da verificare sono stati scelti in base ai meccanismi già utilizzati in ambito industriale. Nei protocolli Ethernet real-time si è spesso utilizzato TDMA, quindi sono stati integrati test sull'IEEE 802.1Qbv. Per utilizzare i vantaggi degli switch in modalità cut-through alcuni protocolli impongono topologie a catena, grazie allo standard IEEE 802.1Qbu si possono avere gli stessi vantaggi, ma mantenendo libertà di scelta sulla topologia.

Si è ipotizzato di poter utilizzare PTP come mezzo per sincronizzare dispositivi e misurare la latenza. Utilizzando dei fronti di salita su contatti si è verificato che PTP non influisce sull'esecuzione dei test, ed è quindi utilizzabile all'interno di ambienti di test.

Questo ambiente di test può essere espanso, aggiungendo test che tengano in considerazione anche altri standard IEEE TSN, valutando se la topologia rimane valida o se è necessario aggiungere dispositivi.

Appendice A

OpenTSN

OpenTSN è presentato in un paper[28], il quale fa riferimento ad una repository gitee per il sorgente. È stato trovato un repository github che fa riferimento allo stesso repository gitee.

All'interno del repository è possibile notare diversi sorgenti verilog e C, in cui apparentemente vengono forniti sia switch che end-point. Analizzando il codice nella repository si può notare che tutti i moduli verilog si interfacciano verso un componente unico, il quale è anche l'unico componente ad avere segnali instradati verso i contatti dell'FPGA.

Questo componente è chiamato FastOS, ed è possibile notarlo anche nel paper di presentazione di openTSN, nella figura A.1 qui riportata. Tutto quello che segue è stato ipotizzato studiato autonomamente il sorgente trovato in openTSN. FastOS è un IPcore generato da un tool chiamato Fast-Framework, strumento proprietario della Fast. Si utilizza FastFramework per gestire gli IPcore di libreria, come per esempio tri-state-Ethernet e code FIFO. Siccome tutti i moduli verilog forniti si interfacciano con i componenti presenti in FastOS è necessario un attento lavoro per adeguare quei moduli ad utilizzare componenti di librerie differenti da FastOS.

Vengono date alcuni FastOS generati per, apparentemente, utilizzare da una a quattro connessioni GMII. Non è quindi possibile compilare openTSN per hardware differente da quello per cui è fornito.

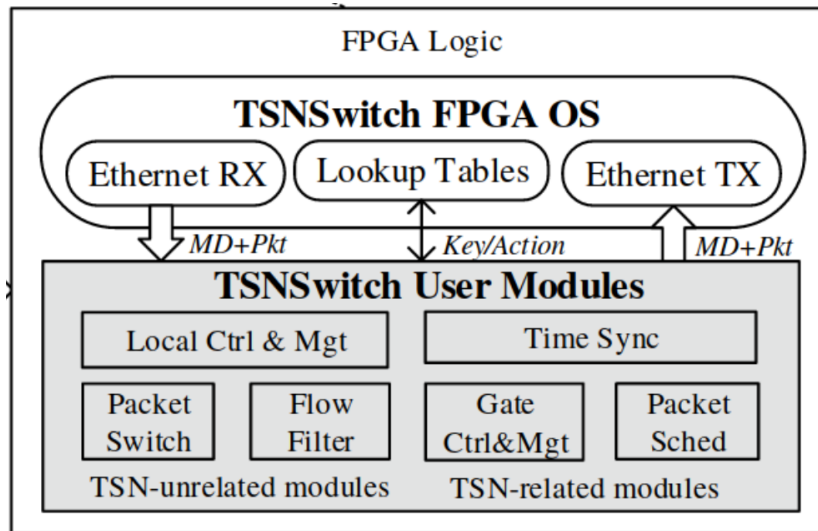


Figura A.1: struttura di OpenTSN riportata nel paper

Chiedendo informazioni agli sviluppatori si è riusciti ad ottenere informazioni su openTSN[32]. Hanno illustrato che attualmente openTSN implementa 802.1AS, 802.1Qbv, 802.1Qcc, 802.1Qch, ma manca 802.1Qbu, per questo motivo non è possibile utilizzarlo per gli scopi prefissati. Si poteva espandere openTSN, ma dato il carico di lavoro non finalizzato agli scopi di questa tesi, si è deciso di abbandonare questa strada.

Bibliografia

- [1] Anna Arestova, Kai-Steffen Jens Hielscher e Reinhard German. «Simulative Evaluation of the TSN Mechanisms Time-Aware Shaper and Frame Preemption and Their Suitability for Industrial Use Cases». In: *2021 IFIP Networking Conference (IFIP Networking)*. IEEE. 2021, pp. 1–6.
- [2] Armando Astarloa et al. «FPGA implemented cut-through vs store-and-forward switches for reliable Ethernet networks». In: *Design of Circuits and Integrated Systems*. IEEE. 2014, pp. 1–6.
- [3] Shahram Babaie et al. «Double bits error correction using CRC method». In: *2009 Fifth International Conference on Semantics, Knowledge and Grid*. IEEE. 2009, pp. 254–257.
- [4] Silviu S Craciunas et al. «Scheduling real-time communication in IEEE 802.1 Qbv time sensitive networks». In: *Proceedings of the 24th International Conference on Real-Time Networks and Systems*. 2016, pp. 183–192.
- [5] Paula Doyle. «Introduction to real-time Ethernet I». In: *Contemporary Control Systems* 5.4 (2004).
- [6] José Duato. «A necessary and sufficient condition for deadlock-free routing in cut-through and store-and-forward networks». In: *IEEE Transactions on Parallel and Distributed Systems* 7.8 (1996), pp. 841–854.

-
- [7] Max Felsler. «Real-time ethernet-industry prospective». In: *Proceedings of the IEEE* 93.6 (2005), pp. 1118–1129.
- [8] Ahmad Fraihat. «Computer Networking Layers Based on the OSI Model». In: (2020).
- [9] Brendan Galloway e Gerhard P Hancke. «Introduction to industrial control networks». In: *IEEE Communications surveys & tutorials* 15.2 (2012), pp. 860–880.
- [10] Geoffrey M Garner e Hyunsurk Ryu. «Synchronization of audio/video bridging networks using IEEE 802.1 AS». In: *IEEE Communications Magazine* 49.2 (2011), pp. 140–147.
- [11] Mandeep Ghuman e Shammi Bahel. «QoS Estimation of IEEE 802.3 under the Influence of Intra-Domain Routing Protocols». In: *IJCSC* 4 (2013), pp. 1–7.
- [12] Y Haibo et al. «The architecture and realtime communication of CNC systems based on switched Ethernet». In: *Computer Engineering and Technology (ICCET), 2010 2nd International Conference*. Vol. 1. 2010, pp. 169–173.
- [13] Liu Jianxin, Guan Xuefeng e Tan Ping. «The analysis and test of real-time performance for time-triggered CAN bus». In: *2008 IEEE International Conference on Automation and Logistics*. IEEE. 2008, pp. 3005–3009.
- [14] Moonbeom Kim et al. «TAS Scheduling for Real-Time Forwarding of Emergency Event Traffic in TSN». In: *2020 International Conference on Information and Communication Technology Convergence (ICTC)*. IEEE. 2020, pp. 1111–1113.
- [15] Yuetsu Kodama et al. «GNET-1: Gigabit Ethernet network testbed». In: *2004 IEEE International Conference on Cluster Computing (IEEE Cat. No. 04EX935)*. IEEE. 2004, pp. 185–192.

- [16] Milos Kovacevic et al. «Towards implementation of frame preemption mechanism on FPGA platform». In: *2020 19th International Symposium INFOTEH-JAHORINA (INFOTEH)*. IEEE. 2020, pp. 1–7.
- [17] Seok-Kyu Kweon e Kang G. Shin. «Statistical real-time communication over Ethernet». In: *IEEE Transactions on parallel and distributed systems* 14.3 (2003), pp. 322–335.
- [18] Physical Layer. «IEEE Standard for Ethernet». In: (2018).
- [19] What You Will Learn. «Priority Flow Control: Build Reliable Layer 2 Infrastructure». In: (2009).
- [20] Duke Lee et al. «FLORAX-flow-rate based hop by hop backpressure control for IEEE 802.3 x». In: *5th IEEE International Conference on High Speed Networks and Multimedia Communication (Cat. No. 02EX612)*. IEEE. 2002, pp. 202–207.
- [21] Hyung-Taek Lim et al. «IEEE 802.1 AS time synchronization in a switched Ethernet based in-car network». In: *2011 IEEE Vehicular Networking Conference (VNC)*. IEEE. 2011, pp. 147–154.
- [22] Igor Mohor. «Ethernet IP Core Specification». In: *Rev04 octobre* (2002).
- [23] Ahmed Nasrallah et al. «Performance comparison of IEEE 802.1 TSN time aware shaper (TAS) and asynchronous traffic shaper (ATS)». In: *IEEE Access* 7 (2019), pp. 44165–44181.
- [24] Peter Neumann e Axel Poschmann. «Ethernet-based real-time communications with PROFINET IO.» In: *WSEAS Transactions on Communications* 4.5 (2005), pp. 235–245.
- [25] Davide Pedretti et al. «Nanoseconds timing system based on IEEE 1588 FPGA implementation». In: *IEEE Transactions on Nuclear Science* 66.7 (2019), pp. 1151–1158.
- [26] Gunnar Prytz. «A performance analysis of EtherCAT and PROFINET IRT». In: *2008 IEEE International Conference on Emerging Technologies and Factory Automation*. IEEE. 2008, pp. 408–415.

-
- [27] Henning Puttnies et al. «A Simulation Model of IEEE 802.1 AS gPTP for Clock Synchronization in OMNeT++.» In: *OMNeT++*. 2018, pp. 63–72.
- [28] Wei Quan et al. «OpenTSN: an open-source project for time-sensitive networking system development». In: *CCF Transactions on Networking* 3.1 (2020), pp. 51–65.
- [29] Jung Hwan Seo e Jae Wook Jeon. «Comparison of IEEE802. 1Q and IEEE802. 1AVB in multi switch environment in embedded system». In: *2017 17th International Conference on Control, Automation and Systems (ICCAS)*. IEEE. 2017, pp. 489–493.
- [30] Jeff Stribling et al. «Implementing QoS in SIEPON». In: *IEEE Communications Magazine* 50.9 (2012), pp. 128–135.
- [31] Michael D Johas Teener e Geoffrey M Garner. «Overview and timing performance of IEEE 802.1 AS». In: *2008 IEEE International Symposium on Precision Clock Synchronization for Measurement, Control and Communication*. IEEE. 2008, pp. 49–53.
- [32] *TSN standards supported by OpenTSN*. <https://web.archive.org/web/20211124124850/https://github.com/fast-codesign/OpenTSN3.0/issues/1>. Accessed: 2021-11-24.
- [33] Jerzy Wechta, Martin Fricker e Fred Halsall. «Hop-by-hop flow control as a method to improve QoS in 802.3 LANs». In: *1999 Seventh International Workshop on Quality of Service. IWQoS'99.(Cat. No. 98EX354)*. IEEE. 1999, pp. 239–247.

Ringraziamenti

In primis, vorrei ringraziare i correlatori che mi hanno seguito durante questo lavoro: Guido Munari e Davide Berardi. Sono stati sempre presenti e sempre disponibili, accompagnandomi durante il tirocinio e durante la stesura di questo documento, dandomi modo di provare e imparare dai miei errori.

Vorrei ringraziare la mia famiglia, che mi ha supportato durante questi anni di studio, e che ha sempre creduto in me, e nelle mie scelte.

Vorrei ringraziare i miei amici, buona parte dei quali conosciuti durante questo percorso iniziato sei anni fa.

E ultimo ma non ultimo, vorrei ringraziare Mauro Baldazzi, che ha contribuito alla mia formazione tanto quanto l'università, se non anche in misura maggiore. Grazie a cui ho scoperto il significato della parola “meccatronica”.

Grazie.