DIPARTIMENTO DI INGEGNERIA DELL'ENERGIA ELETTRICA E

DELL'INFORMAZIONE "GUGLIELMO MARCONI" - DEI

CORSO DI LAUREA IN INGEGNERIA ELETTRONICA

"ANALISI DELLE POTENZIALITÀ DELLE SCHEDE DI RETE: FUNZIONALITÀ, OFFLOADING E PROSPETTIVE FUTURE"

Tesi di laurea in Comunicazioni Digitali e Internet

Relatore

Prof. Walter Cerroni

Presentata da

Giulio Flamigni

Sessione di settembre 2025

Anno Accademico 2024/2025

INDICE

INDICE DELLE FIGURE

INTRODUZIONE

L'Informazione gioca un ruolo fondamentale nella moderna società globale dal punto di vista socioeconomico, rendendo possibile un vasto e rapido scambio di notizie e opinioni e creando consapevolezza delle dinamiche dei diversi luoghi del pianeta in tempo reale.

I dati viaggiano all'interno di reti di utenti e la più estesa architettura di rete è **Internet**, spesso riferita anche come "La Rete", la quale possiede una copertura globale.
Sebbene in molti abbiano tentato di quantificare l'entità di Internet, ad esempio di misurare il numero di dispositivi connessi e di dati scambiati, è tuttavia impossibile conoscere esattamente tali valori in quanto esso è topologicamente e tecnologicamente variegato e strutturato come un insieme di sottosistemi collegati ma autonomi, ovvero una "rete di reti" [1].

Secondo una stima dell'*International Data Corporation* (IDC) del gennaio 2021, nel 2025 si sarebbero raggiunti ben 55.7 miliardi di dispositivi connessi all'Internet of Things, i quali avrebbero generato una quantità di quasi 80 Zettabyte di dati [2].

Un sondaggio di *Statista* del giugno 2024 affermava che il numero di apparati collegati nel 2023 era 15.9 miliardi ed era previsto quasi un raddoppiamento per il 2030, raggiungendo la quota di 32.1 miliardi [3].

L'edizione dodici del 2024 di *Data Never Sleeps*, curata da *DOMO*, sosteneva che Internet avesse raggiunto ben 5.52 miliardi di utenti ed entro la fine dell'anno sarebbero stati "creati, catturati, copiati e consumati", ben 149 Zettabyte di dati, quantità destinata ad aumentare fino a 394 nel 2028 [4].

Sempre per quanto concerne l'utenza, l'indagine promossa dall'*Unione Internazionale delle Telecomunicazioni* (ITU) riporta che nel 2024 il 67% delle persone del globo era connessa a Internet [5].

Questi sono solo alcuni tentativi di quantificazione del fenomeno e da tutti traspaiono dei fattori comuni: le informazioni che transitano sono ingenti e la Rete è capillarmente diffusa sul pianeta.

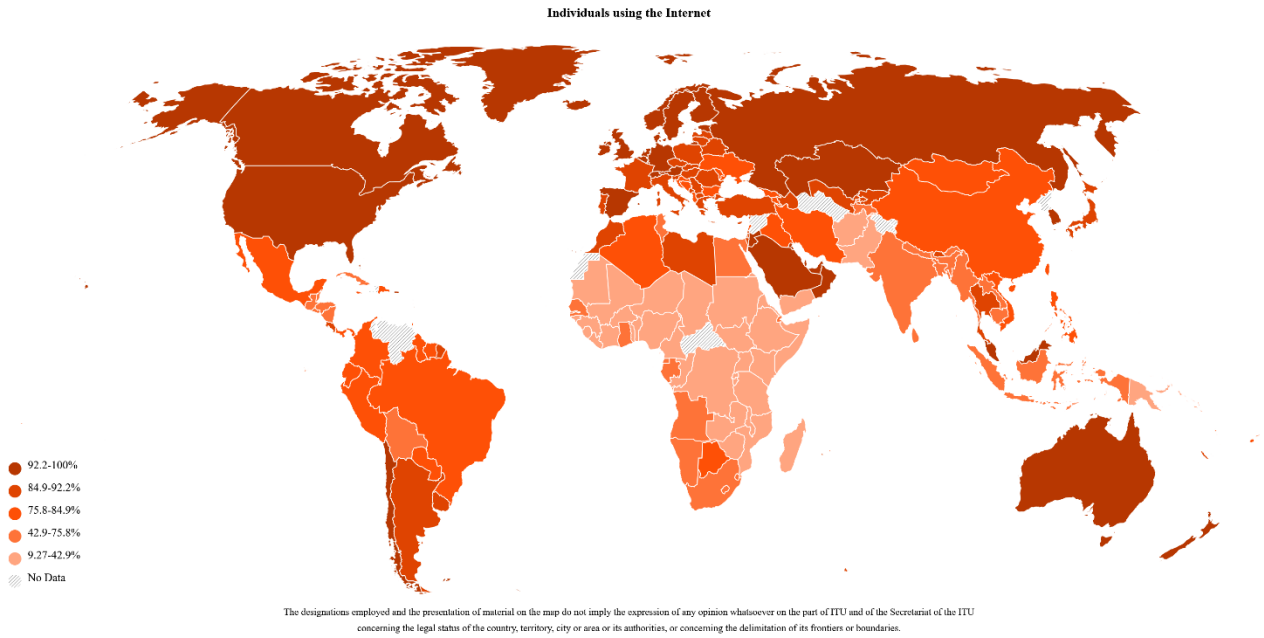


Figura 1: Mappa degli utenti connessi a Internet [5]

Un funzione significativa in questo contesto è giocata dalle telecomunicazioni, intese come qualsiasi procedimento grazie al quale sia possibile trasmettere a distanza informazioni [6].

Le **infrastrutture di telecomunicazione** rappresentano la struttura portante delle reti e, grazie alle tecnologie moderne, lo scambio di dati avviene tra luoghi remoti con velocità estremamente elevate e mediante un unico linguaggio, quello binario.
Un'infrastruttura di rete prevede l'esistenza di un apparato di smistamento e indirizzamento dei messaggi e la presenza di terminali che possano trasmetterli e/o riceverli. Pertanto, per essere efficiente, un'architettura deve mantenere aggiornati entrambi i componenti.

Un dispositivo elettronico fondamentale a questo proposito è la **scheda di rete**, componente necessario a qualunque apparato che debba collegarsi ad Internet, dalla smart tv, allo smartphone, al personal computer, fino ai server dei fornitori di servizi multimediali.

Per garantire un corretto funzionamento della connessione, ridurre i tempi e i rischi di fallimenti, sono necessarie operazioni di controllo a diversi livelli di astrazione, definiti dal modello di comunicazione adottato.
Queste operazioni rappresentano un "costo computazionale" gravoso per le risorse dei calcolatori, quali la CPU, portando ad inefficienza in un sistema complesso e specialmente multitasking.

Recentemente numerosi produttori di schede di rete hanno introdotto componenti in grado di svolgere parzialmente o integralmente queste operazioni, in una modalità che prende il nome di “Offloading”.

Avendo trovato interessanti queste nuove tecnologie, ho deciso di sviluppare la tesi intorno ad esse, con l’obiettivo di fare chiarezza sulle loro utilità e potenzialità.

Il primo capitolo espone una breve panoramica dei protocolli di comunicazione ISO/OSI e TCP/IP, in preparazione per i contenuti successivi. Nel secondo capitolo vengono introdotte le schede di rete, relative a reti cablate e wireless, con approfondimento sugli ultimi prodotti tecnologici. Quindi il terzo capitolo elenca le funzionalità delle schede di rete, partendo da quelle classiche per giungere alle frontiere dell’offload, avendo rivolto lo sguardo verso i vantaggi e gli svantaggi associati.
Infine, nel quarto, riporto i risultati di due esperimenti di laboratorio per la valutazione delle performance di alcune funzionalità di offload.

1 MODELLI DI INTERNET

Nonostante l'ampia varietà topologica e tecnologica delle infrastrutture delle reti, che possono essere considerevolmente differenti tra di loro, Internet tenta di far comunicare i dispositivi indipendentemente dalla realtà fisica a cui appartengono. Per rendere possibile ciò, in passato sono stati definiti modelli che fossero universalmente applicabili e implementabili in modo versatile. I due modelli principali sono l'ISO/OSI e il TCP/IP.

1.1 ISO/OSI

Il modello ISO/OSI fu definito per la prima volta nel 1984 dall'*International Organization for Standardization* (ISO) e successivamente revisionato nel 1994 [7].

Esso organizza l'architettura di un apparato di rete in sette livelli di gestione della comunicazione, i quali possono essere elencati in ordine di astrazione decrescente da quello più prossimo all'utente a quello aderente alle tecnologie meccaniche ed elettriche:

Livello 7: è chiamato livello di **Applicazione**, è il livello massimo, che si interfaccia con i processi applicativi in esecuzione sul dispositivo utilizzato dall'utente. Le sue funzioni annoverano l'identificazione dei partner di comunicazione, la definizione della qualità di servizio (QoS), la sincronizzazione delle applicazioni cooperanti, l'accordo sulla gestione degli errori e il trasferimento dei dati sicuro. Esso sfrutta protocolli per garantire l'autenticazione, l'integrità e il controllo di accesso. Viene inoltre decisa la modalità di dialogo e la sintassi dei dati scambiati.

Livello 6: noto come **Presentazione**, si occupa di adattare la sintassi dei messaggi provenienti dal livello applicativo senza corromperne la semantica, tramite le funzioni di compressione, traduzione del formato e cifratura dei contenuti.

Livello 5: chiamato livello di **Sessione**, ha la funzione di gestire le richieste di invio di messaggi provenienti da più processi in esecuzione, creando delle sessioni di dialogo. Ciò si articola nelle funzioni di apertura, sincronizzazione e chiusura ordinate delle sessioni, attraverso i protocolli di connessione. Stabilisce inoltre i criteri per lo scambio ordinato di dati, secondo dei protocolli di comunicazione. A tale proposito impiega dei *punti di sincronizzazione* che permettono, nel caso di errori, di far ripartire ordinatamente il dialogo;

Livello 4: conosciuto come **Trasporto**, provvede a un trasferimento trasparente dei dati svincolando gli strati superiori dalle problematiche di rete. Ciò si traduce nella realizzazione di un canale di comunicazione end-to-end tra mittente e destinatario, che può essere di tipo connection-oriented o connectionless, a seconda che i dati siano trasferiti come un flusso controllato sequenzialmente oppure inviati come datagrammi autonomi.

La modalità priva di connessione è in grado di implementare alcune funzionalità come la creazione di segmenti di lunghezza prefissata, la multiplazione di più connessioni mediante l'ausilio di porte e il controllo della QoS, senza tuttavia potere agire direttamente su di essa. È inoltre presente un sistema di controllo dell'errore, privo della possibilità di ripetizione del messaggio errato.

La modalità con connessione invece possiede ulteriori funzioni, quali il controllo di flusso e di sequenza, la sospensione e ripresa della comunicazione e la gestione di dati prioritari. È così possibile anche influenzare la QoS. Nel caso in cui un pacchetto non sia riscontrato positivamente dal ricevitore, perché smarrito o giunto errato, il trasmettitore provvederà a reinviarlo dopo un tempo di attesa.

Livello 3: **Rete**, ha il compito di smistare all'interno della Rete le unità informative, scegliendo in maniera ottimale il percorso da effettuare tra i nodi della rete mediante uno schema di indirizzamento universale.

A tale fine sono implementate le funzioni di inoltro e indirizzamento che sfruttano algoritmi di routing e metodi di frammentazione dei pacchetti.
Gli algoritmi permettono di definire e aggiornare le tabelle di indirizzamento dei diversi nodi della rete, mentre i metodi rendono possibile il passaggio dei pacchetti nei tratti della rete in cui la dimensione massima consentita è limitante.
Inoltre, questo livello prevede anche l'utilizzo di sistemi per il rilevamento e recupero dell'errore e un servizio per il mapping tra indirizzi di rete e indirizzi di livello di Collegamento, il protocollo ARP.

Livello 2: noto come **Collegamento**, ha la funzione di rendere affidabile la comunicazione link-by-link, ovvero tra due nodi adiacenti, gestendo la creazione di trame adatte al mezzo trasmissivo e funzioni di controllo e recupero dell'errore, controllo di flusso e di congestione. Infine, sovrintende l'accesso al mezzo condiviso, mediante protocolli di gestione delle collisioni.

Livello 1: è il livello **Fisico**, prossimo alla realtà elettromeccanica del sistema di telecomunicazione. Esso ha la funzione di trasferire i bit logici mediante l'ausilio di segnali elettromagnetici adatti al mezzo trasmissivo.

A tal fine deve gestire problematiche meccaniche, elettriche, procedurali e funzionali proprie dello specifico mezzo di comunicazione elettrico, ottico o radio.
Tra le funzioni si annoverano la gestione e il mantenimento della connessione fisica e la notifica ai livelli superiori di eventuali guasti nella linea di trasmissione.

In conclusione, l'articolazione nei sette strati di astrazione ha il fine di dividere le necessità della comunicazione a più livelli per garantire una certa qualità di servizio (QoS). È così possibile realizzare ogni livello autonomamente, disponendo di interfacce che permettano il dialogo tra livelli adiacenti, chiamate *Service Access Point* (SAP).

1.2 TCP/IP

Il modello oggi utilizzato per la gestione di Internet è il TCP/IP, che prende i passi dal modello ISO/OSI adottandone le funzioni, ma dividendo la comunicazione su quattro livelli invece di sette. Esso è definito dalla *Internet Engineering Task Force* (IETF) all'interno di documenti diffusi online con la sigla *Request For Comment* (RFC) che con il passare del tempo hanno subito aggiornamenti fino a diventare, alcuni, standard.

Esso prevede quattro livelli di astrazione: Applicazione, Trasporto, Rete e Collegamento. Il livello di Collegamento comprende le funzioni dei due livelli inferiori dell'ISO/OSI, il livello di Rete è analogo all'omonimo, mentre i livelli di Trasporto e Applicazione, oltre a gestire rispettivamente le funzioni degli strati quattro e sette, si spartiscono anche le proprietà dei livelli di Sessione e Presentazione.

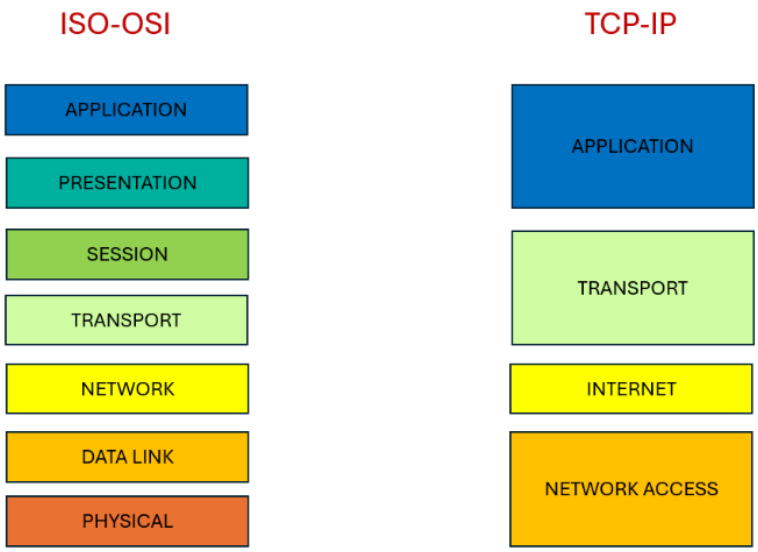


Figura 2: Modelli di rete a confronto

Le schede di rete implementano importanti funzioni legate al livello di Collegamento ma talvolta possono svolgere anche calcoli di *offload* legati ai livelli di Rete e Trasporto. Le funzioni del livello applicativo sono molto astratte ed eseguite via software.

Di conseguenza, risulta opportuno approfondire la natura dei tre livelli inferiori.

1.2.1 TCP e UDP

Lo strato di Trasporto prevede due protocolli, il *Transmission Control Protocol* (TCP) [8] [9], che instaura connessioni affidabili e lo *User Datagram Protocol* (UDP) [10] che invece comunica mediante datagrammi.
Talvolta, il protocollo UDP potrebbe collaborare con un protocollo di livello applicativo ausiliario, il QUIC, che fornisce multiplexing, controllo di flusso e di sicurezza [11].

Come precedentemente descritto per il livello quattro dello standard ISO/OSI, le funzioni del TCP sono maggiori di quelle dell'UDP, pertanto verranno descritte quelle del primo, includendo così anche quelle del secondo.

Un'importantissima funzione è la frammentazione delle sequenze di dati provenienti dai processi applicativi, che possono possedere lunghezze imprevedibili, producendo dei segmenti TCP di dimensione prefissata pari al più alla *Maximum Transmission Unit* (MTU). Questa funzione viene usualmente implementata in offload nelle schede di rete con il nome di **Segmentation Offload**.

Un'altra funzione rilevante è la multiplazione e successiva demultiplazione delle connessioni dell'host, che avviene mediante *porte* specifiche per il processo applicativo. Ogni pacchetto TCP fa riferimento a una porta di invio del mittente e una porta di ricezione del destinatario, definite in due campi distinti di 16 bit ciascuno nell'intestazione del pacchetto.

Per quanto riguarda il controllo degli errori, esso viene eseguito mediante l'algoritmo dell'*Internet Checksum*. Essendo la computazione abbastanza semplice ma onerosa, le schede di rete frequentemente possiedono logiche per la sua gestione, ovvero la funzionalità di **Checksum Offload**.

Ci sono poi il controllo di sequenza e la ritrasmissione delle trame non riscontrate, il controllo di flusso e di congestione.

Per verificare la corretta sequenzialità dei pacchetti è presente un campo a 32 bit nell'intestazione TCP, per la numerazione univoca delle trame.

Per quanto riguarda il controllo di flusso, esso viene implementato mediante un meccanismo a finestra scorrevole gestito a credito secondo cui il ricevitore costantemente comunica al trasmettitore la massima quantità di segmenti che esso può inoltrargli senza riscontro, ovvero la *Advertised Window* (AW). A tale funzione sono riservati 16 bit di intestazione.

Infine, il controllo di congestione è perseguito mediante la regolazione di una *Congestion Window* (CW), che viene aggiornata dal trasmettitore stesso, analizzando i tempi di riscontro delle trame.

Queste funzioni non sono frequentemente implementate nelle schede di rete, potrebbero essere presenti in quelle che possiedono la dicitura **Full TCP Offload**.

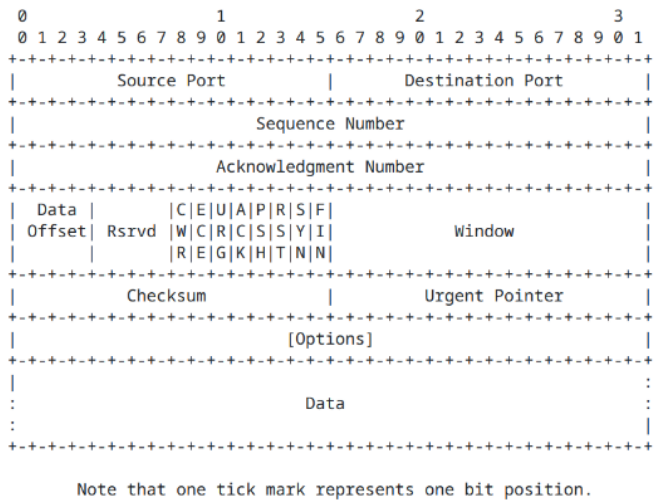


Figura 3: Intestazione TCP [9]

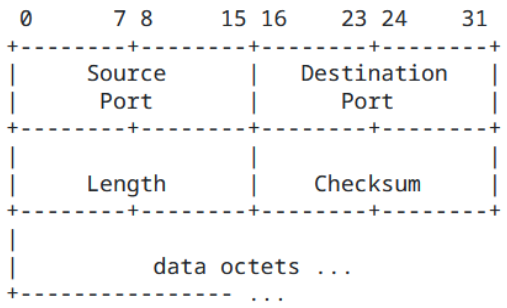


Figura 4: Intestazione UDP [10]

1.2.2 IP

Lo scopo principale di questo livello, come precedentemente detto, è quello di far giungere correttamente le unità informative dal mittente al destinatario scegliendo opportunamente il percorso attraverso la rete.
Ciò avviene mediante due operazioni: l'instradamento, e l'inoltro.

L'**instradamento** consiste nella scelta del canale in cui inoltrare il pacchetto. La decisione è affidata a una tabella di routing, che determina, sulla base del destinatario finale, il nodo adiacente su cui inoltrare il pacchetto. Le tabelle sono generalmente programmate per aggiornarsi continuamente, grazie allo scambio di pacchetti tra routers in varie modalità.

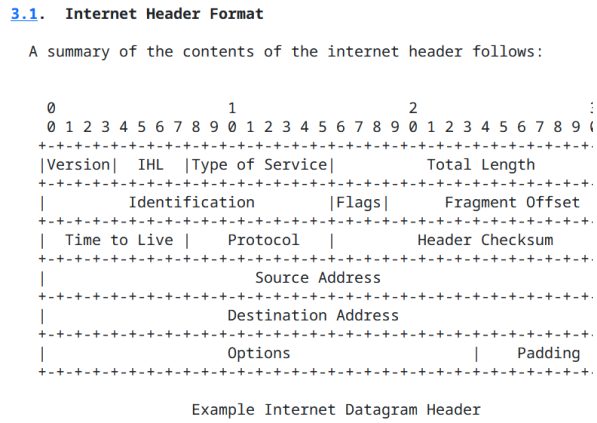
Successivamente avviene l'**inoltro**, ovvero il trasferimento fisico dell'unità informativa dalla porta di ingresso alla porta di uscita, funzione svolta dalla scheda di rete, che si interfaccia mediante un'opportuna interfaccia al mezzo di comunicazione.

Esistono due versioni del protocollo IP: *IPv4* [12] e *IPv6* [13]. Sebbene la versione sei aggiorni la quattro, quest'ultima non è affatto in disuso.
I due protocolli differiscono per le funzionalità implementate, che richiedono bit di controllo aggiuntivi. Conseguentemente, le intestazioni sono differenti.

L'indirizzamento degli host si evolve passando dagli indirizzi precedenti a 32 bit ai nuovi indirizzi a 128 bit i quali possono essere di tre tipi, analogamente alla modalità di instradamento. Ciò si basa sulla sintassi degli indirizzi, dotati di prefisso di rete e suffisso di host, grazie alla quale diventa possibile gestire lo *scope* dell'indirizzamento. Le tre modalità sono *unicast*, la quale prevede l'indirizzamento univoco a uno specifico nodo, *anycast*, nella quale un indirizzo è associato a più interfacce e il router ne sceglie una sulla base della distanza minima del nodo corrispondente. Infine, la modalità *multicasting*, nella quale l'indirizzo specifica un insieme di interfacce e il pacchetto è inoltrato ad ognuna di esse.

Viene introdotto l'etichettamento dei flussi e aggiunti dei bit di controllo per supportare autenticazione, integrità e confidenzialità, funzioni tipiche del livello di Applicazione ma che sono diventati argomento di interesse anche per il livello di Rete.
Alcune funzionalità sono rese opzionali per risparmiare spazio nell'intestazione e, infine, la Checksum di livello di Rete viene rimossa.
Infatti, il TCP include nel campo di interesse della propria Checksum anche uno pseudo-header IP, diminuendo così la ridondanza dell'operazione.

Nel caso in cui sia utilizzato il protocollo IPv4 e quindi sia presente la Checksum di Rete, diventa interessante la disponibilità del relativo offload da parte della scheda di rete.



3.1. Internet Header Format

A summary of the contents of the internet header follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Version|  IHL  |Type of Service|          Total Length         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Identification        |Flags|      Fragment Offset    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Time to Live |    Protocol   |         Header Checksum       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Source Address                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Destination Address                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Options                    |    Padding    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Example Internet Datagram Header

Figura 5: Intestazione IPv4 [12]

. IPv6 Header Format

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Version| Traffic Class |           Flow Label                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|         Payload Length        |  Next Header  |   Hop Limit   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                                                               +
|                                                               |
+                         Source Address                        +
|                                                               |
+                                                               +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                                                               +
|                                                               |
+                      Destination Address                      +
|                                                               |
+                                                               +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figura 6: Intestazione IPv6 [13]

Talvolta, l'header IPv6 potrebbe presentare molteplici estensioni, dal momento che è personalizzabile.

1.2.3 Livello di Collegamento

A causa della variabilità delle implementazioni fisiche delle reti, l'ISO non ha standardizzato il livello di Collegamento. Per quanto concerne le reti locali e metropolitane, uno standard ampiamente diffuso è quello di IEEE, al quale viene fatto riferimento nel corso della tesi. Esso propone i sottolivelli *Logical Link Control* (LLC) e *Medium Access Control* (MAC). Il primo ha il compito di gestire i collegamenti logici, mentre il secondo governa le procedure di accesso al messo condiviso.

Per garantire una connessione affidabile link-by-link vengono implementate le funzioni di strutturazione del flusso di dati in unità informative denominate trame, rivelazione e recupero degli errori, controllo di flusso, sequenza e accesso al mezzo condiviso.

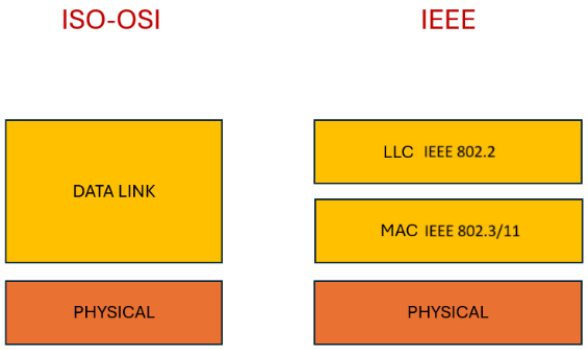


Figura 7: Sottolivelli definiti da IEEE

Di seguito vengono riportati i formati delle trame degli standard di interesse per gli argomenti successivi, ovvero IEEE 802.3, Ethernet, e IEEE 802.11, Wi-Fi.

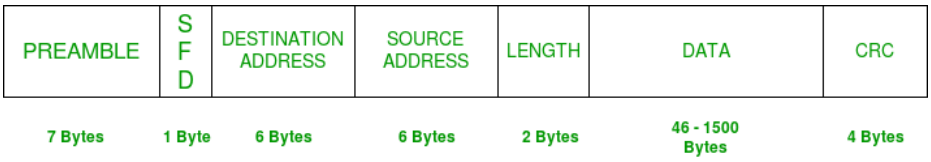


Figura 8: Trama IEEE 802.3 [14]

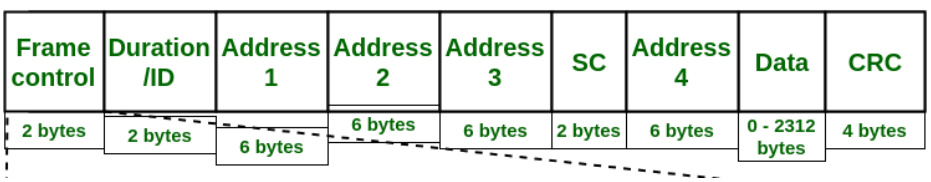


Figura 9: Trama IEEE 802.11 [15]

2 SCHEDE DI RETE

La scheda di rete è il componente elettronico che funge da interfaccia tra un calcolatore digitale e un'architettura di rete con la quale intende comunicare, inviando e ricevendo dati.
Relativamente alla realtà fisica della rete che può essere cablata oppure wireless, esistono diverse tipologie di schede.
Un importante obiettivo per i produttori delle schede di rete è la realizzazione di interfacce di rapido accesso e in grado di gestire ingenti quantità di dati. Ciò è più facilmente perseguibile nelle reti cablate, grazie al meccanismo della propagazione guidata, piuttosto che nelle reti wireless, dove un parametro aggiuntivo da tenere in considerazione è la mobilità dei terminali.

2.1 Reti cablate

Per quanto concerne le reti cablate, i principali protocolli di comunicazione sono *Ethernet*, *Token ring* e *Token bus*. Mentre i protocolli token sono obsoleti e sporadicamente continuano ad esistere in alcune reti private, il protocollo Ethernet è ampiamento usato in molteplici contesti, dalle reti domestiche a quelle aziendali.

Nelle reti cablate è presente un mezzo guidante attraverso cui il segnale elettromagnetico si propaga. Al giorno d'oggi le guide d'onda più diffuse sono i mezzi trasmissivi in rame e la fibra ottica.

I **mezzi trasmissivi in rame** sono efficaci nella trasmissione di onde elettromagnetiche di natura passabasso.
Di questa categoria fanno parte il cavo coassiale e il doppino telefonico.

Il primo è ormai in disuso nelle reti di computer ma fu utilizzato nelle prime infrastrutture Ethernet, mentre il secondo è ampiamente impiegato.
Del doppino telefonico esistono diverse versioni, relativamente alla tecnologia e quantità di rame utilizzato.
Il modello più semplice prevede due fili di rame isolati, intrecciati e avvolti da una guaina esterna in plastica (*Unshielded Twisted Pair*). Una versione successiva prevede uno schermo metallico per ognuno dei conduttori (*Shielded Twisted Pair*), mentre un'altra possiede un unico schermo che avvolge l'intera coppia intrecciata (*Foiled Twisted Pair*). Infine, possono essere presenti entrambe le schermature (*Shielded Foiled Twisted Pair*).

L'aggiunta di conduttori schermanti implica un aumento del costo del prodotto, che è tuttavia motivato dal miglior isolamento elettromagnetico del mezzo di

comunicazione nei confronti delle interferenze esterne. Infatti, ne risulta un aumento della larghezza di banda disponibile per la comunicazione.

Per quanto concerne la **fibra ottica**, essa favorisce la propagazione di onde elettromagnetiche di natura passabanda e frequenze prossime allo spettro infrarosso.
Essa è sfruttata soprattutto per la trasmissione di dati su lunghe tratte, ad esempio tra router di bordo che consentono la connettività tra diverse aree geografiche, al contrario, i cavi Ethernet sono perfettamente integrabili in reti di ridotta estensione, in particolare reti locali (LAN). Conseguentemente, i computer che costituiscono i terminali di queste reti, possiedono porte per i connettori Ethernet.

2.1.1 Evoluzione di Ethernet

Il primo abbozzo del protocollo Ethernet venne definito nel 1973 nel centro di ricerca di Palo Alto (PARC) in California. L'inventore, l'informatico Metcalfe, era intento a realizzare un sistema che permettesse la collaborazione tra computer e altri dispositivi dell'azienda, quali stampanti, sicché rappresentò la sua idea in uno sketch, che mostrò alla National Computer Conference.

Al giorno d'oggi, Ethernet è diventato lo standard per le reti locali (LAN), sviluppandosi negli anni in affidabilità ed efficienza. A sancirne il successo definitivo fu lo standard IEEE 802.3 pubblicato dall'*Institute of Electrical and Electronics Engineers* (IEEE) nel giugno 1983, che provvide a realizzarne una cornice tecnologica, rendendolo oggi standard universale.

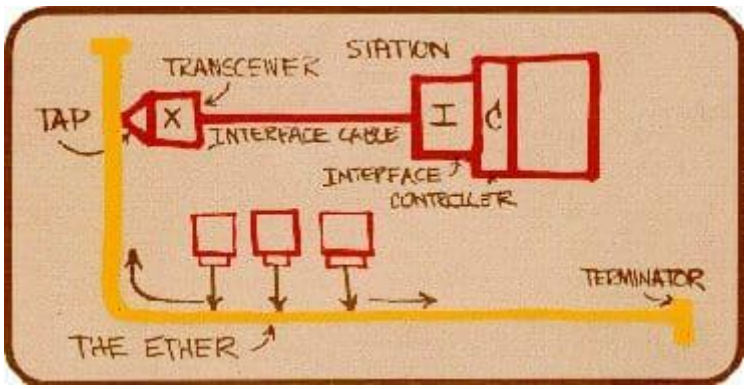


Figura 10: Primo disegno di una rete Ethernet [16]

Le prime versioni di Ethernet prevedevano l'utilizzo del cavo coassiale come mezzo di propagazione e solo successivamente venne introdotto il doppino in rame. I primi doppini non erano schermati e garantivano una velocità di trasmissione di 10 Megabit per secondo. Con l'evolversi del doppino, anche le velocità aumentarono e, al giorno d'oggi, impiegando lo *Shielded Foiled Twisted Pair* si possono di raggiungere velocità fino a 40 Gbit per secondo. Talvolta, esistono connettori ancora più performanti, capaci di ottenere velocità dell'ordine dei 100, 200, 400 Gbit/s e sono in corso studi per la realizzazione di connettori da 800 Gbit/s e 1,6 Tbit/s [16].

2.1.2 Schede di rete Ethernet

Tra i principali produttori globali di schede di rete Ethernet si annoverano Intel, Realtek, Broadcom, NVIDIA, TP-Link, Marvell, ASUS e Chelsio. Esiste un'ampia gamma di prodotti sul mercato, che spazia dalle soluzioni più semplici, che garantiscono velocità di trasmissione comprese tra le centinaia di Megabit fino a una decina di Gigabit per secondo, adatte ai computer personali e database ordinari, arrivando alle schede più performanti, adottate in ambienti orientati al machine learning (ML) e all'intelligenza artificiale (AI) che permettono scambi di dati massivi raggiungendo velocità dell'ordine delle centinaia di Gigabit per secondo.

Con riferimento ai produttori sopracitati, esempi di schede della prima categoria sono Realtek 8126 [17], Intel E610 [18], Broadcom N425G [19] , TP-Link TX401 [20], Marvell AQC115C [21] e ASUS XG100 C [22], mentre tra i prodotti avanzati si annoverano Intel E830 [18], NVIDIA ConnectX-7 [23], Broadcom P1400 GD [24], Marvell Alaska A 1.6T [25] e Chelsio T62100-CR [26].

Figura 11: Scheda di rete Ethernet [27]

2.2 Reti wireless

Mentre per le reti cablate lo standard Ethernet si è ormai imposto da anni come unica soluzione, nel campo delle reti wireless esistono molteplici protocolli adatti a diverse situazioni. Tra di essi Wi-Fi, rivolto a reti domestiche o d'ufficio, la cui evoluzione è Wimax, dedicato a realtà metropolitane, Bluetooth, utilizzato per lo scambio di dati in prossimità, in particolar modo file di piccole dimensioni, immagini e documenti. Esistono inoltre protocolli orientati all'Internet of Things, quali LoRa e ZigBee, oppure finalizzati alla connettività dei cellulari quali 4G, 5G e 6G.

Sebbene distinti, tutti questi protocolli possiedono una caratteristica comune: lo scambio di informazioni avviene sottoforma di onde elettromagnetiche di natura prettamente passabanda in propagazione libera, tramite l'ausilio di apposite antenne.
A causa della natura condivisa del mezzo radio, risulta necessario un uso consapevole dello spettro delle frequenze, occupando specifiche bande riservate o pubbliche. Le interferenze sono infatti fortemente influenti sulla qualità di servizio (QoS).

Mi limiterò a trattare le schede di rete orientate alla comunicazione Wi-Fi, essendo esso il protocollo usato nella comunicazione tra i dispositivi e i router per l'accesso a Internet. Un computer senza un'interfaccia apposita potrebbe avere limitazioni nell'uso della Rete, con gli svantaggi conseguenti.

2.1.3 Evoluzione di Wi-Fi

Con il termine Wi-Fi si indica comunemente una rete locale senza cavi (Wireless LAN).

Il primo standard tecnico, ovvero IEEE 802.11, risale al 1997 e prevedeva una velocità di comunicazione fino a 2 Mbit/s e usufruiva unicamente della banda 2.4 GHz [28]. La pubblicazione del successivo standard IEEE 802.11b, che estendeva la velocità a 11 Mbit/s, portò alla commercializzazione del primo *access point* da parte di Apple noto come *AirPort* [29]. Iniziava così la "Rivoluzione Wireless".

Figura 12: Airport, primo access point commerciale [30]

Un grande passo avanti nello standard 802.11 venne fatto con l'introduzione di una seconda banda centrata nella frequenza 5 GHz, innovazione che diede vita a *Wi-Fi 2*.

Al giorno d'oggi la versione corrente è *Wi-Fi 7* [31], rilasciata nel 2024, la quale raggiunge velocità fino a 40 Gbit/s e canalizzazioni con banda pari a 320 MHz. Inoltre, dispone di un'ulteriore banda a 6 GHz. Talvolta, possono essere sfruttate più bande contemporaneamente per una maggiore canalizzazione, in particolar modo quelle a 5 e 6 GHz. Per quanto riguarda la tecnica di modulazione, viene impiegata la 4096 QAM, ovvero una modulazione di ampiezza e fase in quadratura con 4096 possibili livelli per ogni segnale trasmesso.

2.1.4 Schede di rete Wi-Fi

Tra i più noti produttori di schede di rete Wi-Fi al mondo si annoverano Intel, Qualcomm, Broadcom, Realtek, TP-Link e ASUS .

Sul mercato esistono connettori Wi-Fi tramite porta USB, quali Netgear AXE3000 [32], TP-Link Archer TBE400UH [33] e MSI BE 6500 [34].
Ma più performanti sono le schede di rete integrabili nella scheda madre del computer con la quale comunicano mediante il protocollo *Peripheral Component Interconnect express* (PCIe). Tra le più moderne, che supportano Wi-Fi 7, si possono identificare Intel Killer Wi-Fi BE 1750 [35], Qualcomm FastConnect 7900 [36], Broadcom BCM47722 [37], Realtek RTL8922A [38].

Alcuni modelli prevedono anche un'antenna estensibile, come TP-Link Archer TBE550E [39] e ASUS PCE-BE92BT [40].

Figura 13: Scheda di rete wireless di ultima generazione [39]

2.3 Smart NICS

Le *Smart NICs* (Smart Network Interface Cards), o schede di rete intelligenti, sono evoluzioni avanzate delle classiche schede di rete (NIC), progettate per alleggerire il carico della CPU e ottimizzare le prestazioni di rete in ambienti ad alta intensità di dati, quali high performance computing, data center e cloud computing [41].

Esse possiedono dei circuiti elettronici moderni programmabili detti FPGA oppure processori dedicati al calcolo intensivo e talvolta orientati al machine learning noti come DPU. Queste tecnologie all'avanguardia permettono alla smart NIC di operare autonomamente a diversi strati di rete senza coinvolgere la CPU e con velocità elevate.

I vantaggi introdotti da queste tecnologie riguardano la programmabilità e personalizzabilità via software. Rendono possibile impostare funzionalità di gestione di firewall, bilanciamento del carico, tunneling, crittografia, supporto per ambienti virtualizzati e containerizzati e ovviamente offload delle funzionalità di rete.
Gestendo crittografia e protocolli di sicurezza a livello hardware, si riduce la vulnerabilità a minacce informatiche rispetto a soluzioni software. In particolare, controllando il flusso dei dati prima dell'ingresso in CPU, si limita il rischio di sovraccarichi malevoli del sistema.
La gestione separata e autonoma di queste funzioni aumenta la scalabilità del calcolatore ed essendo più performanti della CPU, la loro presenza si traduce in una riduzione del consumo energetico complessivo. Considerandone l'impiego in un data center, questo risparmio è notevole.

Viene di seguito fatto un breve accenno alle due tecnologie precedentemente citate in applicazioni orientate alle schede di rete e telecomunicazioni.

2.3.1 DPU

Le *Data Processing Unit* (DPU) sono acceleratori hardware specializzati nella gestione di attività di rete, archiviazione e calcolo grazie a diversi componenti chiave. Elemento vitale è il processore multicore ad alte prestazioni e programmabile, spesso basato su architetture ARM, integrato con altri componenti, quali una memoria per la gestione efficiente dei dati e hardware specializzati in attività che vanno dalla crittografia alla compressione dei dati [42]. L'esatta composizione può variare a seconda dell'applicazione e del design specifici, ma in generale è progettata per le operazioni di rete, sicurezza e archiviazione.

Sebbene una DPU possa funzionare come processore embedded indipendente, è più comunemente integrato in una Smart Nic e pertanto deve possedere un'interfaccia di rete ad alte prestazioni [43]. La loro presenza nel sistema si traduce in un miglioramento dell'efficienza energetica, della scalabilità e della sicurezza.

Per comprendere l'attualità di questa nuova tecnologia, possiamo prendere in considerazione quanto detto dal CEO di NVIDIA Jensen Huang durante un discorso risalente a maggio 2020: "Questo [una DPU] rappresenterà uno dei tre principali pilastri dell'informatica in futuro" (insieme a CPU e GPU). [44]

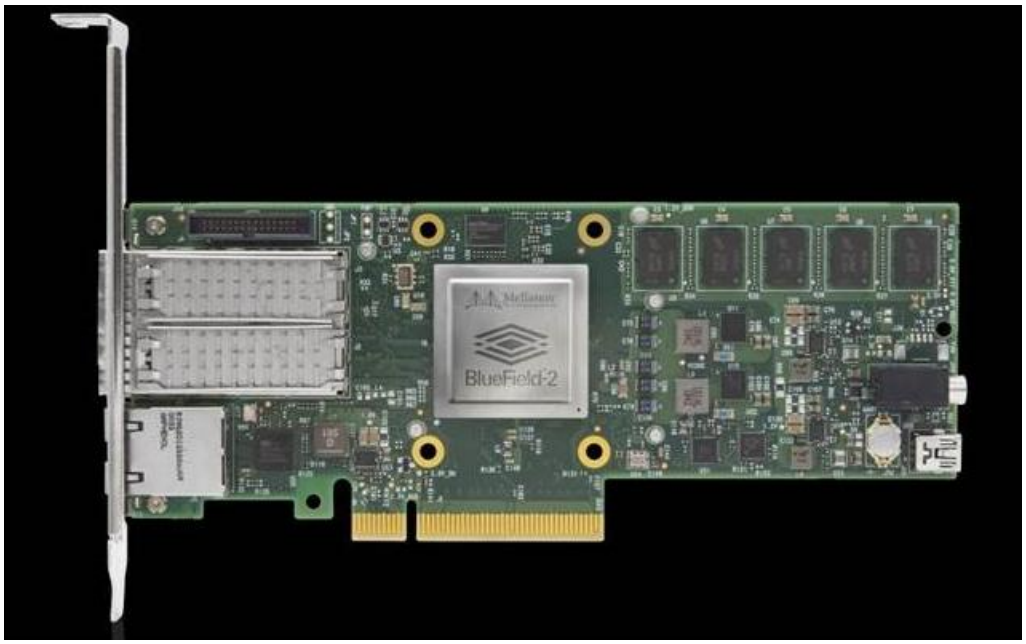


Figura 14: Scheda di rete NVIDIA dotata della DPU BlueField-2 [44]

2.3.2 FPGA

Una scheda di rete classica è realizzata come *Application Specific Integrated Circuit* (ASIC), ovvero un circuito integrato progettato su misura per un'applicazione specifica al fine di migliorarne l'efficienza. Tuttavia, la necessità di riprogrammare le funzioni di rete ha portato all'introduzione nel campo delle telecomunicazioni delle FPGA, ovvero *Field Programmable Gate Array* (FPGA). Si tratta di circuiti digitali integrati riprogrammabili via software per svolgere diversi compiti.

Essi sono costituiti da blocchi logici e interconnessioni programmabili per implementare particolari reti combinatorie e sequenziali.
Sono presenti anche blocchi di memoria, con dimensioni che spaziano dai singoli flip-flop ad array di celle molto densi, per la memorizzazione dei dati digitali all'interno del dispositivo. Facoltativamente, possono essere dotate di blocchi per il *Digital Signal Processing* (DSP) che implementano funzioni aritmetiche ad alte prestazioni.

Versatili per natura, gli FPGA sono adatti per molte applicazioni diverse e la presenza dei DSP ne permette un ampio uso nelle telecomunicazioni, facilitando operazioni onerose come il filtraggio digitale, la convoluzione, le trasformate di Fourier e i calcoli trigonometrici [45].

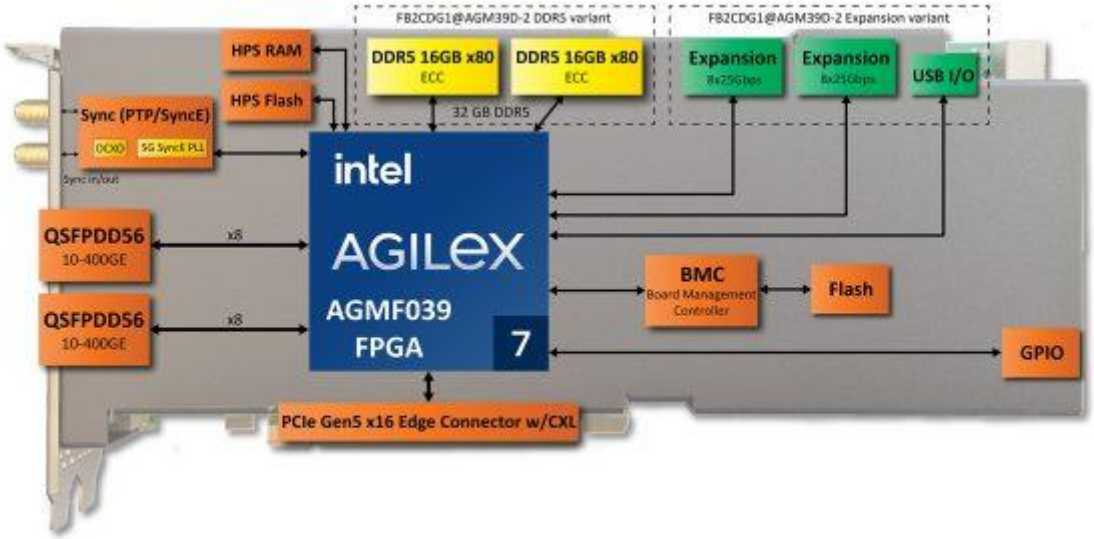


Figura 15: Schema di una Smart NIC dotata di FPGA [46]

3 FUNZIONALITÀ DELLE SCHEDE DI RETE

Affinché lo scambio di dati tra la rete e il calcolatore avvenga efficientemente sono necessarie molteplici operazioni che una scheda di rete deve effettuare sui segnali inviati e ricevuti. L'obiettivo da perseguire è un'ottimale qualità di servizio.

Queste funzioni operano a diversi livelli della comunicazione e per comodità possiamo dividerle nei due gruppi, non nettamente separati, di operazioni generali e di offload.

Per quanto riguarda le operazioni generali ai annoverano:
Codifica di canale, multiplazione, strutturazione delle trame e controllo degli errori, controllo del flusso, accesso al mezzo condiviso, risparmio energetico, Wake on LAN, indirizzamento tramite MAC, supporto per reti virtuali e gestione delle priorità.

Per quanto riguarda quelle avanzate: Internet checksum, Large Send e Large Receive offload, gestione dei Jumbo Frame, Direct Memory Access, ARP offload, TLS offload, Receive e Transmit Side Scaling, Sigle Root Input/Output Virtualization.

3.1 FUNZIONI GENERALI

3.1.1 Codifica di canale

Un contenuto digitale, per poter essere comunicato a un ricevitore, deve essere trasferito fisicamente in forma analogica. La responsabilità di ciò ricade sulla scheda di rete che deve effettuare efficientemente la conversione tramite codifica di linea, specifica per il canale di comunicazione.

Viene di seguito riportato il modello a blocchi di un sistema di comunicazione numerico passabasso per comprendere il funzionamento della codifica di linea all'interno della comunicazione.

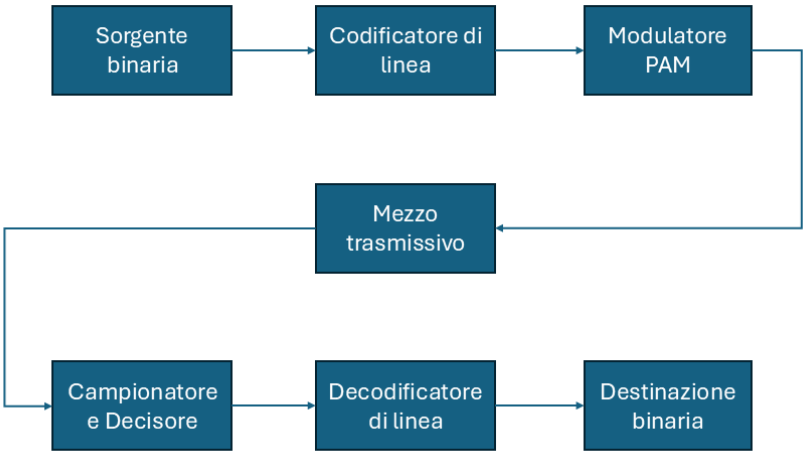


Figura 16: Modello semplificato di una comunicazione passa-basso digitale

Inoltre, potrebbe essere presente un modulatore a portante sinusoidale per ottenere segnali passabanda. Esso sarebbe utile nel caso in cui si desiderassero utilizzare diverse bande separate di comunicazione.

Due codifiche, relativamente semplici, sono *Non Return Zero* (NRZ) e *Manchester*. Nella prima gli 0 e 1 logici sono modulati con impulsi rettangolari di durata pari all'intero tempo di bit, mentre nella seconda ogni bit è rappresentato come una transizione: lo "0" logico diventa una transizione basso → alto, mentre l' "1" logico una variazione alto → basso. Questa codifica richiede una bit-rate doppia rispetto alla codifica NRZ ma semplifica lato ricevitore la sincronizzazione con il segnale ricevuto.

Per questo motivo Manchester fu effettivamente utilizzata nei primi standard Ethernet, in particolare 10-BASE T, anche se al giorno d'oggi è in disuso in favore di codifiche più efficienti, quale *PAM-16* [47], ovvero una modulazione di ampiezza a 16 livelli sfruttata nei protocolli Ethernet ad alta velocità, come

10GBASE-T (Ethernet a 10 Gigabit su rame). Mediante i sedici livelli è possibile ottenere ben quattro bit per simbolo.

Di seguito viene riportata una rappresentazione tramite Matlab del funzionamento delle due codifiche NRZ e Manchester avendo considerato la sequenza di bit :

b = [0,1,0,1,0,1,1,0,0,0,1,0,1,1,0,0,1,0,1,0,1,0,0,1,1]

e una bit-rate pari a 1kHz.

Quindi, ogni bit si tradurrà in un segnale di durata 1 ms.

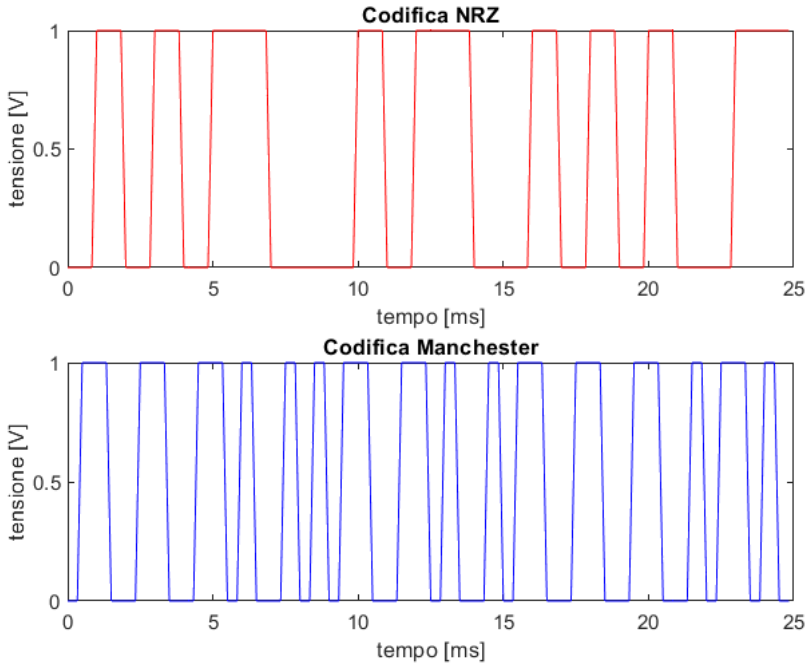


Figura 17: Plot delle codifiche di linea realizzati con Matlab

3.1.2 Multiplazione

In molte occasioni un dispositivo deve comunicare contemporaneamente con diversi interlocutori, pertanto, può essere conveniente implementare molteplici canali di comunicazione su un unico mezzo trasmissivo, gestendo i diversi flussi per evitare conflitti.

Questa funzione prende il nome di Multiplazione e può essere gestita in quattro domini: tempo, frequenza, codice e spazio.

Nella *Multiplazione a Divisione di Frequenza* (FDM) i canali sfruttano bande in frequenza distinte. Una versione notevole usata nelle fibre ottiche è la *Wave*

Division Multiplexing (WDM), ovvero la multiplazione a divisione di lunghezza d'onda.

Nella *Multiplazione a Divisione di Tempo* (TDM) la comunicazione sul canale viene gestita nel tempo, per cui ogni dispositivo può trasmettere in specifici intervalli temporali. Questi intervalli possono essere divisi in slot di durata fissa, essere assegnati staticamente oppure dinamicamente e prevedere una organizzazione in frame.

Nella *Multiplazione a Divisione di Codice* (CDM) i flussi di dati sfruttano codici univoci e distinti, ortogonali e pseudo-casuali, che permettono di trasmettere contemporaneamente sulla stessa banda di frequenze. Il ricevitore, infatti, è in grado di decodificare il segnale corretto, sulla base del codice, ignorando gli altri.

Nella *Multiplazione a Divisione di Spazio*, vengono usati canali fisici separati spazialmente. Ad esempio, nelle fibre multicore vengono inseriti molteplici core all'interno di un unico cladding.

3.1.3 Strutturazione delle trame e controllo degli errori

I dati da inviare devono essere organizzati in trame che siano facilmente gestibili nella comunicazione. In particolare, i segmenti prodotti dal TCP del trasmettitore devono essere aggiornati per il dialogo link-by-link con l'aggiunta di intestazioni di Collegamento che consentano le funzioni di **sincronizzazione**, **numerazione** e **delimitazione** delle trame, mediante opportuni campi, come mostrato nelle figure 8 e 9.

In realtà, questa funzione potrebbe essere eseguita dalla scheda di rete, con il nome di *Large Send Offload*, a cui si fa riferimento nella sezione successiva. Tuttavia, è comodo trattarla in questo momento per rendere comprensibile il controllo degli errori.

Con riferimento allo standard 802.3 di Ethernet, affinché il ricevitore riconosca l'inizio di un pacchetto e possa correttamente campionare il segnale in ingresso, è necessaria una sequenza di bit di sincronizzazione detta *preambolo di sincronismo*, costituita da "1" e "0" alternati. Sono importanti anche i campi *Start Of Frame* (SOF) e *Length*, che permettono di identificare i bit di informazione nella trama.

Per quanto riguarda il campo CRC, esso è necessario per il controllo degli errori. Infatti, l'algoritmo utilizzato è il *Cyclic Redundancy Check* (CRC – 32), ovvero un codice a blocco sistematico polinomiale che sfrutta un polinomio

generatore di 32 bit. Il polinomio attualmente utilizzato dallo standard IEEE 802.3 è, in formato esadecimale, 0x04C11DB7 [48].

Il funzionamento dell'algoritmo è il seguente: il trasmettitore calcola il resto della divisione tra la sequenza di bit che intende inviare e il polinomio. Quindi shifta la sequenza a sinistra di 32 bit per concatenarle il resto della divisione. Quando la sequenza verrà ricevuta dal destinatario, esso effettuerà la divisione per il polinomio generatore e, se otterrà un resto nullo, assumerà il contenuto come corretto, al contrario come errato. Questo algoritmo non è infallibile, infatti esistono casi in cui il controllo fallisce: se l'errore è un multiplo del polinomio generatore, il ricevitore calcolerà un resto nullo e non rileverà il problema.

3.1.4 Controllo del flusso

Le schede di rete possono gestire anche il controllo del flusso link-by-link tipico del livello di Collegamento. Il protocollo Wi-Fi possiede il *Sequence Control* (SC), costituito da due sottocampi, ovvero il *Sequence Number* di 12 bit e il *Fragment Number* di 4.

Il controllo è implementato secondo le modalità tipiche del protocollo *Automatic Repeat reQuest* (ARQ), per cui una trama non riscontrata deve essere rinviata.

Talvolta, le schede di rete possiedono dei buffer di memoria per mantenere i pacchetti in attesa di elaborazione e gestire le priorità. La versione 802.1Q di Ethernet [49], infatti, possiede un campo *Priority*.

Talvolta, in Ethernet 802.3x, il ricevitore può completamente sospendere la trasmissione inviando al mittente un messaggio di *pause frame*, che coincide nell'ARQ alla proposta di una Advertised Window nulla.

3.1.5 Accesso al mezzo condiviso

Nel caso in cui il mezzo trasmissivo sia condiviso, diventa necessario un accesso consapevole per evitare conflitti con altri usufruitori. A causa della diversità delle infrastrutture, sono stati definiti due protocolli di accesso multiplo casuale con rilevazione del canale noti come *Carrier Sense Multiple Access* (CSMA).

Nelle reti wireless è sfruttato il protocollo *Collision Avoidance* (CSMA - CA), mentre nelle reti cablate half-duplex è utilizzato il protocollo *Collision Detection* (CSMA – CD). Dal momento che il protocollo Ethernet è full-duplex e dispone di due mezzi separati per la comunicazione tra i due dispositivi, le collisioni non possono verificarsi e quindi non necessita di controllo. Diverso è per le reti che

possiedono, ad esempio, un cavo coassiale condiviso, che necessitano del protocollo [50].

Per quanto concerne il *Collision Detection*, esso mira al rilevamento delle collisioni: il trasmettitore, mentre invia la trama, rimane in ascolto sul canale e, nel caso in cui rilevi un segnale differente dal proprio, comprende l'avvenuta collisione e provvede al rinvio dei dati.

Riguardo al *Collision Avoidance*, esso è finalizzato ad evitare le collisioni, pertanto, prima di iniziare una trasmissione, avviene la "prenotazione del canale" mediante la quale il trasmettitore avverte tutti gli altri nodi della rete della propria intenzione di trasmettere. Il metodo prende anche il nome di RTS/CTS in quanto, in primo luogo, il trasmettitore richiede di trasmettere (*Request To Send*) e successivamente il ricevitore accetta la richiesta avvisando tutti i nodi ad esso connessi (*Clear To Send*).

3.1.6 Risparmio energetico e Wake on LAN

Per rispondere all'attuale necessità di riduzione del consumo energetico di un host, specialmente nei momenti di scarsa attività, la maggior parte delle schede di rete implementa la *Energy-Efficient Ethernet* (EEE) [51], meglio conosciuta come *Green Ethernet*. Questo meccanismo consiste in una riduzione o totale rimozione dell'alimentazione, mantenendo però attiva la scheda di rete nell'evenienza in cui qualcuno volesse comunicare.

Spesso questa funzione è sfruttata in concomitanza con la *Wake on LAN* (WoL), anche nota come *Wake on Magic Packet* che permette l'accensione di un dispositivo a distanza, mediante un *magic packet* [52]. Ricevuto questo pacchetto, la scheda di rete invierà un segnale alla scheda madre per attivare il resto del calcolatore.

Generalmente, il pacchetto magico contiene una sequenza di bit di sincronizzazione seguita dalla ripetizione dell'indirizzo MAC della scheda di rete del computer che si intende "svegliare".

Queste due tecnologie possono essere utilizzate in concomitanza perché, nel caso in cui un dispositivo sia sospeso per risparmiare energia, deve comunque poter rispondere a una richiesta di comunicazione imprevista.
A tal proposito, all'interno della rete LAN di cui fa parte, deve essere presente un nodo a conoscenza del suo indirizzo MAC, a cui può inviare il pacchetto magico per riattivarlo.

3.1.7 Indirizzamento MAC

Ogni scheda di rete deve poter essere identificabile univocamente all'interno di qualunque rete fisica essa possa fare parte tramite un identificativo MAC (*Medium Access Control*) assegnatole univocamente dal produttore e che talvolta può essere modificato via software per scopi di privacy.
Si tratta di un codice di 48 bit di cui i primi 24 identificano il produttore della scheda e i successivi 24 il numero di serie nella produzione.

Quando un pacchetto arriva a una scheda di rete, essa verifica che l'indirizzo MAC di destinazione coincida con il proprio e in tal caso mantiene il pacchetto, in caso contrario lo scarta. Talvolta può essere attiva una modalità promiscua, per cui la scheda intercetta tutto il traffico di rete compiendo uno *sniffing*.

3.1.8 Gestione delle reti virtuali e delle priorità

Talvolta, per dividere i domini di broadcast all'interno di un'architettura di rete, possono essere istituite delle reti virtuali (VLAN), che condividono il Router, il quale però gestisce la comunicazione con Internet e le due reti separandone i domini.

Per supportare questa funzione, è stato emanato lo standard IEEE 802.1Q che prevede una modifica nell'intestazione di Ethernet con l'aggiunta di campi specifici: il *VLAN identifier*, in particolare, permette di identificare la rete virtuale a cui fa riferimento il pacchetto.

Sempre a proposito di questa versione di Ethernet, è presente un campo *Priority*, già descritto nel paragrafo del controllo di flusso, molto utile per gestire la priorità di più flussi di dati.

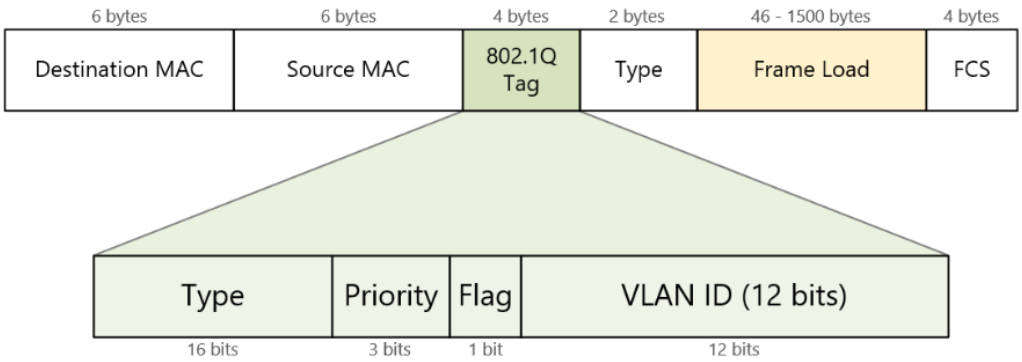


Figura 18: Formato trama IEEE 802.1Q [49]

3.2 FUNZIONALITÀ DI OFFLOADING

Come preannunciato tante volte in precedenza, le schede di rete possono svolgere delle funzionalità che normalmente verrebbero computate a livello software, ottimizzando il funzionamento del sistema.

Di seguito viene riportato un semplice disegno che rappresenta il funzionamento di una scheda di rete "tradizionale", dotata solamente delle funzionalità dei livelli bassi quali Collegamento e Fisico, contro una "avanzata" fornita di funzionalità di offload [53].

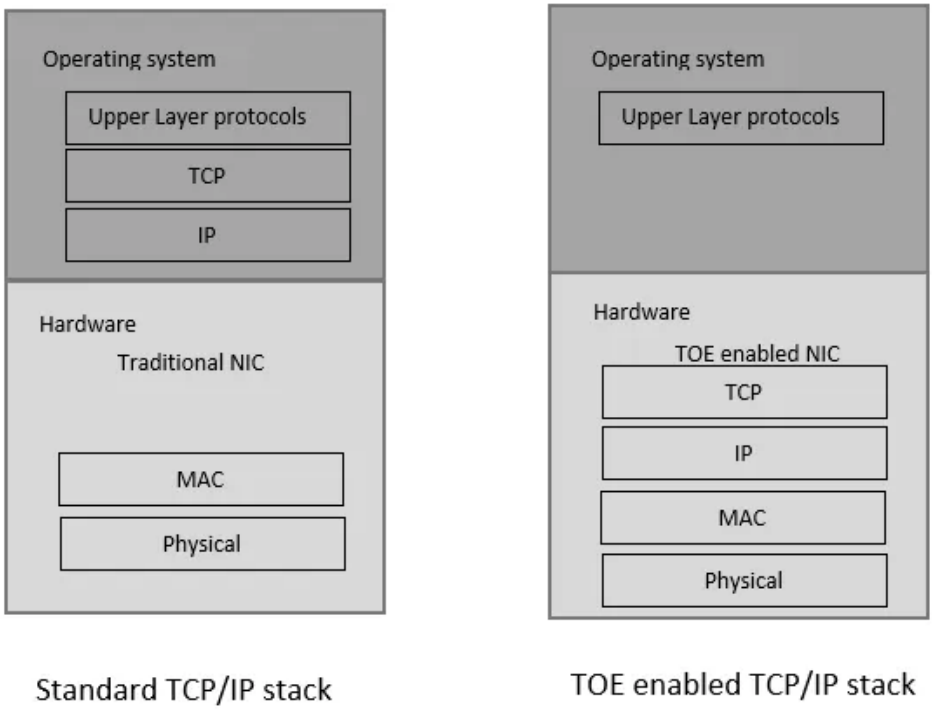


Figura 19: Gestione del TCP/IP tradizionale e mediante offload [53]

Spesso il set di funzioni presenti nelle schede di rete viene indicato con il nome *TCP Offload Engine* (TOE), simboleggiante un "motore di scarico" delle funzioni dello strato TCP (e IP).

3.2.1 Storia

Per risalire all'origine delle funzionalità di offload è necessario tornare al 1990, quando *Auspex Systems* pubblicò uno dei primi brevetti in questo campo, rivolto all'offload delle funzionalità del protocollo di Trasporto UDP.
In seguito, constatata l'enorme diffusione di TCP, queste funzionalità vennero estese anche ad esso, dando vita al *TCP chimney offload* [54].

Inizialmente, molti ritennero svantaggiosa l'elaborazione delle operazioni in offload sostenendo che l'esecuzione centralizzata da parte della CPU fosse migliore, specialmente in termini di gestione delle operazioni e supervisione dei dati. Tuttavia, con il passare degli anni, molti produttori delle schede di rete iniziarono a commercializzare prodotti sempre più dotati di queste funzioni, avendo sperimentato e simulato i vantaggi reali. Infatti, l'efficacia di queste tecnologie è apparsa sempre più evidente e al giorno d'oggi, tutte le schede di ultima generazione ne dispongono ampiamente.

3.2.2 Vantaggi e svantaggi

Per parlare della reale efficacia di una scheda di rete in una comunicazione bisogna identificare a priori il collo di bottiglia della rete, per cui una scheda molto avanzata potrebbe essere molto sottoutilizzata in una rete antiquata, tanto quanto una scheda di rete obsoleta potrebbe essere un importante punto di congestione in una rete prestante. Inoltre, è necessario definire lo scopo del calcolatore di cui essa fa parte: nel caso in cui ricevesse enormi quantità di dati dalla rete e dovesse nel frattempo eseguire altre importanti operazioni, la scheda diventerebbe fondamentale.

Un vantaggio evidente risiede nella gestione della scheda da parte della CPU. Essa è controllata tramite "interruzioni", o *interrupt*, per cui il trasferimento dei dati tra i buffer della scheda e i registri centrali avviene sospendendo l'attività del calcolatore. Ciò si traduce in continui salti a routine di gestione della periferica, per poi tornare al programma in attesa.

Se i calcoli venissero delegati alla scheda, le routine di gestione diventerebbero più semplici e meno frequenti. Conseguentemente, anche un risparmio energetico sarebbe introdotto.

Qualcuno potrebbe avanzare l'idea che esistano anche degli svantaggi legati a queste tecnologie, in particolare associati al prezzo maggiorato delle schede avanzate e problemi di compatibilità con i sistemi operativi che potrebbero non supportare correttamente le funzioni di offload.
A tal proposito, è necessario specificare che schede molto prestanti comportano un vantaggio in applicazioni mirate, nelle quali grandi quantità di dati arrivano al calcolatore e in questo contesto il risparmio di tempo e di energia sono evidenti. Pertanto, il loro utilizzo deve essere ponderato all'applicazione del dispositivo, il quale dovrà possedere un sistema operativo opportuno.

Un'altra critica afferma che ci sono delle falle di sicurezza e che la gestione software permetta una supervisione maggiore sui messaggi malevoli. Su questo argomento tuttavia vale il contrario, infatti, come precedentemente spiegato per le DPU, la gestione dei flussi a monte dell'introduzione nella CPU permette di bloccare attacchi malevoli senza causare l'interruzione del server che altrimenti vedrebbe il processore essere saturato dai dati di rete.

Ad ogni modo, esiste un valido compromesso tra le modalità software e offload, ovvero il *TCP partial offload*, secondo il quale il sistema operativo gestisce la comunicazione nelle fasi di instaurazione e chiusura della connessione tra i terminali, delegando poi le attività della fase di scambio dati.

Nei paragrafi successivi saranno elencate le principali funzionalità di offload.

3.2.3 Internet Checksum

L'Internet Checksum è un codice a blocco sistematico usato dai protocolli di Internet TCP e IP per il controllo dell'errore. Esso è fondamentale per garantire una corretta condivisione dei dati ma comporta dei calcoli sia da parte del trasmettitore, sia del ricevitore.

Un *codice a blocco* è un codice che definisce dei bit di controllo dell'errore come funzione combinatoria dei bit di informazione. Un codice a blocco si dice *sistematico* se, all'interno del segmento inviato, i bit di controllo sono tenuti separati dai bit di informazione.

Algoritmo

L'algoritmo, definito dall'*IETF*, si sviluppa in tre fasi:

In outline, the Internet checksum algorithm is very simple:

(1) Adjacent octets to be checksummed are paired to form 16-bit integers, and the 1's complement sum of these 16-bit integers is formed.

(2) To generate a checksum, the checksum field itself is cleared, the 16-bit 1's complement sum is computed over the octets concerned, and the 1's complement of this sum is placed in the checksum field.

(3) To check a checksum, the 1's complement sum is computed over the same set of octets, including the checksum field. If the result is all 1 bits (-0 in 1's complement arithmetic), the check succeeds.

Figura 20: Algoritmo Internet Checksum [55]

Il **trasmettitore**, che possiede il segmento da trasferire, deve:

1) Dividere il contenuto in ottetti e accoppiarli in righe da 16 bit;
2) Calcolare la somma complemento a 1 delle righe: esse vengono sommate e il riporto finale aggiunto all'inizio del risultato, quindi invertiti zeri e uni.
   Inserire il risultato nel campo riservato alla Checksum dell'intestazione TCP/IP;

Il **ricevitore** deve:

3) Controllare che il segmento sia corretto, quindi ricalcolare la somma complemento a 1 sulla sequenza di bit includendo anche il campo Checksum. Se il risultato è una sequenza completa di uni allora il segmento non possiede, apparentemente, errori.

1. Proprietà

L'algoritmo gode di alcune proprietà matematiche:

1) Proprietà commutativa:
   [A,B] + [C,D] = [C,D] + [A,B]

2) Proprietà associativa:
   ([A,B] + [C,D]) + [E,F] = [A,B] + ([C,D] + [E,F])

Esse sono fondamentali perché rendono il calcolo indipendente dalla rappresentazione dei dati che potrebbe essere sia di tipo Big-endian sia Little-endian: se [A,B] + [C,D] = [X,Y] , allora [C,D] + [A,B] = [Y,X]

3) Se l'architettura del calcolatore ha un parallelismo multiplo di 16 bit, si può effettuare una somma in parallelo, che rende il calcolo più efficiente. Ad esempio, in una architettura a 32 bit, si può effettuare una somma a 32 bit e poi "ripiegare" a un risultato a 16 bit.

4) Proprietà dell'aggiornamento incrementale, per la quale, dato [A,B] + [C,D] = [X,Y], se [A,B] viene modificata in [A',B'], allora si può ricalcolare la nuova somma come [X',Y'] = [X,Y] – ([A',B'] – [A,B]).
   Quest'ultima è una proprietà molto utile perché permette, di fronte a una modifica di qualche bit del contenuto del segmento, di non dover ricalcolare completamente l'intero risultato [56].

2. Esempio

Viene mostrato di seguito un esempio di utilizzo della Checksum, sia dal punto di vista del trasmettitore che del ricevitore. Consideriamo una breve sequenza di bit, che non ha a che vedere con le reali dimensioni delle trame di Internet:

- Sequenza che deve inviare il trasmettitore:

00000000 00000001 11110010 00000011 11110100 11110101 11110110 11110111

Dati generati		0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	1
		1	1	1	1	0	0	1	0		0	0	0	0	0	0	1	1
		1	1	1	1	0	1	0	0		1	1	1	1	0	1	0	1
		1	1	1	1	0	1	1	0		1	1	1	1	0	1	1	1
Riporti	10	0	0	1	0	1	1	0		1	1	1	1	1	0	0	0	
Somma		1	1	0	1	1	1	0	0		1	1	1	1	0	0	0	0
									1								1	0
Complemento a 1		1	1	0	1	1	1	0	1		1	1	1	1	0	0	1	0
Da trasmettere		0	0	1	0	0	0	1	0		0	0	0	0	1	1	0	1

- Sequenza giunta al ricevitore:

  00000000 00000001 11110010 00000011 11100100 11110101 11110110 11110111 00100010 00001101

In cui è presente un errore.

Dati generati		0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	1
		1	1	1	1	0	0	1	0		0	0	0	0	0	0	1	1
		1	1	1	0	0	1	0	0		1	1	1	1	0	1	0	1
		1	1	1	1	0	1	1	0		1	1	1	1	0	1	1	1
		0	0	1	0	0	0	1	0		0	0	0	0	1	1	0	1
Riporti	10	0	0	1	0	1	1	0		1	1	1	1	1	0	0	0	
Somma		1	1	1	0	1	1	1	0		1	1	1	1	1	1	0	1
									1								1	0
Complemento a 1		1	1	1	0	1	1	1	1		1	1	1	1	1	1	1	1
Esito	NEGATIVO																	

In questa occasione il controllo è riuscito a rilevare l'errore, ma esiste il caso in cui la Checksum fallisce: se coesistono un numero pari di errori su un'unica colonna l'algoritmo non li rileverà, lasciando intendere che il pacchetto sia corretto.

La Checksum viene sempre computata a livello di Trasporto e può talvolta essere eseguita anche a livello di Rete, dipendentemente dalla versione del protocollo IP adottato.

Checksum TCP/UDP

A livello di Trasporto la Checksum viene eseguita su tutto il segmento, contenente l'intestazione TCP/UDP, i dati e lo pseudo-header IP, ovvero una porzione di intestazione IP.

Lo pseudo-header IP è differente nelle due versioni IPv4 e IPv6: la prima contiene 96 bit, la seconda ben 320.

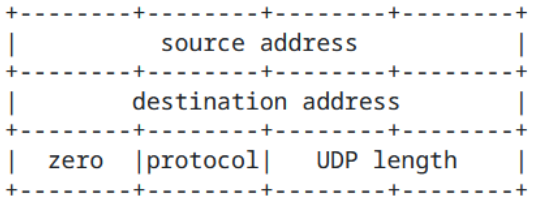


Figura 21: Pseudo-header IPv4 [8]

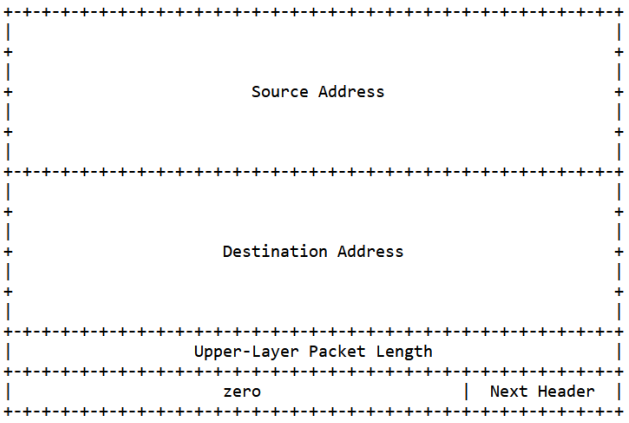


Figura 22: Pseudo-header IPv6 [13]

Per quanto riguarda le intestazioni TCP e UDP [9] [10], il primo possiede 256 bit, mentre il secondo 128 bit.

Infine, a questi contributi, devono essere sommati i bit di informazione, in quantità variabili a seconda della dimensione del segmento, che talvolta potrebbe essere un Jumbo gramma, di cui si parlerà successivamente.

Il contributo dei dati è generalmente preponderante.

Checksum IP

La Checksum può essere calcolata link-by-link a livello di Rete, a seconda della versione del protocollo e dell'implementazione dei router intermedi che possono non essere in grado di compiere questo calcolo.

La versione IPv4 è dotata di un campo Checksum, ed essa viene calcolata unicamente sull'intestazione IP, lunga 196 bit. Quindi, il calcolo relativo è significativamente inferiore a quello del protocollo TCP.

Invece, il protocollo IPv6 ha eliminato questo passaggio, come si evince dall'intestazione stessa, in cui non compare nessun campo relativo.

3.2.4 Large Send Offload (LSO) e Large Receive Offload (LRO)

Il *Large Send Offload*, anche conosciuto come *TCP Segmentation Offload* (TSO), permette alla scheda di rete di "smembrare" i dati provenienti dal processo applicativo in opportune trame adatte al trasporto fino al destinatario, svincolando quindi la CPU da questo compito. Essa, infatti, può limitarsi a inviare i dati alla scheda nel momento esatto in cui sono forniti dal processo applicativo, disinteressandosi delle problematiche dei livelli inferiori [57].
A questa funzione è stato fatto riferimento nella sezione precedente, per quanto riguardava la strutturazione delle trame di Collegamento.

Talvolta, se attivata la relativa impostazione, è possibile creare trame di dimensione superiore rispetto alla Maximum Transmission Unit dello standard di Collegamento, Ethernet specialmente, creando dei Jumbo frame.

Una funzione analoga esiste per il protocollo UDP, la quale prende il nome di *UDP Fragmentation Offload* (UFO).

Esiste la possibilità di avere una gestione software di queste operazioni, generalmente sconveniente. Alcuni sistemi operativi permettono di disattivare questa impostazione e in Linux, per esempio, l'utente è libero di preferirgli la *Generic Segmentation Offload* (GSO), nella quale le operazioni di segmentazione sono eseguite via software, prima che i dati siano trasmessi alla scheda di rete. Ciò può rivelarsi utile nel caso in cui la scheda sia poco performante e si preferisca utilizzare la CPU.

Per quanto riguarda il ricevitore, esistono funzioni analoghe di riaggregazione dei pacchetti. La funzione di offload analoga a LSO prende il nome di *Large Receive Offload* (LRO) [58]. Il principale vantaggio che ne deriva è la moderazione degli interrupt di CPU, dal momento che essi verranno prodotti solamente una volta riassemblato il segmento di Trasporto completo e pertanto saranno meno frequenti.

Come LSO, anche per LRO esiste una variante software, il *Generic Receive Offload* (GRO).

3.2.5 Gestione dei Jumbo frames

Quando si parla di *Jumbo frame* si intende un concetto distinto dal *Jumbo gramma*, sebbene la somiglianza dei nomi.

Un Jumbo gramma è un pacchetto IP di dimensioni eccedenti quella massima definita dal protocollo IPv4, pari a 65.536 byte. La versione IPv6, infatti, ha introdotto un campo facoltativo *Jumbo Payload* che estende la lunghezza massima a 4 Gigabyte.
I Jumbo grammi possiedono una grave problematica, nonché motivo del loro quasi totale inutilizzo, essi sono trattati come singoli segmenti TCP e conseguentemente la Checksum è eseguita su tutta la loro estensione, diventando estremamente inefficiente.

Un Jumbo frame, invece, è una trama Ethernet più estesa del Maximum Transmission Unit (MTU) stabilito dal protocollo di Collegamento.
La dimensione massima predefinita di una trama Ethernet è di 1500 byte di dati informativi, ma l'utente può optare per la generazione di trame con payload superiori, fino a valori nell'intorno dei 9000 bytes [59].

3.2.6 Direct Memory Access (DMA)

Se i pacchetti provenienti dalla scheda di rete contengono informazioni relative a processi applicativi in esecuzione, residenti in memoria principale, questa impostazione diventa molto utile, permettendo il trasferimento dei dati dai buffer della scheda direttamente nella RAM, senza l'intermediazione della CPU. Normalmente, infatti, i dati dovrebbero transitare attraverso i registri, previa essere immagazzinati in memoria. L'accesso diretto permette di evitare interruzioni della CPU, riducendo latenza e migliorando l'efficienza.

Il trasferimento avviene seguendo un flusso di operazioni stabilite che prevedono la mediazione di un *Controller DMA* il quale gestisce lo spostamento in modo autonomo [60]. Innanzitutto, una periferia che intende accedere alla memoria per registrare o leggere dei dati, invia una richiesta di accesso al bus dati al Controllore DMA. Esso monitora il traffico sul bus e, nel caso in cui sia libero, ne prende il controllo ed effettua il trasferimento tra i buffer della periferica e la memoria. Una volta completato, ne dà notifica alla CPU mediante un interrupt, quindi rilascia il controllo del bus.

Questa tecnologia, prima dell'avvento nelle schede di rete, era già diffusa nelle *Graphic Processing Units* (GPU) [61], dove permette di trasferire i dati dai buffer della periferica in specifiche zone della RAM. Solo in seguito al trasferimento viene notificata alla CPU la disponibilità dei dati.

3.2.7 ARP offload

L'*Address Resolution Protocol* (ARP) è un protocollo collocabile a cavallo tra lo strato di Rete e lo strato di Collegamento, la cui funzione è quella di tradurre gli indirizzi IP degli host nei rispettivi indirizzi MAC [62].
Questa operazione risulta fondamentale nel momento in cui sia in corso il trasferimento di un pacchetto tra due nodi e il mittente non conosca l'indirizzo MAC da inserire nell'intestazione di livello di Collegamento. Ciò avviene frequentemente, specialmente alla creazione di una nuova rete LAN o all'accesso di un nuovo dispositivo in una preesistente, della quale non conosce gli altri partecipanti. Infatti, nella fase di indirizzamento, la decisione del *next hop* avviene sulla base delle informazioni contenute nella tabella di Routing, che sfrutta gli indirizzi IP. Quindi, è necessario determinare l'indirizzo MAC del destinatario a partire da quello IP.

Generalmente, i nodi possiedono dei buffer che memorizzano le correlazioni tra i due indirizzi, sicché le operazioni di ARP possono essere eseguite meno frequentemente.

Questo offload è particolarmente utile in quei dispositivi in cui è in esecuzione un'applicazione che richiede una perenne consapevolezza degli host raggiungibili e pertanto vengono ripetute periodicamente operazioni ARP [63].

3.2.8 TLS offload

Il *Transport Layer Security* (TLS) [64] è un protocollo crittografico di livello di Trasporto che permette l'instaurazione di comunicazioni sicure tra host in concomitanza con *Hyper Text Transfer Protocol Secure* (HTTPS).
Nella prima fase della comunicazione tra terminali si verifica l'*hand-shake TLS*, in cui avviene l'autenticazione e si stabiliscono i parametri e le chiavi per la crittografia. Questa procedura è abbastanza onerosa in termini di risorse, pertanto solamente schede di rete molto performanti la implementano.

3.2.9 Receive Side Scaling (RSS) e Transmit Side Scaling (TSS)

Il *Transmit Side Scaling* e il *Receive Side Scaling*, ovvero il ridimensionamento lato trasmettitore e ricevitore, sono due funzioni che permettono di distribuire l'elaborazione dei pacchetti inviati o ricevuti su più cores di una CPU [65].
Nella maggior parte delle architetture moderne sono presenti processori multicore, pertanto, risulta ragionevole adoperarli contemporaneamente eseguendo operazioni parallele.

In Windows, la distribuzione dei pacchetti è gestita dal lavoro combinato di scheda di rete e driver miniport.

Un *driver miniport* è un'interfaccia software che permette alla scheda di rete di comunicare con entità di livello superiore [66]. Quando un pacchetto è pronto per essere inviato alla CPU, la scheda di rete gli associa un *hash*, ovvero una stringa che identifica un core. Quindi il driver miniport instrada il pacchetto nella coda del core stabilito.

Il RSS può essere usato in combinazione con l'*accelerated Receive Flow Steering* (aRFS) [67], che permette di ottimizzare ulteriormente l'utilizzo dei core delle CPU. Infatti, l'elaborazione dei pacchetti relativi a una connessione potrebbe essere in esecuzione su uno specifico core e pertanto risulterebbe conveniente continuare a instradare ad esso i pacchetti relativi a quel flusso. Se presente, RFS permette di tenere traccia dei processi sui core in modo da indirizzare i pacchetti efficacemente, risultando in un ottimizzazione dell'accesso alla cache della CPU, migliorando la latenza complessiva.

Per quanto riguarda il *Transmit Side Scaling*, esso è l'analoga implementazione lato trasmettitore, che gestisce la produzione contemporanea su più core di pacchetti da inviare all'interfaccia di rete.

3.2.10 Single Root Input/Output Virtualization SRIOV

Il *Single Root Input/Output Virtualization* è una tecnica di virtualizzazione delle schede di rete facente parte delle funzioni di virtualizzazione dei calcolatori.

Quando si parla di virtualizzazione si intendono tecnologie che permettono la creazione di istanze virtuali su un'unica entità fisica per ottenere dei benefici tecnologici.

I componenti chiave della virtualizzazione sono la Physical Machine (PM), ovvero il server/computer fisico, le Virtual Machines (VMs), cioè le istanze virtuali, e l'Hypervisor, ovvero il gestore delle relazioni tra le VMs e la PM. Esso coordina le attività in esecuzione sulle VMs e funge da interfaccia tra esse e la macchina fisica, permettendo accessi alle risorse senza interferenze reciproche.

La virtualizzazione comporta dei benefici, quali una maggiore efficienza nell'uso delle risorse, gestione più semplice e minori tempi di inattività degli host. Infine, la possibilità di spostare le macchine virtuali da un calcolatore a un altro permette, nell'evenienza di malfunzionamenti, di traslare l'operatività senza interdire il funzionamento del server [68].

Per quanto riguarda le schede di rete, esistono modelli per i quali è possibile gestire la virtualizzazione in offload e ciò prende un nome specifico, ovvero *Single Root Input/Output Virtualization*. Normalmente, infatti, verrebbero prodotte interfacce virtuali della scheda, simulanti l'esistenza di altrettante istanze fisiche. Questo modello comporta il consumo di notevoli risorse sulla macchina host per gestire il traffico sulle interfacce virtuali, con conseguente aumento del carico del server.

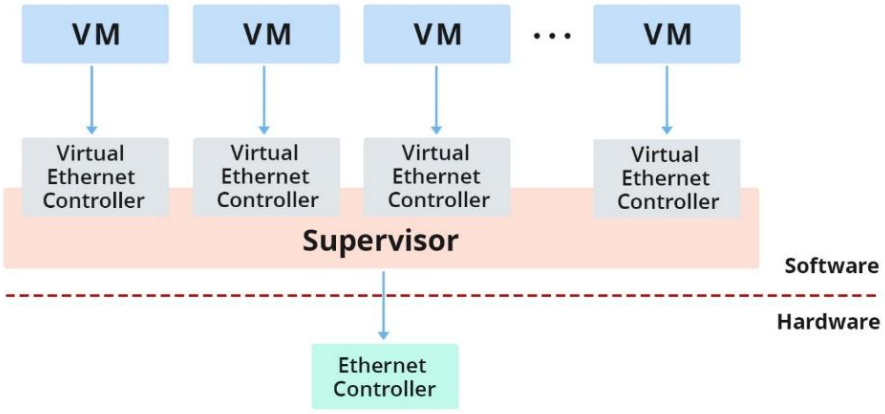


Figura 23: Virtualizzazione classica della scheda di rete [69]

La tecnologia SR-IOV propone un approccio fondamentalmente diverso dalla virtualizzazione tradizionale, permettendo la condivisione delle risorse della scheda, ovvero l'accesso all'interfaccia PCIe senza l'emulazione software.

A tal proposito le uscite PCIe vengono divise in *Physical Functions* (PF) che fanno riferimento a delle analoghe *Virtual Functions* (VF), garantendo a ciascuna macchina virtuale l'accesso diretto a risorse hardware uniche.

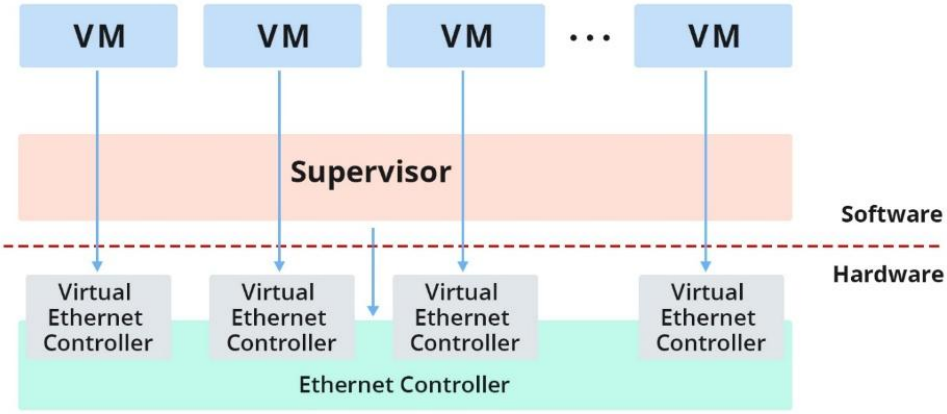


Figura 24: Virtualizzazione mediante SR-IOV [69]

I vantaggi correlati a SR-IOV sono:

- **Prestazioni migliorate:** significativa riduzione dell'overhead, perché le macchine virtuali possono comunicare direttamente con il dispositivo fisico, bypassando il livello di virtualizzazione;
- **Riduzione dell'utilizzo della CPU:** i metodi di virtualizzazione tradizionali comportano un'ampia elaborazione da parte della CPU per le operazioni di I/O. SR-IOV scarica questi compiti sull'hardware;
- **Sicurezza migliorata:** isolamento tra le macchine virtuali che condividono lo stesso dispositivo fisico, in maniera da mantenere i dati isolati e sicuri, come se ogni VM avesse un proprio hardware dedicato.
- **Qualità del servizio (QoS):** gli amministratori possono assegnare le funzioni virtuali a specifiche macchine virtuali per gestire la priorità di accesso alle risorse e garantire una QoS coerente alle applicazioni critiche;
- **Efficienza delle risorse:** per ottenere le medesime prestazioni sono necessari meno dispositivi hardware, con conseguente riduzione dei costi e del consumo energetico.

Vista l'applicazione di nicchia di questa tecnologia, solamente le schede di rete maggiormente prestanti la supportano.

Talvolta, in Windows, questa tecnologia viene implementata mediante l'Ipervisore Hyper–V e l'interfacciamento tra Funzioni Fisiche e Virtuali avviene mediante i Driver Miniport, precedentemente identificati nelle funzionalità di Transmit e Receive Side Scaling [70].

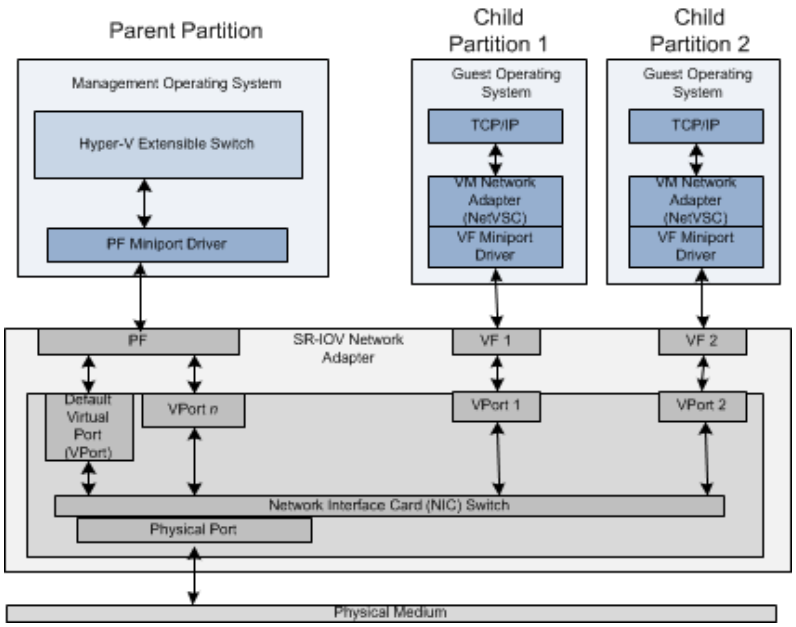


Figura 25: Architettura SR-IOV in Windows [70]

Alcune funzionalità classiche e di offload delle schede di rete Wi-Fi ed Ethernet sono riportate di seguito:

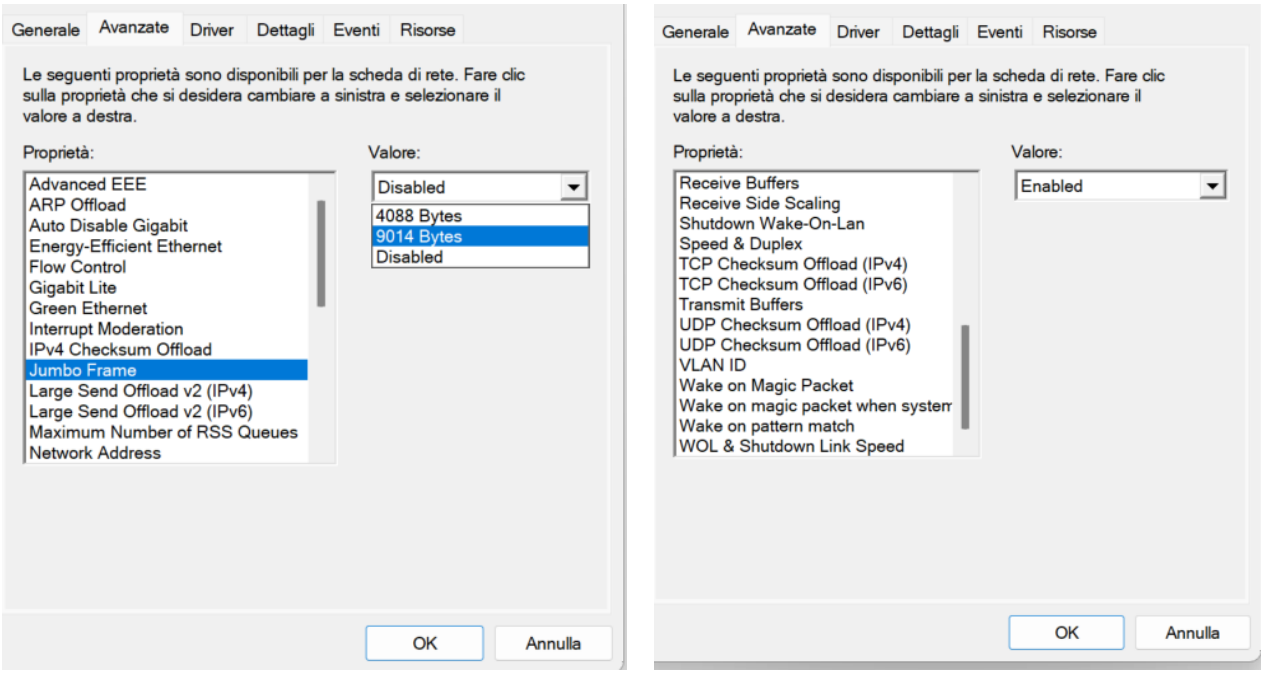


Figura 26: Funzionalità scheda di rete Ethernet

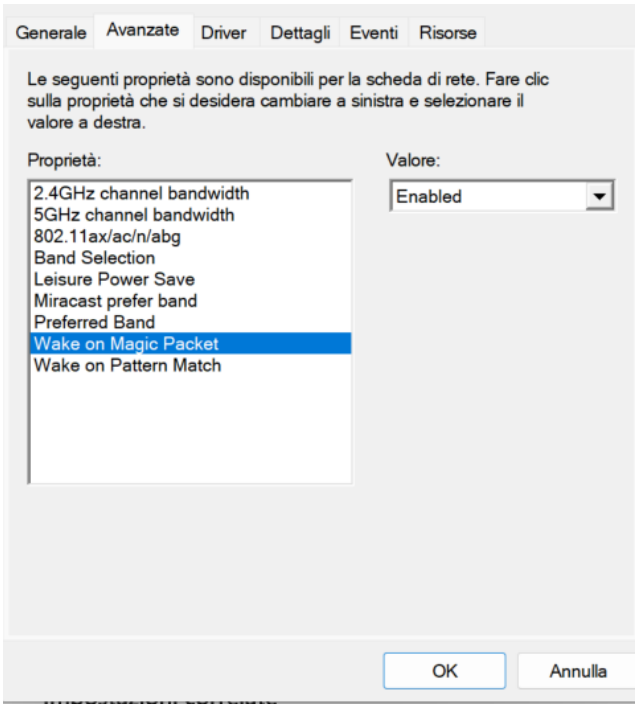


Figura 27: Funzionalità scheda di rete Wi-Fi

4 VALUTAZIONE FUNZIONALITÀ

Obiettivo

Essendomi stati resi disponibili dei calcolatori all'interno di due laboratori universitari, a Cesena e a Bologna, ho potuto effettuare studi relativi alle funzionalità di offloading delle schede di rete, analizzando due parametri delle connessioni, **latenza** e **bit rate** di canale. Queste, come si vedrà di seguito, risultano essere influenzate dalle prestazioni e dalle funzionalità di offload delle schede di rete. Infine, ho osservato anche un effetto sull'**utilizzo della CPU**.

La latenza della comunicazione misura il ritardo che subisce un pacchetto nel trasferimento tra due host in una rete.

La bit rate quantifica la larghezza di banda della connessione, ovvero il volume di dati che è possibile trasferire attraverso la rete.

Risorse

Per eseguire l'esperimento ho sfruttato due coppie di calcolatori collocati nei laboratori di elettronica e telecomunicazioni presenti a Cesena e Bologna. I computer accoppiati sono collegati direttamente mediante un cavo Ethernet.

Viene riportato di seguito un semplice schema rappresentante il trasferimento di un pacchetto tra i due host di Cesena agli indirizzi IP 10.0.0.1 e 10.0.0.2 e porta TCP del server 1234.

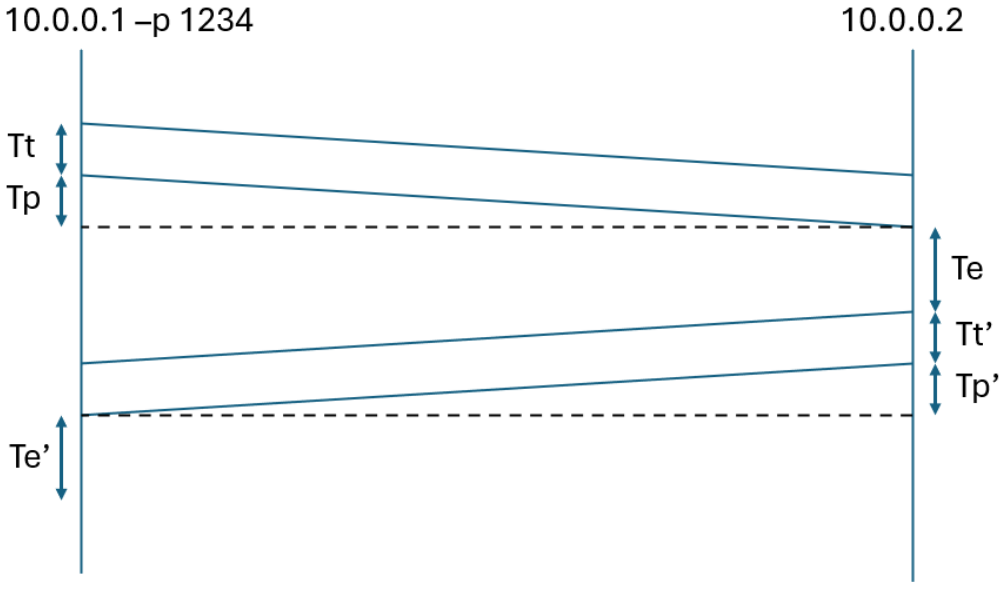


Figura 28: Latenze influenti lo scambio di due pacchetti tra i calcolatori di Cesena

Lo schema relativo ai calcolatori di Bologna differisce unicamente per gli indirizzi IP che sono, rispettivamente, 10.0.0.13 e 10.0.0.14.

Nella comunicazione tra i due dispositivi sono presenti delle **latenze additive** che concorrono alla tempistica complessiva dello scambio di pacchetti:

- *Tempo di trasmissione* (Tt): è il tempo necessario affinché il mittente completi l'inserimento di tutti i bit sul mezzo trasmissivo.
- *Tempo di propagazione* (Tp): è il tempo necessario affinché i dati, partendo dalla sorgente, raggiungano il destinatario propagandosi fisicamente nel mezzo trasmissivo.
- *Tempo di elaborazione* (Te): è il tempo necessario affinché il destinatario, ricevuti e immagazzinati in delle memorie di buffering tutti i bit della trama, ne elabori il contenuto. Un contributo significativo è dato dal *tempo di accodamento*, ovvero il tempo che intercorre tra l'istante in cui il bit arriva al destinatario ed è accodato nel buffer e il momento in cui viene letto.

Questi tempi impattano duplicemente la comunicazione in quanto presenti in ambedue le direzioni. Infatti, adottando il protocollo TCP, l'esito dell'invio di un pacchetto deve essere riscontrato mediante un pacchetto di ritorno.

È importante sottolineare che i tre contributi temporali possono essere, e generalmente sono, differenti nelle due direzioni, in misura tanto maggiore quanto più eterogenei sono i dispositivi della rete.

Possiamo così definire il *Round Trip Time* (RTT), ovvero il tempo totale necessario alla buona riuscita, oppure al riscontro di un fallimento, dello scambio del pacchetto. Esso influenza direttamente la latenza della comunicazione.

La durata del RTT è pari a:

$$RTT = (Tt + Tp + Te) + (Tt' + Tp' + Te')$$

L'unità di misura più appropriata per definire il RTT all'interno della rete oggetto di studio è il millisecondo [ms].

I calcolatori dei due laboratori sono costituiti da risorse hardware differenti tra di loro, le quali influenzano i risultati degli esperimenti. Vengono riportati di seguito i processori, le memorie principali e le schede di rete.

	CESENA	BOLOGNA
CPU	Intel(R)Core(TM) i5-4460CPU@3.20GHz **Frequenza**: 3.2 GHz **Cores/threads**: 4/4 **Cache**: L1d 128 KiB, L1i 128 KiB, L2 1 MiB L3 6 MiB	Intel Atom Processor (SnowRidge) **Frequenza**: 2.2 GHz **Cores/threads**: 4/4 **Cache**: L1d 128 KiB, L1i 128 KiB, L2 16 MiB, L3 64 MiB (valori virtualizzati)
RAM	Produttore: Hynix/Hyundai **Tipo:** DDR3 **Capacità:** 8 GB **Velocità:** 1600 MT/s	Produttore: Kingston **Tipo:** DDR4 **Capacità:** 16 GB **Velocità:** 3200 MT/s
SCHEDA DI RETE	Realtek Semiconductor RTL8111/8168/8411 PCI Express Gigabit Ethernet Controller **Speed**: 1000Mb/s	Realtek Semiconductor RTL8125 2.5GbE Controller **Speed**: 2500Mb/s

Snow Ridge è un processore dotato di virtualizzazione che si appoggia su una CPU fisica con un numero maggiore di cores rispetto ai quattro utilizzati. Esso è ideale per calcolatori utilizzati come server, anche se per la rete oggetto di studio non mostra i vantaggi di cui godrebbe in un server reale, che dovrebbe accettare molteplici connessioni, per esempio. Al contrario, il processore i5-4460 è una CPU general-purpose molto prestante.

Per quanto riguarda le memorie e le schede di rete, dalle caratteristiche elencate si deduce la superiorità dei computer di Bologna.

Quindi, complessivamente, è attendibile una prestazione migliore della connessione nel laboratorio di Bologna.

Funzioni oggetto di studio

Le schede di rete dei calcolatori possiedono alcune delle funzionalità di offloading che sono state definite precedentemente, ovvero:

- TCP segmentation offload (TSO)
- Scatter-Gather (SG)

Esso è uno speciale tipo di *Direct Memory Access* in cui vengono trasferiti vari vettori di dati tra la CPU e i buffer di memoria della scheda di rete. Supponendo di avere dei dati non contigui da trasferire, infatti, normalmente bisognerebbe effettuare molteplici operazioni di read e write, mentre la tecnica DMA permette di effettuare uno *zero-copy* per cui vengono risparmiati tempi di clock preziosi.

- Checksum al ricevitore (RX)
- Checksum al trasmettitore (TX)
- Generic Segmentation Offload (GSO)
- Generic Receive Offload (GRO)

La scheda non permette di effettuare il *Large Receive Offload*, quindi, TSO dovrà essere combinato con il *Generic Receive Offload*.

Dal terminale di Linux è possibile vedere le funzionalità mediante il comando:

ethtool -k “nome scheda”

A cui il sistema risponde con:

```
Features for “nome scheda”:
rx-checksumming: off
tx-checksumming: off
        tx-checksum-ipv4: off
        tx-checksum-ip-generic: off [fixed]
        tx-checksum-ipv6: off
        tx-checksum-fcoe-crc: off [fixed]
        tx-checksum-sctp: off [fixed]
scatter-gather: off
        tx-scatter-gather: off
        tx-scatter-gather-fraglist: off [fixed]
tcp-segmentation-offload: off
        tx-tcp-segmentation: off
        tx-tcp-ecn-segmentation: off [fixed]
        tx-tcp-mangleid-segmentation: off
        tx-tcp6-segmentation: off
generic-segmentation-offload: off
generic-receive-offload: off
large-receive-offload: off [fixed]
rx-vlan-offload: on
tx-vlan-offload: on
ntuple-filters: off [fixed]
receive-hashing: off [fixed]
highdma: on [fixed]
rx-vlan-filter: off [fixed]
vlan-challenged: off [fixed]
tx-lockless: off [fixed]
netns-local: off [fixed]
tx-gso-robust: off [fixed]
tx-fcoe-segmentation: off [fixed]
tx-gre-segmentation: off [fixed]
tx-gre-csum-segmentation: off [fixed]
tx-ipxip4-segmentation: off [fixed]
tx-ipxip6-segmentation: off [fixed]
tx-udp_tnl-segmentation: off [fixed]
tx-udp_tnl-csum-segmentation: off [fixed]
tx-gso-partial: off [fixed]
tx-tunnel-remcsum-segmentation: off [fixed]
tx-sctp-segmentation: off [fixed]
tx-esp-segmentation: off [fixed]
tx-udp-segmentation: off [fixed]
tx-gso-list: off [fixed]
fcoe-mtu: off [fixed]
tx-nocache-copy: off
loopback: off [fixed]
rx-fcs: off
rx-all: off
tx-vlan-stag-hw-insert: off [fixed]
rx-vlan-stag-hw-parse: off [fixed]
rx-vlan-stag-filter: off [fixed]
l2-fwd-offload: off [fixed]
hw-tc-offload: off [fixed]
esp-hw-offload: off [fixed]
esp-tx-csum-hw-offload: off [fixed]
rx-udp_tunnel-port-offload: off [fixed]
tls-hw-tx-offload: off [fixed]
tls-hw-rx-offload: off [fixed]
rx-gro-hw: off [fixed]
tls-hw-record: off [fixed]
rx-gro-list: off
macsec-hw-offload: off [fixed]
rx-udp-gro-forwarding: off
hsr-tag-ins-offload: off [fixed]
hsr-tag-rm-offload: off [fixed]
hsr-fwd-offload: off [fixed]
hsr-dup-offload: off [fixed]
```

In verde sono state colorate le impostazioni di interesse. Quelle caratterizzate dalla stringa [fixed] non possono essere modificate dall’utente via software.

Le funzionalità sono identiche per le schede di rete di Bologna e quelle di Cesena.

Mediante opportuni comandi da terminale è possibile attivare e disattivare selettivamente le funzionalità. Quindi ho definito otto modalità di operazione dei calcolatori, ovvero:

1) TUTTO off

sudo ethtool -K “nome scheda” tx off rx off sg off tso off gso off gro off

2) CHECKSUM on

sudo ethtool -K “nome scheda” tx on rx on sg off tso off gso off gro off

3) SG on

sudo ethtool -K “nome scheda” tx off rx off sg on tso off gso off gro off

4) CHECKSUM, SG on

sudo ethtool -K “nome scheda” tx on rx on sg on tso off gso off gro off

5) CHECKSUM, GSO, GRO on

sudo ethtool -K “nome scheda” tx on rx on sg off tso off gso on gro on

6) CHECKSUM, SG, GRO, TSO on

sudo ethtool -K “nome scheda” tx on rx on sg on tso on gso off gro on

7) CHECKSUM, SG, TSO on

sudo ethtool -K “nome scheda” tx on rx on sg on tso on gso off gro off

8) TUTTO on

sudo ethtool -K “nome scheda” tx on rx on sg on tso on gso on gro on

Le quali possono essere visualizzate nella seguente tabella:

MODALITÀ	CHECKSUM	SG	GSO	GRO	TSO
1					
2					
3					
4					
5					
6					
7					
8					

Strumenti di misura

Sebbene esistano numerosi programmi che permettono di effettuare scambi intensivi di pacchetti, molti non soddisfano i requisiti per questa ricerca. In particolare, i pacchetti da inviare devono essere provvisti di intestazioni di Trasporto, mentre molti servizi usano pacchetti ICMP, che ne sono sprovvisti.

Per lo studio della latenza ho scelto l'applicazione *Nmap*, mentre per lo studio della bit-rate ho adottato *Iperf3*.

Nmap

Nmap significa *Network Mapper* ed è un servizio gratuito per analisi e controllo della sicurezza delle reti, che permette ad esempio di determinare l'esistenza di host e contattarli per verificare la presenza di servizi attivi sulle loro porte (*port sniffing*), quali sistemi operativi eseguono, se dispongono di firewall e altre caratteristiche ancora [71].

In particolare, il servizio che utilizzerò sarà **Ncat**, il quale permette di creare un processo server in grado di accettare connessioni da un processo client. L'obiettivo è quello di instaurare una connessione affidabile tra gli host e scambiare dei pacchetti che possiedano delle significative intestazioni TCP-IP.

Esso può far operare un host in due modalità base: *Connect* e *Listen*, che possono essere intese rispettivamente come modalità client e modalità server.

Iperf3

Iperf3 è uno strumento di misurazione e ottimizzazione delle prestazioni delle reti di natura multipiattaforma che possiede, come Nmap, funzionalità client e server. L'obiettivo sarà quello di creare flussi di dati e misurare il throughput della connessione. È possibile fissare la quantità di dati da trasferire oppure un intervallo temporale durante il quale il programma tenterà di trasferire più dati possibile.

Iperf3, come Ncat, permette a un host di operare in modalità client oppure server.

In entrambi gli esperimenti sarà necessario instaurare una connessione affidabile TCP che prevederà l'esistenza di un host server e un host client.

Il processo server dovrà essere in ascolto su una specifica porta, o *socket*. La porta da me scelta è 1234 e il motivo è il seguente.

Le porte sono distinte in tre categorie [72]:

- *Well-known ports* da 0 a 1023;
- *Registered ports* da 1024 a 49151;
- *Dynamic and/or Private ports* da 49152 a 65535.

Le *Well-Known Ports* sono porte riservate unicamente ai server e assegnate dall'*Internet Assigned Numbers Authority* (IANA) a processi specifici, come la 80, per HTTP, e la 22, adibita a SSH. Inoltre, per poterle utilizzare su Linux sono necessari i permessi di *Root*. Essendo più flessibile, ho optato per una registered port, ovvero 1234.

PRIMO ESPERIMENTO: LATENZA

Il primo esperimento è finalizzato a determinare la latenza di comunicazione tra le coppie di dispositivi.
Di seguito riporto i passaggi per effettuare i trasferimenti e l'elaborazione dei di dati:

1. Tramite Ncat attivo il processo server sul primo computer, che viene inizializzato in modalità Listen come segue:

   ```
   ncat -l 10.0.0.1 1234 -k > ricevuti.txt
   ```

   L'host 10.0.0.1 si mette in ascolto sulla porta 1234 accettando connessioni successive, una alla volta, e salvando i dati ricevuti all'interno del file *ricevuti.txt*;
2. A questo punto il secondo host tenterà il Connect:

   ```
   #!/bin/bash
   for ((i=1; i<=1000; i++)); do
                   start=$(date +%s%N)
                   ncat 10.0.0.1 1234 < dati.txt
                   end=$(date +%s%N)
                   elapsed=$(((end - start) / 1000))
                   echo "Tempo totale: $elapsed us"
                   printf "%d\n" "$elapsed" >> tempi#.dat
   done
   ```

   Vengono instaurate mille connessioni ciascuna delle quali comporta il trasferimento di un file di testo *dati.txt* contenente dieci milioni di caratteri ASCII pseudo-casuali. Esso ha una dimensione di circa dieci milioni di bytes.
   I tempi di ogni transazione sono salvati in us all'interno dei file *tempi#.dat*;
3. Gli otto file tempi#.dat vengono elaborati mediante Matlab per estrarne valore medio, deviazione standard e incertezza di tipo A. Quindi, viene realizzato un grafico con barre degli errori.

Risultati

Calcolatori di Cesena

u.m. [ms]

1) Valore medio: 155.498 Incertezza: 1.067
2) Valore medio: 147.656 Incertezza: 1.109
3) Valore medio: 216.940 Incertezza: 0.359
4) Valore medio: 215.480 Incertezza: 0.330
5) Valore medio: 156.507 Incertezza: 0.949
6) Valore medio: 145.145 Incertezza: 0.295
7) Valore medio: 145.729 Incertezza: 0.266
8) Valore medio: 143.159 Incertezza: 0.262

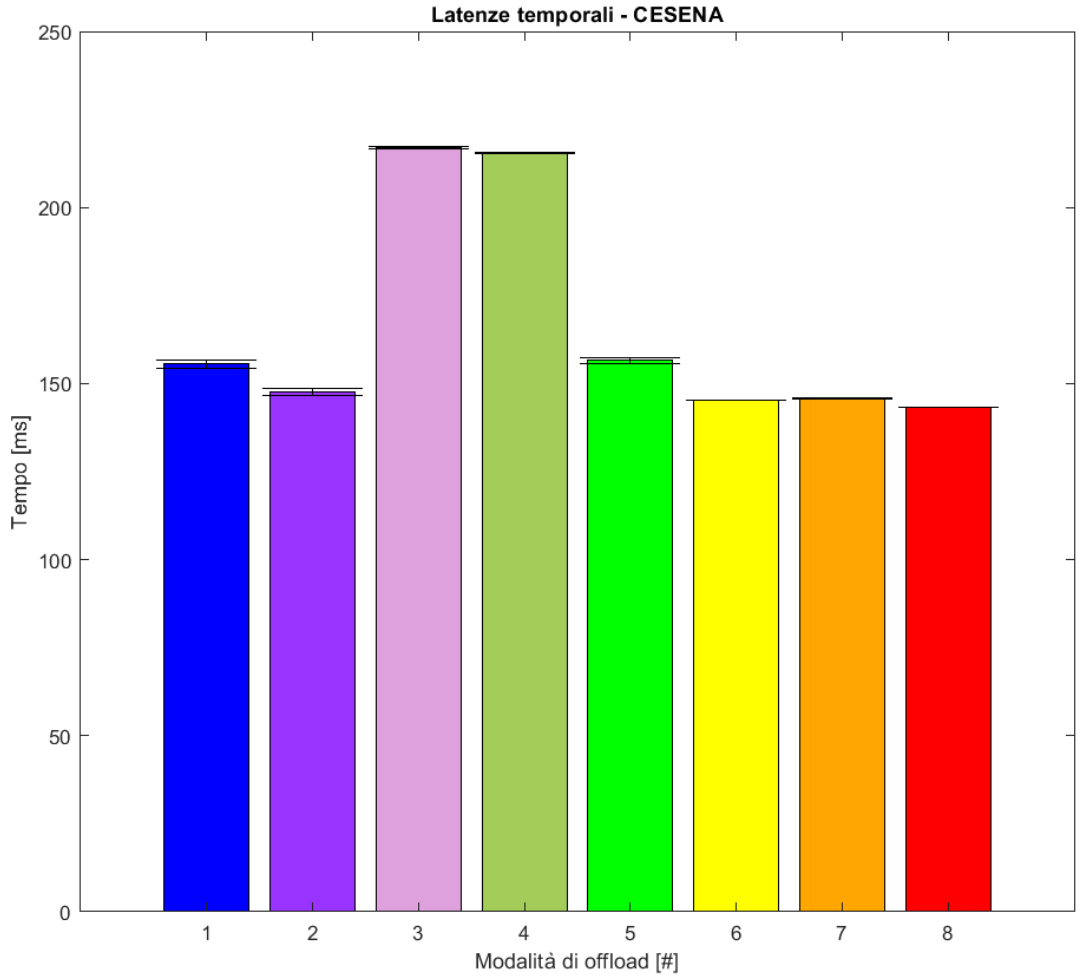


Figura 29: Latenze relative alle diverse impostazioni delle schede di rete dei computer di Cesena

Calcolatori di Bologna

u.m. [ms]

1) Valore medio: 49.839 Incertezza: 0.077
2) Valore medio: 49.589 Incertezza: 0.048
3) Valore medio: 66.930 Incertezza: 0.038
4) Valore medio: 66.797 Incertezza: 0.220
5) Valore medio: 48.301 Incertezza: 0.025
6) Valore medio: 47.811 Incertezza: 0.142
7) Valore medio: 47.549 Incertezza: 0.114
8) Valore medio: 47.133 Incertezza: 0.091

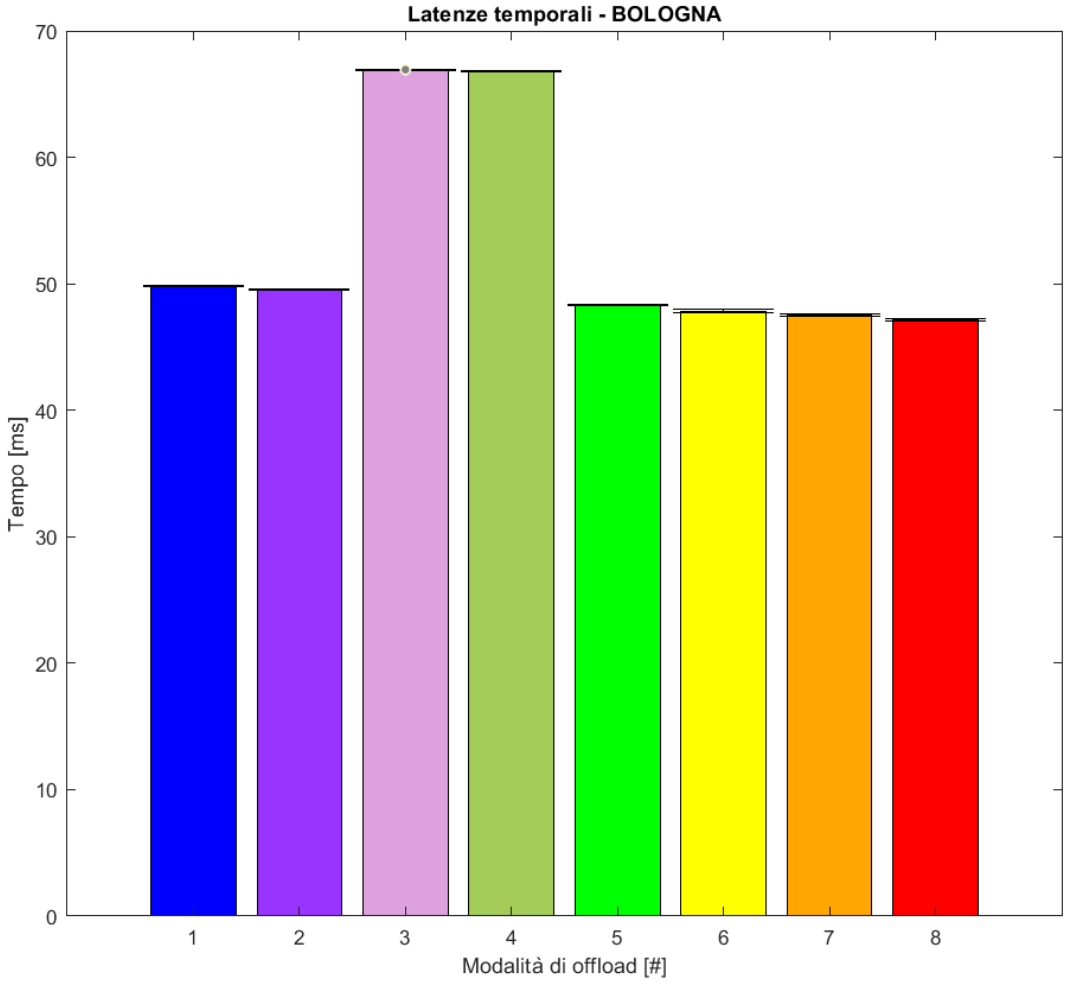


Figura 30: Latenze relative alle diverse impostazioni delle schede di rete dei computer di Bologna

- Dai primi quattro risultati si deduce che l'attivazione di **Scatter-Gather** aumenta le latenze, quindi è sconveniente attivarla singolarmente. Esso può essere vantaggioso in combinazione con TSO, per il quale è imprescindibile;
- Per quanto riguarda la **Checksum**, essa migliora leggermente i tempi;
- Nel complesso si evince come il massimo rendimento si abbia nel caso in cui tutte le funzionalità siano attivate contemporaneamente;
- Confrontando i due laboratori, si nota una differenza di latenza sostanziale a favore di quello di Bologna, motivabile dalla performance migliore delle **schede di rete**.

È possibile spiegare il rallentamento associato a Scatter-Gather addentrandosi nelle problematiche relative alla bufferizzazione dei dati scambiati tra la CPU e la scheda di rete.

Consideriamo che si intenda inviare un messaggio, allora il flusso dei bit procederà dalla CPU verso la periferica.

Se le operazioni necessarie ai protocolli di Rete e Trasporto sono eseguite dalla CPU, questa dovrà attendere il completo arrivo dei dati applicativi per effettuare i calcoli e solo allora invierà il pacchetto alla scheda di rete, che lo immagazzinerà in una propria memoria prima di inviarlo.

Nel caso in cui si intenda far eseguire le operazioni in offload alla scheda, non sarà più necessario che i bit dalla CPU siano inviati in un'unica operazione, quindi, sarà conveniente farli fluire gradualmente appena disponibili sottoforma di più vettori frammentati (*Scatter*). Una volta trasferiti integralmente, la scheda dovrà riassemblare i vettori per creare il pacchetto (*Gather*).

Per poter eseguire efficacemente lo Scatter-Gather la scheda deve disporre di hardware specifico in grado di eseguire calcoli, i quali introducono dei ritardi. In aggiunta, sono necessari molteplici accessi in memoria, in misura proporzionale al numero di frammenti da spostare. Infine, i vettori contengono informazioni aggiuntive relative alle locazioni di memoria e alle lunghezze per permettere la ricostruzione del pacchetto, creando quindi un overhead.

Per questo motivo, lo Scatter-Gather è utile unicamente nel momento in cui la scheda debba eseguire delle operazioni di offload, altrimenti risulterà conveniente far completare i calcoli alla CPU e solamente dopo trasferire l'intero pacchetto [73].

SECONDO ESPERIMENTO: BIT-RATE

Il secondo esperimento mira a quantificare la bit-rate del canale di comunicazione tra le coppie di dispositivi al variare delle funzionalità di offload.

Tramite Iperf3 instauro la connessione tra i due computer:

1. Il primo host viene posto in modalità server (daemon) in ascolto sulla porta 1234:

   ```
   iperf3 -s (-D) -p 1234
   ```

2. Il secondo viene messo in modalità client e comincia lo scambio di dati reciproco che viene gestito da Iperf3 stesso:

   ```
   iperf3 -c 10.0.0.13 -p 1234 -f k -t 100 --logfile risultato.txt
   ```

   Nel secondo host vengono salvati i valori della banda in kbit/s all'interno del file *risultato.txt*. Il trasferimento dura cento secondi e viene registrato un campione al secondo.

Risultati

Calcolatori di Cesena

u.m. [kbit/s]

1) Valore medio: 773690 Incertezza: 530
2) Valore medio: 773910 Incertezza: 460
3) Valore medio: 510990 Incertezza: 480
4) Valore medio: 512440 Incertezza: 500
5) Valore medio: 758380 Incertezza: 800
6) Valore medio: 824030 Incertezza: 670
7) Valore medio: 826690 Incertezza: 930
8) Valore medio: 827060 Incertezza: 440

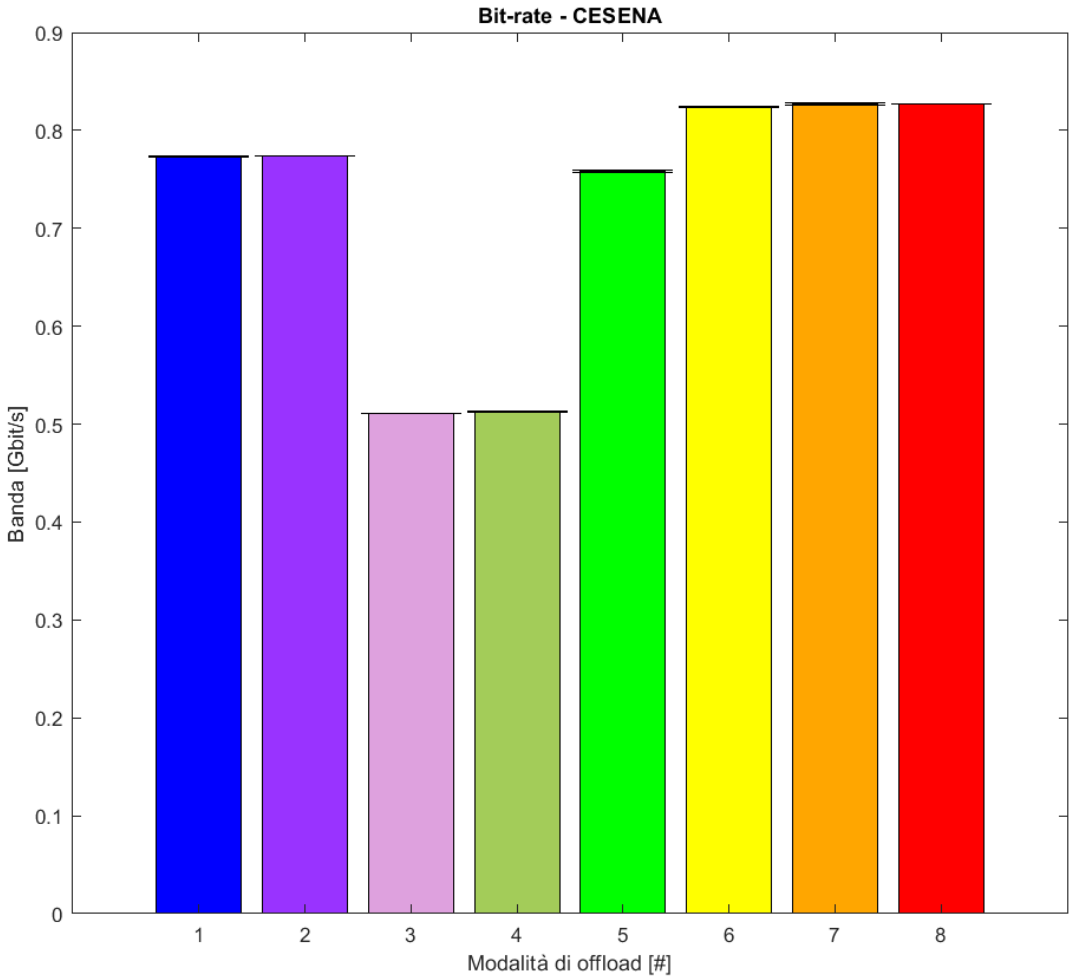


Figura 31: Bit-rate relative alle diverse impostazioni delle schede di rete dei computer di Cesena

Calcolatori di Bologna

u.m. [kbit/s]

1) Valore medio: 2198370 Incertezza: 590
2) Valore medio: 2199610 Incertezza: 640
3) Valore medio: 1496970 Incertezza: 530
4) Valore medio: 1495710 Incertezza: 580
5) Valore medio: 2265780 Incertezza: 660
6) Valore medio: 2353750 Incertezza: 450
7) Valore medio: 2353830 Incertezza: 590
8) Valore medio: 2353970 Incertezza: 520

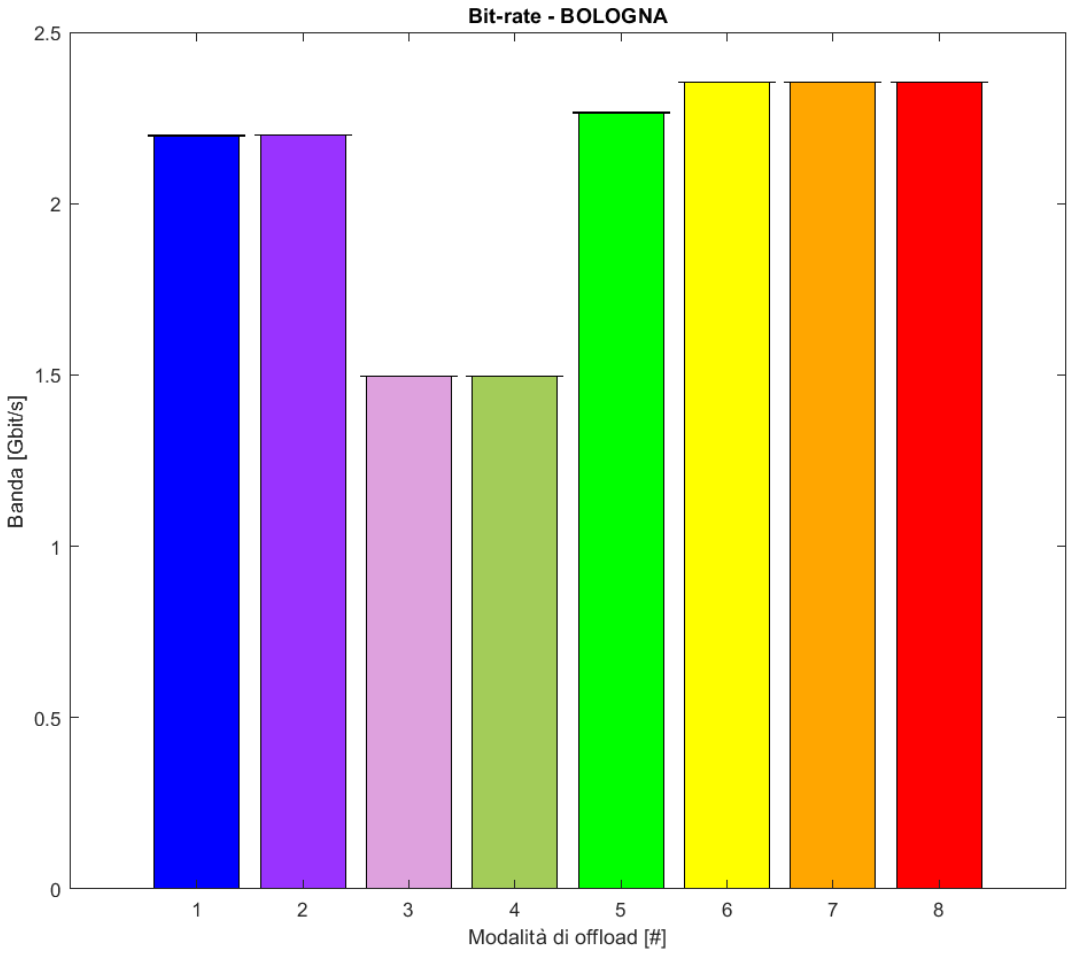


Figura 32: Bit-rate relative alle diverse impostazioni delle schede di rete dei computer di Bologna

- I primi quattro risultati confermano le deduzioni dell'esperimento uno riguardanti **Scatter-Gather**: si assiste a un peggioramento della bit-rate di canale se usato singolarmente;
- Per quanto riguarda la **Checksum**, non si osservano influenze dirette sulla bit-rate;
- Nel complesso si conferma che la performance ottima si ha nel caso in cui tutte le funzionalità siano attivate contemporaneamente;
- Confrontando i risultati dei due laboratori si osserva che la banda del collegamento di Bologna è nettamente superiore rispetto a quella di Cesena, coerentemente al fatto che le **schede di rete** dei primi hanno una portata del gigabit/s, mentre quelle dei secondi pari a 2.5 gigabit/s.

OSSERVAZIONE: IMPIEGO DELLA CPU

Sfruttando il comando del terminale Linux:

```
top
```

Ho avuto modo di osservare l'impiego di CPU da parte del processo Iperf3, constatando dei valori interessanti e coerenti con le aspettative descritte nei paragrafi precedenti della tesi: avevo inizialmente ipotizzato infatti che le funzionalità di offload diminuissero l'utilizzo di CPU, moderando le richieste di interrupt provenienti dalla scheda di rete.

Ho quindi deciso di registrare i valori di utilizzo della stessa durante l'esecuzione del programma Iperf3 per trarne delle statistiche.

Mentre Iperf3 scambia i dati tra i due host, sul server ho messo in esecuzione un programma che iterativamente registra in un file di testo l'impiego di CPU da parte del processo Iperf3 (id del processo 156799, ma variabile ogni volta che viene avviato).

```
#!/usr/bin/bash
PID=156799
OUTPUT="dati.txt"
> "$OUTPUT"
while true; do
        CPU=$(top -bn1 -p $PID | grep "$PID" | awk '{print $9}')
        echo "$CPU" >> "$OUTPUT"
done
```

Il servizio *top* calcola il valore di utilizzo della CPU, restituendo un output con più dati che, mediante manipolazione, viene ridotto al dato di interesse. Questo viene quindi salvato nel file *dati.txt*.

Risultati

CESENA

u.m. [%]

1) Valore medio: 13.66	Incertezza: 0.32
2) Valore medio: 14.37	Incertezza: 0.31
3) Valore medio: 12.91	Incertezza: 0.27
4) Valore medio: 12.88	Incertezza: 0.28
5) Valore medio: 2.79	Incertezza: 0.14
6) Valore medio: 2.81	Incertezza: 0.14
7) Valore medio: 8.61	Incertezza: 0.35
8) Valore medio: 2.92	Incertezza: 0.14

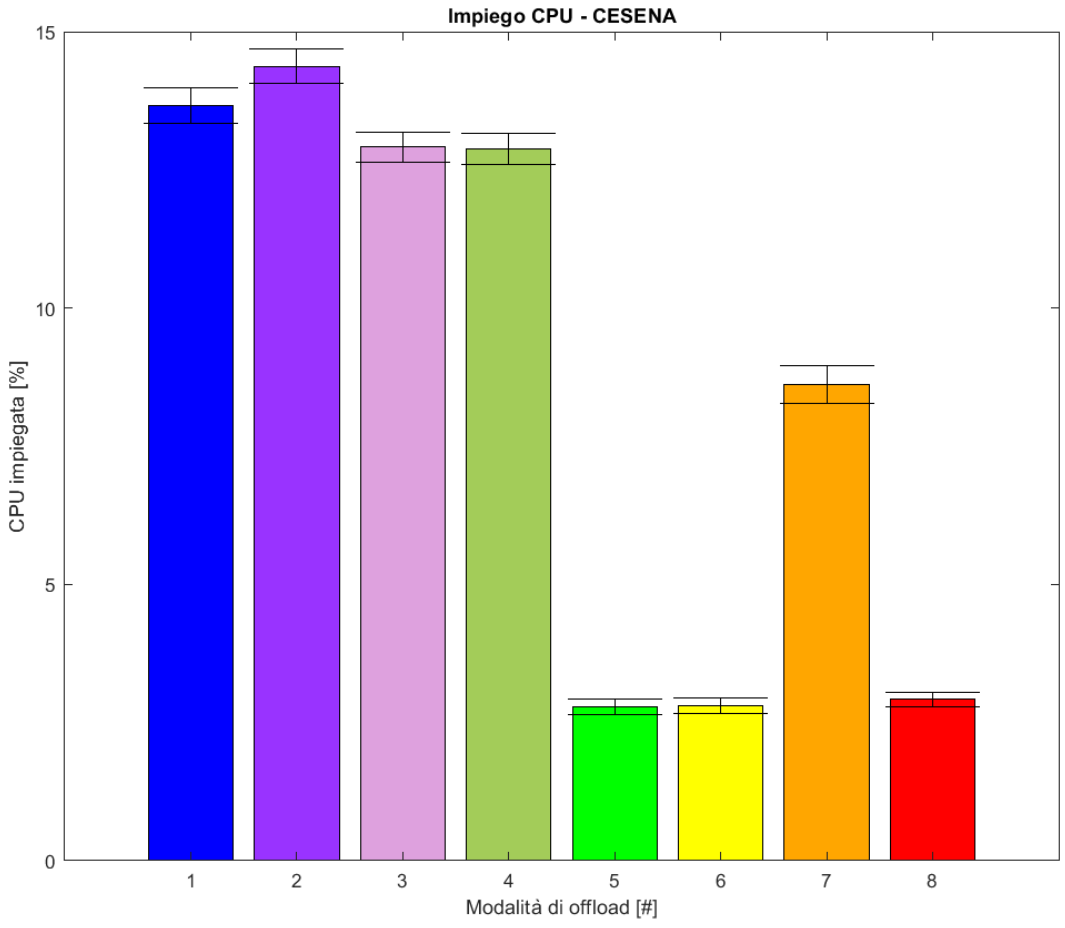


Figura 33:Impiego della CPU relativo alle diverse impostazioni delle schede di rete dei computer di Cesena

BOLOGNA

u.m. [%]

1) Valore medio: 20.80 Incertezza: 0.22
2) Valore medio: 20.72 Incertezza: 0.22
3) Valore medio: 17.31 Incertezza: 0.20
4) Valore medio: 17.44 Incertezza: 0.19
5) Valore medio: 4.38 Incertezza: 0.16
6) Valore medio: 3.66 Incertezza: 0.15
7) Valore medio: 20.74 Incertezza: 0.20
8) Valore medio: 3.64 Incertezza: 0.15

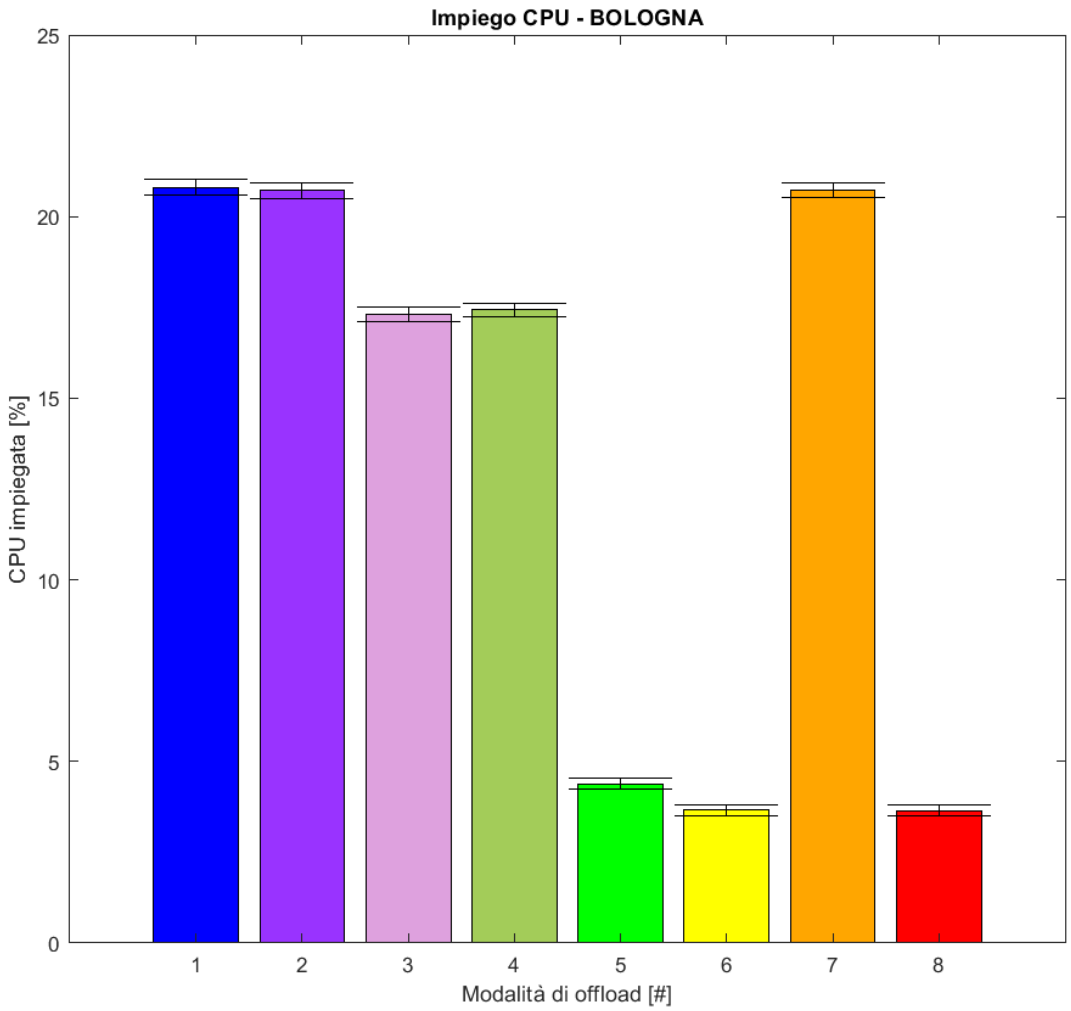


Figura 34:Impiego della CPU relativo alle diverse impostazioni delle schede di rete dei computer di Bologna

- Si nota come la **modalità 7** in entrambe le simulazioni mostri un utilizzo del processore inefficiente. Essa è caratterizzata dalla privazione di GSO e GRO;
- Nel complesso, le **modalità 5,6 e 8** risultano essere le migliori. Non è possibile definire un podio esatto, dal momento che l'incertezza della misura è significativamente elevata a tal punto da rendere la differenza tra i risultati non completamente attendibile.
- Infine, è da evidenziare un consumo della CPU più basso nei calcolatori di Cesena, motivato dalla quantità di dati trasferiti che è circa tre volte inferiore e dalla CPU che possiede una frequenza di calcolo superiore, sebbene sia dotata di minori memorie cache e sia presente una RAM meno performante.

TABELLE RIASSUNTIVE

Per semplicità di visualizzazione ho realizzato delle tabelle che riassumono i risultati dei due esperimenti e dell'osservazione.

Latenze [ms]

Modalità	1	2	3	4	5	6	7	8
CESENA	155.5	147.6	216.9	215.5	156.5	145.1	145.7	143.1
BOLOGNA	49.8	49.6	66.9	66.8	48.3	47.8	47.5	47.1

Bit-rate [Gbit/s]

Modalità	1	2	3	4	5	6	7	8
CESENA	0.774	0.774	0.511	0.512	0.758	0.824	0.827	0.827
BOLOGNA	2.198	2.200	1.497	1.496	2.266	2.354	2.354	2.354

CPU [%]

Modalità	1	2	3	4	5	6	7	8
CESENA	13.66	14.37	12.91	12.88	2.79	2.81	8.61	2.92
BOLOGNA	20.80	20.72	17.31	17.44	4.38	3.66	20.74	3.64

5 CONCLUSIONE

Da quanto descritto nella tesi si evincono numerose qualità delle funzionalità di offloading delle schede di rete, in termini di efficienza temporale ed energetica, in primo luogo, così come scalabilità e sicurezza.

Si sta assistendo ormai da decenni ad una crescente e illimitata domanda di trasmissione ed elaborazione dei dati per assecondare le eterogenee necessità degli esigenti utenti della Rete.
Le continue innovazioni stanno rendendo possibili capacità di calcolo sempre più elevate, seppur trattandosi spesso di architetture general purpose, non ottimizzate per le operazioni di rete. Conseguentemente, è lecito pensare che le prospettive di diffusione delle Smart Nics, dotate di DPU e FPGA, siano veramente eccellenti. Queste, infatti, si stanno dimostrando strategiche non solo nell'ambito delle telecomunicazioni e dei data center, ma anche in settori specifici come l'Intelligenza Artificiale.

Esistono già personal computer con processori specializzati per il machine learning, ovvero le NPU e ancora più diffusi sono quelli dedicati all'elaborazione di immagini, ovvero le GPU. Non sarebbe strano se, nel futuro più prossimo, trovassimo computer dotati di Smart Nics e magari fossero proprio queste uno dei criteri di scelta principali per molti consumatori.

Di fronte a questo nuovo mondo si aprono anche nuove sfide, sia in termini di progettazione hardware e software, sia in ambito di standardizzazione, interoperabilità e gestione. Restiamo in attesa di conoscere quello che sarà il disegno dell'infrastruttura informatica del futuro.

6 BIBLIOGRAFIA

Introduzione

[1] Kahn Robert. “Internet”. Britannica, 5 giugno 2025,
https://www.britannica.com/technology/Internet

[2] Hojlo Jeffrey. “Future of Industry Ecosystems: Shared Data and Insights”. International Data Corporation, 6 gennaio 2021,
https://blogs.idc.com/2021/01/06/future-of-industry-ecosystems-shared-data-and-insights/

[3] Vailshery Lionel Sujay. “Number of Internet of Things (IoT) connections worldwide from 2022 to 2023, with forecasts from 2024 to 2034”. Statista, 26 maggio 2025,
https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/

[4] “Data Never Sleeps 12.0”. Domo, 2024,
https://www.domo.com/learn/infographic/data-never-sleeps-12

[5] “Individuals using the Internet”. International Telecommunication Union,
https://datahub.itu.int/data/?e=701&i=11624&v=

[6] “Telecomunicazione”. Enciclopedia Treccani, 2018,
Telecomunicazione - Significato ed etimologia - Vocabolario - Treccani

1. Modelli di Internet

[7] “ISO/IEC 7498-1:1994”. International Organization for Standardization, 1994,
https://www.iso.org/obp/ui/en/#iso:std:iso-iec:7498:-1:ed-1:v2:en

[8] Postel Jon. “RFC: 793 TRANSMISSION CONTROL PROTOCOL”. Internet Engineering Task Force, settembre 1981,
https://datatracker.ietf.org/doc/html/rfc793

[9] Wesley Eddy. “RFC: 9293 Transmission Control Protocol (TCP)”. Internet Engineering Task Force, agosto 2022,
https://datatracker.ietf.org/doc/html/rfc9293

[10] Postel Jon. “RFC: 768 User Datagram Protocol”. Internet Engineering Task Force, 28 agosto 1980,
https://datatracker.ietf.org/doc/html/rfc768

[11] Grisogani Luca. “Protocollo QUIC ed evoluzione di http”. AMSlaurea, 16 giugno 2022,
https://amslaurea.unibo.it/id/eprint/26162/

[12] Postel Jon. “RFC: 791 INTERNET PROTOCOL”. Internet Engineering Task Force, settembre 1981,
https://datatracker.ietf.org/doc/html/rfc791

[13] Hinden Bob, Dr. Steve E. Deering, “RFC: 2460 Internet Protocol, Version 6 (IPv6)”. Internet Engineering Task Force, dicembre 1998,
https://datatracker.ietf.org/doc/html/rfc2460

[14] “Ethernet Frame Format“. GeeksforGeeks, 28 dicembre 2024, https://www.geeksforgeeks.org/ethernet-frame-format/

[15] “IEEE 802.11 Mac Frame”. GeeksforGeeks, 23 aprile 2025, https://www.geeksforgeeks.org/ieee-802-11-mac-frame/

2. Schede di rete

[16] “Ethernet Through the Years: Celebrating the Technology’s 50th Year of Innovation”. IEEE Standards Association, 24 maggio 2023, https://standards.ieee.org/beyond-standards/ethernet-50th-anniversary/

[17] “High Speed 5GbE NIC Solution (RTL8126-VB)”. Realtek, https://www.realtek.com/Product/ProductHitsDetail?id=4425&menu_id=643

[18] “Introducing Two New Ethernet Controllers and Network Adapter Products”. Intel, https://www.intel.com/content/www/us/en/products/details/ethernet.html

[19] “N425G - 4 x 25/10GbE OCP 3.0 Adapter”. Broadcom, https://www.broadcom.com/products/ethernet-connectivity/network-adapters/n425g

[20] “TX401”. TP-Link, https://www.tp-link.com/it/home-networking/pci-adapter/tx401/

[21] “Marvell® Scalable mGig AQC113/AQC114/AQC114CS/AQC115C”. Marvell, agosto 2022, https://www.marvell.com/content/dam/marvell/en/public-collateral/ethernet-adaptersandcontrollers/marvell-fastLinq-edge-product-brief.pdf

[22] “XG-C100C”. ASUS, https://www.asus.com/it/networking-iot-servers/wired-networking/all-series/xg-c100c/

[23] “ConnectX-7 400G Adapters”. NVIDIA Corporation, https://resources.nvidia.com/en-us-accelerated-networking-resource-library/connectx-7-datasheet

[24] “P1400GD — 1 x 400GbE PCIe NIC”. Broadcom, https://www.broadcom.com/products/ethernet-connectivity/network-adapters/p1400g

[25] “Alaska® A 1.6T PAM4 DSP for Active Electrical Cable (AEC)”. Marvell, giugno 2024, https://www.marvell.com/content/dam/marvell/en/public-collateral/phys-transceivers/marvell-alaska-a-1-6t-pam4-dsp-product-brief.pdf

[26] “T62100-CR”. Chelsio, https://www.chelsio.com/nic/unified-wire-adapters/t62100-cr/

[27] Molkin Denis. “Scheda di rete e un cavo con un connettore di raccordo su uno sfondo bianco”. Alamy, 31 gennaio 2010, https://www.alamy.it/foto-immagine-scheda-di-rete-e-un-cavo-con-un-connettore-di-raccordo-su-uno-sfondo-bianco-52502399.html?imageid=63379585-EF68-4A3B-A7EA-73C5BBE42065&p=172813&pn=1&searchId=ba5500b490cc25991d434b96697e8c91&searchtype=0

[28] “The Evolution of Wi-Fi Technology and Standards”. IEEE Standards Association, 16 maggio 2023, https://standards.ieee.org/beyond-standards/the-evolution-of-wi-fi-technology-and-standards/

[29] “AirPort”. Apple Computer Inc., agosto 1999,
https://dn790000.ca.archive.org/0/items/ads_L04183A_AirPort_DS/L04183A_AirPort_DS.pdf

[30] Jared C. Benedict. “File: Apple graphite airport base station front.jpg”. WikipediaFile: Appleere 2004,
https://it.m.wikipedia.org/wiki/File:Apple_graphite_airport_base_station_front.jpg

[31] Koziol Michael. “What Is Wi-Fi 7?”. IEEE Spectrum, 27 maggio 2022,
https://spectrum.ieee.org/what-is-wifi-7

[32] “Adattatore USB 3.0 WiFi 6E Nighthawk® AXE3000”. Netgear,
https://www.netgear.com/it/home/wifi/adapters/a8000/

[33] “Archer TBE400UH”. TP-Link,
https://www.tp-link.com/it/home-networking/high-power-adapter/archer-tbe400uh/

[34] “BE6500 WiFi 7 USB Adapter”. MSI,
https://www.msi.com/Networking/BE6500-WiFi-7-USB-Adapter

[35] “Intel® Killer™ Wi-Fi 7 BE1750”. Intel,
https://www.intel.com/content/www/us/en/products/sku/230083/intel-killer-wifi-7-be1750-is/specifications.html

[36] “FastConnect 7900”. Qualcomm,
https://www.qualcomm.com/products/technology/wi-fi/fastconnect/fastconnect-7900

[37] “BCM47722”. Broadcom,
https://www.broadcom.com/products/wireless/wireless-lan-infrastructure/bcm47722

[38] “Realtek to Showcase Advanced Comprehensive Multimedia and Communication Network Solutions at IBC2023”. Realtek, 13 settembre 2023,
https://www.realtek.com/Article/NewsDetail?id=4352&app_id=18

[39] “Archer TBE550E”. Acer,
https://www.tp-link.com/it/home-networking/pci-adapter/archer-tbe550e/

[40] “PCE-BE92BT”, ASUS,
https://www.asus.com/it/networking-iot-servers/adapters/all-series/pce-be92bt/

[41] “What Are Smart NICs?”. Supermicro,
https://www.supermicro.com/en/glossary/smart-nics

[42] “What Is a Data Processing Unit (DPU)?”. Supermicro,
https://www.supermicro.com/en/glossary/data-processing-unit

[43] Howard. “DPU: Uno dei tre Pilastri dell'Informatica per il Futuro“. FS Community, 6 febbraio 2024,
https://community.fs.com/it/article/dpu-one-of-the-three-pillars-of-computing-going-forward.html

[44] Deierling Kevin. “What Is a DPU?”. NVIDIA Corporation, 20 maggio 2020,
https://blogs.nvidia.com/blog/whats-a-dpu-data-processing-unit/

[45] Schneider Josh, Smalley Ian. “Cos'è un field programmable gate array (FPGA)?”. IBM Corporation, 8 maggio 2024,
https://www.ibm.com/it-it/think/topics/field-programmable-gate-arrays

[46] “FPGA SmartNIC FB2CDG1@AGM39D-2 Agilex™ M-series FPGA Based”. Silicom Ltd., https://www.silicom-usa.com/pr/server-adapters/programmable-fpga-server-adapter/fpga-intel-based-2/fpga-intel-agilex-based/fpga-smartnic-fb2cdg1agm39d-2-intel-based/

3. Funzionalità delle schede di rete

[47] Bhatt Vipul, Bhoja Sudeep, Farhood Arash, Nicholl Gary. “PAM-N Comparison”. IEEE802.org, settembre 2012, https://www.ieee802.org/3/bm/public/sep12/bhatt_01a_0912_optx.pdf

[48] Prieto Dunia, Pérez Rubén. “Study of Undetected Error Probability of IEEE 802.3 CRC-32 code for MTTFPA analysis”. IEEE 802.3bv Task Force, gennaio 2015, https://www.ieee802.org/3/bv/public/Jan_2015/perezaranda_3bv_2_0115.pdf

[49] “VLAN Trunking”. Network Academy, VLAN Trunking | NetworkAcademy.io

[50] “CSMA/CD: spiegazione del processo”. IONOS, 7 maggio 2019, https://www.ionos.it/digitalguide/server/know-how/csmacd/

[51] Bennett Mike. “IEEE 802.3az Energy Efficient Ethernet”. IEEE802.org, 16 luglio 2008, https://www.ieee802.org/3/maint/public/joint_bennett_1_0708.pdf

[52] Twingate Team. “What is a Magic Packet? How It Works & Examples”. Twingate, 1 agosto 2024, https://www.twingate.com/blog/glossary/magic%20packet

[53] Gupta Shradha. “The TCP Offload Engine”. Medium.com, 7 aprile 2021, https://shradha741.medium.com/the-tcp-offload-engine-35fe844dcade

[54] “TCP offload engine”. Wikipedia.org, 28 maggio 2025, https://en.wikipedia.org/wiki/TCP_offload_engine

[55] Braden Bob, Borman D., Partridge C., “RFC: 1071 Computing the Internet Checksum”. Internet Engineering Task Force, settembre 1988, https://datatracker.ietf.org/doc/html/rfc1071

[56] Rijsinghani Anil. “RFC: 1624 Computation of the Internet Checksum via Incremental Update”. Internet Engineering Task Force, maggio 1994, https://datatracker.ietf.org/doc/html/rfc1624

[57] “TCP large send offload”. IBM Corporation, 20 gennaio 2025, https://www.ibm.com/docs/kk/aix/7.3.0?topic=tuning-tcp-large-send-offload

[58] “8.10. NIC Offloads”. RedHat inc., https://docs.redhat.com/en/documentation/red_hat_enterprise_linux/6/html/performance_tuning_guide/network-nic-offloads

[59] “Ultimate Guide to JUMBO Frames: Configuration and Best MTU Size”. StoneFly.com, https://stonefly.com/resources/jumbo-frames-configuration-and-best-mtu-size/

[60] BasuMallick Chiradeep. “What Is Direct Memory Access (DMA)? Meaning, Types, Principles, Working, and Benefits”. Spiceworks Inc., 14 marzo 2024, https://www.spiceworks.com/tech/hardware/articles/direct-memory-access/

[61] “Enhancing Data Movement and Access for GPUs”. NVIDIA Corporation, https://developer.nvidia.com/gpudirect

[62] “Address Resolution Protocol”. Cisco.com, https://www.cisco.com/c/en/us/td/docs/ios-xml/ios/ipaddr_arp/configuration/xe-3s/arp-xe-3s-book/arp-config-arp.pdf

[63] “Protocol Offloads for NDIS Power Management”. Microsoft.com, 14 marzo 2023, https://learn.microsoft.com/en-us/windows-hardware/drivers/network/protocol-offloads-for-ndis-power-management

[64] Rescorla E.. “RFC: 8446 The Transport Layer Security (TLS) Protocol Version 1.3”. Internet Engineering Task Force, agosto 2018, https://datatracker.ietf.org/doc/html/rfc8446

[65] Herbert Tom, de Bruijn Willem. “Scaling in the Linux Networking Stack”. The kernel development community, https://www.kernel.org/doc/html/v5.1/networking/scaling.html

[66] “Driver miniport”. Microsoft.com, 27 settembre 2024, https://learn.microsoft.com/it-it/windows-hardware/drivers/network/ndis-miniport-drivers2

[67] “8.9. Accelerated RFS”. RedHat inc., https://docs.redhat.com/en/documentation/red_hat_enterprise_linux/6/html/performance_tuning_guide/network-acc-rfs

[68] Susnjara Stephanie, Smalley Ian. “What is virtualization?”. IBM Corporation, 9 aprile 2025, https://www.ibm.com/think/topics/virtualization

[69] Howard. “Cos'è la Virtualizzazione I/O a Radice Singola (SR-IOV)?”. FS Community, 19 ottobre 2023, https://community.fs.com/IT/blog/what-is-single-root-io-virtualization-sriov.html

[70] “architettura SR-IOV”. Microsoft.com, 25 maggio 2025, https://learn.microsoft.com/it-it/windows-hardware/drivers/network/sr-iov-architecture

4. Valutazione Funzionalità

[71] “Nmap: Discover your network”. Nmap.org, https://nmap.org/

[72] Raber Yan. “Porta TCP/IP: cosa sono, lista, protocollo”. HTML.it, 7 aprile 2004, https://www.html.it/articoli/le-porte-del-protocollo-tcpip/

[73] McLoughlin Mark. “Checksums, Scatter-Gather I/O and Segmentation Offload”. blogs.gnome.org, 28 maggio 2008, https://blogs.gnome.org/markmc/2008/05/28/checksums-scatter-gather-io-and-segmentation-offload/