**RETI DI CALCOLATORI - LABORATORIO**

Sommario

[LEZIONE 1 1](#_Toc160315426)

[Identificazione Endpoint 1](#_Toc160315427)

[Indirizzi IP 1](#_Toc160315428)

[TCP E UDP 3](#_Toc160315429)

[Modello Client-Server 7](#_Toc160315430)

# LEZIONE 1

## Identificazione Endpoint

Ogni qual volta due endpoint vogliono comunicare devono identificarsi univocamente

Tale identificazione avviene attraverso due livelli di indirizzamento:

* Il primo determina l'host su cui e' in esecuzione il processo
* Ad ogni host di una rete IP e' associato un indirizzo IP
* Il secondo determina il processo con cui si vuole comunicare
* Ad ogni applicazione in esecuzione su un host è associato un numero di porta.

## Indirizzi IP

Un Indirizzo IP è un numero che identifica univocamente un dispositivo collegato ad una rete informatica che utilizza lo standard IP (Internet Protocol).

Gli indirizzi IPv4 sono costituiti da 32 bit (4 byte), e vengono descritti con 4 numeri decimali rappresentati su 1 byte (quindi ogni numero varia tra 0 e 255) separati da un punto. Es: 192.168.1.134

Questa rappresentazione limita lo spazio di indirizzamento a 4,294,967,296 indirizzi univoci possibili ( 2^32). Inoltre la rete internet esclude 18.000.000 indirizzi utilizzati per le reti private.

Per ovviare al problema della mancanza di indirizzi IP dovuta alla costante crescita di Internet è stato introdotto l'IPv6

Nell’IPv6 gli indirizzi sono lunghi 128 bit (quindi 2^128 possibili indirizzi).

L’indirizzo viene suddiviso in 8 blocchi di 16 bit ciascuno. I blocchi sono separati da “:” e vengono rappresentati in notazione esadecimale.

Es: **3ffe:1001:0001:0100:0a00:20ff:fe83:5531**

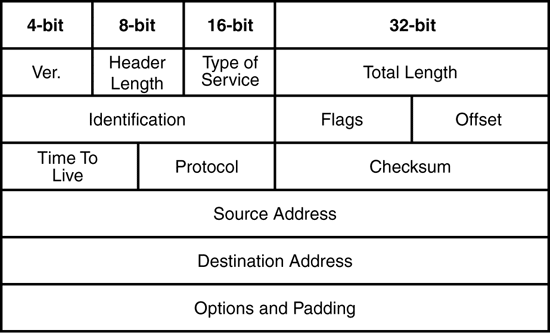
Es: **3ffe:1001:0001:0000:0000:0000:0000:0001**

Esistono delle semplificazioni:

* **si possono omettere gli zeri iniziali** 
  + **3ffe:1001:1:100:a00:20ff:fe83:5531**
* **si possono sostituire gruppi di zeri con** “**::**”
  + **3ffe:1001:1::1**

Gli indirizzi IPv6 compatibili IPv4 si scrivono: “**::163.162.170.171”**

IP HEADER:



In un ambiente multitasking più processi in esecuzione su uno stesso host devono poter comunicare mediante lo stesso sottosistema di rete. E' necessario consentire più connessioni simultaneamente.

Per poter tenere distinte le diverse connessioni su uno stesso host si utilizzano le **porte**

Numeri di Porta  
Le porte sono interi a 16 bit da 0 a 65535

* Da 0 a 1023: porte riservate (ai processi di root)
* Da 1024 a 49151: porte registrate
* Da 49152 a 65535: porte effimere (per i client, ai quali non interessa scegliere una porta specifica)

Esempi di porte riservate

* 21 ftp (trasferimento file)
* 22 ssh (login remoto sicuro)
* 25 smtp (invio email)
* 80 http (web)
* 143 imap (lettura email)

## TCP E UDP

I due principali protocolli relativi al livello di trasporto sono:

* **TCP** (Transport Control Protocol)
* **UDP** (User Datagram Protocol);

**TCP** o Transmission Control Protocol è un protocollo che ha le seguenti caratteristiche:

* **Servizio connesso**: Il protocollo TCP crea un canale logico (connessione logica) tra due host.

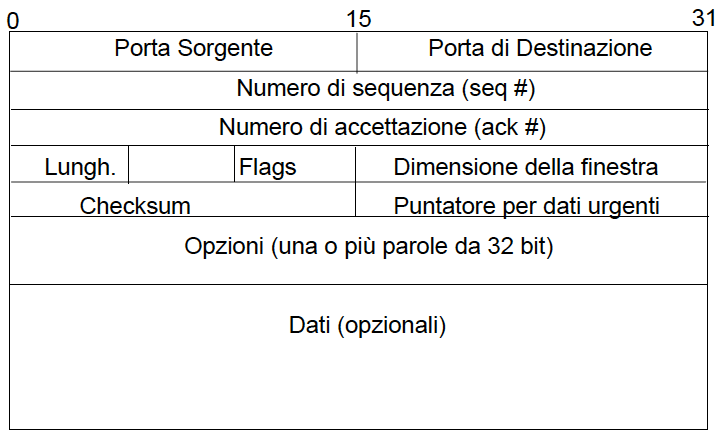
Ciò implica che il TCP lavora in unicast (1:1, un host deve potersi collegare con un altro host a livello logico). La connessione viene instaurata in prima istanza attraverso una tecnica chiamata Three-Way-Handshake e rilasciano la connessione attraverso una tecnica chiamata Four-Way-End-Connection.

* **Controllo dell’errore:** Ogni singolo segmento veicolato dal TCP è sottoposto al controllo dell’errore (impostato nell’header TCP).
* **Affidabile:** Assegna un numero di sequenza ad ogni byte trasmesso, attendendo una conferma di avvenuta ricezione (ACK). ll TCP ricevente, quando restituisce un ACK al mittente, invia anche il **numero massimo di sequenza** che può ricevere nella prossima trasmissione.
* **Ordine dei segmenti:** Il protocollo TCP è in grado di riordinare in automatico i segmenti TCP.

Di conseguenza i segmenti arrivano sempre ordinati.

* **Controllo del flusso:** L’host mittente e destinatario si mettono d’accordo per far si che un host non inondi di segmenti l’altro host che non è in grado di gestire.

Un segmento consiste in un preambolo (header) di 20 byte, più un campo opzionale di 4 byte, seguito da zero o più byte di dati;



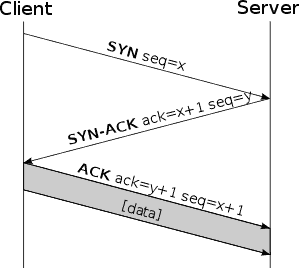
Il protocollo TCP viene utilizzato dai protocolli dei livelli superiori che necessitano di usufruire delle caratteristiche offerte dal TCP. Alcuni protocolli sono: Telnet, HTTP, SMTP, FTP-data.

**Three Way Handshake**

Un handshake a tre vie (Three-way handshake) è un metodo utilizzato in una rete TCP/IP per creare una connessione tra due host. Si tratta di un metodo in tre fasi che richiede ad entrambi gli host di scambiare pacchetti SYN e ACK (riconoscimento-acknowledgment) prima dell’inizio della comunicazione dei dati.

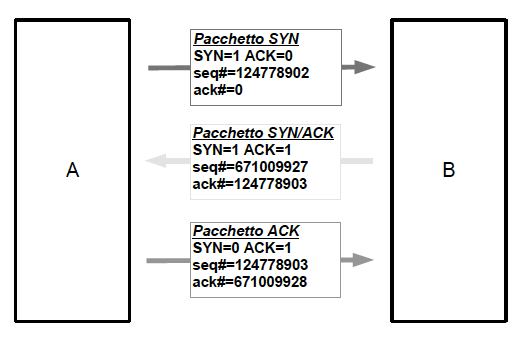
**Funzionamento**

* Un client invia un pacchetto dati SYN su una rete IP a un server sulla stessa rete o a una rete esterna. L’obiettivo di questo pacchetto è di chiedere se il server è disponibile per nuove connessioni.
* Il server di destinazione deve disporre di porte aperte in grado di accettare e avviare nuove connessioni. Quando il server riceve il pacchetto SYN dal nodo client, risponde e restituisce una ricevuta di conferma, il pacchetto ACK o SYN/ACK.
* Ilclient riceve il SYN/ACK dal server e risponde con un pacchetto ACK.
* Al termine di questo processo, viene creata la connessione e l’host e il server sono in grado di comunicare.



1. **A invia un segmento SYN a B** – il *flag SYN* è impostato a 1 e il campo *Sequence number* contiene il valore *x* che specifica l’*Initial Sequence Number* di A;
2. **B invia un segmento SYN/ACK ad A** – i *flag SYN* e *ACK* sono impostati a 1, il campo *Sequence number* contiene il valore *y* che specifica l’*Initial Sequence Number* di B e il campo *Acknowledgment number*contiene il valore *x+1* confermando la ricezione del ISN di A;
3. **A invia un segmento ACK a B** – il *flag ACK* è impostato a 1 e il campo *Acknowledgment number* contiene il valore *y+1* confermando la ricezione del ISN di B.

Il terzo segmento non sarebbe, idealmente, necessario per l’apertura della connessione in quanto già dopo la ricezione da parte di A del secondo segmento, entrambi gli host hanno espresso la loro disponibilità all’apertura della connessione. Tuttavia, esso risulta necessario al fine di permettere anche all’host B una stima del timeout iniziale, come tempo intercorso tra l’invio di un segmento e la ricezione del corrispondente ACK.



**UDP** o **User Datagram Protocol** è un protocollo senza connessione che si trova nel livello di trasporto del modello TCP/IP. Non stabilisce una connessione né controlla se l’host ricevente è pronto per ricevere o meno; invia semplicemente i dati direttamente. UDP viene utilizzato per trasferire i dati a una velocità maggiore. È meno affidabile e quindi utilizzato per la trasmissione di dati come file audio e video, VoIP

UDP non garantisce la consegna dei dati, né ritrasmette i pacchetti persi. È solo un protocollo wrapper che facilita l’applicazione nell’accesso all’IP.

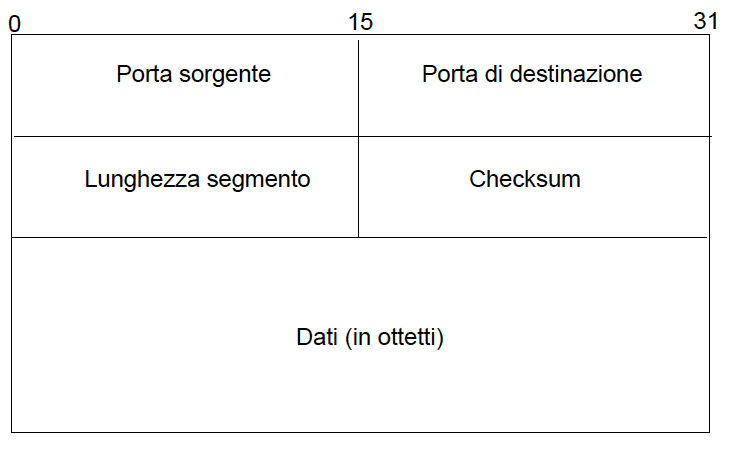
L'UDP fornisce soltanto i servizi basilari del [livello di trasporto](https://it.wikipedia.org/wiki/Livello_di_trasporto), ovvero:

* [multiplazione](https://it.wikipedia.org/wiki/Multiplazione) delle [connessioni](https://it.wikipedia.org/wiki/Connessione_(informatica)), ottenuta attraverso il meccanismo di assegnazione delle [porte](https://it.wikipedia.org/wiki/Porta_(reti));
* verifica degli errori ([integrità dei dati](https://it.wikipedia.org/wiki/Integrit%C3%A0_dei_dati)) mediante una [checksum](https://it.wikipedia.org/wiki/Checksum" \o "Checksum), inserita in un campo dell'intestazione ([header](https://it.wikipedia.org/wiki/Header" \o "Header)) del pacchetto, mentre TCP garantisce anche il trasferimento affidabile dei dati, il [controllo di flusso](https://it.wikipedia.org/wiki/Controllo_di_flusso) e il [controllo della congestione](https://it.wikipedia.org/wiki/Controllo_della_congestione).

Un datagram UDP consiste in un header di 8 byte (64 bit), seguito dai dati.

Il preambolo contiene quattro campi ciascuno della lunghezza di 2 byte (16 bit):

1. Porta sorgente (sprt) 🡪 intero a 16 bit (da 0 a 65535)
2. Porta destinazione (dprt) 🡪 intero a 16 bit (da 0 a 65535).
3. Lunghezza UDP 🡪 espressa in byte e comprende sia il preambolo che i dati
4. Checksum UDP 🡪 stringa di 16 bit utilizzata per il controllo degli errori.



Ogni datagramma UDP è inviato in un singolo datagramma IP. Quest'ultimo può venire frammentato durante una trasmissione e riassemblato dall'IP ricevente prima di essere presentato al livello UDP.

## Modello Client-Server

Nel modello client-server si distinguono due entità

* I programmi che forniscono un servizio, chiamati server
* I programmi di utilizzo, detti client che effettuano le richieste

Un server può (di norma deve) essere in grado di rispondere a più di un client.

Inoltre, distinguiamo due classi di server:

* **Concorrenti:** Un server concorrente gestisce più richieste dai client utilizzando il concetto di concorrenza, che è la capacità di eseguire più attività in parallelo.
* **Iterativi**: processa le richieste di servizio una alla volta. Possibile basso utilizzo delle risorse, in quanto non c’è sovrapposizione tra elaborazione ed I/O.

La classica implementazione di un server concorrente prevede ad ogni nuova richiesta client venga generato un **processo** che la gestisce.

TCP e UDP usano 4 informazioni per identificare una comunicazione

* Indirizzo IP del server.
* Numero di porta del servizio lato server
* Indirizzo IP del client
* Numero di porta del servizio lato client

Un endpoint è una coppia (indirizzo IP, porta).

Una connessione è una coppia di endpoints (endpoint sorgente,endpoint destinazione)