**UDP (User Datagram Protocol)**

UDP è un protocollo del livello Transport definito nell’**RFC 768**, che non prevede connessione tra host mittente e destinatario: ogni datagramma è trattato indipendentemente. Offre un servizio **Best Effort**, senza garanzie di consegna o ordine dei pacchetti, simile a quello di IP. Può essere usato direttamente da alcune applicazioni o come base per protocolli più complessi.

[Protocollo UDP - Funzionamento e Utilizzi](https://youtu.be/udnYVLw_Sx8?si=N_ZGf6-7UpGT48rX)

-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_

### **Caratteristiche principali**

* **Connectionless:** nessuna connessione preliminare.
* **Non affidabile:** non garantisce consegna o ordine.
* **Senza controllo di flusso e congestione.**
* **Multiplexing/Demultiplexing:** usa numeri di porta per gestire più comunicazioni.
* **Checksum opzionale:** verifica l’integrità (obbligatorio in IPv6).

-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_

### **Datagramma UDP**

Header di **8 byte** composto da:

* Source Port (16 bit)
* Destination Port (16 bit)
* Length (16 bit): lunghezza totale (8–65.535 byte)
* Checksum (16 bit): controllo d’integrità su header, dati e pseudo-header IP

Segue il **campo dati** (payload).  
La dimensione massima effettiva è di **65.507 byte** (65.535 totali – 8 byte header UDP – 20 byte header IP minimo).

### **UDP-Lite**

Versione introdotta nel kernel Linux 2.6.20.  
Sostituisce il campo Length con **Checksum Coverage Length**, che indica quanti byte controllare (da 8 byte dell’header fino a tutto il datagramma).  
Consente di usare parti non corrotte dei pacchetti — utile per applicazioni come **VoIP** o **streaming audio/video**.

-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_

### **Vantaggi**

* Nessun ritardo di connessione (no handshaking).
* Nessuno stato di connessione → gestione efficiente di molti client.
* Overhead minimo (header da 8 byte).
* Controllo totale da parte dell’applicazione.
* Alta velocità → ideale per applicazioni real-time.
* Supporto a **broadcast** e **multicast**.

-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_

### **Svantaggi**

* Possibile perdita o disordine dei pacchetti.
* Nessun controllo di congestione → rischio di saturare la rete.
* L’applicazione deve gestire affidabilità ed errori.
* Dimensione massima limitata del datagramma.

-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_-\_

### 

### 

### **Casi d’uso principali**

* **Streaming multimediale:** VoIP, videoconferenze, streaming live.
* **Gaming online:** comunicazioni rapide e tolleranti alle perdite.
* **Broadcast/Multicast:** IPTV, trasmissioni di gruppo.