Fondamenti_reti_di_calcolatori

TCP/IP

è un insieme di regole (protocolli) che permette ai dispositivi di comunicare su Internet e sulle reti locali. Per capirlo in modo semplice, immagina il TCP/IP come il servizio postale di Internet.

Come funziona?

TCP/IP è diviso in due parti principali:

1. **TCP (Transmission Control Protocol)** → Si occupa di spezzettare i dati in piccoli pacchetti prima di inviarli e di ricomporli nell'ordine corretto una volta ricevuti. Assicura che tutti i dati arrivino senza errori e nella giusta sequenza.
   - 📦 Immaginalo come un corriere che suddivide un pacco grande in più scatole numerate e le ricompone a destinazione.
2. **IP (Internet Protocol)** → Si occupa di trovare il miglior percorso per inviare i pacchetti attraverso la rete. Ogni dispositivo ha un indirizzo IP, che è come un indirizzo di casa.
   - 🗺️ Funziona come un navigatore GPS che decide quale strada far percorrere ai pacchetti di dati.

Esempio pratico

Quando invii un messaggio su WhatsApp:
3. Il messaggio viene diviso in pacchetti (TCP).
4. Ogni pacchetto viene inviato al destinatario seguendo il percorso più veloce (IP).
5. Una volta arrivati, i pacchetti vengono ricomposti per mostrarti il messaggio completo (TCP).

In poche parole, TCP/IP è il sistema che fa funzionare Internet, permettendo ai dispositivi di scambiarsi informazioni in modo affidabile!
-tcp/ip = è un insieme di protocolli connection oriented sicuro (uno stack), è sia implementazione iso osi che a tutti gli effetti la sua ispirazione, siccome è nato prima il tcp/ip

I 4 LIVELLI DEL TCP/IP

1. **Livello Applicazione** (Application Layer)
   - 📱 È il livello con cui interagisci direttamente: app, browser, email, ecc.
   - Usa protocolli come **HTTP (per siti web), FTP (per trasferire file), SMTP (per email)**.
   - **Esempio**: Apri un sito web con il browser (usa il protocollo HTTP).

2. **Livello Trasporto** (Transport Layer)
   - 📦 Si occupa di spezzettare i dati in pacchetti e garantirne la corretta consegna.
   - Protocolli: **TCP** (affidabile, per email, web) e **UDP** (più veloce, per videochiamate e giochi online).
   - **Esempio**: Il sito web che hai aperto viene suddiviso in pacchetti di dati (TCP).
3. **Livello Internet** (Internet Layer)
   - 🗺️ Decide il miglior percorso per i pacchetti attraverso la rete.
   - Protocolli: **IP** (indirizzi dei dispositivi), **ICMP** (ping per controllare se un dispositivo è attivo).
   - **Esempio**: I pacchetti viaggiano da un server al tuo computer attraverso la rete.
4. **Livello Accesso alla Rete** (Network Access Layer)
   - 📡 Converte i pacchetti in segnali elettrici o onde radio per trasmetterli sulla rete fisica (Wi-Fi, cavi, fibra).
   - Coinvolge tecnologie come **Ethernet, Wi-Fi, LTE**.
   - **Esempio**: Il tuo router invia il segnale via Wi-Fi al tuo computer.

🎯 Esempio pratico: Aprire un sito web

1. **Livello Applicazione** → Digiti "www.google.com" nel browser (HTTP).
2. **Livello Trasporto** → Il sito viene diviso in pacchetti (TCP).
3. **Livello Internet** → I pacchetti viaggiano attraverso la rete (IP).
4. **Livello Accesso alla Rete** → Il tuo Wi-Fi trasmette i dati al router e poi a Internet.
   Ogni livello ha il suo compito per assicurarsi che i dati partano, viaggino e arrivino a destinazione senza problemi!

modello osi

modello di riferimento architetturale per interconnessioni

Il **modello OSI** (Open Systems Interconnection) è uno schema teorico che descrive come i dati viaggiano su una rete. È diviso in **7 livelli**, ognuno con una funzione specifica.

> 📌 **Differenza con TCP/IP:**
>
> - **OSI** è un modello teorico (serve per capire il funzionamento delle reti).
> - **TCP/IP** è un modello pratico (usato davvero su Internet).

🛠️ I 7 LIVELLI DEL MODELLO OSI

1️⃣ Livello Fisico ⚡

- Trasmette i dati come segnali elettrici, onde radio o impulsi luminosi.
- Hardware: **cavi, Wi-Fi, fibra ottica, router, switch**.
- 📌 **Esempio:** Un cavo Ethernet invia segnali elettrici per trasmettere i dati.

2️⃣ **Livello Collegamento Dati** 🔗

- Organizza i dati in "frame" e corregge errori di trasmissione.
- Usa **MAC address** per identificare i dispositivi sulla rete locale.
- 📌 **Esempio:** Il tuo router invia dati al tuo PC usando l'indirizzo MAC della scheda di rete.

3️⃣ **Livello Rete** 🌍

- Indirizza i pacchetti di dati attraverso la rete.
- Protocollo principale: **IP (Internet Protocol)**.
- 📌 **Esempio:** Il tuo pacchetto viaggia tra diversi router per raggiungere Google.

4️⃣ **Livello Trasporto** 📦

- Gestisce la trasmissione affidabile dei dati.
- Protocolli: **TCP** (affidabile) e **UDP** (veloce, ma senza controllo degli errori).
- 📌 **Esempio:** TCP assicura che un'email arrivi completa e nell'ordine giusto.

5️⃣ **Livello Sessione** 🛠️

- Apre e gestisce la connessione tra due dispositivi.
- 📌 **Esempio:** Mantiene attiva una sessione di login su un sito senza farti disconnettere.

6️⃣ **Livello Presentazione** 🎭

- Converte e cripta i dati in un formato leggibile dal destinatario.
- Formati: **JPEG, MP3, PDF, HTML, SSL/TLS (crittografia)**.
- 📌 **Esempio:** Quando navighi su un sito HTTPS, i dati vengono crittografati per sicurezza.

7️⃣ **Livello Applicazione** 🖥️

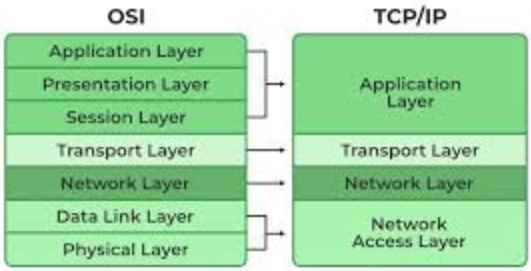
- L'interfaccia che gli utenti usano per interagire con la rete.
- Protocolli: **HTTP (web), FTP (file), SMTP (email)**.
- 📌 **Esempio:** Quando navighi su Google con il browser, usi il protocollo HTTP.

🔍 Esempio pratico: Aprire un sito web

1. **L'utente digita www.google.com → (Livello Applicazione)**
2. **I dati vengono criptati se è HTTPS → (Livello Presentazione)**
3. **Si stabilisce la connessione tra il browser e il server → (Livello Sessione)**
4. **I dati vengono divisi in pacchetti TCP → (Livello Trasporto)**
5. **Ogni pacchetto riceve un indirizzo IP per il percorso → (Livello Rete)**
6. **I pacchetti viaggiano attraverso la rete → (Livello Collegamento Dati)**
7. **I pacchetti vengono convertiti in segnali elettrici o Wi-Fi → (Livello Fisico)**

🎯 Conclusione:

Il modello OSI è una guida per capire come funzionano le reti. Anche se Internet usa **TCP/IP**, il modello OSI aiuta a visualizzare il processo di trasmissione dei dati!



-stack= è chiamata pila perché avviene o verso l'alto o verso il basso

-sopra il livello 7 del modello osi c'è l'utente, stessa cosa sopra il livello 4 del tcp-ip

indirizzamento logico: serve indirizzo ip, maschera di rete e il default getaway: cerco google, non è all'interno della mia rete quindi vado al default getaway

-LIVELLO TRANSPORT (3 livello tcp-ip) = fornisce una connessione hand to hand: da un punto all'altro direttamente (immaginiamoci come un tubo, crea una connessione diretta tra l'inizio e la fine, e posso immettere l'acqua sia da un lato che dall'altro)
LIVELLO DI TRASPORTO: permette di avere un trasporto efficace dall'host di origine a quello di destinazione, indipendentemente dalla rete utilizzata

- **mac address**= si occupa dell'indirizzo fisico, un numero di identificazione univoco che ti aiuta a rintracciare il tuo dispositivo in una rete
- connessione hand to hand tra processi su host diversi (hanno un diverso ip)
- **UDP** : non è oriented connection (non stabilisce una connessione prima di inviare dati), è semplice, utilizza gli ip ma in modo molto leggero, non ha meccanismi di controllo di flusso, non è affidabile, non garantisce il corretto invio ne la ricezione
- **vantaggi udp:** overhead minimo, minore latenza, rapidità di elaborazione elevata
- svantaggi udp: non affidabilità!
- **TCP**: è un protocollo di rete a pacchetto, si occupa del controllo della trasmissione, è oriented connection, ha delle capacità di controllo di flusso, serve a rendere affidabile la comunicazione dati in rete tra mittente e destinatario, ha caratteristiche di segmentazione, cioè può 'sminuzzare' il flusso di byte in flussi più piccoli, ovviamente è molto più lento del UDP perché deve appunto fare molte più robe
- **3 way handshak**e:La procedura utilizzata per instaurare in modo affidabile una connessione TCP tra due host è chiamata *three-way handshake* (stretta di mano in 3

passaggi), indicando la necessità di scambiare 3 messaggi tra host mittente e host ricevente affinché la connessione sia instaurata correttamente

PORTA:

- è un numero a 16 bit che identifica un endpoint di connessione, cioè identificano le applicazioni o i servizi in esecuzione su un dispositivo connesso a una rete e indirizza i dati a un servizio o processo specifico (porta 80: servizi web, comunemente detta http (hyper text transfer protocol), ssh: porta 22, https: porta 443) I numeri di porta sono compresi tra 0 a 65535 (perché 16 bit permettono 216=65.5362^{16} = 65.536216=65.536 combinazioni), i numeri da 1 a 1023 sono assegnati ai servizi e da 1024 a 65535 sono porte effimere utilizzate per identificare una fonte.
- Quando un computer invia o riceve dati, usa **un indirizzo IP** (che identifica il dispositivo) e **una porta** (che indica il servizio o l'app).
  Le porte permettono a più applicazioni di comunicare sulla stessa rete senza confondersi.
- 📜 **Classificazione delle porte**
  Le porte sono divise in tre categorie principali:
  1️⃣ **Porte Well-Known (0-1023)**
- Usate da servizi noti come **HTTP (80), HTTPS (443), FTP (21), SSH (22), SMTP (25)**.
- Sono assegnate ufficialmente da **IANA (Internet Assigned Numbers Authority)**.
  2️⃣ **Porte Registrate (1024-49.151)**
- Usate da software e applicazioni specifiche.
- Esempi: **Steam (27015-27030), MySQL (3306), BitTorrent (6881-6889)**.
  3️⃣ **Porte Dinamiche o Private (49.152-65.535)**
- Usate temporaneamente dal sistema operativo per gestire connessioni momentanee.
- Vengono assegnate in modo casuale quando un'app si connette a Internet.

/ APPLICATION LAYER: protocolli applicativi che servono a cifrare, controllare ecc, protocolli che gestiscono le sessioni, protocolli che gestiscono osi a livello applicativo.*
Con il livello application completiamo il livello URI (Uniform Resource Identifier) cioè un indirizzo espresso attraverso una stringa di caratteri per identificare una risorsa
ES: voglio accedere a 'www.google.de.' Cosa faccio?

1. *apro il browser*
2. *digito 'www.google.de' (sto richiedendo un protocollo applicativo http che generalmente sta sulla porta 80) e mi trovo di fronte a Google tedesco*
3. *cosa è successo? utilizzando il dns (Domain Name System: traduce nomi comprensibili come 'www.amazon.com' in indirizzi IP utilizzati dai computer, ad esempio 192.0.2.44, una specie di file con i nomi e gli indirizzi ai quali corrispondono), che è un protocollo applicativo/applicazione dove, dato in input un nome, ci restituisce un indirizzo, lo fa*

attraverso il network usando il layer di trasporto e una connessione hand to hand
ES:
192.168.0.17/24: indirizzo ip della macchina che stiamo utilizzando, /24 (notazione CIDR: rappresenta un indirizzo ip e un suffisso (/24) che indica i bit identificativi di rete in un formato specifico) vuol dire che se io trasformo il mio indirizzo ip in decimale puntato, otterrò 24 bit pari a 1 (i primi 24 1 e i restanti a 0) (192.168.0= identificativo rete, 17= identificativo host)
host id = 0 è l'indirizzo della rete! (192.168.0.0)
per calcolarla: network mask calculator su google
l'intervallo di indirizzo ip varia da 0 a 255: 0.0.0.0 a 255.255.255.255
maschera di rete: è utilizzata nel protocollo tcp/ip per determinare se un host si trovi su una sottorete locale o su una rete remota. /

DNS:

domain name system, è un protocollo applicativo, (livello 4) gestisce semplicemente quegli applicativi che hanno delle risoluzioni dei nomi e gli dice come comportarsi.
insomma il sistema che traduce i nomi dei siti web (come www.google.com) negli indirizzi IP (come **142.250.184.14**) necessari per connettersi ai server su Internet.
Gli esseri umani ricordano meglio i **nomi di dominio**, mentre i computer e i router usano solo **indirizzi IP** per identificare i server. Il **DNS funziona come una rubrica telefonica** che associa i nomi agli indirizzi.

Come funziona il DNS? (Passaggi semplificati)

Immagina di voler visitare www.google.com:
1️⃣ **Il tuo browser controlla la cache** 🧠

- Se hai già visitato Google, il tuo PC potrebbe ricordare l'indirizzo IP e andare direttamente al sito.
  2️⃣ **Il tuo computer chiede a un server DNS** 📞
- Se l'IP non è in cache, il PC invia una richiesta a un server DNS (di solito fornito dal tuo ISP o da servizi come Google DNS **8.8.8.8**).
  3️⃣ **Il server DNS trova l'IP corrispondente** 📋
- Se il server DNS conosce già l'IP di www.google.com, lo restituisce al tuo PC.
  4️⃣ **Il browser si connette al sito** 🚀
- Ora il PC ha l'IP giusto e può comunicare con il server di Google per caricare la pagina.

Tipi di Server DNS

Il processo di ricerca DNS può coinvolgere più server:
1️⃣ **Resolver DNS** (il primo server a cui il PC chiede l'IP)
2️⃣ **Root Server** (indirizza alla zona giusta, come ".com")
3️⃣ **Server TLD (Top-Level Domain)** (trova i domini ".com", ".it", ecc.)
4️⃣ **Server Autoritativo** (ha l'IP esatto per il dominio richiesto)

Esempio pratico

🧍 **Tu**: "Voglio visitare www.amazon.com"
📞 **DNS**: "Aspetta, controllo…"
📋 **DNS**: "Trovato! L'IP di Amazon è **176.32.103.205**"
🖥️ **PC**: "Grazie! Ora mi collego!"

Il **DNS è il "traduttore" di Internet**, che permette di accedere ai siti senza dover ricordare numeri complicati. Senza il DNS, saresti costretto a digitare gli indirizzi IP manualmente!

viaggia sulla porta 53 sia per le richieste in UDP che in TCP (UDP 53 usato per la maggior parte delle richieste DNS perchè più veloce e leggero, TCP 53 usato solo in casi particolari, tipo quando la risposta è troppo grande (oltre i 512 byte))

NOME DI DOMINIO: www. example .com, sono gestiti da degli enti che danno la possibilità ai nomi di dominio di essere online (gestiti dall'icam), sono composti da

```
###### .com = 1° livello
###### example = 2° livello (dominio)
###### www. = 3° livello (posso parlare di web anche senza www)
```

connessione end to end= da un capo all'altro

DHCP:

Dynamic Host Configuration Protocol, è un protocollo permette di assegnare automaticamente ad un dispositivo (client) che si connette alla rete, il primo indirizzo ip valido disponibile tra quelli che sono stati definiti nei parametri di configurazione
💡 **Senza DHCP**, dovresti configurare manualmente l'IP su ogni dispositivo!

Come funziona il DHCP?

Quando un dispositivo (PC, smartphone, smart TV) si collega a una rete, segue questi 4 passaggi:
1️⃣ **Discovery** 🔎

- Il dispositivo invia una richiesta ("Chi mi dà un IP?") sulla rete.
  2️⃣ **Offer** 🎁
- Il server DHCP risponde offrendo un indirizzo IP disponibile.
  3️⃣ **Request** ✅
- Il dispositivo accetta l'IP offerto.
  4️⃣ **Acknowledgment** 📩
- Il server conferma e assegna l'IP per un certo periodo di tempo (**lease time**).

Cosa assegna il DHCP?

Oltre all'**indirizzo IP**, il DHCP fornisce anche:
✓ **Subnet Mask** (per capire quali dispositivi sono nella stessa rete).
✓ **Gateway** (l'indirizzo del router per uscire su Internet).
✓ **DNS** (i server da usare per tradurre i nomi di dominio in IP).

> ## per diventare un host serve:
>
> indirizzo ip
> subnet mask
> default gateway
> dns
> nome host

Cos'è un indirizzo IP?

Un **indirizzo IP (Internet Protocol)** è un numero che identifica in modo univoco un dispositivo in una rete. È come un **indirizzo di casa**, ma per i computer e i dispositivi connessi a Internet o a una rete locale.

🌍 Tipi di indirizzi IP

Gli indirizzi IP si dividono in **due categorie principali**:
1️⃣ **IPv4 (Internet Protocol versione 4)** ⌨️

- Formato: **quattro numeri separati da punti** (es. **192.168.1.1**) (4 terzine di numeri)
- Ogni numero va da **0 a 255**.
- Totale: **4,3 miliardi di IP disponibili** → oggi quasi esauriti!
  2️⃣ **IPv6 (Internet Protocol versione 6)** 🔢
- Formato: **otto gruppi di numeri esadecimali separati da due punti** (es. **2001:0db8:85a3::8a2e:0370:7334**)
- Molto più lungo di IPv4.

- Totale: **340 trilioni di trilioni di IP disponibili!**

IP Privati vs. IP Pubblici

- **IP Privati** 🏠 → Usati nelle reti locali (LAN). Non sono accessibili da Internet.
  - Esempi: **192.168.x.x**, **10.x.x.x**, **172.16.x.x – 172.31.x.x**
- **IP Pubblici** 🌍 → Assegnati da un provider (ISP) e visibili su Internet.
  Esempio:
- Il tuo **router ha un IP pubblico** (visibile da Internet).
- Il tuo **PC e telefono hanno IP privati** assegnati dal router tramite **DHCP**.

IP Statici vs. IP Dinamici

- **IP Statico** 📌 → Non cambia mai (usato per server, telecamere di sicurezza, ecc.).
- **IP Dinamico** 🔄 → Cambia ad ogni connessione (assegnato dal provider).
  Esempio:
- Il tuo ISP ti assegna un **IP dinamico** per navigare.
- Un sito web usa un **IP statico** per essere sempre raggiungibile

ISP:

internet service provider: chiunque fornisca internet (Vodafone, Tim...)

come fa a tornare un'informazione alla mia macchina?
tramite l'ip del router e tramite il nat (effettua una lista formata da una coppia= ip:porta)

NAT:

memorizza una porta(random) e mette in piedi una connessione end to end (serve per utilizzare più indirizzi ip collegati a internet dallo stesso posto). In pratica, il NAT permette a un gruppo di dispositivi di condividere un singolo indirizzo IP pubblico per comunicare con l'esterno.

Come funziona il NAT?

Quando un dispositivo nella rete locale (LAN) invia una richiesta a Internet, il router applica il NAT, sostituendo l'indirizzo IP privato del dispositivo con un indirizzo IP pubblico. Quando la risposta torna al router, il NAT riconosce quale dispositivo nella LAN ha fatto la richiesta e inoltra correttamente la risposta a quel dispositivo.

Esempio pratico:

Immagina di avere 5 dispositivi nella tua rete locale (es. 192.168.1.2, 192.168.1.3, ecc.), ma un solo indirizzo IP pubblico (es. 203.0.113.1) per accedere a Internet.

4. **Dispositivo 1 (192.168.1.2)** invia una richiesta su Internet.
5. Il router sostituisce l'indirizzo IP privato **192.168.1.2** con l'IP pubblico **203.0.113.1**.
6. Quando la risposta torna al router, il NAT identifica che la richiesta proveniva da **192.168.1.2** e invia la risposta a quel dispositivo.

Cos'è l'HTTP?

HTTP (HyperText Transfer Protocol) è il protocollo di comunicazione utilizzato per il trasferimento di **dati** tra un **client** (come un browser) e un **server web**. È il fondamento della navigazione su Internet. Quando accedi a una pagina web, il tuo browser utilizza HTTP per richiedere i dati dal server che ospita quella pagina.

Cos'è la Subnet Mask?

La **subnet mask** (maschera di sottorete) è un valore numerico che definisce quale parte di un indirizzo IP è dedicata alla **rete** e quale parte è dedicata agli **host** (dispositivi all'interno della rete). In altre parole, la subnet mask aiuta a suddividere un indirizzo IP in due sezioni: quella che identifica la rete e quella che identifica il dispositivo specifico all'interno della rete (host). Un indirizzo IP è composto da 4 numeri (ottetti) separati da punti, come ad esempio **192.168.1.10**. Ogni numero è formato da 8 bit, quindi un indirizzo IP ha **32 bit totali**. La **subnet mask** viene utilizzata per "mascherare" (nascondere) la parte dell'indirizzo IP che riguarda l'host, lasciando visibile solo la parte della rete.

Esempio:

Supponiamo di avere un indirizzo IP **192.168.1.10** con la **subnet mask 255.255.255.0**.

- La **subnet mask 255.255.255.0** corrisponde a **11111111.11111111.11111111.00000000** in binario.
- Questo significa che i primi 24 bit (gli "1" nella subnet mask) appartengono alla **rete** e gli ultimi 8 bit (gli "0") appartengono all'**host**.

Quindi, nel caso dell'indirizzo **192.168.1.10**, la parte **192.168.1** identifica la rete e **10** identifica il dispositivo specifico (host) nella rete.

Formato delle Subnet Mask

Una subnet mask viene scritta in notazione decimale a 4 ottetti (come l'indirizzo IP). I valori più comuni per una subnet mask sono:

- **255.0.0.0** (Classe A)
- **255.255.0.0** (Classe B)
- **255.255.255.0** (Classe C)
  Questi valori determinano la dimensione della rete e il numero di dispositivi che possono essere ospitati all'interno di quella rete.

Esempi:

1. **255.255.255.0** (Classe C):
   - Permette di avere **254 dispositivi** in rete (gli indirizzi da 1 a 254).
2. **255.255.0.0** (Classe B):
   - Permette di avere **65.534 dispositivi**.
3. **255.0.0.0** (Classe A):
   - Permette di avere **16.777.214 dispositivi**.

Perché la Subnet Mask è importante?

1. **Ottimizzazione delle risorse IP**:
   La subnet mask consente di suddividere una rete in **sottoreti** più piccole, ottimizzando l'utilizzo degli indirizzi IP e migliorando l'efficienza della rete.
2. **Gestione della rete**:
   Con una buona configurazione della subnet mask, puoi separare diverse aree della rete per motivi di sicurezza o per una gestione più facile.
3. **Sicurezza e isolamento**:
   Le subnet mask permettono di **isolamento delle reti**, migliorando la sicurezza. I dispositivi all'interno di una sottorete non possono comunicare direttamente con dispositivi in un’altra sottorete senza passare attraverso il router.

Cos'è il Default Gateway?

Il **default gateway** (gateway predefinito) è un dispositivo di rete, solitamente un **router**, che funge da punto di accesso tra una rete locale (LAN) e altre reti, come **Internet** o altre sottoreti. Quando un dispositivo (come un computer, una stampante, o un altro dispositivo di rete) non sa come raggiungere una destinazione al di fuori della sua rete locale, invia i dati al **default gateway**, che si occupa di instradarli correttamente.

Come funziona il Default Gateway?

Immagina che il tuo dispositivo (ad esempio, un computer con l'indirizzo IP **192.168.1.10**) debba inviare un pacchetto a un altro dispositivo con un indirizzo IP che non fa parte della sua stessa rete locale. In questo caso, il computer non sa come raggiungere direttamente l'indirizzo

di destinazione. Ecco come interviene il **default gateway**:

4. **Controllo della rete di destinazione**:
Il computer verifica se l'indirizzo IP di destinazione appartiene alla sua stessa rete locale. Se l'indirizzo non appartiene alla sua rete (ad esempio **192.168.2.15**), invia il pacchetto al default gateway.

5. **Invio al Default Gateway**:
Il pacchetto di dati viene inviato dal computer al router, che funge da default gateway. Il router si occupa di instradare il pacchetto verso la rete di destinazione, usando le informazioni di routing appropriate.

6. **Rete esterna**:
Se la destinazione è fuori dalla rete locale (ad esempio su Internet), il gateway invia il pacchetto verso un router esterno che si occupa di trovare la strada giusta fino al destinatario finale.

Esempio pratico:

Immagina una rete domestica con questi dettagli:

- **Indirizzo IP del dispositivo**: 192.168.1.10
- **Subnet Mask**: 255.255.255.0
- **Default Gateway (Router)**: 192.168.1.1
  Se il dispositivo 192.168.1.10 deve comunicare con un altro dispositivo che si trova fuori dalla rete **192.168.1.0/24** (ad esempio, una pagina web su Internet con IP 8.8.8.8), invierà il pacchetto al **default gateway** (192.168.1.1). Il router (192.168.1.1) si occuperà di instradare il pacchetto verso la rete esterna, come Internet

Senza il default gateway, un dispositivo non sarebbe in grado di comunicare con altri dispositivi al di fuori della sua rete locale, come accedere a Internet. In pratica, è la "porta" che consente di uscire dalla rete locale e accedere a risorse esterne.