Università degli Studi di Trieste

Corso di Laurea Magistrale in
INGEGNERIA CLINICA

UNIVERSITÀ
DEGLI STUDI
DI TRIESTE

Dipartimento di
Ingegneria e Architettura

Reti di calcolatori

Corso di Informatica Medica

Docente: Aleksandar Miladinović

Livello 3 — Rete (IP e routing)

Indirizzi logici, pacchetti e scelta del percorso

Livello 3 – Network

UNIVERSITÀ DEGLI STUDI DI TRIESTE

- Si preoccupa dell'instradamento nella subnet di comunicazione
- Il suo ruolo è permettere ad un host di iniettare pacchetti in una qualunque rete
- Il livello poi si deve occupare di farli viaggiare fino a destinazione

- E' a questo livello che ad esempio è necessario determinare:
  - ✓ come identificare i nodi (**indirizzi**)
  - ✓ come gestire il **routing** (qual è la strada migliore)

- E' il livello di fatto più importante nelle reti di calcolatori
- Nella pratica si è adottata una sola strategia (offrire un servizio *connectionless*)

Definizioni

UNIVERSITÀ DEGLI STUDI DI TRIESTE

- **Node (nodo) ->** Qualsiasi entità di rete capace di inviare, ricevere o inoltrare traffico: PC, server, smartphone, stampante, **router**, **switch**, **AP Wi-Fi**, oppure anche **una singola interfaccia** di questi apparati. In pratica: se è un punto della rete che “partecipa” alla comunicazione, è un *nodo*.

- **Host ->** Un *nodo* **terminale** che **origina/consuma dati applicativi** (utente): PC, server, telefono, telecamera IP, stampante (quando riceve job), ecc. Normalmente **non** inoltra pacchetti per altri (a differenza dei router).

  Tutti gli host sono nodi; non tutti i nodi sono host (es. router/switch sono nodi, ma non host “utente”).

- **Router** — Nodo che lavora a **L3**: legge l’IP, sceglie la **rotta** e **ricapsula** nel frame del prossimo tratto (quindi **cambiano i MAC**).

- **Switch** — Nodo che lavora a **L2**: guarda il **MAC destinazione** e inoltra **solo** sulla porta corretta; **non** cambia gli IP e **non** attraversa le sottoreti.

HOST vs NODE

UNIVERSITÀ DEGLI STUDI DI TRIESTE

NODO
Qualsiasi entità che partecipa alla rete: invia, riceve
OPPURE inoltra traffico (punto di rete)

HOST (endpoint)	NODO NON-HOST (FORWARDER) (inoltra per altri)
• PC / laptop	• Router (L3)
• Server applicativo	• Switch (L2)
• Smartphone / tablet	• Access point Wi-Fi
• Stampante / telecamera IP	• Firewall / Load balancer / Proxy

Cosa fa L3

- **Indirizza** **END-TO-END** con **IP** (sorgente/destinazione finali).
- **Incapsula** i dati in un **pacchetto IP** (che viaggia dentro al frame L2).
- **Instrada** tra reti diverse usando **router** e **tabelle di routing**.

Pacchetto IP vs (Ethernet) Frame

- **Frame (L2)**: recapito **sul link** (MAC cambia per tratto) – NODE TO NODE.
- **Pacchetto IP (L3)**: recapito **tra reti** (IP resta sorgente/destinazione finali) – END TO END.
- Il pacchetto IP è il **payload** del frame.

Cos'è l'IP (Livello 3)

Indirizzo **logico** usato dai router per instradare i **pacchetti end-to-end** tra reti diverse.
È scritto nell'**header IP** (sorgente/destinazione finali) e **resta uguale** per tutto il percorso

Esempi IP:
- 192.168.1.10 (IPv4)
- 2001:0db8:85a3:0000:0000:8a2e:0370:7334 (IPv6)

IPv4

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Lunghezza: 32 bit totali.

- È composto da **4 ottetti** (8 bit ciascuno).
- Tipicamente è rappresentato in **notazione decimale puntata**, ad esempio: **192.168.1.10**

Struttura

Un indirizzo IPv4 è diviso in:

- **Parte di rete (network)**
- **Parte di host**

Spazio di indirizzi

- Possibili indirizzi: **2^{32} ≈ 4,29 miliardi**
- Per questo numero limitato, oggi **IPv4 è quasi esaurito**.

Secondo il report GreenIT (2025), ci sono circa 30,5 miliardi di dispositivi.
https://iot-analytics.com/number-connected-iot-devices

4,29 miliardi < 30,5 miliardi 【 · _ · ?】

4,29 miliardi < 30,5 miliardi 【 · _ · ?】

A prima vista sembra impossibile assegnare un IP univoco a ciascun dispositivo.

Ecco come si risolve il problema:

1) **IPv6**

   - IPv6 offre **2^{128} indirizzi**, quindi ogni dispositivo può avere un IP unico.

   !!! Tutti parlano del fatto che l'implementazione di IPv6 è pronta (da più di 20 anni)!!!

UNIVERSITÀ DEGLI STUDI DI TRIESTE

4,29 miliardi < 30,5 miliardi 【 · _ · ?】

A prima vista sembra impossibile assegnare un IP univoco a ciascun dispositivo.

Ecco come si risolve il problema:

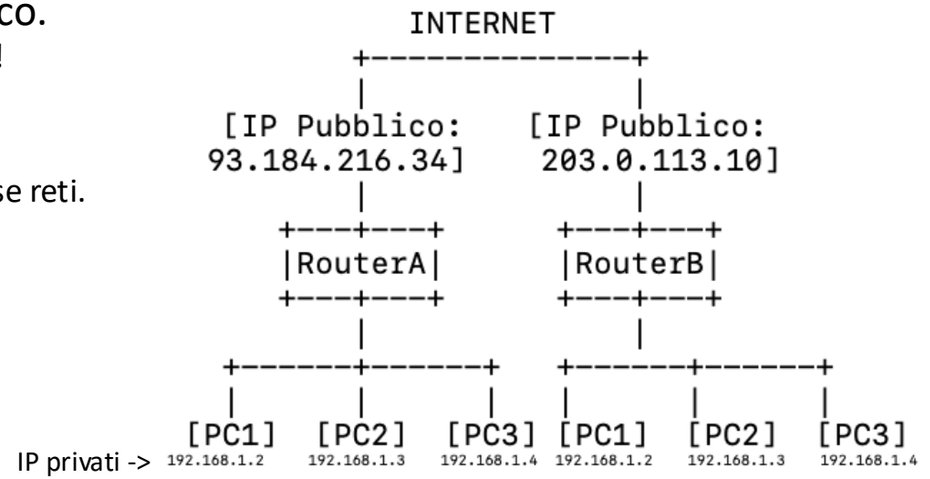
1) IPv6

IPv6 offre **2^{128} indirizzi**, quindi ogni dispositivo può avere un IP unico.

!!! Tutti parlano del fatto che l'implementazione di IPv6 è pronta (da più di 20 anni)!!!

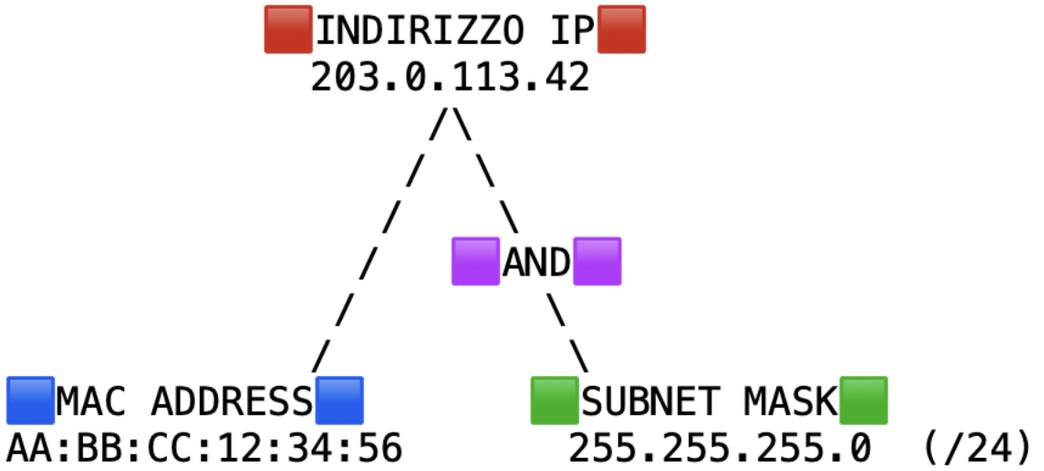
2) Avere un unico IP per accedere a Internet

- Si può creare una **rete privata** con molti dispositivi.
- Gli indirizzi dei dispositivi sono **"nascosti"** (IP privati) e **possono essere ripetuti** in diverse reti.
- Tutti i dispositivi condividono **un unico IP pubblico** per comunicare su Internet.
- Questa tecnica permette di **superare il limite degli indirizzi IPv4 disponibili**.



La triade: IP • MAC • Subnet Mask

- **IP (L3)**: indirizzo **logico end-to-end** usato dai router per instradare i pacchetti tra reti diverse.
- **MAC (L2)**: indirizzo **fisico locale al link**; serve a consegnare il frame al **prossimo hop** sul tratto corrente.
- **Subnet Mask**: “maschera” che **separa** i bit **rete** dai bit **host**; permette all’host di decidere se la destinazione è **nella mia rete** (invio diretto) oppure **fuori** (invio al **gateway**).



IP pubblico/privato e prefissi

UNIVERSITÀ DEGLI STUDI DI TRIESTE

- **Pubblici**: instradabili su Internet.
- **Privati**: 10.0.0.0/8 • 172.16.0.0/12 • 192.168.0.0/16 (non instradati su Internet).
- **prefisso**: /24 significa 24 bit di rete; si scrive **rete/prefisso**.

Subnet mask: locale o fuori?

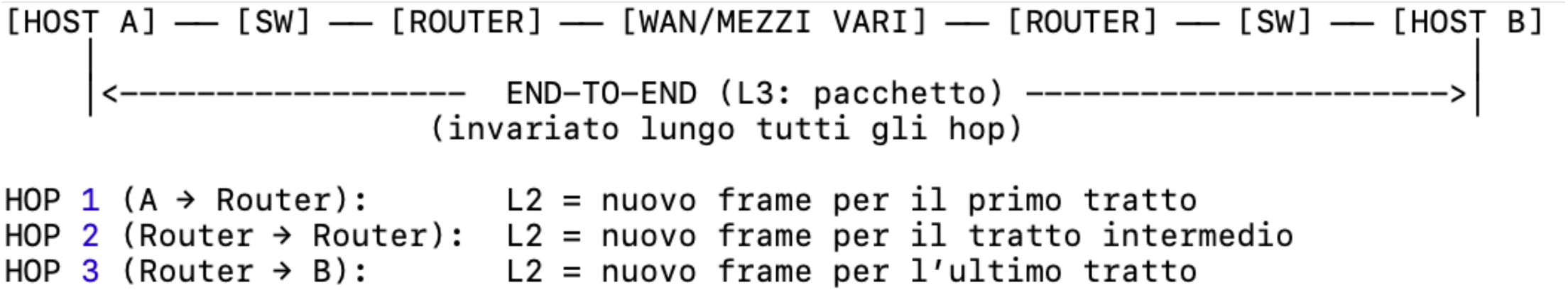
L’host calcola: **(IP_mio & mask) == (IP_dest & mask)?**

Sì → stessa sottorete → invio **diretto** (MAC del destinatario via ARP).
No → fuori sottorete → invio al **default gateway** (MAC del gateway via ARP).

Gli **IP** restano end-to-end; i **MAC** cambiano a ogni tratto.

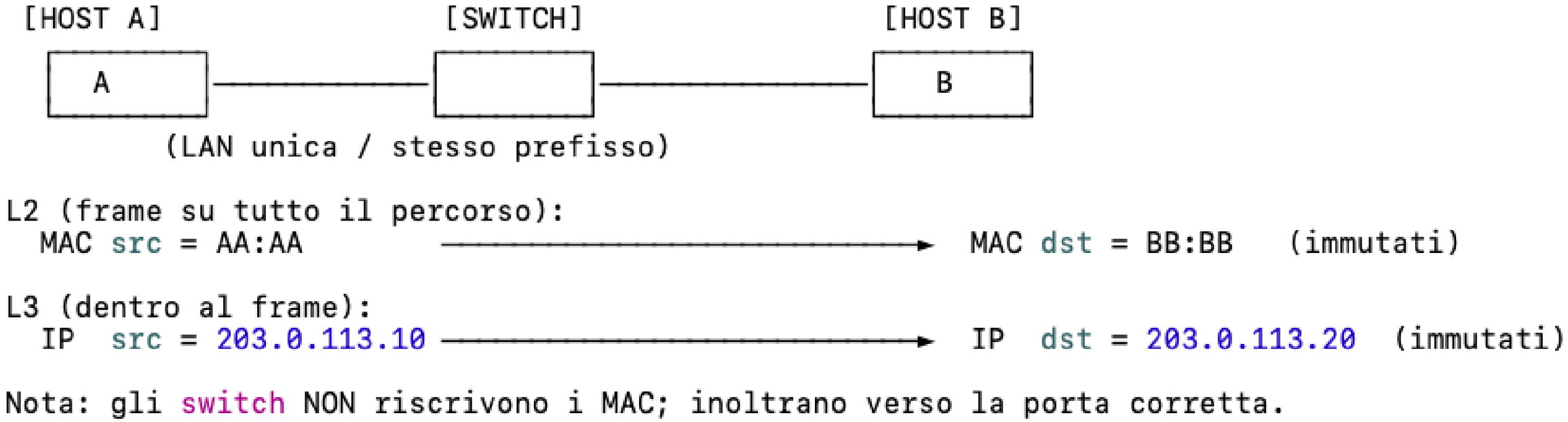
End-to-end

- **End-to-end (L3):** il **pacchetto** rimane lo stesso dalla sorgente alla destinazione.
- **Hop-by-hop/node-to-node (L2):** il **frame** viene **ricostruito** a ogni tratto quando si attraversa un **router** (nuovo link ⇒ nuova intestazione L2).

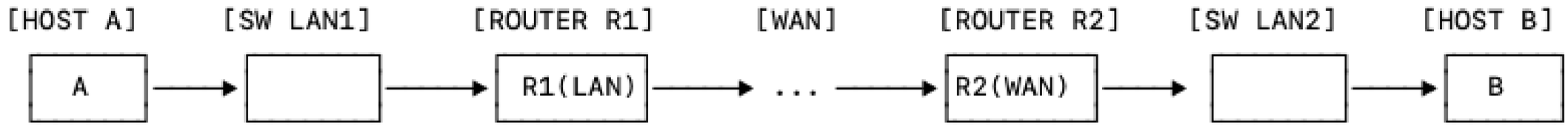


Stessa sottorete (solo switch: il MAC non cambia “in mezzo”)

Solo nella **stessa LAN**, il pacchetto (L3) è end-to-end; il **frame L2** mantiene la stessa destinazione attraverso gli switch.



Reti diverse (router in mezzo: il MAC cambia a ogni hop)




```
HOP 1 (A → R1 su LAN1)
  L2: MAC src = AA:AA          ►  MAC dst = R1L
  L3: IP  src = 203.0.113.10   ►  IP  dst = 198.51.100.20   (restano uguali)

HOP 2 (R1 → R2 sulla WAN)
  L2: MAC src = R1W            ►  MAC dst = R2W
  L3: IP  src = 203.0.113.10   ►  IP  dst = 198.51.100.20   (restano uguali)

HOP 3 (R2 → B su LAN2)
  L2: MAC src = R2L            ►  MAC dst = BB:BB
  L3: IP  src = 203.0.113.10   ►  IP  dst = 198.51.100.20   (restano uguali)

Messaggio chiave:
  • Il frame L2 vale solo per il tratto: attraversando un router, i MAC si riscrivono.
  • Gli IP (sorgente/destinazione finali) restano identici end-to-end.
```

Subnet Mask: idea in 1 riga

Una maschera è una sequenza di **1** (parte **rete**) seguiti da **0** (parte **host**).

Esempio: /24 →

255. 255. 255. 0 =

11111111.11111111.11111111.00000000

Come si usa la maschera (AND binario)

L’host calcola: **(IP_mio & mask)** e **(IP_dest & mask)**.

Se i due risultati coincidono ⇒ **stessa sottorete**; altrimenti **fuori sottorete**.

IP mio : 192.168.10.42
IP dest : 192.168.10.77
Mask (/24): 255.255.255.0

Binario (solo concetto):
IP_mio 11000000.10101000.00001010.00101010
IP_dest 11000000.10101000.00001010.01001101
Mask 11111111.11111111.11111111.00000000
AND → 11000000.10101000.00001010.00000000 (= 192.168.10.0 per entrambi)

⇒ Stessa sottorete → invio diretto (ARP sul MAC del destinatario).

IP mio : 192.168.10.42
IP dest : 192.168.10.77
Mask (/24): 255.255.255.0

Binario (solo concetto):
IP_mio 11000000.10101000.00001010.00101010
IP_dest 11000000.10101000.00001010.01001101
Mask 11111111.11111111.11111111.00000000
AND → 11000000.10101000.00001010.00000000 (= 192.168.10.0 per entrambi)

⇒ Stessa sottorete → invio diretto (ARP sul MAC del destinatario).

Subnet Mask: rete, broadcast, host validi

Rete di partenza: 192.168.10.0/24 (256 indirizzi totali)
Rete = 192.168.10.0
Broadcast = 192.168.10.255
Host validi = 192.168.10.1 … 192.168.10.254

Esempio: dividere una /24 in quattro sottoreti /26 (2 bit per subnetting)

192.168.10.0/26 host .1–.62 broadcast .63
192.168.10.64/26 host .65–.126 broadcast .127
192.168.10.128/26 host .129–.190 broadcast .191
192.168.10.192/26 host .193–.254 broadcast .255
(ogni /26 ha 62 host utilizzabili)

Esempio di subnetting (già visto in classe)

Dati di partenza:

- Rete: **192.168.10.0/24**
- Vuoi creare **4 sottoreti**

Esempio di subnetting (già visto in classe)

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Dati di partenza:

- Rete: **192.168.10.0/24**
- Vuoi creare **4 sottoreti**

Passaggi:

- /24 → 8 bit per gli host
- Servono 4 sottoreti → **2 bit** per subnetting ($2^2 = 4$)
- Nuovo prefisso: **/26**

Esempio di subnetting (già visto in classe)

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Dati di partenza:
- Rete: **192.168.10.0/24**
- Vuoi creare **4 sottoreti**

Passaggi:
- /24 → 8 bit per gli host
- Servono 4 sottoreti → **2 bit** per subnetting ($2^2 = 4$)
- Nuovo prefisso: **/26**

Sottorete	Indirizzo	Range Host	Broadcast
1	192.168.10.0/26	192.168.10.1 – 192.168.10.62	192.168.10.63
2	192.168.10.64/26	192.168.10.65 – 192.168.10.126	192.168.10.127
3	192.168.10.128/26	192.168.10.129 – 192.168.10.190	192.168.10.191
4	192.168.10.192/26	192.168.10.193 – 192.168.10.254	192.168.10.255

Router: cosa fa in pratica

- **Legge L3**, decide l'**uscita** consultando la **routing table** (longest prefix match).
- **Decrementa TTL** e aggiorna l'header (checksum IPv4).
- **Ricapsula** nel **nuovo frame L2** per il tratto di uscita (nuovi MAC).
- **Non inoltra broadcast L2** tra reti diverse.

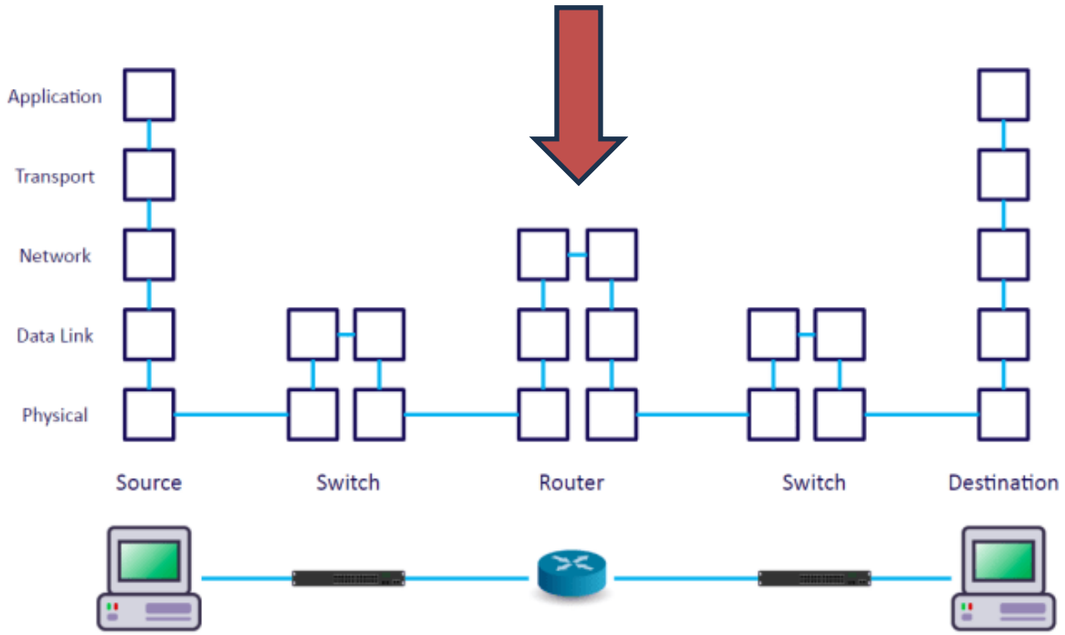


Tabella di routing: da dove arriva

UNIVERSITÀ DEGLI STUDI DI TRIESTE

- **Directly Connected** — reti delle **sue interfacce**.
- **Static routes** — inserite **a mano** dall'amministratore.
- **Routing dinamico** — scambio tra router.
- **Default route** — "qualsiasi altra destinazione" → **0.0.0.0/0** → **gateway**.

Longest Prefix Match (LPM)

Se esistono più rotte per la stessa destinazione, vince la **più specifica** (prefisso **più lungo**).

Esempio: verso 10.106.6.215

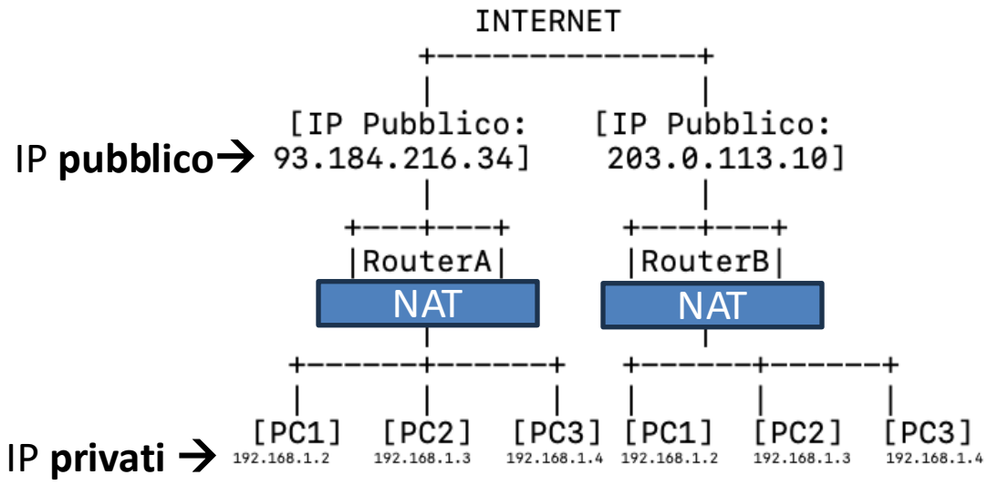
Rotta A: 10.106.0.0/16 via R1

Rotta B: 10.106.6.0/24 via R2
→ **Scelgo /24** (più specifica) → **R2**.

Detagli: https://www.youtube.com/watch?v=L6bDA5FK6gs&list=PLA9WbCkSbKM4nPaw0O92-VKcry3yYIuA7

NAT – Network Address Translation

- In LAN uso IP **privati**; “fuori” esco con **un IP pubblico** del router.
- **NAT** traduce molti IP/porte **interni** in **un solo** IP **pubblico** (porta esterna distinta).



Cosa sapere

UNIVERSITÀ DEGLI STUDI DI TRIESTE

- Distinguere **frame (L2)** vs **pacchetto (L3)**
- Capire **routing table**
- Applicare **subnetting** semplice (es. /24 → 4×/26)
- Sapere cosa fa **NAT** e quando serve.