

# **Introduzione alle reti e all'architettura TCP/IP**

# Indice dei contenuti:

## 1) Introduzione alle reti

- ↴ Topologia di Internet e topologie di rete
- ↴ I mezzi fisici
- ↴ Il software di rete: architetture a livelli
- ↴ Reti LAN: tipologie e standard
- ↴ Interconnessione di LAN: hub, switch, router
- ↴ Tecnologie di WAN: cenni

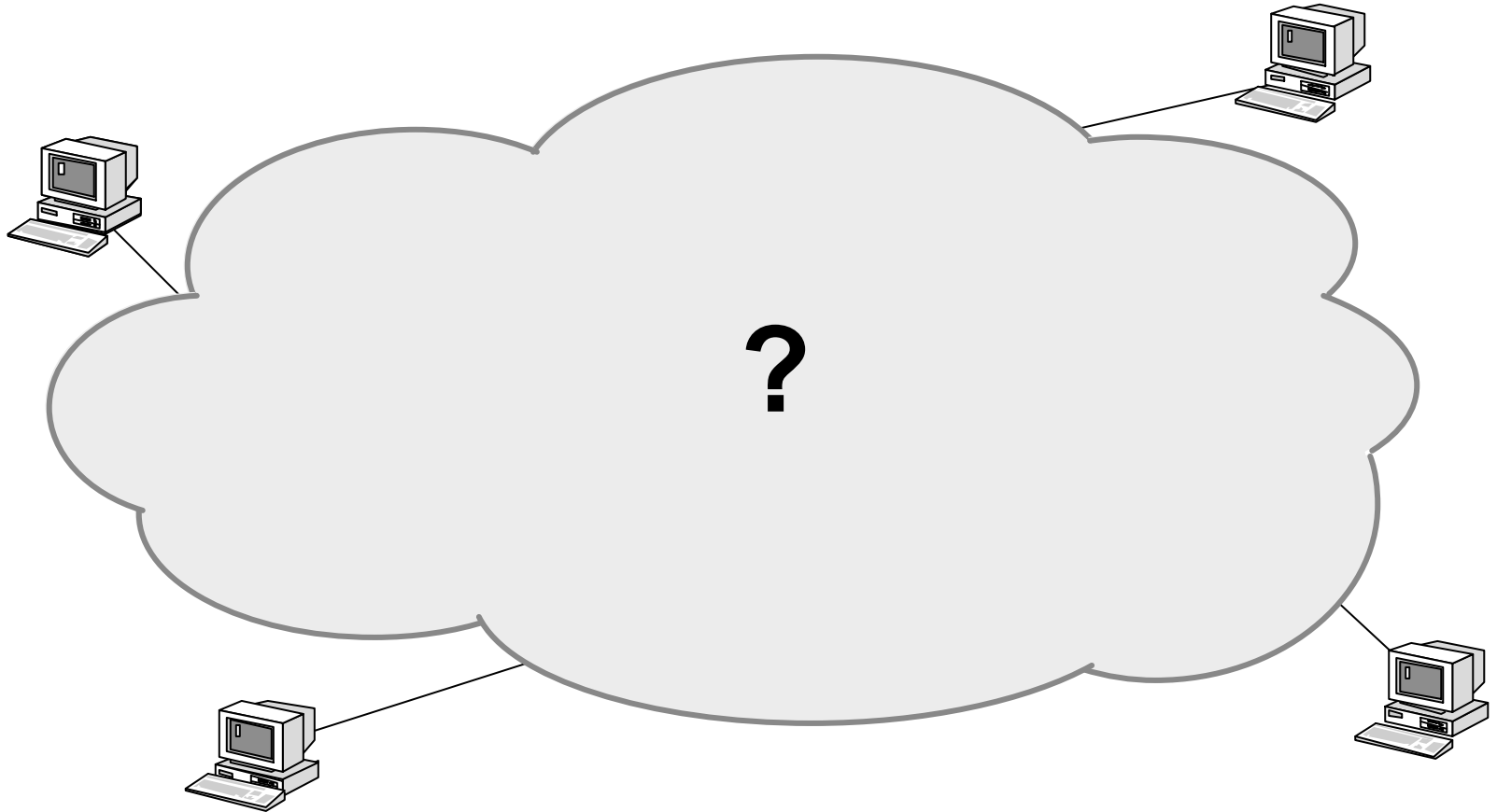
# Indice dei contenuti:

## 2) L'architettura TCP/IP

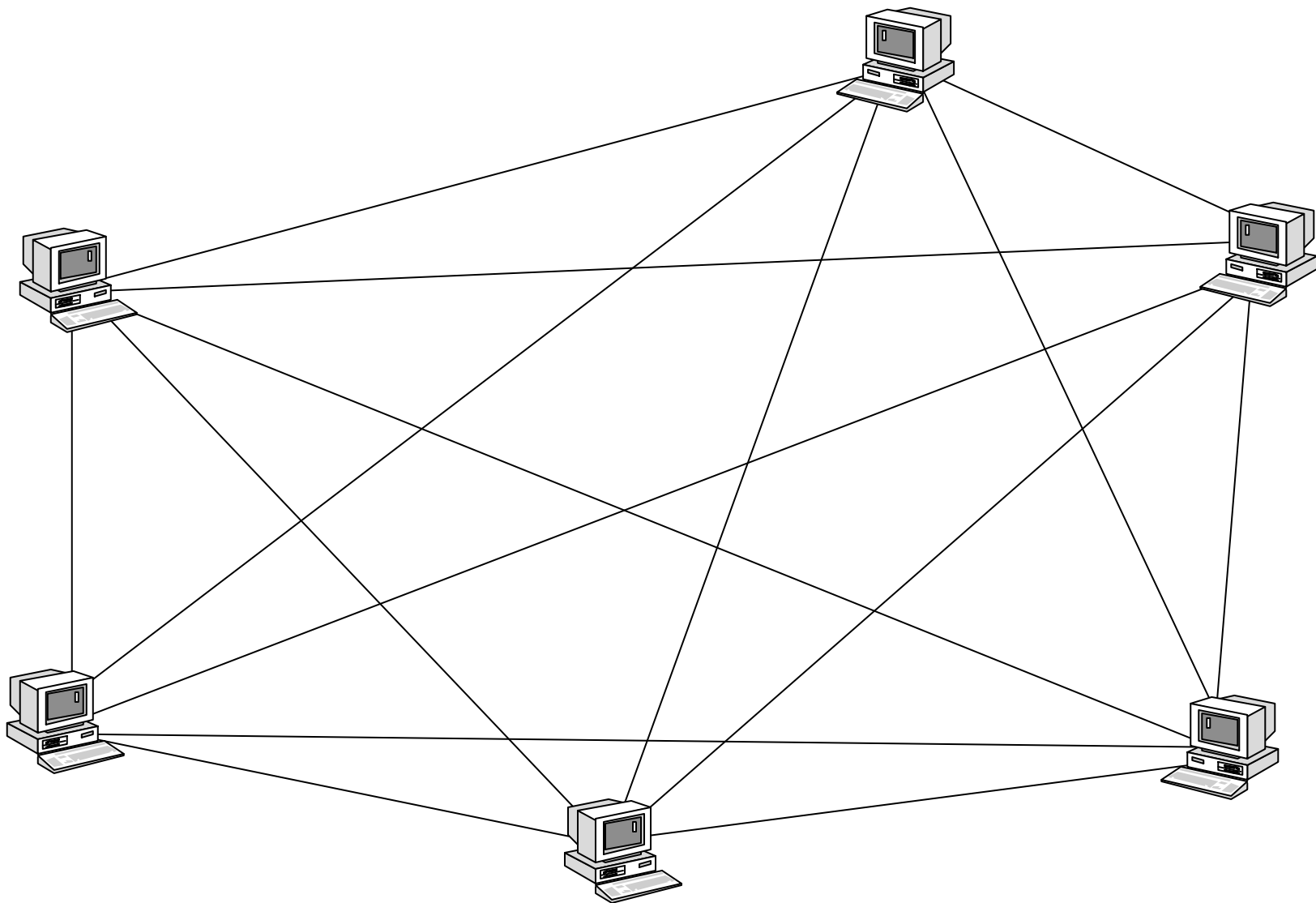
- ⇩ Un po' di storia
- ⇩ Le funzionalità del protocollo IP
- ⇩ Gli indirizzi IP
- ⇩ L'assegnamento degli indirizzi IP (*DHCP included*)
- ⇩ L'inoltro dei pacchetti IP
- ⇩ La tabella di routing
- ⇩ Concetti, algoritmi e protocolli di routing
- ⇩ ICMP: diagnosi su reti IP
- ⇩ I protocolli di trasporto: TCP, UDP
- ⇩ NAT
- ⇩ I servizi applicativi fondamentali: DNS, FTP, SMTP,....

1° PARTE  
INTRODUZIONE ALLA RETI

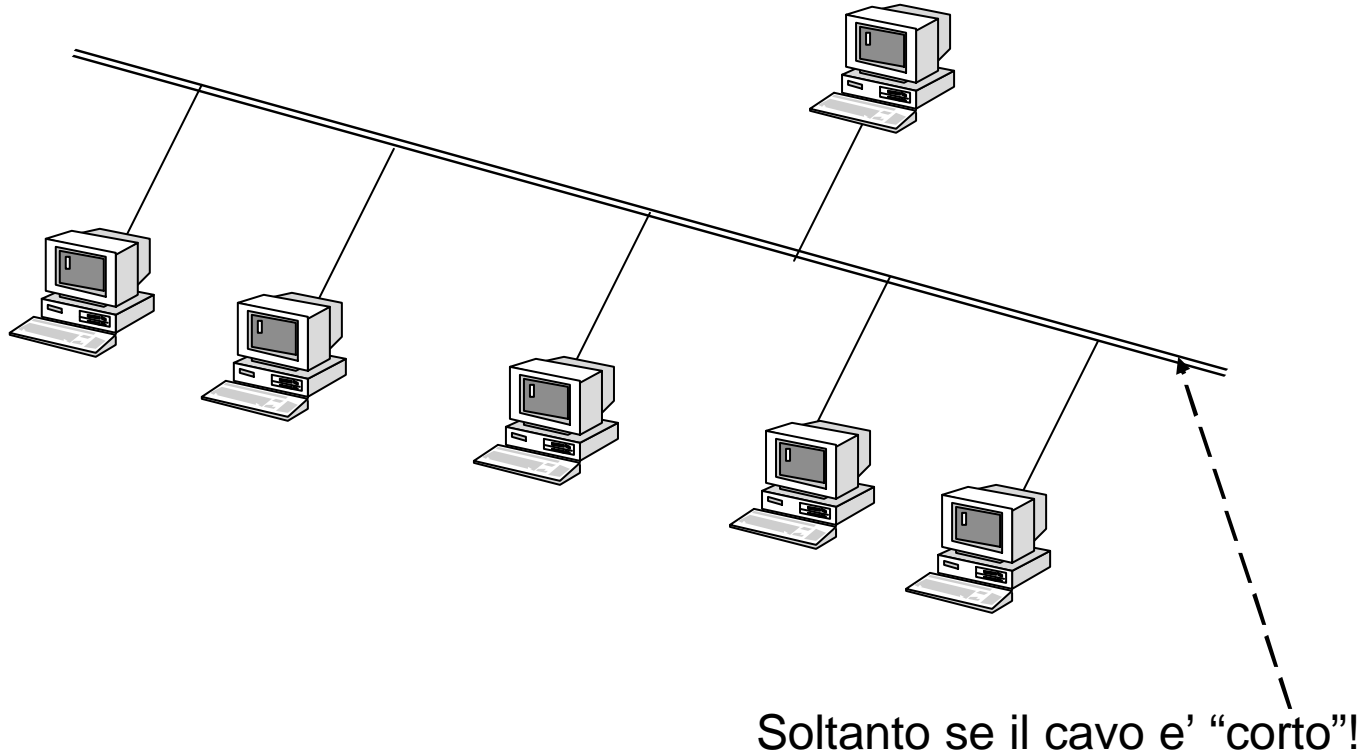
# Ma come è fatta Internet ?



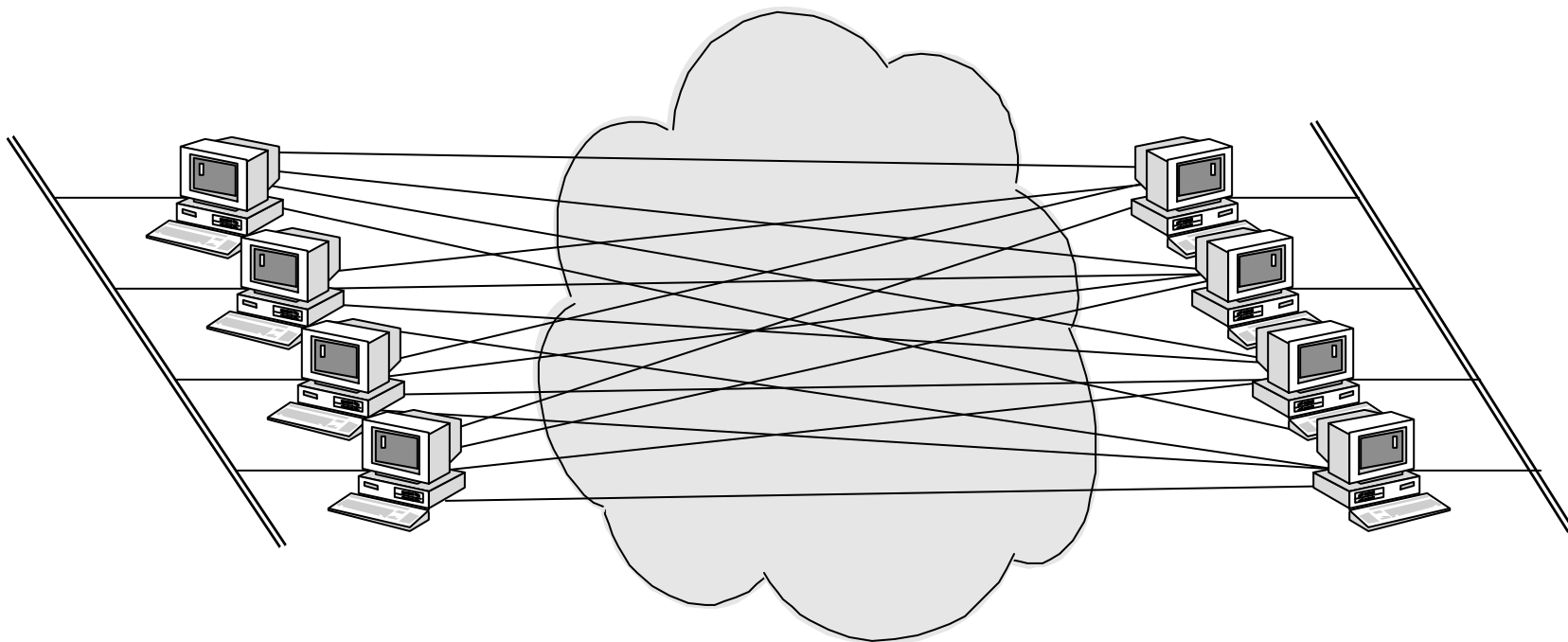
# Una rete locale ?!?



# Una rete locale !

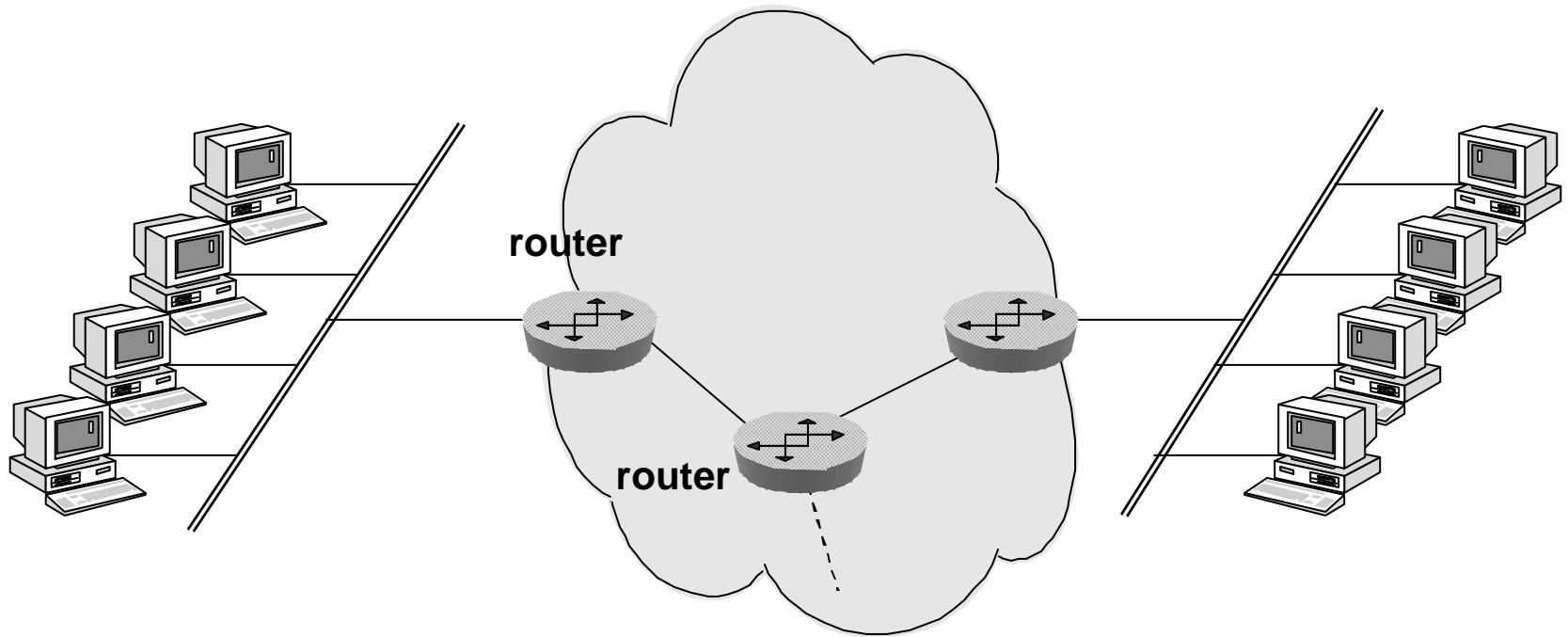


# Internetworking ?!?

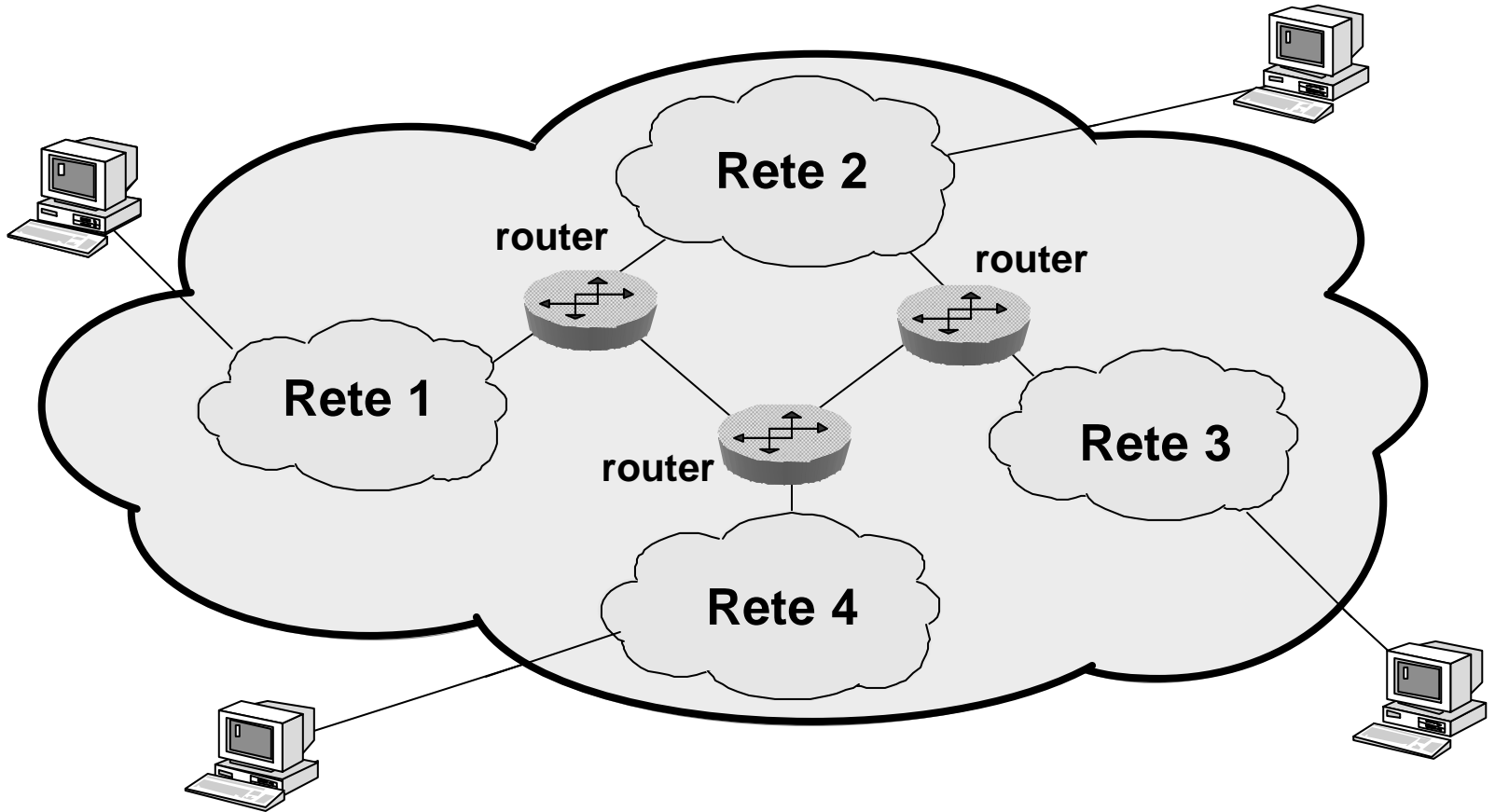




# Internetworking !

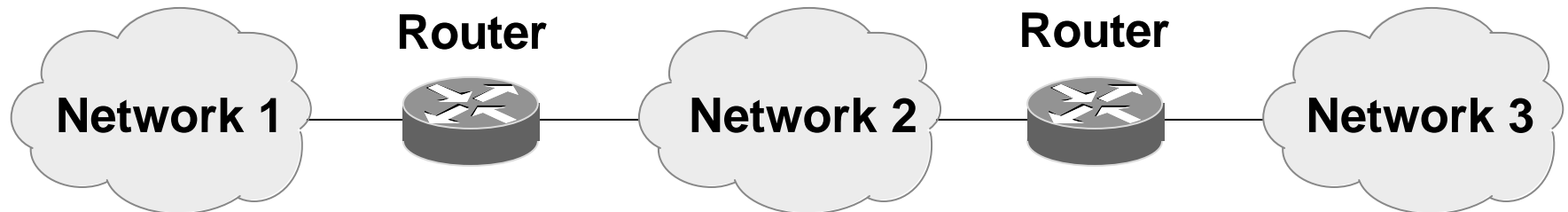


# Internet !



# Topologia di Internet

- ↴ Internet consiste di un insieme di reti, dalle dimensioni contenute e generalmente realizzate con tecnologie di LAN, interconnesse tra di loro mediante WAN (*Wide Area Networks*), che usano tecnologie pensate per i collegamenti geografici.
- ↴ A particolari apparati, detti *router*, è affidato il compito di smistare i pacchetti tra le reti interconnesse.
- ↴ L'utente di Internet percepisce questo insieme di infrastrutture eterogenee come un'unica entità.

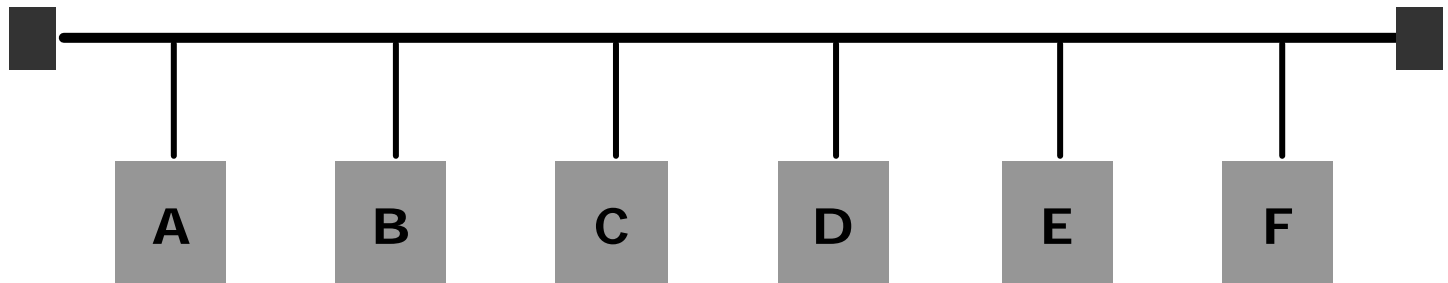


# Topologie di rete

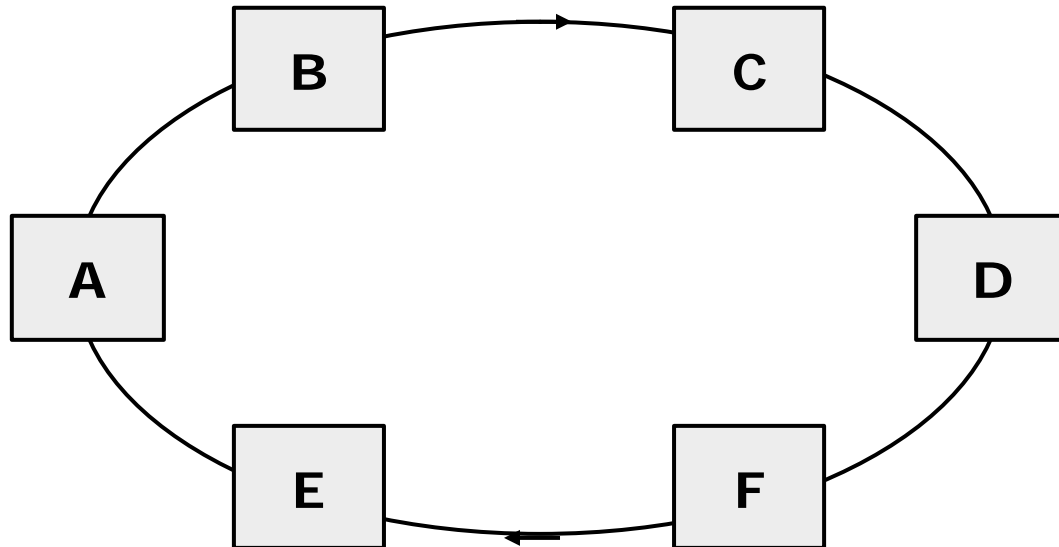
# Topologie di rete

- ⇩ Potremmo classificare le reti in base:
  - a) al modo in cui le informazioni vengono inviate sulla rete
  - b) alle dimensioni
- ⇩ Si osserva, di fatto, una correlazione tra dimensioni e modalità di scambio delle informazioni.
- ⇩ Nel caso a) parleremmo di:
  - Reti broadcast
  - Reti punto-punto
- ⇩ Nel caso b) le divideremmo in:
  - Reti locali (LAN)
  - Reti metropolitane
  - Reti geografiche (WAN)

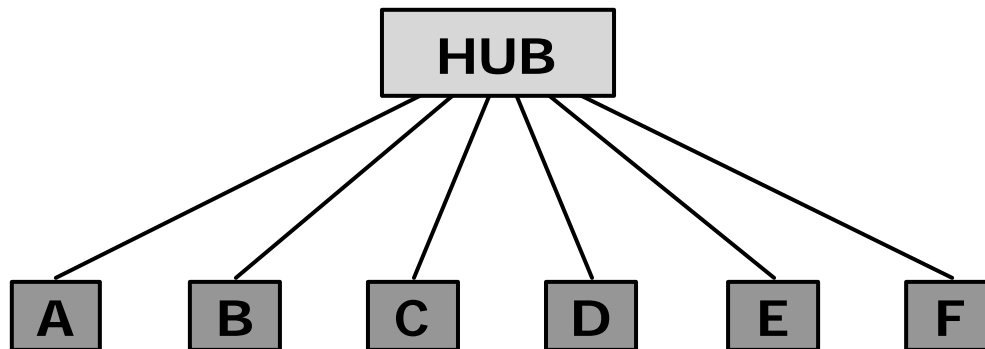
# Topologia logica a bus



# Topologia logica ad anello

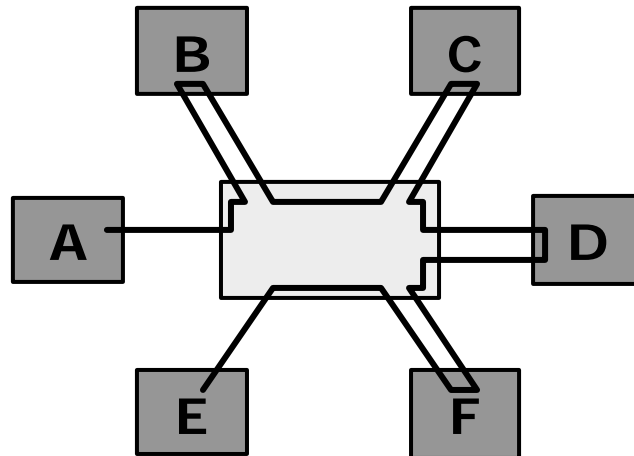
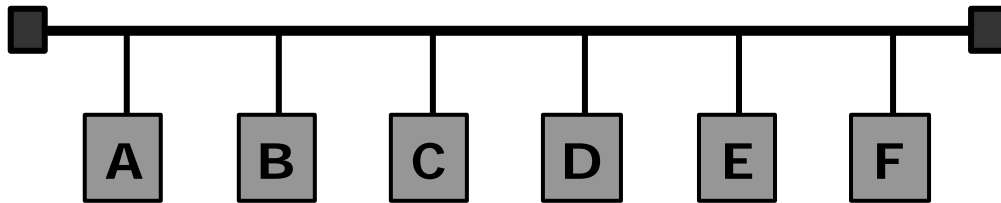


# Cablaggio a stella

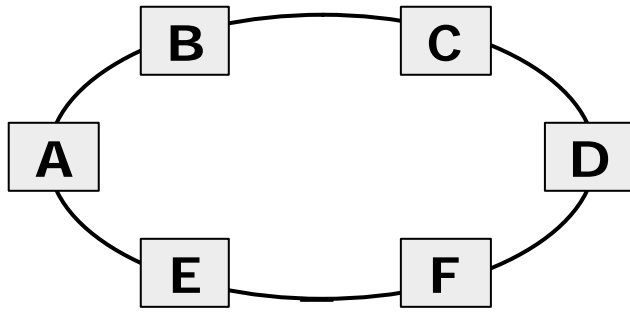




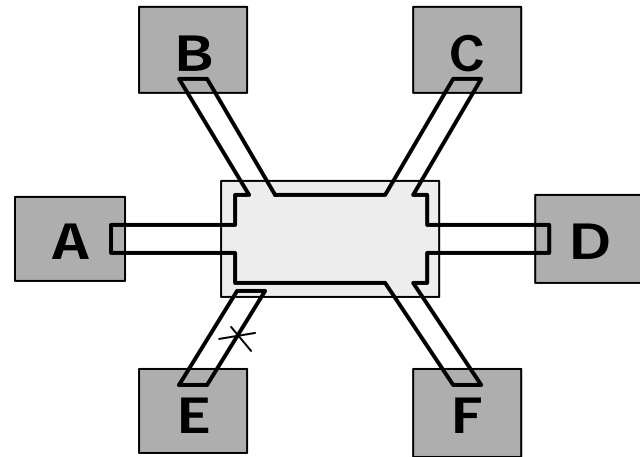
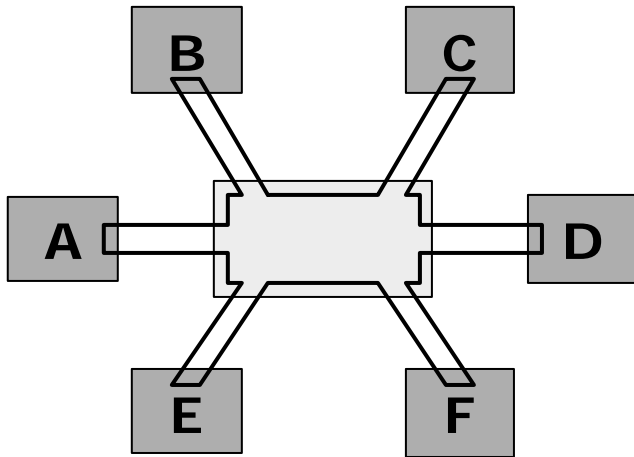
# Topologia a bus cablata a stella



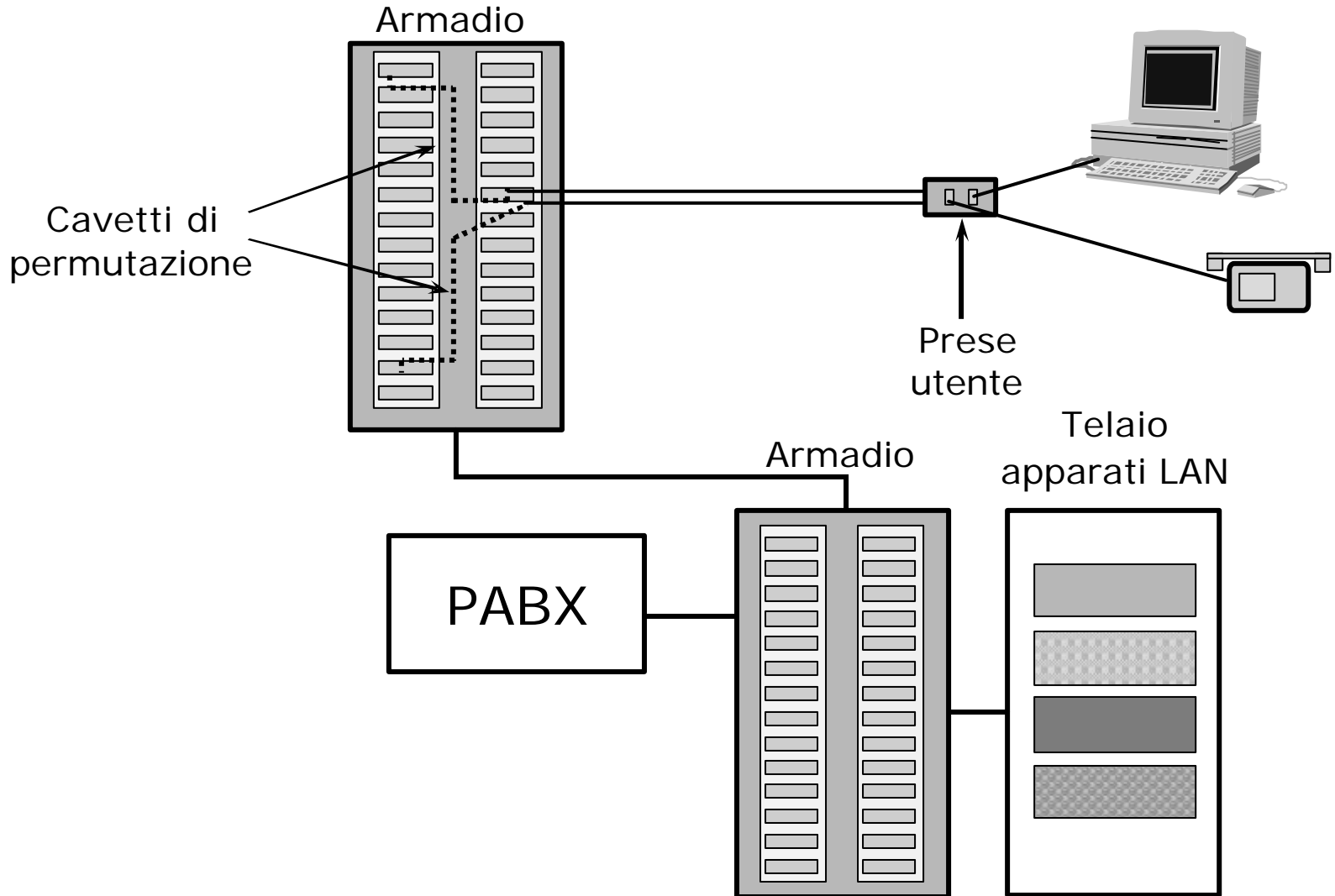
# Topologia ad anello cablata a stella



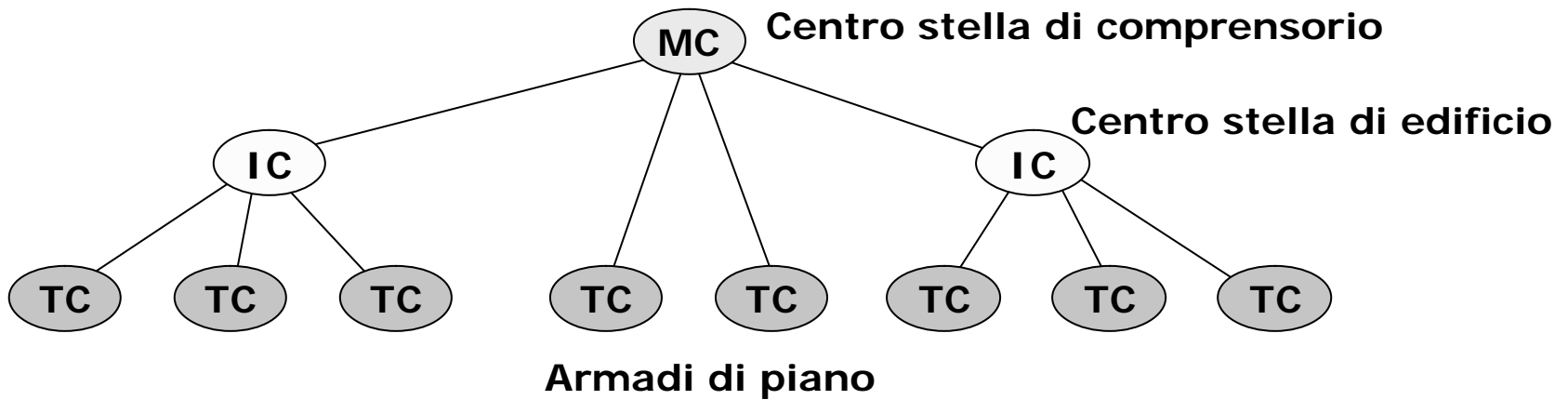
(variante: doppio anello controrotante)



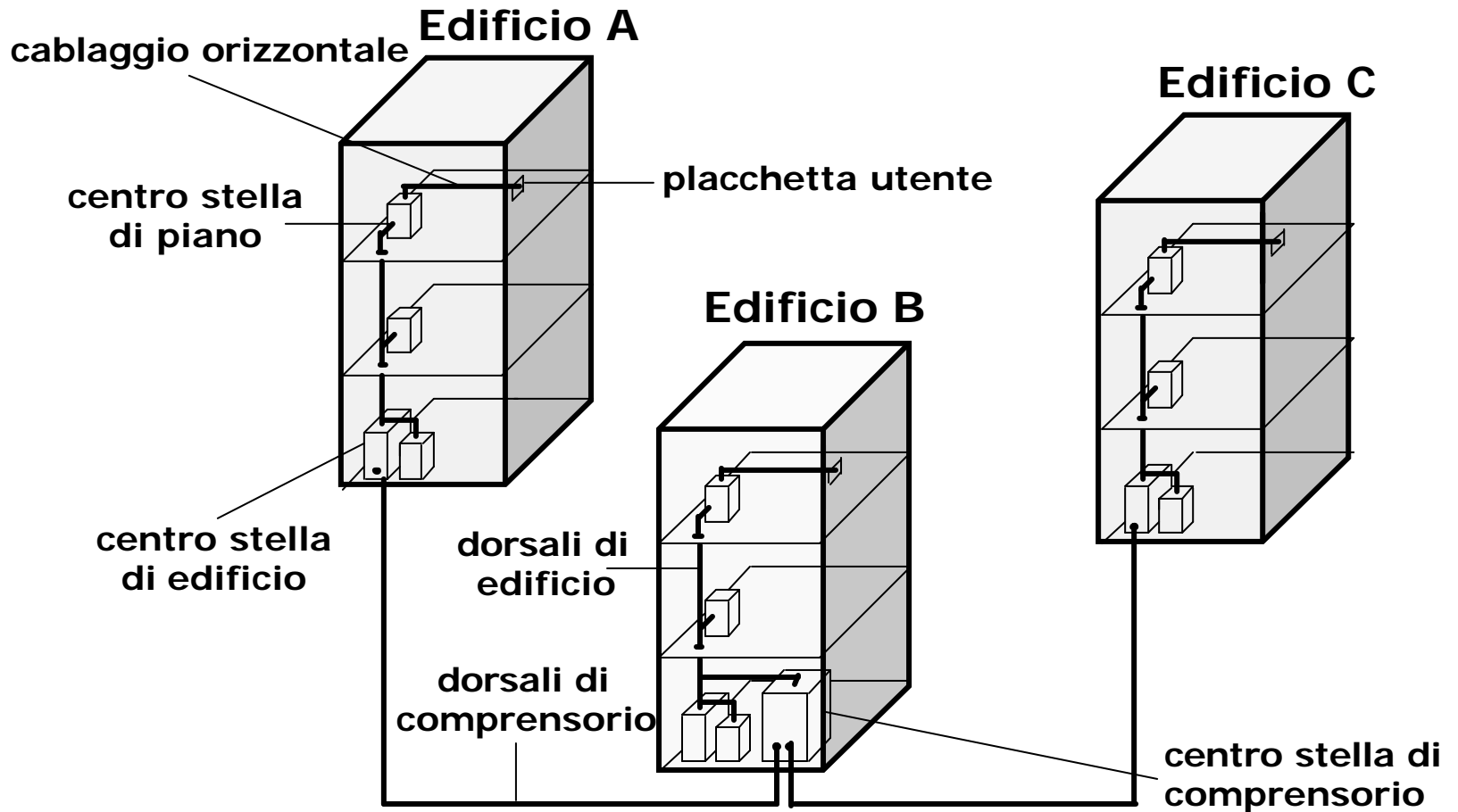
# Il cablaggio strutturato degli edifici



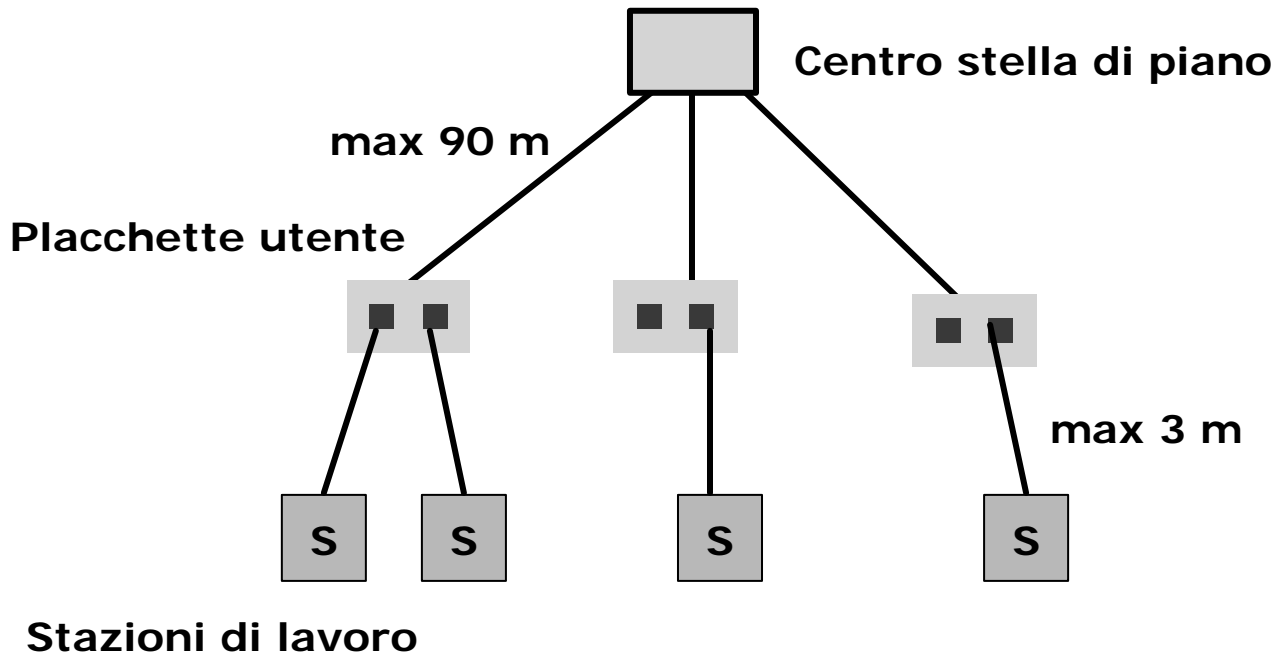
# Standard EIA/TIA 568



# Elementi del cablaggio



# Cablaggio orizzontale



Il livello fisico

# Il livello fisico

↓ Per realizzare una rete è necessario collegare fisicamente (utilizzando cioè un mezzo fisico) le apparecchiature di rete mediante opportuni mezzi trasmissivi, che sono:

- Mezzi elettrici
- Mezzi ottici
- Mezzi elettromagnetici



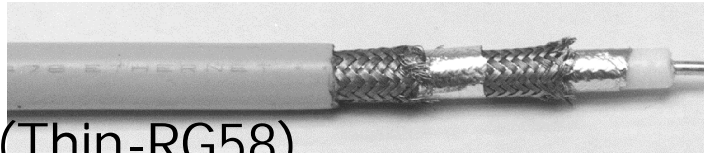
# Le “malattie” del mezzo fisico

↓ I mezzi fisici soffrono di “disturbi”; eccone alcuni:

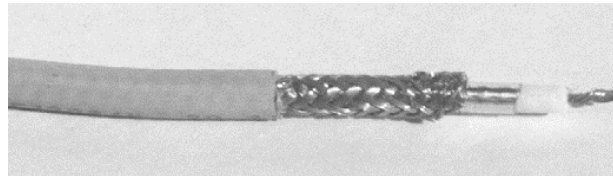
- Attenuazione
- Distorsione
- Rumore, Diafonia

# Mezzi trasmissivi: coassiale

- ⇩ Tipicamente usato per reti a BUS
- ⇩ Dominante sino al '90
- ⇩ Principali tipi:
  - Cavo grosso (Thick-RG213)

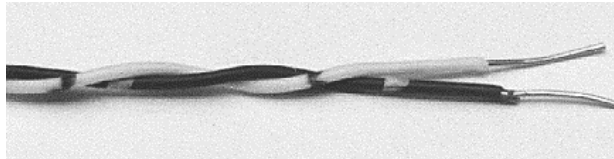


- Cavo sottile (Thin-RG58)



# Mezzi trasmissivi: coppia simmetrica

- ⇩ Prestazioni inferiori al cavo coassiale
- ⇩ Utilizzabili anche per bit rate elevati ( $> 100$  Mbit/s) su brevi distanze ( $\sim 100$  m)
- ⇩ Basso costo e facilità di posa
- ⇩ Adatto a cablaggi strutturati
- ⇩ Enorme diffusione dal '90



# Mezzi trasmissivi: coppia simmetrica

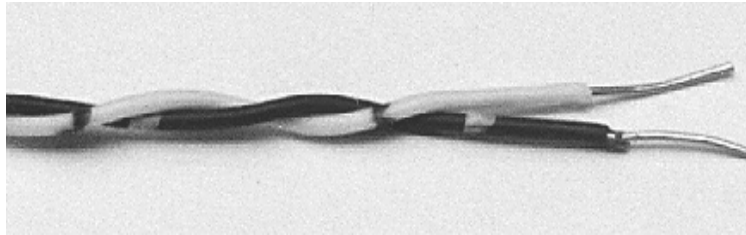
## ⇩ Varianti

- UTP (Unshielded): non schermato
- STP (Shielded): schermato coppia per coppia
- FTP (Foiled): uno solo schermo per tutto il cavetto

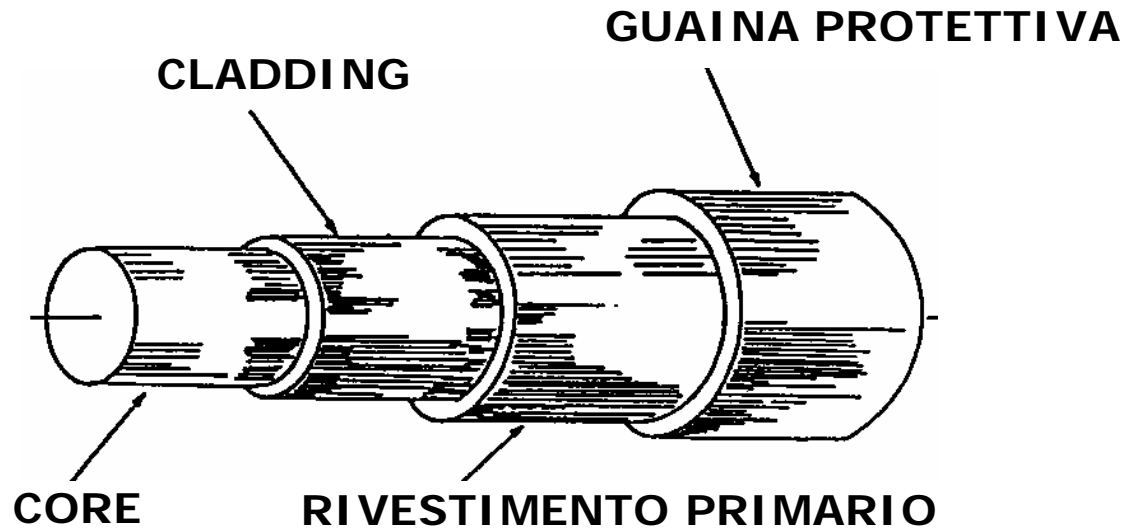
## ⇩ Categorie dei cavi

- 1 - telefonia analogica
- 2 - telefonia numerica (ISDN) e dati a bassa velocità
- 3 - dati sino a 16 MHz di banda
- 4 - dati sino a 20 MHz di banda
- 5 - dati sino a 100 MHz di banda
- 6 - dati sino a 300 MHz di banda

# Mezzi trasmissivi: coppia simmetrica



# Mezzi trasmissivi: fibra ottica



# Mezzi trasmissivi: fibra ottica

- ⇓ Insensibilità al rumore elettromagnetico
- ⇓ Mancanza di emissioni
- ⇓ Bassa attenuazione
- ⇓ Banda passante teoricamente illimitata (fibre monomodali)
- ⇓ Basso costo della fibra
- ⇓ Alto costo per interfacce e connettorizzazioni
- ⇓ Campi di impiego:
  - altissima velocità ( $>150$  Mbit/s)
  - lunghe distanze di interconnessione
  - ambienti con problemi di compatibilità elettromagnetica

# Mezzi trasmissivi: fibra ottica

## ↓ Fibre multimodali

- prestazioni inferiori
- costo più alto
- interfacce relativamente poco costose

## ↓ Fibre monomodali

- prestazioni più elevate
- costo della fibra più basso
- interfacce più costose
- maggiori difficoltà di connettorizzazione



# Tecniche di codifica

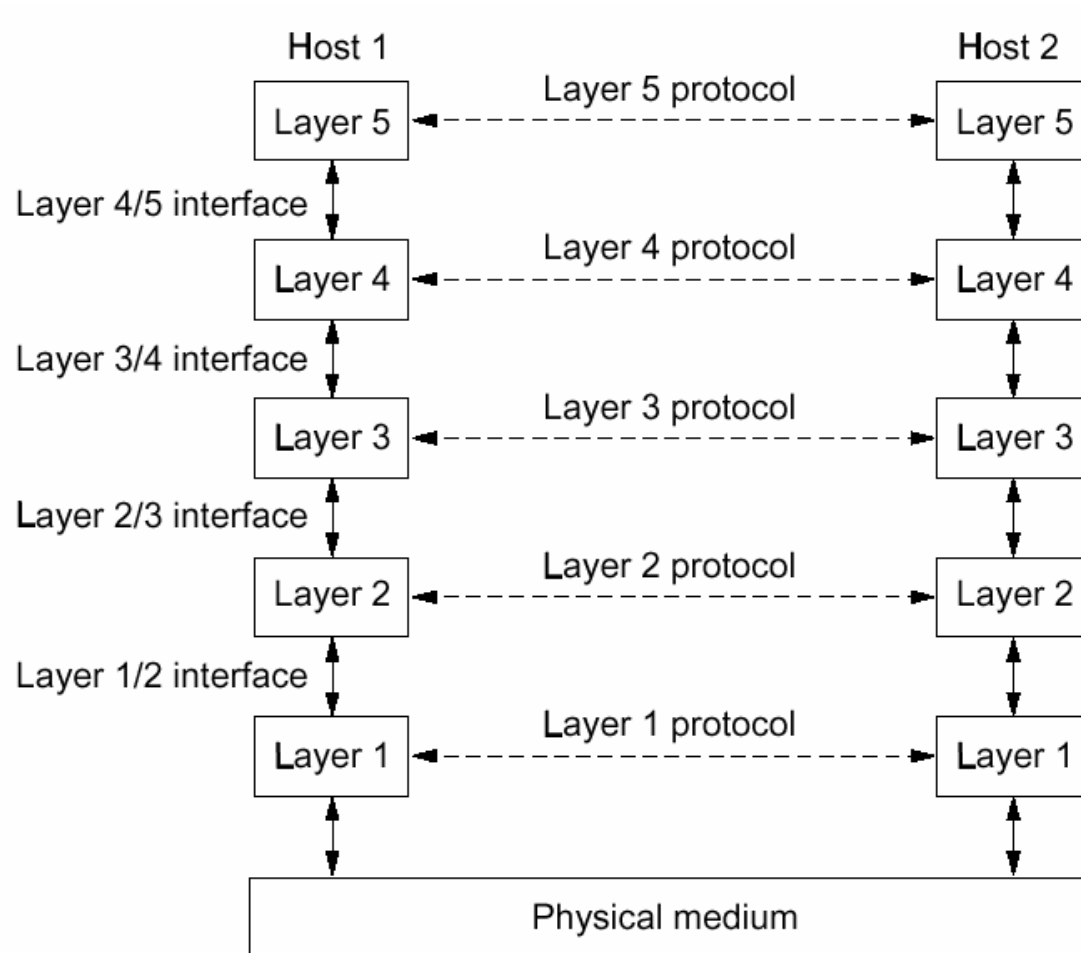
- ⇓ Sono necessarie perché le grandezze fisiche coinvolte nella trasmissione (elettriche, ottiche, elettromagnetiche), che sono analogiche, vengono utilizzate per veicolare informazioni di natura discreta (binarie).
- ⇓ Le informazioni "digitali" verranno associate a particolari valori delle grandezze fisiche.
- ⇓ Es. una semplice codifica consiste nell'associare due particolari valori della grandezza fisica scelta ai valori logici 0 ed 1.

# Architetture a livelli

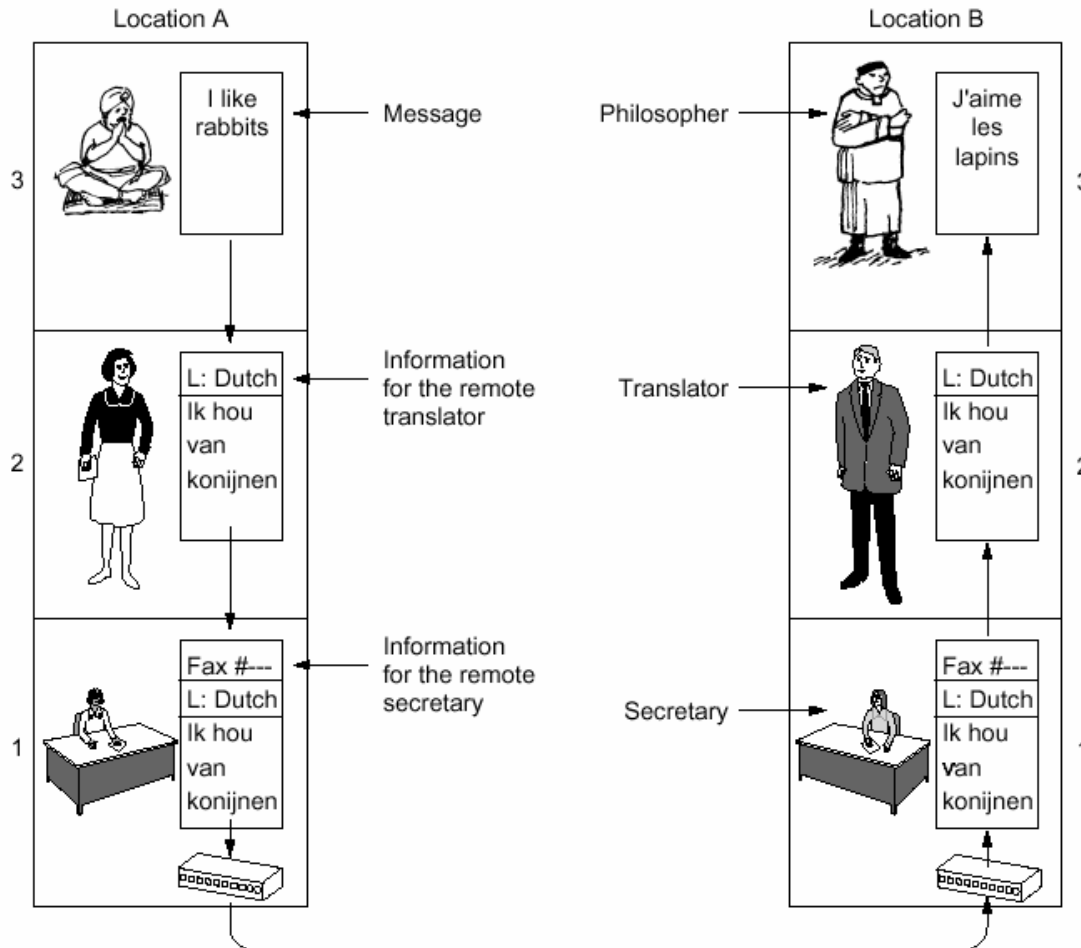
# Architetture di rete modulari

- ⇓ Trattandosi di software complesso, anche a quello che implementa le funzioni di comunicazione tocca in sorte uno sviluppo modulare.
- ⇓ Le regole della buona progettazione del software di comunicazione lo vogliono suddiviso in livelli, a ciascuno dei quali vengono assegnati compiti precisi
- ⇓ Il dialogo tra i vari livelli di una stessa architettura è definito dalle interfacce
- ⇓ Il dialogo tra i livelli pari grado su architetture differenti è regolato dai protocolli

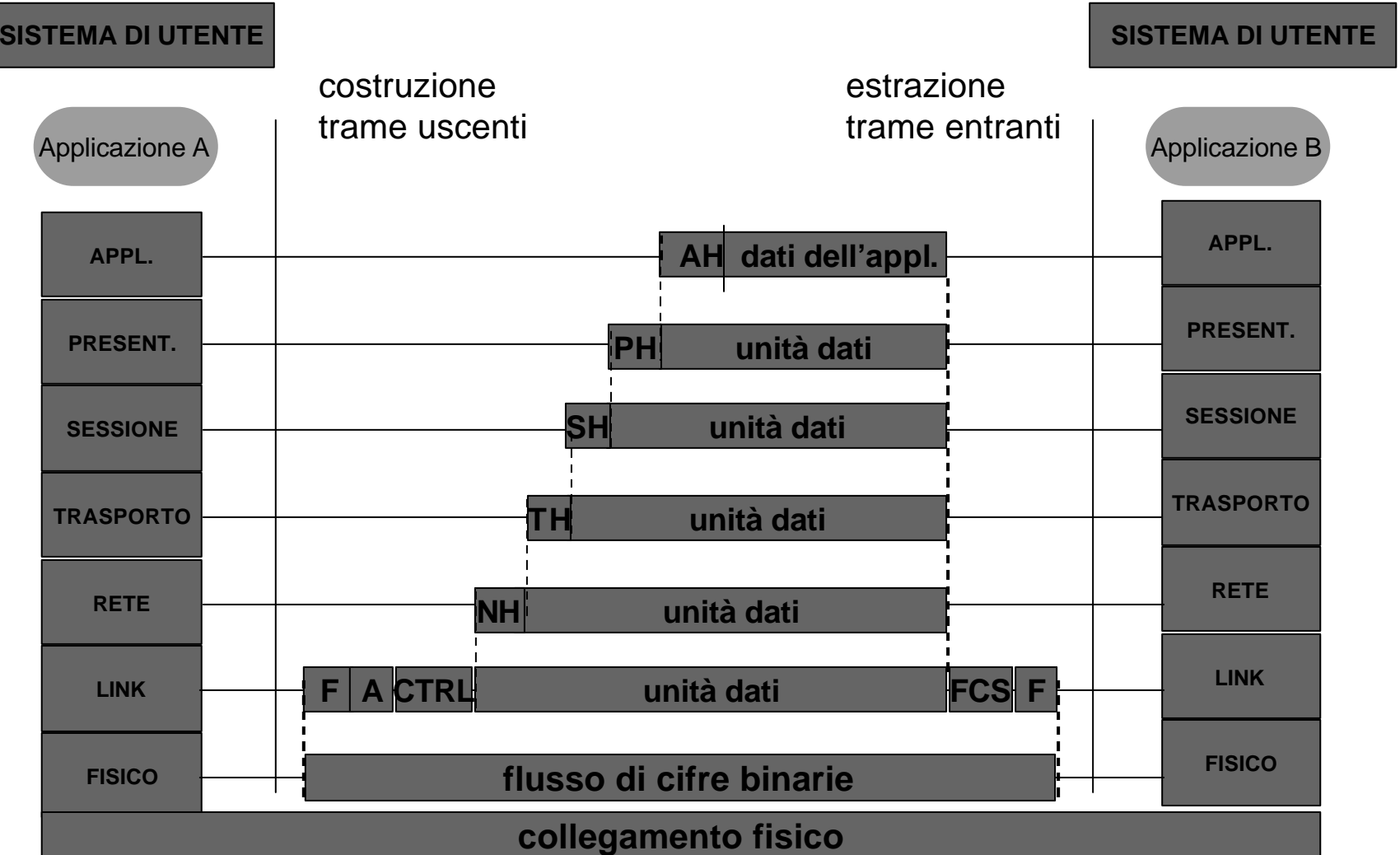
# Protocolli e interfacce



# Analogia



# Protocolli



# I 7 strati del modello OSI (1)

**Open System A**

**Open System B**



# I 7 strati del modello OSI (2)

**Open System A**

**Open System B**



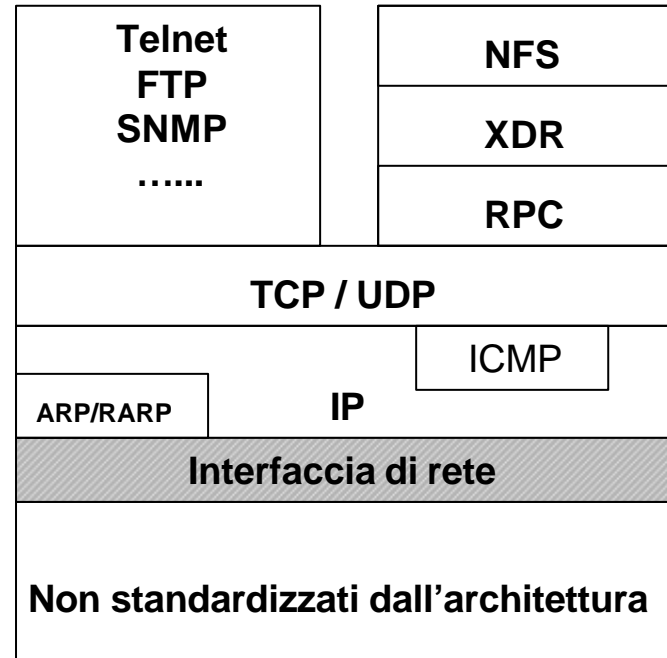


# Confronto OSI – TCP/IP

## OSI

<b>Applicazione</b>
<b>Presentazione</b>
<b>Sessione</b>
<b>Trasporto</b>
<b>Rete</b>
<b>Collegamento</b>
<b>Fisico</b>

## TCP/IP



Reti locali

# Prima di parlare di reti locali...

↓ Diciamo qualcosa sul livello Data Link,

- che svolge funzioni di primaria importanza in tutte le architetture di rete,
- ed in particolare in quelle che utilizzano modalità trasmissive di tipo *broadcast*.

# Data Link

↓ Al livello Data Link sono in genere demandate funzioni importanti come:

- il compito di fornire al livello Network soprastante diverse classi di servizio
- il raggruppamento dei bit in trame
- la gestione degli errori di trasmissione
- il controllo del flusso dei pacchetti (in modo che un ricevente lento non sia inondato da un mittente veloce)

# Data Link: *framing*

- ↴ L'impacchettamento del flusso di bit inviati sul canale fisico consente al ricevente di ricostruire il messaggio (non sarebbe altrimenti possibile stabilirne la lunghezza).
- ↴ La sequenza di bit, prima di essere inviata sul canale, viene suddivisa in trame, di cui viene calcolato il *checksum*, che viene inviato insieme alla trama corrispondente.
- ↴ Il ricevente ha il problema di individuare l'inizio e la fine di ciascuna trama.
- ↴ Il problema viene risolto delimitando le trame con particolari sequenze di bit

# Data Link: *error checking*

- ↓ La sequenza di controllo (*checksum*) accodata alla trama permette al ricevente di validarla, accorgendosi di eventuali alterazioni.
- ↓ E' possibile utilizzare sia codici di controllo che consentono il rilevamento di eventuali errori di trasmissione, sia codici che ne consentono anche la correzione.
- ↓ Ma, se il pacchetto spedito non arriva per niente? Anche il fatto di prevedere dei riscontri in vista di una connessione affidabile risulterebbe inutile!
- ↓ Il problema viene risolto introducendo un *timer* a livello Data Link: il mittente attende un tempo sufficiente l'ACK prima di ritrasmettere il pacchetto.

## Data Link: *flow control*

- ⇓ Tutti i meccanismi di controllo di flusso richiedono, tipicamente, una qualche forma di riscontro da parte del ricevente che informi il mittente sul suo stato e sulle sue capacità di ricezione (“Adesso mi puoi inviare N pacchetti, poi fermati fino a nuove disposizioni”)

# Ed ora... la definizione!

- ↓ Una **LAN** è un sistema di comunicazione che permette
- ad apparecchiature indipendenti di comunicare tra di loro
  - entro un'area delimitata utilizzando
  - un unico canale fisico a velocità elevata e
  - con basso tasso di errore

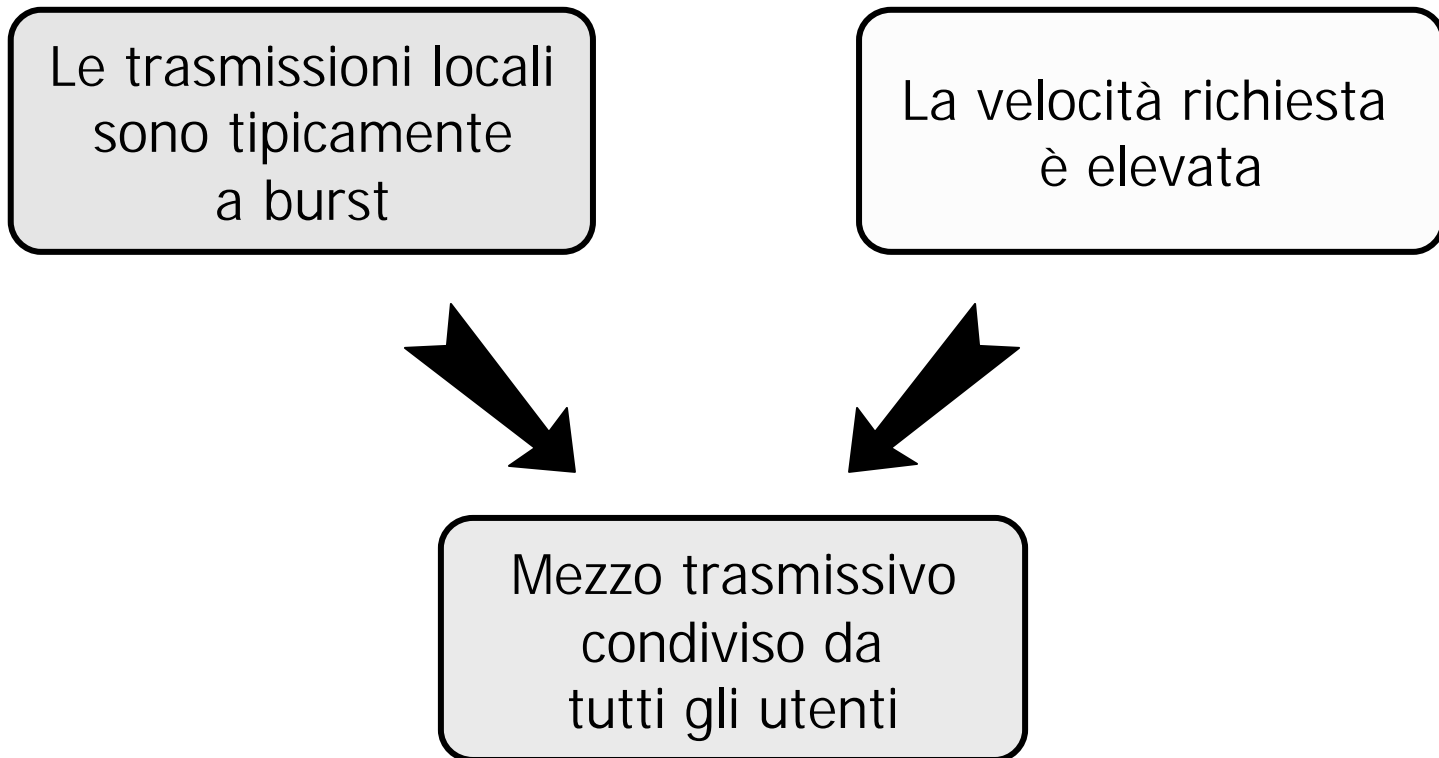


# LAN: conseguenze a livello 2

- ↴ Quando un sistema trasmette diventa temporaneamente proprietario dell'intera capacità trasmissiva della rete
- ↴ La trasmissione è sempre di tipo broadcast: un sistema trasmette e tutti gli altri ricevono
- ↴ E' necessaria la presenza di indirizzi per stabilire chi siano il reale destinatario e il mittente della trasmissione
- ↴ Occorre un meccanismo per arbitrare l'accesso al mezzo trasmissivo
- ↴ Il livello Data Link di una LAN deve farsi carico anche di questi problemi, oltre a quelli già evidenziati

# Mezzo trasmissivo condiviso

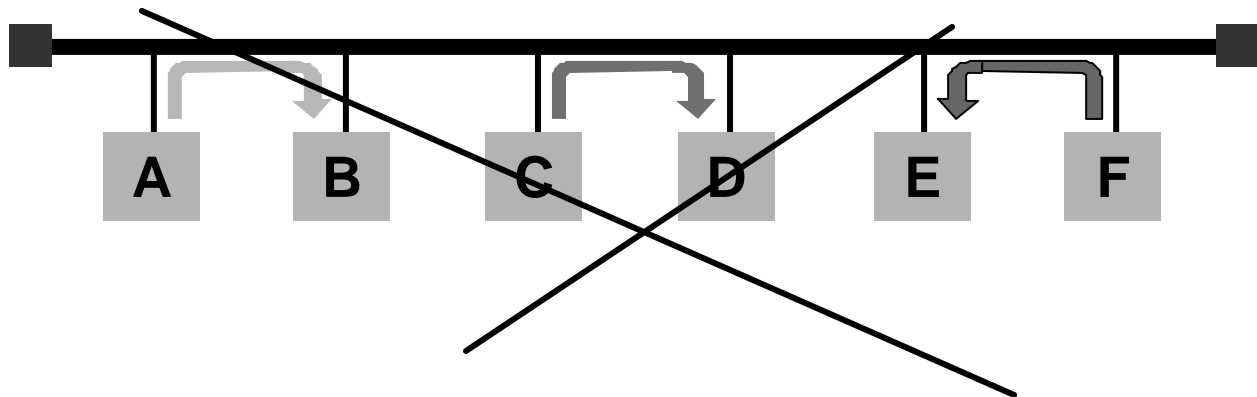
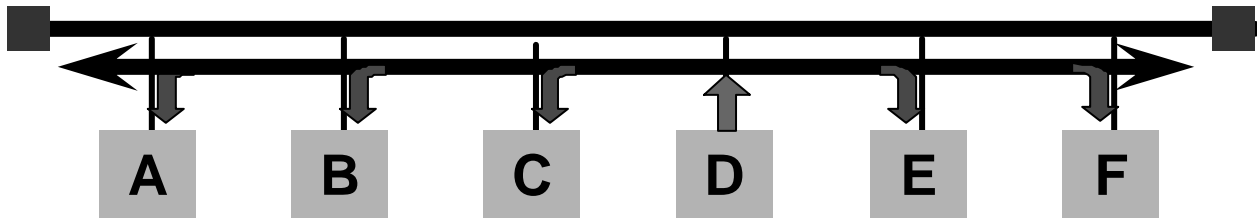
↓ Le LAN hanno un solo canale trasmissivo condiviso nel tempo da tutti i sistemi collegati



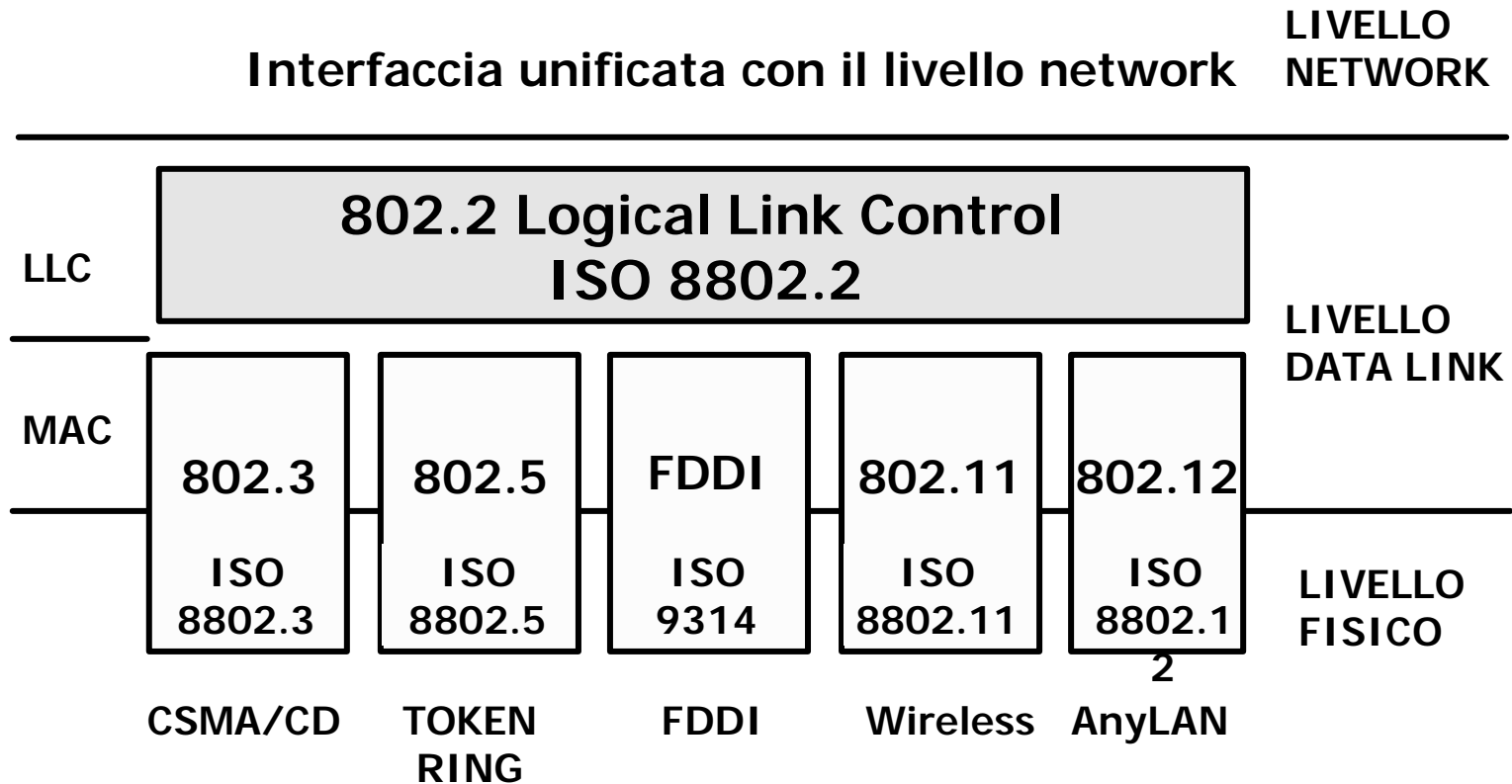
# Limitazioni dell'accesso multiplo

↓ Mezzo trasmissivo ad accesso multiplo

- Problemi di riservatezza delle informazioni
- Riduzione dell'efficienza del mezzo



# Il progetto IEEE 802



**Tecnologie trasmissive differenziate**

# Confronto modello IEEE 802 - OSI

<b>Livelli Superiori (OSI 3-7)</b>			
<b>Data Link</b>		<b>Logical Link Control (LLC)</b>	Recupero errori, controllo di flusso, gestione della connessione logica
		<b>Medium Access Control (MAC)</b>	Controllo di accesso, indirizzamento, 'framing' controllo di errore
<b>Fisico</b>		<b>Fisico</b>	Codifica, sincronizzazione, interfaccia con il mezzo trasmissivo
<b>Modello OSI</b>		<b>Modello IEEE 802</b>	

# Il sottolivello MAC

- ⇓ Il sottolivello MAC è di fondamentale importanza nelle reti di tipo broadcast, in cui ogni sistema riceve tutte le trame inviate dagli altri
- ⇓ Trasmettere in broadcast implica la soluzione di due problemi:
  - in trasmissione: determinare chi deve/può utilizzare il canale
  - in ricezione: discriminare quali messaggi sono destinati alla stazione tramite l'uso di indirizzi
- ⇓ Il MAC normalmente non esiste nelle WAN



**L'indirizzo di livello MAC può essere**

**↗ Individuale es.: 00-50-8b-f3-1b-43**

**↗ Di gruppo (o Multicast) 01-xx-xx-xx-xx-xx**

**↗ Broadcast (ff-ff-ff-ff-ff-ff)**

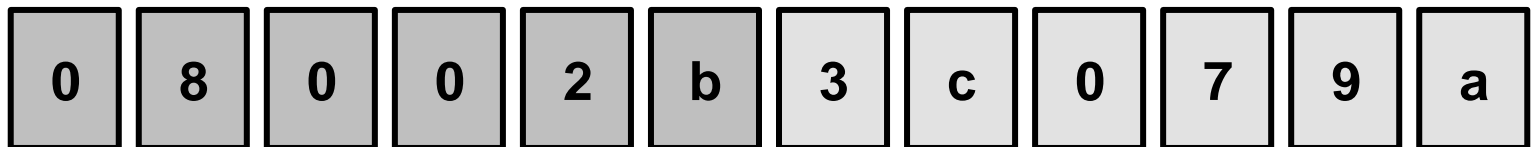
# Indirizzi MAC

↓ Sono univoci a livello mondiale

- sono lunghi 6 byte, cioè 48 bit
- si scrivono come 6 coppie di cifre esadecimali

↓ Si compongono di due parti grandi 3 Byte ciascuna:

- I tre byte più significativi indicano il lotto di indirizzi acquistato dal costruttore della scheda, detto anche OUI (Organization Unique Identifier).
- I tre meno significativi sono una numerazione progressiva decisa dal costruttore



**OUI assegnato dall'IEEE**

**Assegnato dal costruttore**



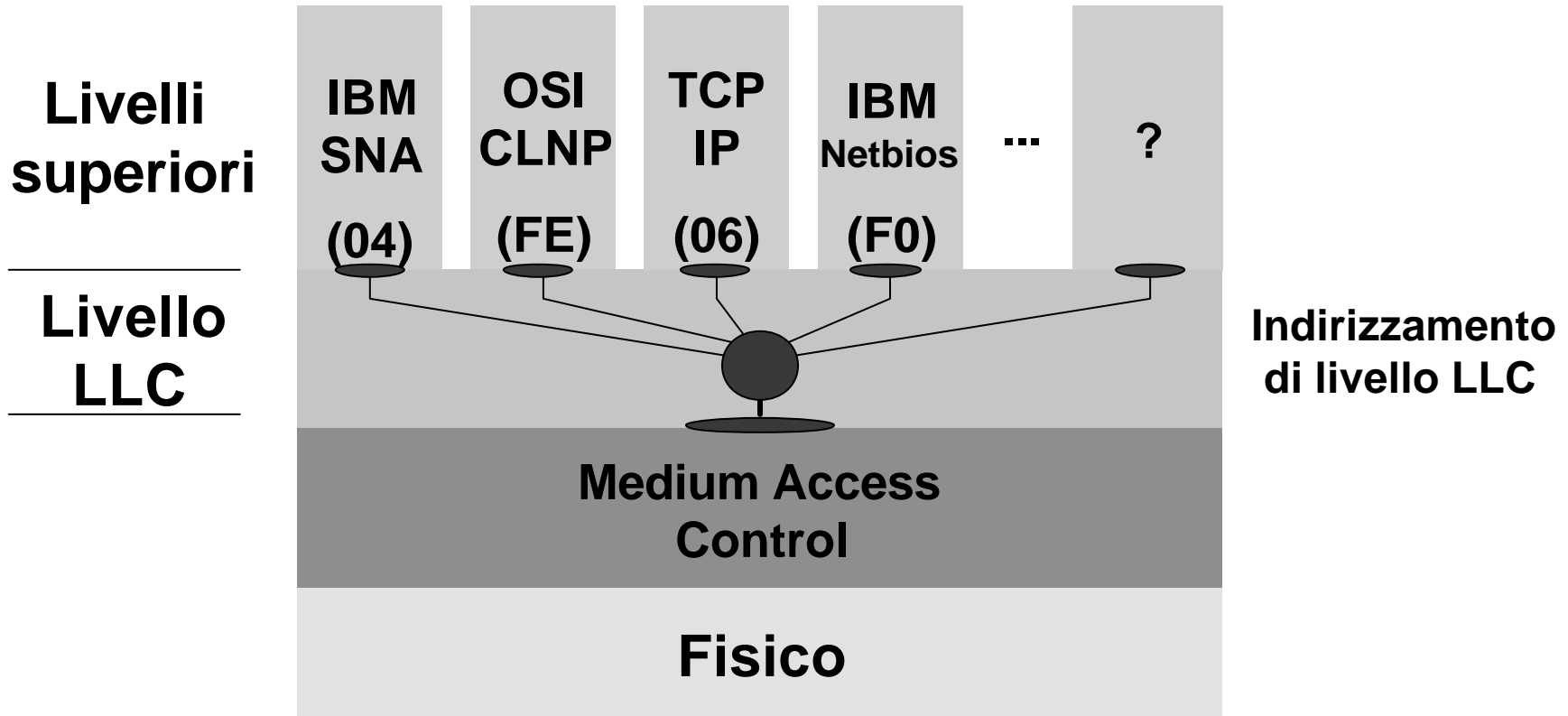
# Alcuni OUI

<b>Organization</b>	<b>Address Block</b>
<b>Cisco</b>	<b>00000Ch</b>
<b>DEC</b>	<b>08002B (e altri)</b>
<b>IBM</b>	<b>08005A (e altri)</b>
<b>Sun</b>	<b>080020h</b>
<b>Proteon</b>	<b>000093h</b>
<b>Bay-Networks</b>	<b>0000A2h</b>

# Indirizzi MAC

- ⇩ Quando una scheda di rete locale riceve un pacchetto, non lo passa automaticamente al livello superiore (LLC), ma effettua una serie di controlli:
- ⇩ verifica che il pacchetto sia integro (FCS corretta) e di dimensioni ammesse
  - analizza l'indirizzo di destinazione (DEST-MAC):
    - ⊛ se il DEST-MAC è broadcast, il pacchetto viene sempre passato al LLC
    - ⊛ se il DEST-MAC è single, il pacchetto viene passato al LLC solo se il DEST-MAC è uguale a quello della scheda
    - ⊛ se il DEST-MAC è multicast, si verifica se la scheda appartiene al gruppo indirizzato

# LLC: supporto multiprotocollo

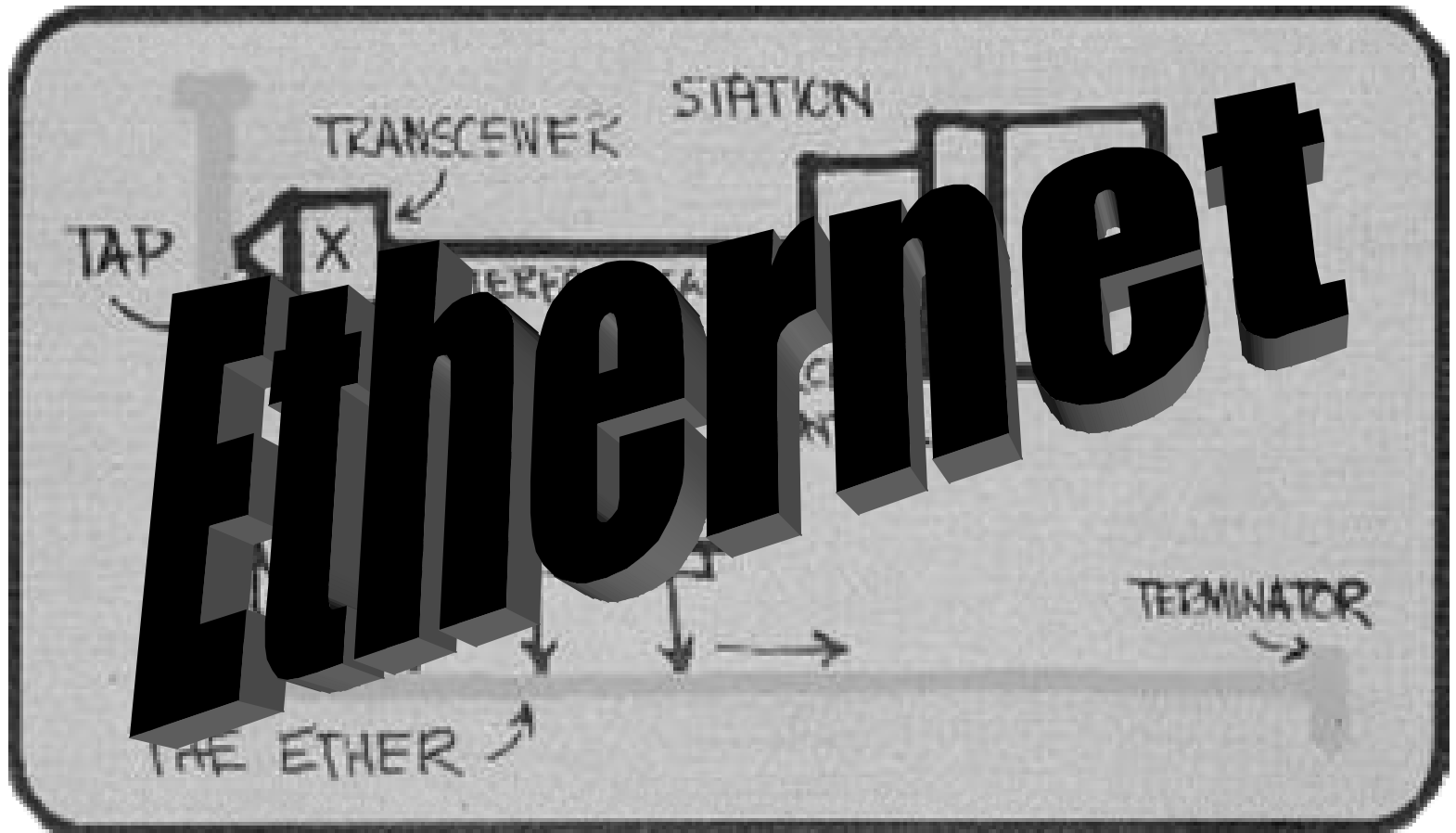


**Entità di livello  
Logical Link Control**



**Service Access Point  
(SAP)**

# Ethernet

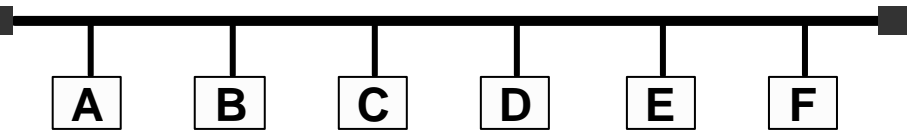
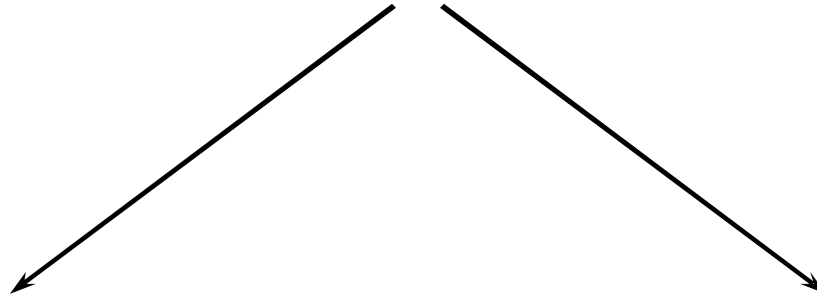
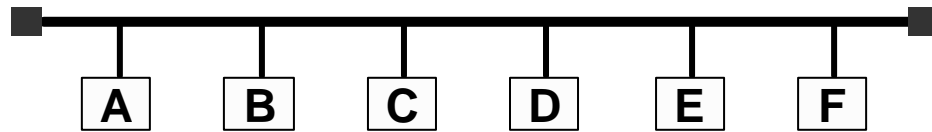


# La rete Ethernet (IEEE 802.3)

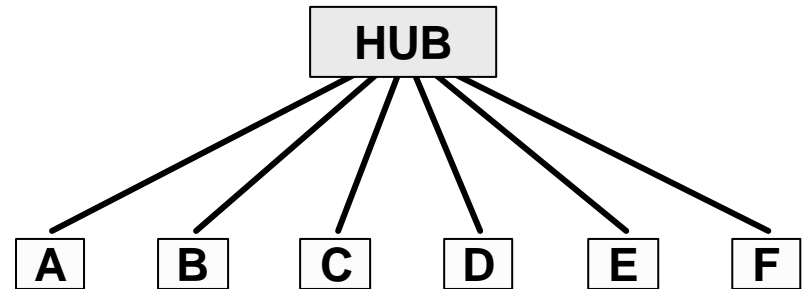
- ↴ Topologia: bus
- ↴ Cablaggio: bus, stella
- ↴ Arbitraggio del canale trasmissivo: tramite contesa
- ↴ Velocità Trasmissiva: 10 Mb/s
- ↴ IEEE 802.3u: versione a 100 Mb/s
- ↴ IEEE 802.3z: versione a 1 Gb/s

# Topologia e cablaggio

## Topologia a bus



## Cablaggio a bus



## Cablaggio a stella

# Metodo di accesso CSMA/CD

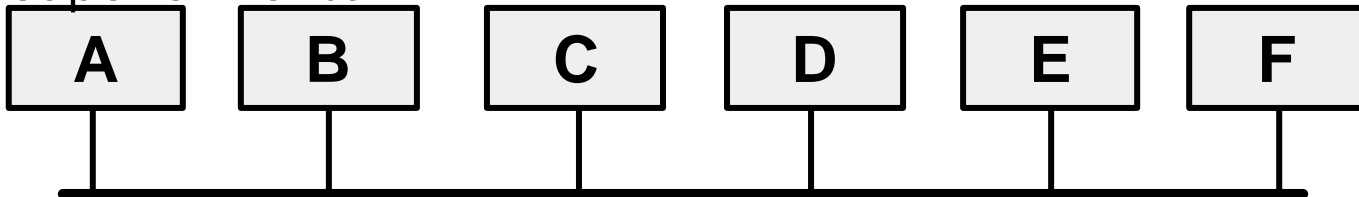
↓ CSMA/CD:

- Carrier Sense
- Multiple Access
- with Collision Detection

↓ Protocollo MAC:

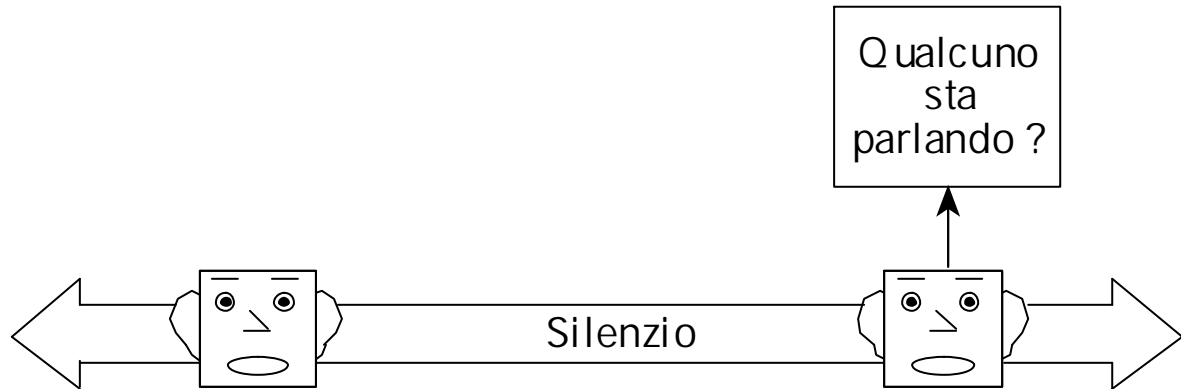
- concepito per topologie a bus
- non deterministico con tempo di attesa non limitato

superiormente

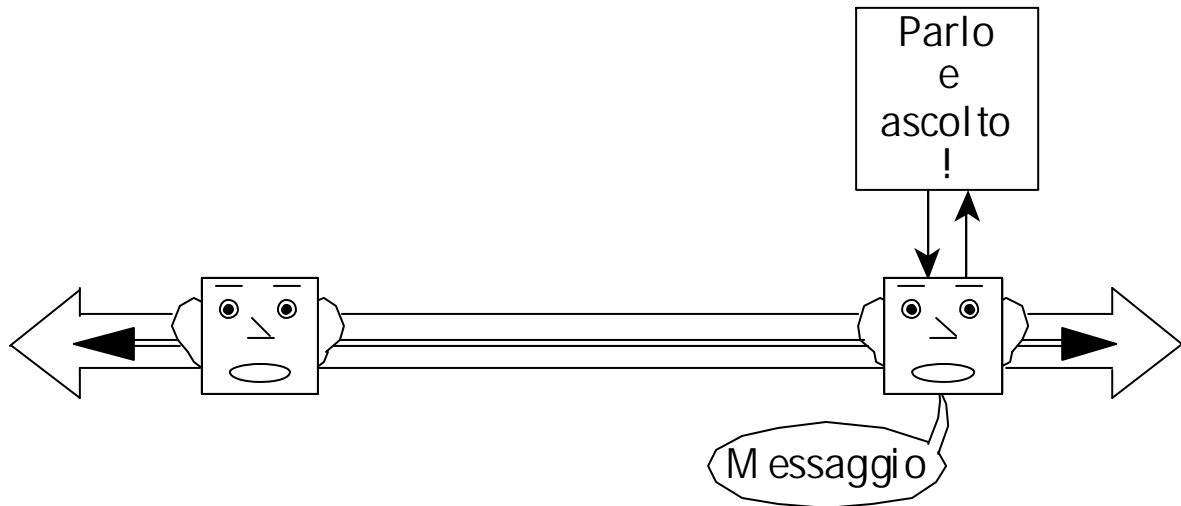


# Trasmissione senza collisione

**Fase1:  
Ascolto**

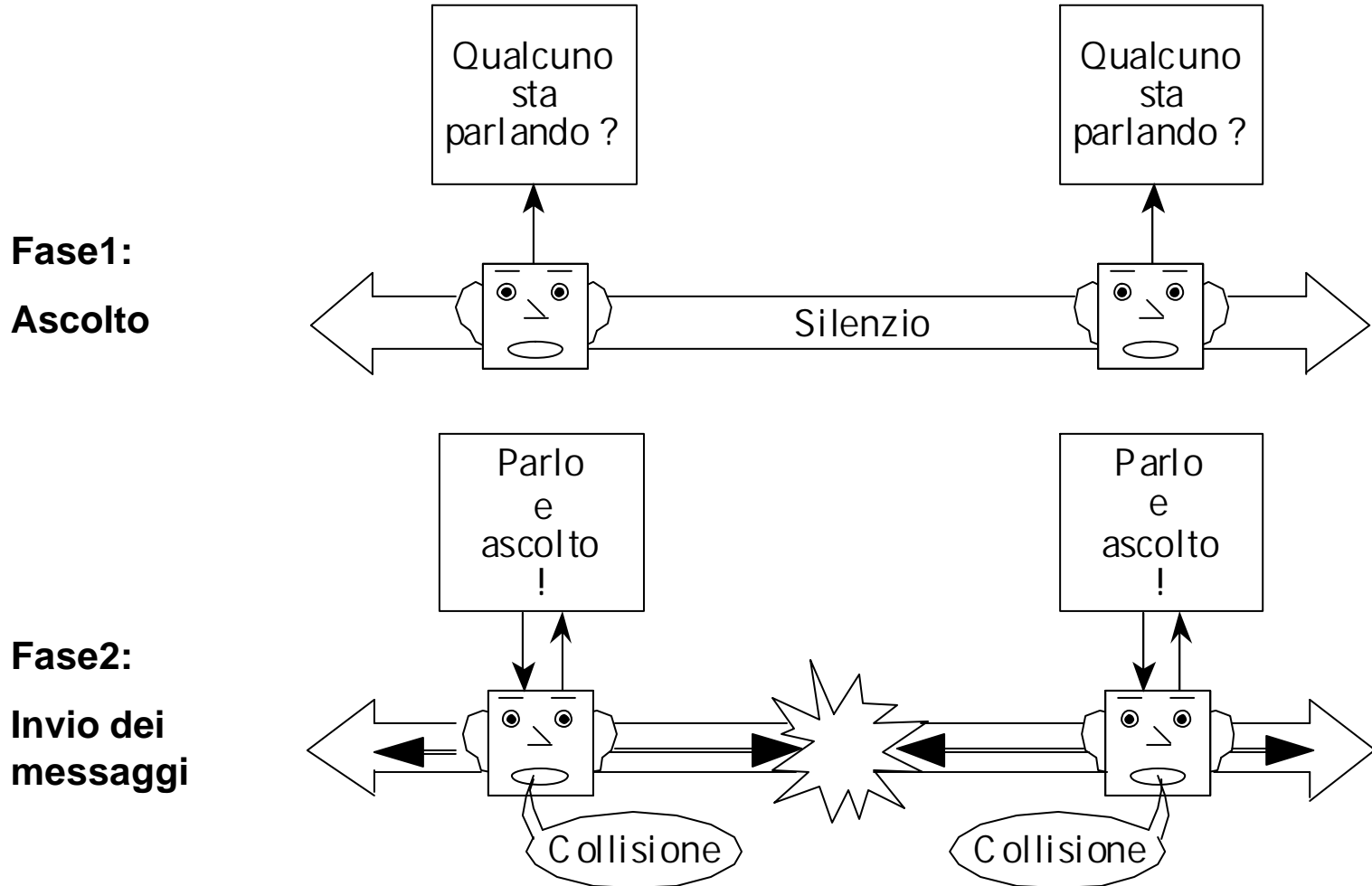


**Fase2:  
Invio dei  
messaggi**





# Trasmissione con collisione



# Dominio di collisione

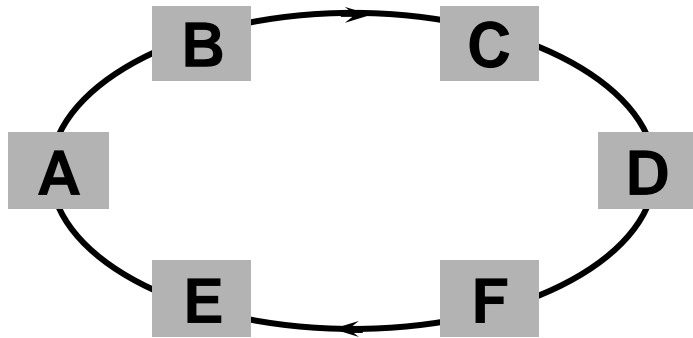
- ⇓ In una rete CSMA/CD al crescere del numero di stazioni e/o del traffico aumenta la probabilità di collisioni e quindi diminuisce l'efficienza della rete
- ⇓ E' possibile suddividere la rete in più sottoreti in modo che la contesa del mezzo avvenga soltanto tra le stazioni appartenenti ad una singola sottorete, la quale rappresenta un singolo dominio di collisione
- ⇓ Le stazioni separate da repeater fanno parte dello stesso dominio di collisione
- ⇓ Fanno parte di domini di collisione diversi le stazioni separate da apparecchiature di rete che lavorano a livelli OSI superiori al fisico (bridge, switch, router) e che sono quindi in grado di decodificare gli indirizzi MAC e filtrare i pacchetti

# La rete Token Ring (IEEE 802.5)

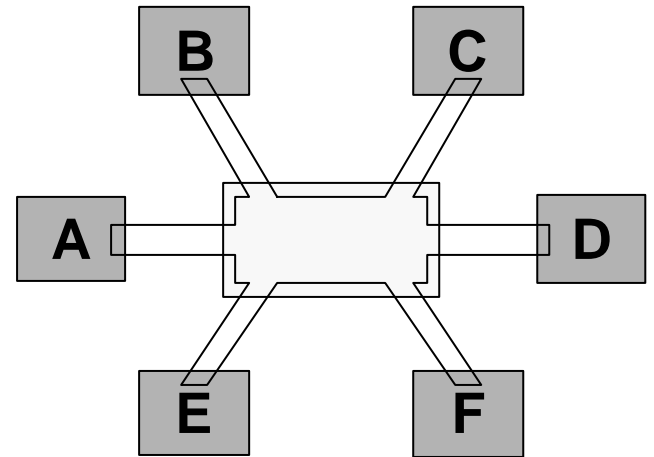
- ↴ Topologia: anello
- ↴ Cablaggio: stella
- ↴ Arbitraggio del canale trasmissivo: token
- ↴ Tipologia del protocollo: deterministico con tempo di attesa limitato superiormente
- ↴ Velocità Trasmissiva: 4 o 16 Mb/s
- ↴ Throughput massimo: 3 o 12 Mb/s
- ↴ Rete proposta da IBM in alternativa a Ethernet

# Topologia e cablaggio

- ↓ Una rete Token Ring consiste in un certo numero di stazioni collegate serialmente tramite un mezzo trasmissivo e rinchiusa ad anello
- ↓ Poiché le stazioni devono ripetere continuamente i pacchetti delle altre stazioni, per ragioni di affidabilità la rete viene cablata a stella

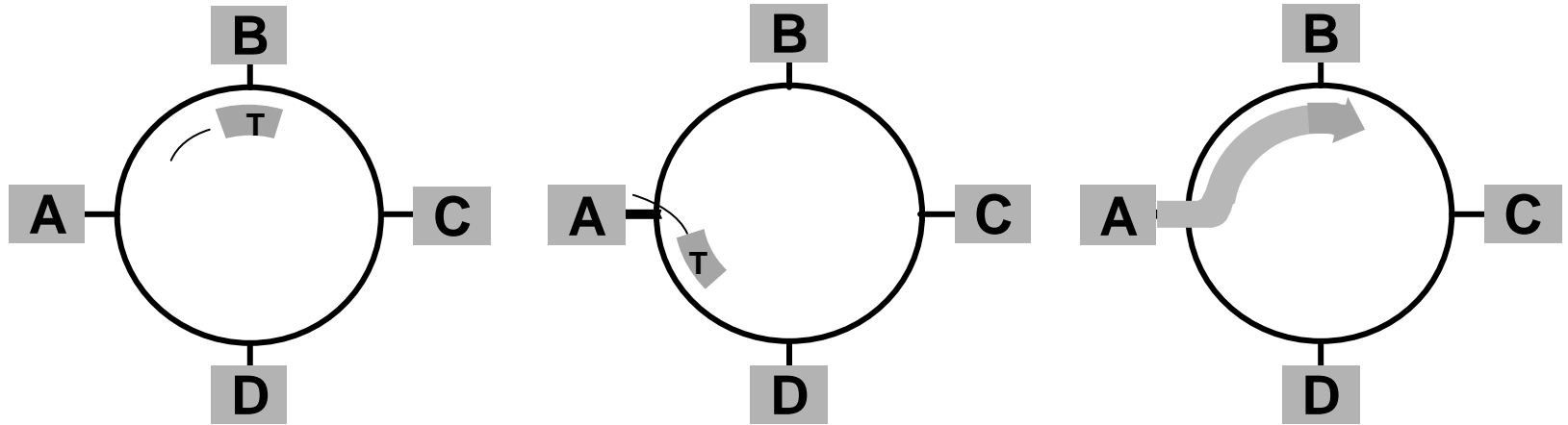


**Topologia ad anello**



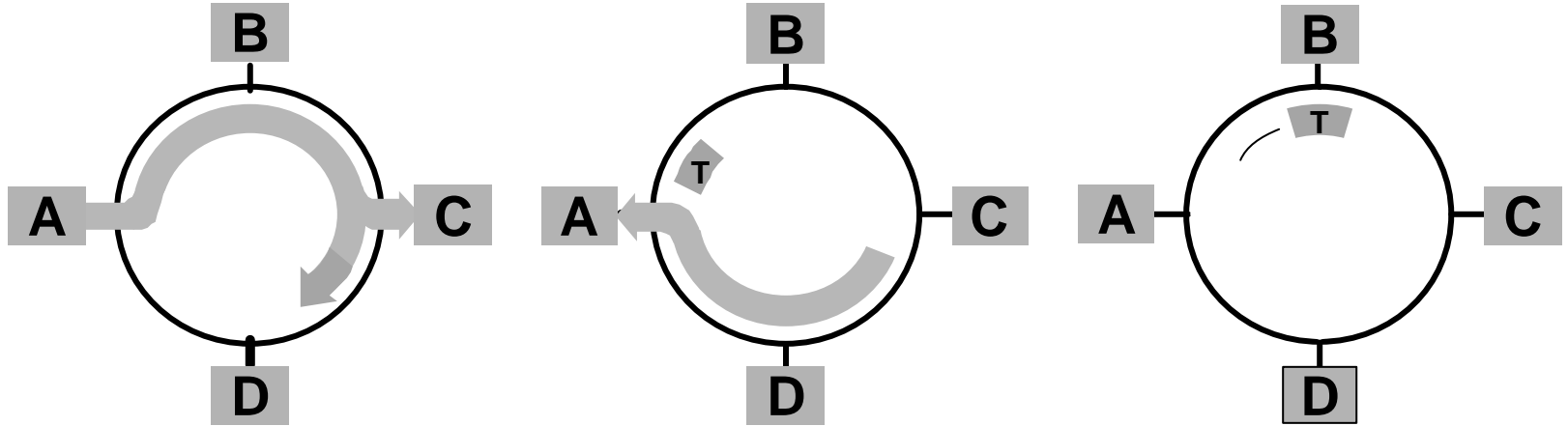
**Cablaggio a stella**

# Metodo di accesso a token (1)



- ⇓ Il nodo mittente (A) attende la ricezione del Token
- ⇓ Modifica il Token "al volo" e gli accorda la trama informativa

## Metodo di accesso a token (2)



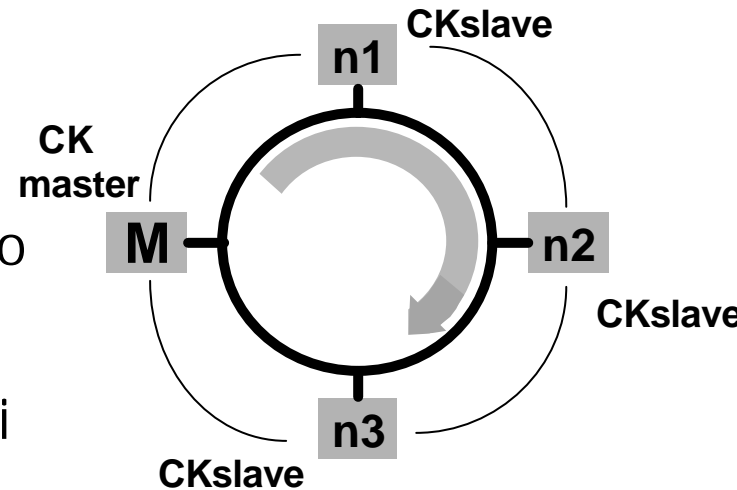
- ↓ Il nodo destinatario (C) riconosce l'indirizzo e legge la trama
- ↓ Il nodo mittente riceve la trama che ha compiuto il giro dell'anello ed emette un nuovo Token che può essere catturato dai nodi a valle

# Nodo di monitor

↓ E' necessaria la presenza di un nodo 'Monitor' che svolge le seguenti funzioni:

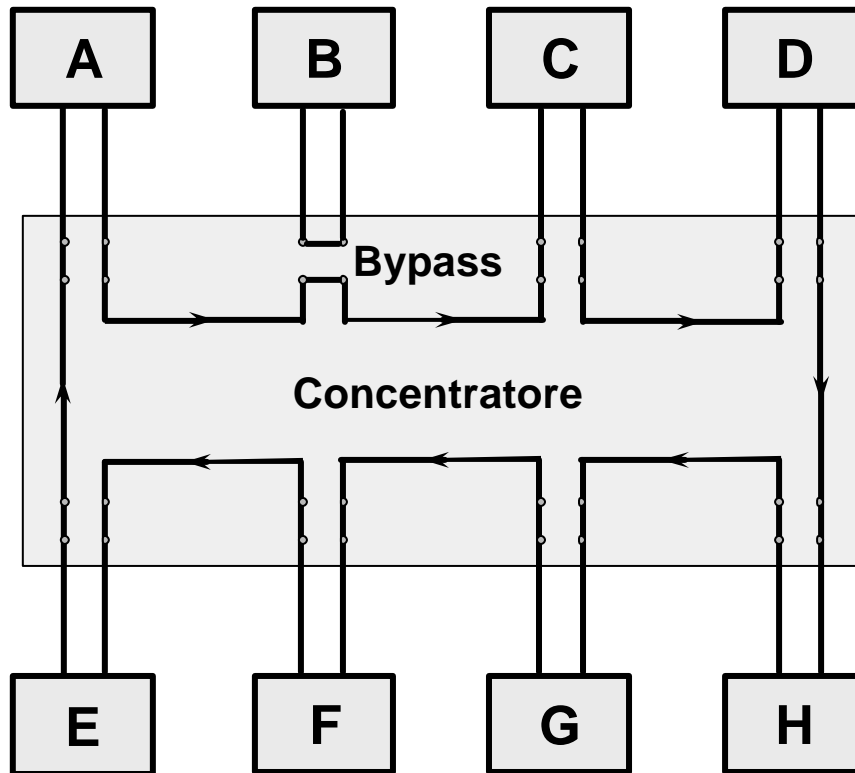
- emette il primo token
- controlla la presenza del token e lo rigenera se necessario
- elimina eventuali trame persistenti sull'anello

↓ Tutte le stazioni possono assumere il ruolo di monitor



# Concentratore (MAU)

↓ Quando un stazione è spenta o guasta, il concentratore la esclude dalla rete (bypass)



**Multistation  
Access  
Unit**



# Interconnessione di LAN

# Interconnessione di LAN

- ⇓ Gli apparati utilizzati per interconnettere le LAN sono:
  - Repeater
  - Switch
  - Router
- ⇓ Switch e Router sono utilizzati per interconnettere tra loro reti con differenti:
  - tecnologie (ad es., reti Ethernet e Token Ring)
  - tipologie (ad es., reti locali e geografiche)
  - e per aumentarne la dimensione

# Generalità

## ↓ I Bridge

- operano a livello 2 OSI (sottolivello MAC)
- hanno algoritmi di instradamento molto semplici
- si utilizzano normalmente per interconnessioni locali

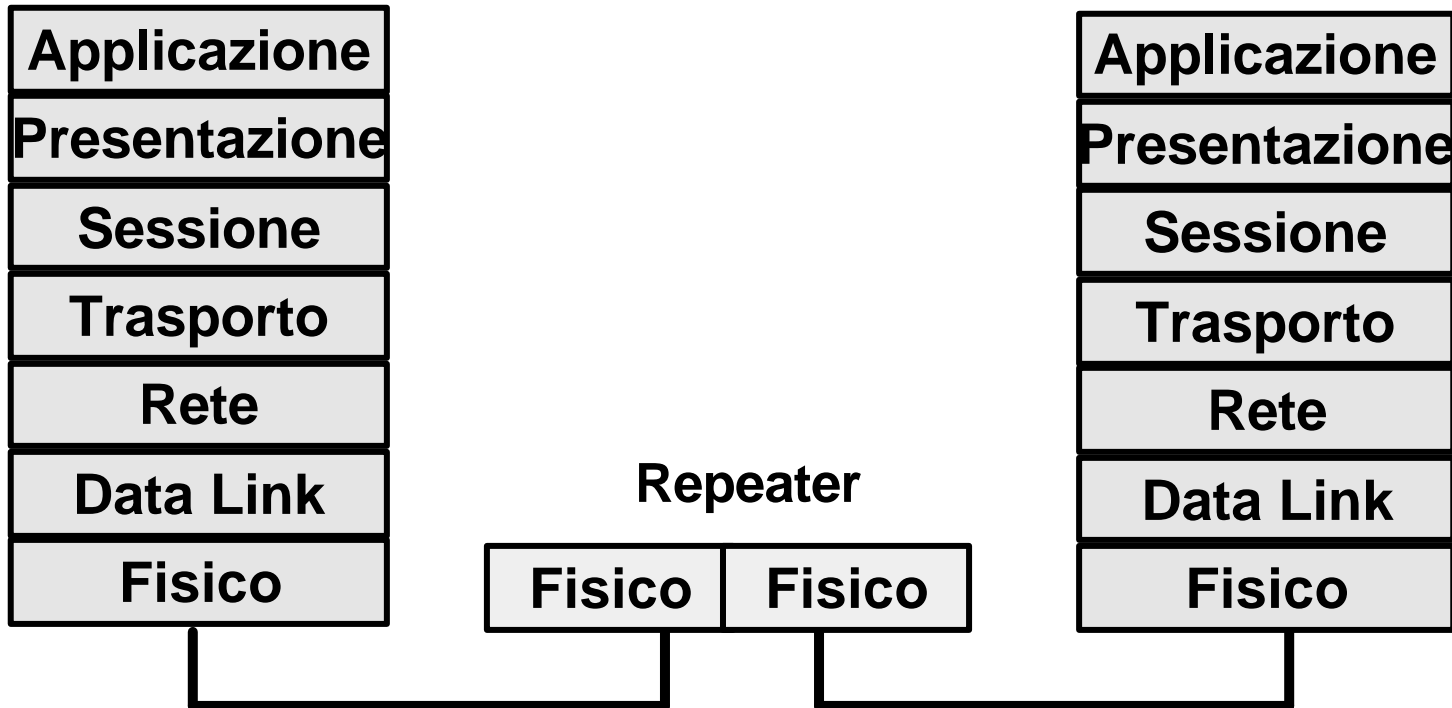
## ↓ Gli Switch

- operano a livello 2 (sottolivello MAC)
- usano diverse tecniche di inoltro del pacchetto
- si utilizzano per realizzare reti locali commutate

## ↓ I Router

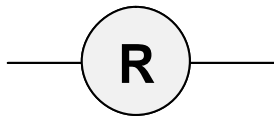
- operano a livello 3
- hanno algoritmi di instradamento sofisticati
- si utilizzano normalmente per interconnessioni geografiche

# Interconnessione con Repeater

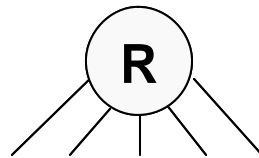


# Repeater

- ↓ Serve per ripetere e rigenerare una sequenza di bit ricevuti da una porta sulle altre porte.
- ↓ Assume il nome di:
  - repeater quando è costituito da 2 porte;
  - multiport repeater quando è costituito da più di 2 porte.



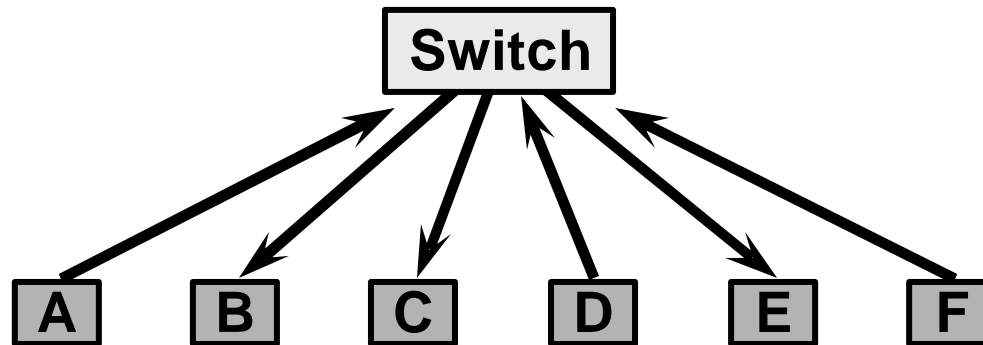
**Repeater**



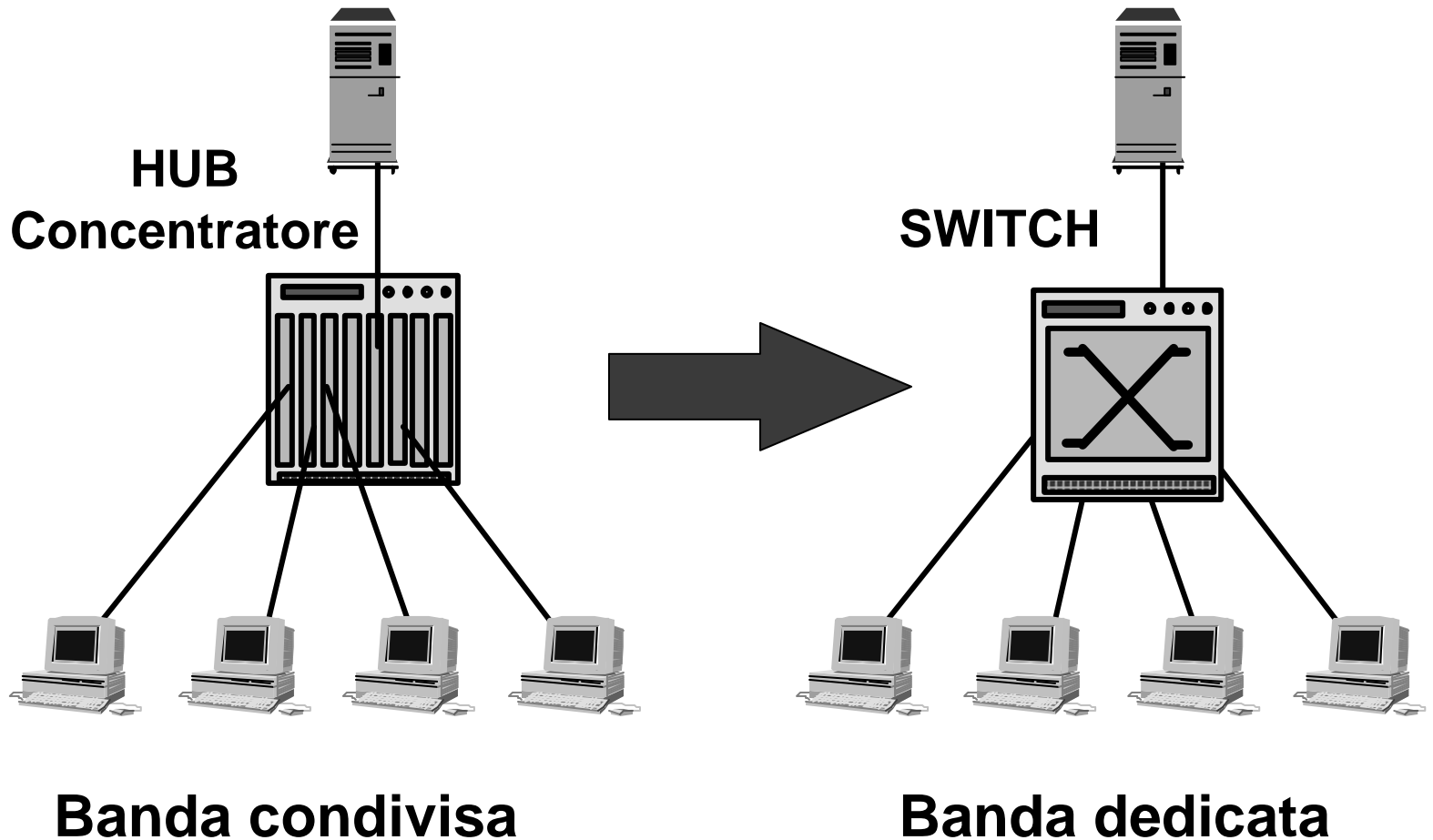
**Multiport-Repeater**

# Switch

- ⇩ Sono dei bridge multi-porta
  - realizzazione in hardware dell'algoritmo
  - molto veloci
- ⇩ Si sostituiscono ai repeater nei centro stella
- ⇩ Hanno una banda aggregata molto superiore a quella della singola porta
  - Molte trasmissioni in contemporanea



# Dal concentratore allo switch



# Modalità di switching

## ⇩ Store-and-Forward

- utilizzata dai bridge (prevista da IEEE 802.1d)
- la trama viene ricevuta interamente e poi ritrasmessa

## ⇩ Cut through

- non appena lo switch comincia a ricevere una trama, ne legge l'indirizzo di destinazione, e inizia immediatamente a trasmettere la trama senza aspettare che questa sia arrivata per intero

## ⇩ Fragment free

- alla pari della modalità cut-through, anche qui non si aspetta l'intera trama prima di iniziare a trasmetterla, però ci si assicura che questa sia almeno lunga 64 byte e si scarta qualsiasi frammento di trama che abbia dimensioni inferiori

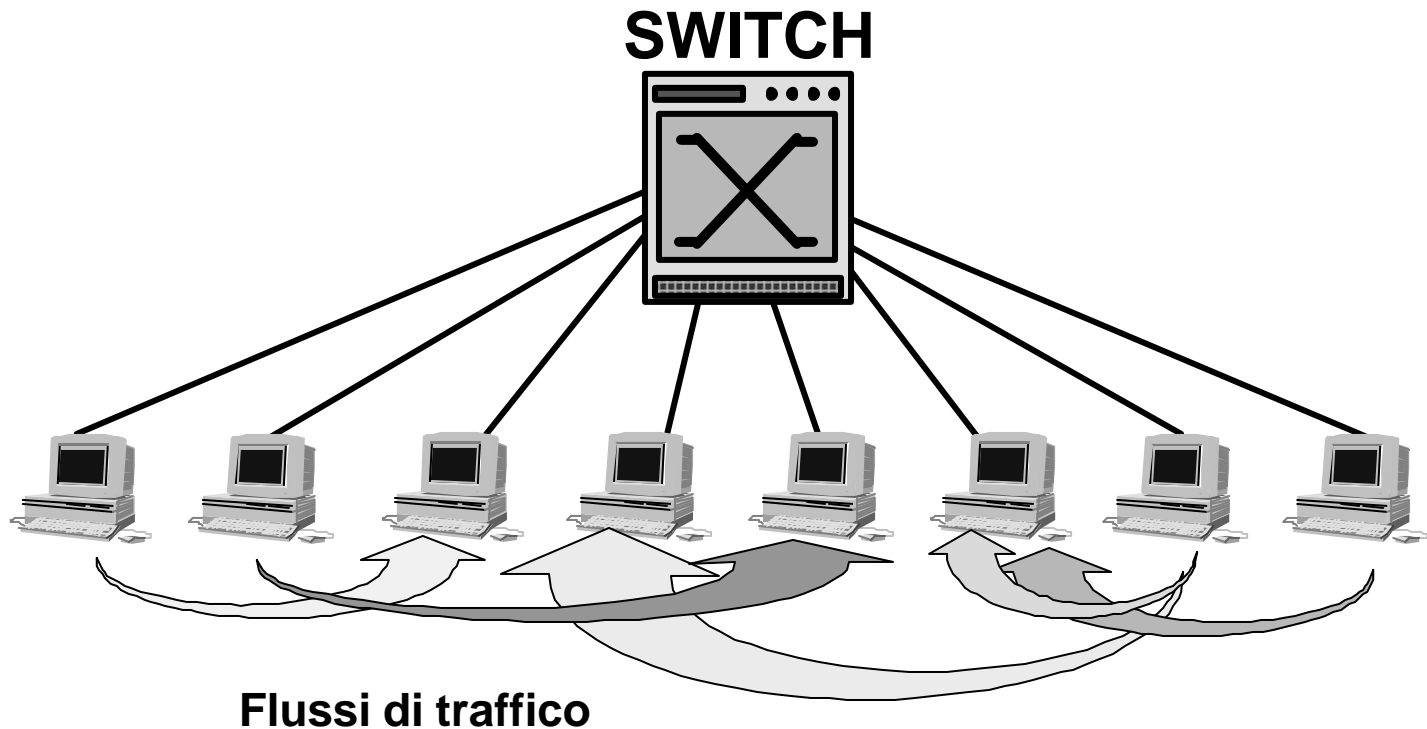


# Modalità di switching

- ↓ Uno switch quando opera in modalità cut-through non può verificare e ricalcolare la FCS prima di aver iniziato la ritrasmissione della trama e quindi non può evitare di inoltrare sulla rete una trama corrotta
- ↓ Esistono alcune condizioni che inibiscono il cut-through e impongono allo switch di operare in modalità store and forward
  - quando uno switch opera tra due reti locali appartenenti a due standard diversi
  - quando uno switch opera tra due reti identiche, ma a velocità diverse
  - quando la porta di destinazione è occupata
  - quando la trama ha un indirizzo di dest. multicast o broadcast

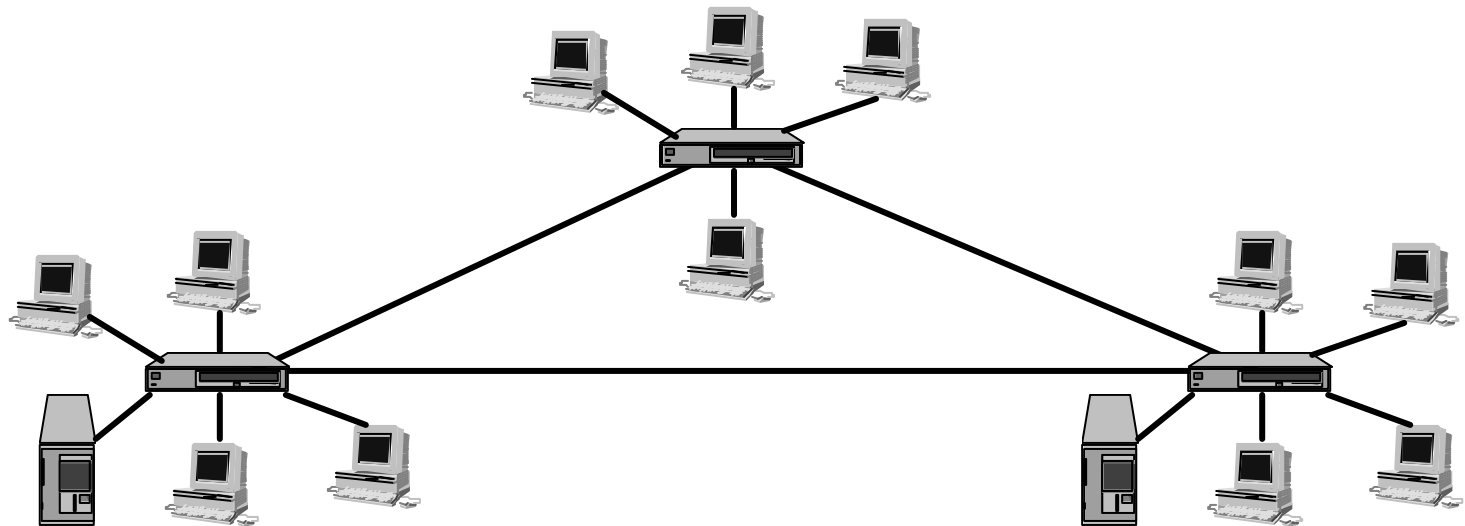
# Ethernet switching

- ⇩ Mezzo dedicato
- ⇩ Più comunicazioni simultanee
- ⇩ Throughput massimo: 10 Mb/s dedicati



# Switched LAN

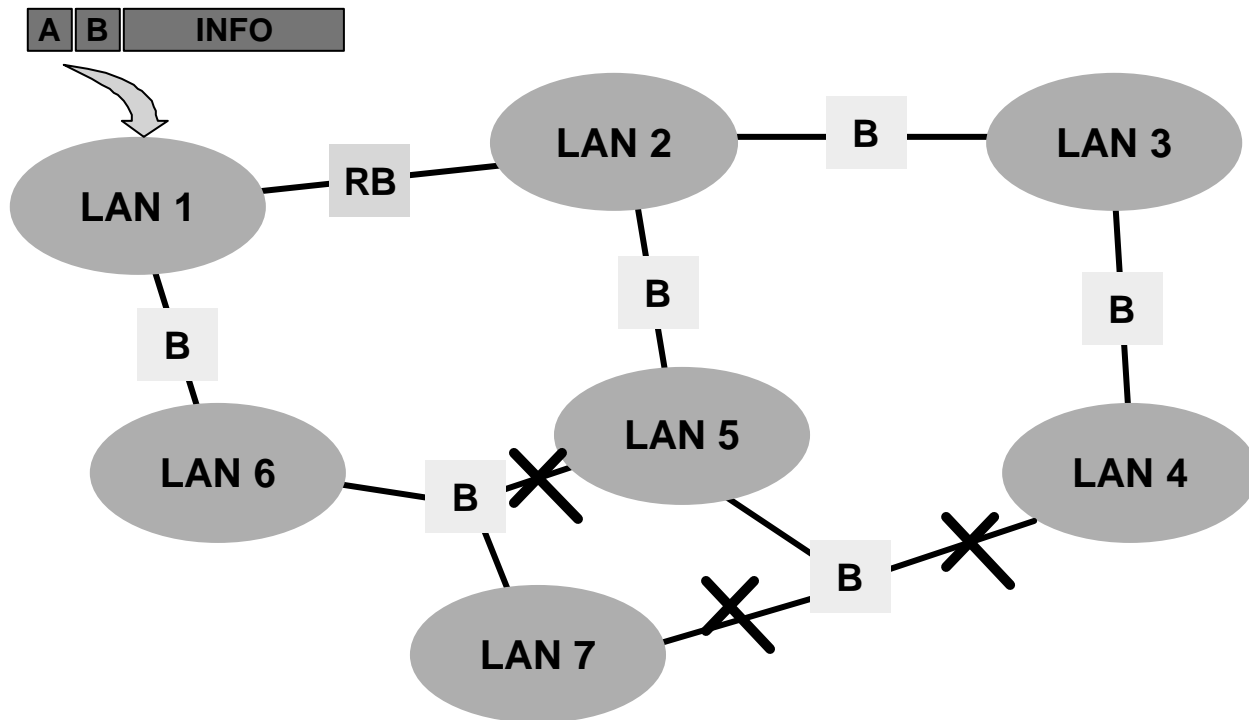
- ⇩ Necessità di interconnettere più Switch
- ⇩ 10 Mb/s costituiscono un collo di bottiglia
- ⇩ Necessità di connessioni ad elevata velocità



# Spanning Tree

↓ La rete magliata deve essere trasformata in albero

- protocollo di spanning tree (IEEE 802.1 D)



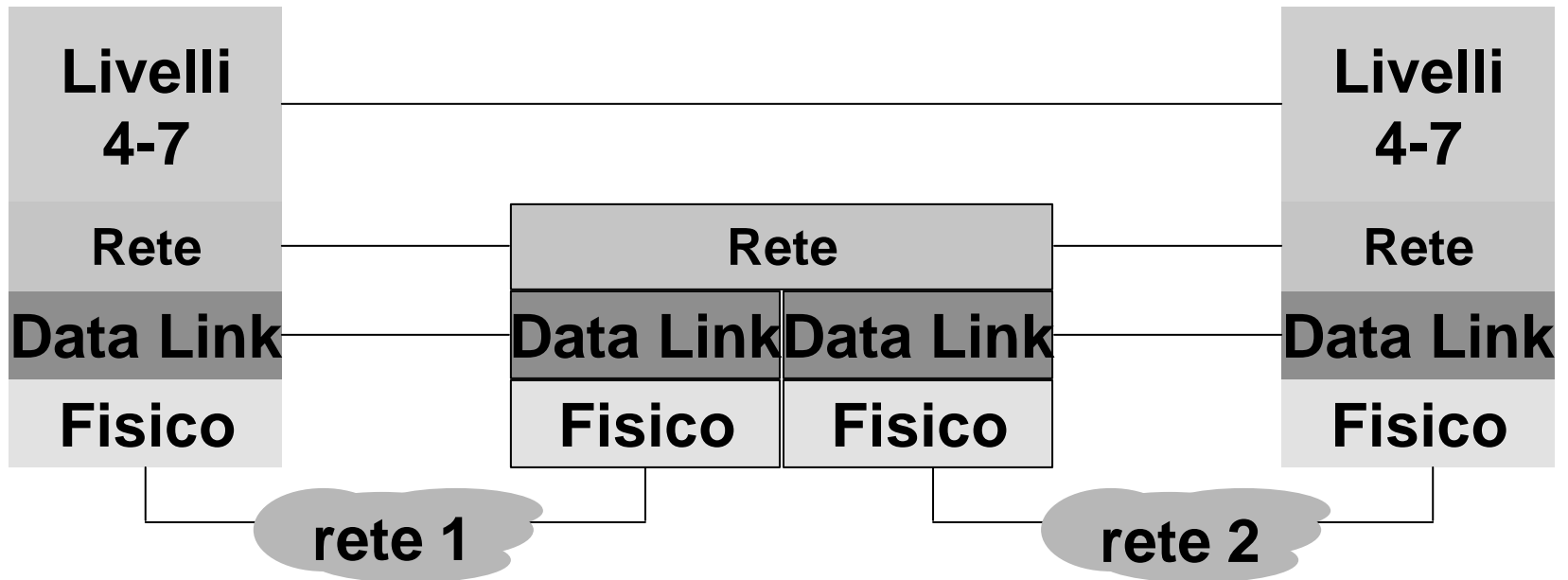
# Il livello di rete

- ⇓ Il livello *network* si occupa di trasmettere pacchetti dalla sorgente alla destinazione.
  - ⇓ Per raggiungere la destinazione può essere necessario attraversare lungo il percorso diversi nodi intermedi.
  - ⇓ Compito diverso da quello del livello *data link* che, come sappiamo,...
- ...ha il più modesto compito di trasportare pacchetti da un estremo all'altro di un cavo.

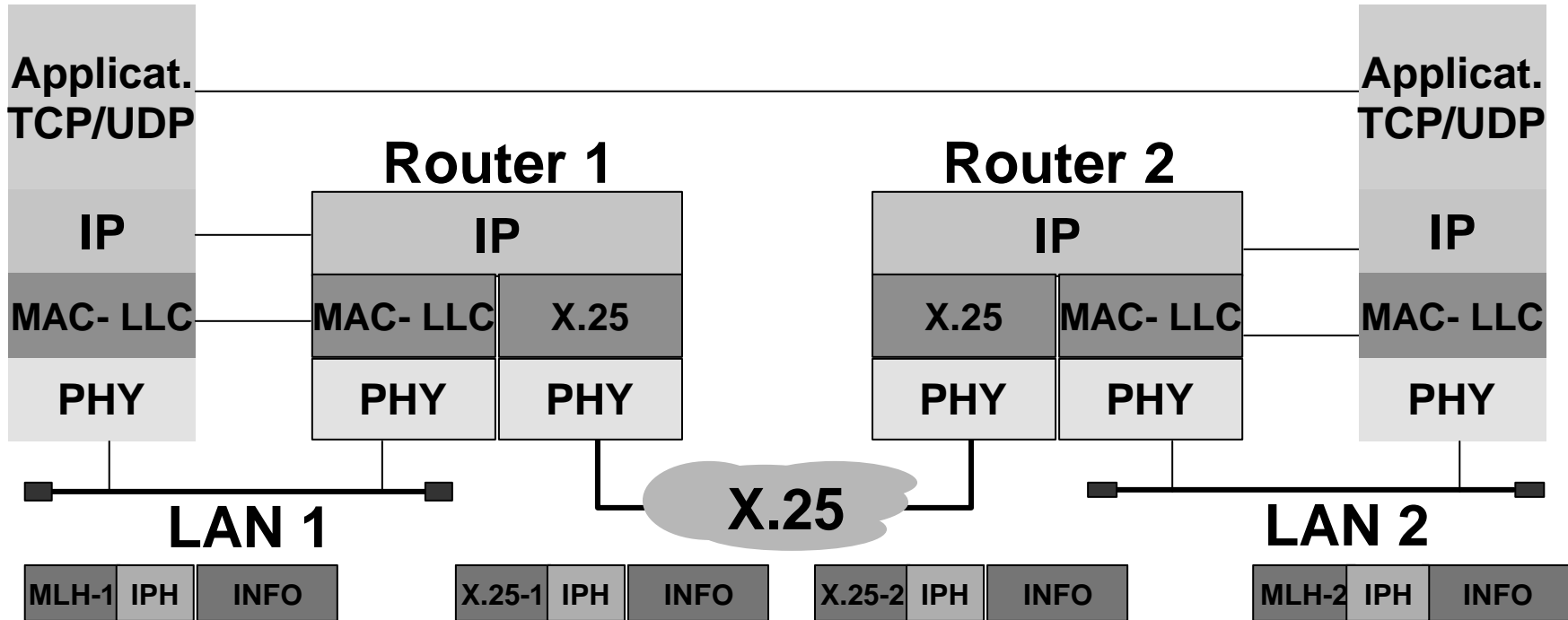
# Interconnessione di LAN tramite router

- ⇩ Operano a livello 3 della pila OSI
- ⇩ Sono indirizzati in modo esplicito
- ⇩ Non sono trasparenti ai nodi terminali (ES)
- ⇩ Analizzano gli indirizzi di livello 3 (es. IP)
- ⇩ Consentono instradamenti ottimizzati
- ⇩ Permettono la gestione di reti complesse
- ⇩ Possono operare con diversi protocolli di rete:
  - ⊗ IP
  - ⊗ SNA
  - ⊗ DECNET
  - ⊗ IPX
  - ⊗ OSI CLNP
  - ⊗ ....

# Router



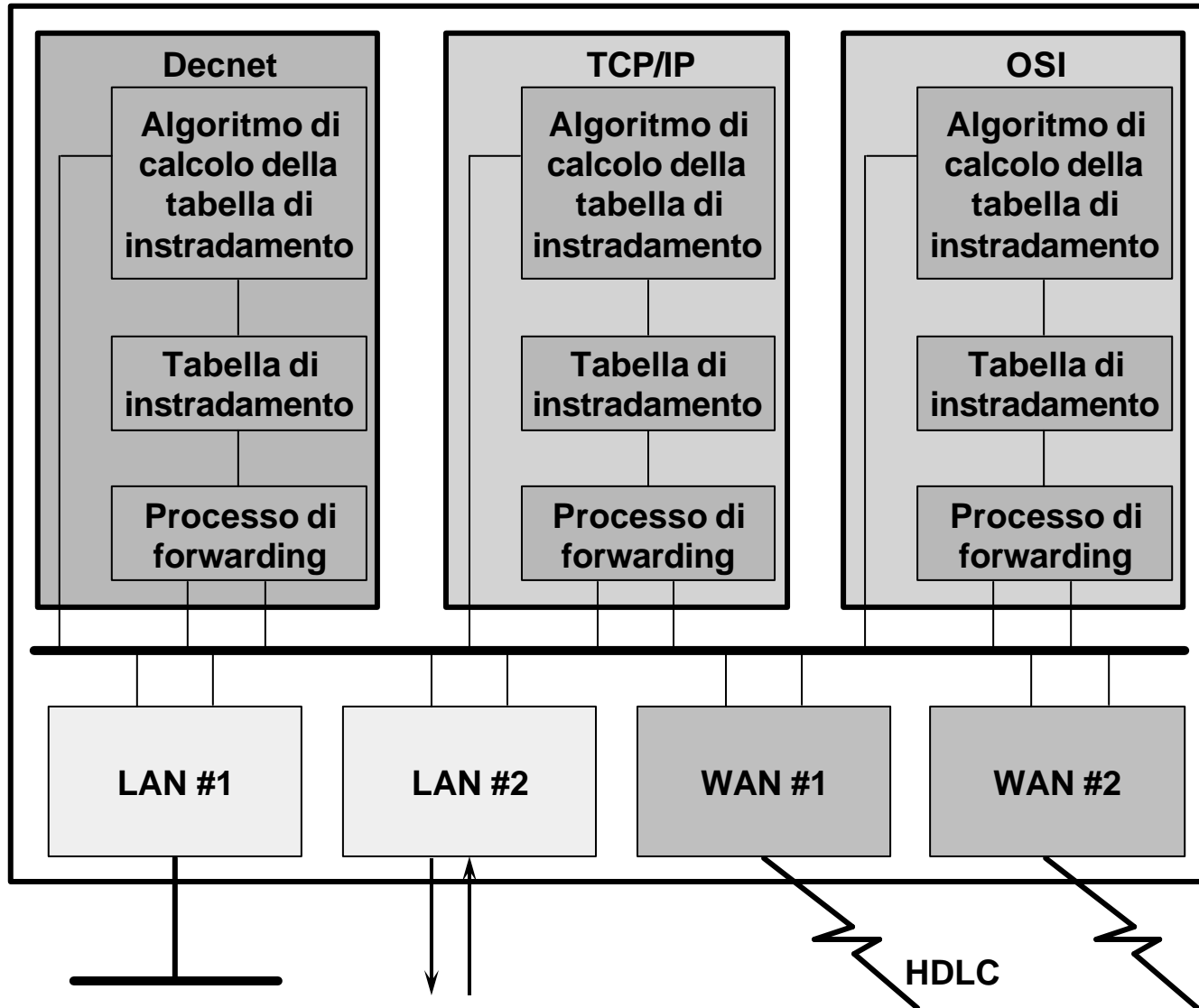
# Interconnessione geografica di LAN



- ⇩ Indirizzamento globale di livello IP (a.b.c.d)
- ⇩ Lo strato IP analizza l'etichetta IP ed instrada il messaggio in base alle proprie tabelle



# Router multiprotocollo



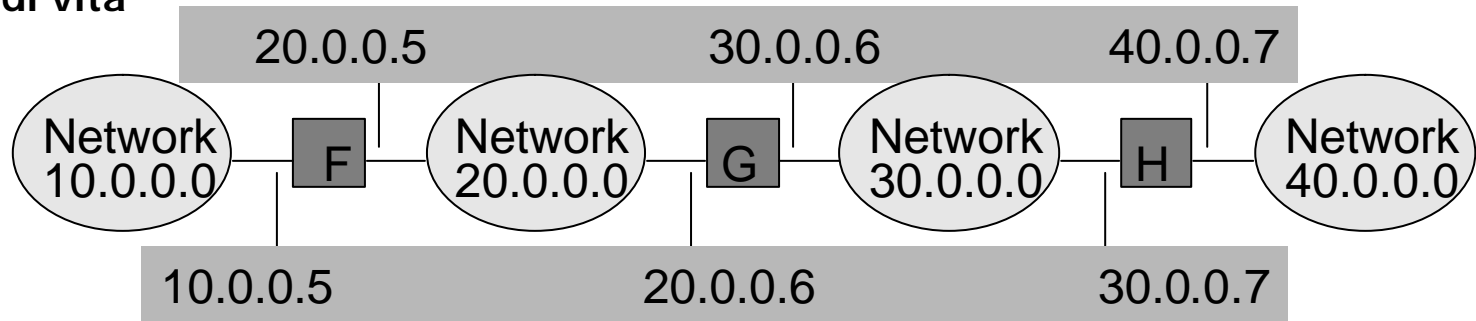
**Livello 3  
(Network)**

**Livelli  
1 e 2**

# Routing IP: tabelle di routing

↓ Per effettuare forwarding indiretto il mittente del messaggio deve disporre di una tabella (routing table) che per ogni riga riporta:

- Network di destinazione
- Router direttamente raggiungibile da attraversare
- Metrica (es. N. di salti)
- Tempo di vita

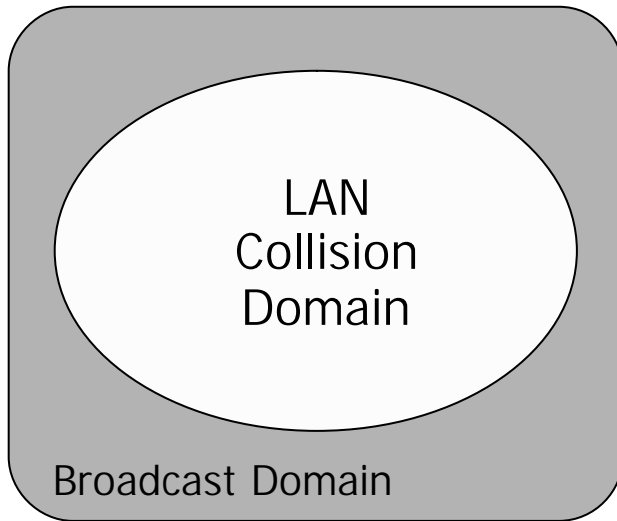


To reach hosts on network	Route to this address	Metric to destination	Time to live
30.0.0.0	Direct	0	-
20.0.0.0	Direct	0	-
10.0.0.0	20.0.0.5	1	60
40.0.0.0	30.0.0.7	1	85

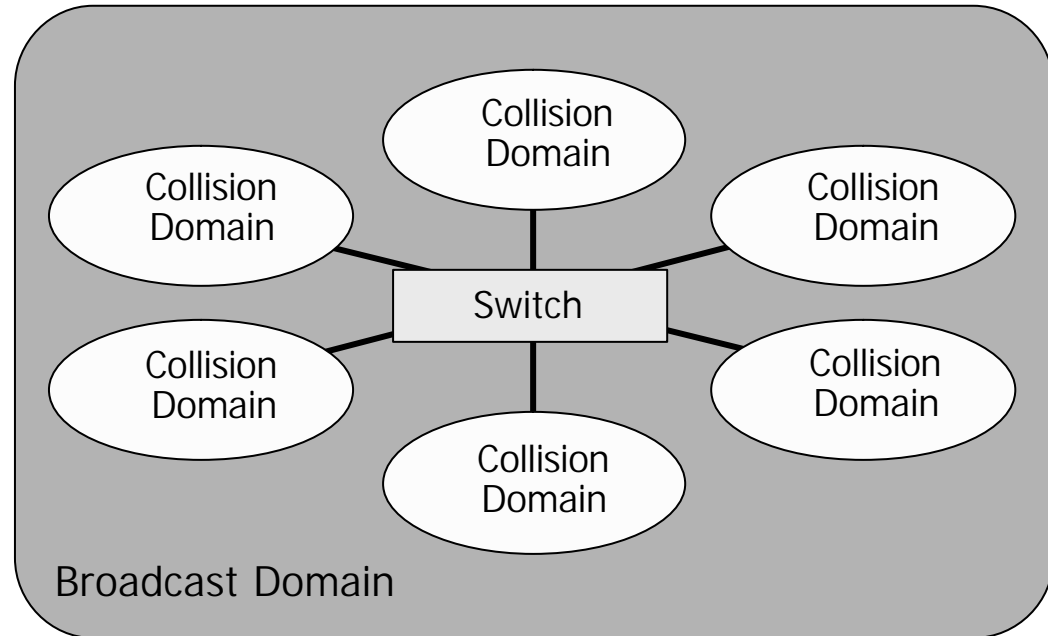
routing table di G

# Segmentazione di LAN tramite switch

Before Switching

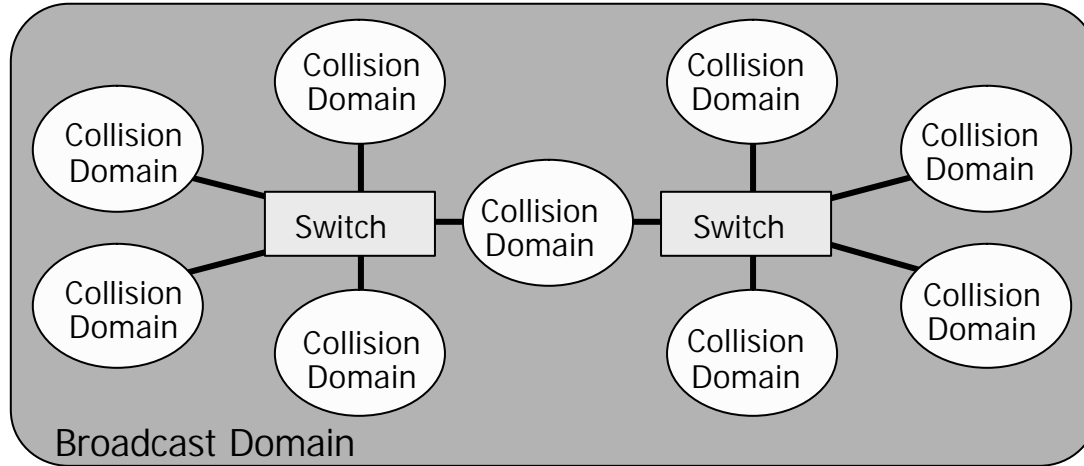


After Switching

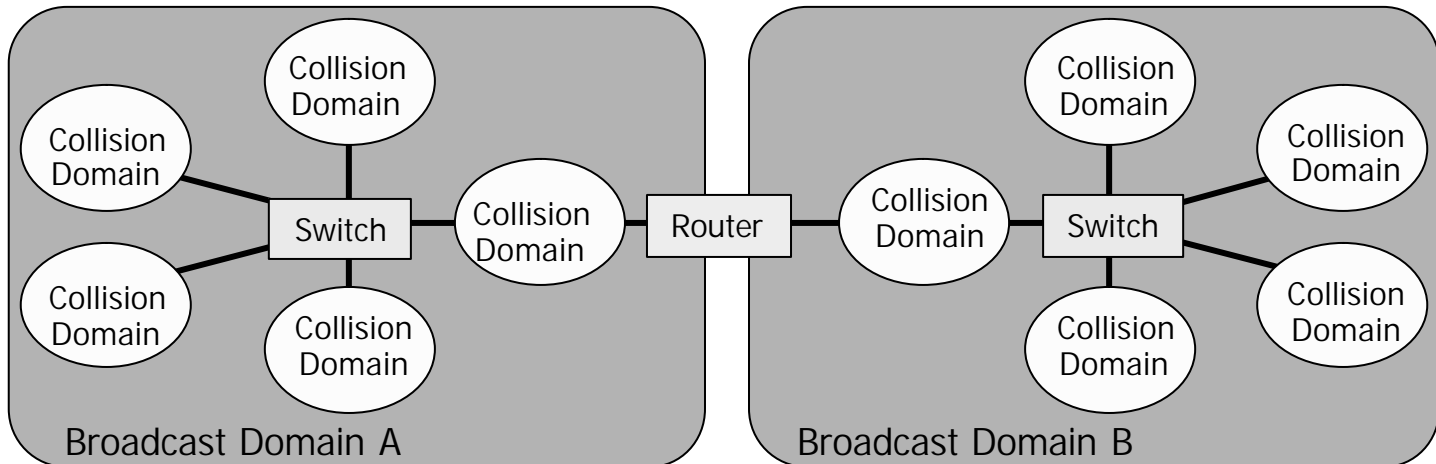


# Segmentazione di LAN tramite router

## Switched Broadcast Domain



## I Routers segmentano i Broadcast Domains



# Router vs. Bridge

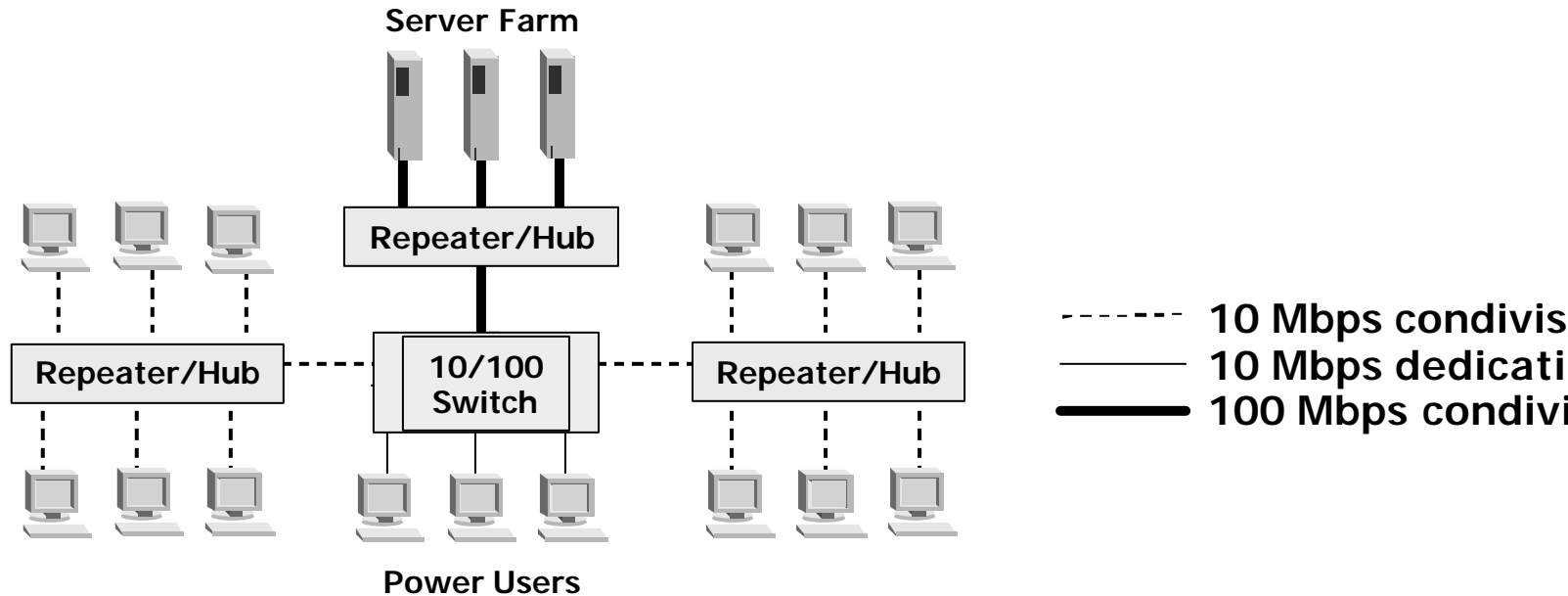
## ↓ **Vantaggi dei router:**

- possono determinare il percorso migliore esistente tra una sorgente ed una destinazione; i bridge sono limitati ad un percorso specifico (denominato spanning tree)
- ricalcolano le proprie tabelle di instradamento, a seguito di modifiche della topologia della rete, molto più velocemente dei bridge
- forniscono una barriera contro i broadcast storms
- frammentano i pacchetti di dimensioni elevate; i bridge scartano i pacchetti troppo grandi per essere inoltrati

## ↓ **Vantaggi dei bridge:**

- richiedono una configurazione minima
- impiegabili con qualsiasi protocollo di livello 3
- instradano i protocolli sprovvisti del livello 3 (ad es., LAT)

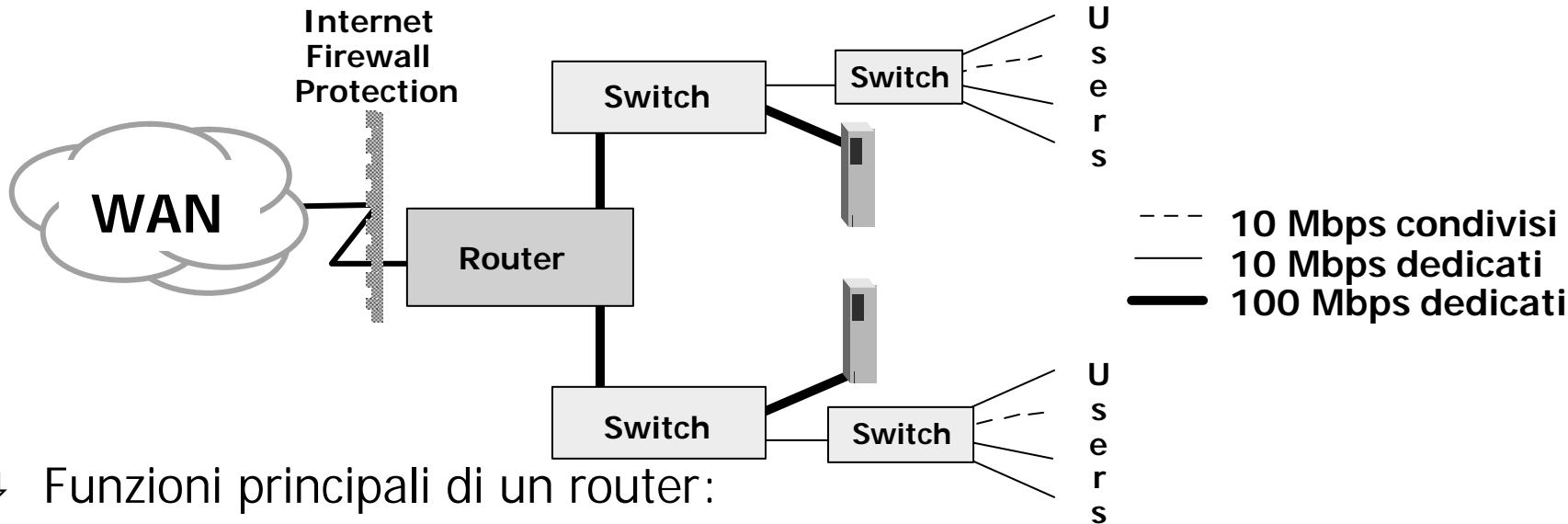
# Quando usare gli switch



↓ Gli Switch risolvono problemi di performance:

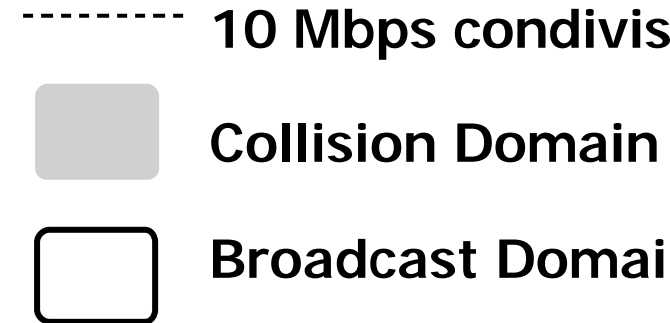
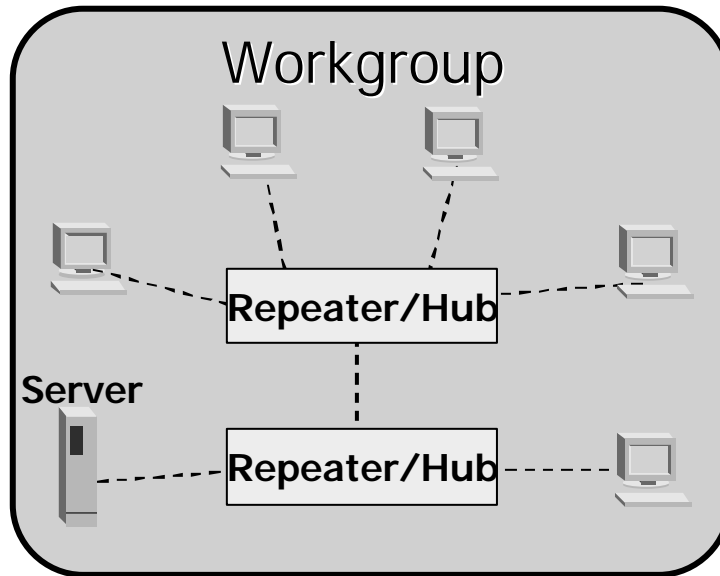
- Domini di collisione più piccoli
- Elevata banda aggregata
- Elevato throughput
- Bassa latenza
- Basso costo per porta

# Quando usare i router



- ↓ Funzioni principali di un router:
- segmenta la rete in differenti domini di broadcast
  - inoltra 'intelligente' dei pacchetti
  - accesso alle WAN
  - supporta percorsi ridondanti
  - sicurezza (firewall)

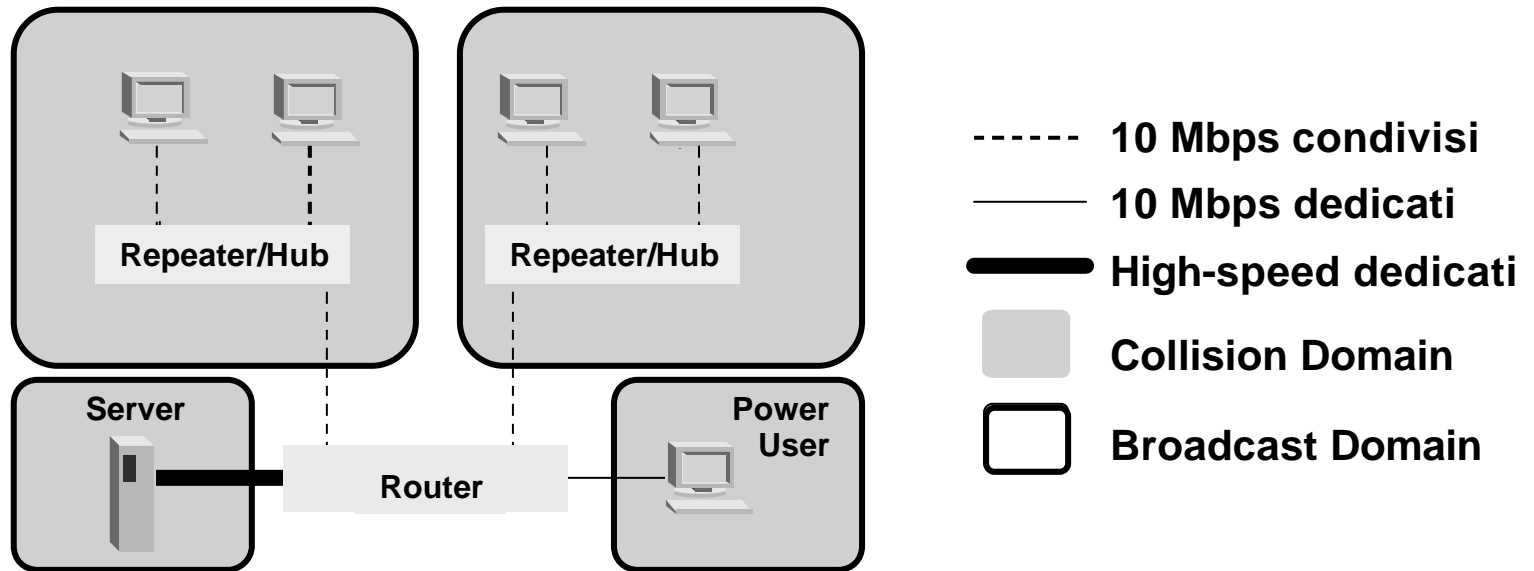
# Un piccolo esercizio di design



- Gruppo di lavoro prima dell'installazione di un dispositivo di internetworking
- Quale dispositivo installare, uno switch o un router, per:
  - ⚙ massimizzare la banda disponibile al server
  - ⚙ suddividere i singoli host in domini di collisione più piccoli con 10 MBps di banda condivisa
  - ⚙ assegnare una banda dedicata di 10 Mbps ad un numero limitato di power users

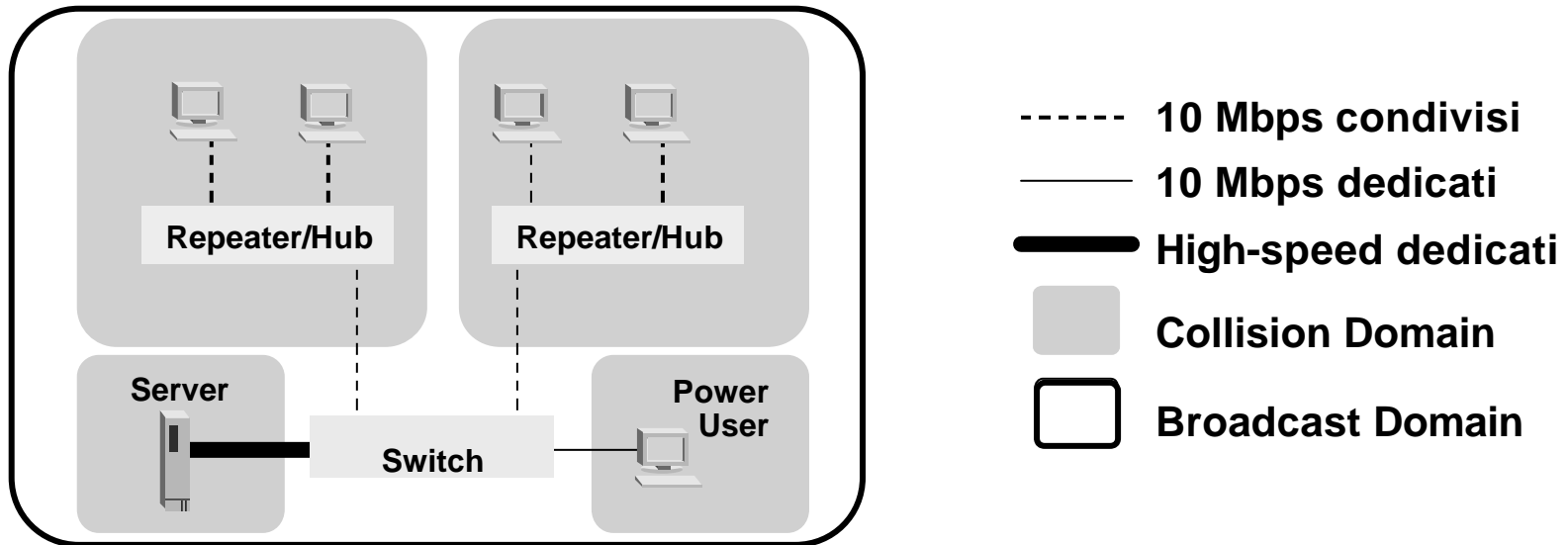


# Soluzione 1: Router



- ↓ Crea molteplici domini di collisione/broadcast
- ↓ Probabilmente non la migliore soluzione in termini economici:
  - elevato costo per porta
- ↓ Probabilmente non la migliore soluzione in termini tecnologici:
  - throughput minore rispetto ad uno switch
  - il livello del traffico di broadcast non giustifica l'utilizzo di un router

## Soluzione 2: Switch



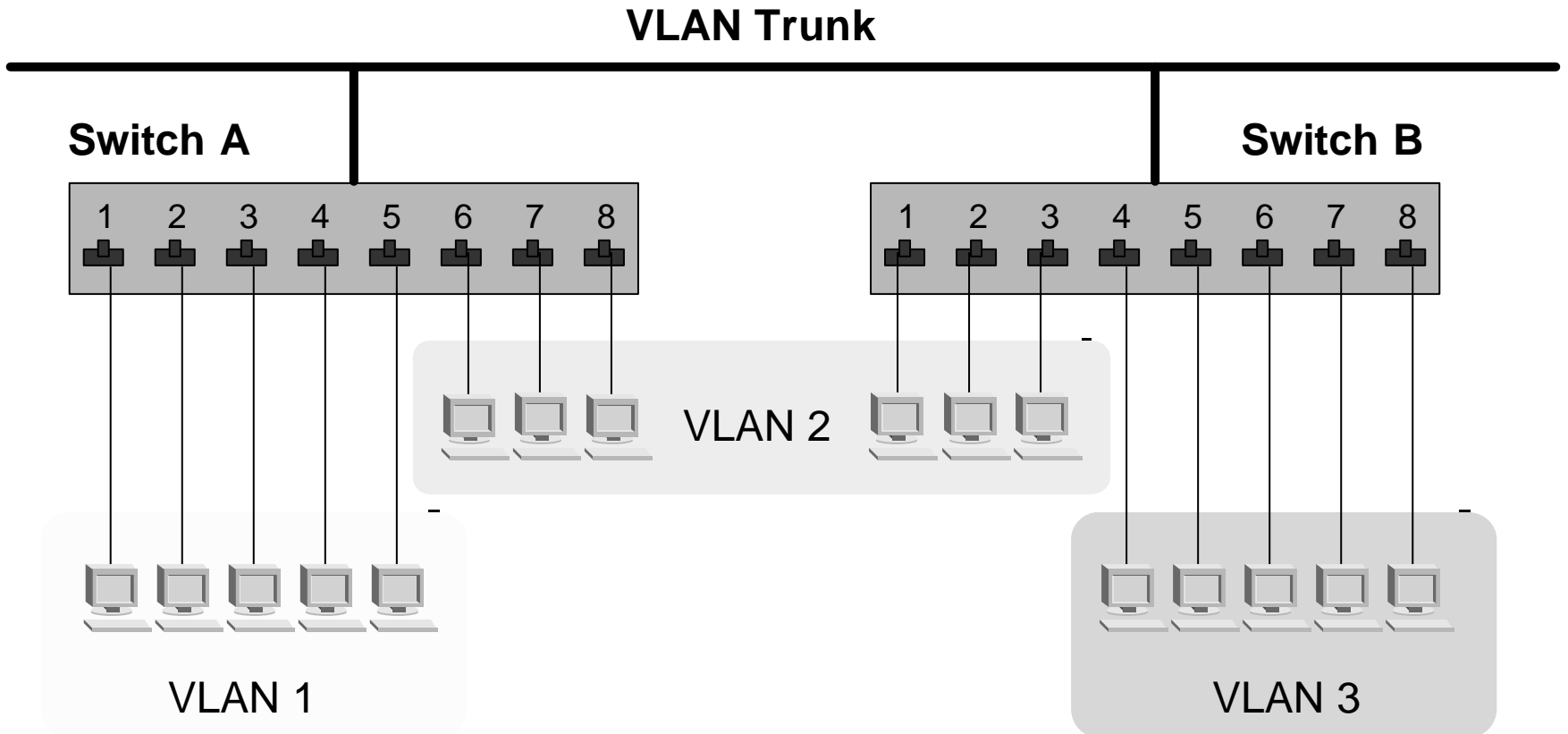
- ⇓ Crea molteplici domini di collisione all'interno di un singolo dominio di broadcast
- ⇓ Uno switch ha diversi vantaggi rispetto ad un router:
  - maggior throughput
  - minor costo per porta
  - più semplice da configurare e gestire

# Lan virtuali (VLAN)

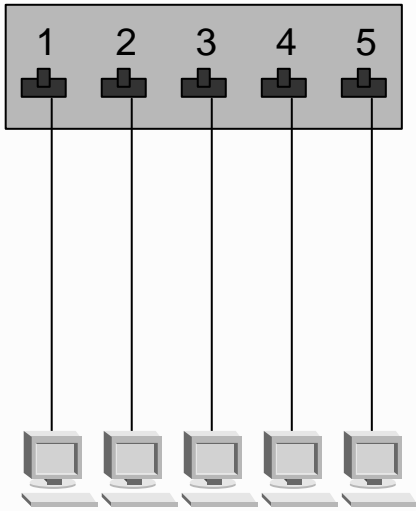
- ⇓ Molti switch moderni consentono di realizzare LAN virtuali (VLAN), cioè insieme di stazioni che si comportano come fossero reti separate (quindi non condividono i broadcast) e che possono essere composte da stazioni collegate a uno o più switch, comunque distribuite all'interno dell'azienda
- ⇓ La realizzazione di una VLAN ha tre vantaggi principali:
  - semplifica la gestione degli spostamenti di stazioni in reti IP
  - consente di controllare il traffico di broadcast
  - aumenta la sicurezza della rete

# VLAN basata su assegnazione delle porte

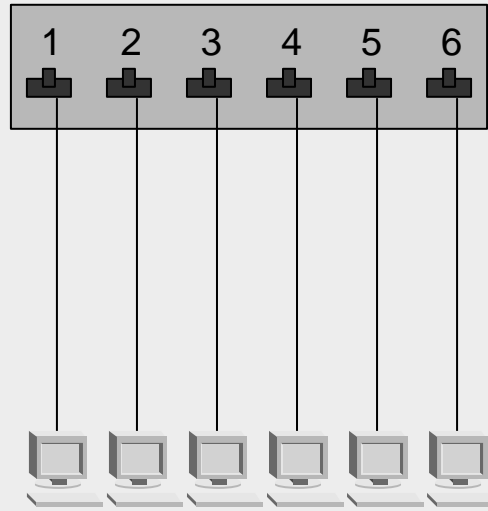
↓ Ogni porta dello switch è assegnata ad una singola VLAN



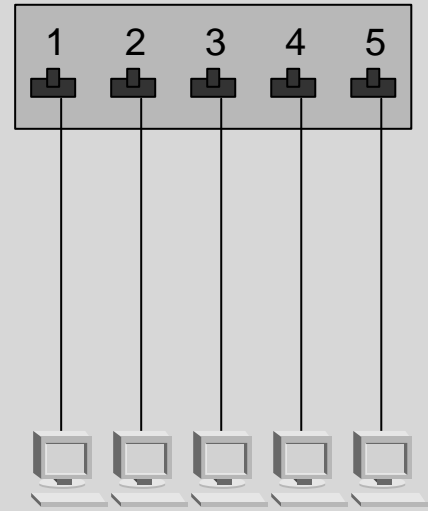
# VLAN: layout logico equivalente



VLAN 1



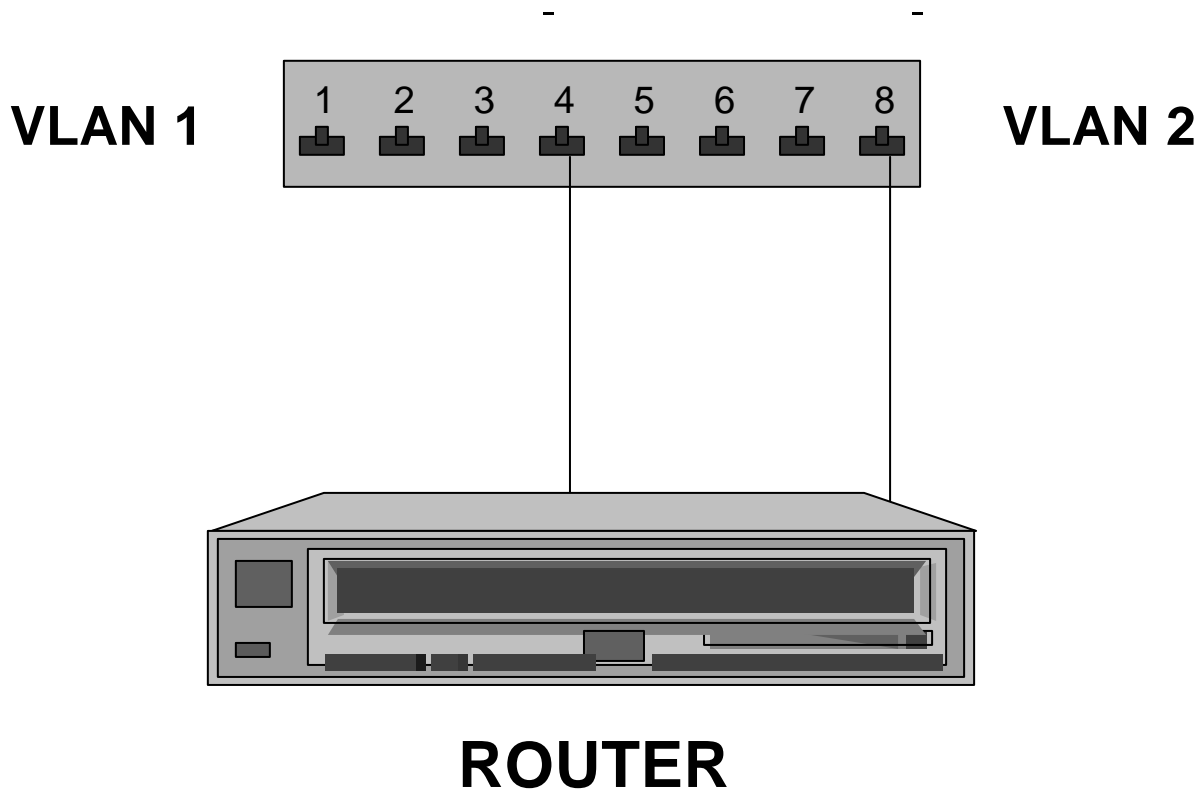
VLAN 2



VLAN 3

# Interconnessione di VLAN

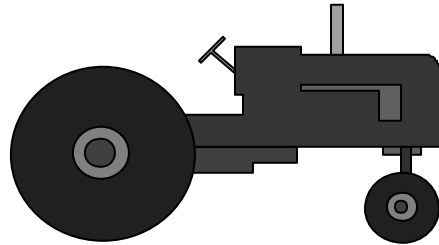
↓ Per interconnettere differenti VLAN è necessario un router



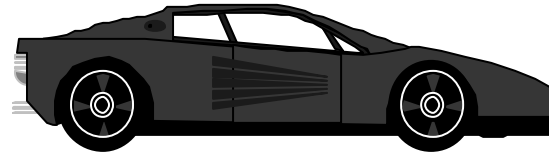
# Tecnologie di WAN

# WAN: technologie (1)

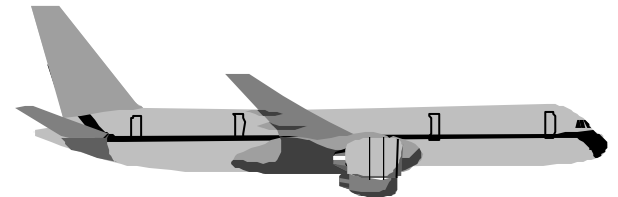
*X.25*



*Frame Relay*



*ATM*





## WAN: tecnologie (2)

<b>Tecnologia</b>	<b>Velocità max.</b>
<b>X.25</b>	<b>64 Kb/s</b>
<b>Frame Relay</b>	<b>155 Mb/s</b>
<b>ATM</b>	<b>2.4 Gb/s</b>

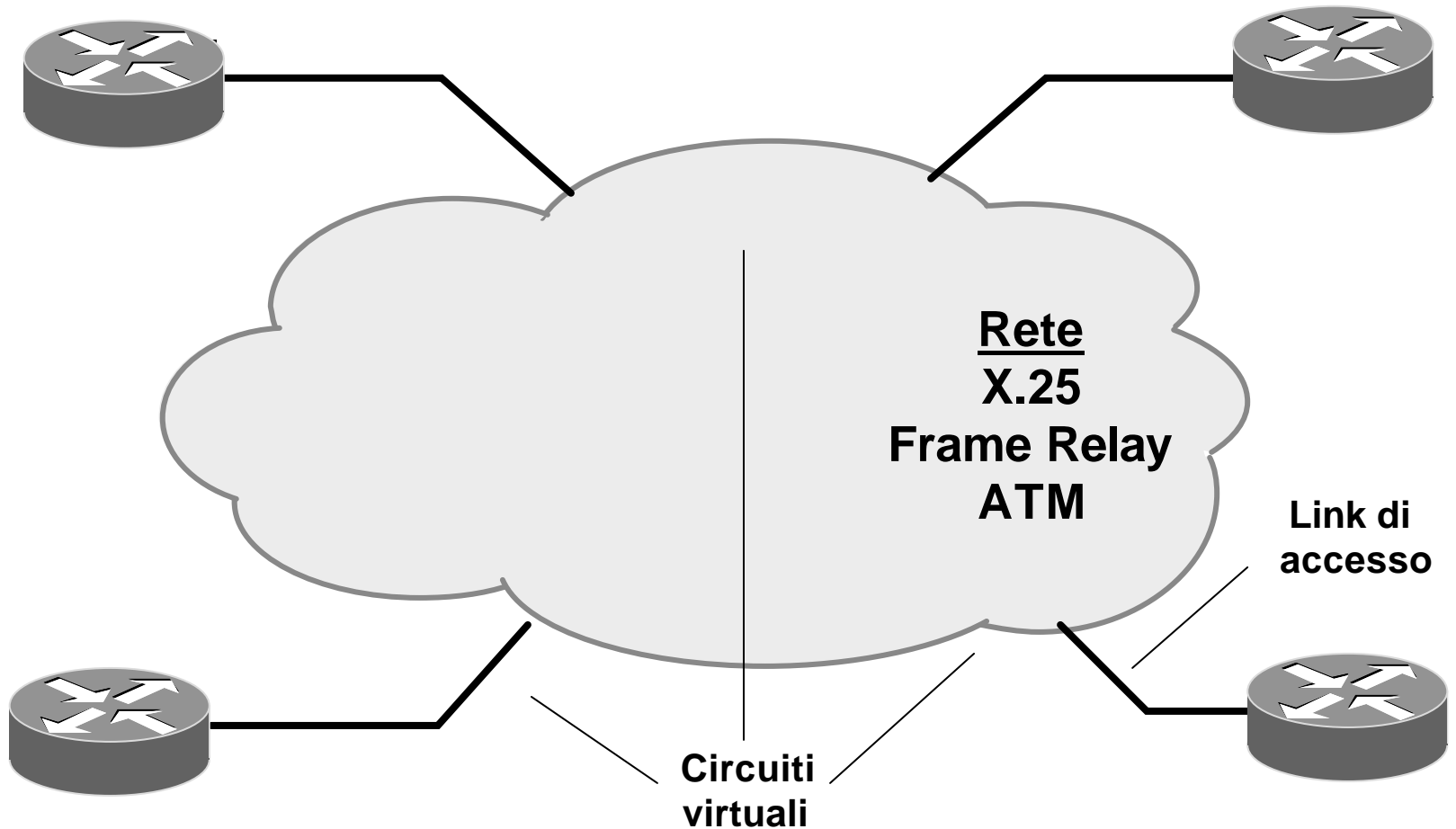
# Connessioni virtuali

- ⇓ X.25, Frame Relay e ATM sono tecniche orientate alla connessione: prima della effettiva trasmissione delle informazioni occorre predisporre il cammino che sarà seguito dai dati attraverso tutta la rete tra i due estremi che devono comunicare tra loro
- ⇓ Le connessioni sono di tipo virtuale nel senso che non prevedono l'allocazione preventiva di risorse trasmissive e/o di commutazione, bensì solo un'associazione logica di corrispondenze tra i vari punti delle connessioni
- ⇓ Le connessioni (o circuiti) virtuali possono essere di due tipi:
  - permanenti (PVC, Permanent Virtual Connection\Circuit), ovvero instaurate tramite apposito centro di gestione
  - su base chiamata (SVC, Switched Virtual Connection\Circuit), ovvero instaurate tramite apposito sistema di segnalazione

# Connessioni virtuali: label

- ↓ Le connessioni virtuali vengono identificate attraverso una etichetta (label), che ha un significato locale all'interfaccia UNI
- ↓ La concatenazione di n label (su n link trasmissivi separati da n-1 nodi) costituisce la connessione virtuale stabilita tra due apparati terminali della rete
- ↓ La label viene denominata in maniera differente in dipendenza della tecnologia di WAN utilizzata:
  - X.25: Logical Channel Identifier (LCI)
  - Frame Relay: Data Link Connection Identifier (DLCI)
  - ATM: Virtual Path Identifier/Virtual Channel Identifier (VPI/VCI)
- ↓ Il nodo di commutazione effettua un "label swapping"

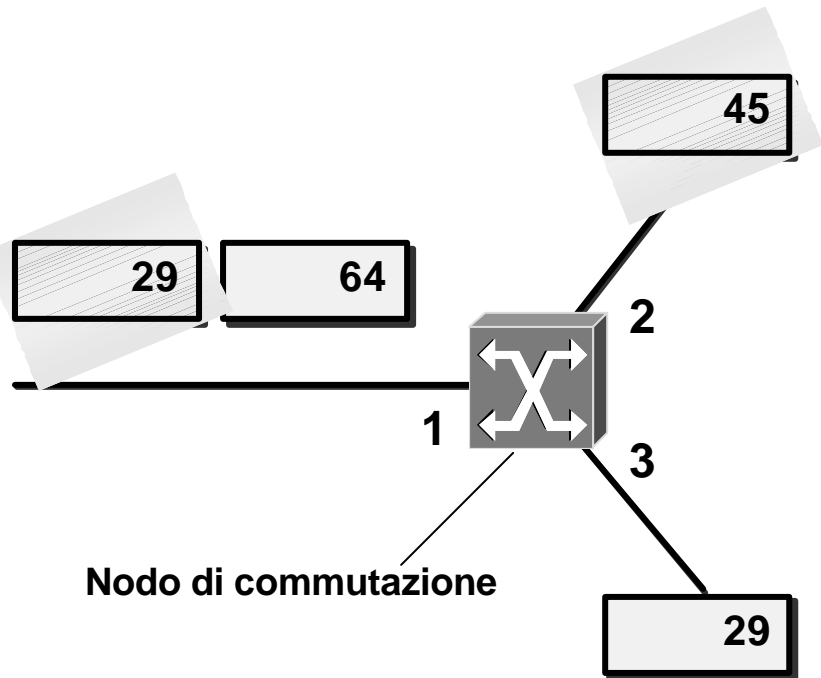
# Circuiti Virtuali



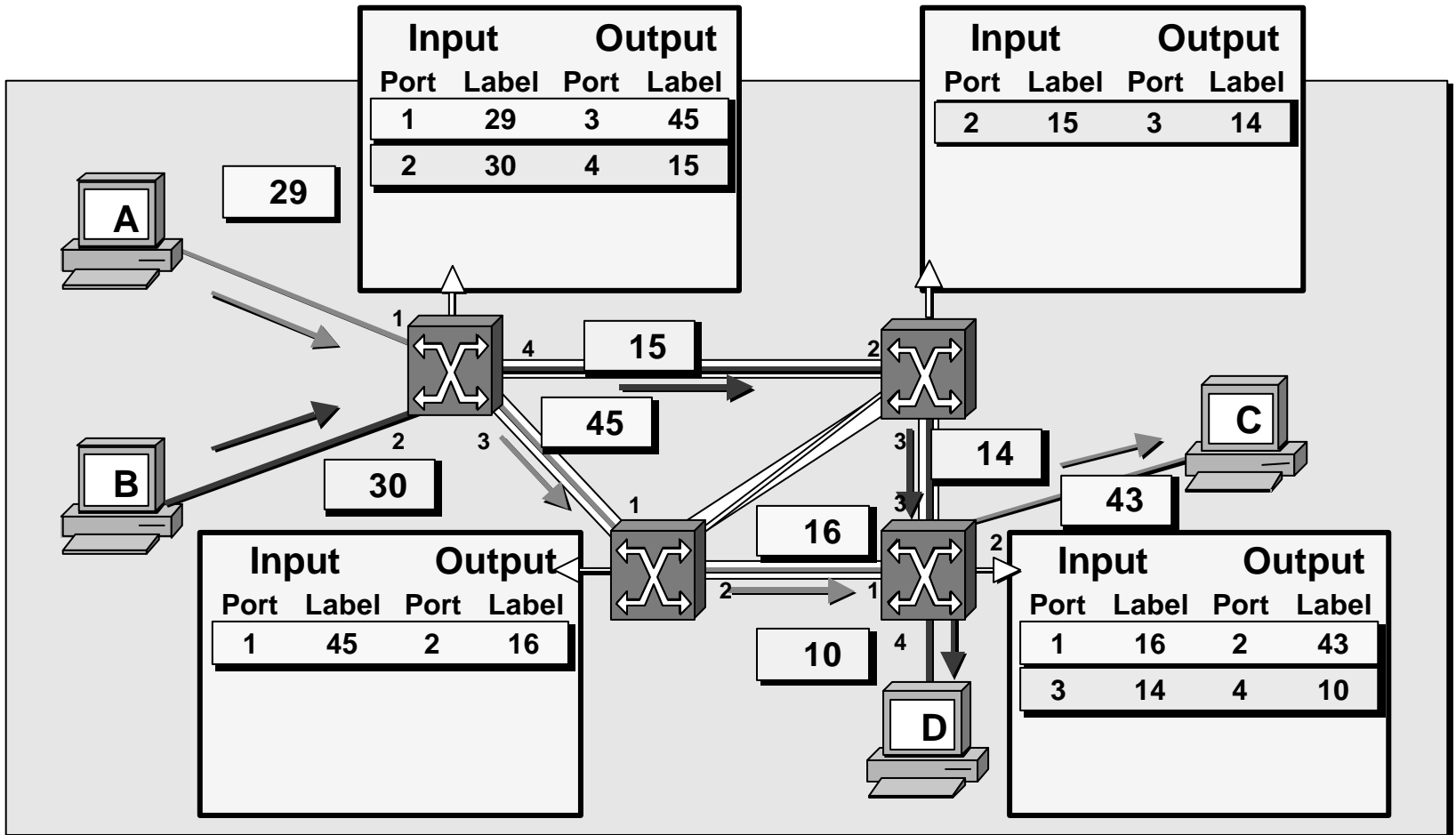
# Label Swapping

Input		Output	
Port	Label	Port	Label
1	29	2	45
2	45	1	29
1	64	3	29
3	29	1	64

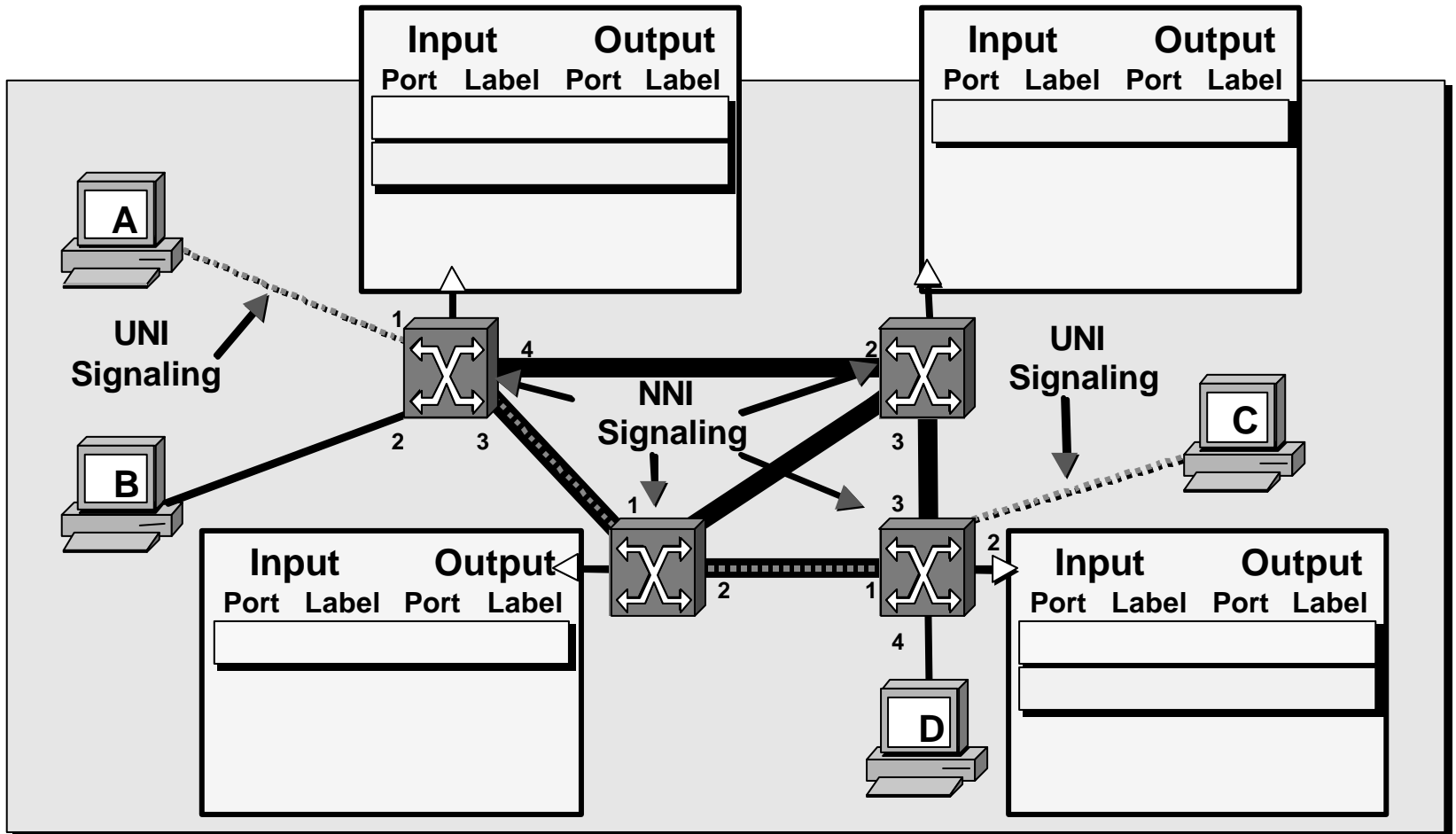
**Look-up Table**



# Permanent Virtual Circuit (PVC)

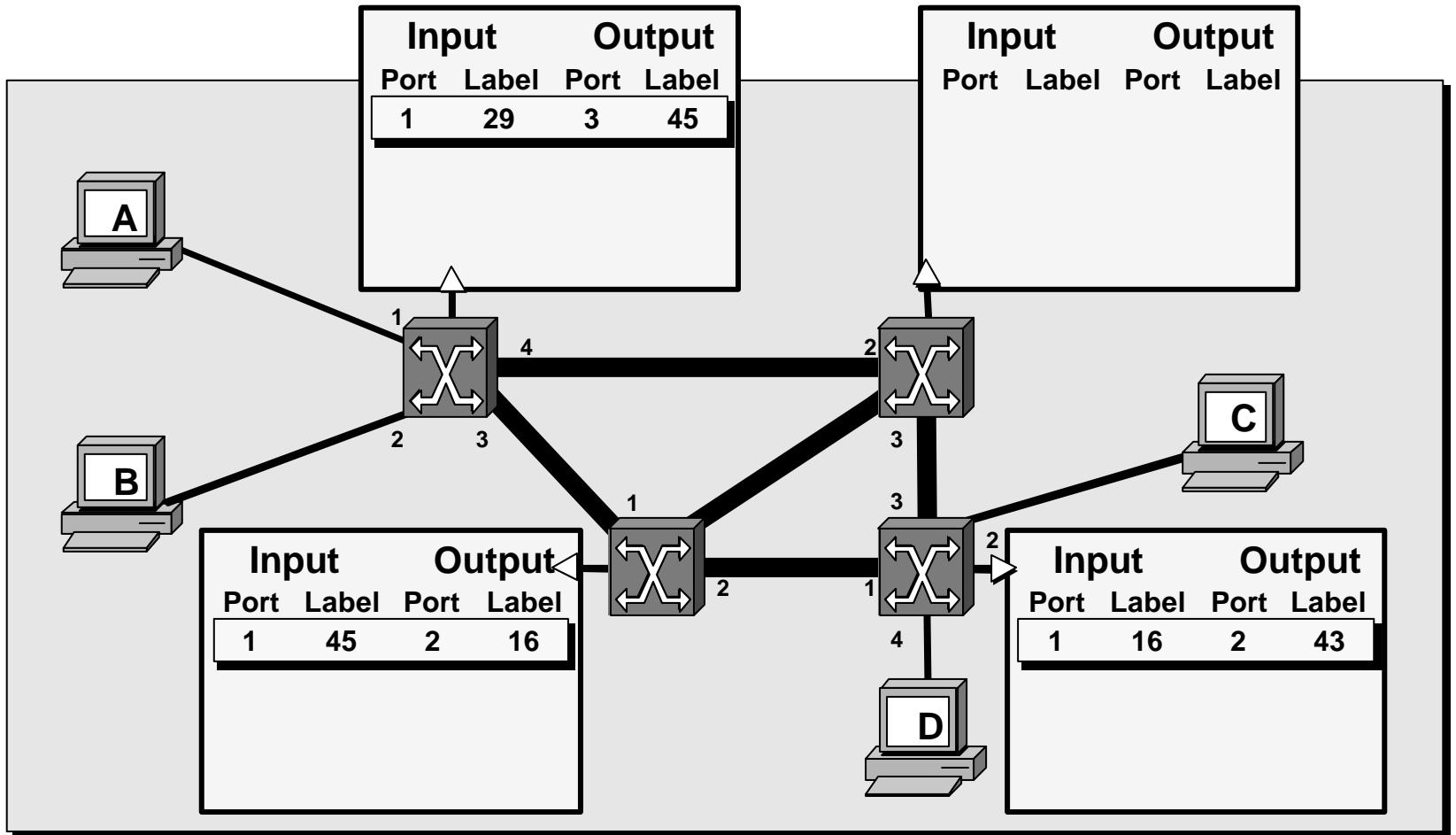


# Switched Virtual Circuit (SVC)



**SVC: fase di apertura della connessione (attraverso segnalazione)**

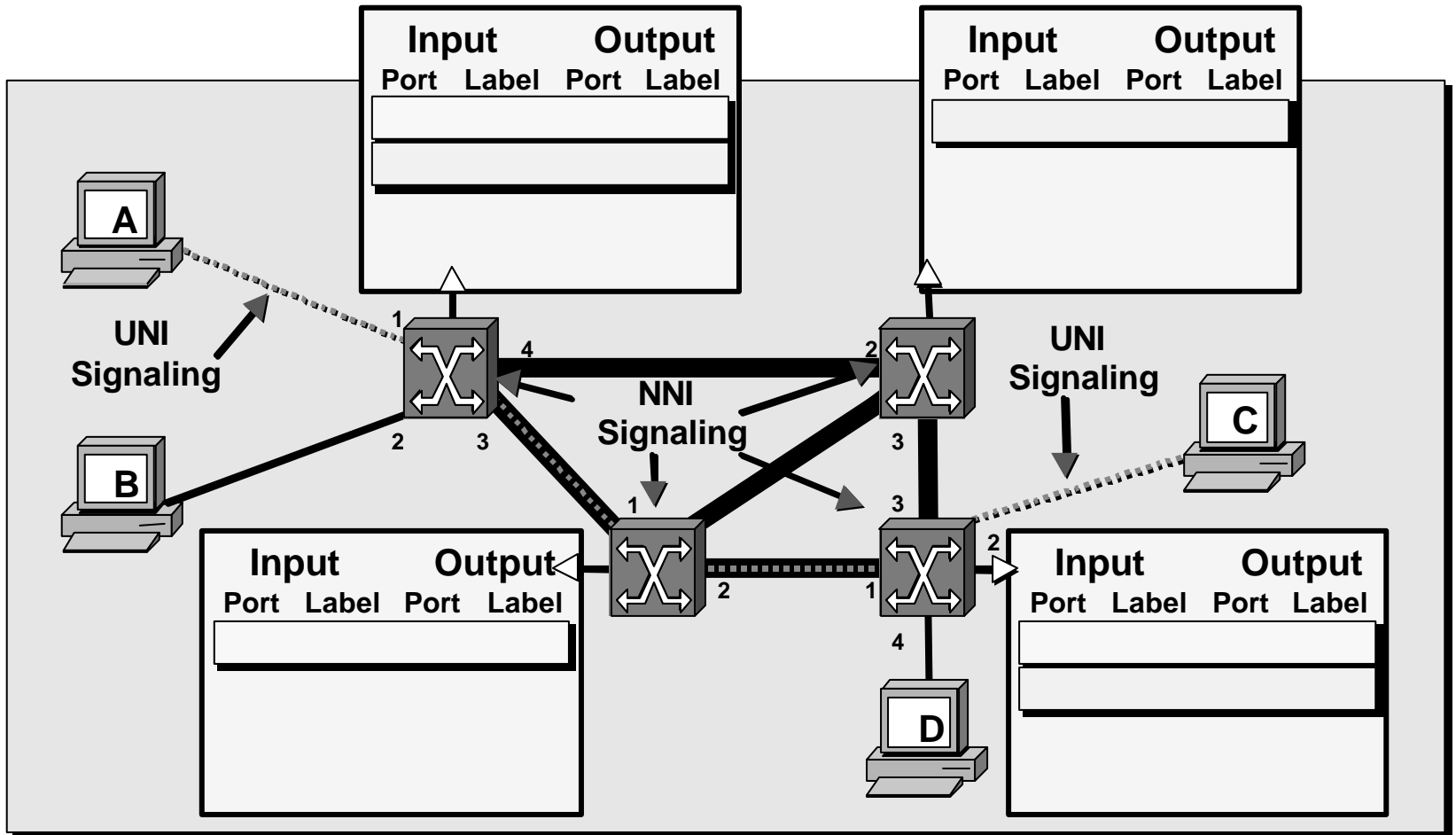
# Switched Virtual Circuit (SVC)



**SVC: fase di trasferimento dati**



# Switched Virtual Circuit (SVC)



**SVC: fase di chiusura della connessione (attraverso segnalazione)**

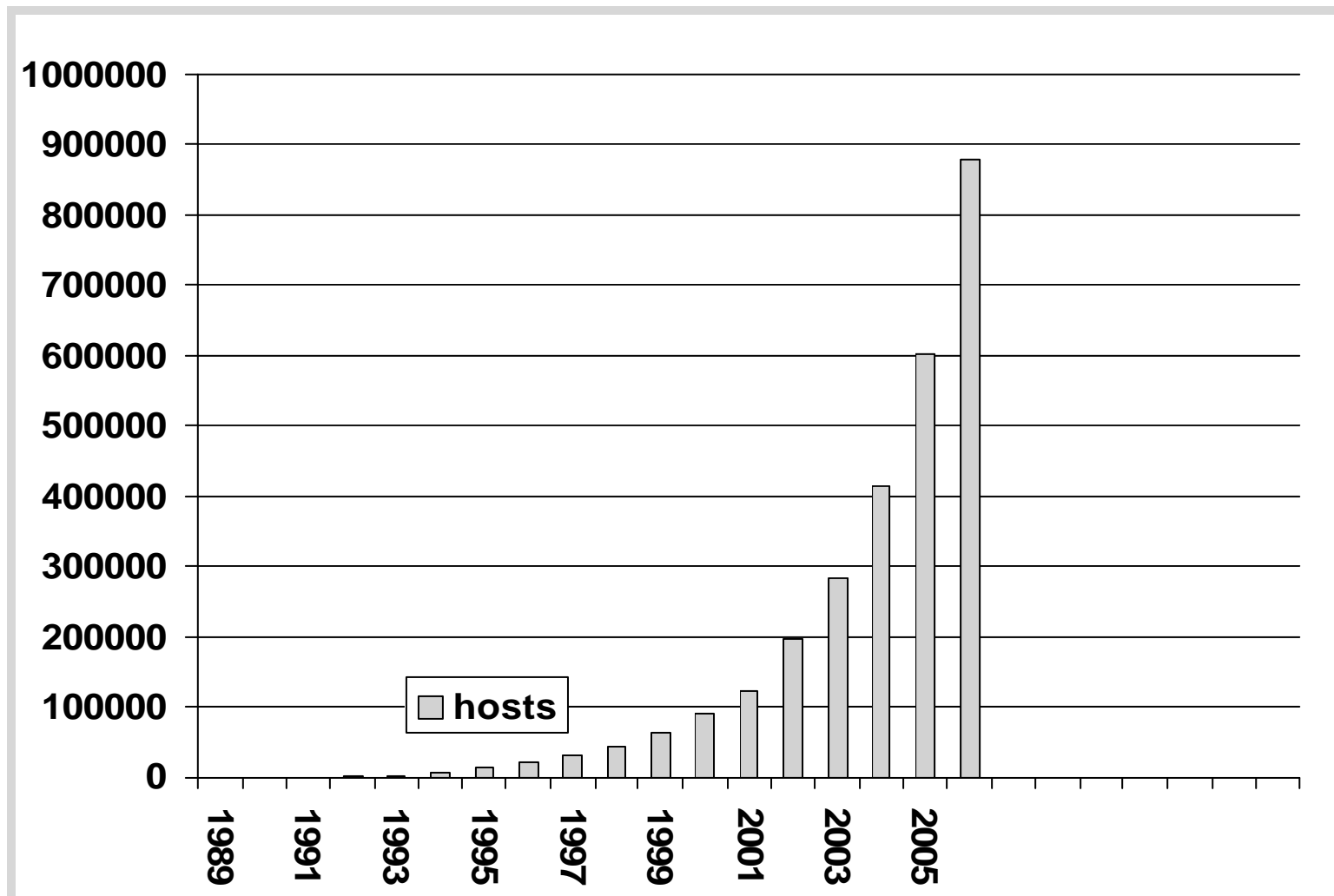
2° PARTE  
L' ARCHITETTURA TCP/IP

# La storia di Internet

# Internet: statistiche

- 5.6 Milioni Domini Internet ([www.nw.com](http://www.nw.com))
- 56 Milioni di Host ([www.nw.com](http://www.nw.com))
- 246 "IP countries" ([www.nw.com](http://www.nw.com))
- 201 Milioni di utenti ([www.nua.ie](http://www.nua.ie))
- 75% del traffico su Internet è WWW
- 3 Milioni di siti Web
- 8000 ISPs nel mondo (4700+ negli U.S.A.)

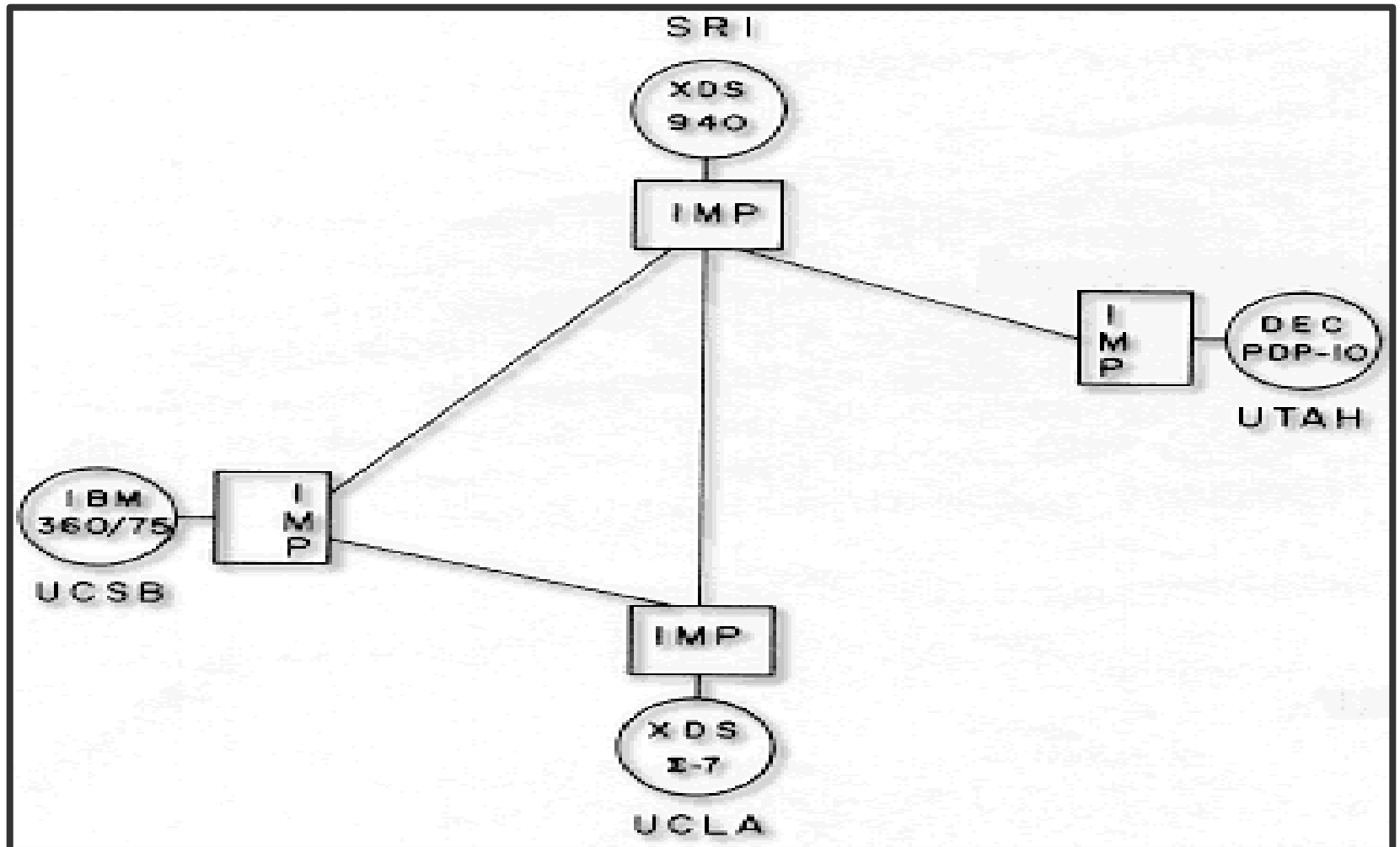
# Internet Hosts (000s) 1989-2006



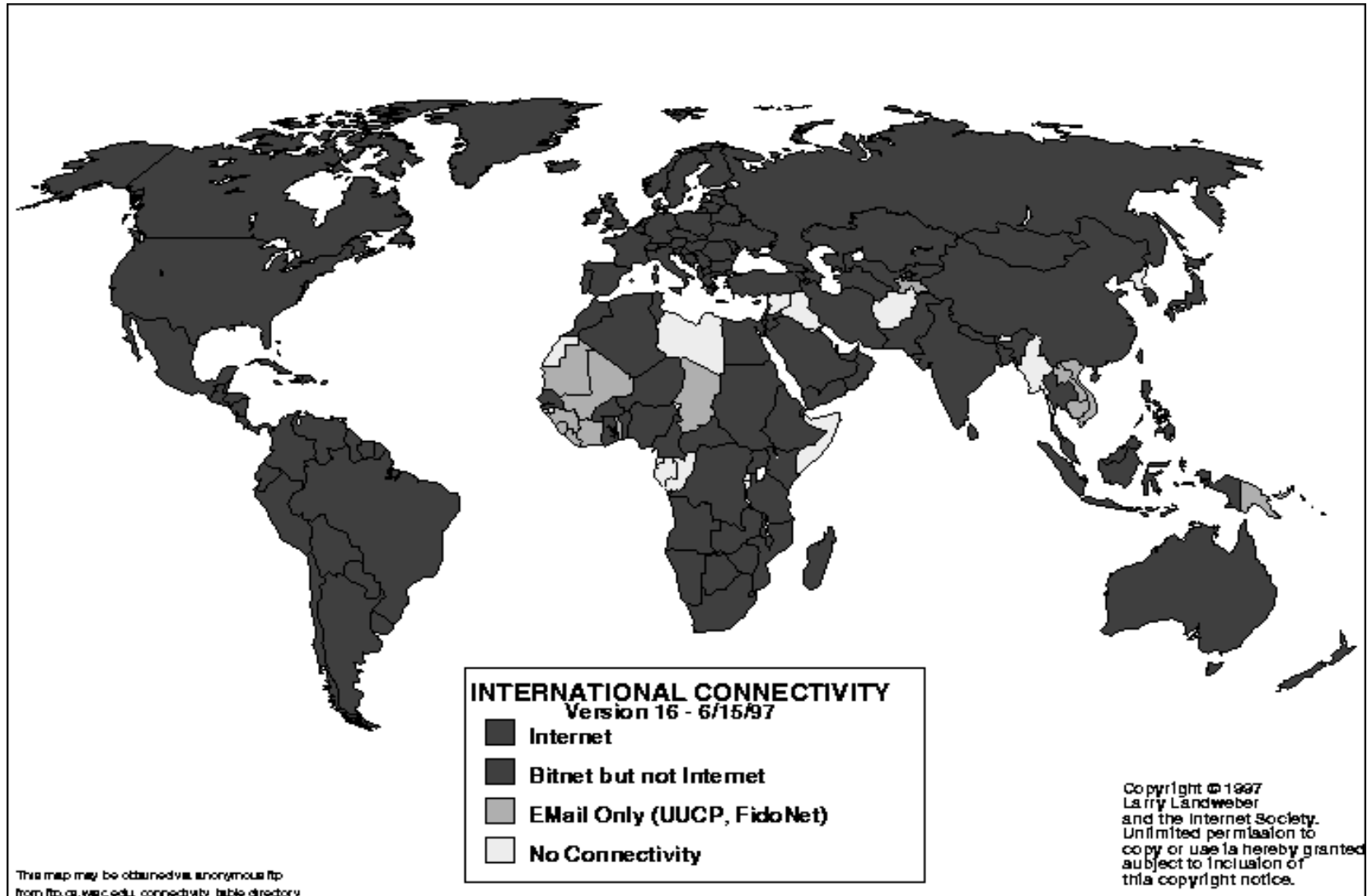
# Internet: storia

- ↴ Le origini di Internet si possono far risalire al progetto DARPA (Defense Advanced Research Project Agency) del DOD Americano (inizi anni 70')
- ↴ Necessità di interconnettere reti dei centri militari, universitari e di ricerca: definizione della rete ARPANET
- ↴ 1973 viene commissionato all'Università di Stanford il progetto di una suite di protocolli standard che garantissero connettività a livello di rete
- ↴ Verso la fine degli anni '70, tale sforzo portò al completamento dell'Internet Protocol Suite, di cui i due protocolli più noti sono il TCP e l'IP
- ↴ Il nome più appropriato per l'architettura di rete rimane quello di Internet Protocol Suite, anche se comunemente si fa riferimento ad essa con la sigla TCP/IP
- ↴ I protocolli appartenenti a questa architettura sono specificati tramite standard denominati RFC (Request For Comments) ([WWW.IETF.ORG](http://WWW.IETF.ORG))

# ARPANET: dicembre 1969

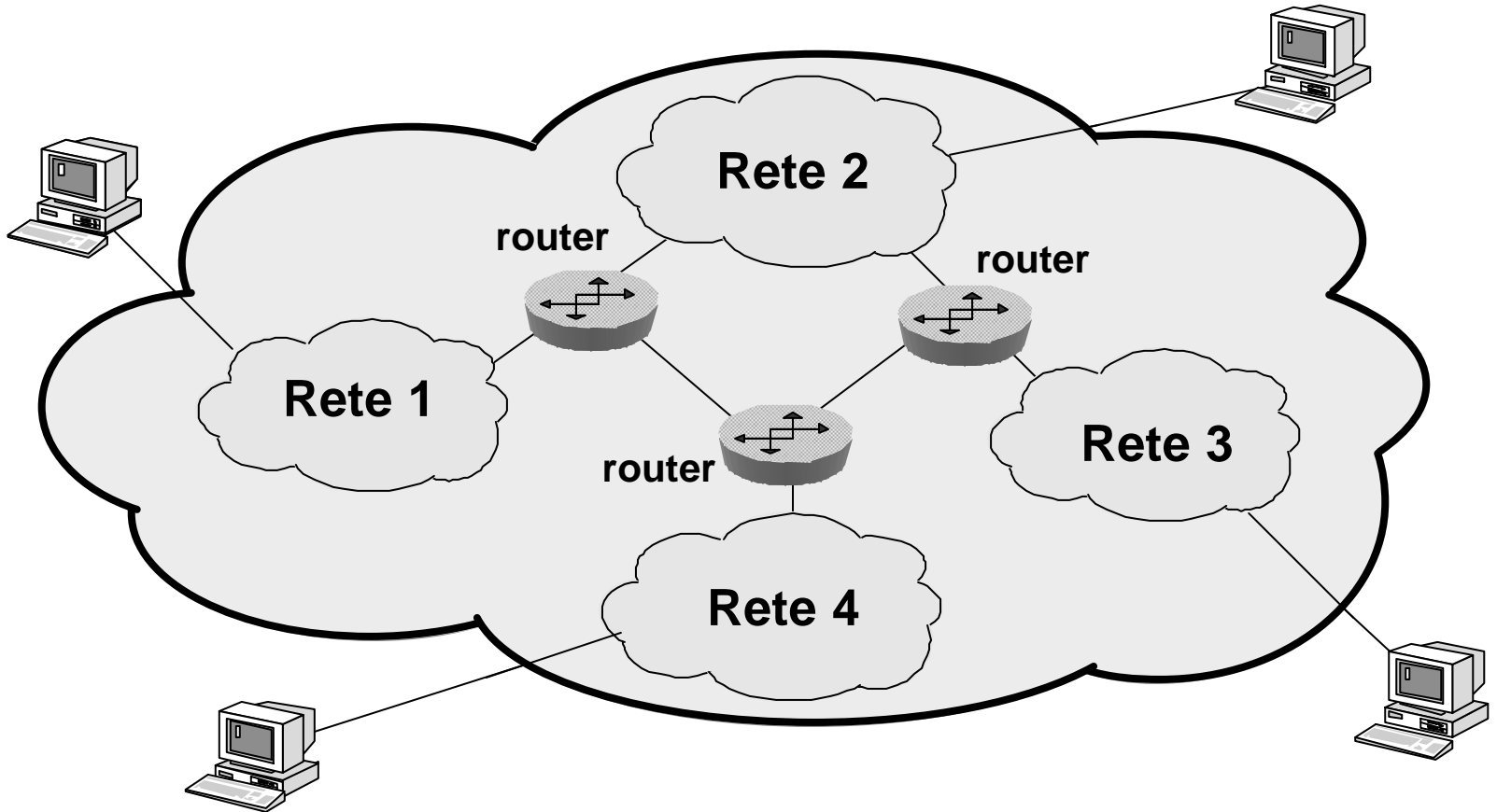


# Internet: dal 1991 al 1997



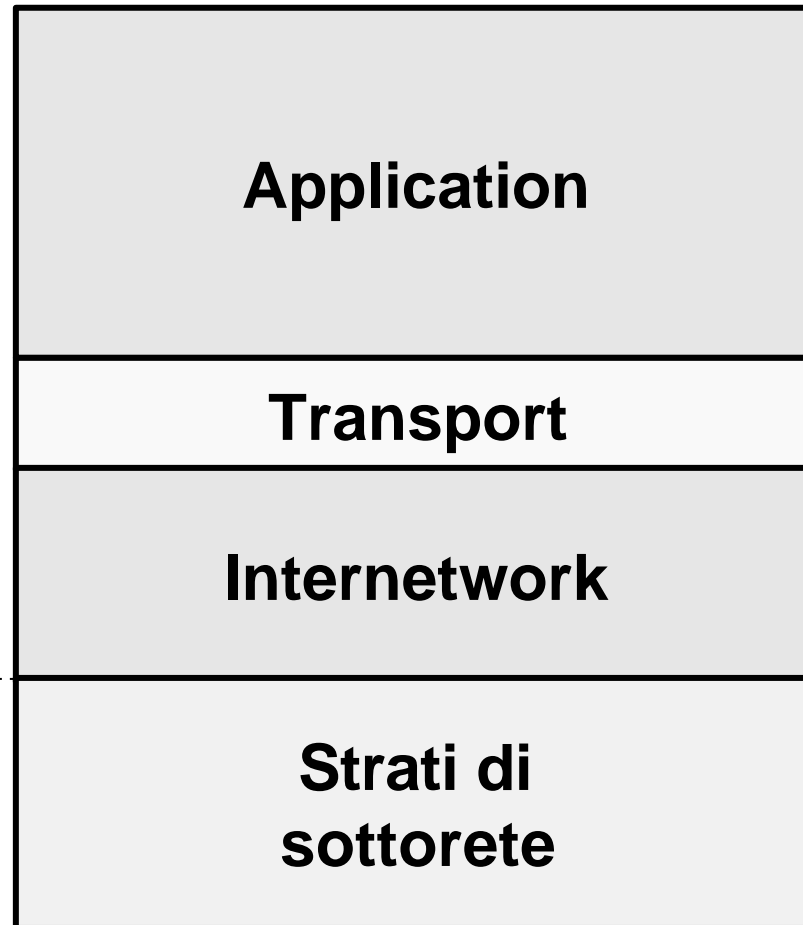


# Internet: topologia



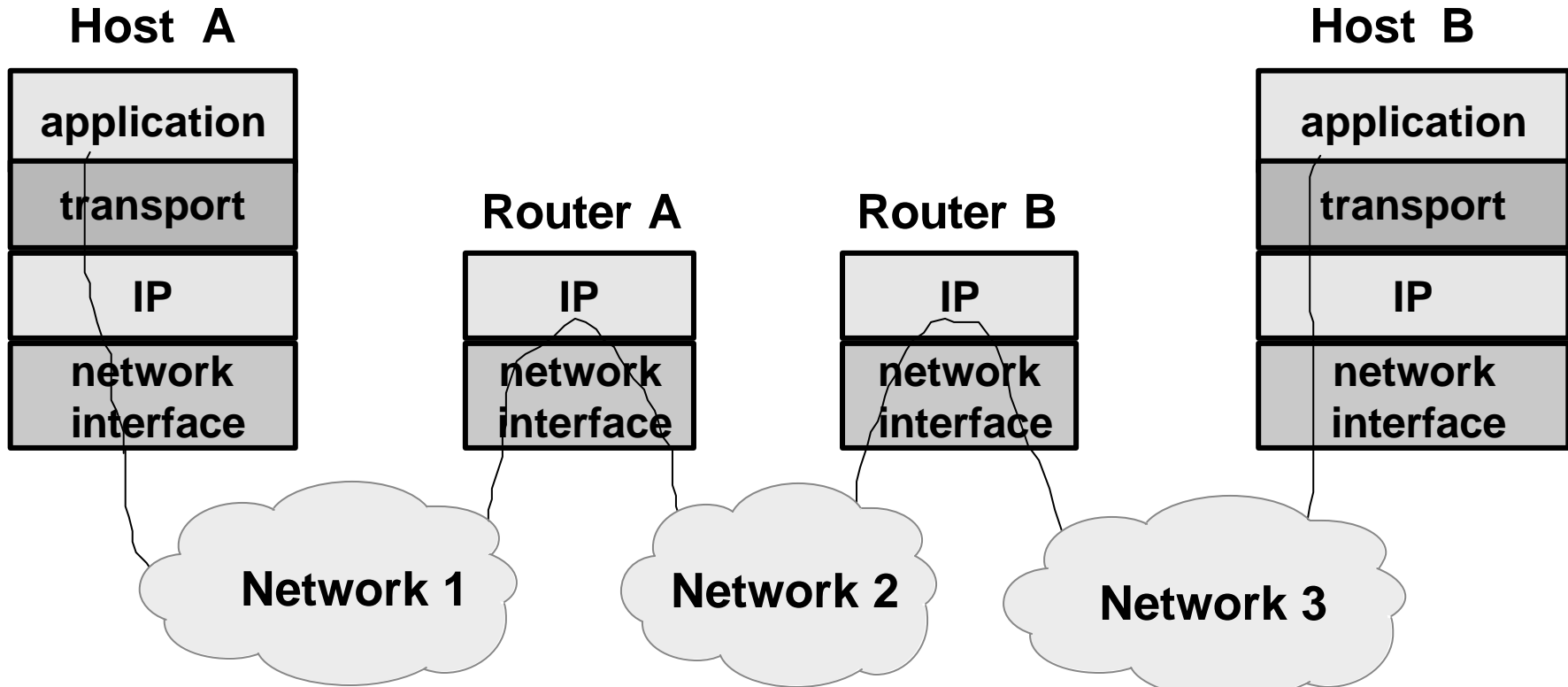
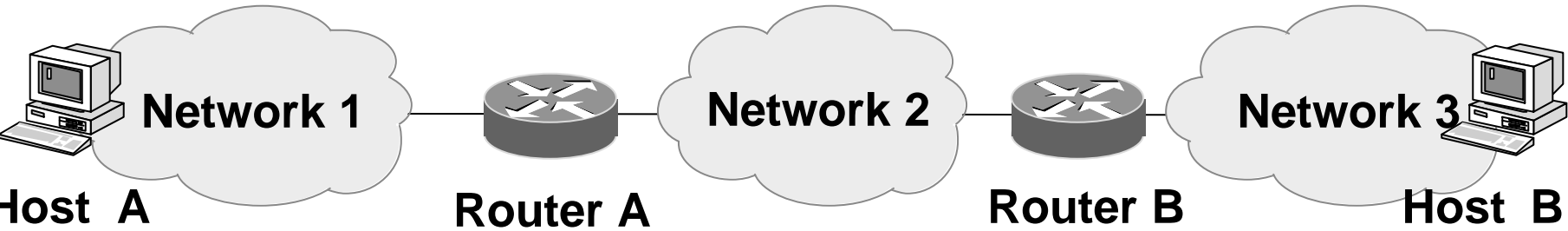
# Architettura TCP/IP

↓ L'architettura di comunicazione ha una struttura stratificata:



**Tecnologie di  
sottorete**

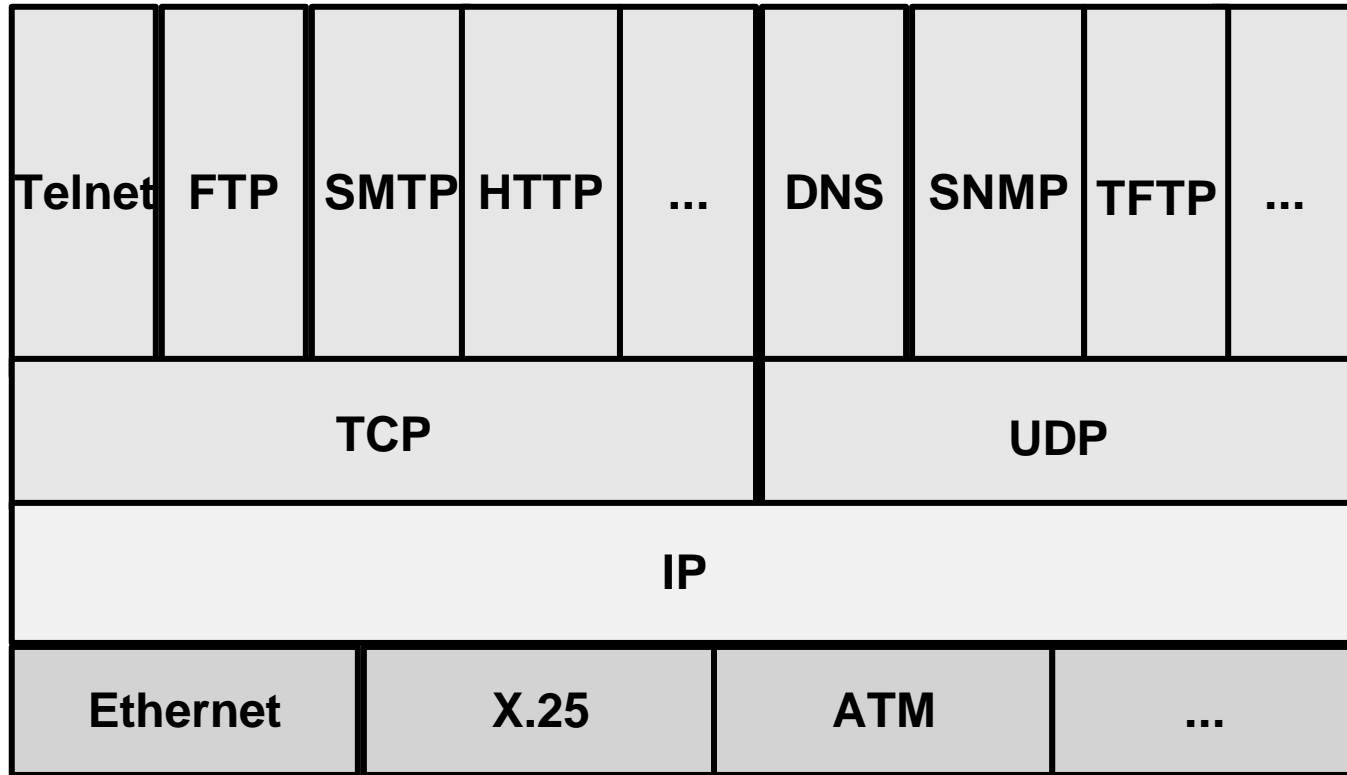
# Architettura TCP/IP



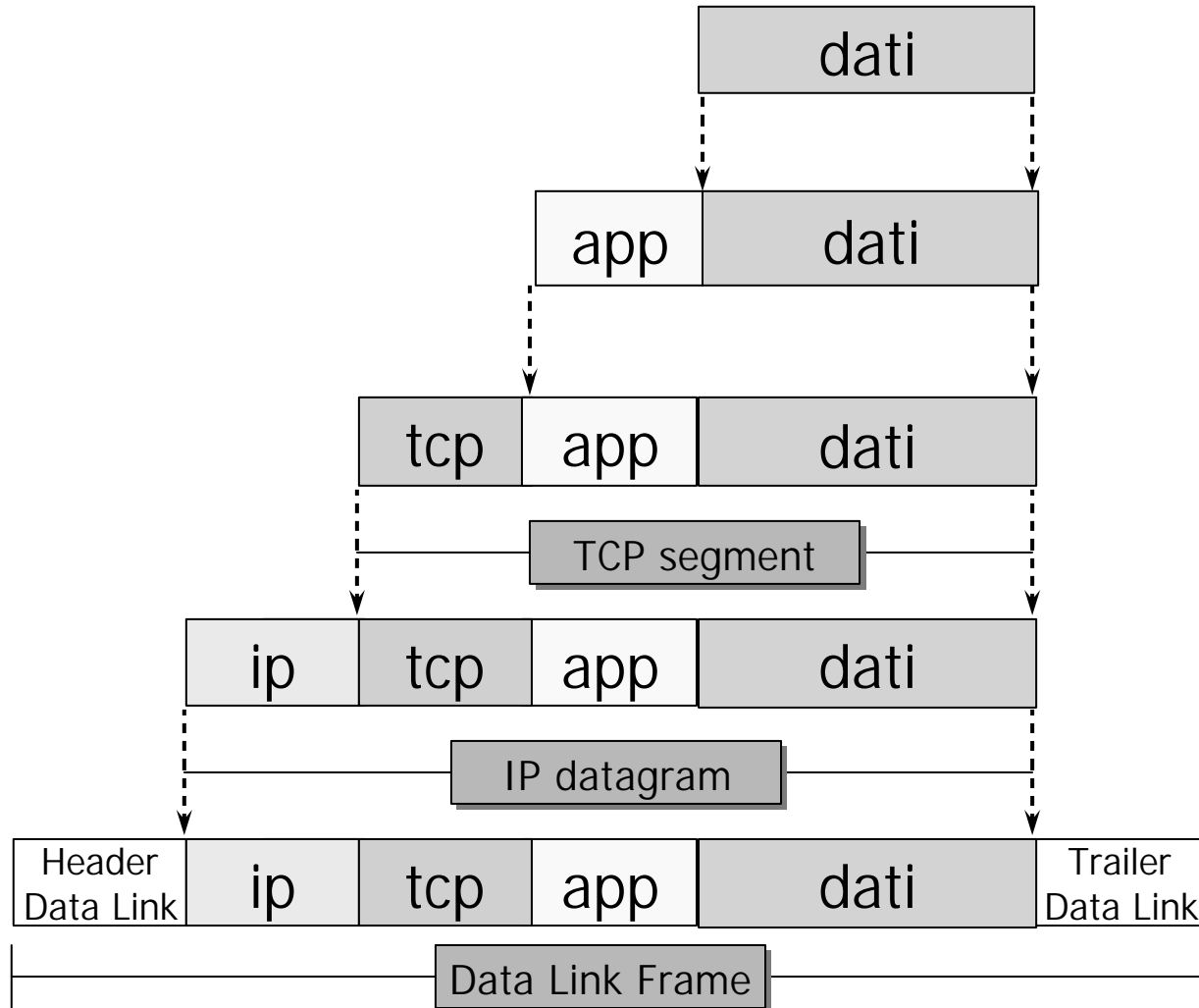
# L'indirizzamento IP

# Architettura TCP/IP

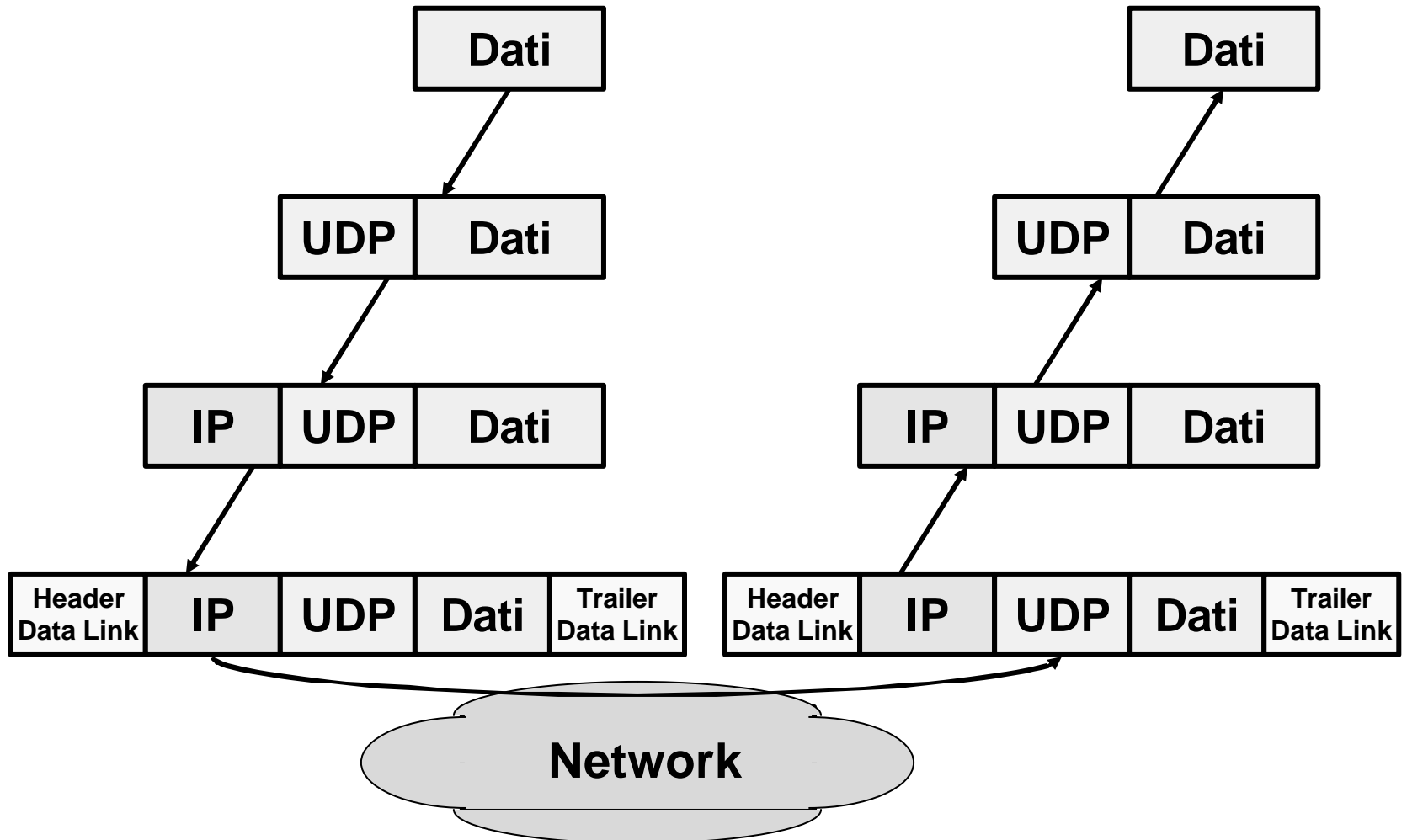
↓ Pila dei protocolli Internet



# Architettura TCP/IP



# Architettura TCP/IP

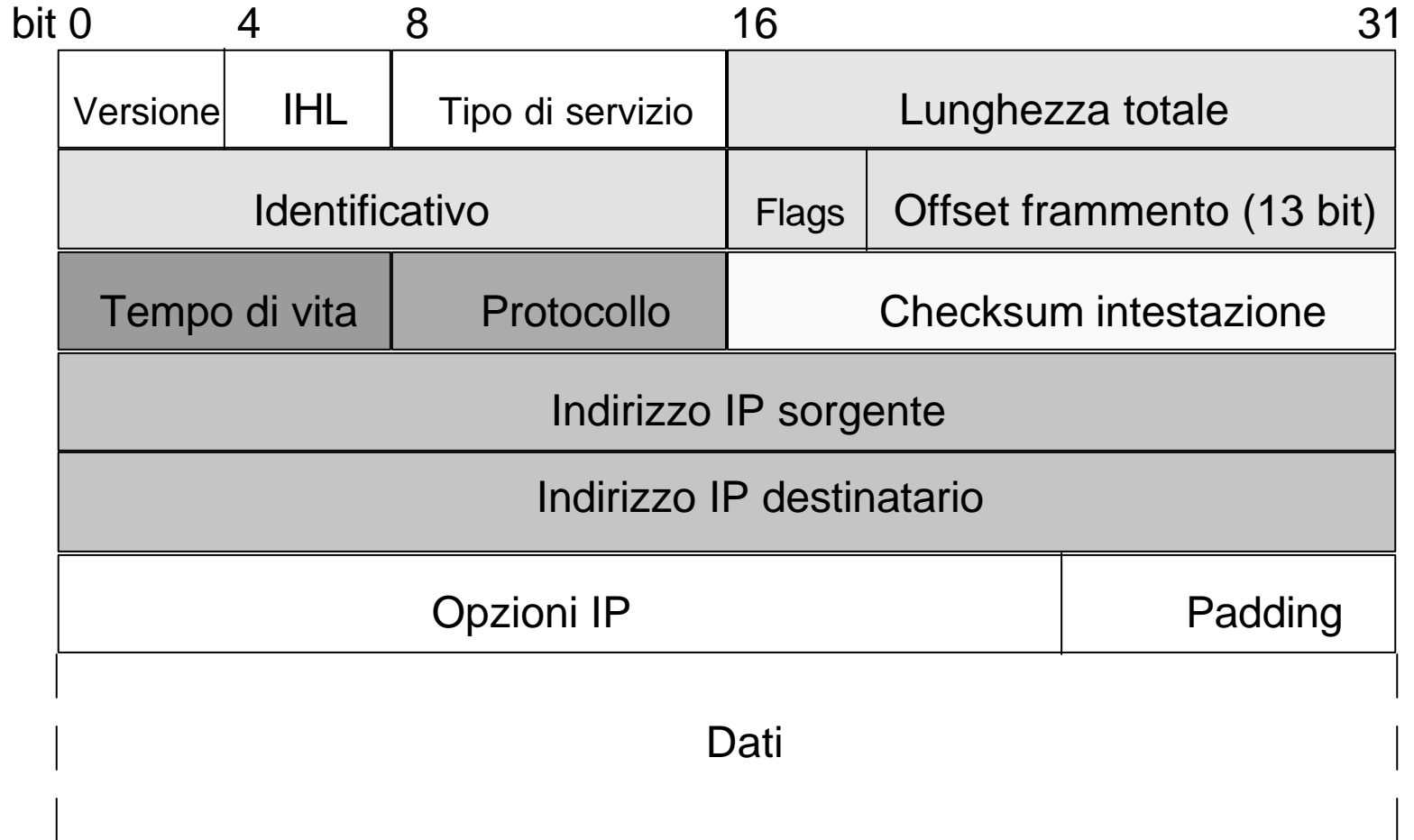


# Il protocollo IP

- ⇓ “IP is the workhorse protocol of the TCP/IP protocol suite” (W. R. Stevens)
- ⇓ Il protocollo IP fornisce un servizio datagram connectionless ed inaffidabile
- ⇓ Il termine inaffidabile significa che non ci sono garanzie che un pacchetto IP giunga a destinazione (servizio best effort)
- ⇓ Il termine connectionless significa che il protocollo IP non mantiene alcuna informazione di stato circa i pacchetti inoltrati. Ciascun pacchetto è trattato indipendentemente da tutti gli altri. Questo significa anche che i datagrammi IP possono essere consegnati fuori sequenza



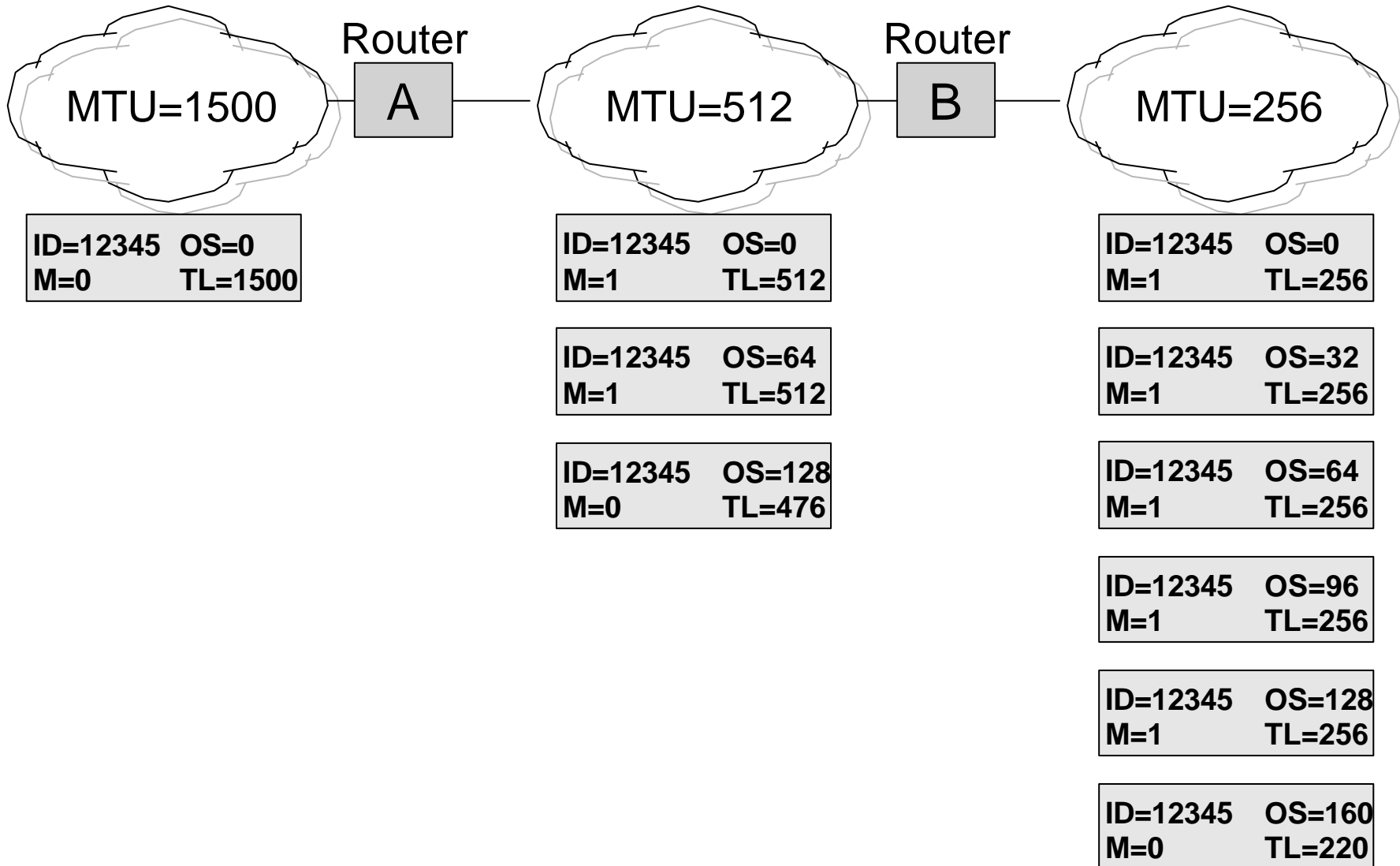
# Header IP



# Header IP

- ↴ Il datagram IP ha una lunghezza variabile:
  - header: 20 - 64byte
  - dati + header  $\leq$  MTU (Maximum Transmission Unit), altrimenti si ha la frammentazione
- ↴ Source address: indirizzo IP mittente
- ↴ Destination address: indirizzo IP destinatario
- ↴ Protocol: protocollo di trasporto
- ↴ Fragment offset: offset del frammento nell'ambito del totale dei dati da trasmettere
- ↴ Identification: identificativo del datagram
- ↴ Flag: indica se il datagram rappresenta un frammento
- ↴ Time-To-Live (TTL): numero di router che un datagram puo' attraversare
- ↴ Header checksum: controllo errore sull'header
- ↴ IHL: lunghezza dell'header
- ↴ Total length: lunghezza totale

# IP: frammentazione



# Indirizzi IP

32 bit

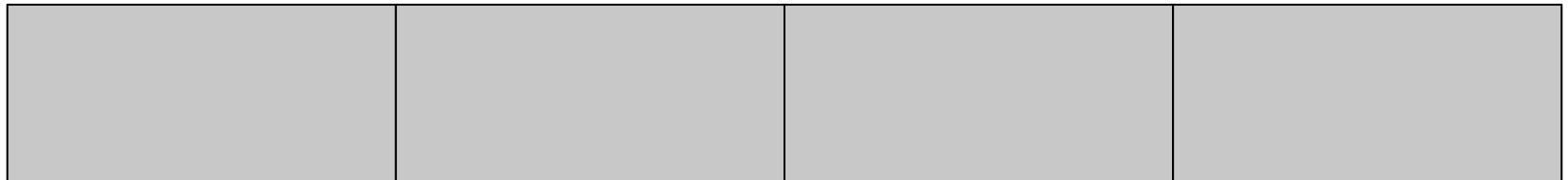


8 bit

8 bit

8 bit

8 bit



**198**

.

**18**

.

**140**

.

**208**

# Indirizzi IP

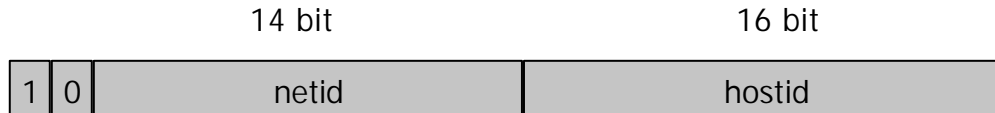
## Classe A

(0.0.0.0 - 127.255.255.255)



## Classe B

(128.0.0.0 - 191.255.255.255)



## Classe C

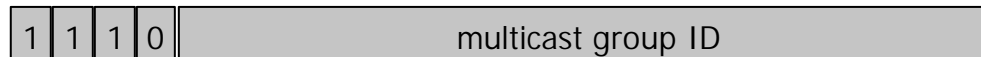
(192.0.0.0 - 223.255.255.255)



## Classe D

28 bit

(224.0.0.0 - 239.255.255.255)



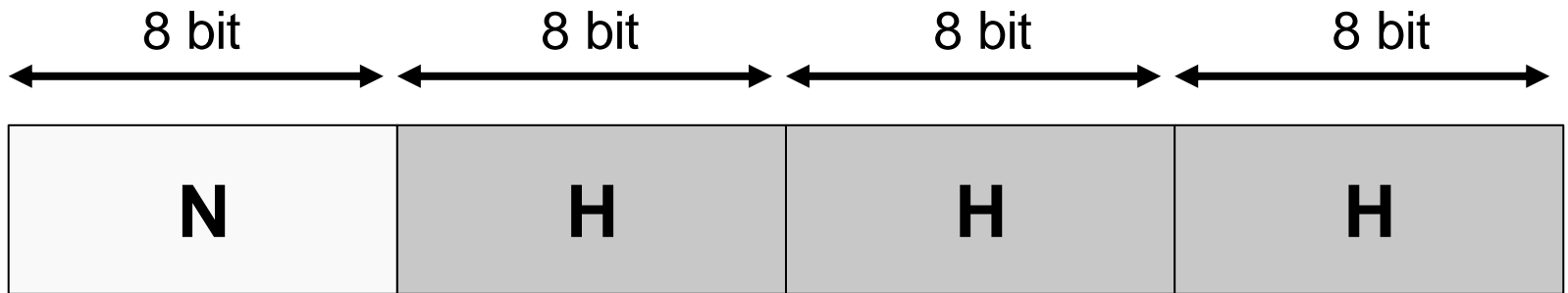
## Classe E

27 bit

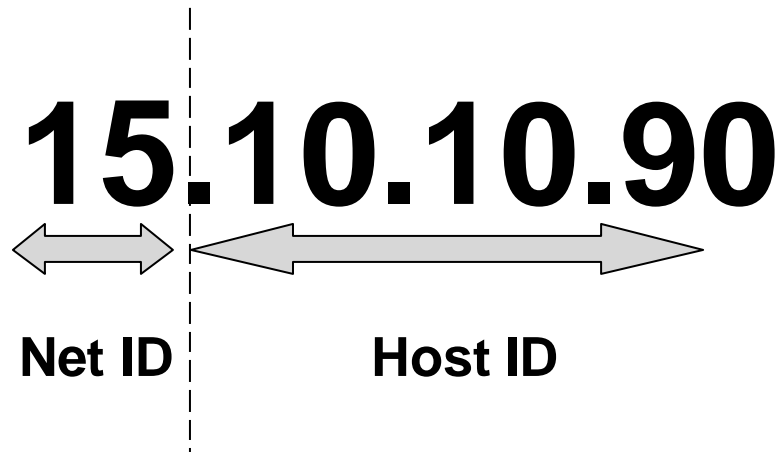
(240.0.0.0 - 255.255.255.254)



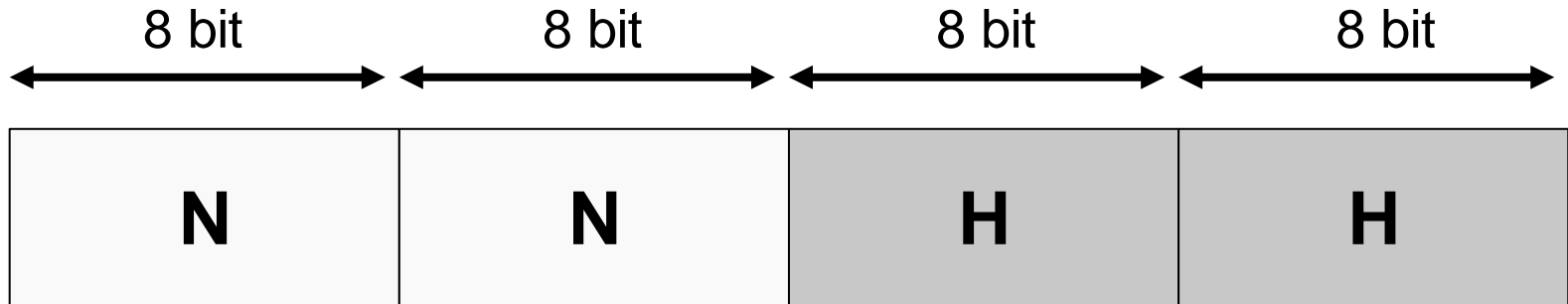
# Indirizzi di classe A



↓ Esempio di indirizzo di classe A:

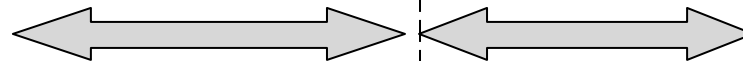


# Indirizzi di classe B



↓ Esempio di indirizzo di classe B:

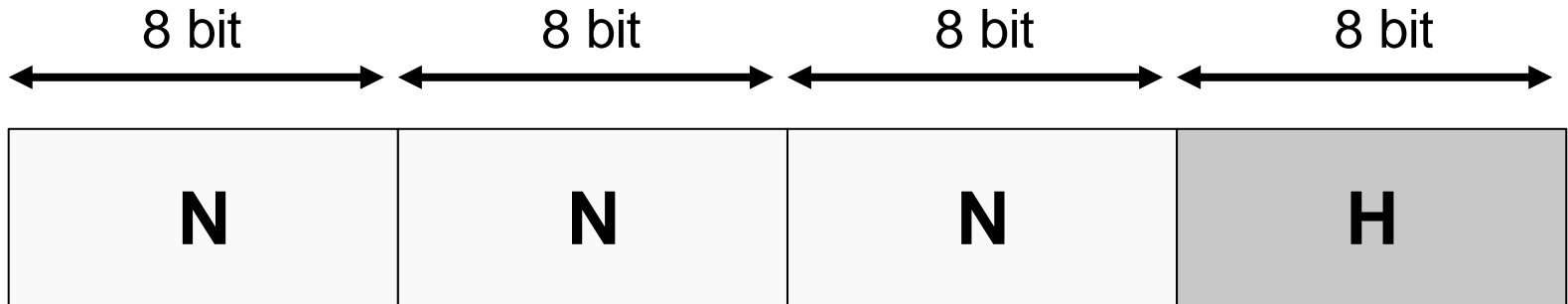
**130.20.18.62**



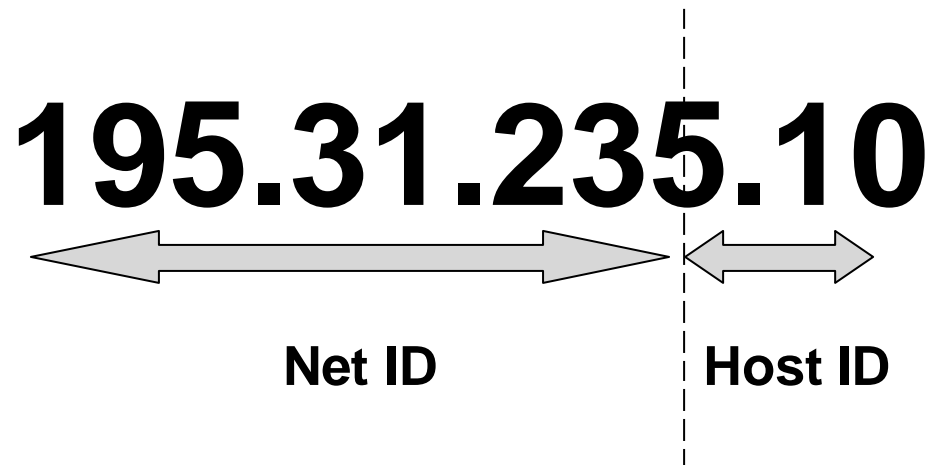
**Net ID**

**Host ID**

# Indirizzi di classe C



⇩ Esempio di indirizzo di classe C:





# Indirizzi IP privati

## **IANA-Allocated, Non-Internet Routable, IP Address Schemes**

<b>Class</b>	<b>Network Address Range</b>
A	10.0.0.0-10.255.255.255
B	172.16.0.0-172.31.255.255
C	192.168.0.0-192.168.255.255

# Indirizzi IP particolari

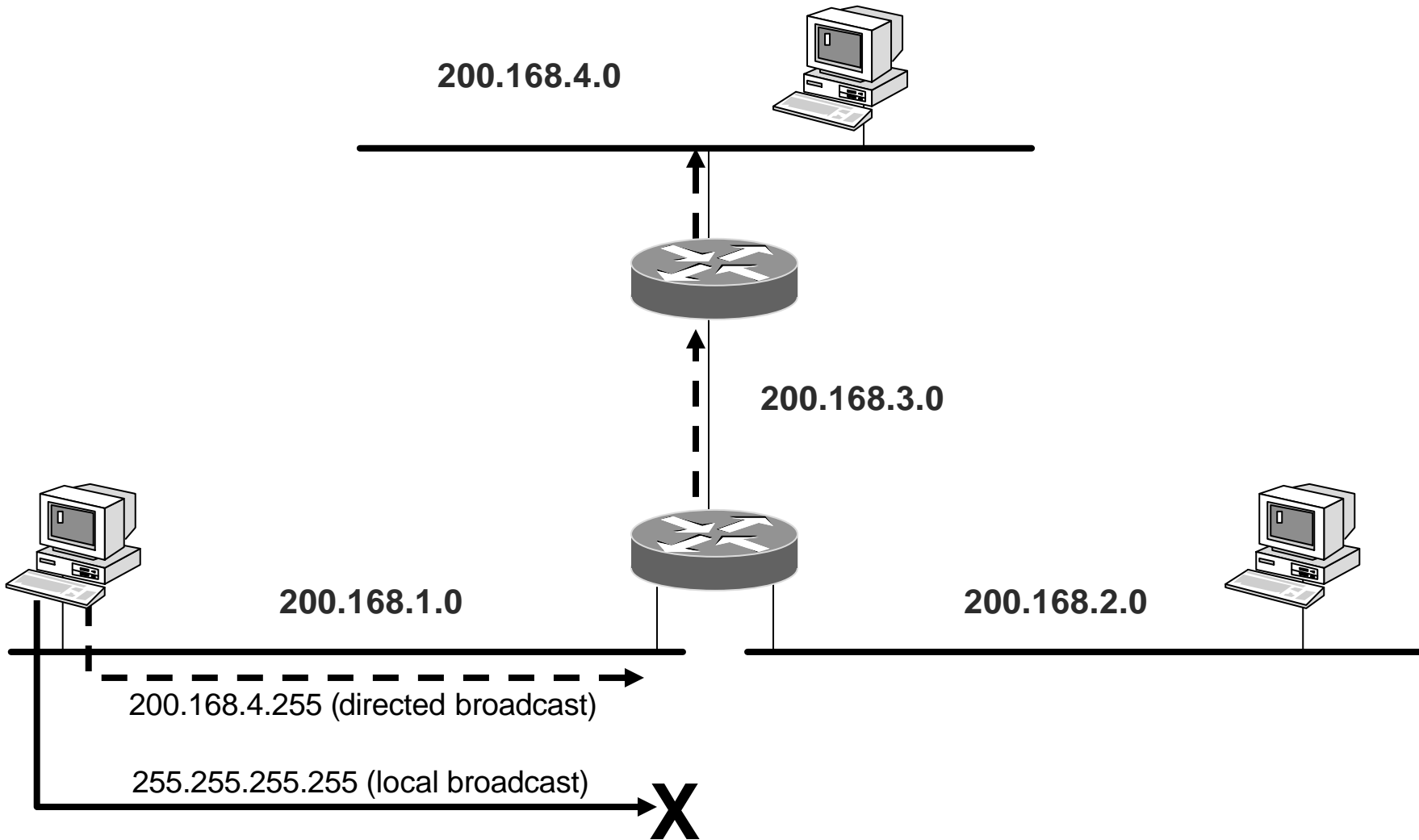
<b>All 0s</b>		<b>This host<sup>1</sup></b>
<b>All 0s</b>	<b>Host</b>	<b>Host on this net<sup>1</sup></b>
<b>All 1s</b>		<b>Limited broadcast (local net)<sup>2</sup></b>
<b>Net</b>	<b>All 1s</b>	<b>Directed broadcast for net<sup>2</sup></b>
<b>127</b>	<b>Anything (often 1)</b>	<b>Loopback<sup>3</sup></b>

<sup>1</sup> Permessso solo al bootstrap ed è usabile solo come indirizzo sorgente

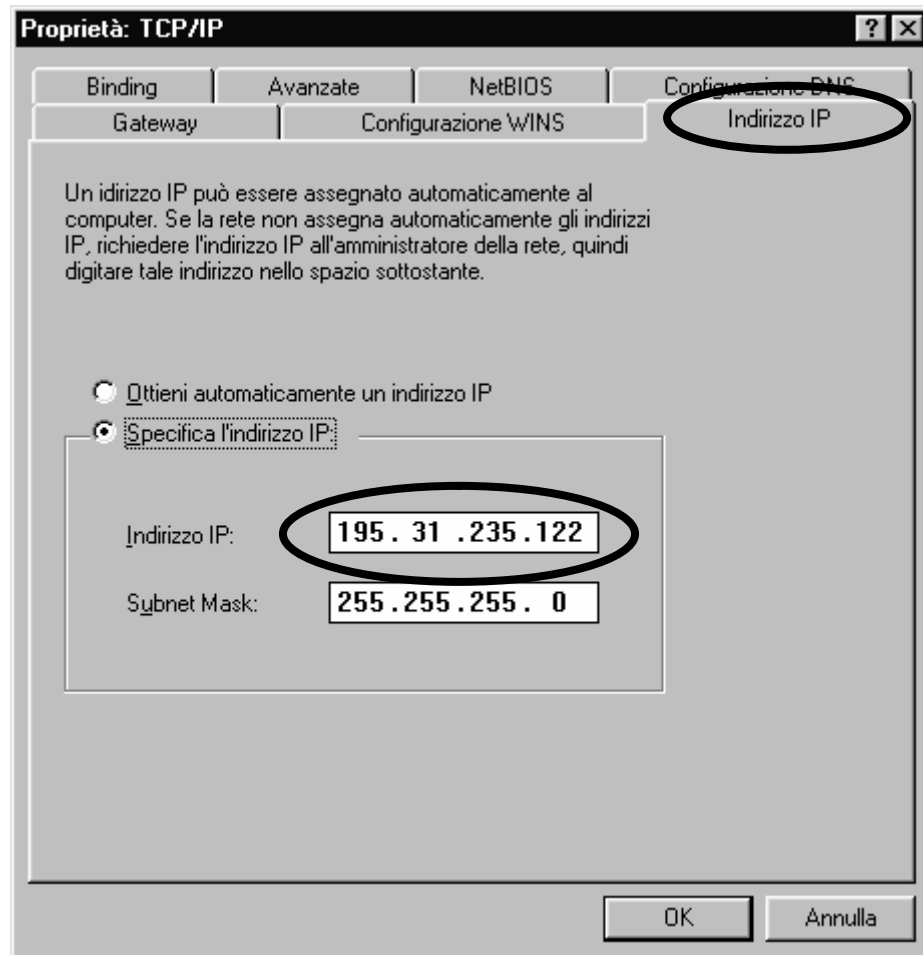
<sup>2</sup> Può essere usato solo come indirizzo destinazione

<sup>3</sup> Non deve essere propagato dai router sulla rete

# Indirizzi di broadcast



# Configurazione Windows 9x: indirizzo IP



# Forwarding di datagrammi IP

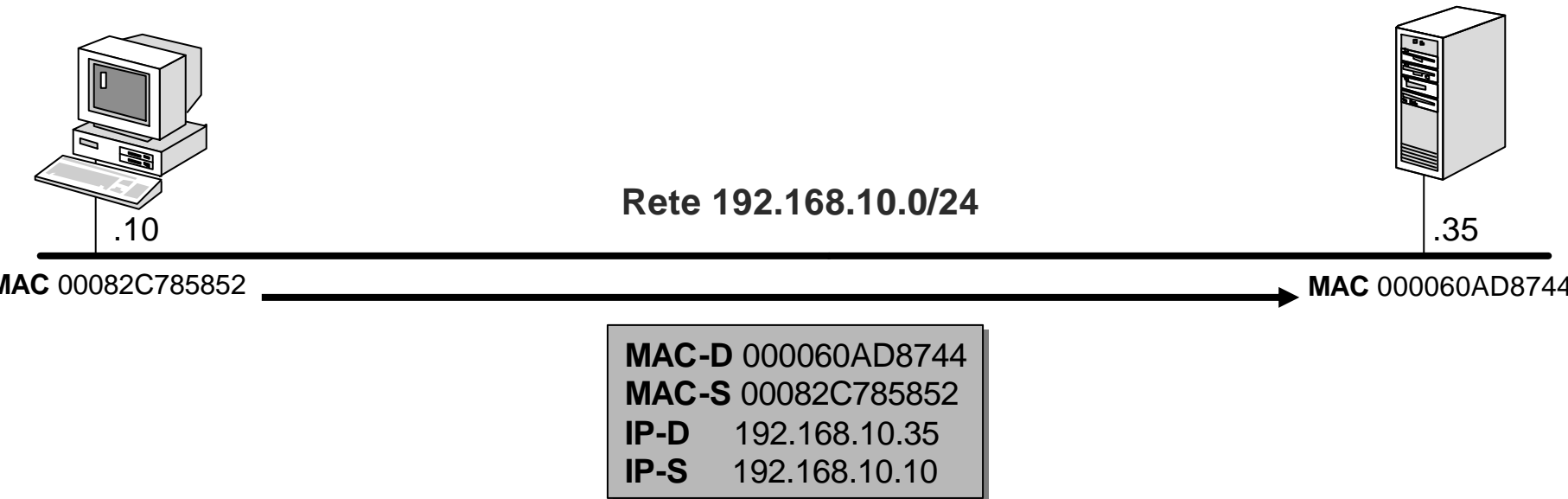
## Forwarding diretto

- ⇓ La trasmissione di un IP datagram tra due host connessi su una stessa rete IP (stesso prefisso) non coinvolge i router
- ⇓ Il trasmettitore incapsula il datagram nel frame fisico e lo invia direttamente all'host destinatario

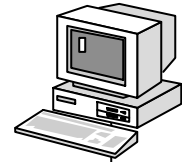
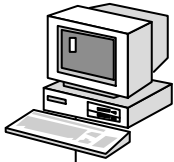
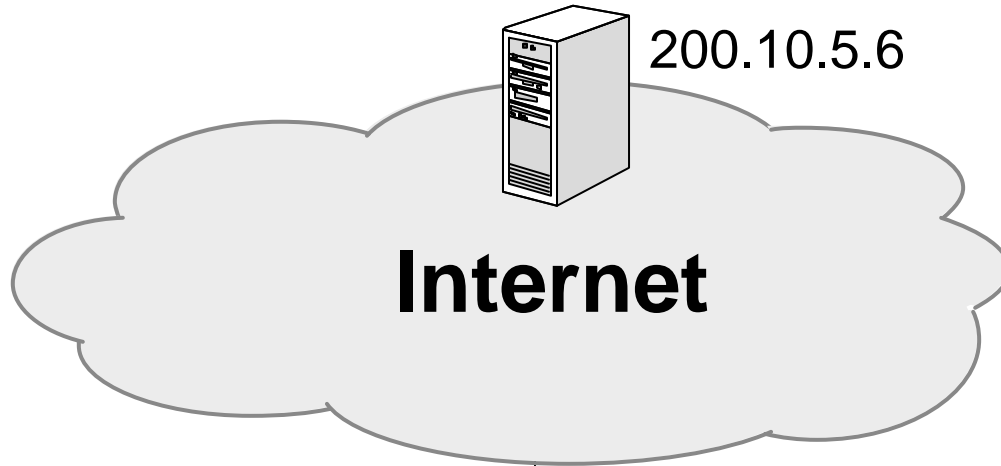
## Forwarding indiretto

- ⇓ La trasmissione di un IP datagram tra due host connessi su differenti reti IP (diverso prefisso) coinvolge i router
- ⇓ Il trasmettitore incapsula il datagram nel frame fisico e lo invia al default gateway
- ⇓ I datagram passano da un router all'altro finchè non raggiungono un router che può trasmetterli direttamente

# Forwarding diretto: esempio



# Forwarding indiretto: esempio



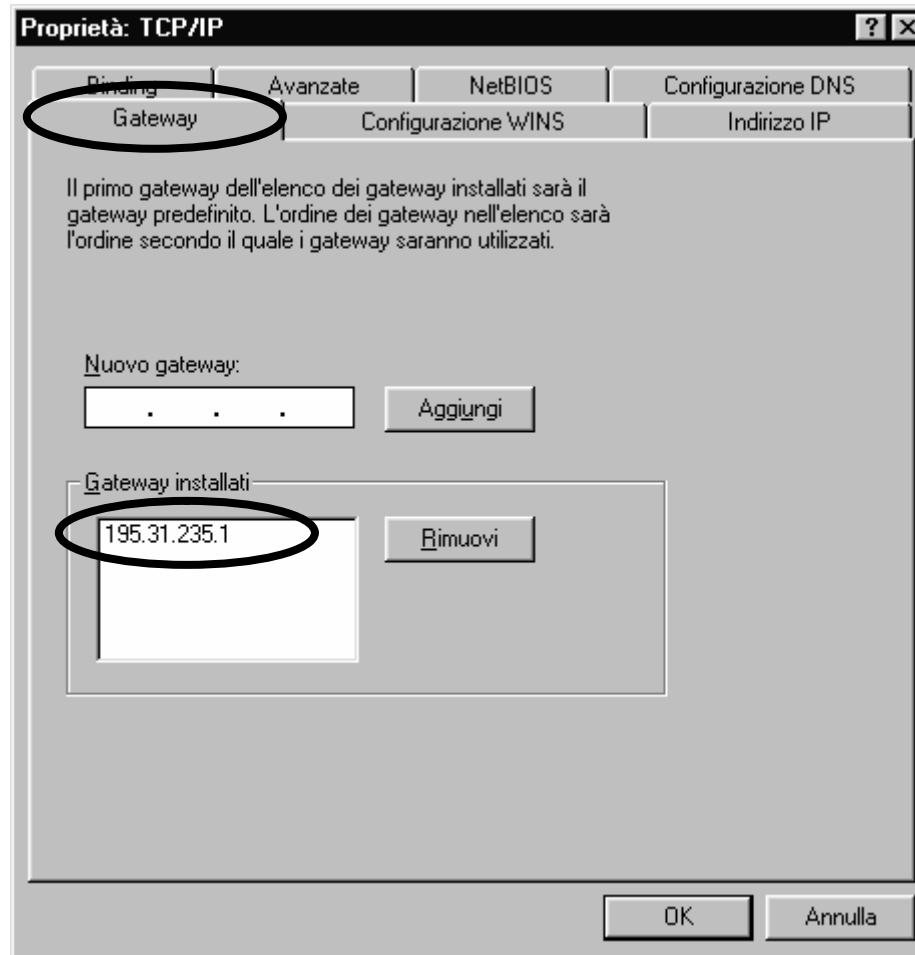
---

IP Add: 172.168.1.10  
Mask: 255.255.255.0  
Default Gateway: 172.168.1.1

**172.168.1.0/24**

IP Add: 172.168.1.120  
Mask: 255.255.255.0  
Default Gateway: 172.168.1.1

# Configurazione Windows 9x: default gateway





# ARP (Address Resolution Protocol)

- ⇓ All'interno di una subnet l'instradamento viene fornito dalla rete fisica
- ⇓ Corrispondenza tra gli indirizzi IP (indirizzi di livello 3) e gli indirizzi di livello 2 gestita da un meccanismo di ARP (Address Resolution Protocol)
- ⇓ Indirizzi di livello 2 possono essere:
  - indirizzi MAC nelle LAN
  - identificatori di circuito virtuale nelle reti X.25, Frame Relay e ATM
  - etc.

# ARP (Address Resolution Protocol)

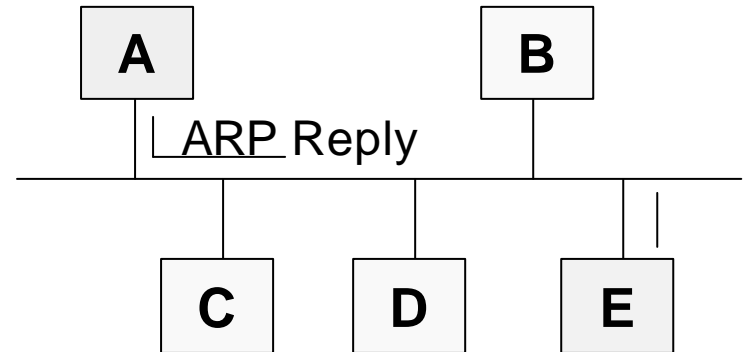
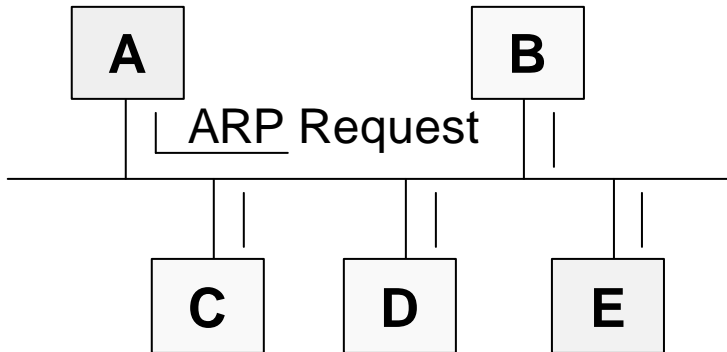
- ⇓ Per effettuare il forwarding diretto è necessario associare l'indirizzo IP del destinatario e indirizzo fisico corrispondente
- ⇓ Mapping statico
  - la tabella di associazione viene predisposta staticamente
- ⇓ Mapping dinamico
  - la tabella viene costruita dinamicamente attraverso un protocollo ARP (Address Resolution Protocol) RFC826

# ARP (Address Resolution Protocol)

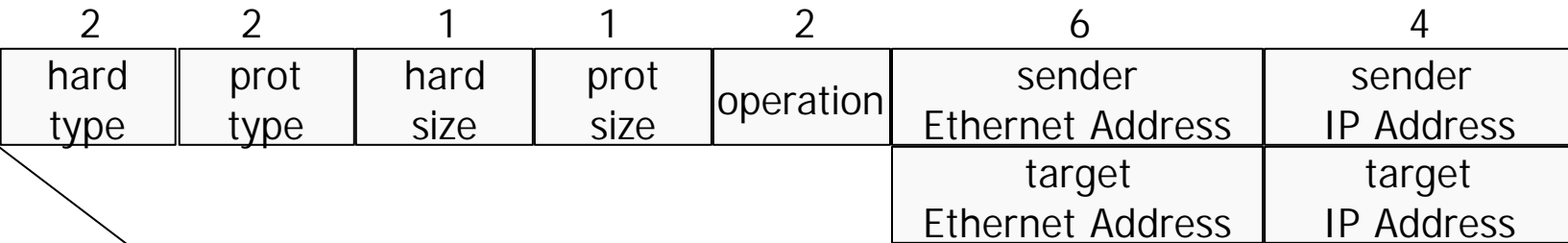
*Campi più significativi della trama MAC*

*Campi più significativi della trama ARP*

<b>MAC broadcast</b>	<b>MAC A</b>	<b>ARP Req</b>	<b>MAC A</b>	<b>IP A</b>	<b>??</b>	<b>IP E</b>
<b>MAC A</b>	<b>MAC E</b>	<b>ARP Reply</b>	<b>MAC E</b>	<b>IP E</b>	<b>MAC A</b>	<b>IP A</b>



# ARP (Address Resolution Protocol)



# ARP Cache

↓ I mapping

**<IP address>**      **<MAC address>**

vengono memorizzati in una cache (ARP Cache)

↓ Quando il driver di rete richiede la spedizione di un pacchetto:

- viene controllato se esiste un mapping per quell'host
- in caso positivo viene generato l'apposita trama MAC
- in caso negativo viene inviata una ARP Request

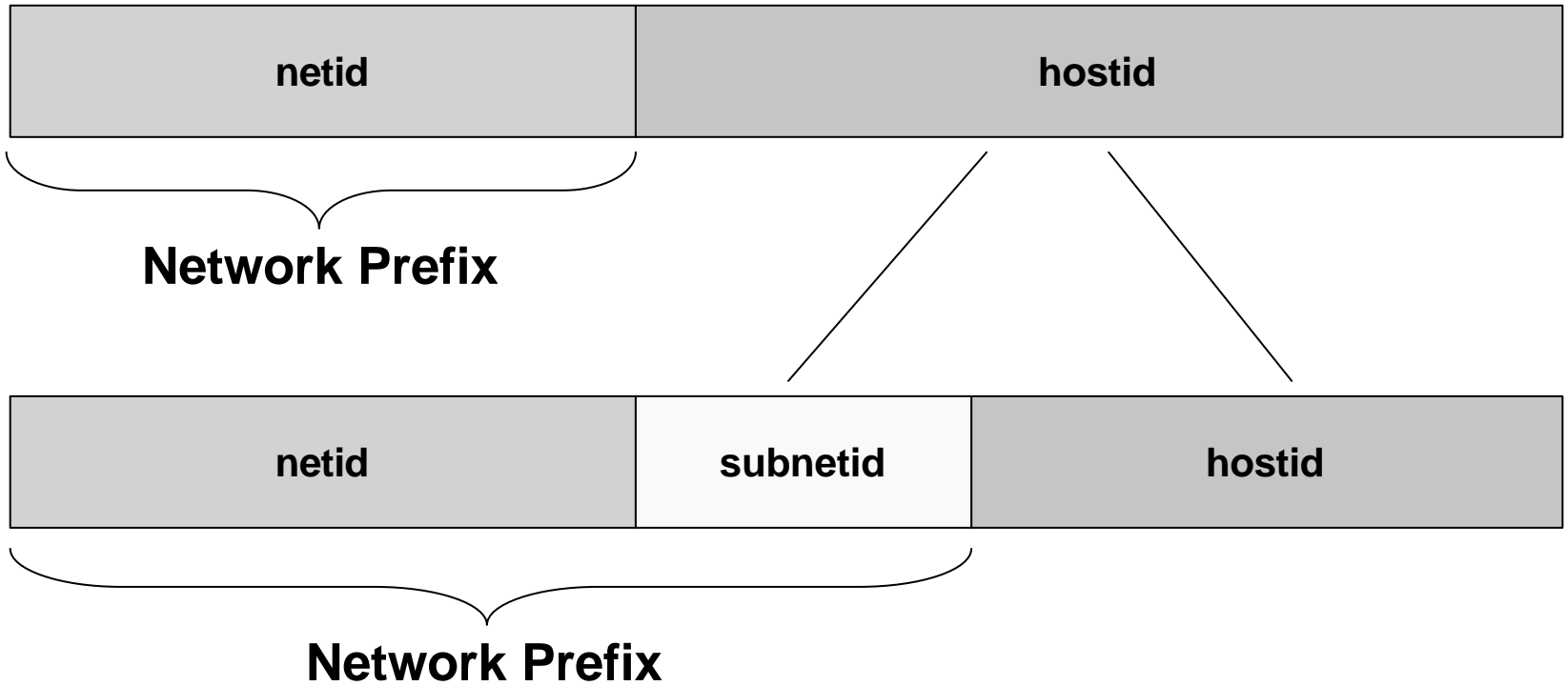
↓ Il comando per visualizzare l'arp-cache è `arp -a`

```
C:\>arp -a
Interface: 195.31.237.213 on Interface 1
  Internet Address      Physical Address      Type
  195.31.237.136        08-00-20-95-1f-d3    dynamic
  195.31.237.138        08-00-20-20-3d-28    dynamic
  195.31.237.140        08-00-20-90-3e-b9    dynamic
  195.31.237.193        00-e0-1e-84-cc-a0    dynamic
```

# Esaurimento degli indirizzi IP

- ⇓ Il progressivo esaurimento degli indirizzi IP unitamente alla rapida crescita delle dimensioni delle tabelle di routing ha spinto l'IETF (*Internet Engineering Task Force*) ad intraprendere delle azioni preventive
- ⇓ Tali misure preventive possono essere raggruppate nelle seguenti categorie:
  - Assegnazione razionale degli indirizzi IP
  - *Classless InterDomain Routing* (CIDR)
  - Indirizzi privati e *Network Address Translation* (NAT)
  - IP versione 6 (IPv6)

# Subnetting



# Classi di indirizzi e netmask

- ⇓ Indirizzo di rete "naturale" è un prefisso con maschera uguale a quella implicita
- ⇓ Subnetting: deriva da una maschera con più bit a 1 rispetto alla maschera naturale

Network																Subnet				Host												
193								205								102				36												
1	1	0	0	0	0	0	1	1	1	0	0	1	1	0	1	1	0	0	1	1	1	0	0	0	1	0	0	1	0	0		
255								255								255				248												
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0

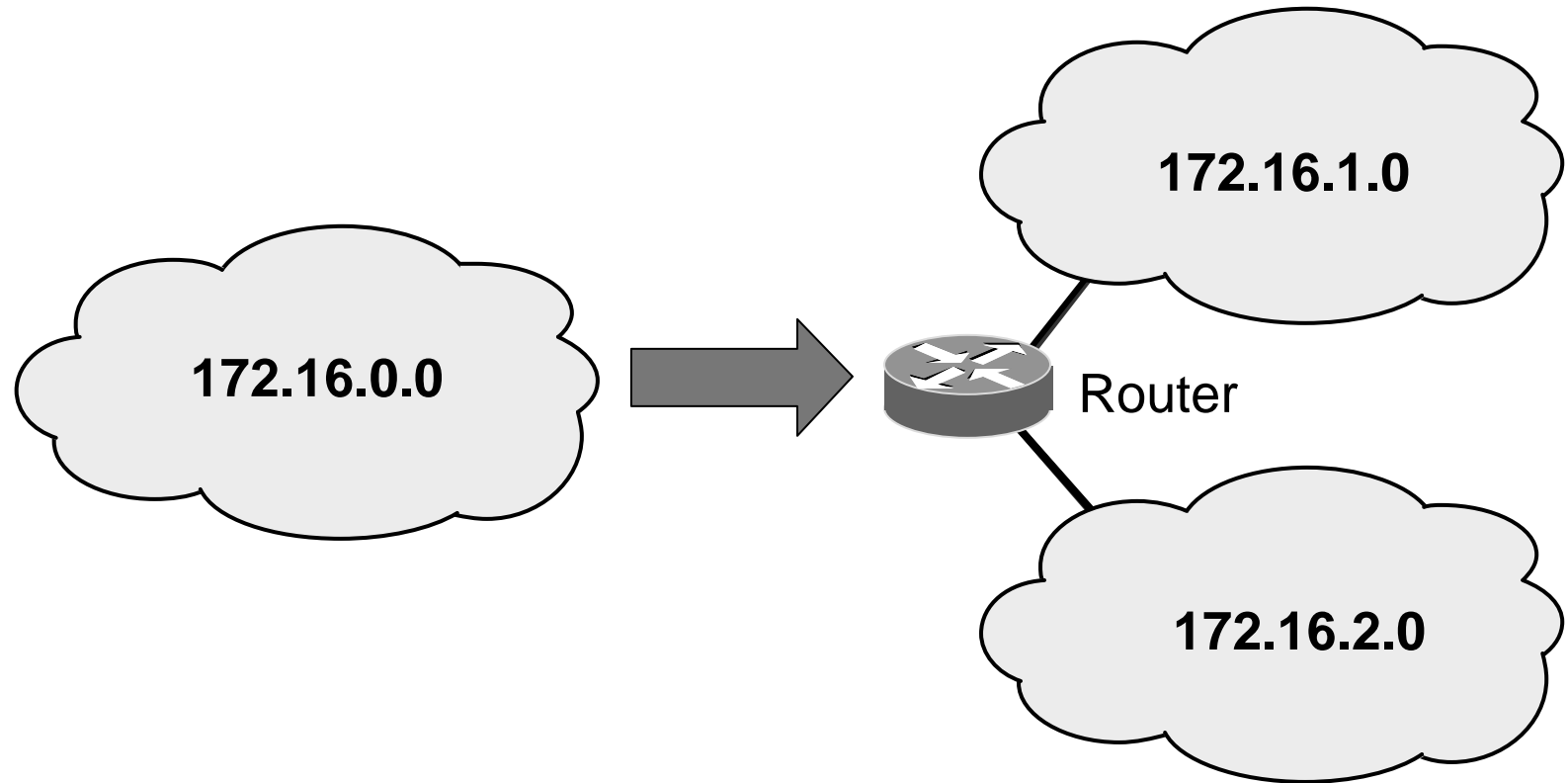


# Netmask: valori decimali leciti

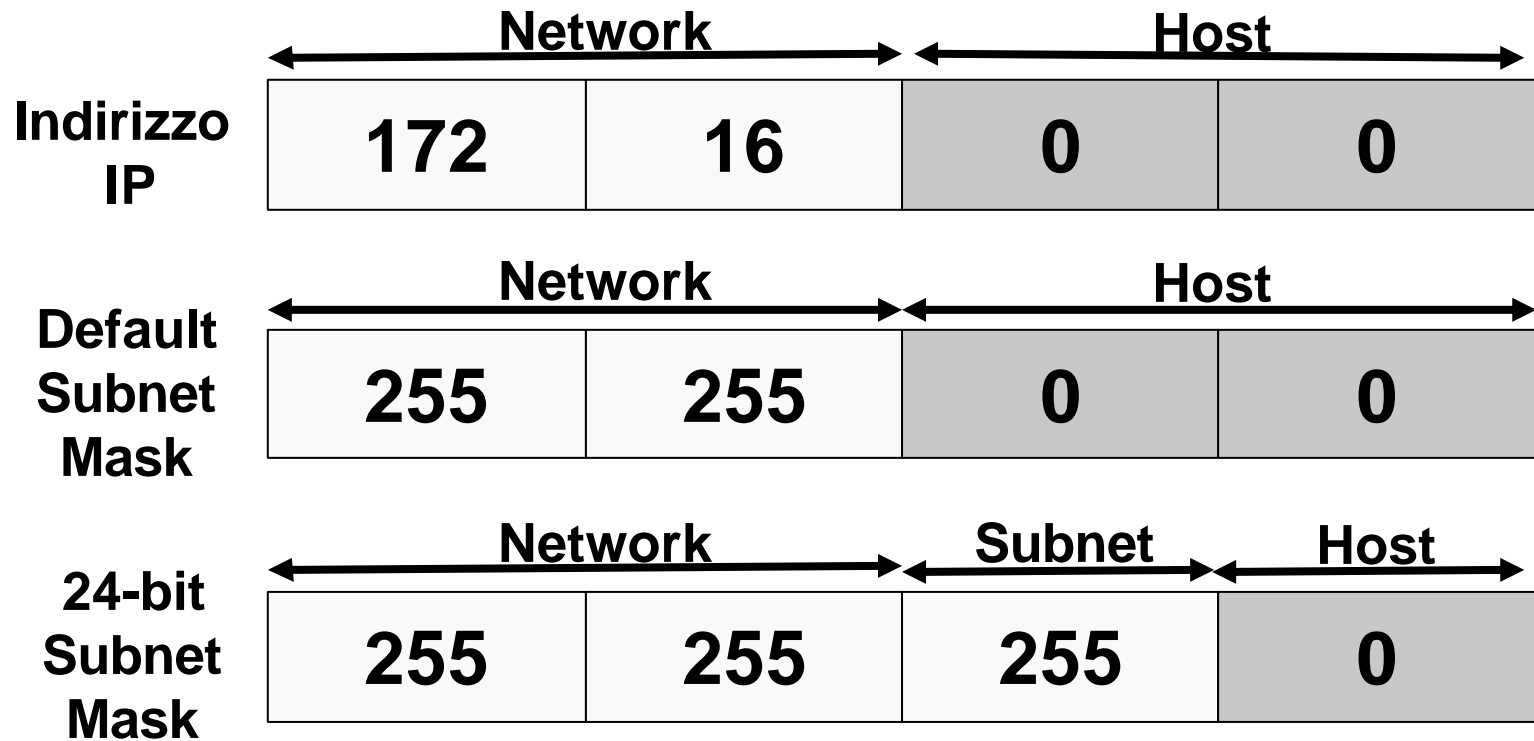
↓ I valori decimali leciti nei 4 byte che costituiscono la netmask sono:

128	1000 0000	(128)
192	1100 0000	(64)
224	1110 0000	(32)
240	1111 0000	(16)
248	1111 1000	(8)
252	1111 1100	(4)
254	1111 1110	(2)
255	1111 1111	(1)

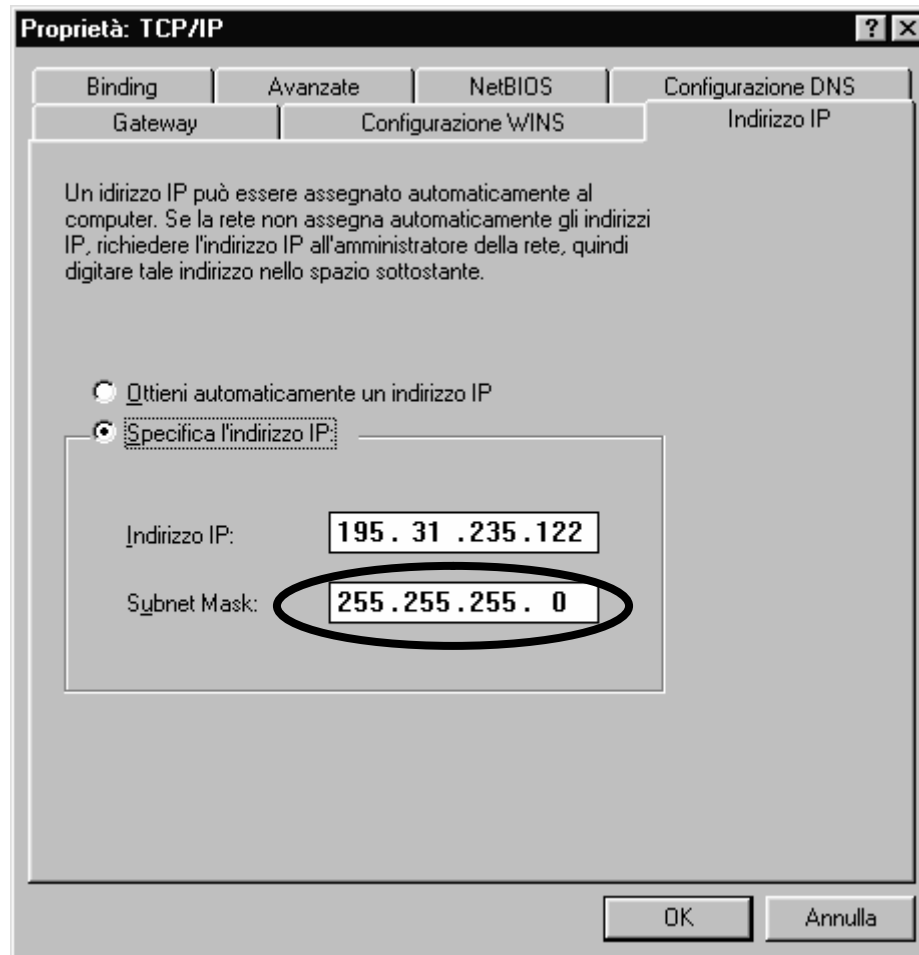
# Esempio: subnetting



# Esempio: Subnet Mask



# Configurazione Windows 9x: subnet mask



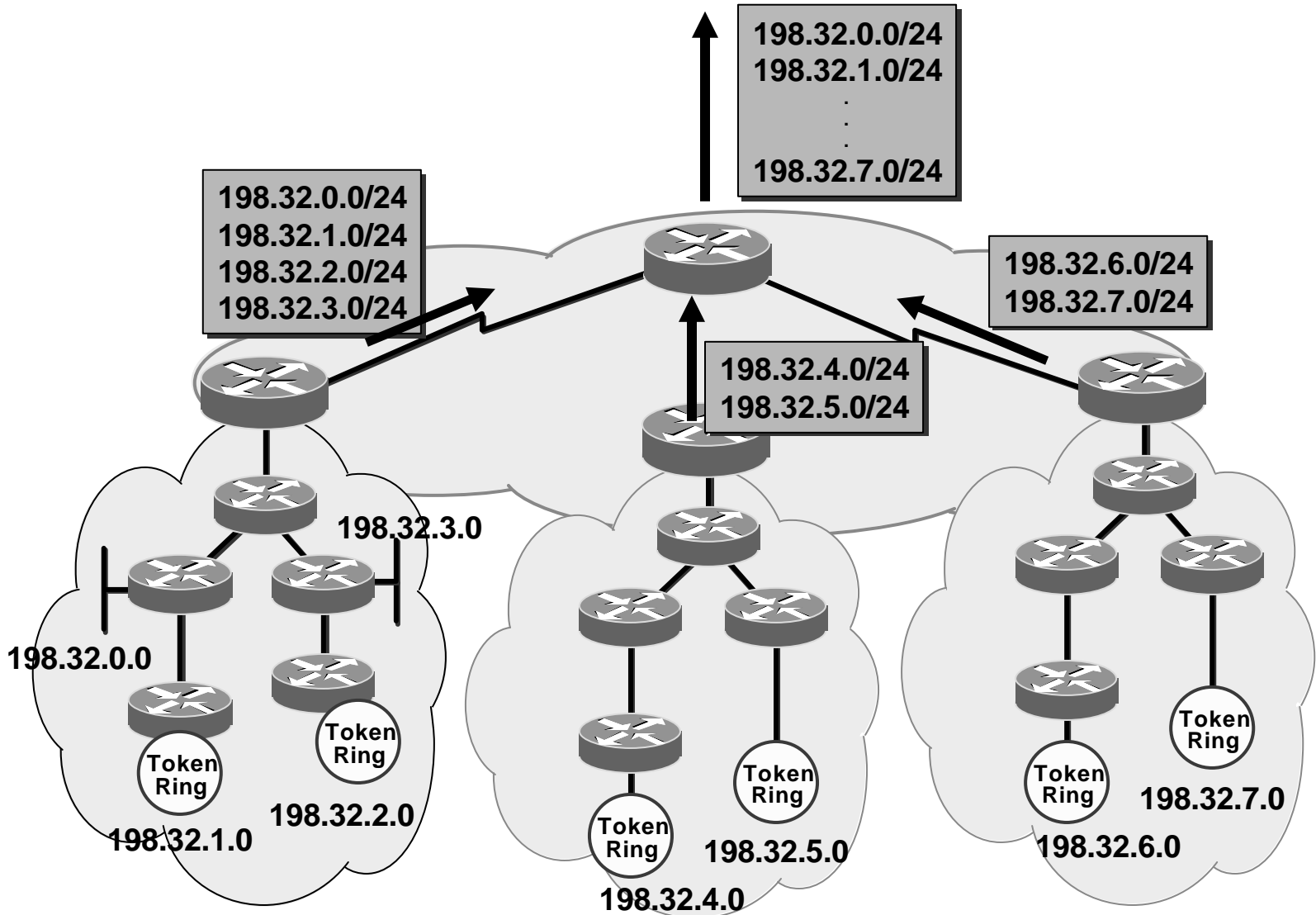
# VLSM

- ⇓ Un grosso limite del subnetting tradizionale è il dover utilizzare una netmask di lunghezza fissa per ogni indirizzo di rete
- ⇓ Una volta che la netmask viene scelta si è vincolati ad avere un numero fisso di sottoreti aventi tutte le stesse dimensioni (in termini di host indirizzabili)
- ⇓ Nel 1987 l'RFC 1009 ha specificato come una rete divisa in sottoreti possa utilizzare più di una netmask
- ⇓ Quando ad una rete viene assegnata più di una netmask, questa viene considerata una rete con maschere di lunghezza variabile (Variable Length Subnet Mask, VLSM)

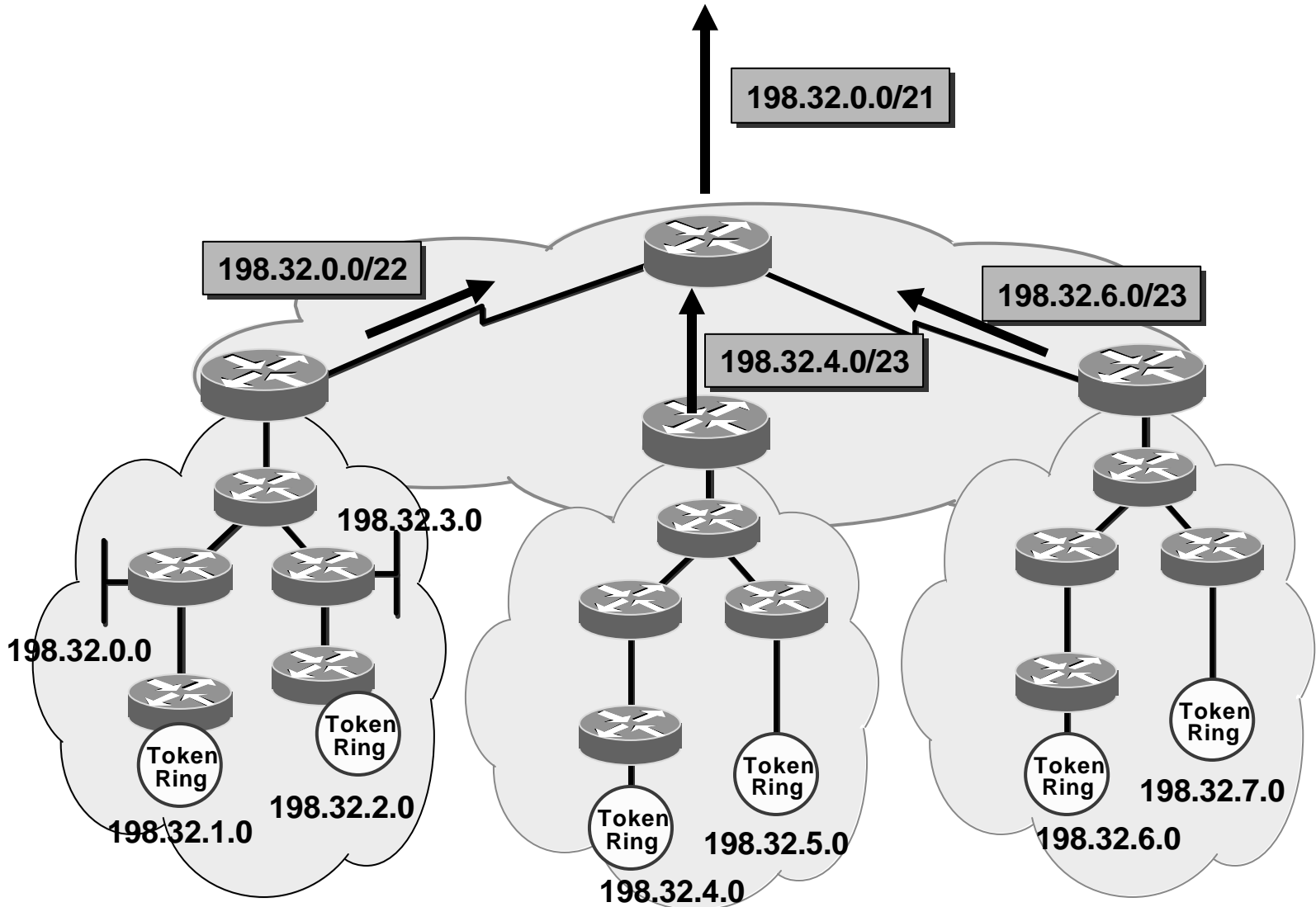
# CIDR

- ↓ Il Classless InterDomain Routing (CIDR) supporta due importanti caratteristiche che hanno portato grossi benefici al sistema di routing di Internet:
- elimina il concetto tradizionale di indirizzi di rete di classe A, classe B e classe C, consentendo un'allocazione efficiente dello spazio degli indirizzi IP
  - supporta l'aggregazione degli indirizzi, consentendo di rappresentare lo spazio di indirizzi di migliaia di reti classful tradizionali in una singola entry nella tabella di routing

# Esempio: senza CIDR



# Esempio: con CIDR



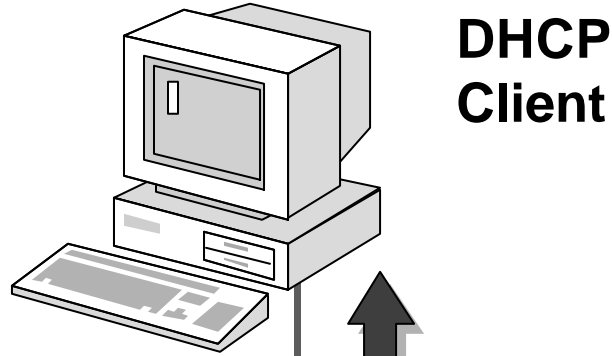


DHCP

# Il protocollo DHCP

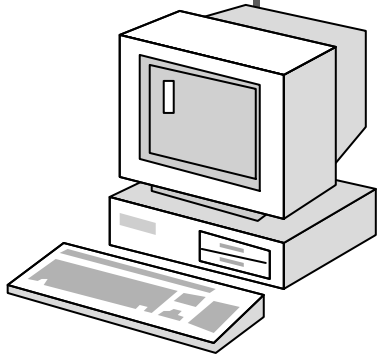
- ⇓ Il protocollo DHCP (Dynamic Host Configuration Protocol) assegna automaticamente indirizzi IP ai computer superando le limitazioni imposte dalla configurazione manuale
- ⇓ Il protocollo DHCP è definito negli RFC 1533, 1534, 1541 e 1542

# DHCP: client e server



**IP Address1**

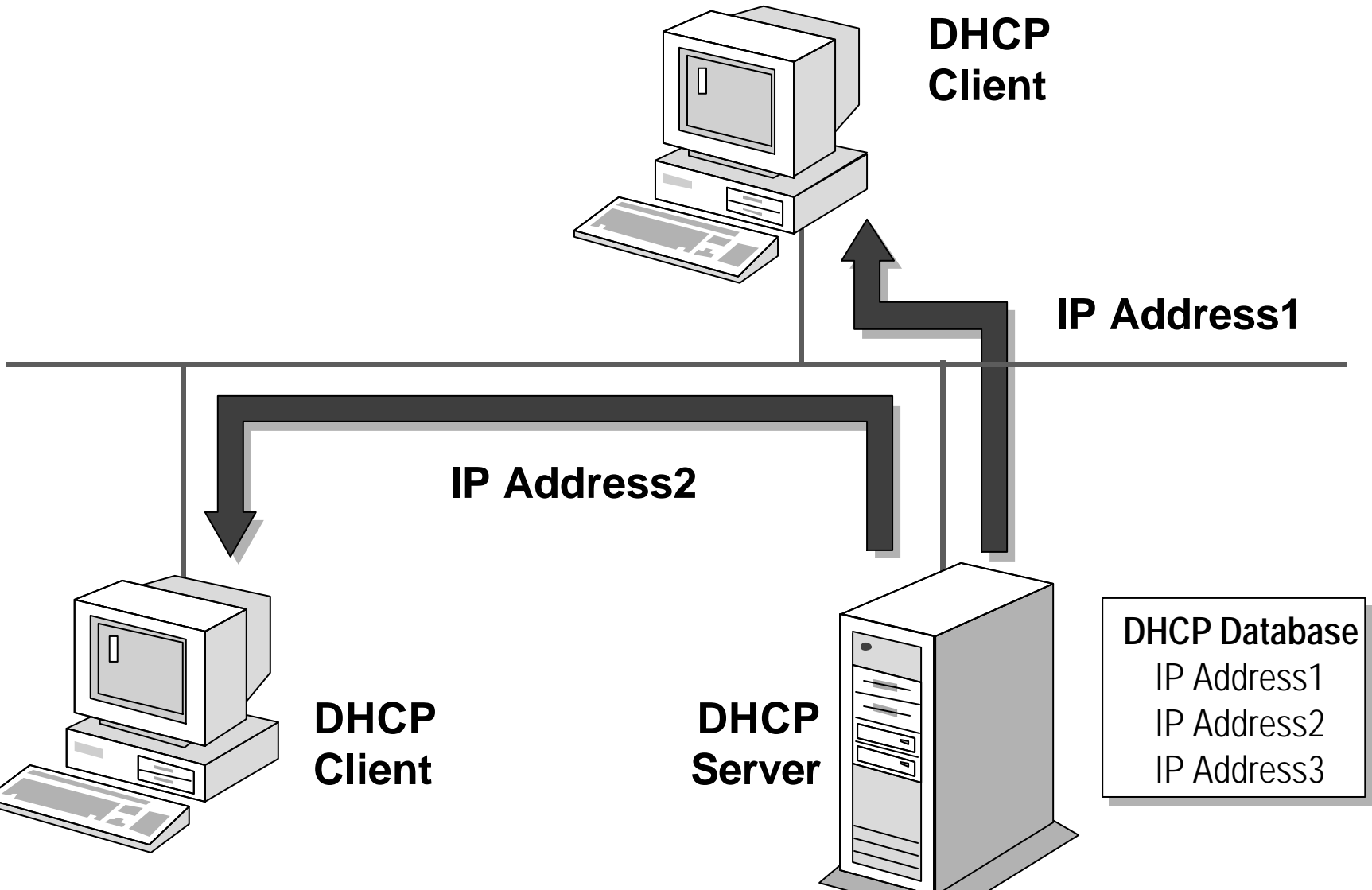
**IP Address2**



**DHCP Server**

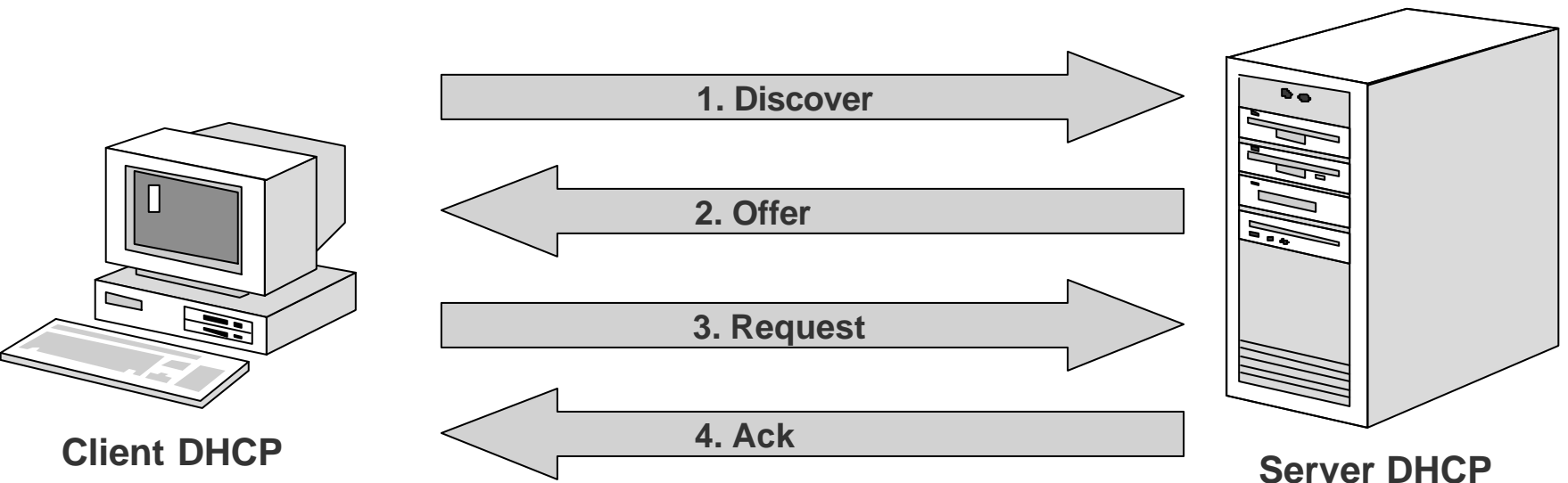
**DHCP Database**

IP Address1  
IP Address2  
IP Address3

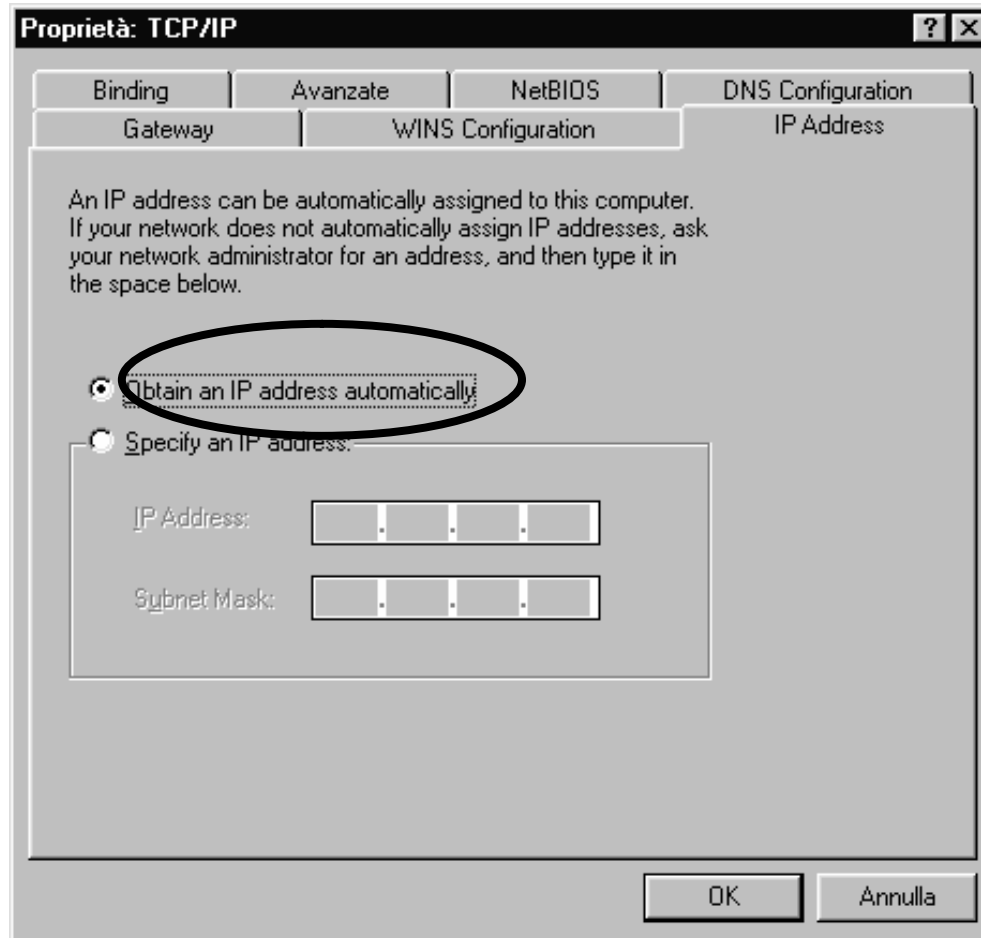


# Il protocollo DHCP

- ↓ DHCP utilizza un processo in quattro fasi per configurare un client DHCP
- ↓ L'intero processo di comunicazione DHCP avviene tramite le porte UDP 67 (server) e 68 (client)



# Configurazione Windows 95/98: client DHCP



# Configurazione Windows NT: server DHCP

**Create Scope - (Local)**

IP Address Pool

Start Address: 131 .107 .2 .1

End Address: 131 .107 .2 .254

Subnet Mask: 255 .255 .255 .0

Excluded Addresses

Exclusion Range:

Start Address: 131 .107 .2 .150 Add ->

End Address: 131 .107 .2 .160 <- Remove

Lease Duration

Unlimited

Limited To: 3 Day(s) 00 Hour(s) 00 Minutes

Name: Subnet 131.107.2.200

Comment: Used by Windows NT Case

OK Cancel Help

# DHCP: richiesta del lease IP

- ⇓ Quando un client viene inizializzato per la prima volta, richiede il lease di un indirizzo IP trasmettendo una richiesta tramite broadcast a tutti i server DHCP
- ⇓ Non avendo un indirizzo IP e non conoscendo l'indirizzo IP di un server DHCP, il client utilizza 0.0.0.0 come indirizzo IP di origine e 255.255.255.255 come indirizzo IP di destinazione
- ⇓ La richiesta di lease viene inviata in un messaggio DHCPDISCOVER, che contiene anche l'indirizzo MAC e il nome del client
- ⇓ Il processo di lease IP viene eseguito quando si verifica una delle seguenti situazioni:
  - lo stack TCP/IP viene inizializzato per la prima volta come client DHCP
  - il client richiede un indirizzo IP specifico che viene rifiutato
  - il client ha rilasciato il lease precedente e ora ne richiede uno nuovo

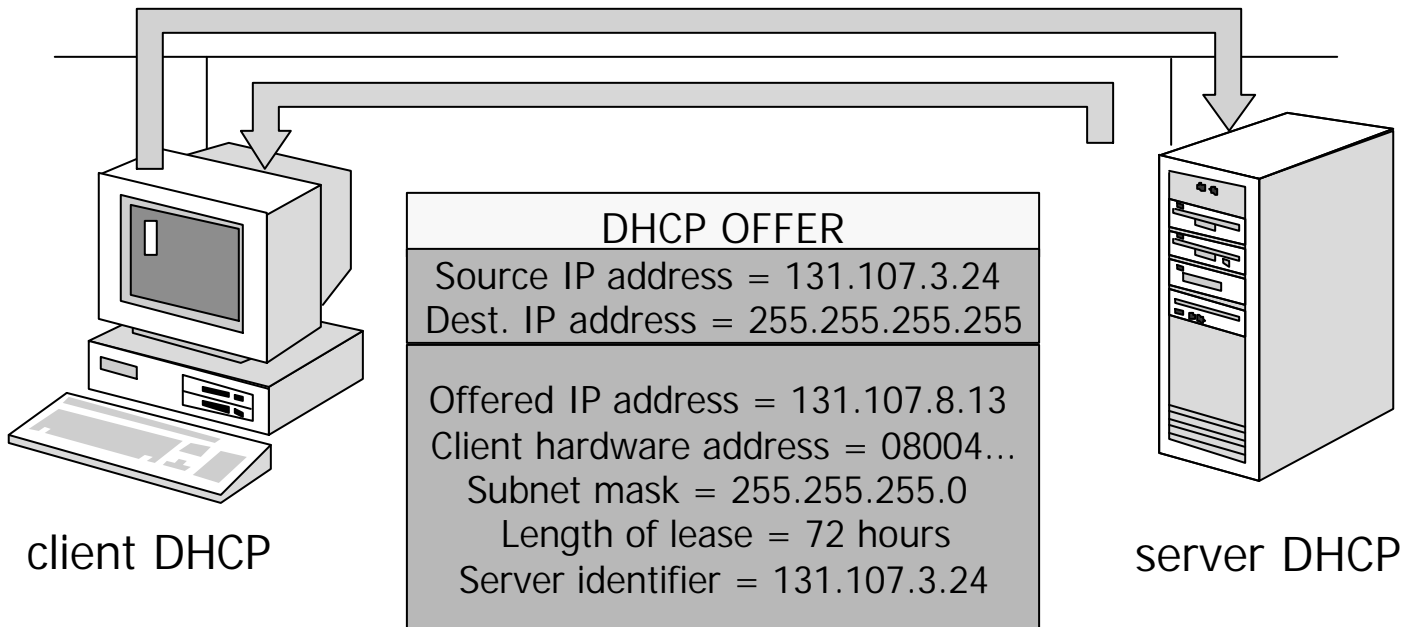
# DHCP: offerta del lease IP

- ⇓ Tutti i server DHCP che ricevono la richiesta e dispongono di una configurazione valida per il client inviano tramite broadcast un'offerta che include le seguenti informazioni:
  - l'indirizzo MAC del client, un'offerta di indirizzo IP, la subnet mask, durata del lease
  - viene inviato inoltre un identificatore del server (indirizzo IP del server che ha inviato l'offerta)
- ⇓ La trasmissione avviene tramite broadcast perchè il client non ha ancora un indirizzo IP. L'offerta viene inviata come messaggio DHCPOFFER
- ⇓ Il client DHCP seleziona l'indirizzo IP dalla prima offerta ricevuta
- ⇓ Il client DHCP attende un'offerta per un secondo. Se non riceve offerte, il client non potrà essere inizializzato e ritrasmetterà la richiesta tramite broadcast per tre volte. Se non riceve alcuna offerta dopo quattro richieste, il client riproverà ogni cinque minuti



# DHCP: Discover e Offer

DHCP DISCOVER
Source IP Address = 0.0.0.0 Dest. IP address = 255.255.255.255
Hardware Address = 08004....



# DHCP: selezione del lease IP

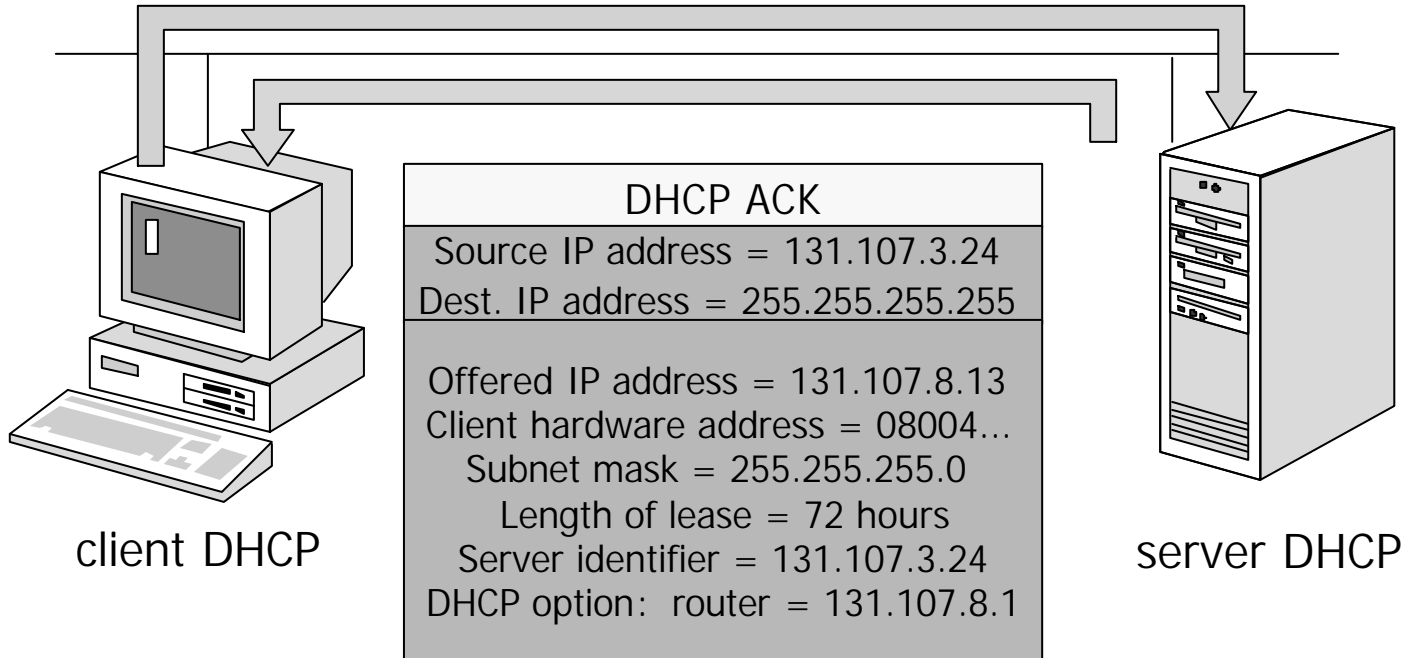
- ⇩ Dopo aver ricevuto un'offerta da almeno un server DHCP, il client comunica tramite broadcast a tutti i server DHCP che ha eseguito una selezione accettando l'offerta
- ⇩ La selezione del lease viene inviata come messaggio DHCPREQUEST e include l'indirizzo IP del server di cui è stata accettata l'offerta
- ⇩ A questo punto, tutti gli altri server DHCP ritirano le rispettive offerte in modo che gli indirizzi IP corrispondenti siano disponibili per la successiva richiesta di lease IP

# DHCP: riconoscimento del lease IP

- ⇓ Il server DHCP di cui è stata accettata l'offerta invia al client tramite broadcast un riconoscimento di operazione riuscita sotto forma di messaggio DHCPACK
- ⇓ Se il client cerca di ottenere il lease dell'indirizzo IP precedente e questo non è più disponibile, viene inviato tramite broadcast un riconoscimento di operazione non riuscita mediante il messaggio DHCPNACK
- ⇓ Un messaggio DHCPNACK viene inviato anche se l'indirizzo IP richiesto non è più valido perchè il client è stato spostato in una diversa sottorete

# DHCP: Request e Ack

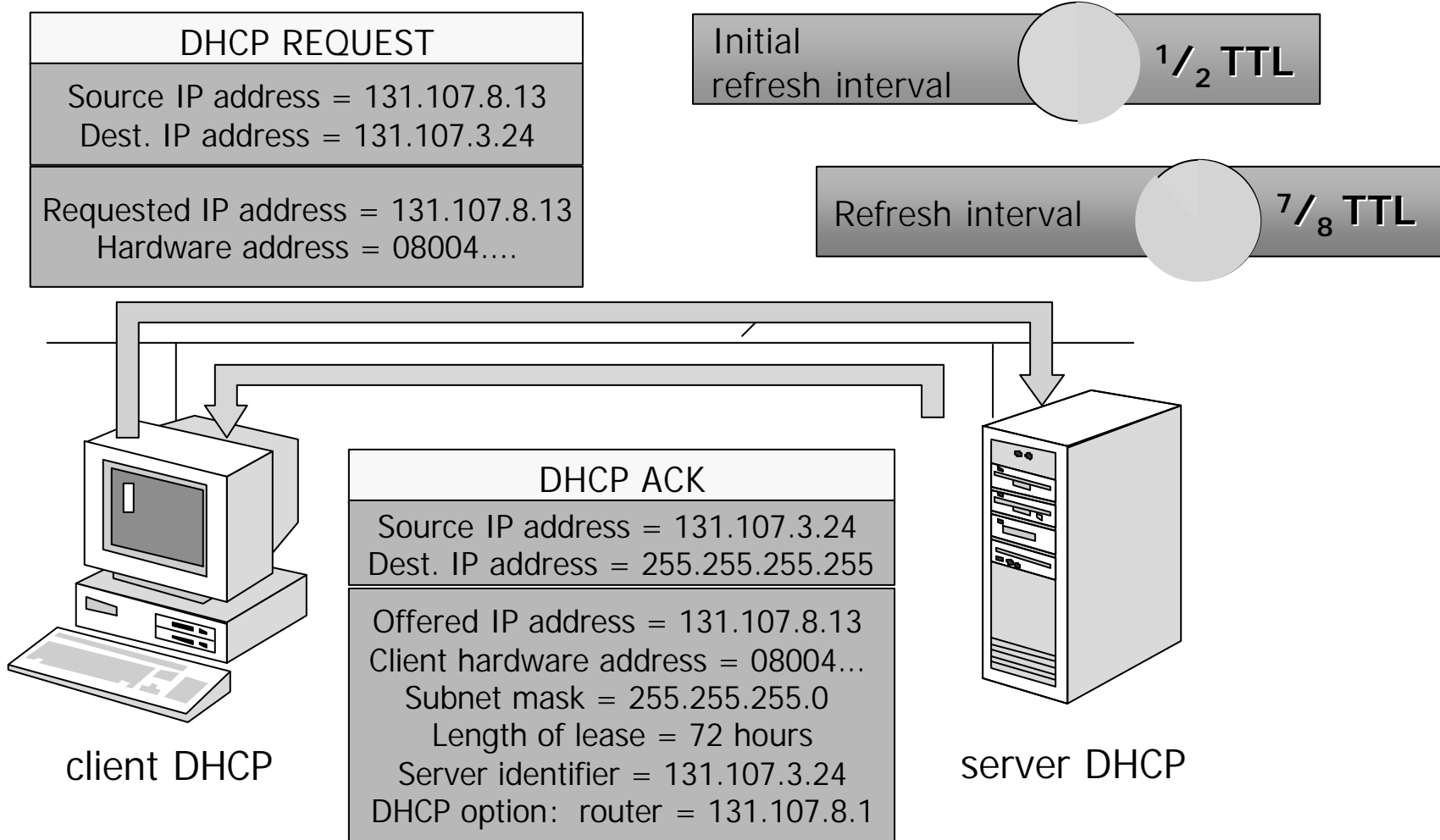
DHCP REQUEST
Source IP address = 0.0.0.0 Dest. IP address = 255.255.255.255
Hardware address = 08004... Requested IP address = 131.107.8.13 Server identifier = 131.107.3.24



# DHCP: rinnovo di un lease IP

- ↓ Tutti i client DHCP provano a rinnovare il proprio lease una volta trascorso il 50 per cento del periodo di lease
- ↓ Per rinnovare il proprio lease, un client DHCP invia un messaggio DHCPREQUEST direttamente al server DHCP da cui ha ottenuto il lease
- ↓ Se il server DHCP è disponibile, rinnoverà il lease e quindi invierà al client un messaggio DHCPACK con il nuovo periodo di lease ed eventuali aggiornamenti dei parametri di configurazione
- ↓ Se il client non riesce a rinnovare il lease allo scadere del 50 per cento della sua durata, il client cerca di contattare qualsiasi server DHCP disponibile una volta trascorso l'87,5 per cento del periodo di lease inviando un messaggio DHCPREQUEST tramite broadcast

# DHCP: rinnovo di un lease IP



# DHCP: sequenza di reboot

- ⇩ Quando un client DHCP effettua il reboot, se ha un indirizzo IP già precedentemente assegnatogli, invia un messaggio DHCPREQUEST tramite broadcast per verificare che il suo indirizzo IP sia ancora valido
- ⇩ Il messaggio DHCPREQUEST è inviato tramite broadcast in modo che se il client viene spostato in una subnet differente (durante il periodo in cui è spento) possa ricevere un indirizzo IP valido in tempi brevi
- ⇩ Il server DHCP locale ricevendo un richiesta di rinnovo per un indirizzo IP non assegnato da lui, risponderà con un messaggio DHCPNACK. Il client inizierà quindi il processo di leasing in modo da ottenere un indirizzo IP valido sulla nuova subnet

# DHCP: rilascio di un lease IP

- ⇓ I client DHCP non rilasciano l'indirizzo IP quando vengono spenti
- ⇓ E' possibile forzare un client a rilasciare il proprio indirizzo IP (il client invia un messaggio DHCPRELEASE al server DHCP per rinunciare al lease)
- ⇓ Ciò risulta utile quando il client deve essere rimosso dalla rete oppure spostato in un'altra subnet



# DHCP in presenza di router

- ⇓ Il caso in cui client e server sono nella stessa subnet è quello più semplice per le operazioni del protocollo DHCP
- ⇓ DHCP diventa più complesso in presenza di router. Il problema nasce dal fatto che molti messaggi DHCP sono trasmessi in broadcast ed i router non inoltrano i broadcast
- ⇓ Sarebbe necessario un server DHCP per ogni subnet
- ⇓ Per evitare ciò, è stata adottata una soluzione (RFC 1542) che implica la possibilità per un router di inoltrare i messaggi DHCP broadcast

# DHCP Relay Agent

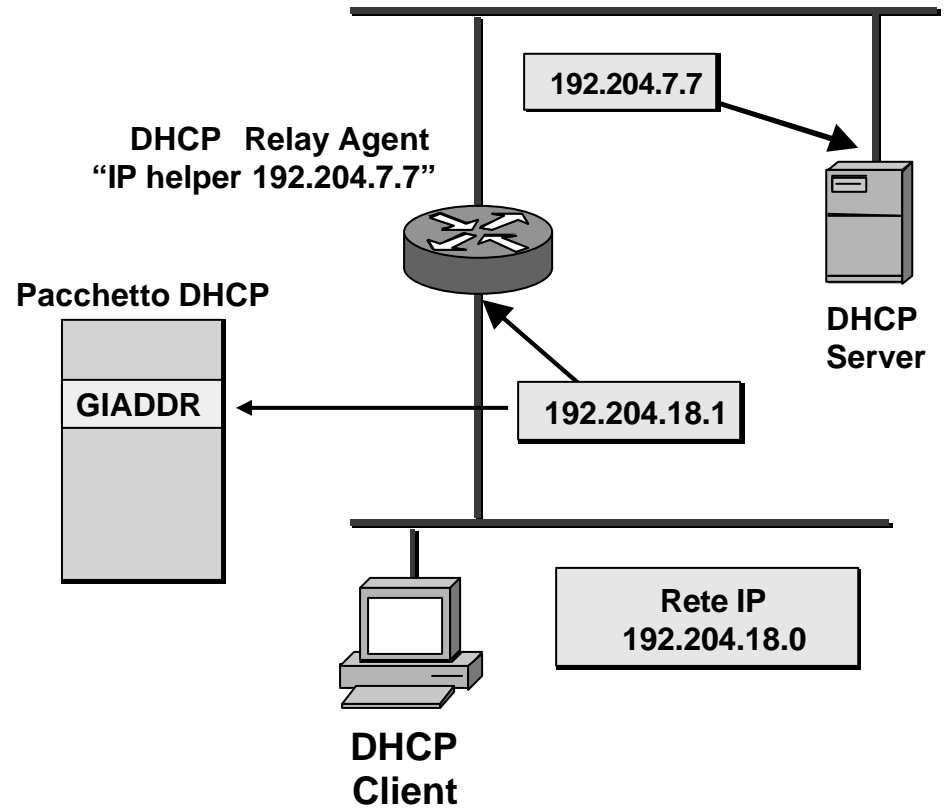
- ⇓ Se non è presente un server DHCP in ogni subnet, i router della rete devono essere configurati per lasciar passare i messaggi DHCP broadcast
- ⇓ Un router può essere configurato come DHCP Relay Agent
  - i messaggi DHCP utilizzano le porte UDP 67 e 68, quindi il router lascerà passare i messaggi inviati come broadcast IP su queste porte
- ⇓ Un DHCP Relay Agent deve essere configurato con l'indirizzo IP del server DHCP al quale inviare i messaggi DHCP (broadcast) provenienti dal client

# DHCP Relay Agent

- ⇩ Un DHCP Relay Agent verifica due proprietà quando riceve un pacchetto IP:
  - indirizzo IP di destinazione 255.255.255.255 e porta UDP di destinazione 67 (messaggio dal client al server)
  - indirizzo IP di destinazione di una delle sue interfacce e porta UDP di destinazione 68 (messaggio dal server al client)
- ⇩ Un DHCP Relay Agent modifica il messaggio DHCP nei seguenti modi:
  - inserisce nel messaggio l'indirizzo IP dell'interfaccia dalla quale ha ricevuto il pacchetto DHCP
  - invia il messaggio in unicast al server DHCP

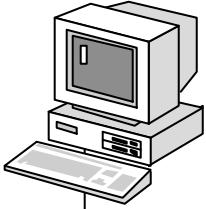
# DHCP Relay Agent: esempio

- ⇩ Il client DHCP invia in broadcast un pacchetto DHCP di Discover
- ⇩ Il router configurato come Relay Agent invia (in unicast) il pacchetto al DHCP server
- ⇩ Il Relay Agent inserisce nel campo GIADDR l'indirizzo IP dell'interfaccia del router dalla quale ha ricevuto il pacchetto
- ⇩ Il Relay agent può essere configurato con indirizzi di più server DHCP
- ⇩ Il server DHCP utilizza il campo GIADDR del pacchetto di Discover per determinare il pool di indirizzi dal quale prelevare l'indirizzo da assegnare al client



# DHCP Relay Agent

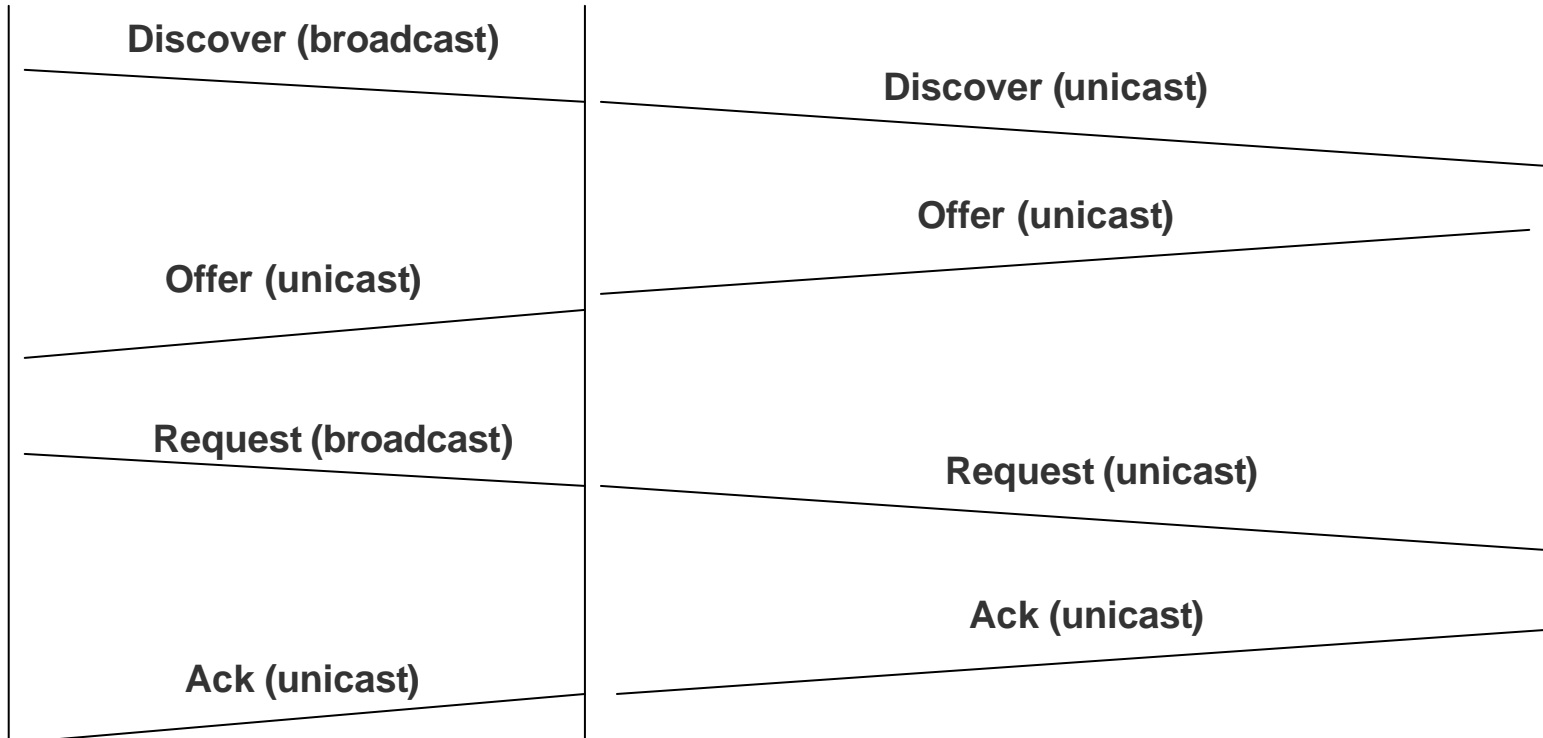
client DHCP



DHCP  
Relay Agent



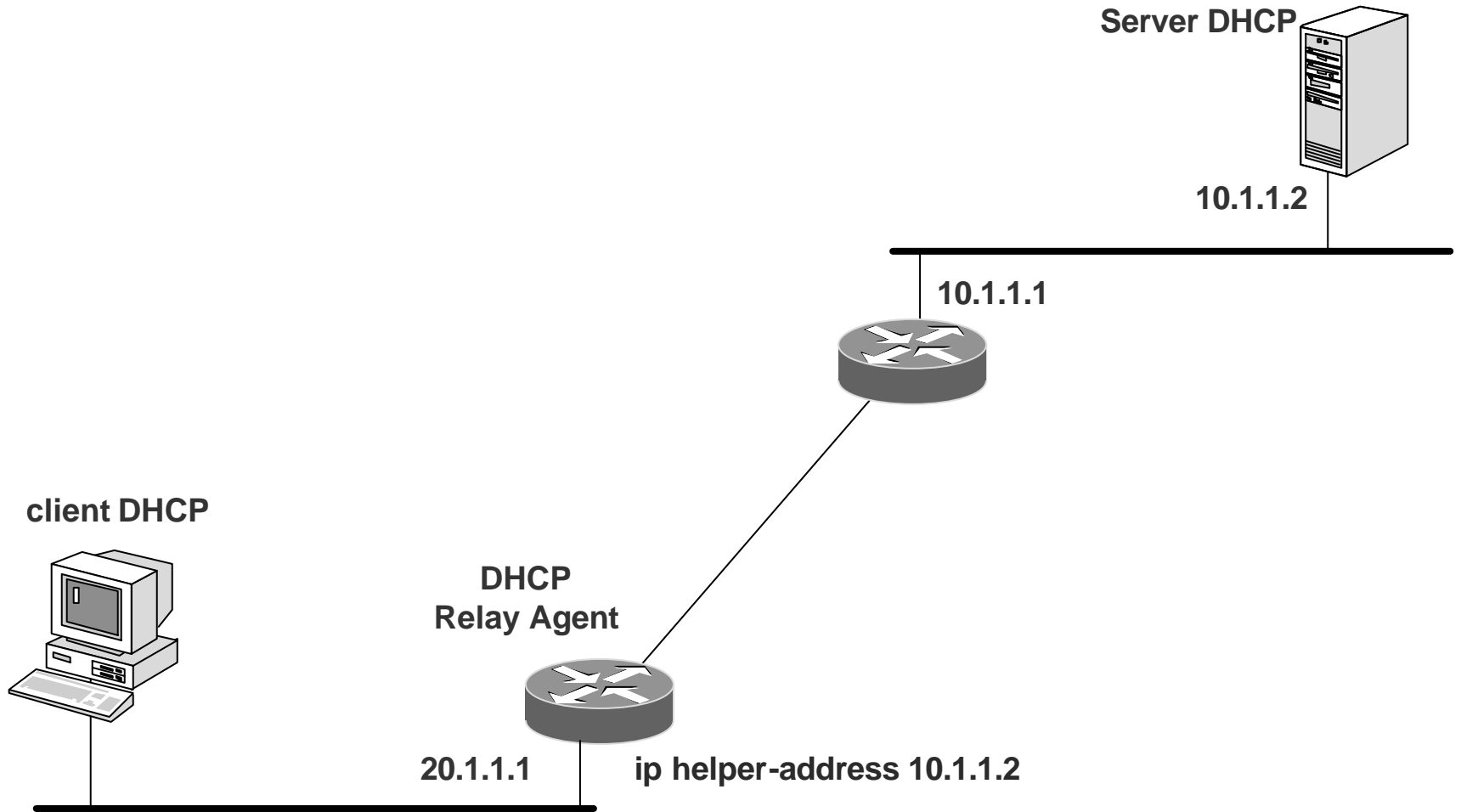
Server DHCP



# Router Cisco come DHCP relay agent

- ⇩ Per utilizzare un router Cisco come DHCP relay agent, deve essere configurato il comando *ip helper-address* sull'interfaccia del router appartenente alla stessa subnet del client DHCP
- ⇩ L'indirizzo usato nel comando *ip helper-address* può essere l'indirizzo specifico di un server DHCP oppure l'indirizzo di una subnet, nel caso in cui siano presenti più server DHCP nello stesso segmento di rete

# Router Cisco come DHCP relay agent



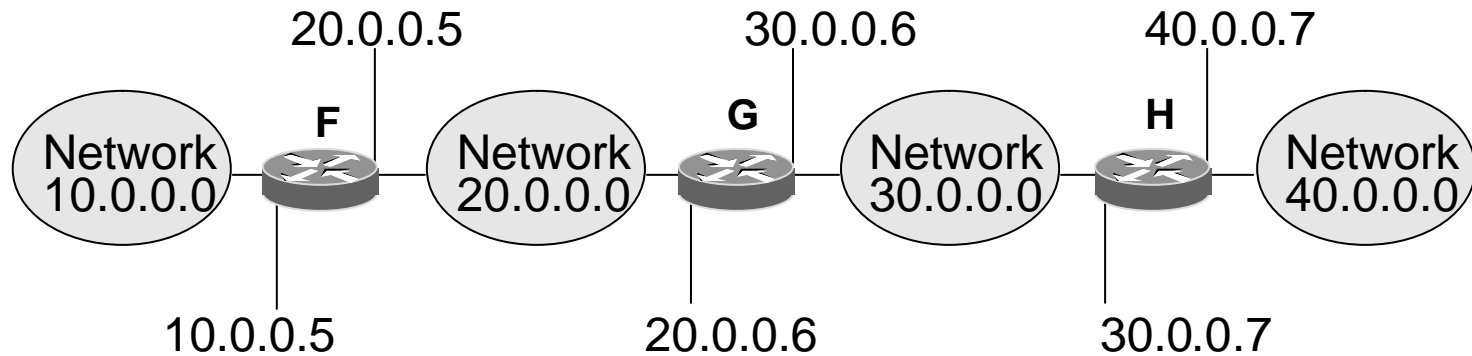
Routing IP



# Routing IP: la tabella di routing

↓ Per effettuare l'instradamento di un pacchetto IP un router dispone di una tabella (routing table) che per ogni riga riporta:

- Network di destinazione
- Router adiacente da attraversare (Next Hop)



Network ID	Next Hop
30.0.0.0	Direct
20.0.0.0	Direct
10.0.0.0	20.0.0.5
40.0.0.0	30.0.0.7

**Tabella di routing di G**

# Algoritmi di routing

↓ Le tabelle possono essere costruite con algoritmi di tipo:

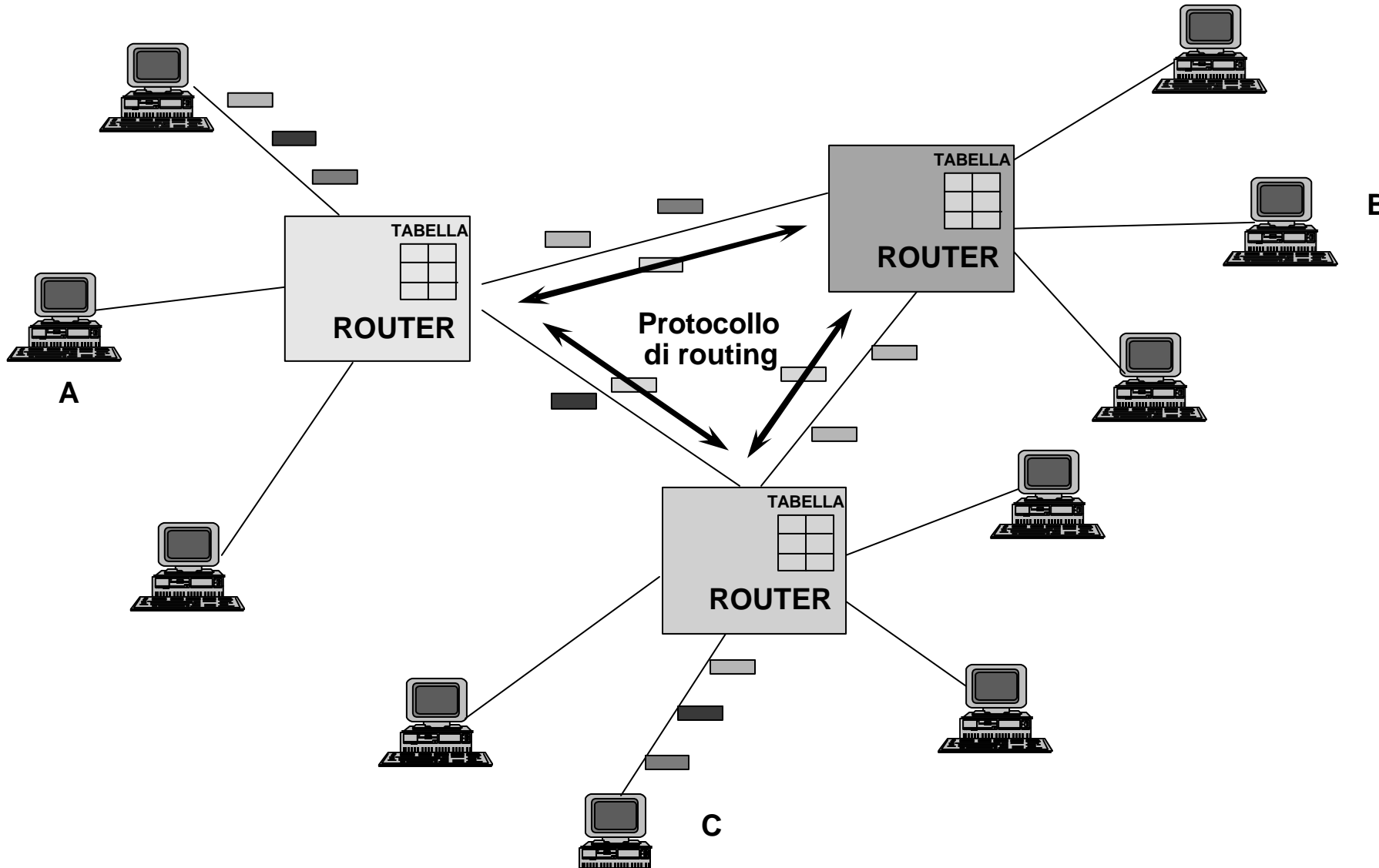
- Statico

- ✧ tabelle definite dal gestore
- ✧ il gestore ha un totale controllo dei flussi di traffico
- ✧ deve intervenire manualmente per riconfigurare la rete
- ✧ utilizzato ad es. nella parte non magliata di reti TCP/IP

- Dinamico

- ✧ tabelle calcolate con appositi algoritmi di routing ed aggiornate automaticamente ad ogni modifica della configurazione della rete

# Routing Dinamico



# Routing Dinamico

- ⇩ Ogni router calcola le proprie tabelle dialogando con gli altri router
- ⇩ Tale dialogo avviene tramite dei protocolli ausiliari a livello 3 o superiore detti protocolli di routing
- ⇩ Esistono due approcci principali al routing distribuito:
  - Algoritmi Distance Vector
    - ⊗ più semplici
    - ⊗ impegnano meno risorse sul router
    - ⊗ meno efficienti
    - ⊗ adatti a reti piccole
  - Algoritmi Link State
    - ⊗ molto più complessi
    - ⊗ molto più efficienti
    - ⊗ impegnano più risorse
    - ⊗ adatti a reti grandi

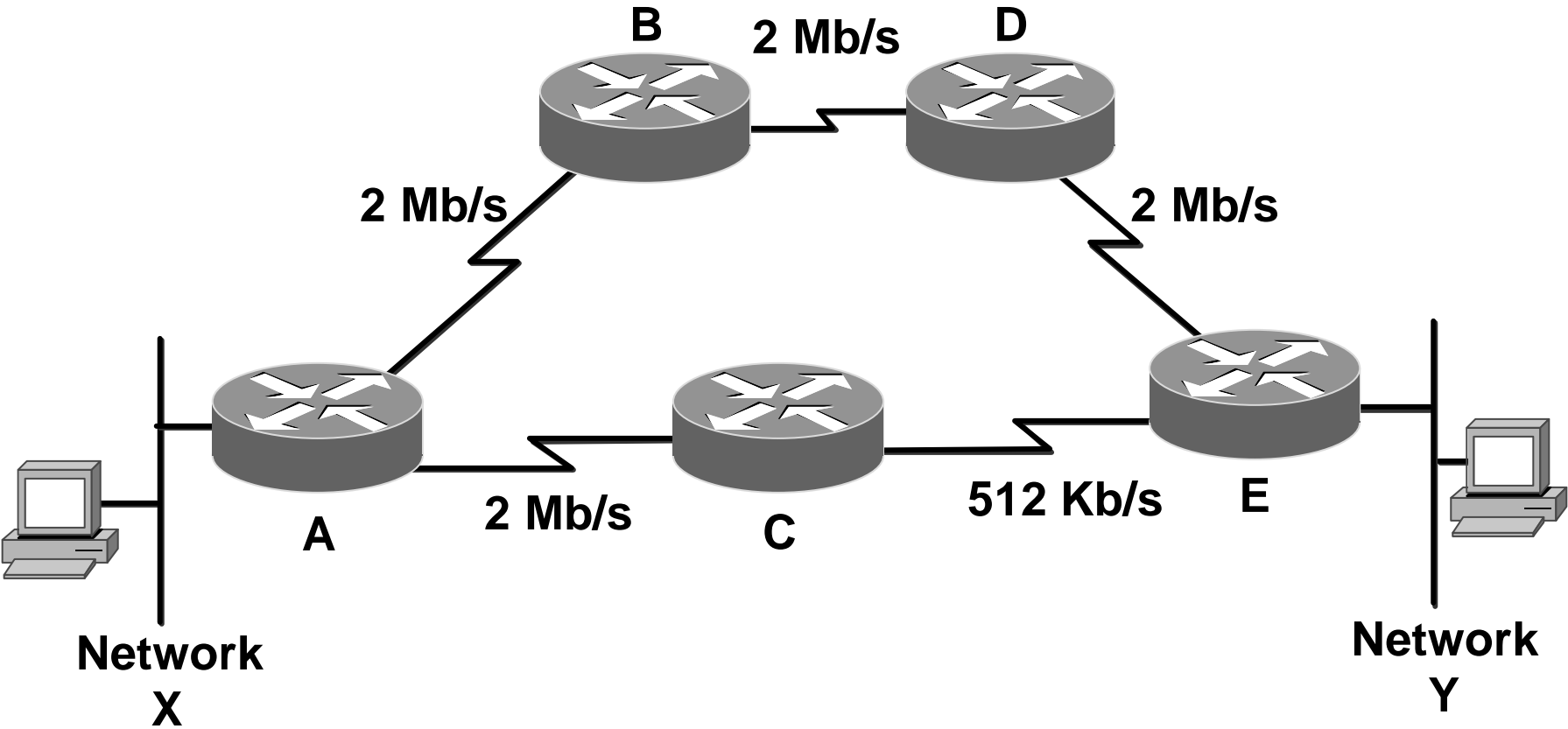
# Protocolli di routing

	<b>Algoritmo</b>	<b>Protocollo</b>
<b>Link State</b>	<b>Dijkstra SPF</b>	<b>OSPF</b>
<b>Distance Vector Tradizionale</b>	<b>Bellman-Ford</b>	<b>RIP IGRP (Cisco)</b>
<b>Distance Vector Avanzato</b>	<b>DUAL</b>	<b>EIGRP (Cisco)</b>

# Metrica

- ⇓ Quando esistono più percorsi per raggiungere la stessa destinazione, un router deve avere un meccanismo per stabilire qual'è il percorso migliore
- ⇓ La metrica è una variabile associata dal router a ciascuna route. Se esistono più route verso una destinazione il router sceglie quella a metrica più bassa
- ⇓ Numerose metriche vengono impiegate dai protocolli di routing. Alcuni protocolli utilizzano una combinazione di più metriche per calcolare il percorso migliore
- ⇓ Le metriche più comuni sono:
  - numero di hop
  - banda
  - carico
  - ritardo
  - affidabilità

# Metrica: esempio

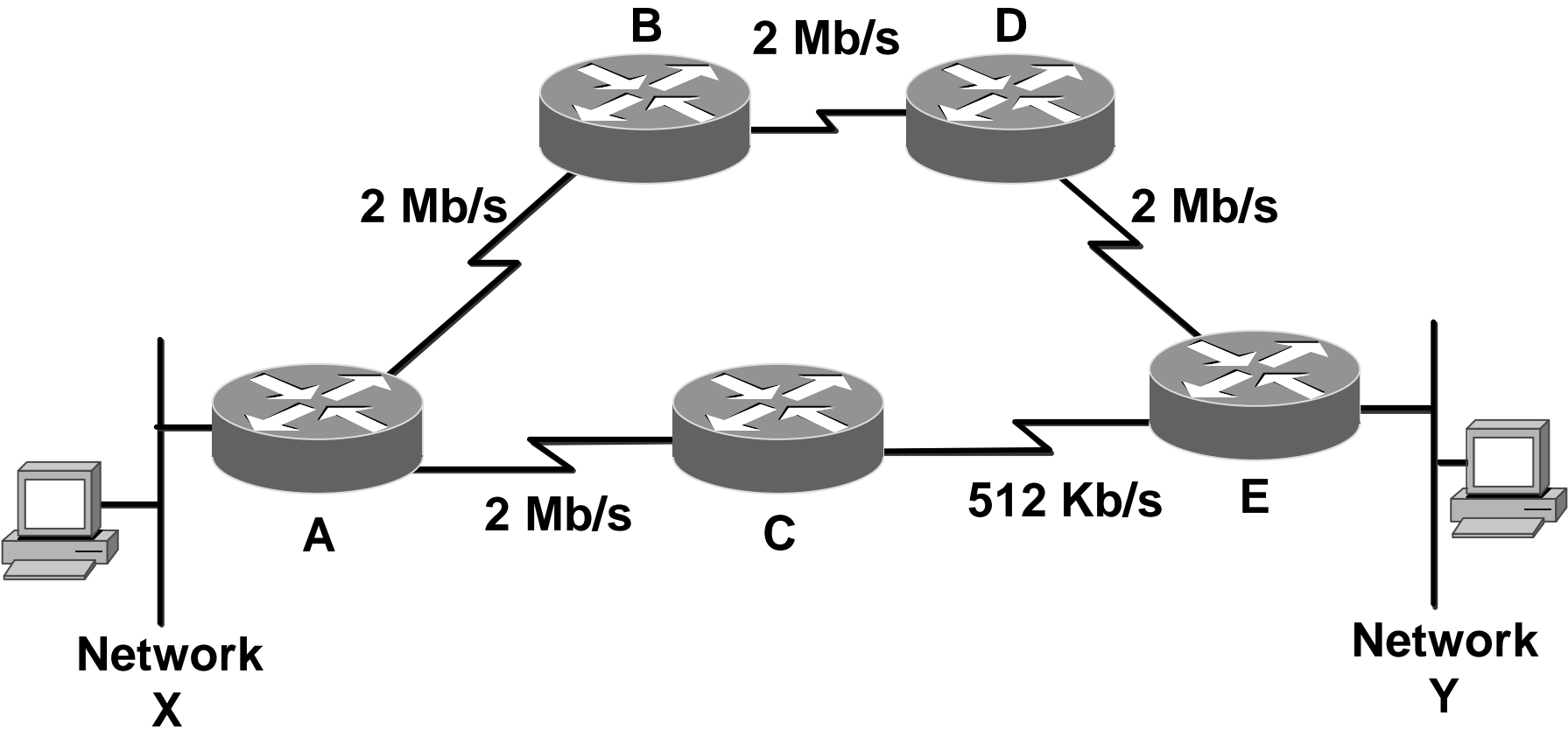


# Load Balancing

- ⇓ Con il termine di load balancing si intende la pratica di distribuire il traffico su più percorsi verso la stessa destinazione allo scopo di utilizzare le risorse di rete in maniera più efficiente
- ⇓ Un ulteriore beneficio del load balancing è il reinstradamento automatico del traffico in caso di guasti
- ⇓ Esistono due modalità di load balancing:
  - Equal-cost load balancing: il traffico viene distribuito su più percorsi aventi la stessa metrica
  - Unequal-cost load balancing: il traffico viene distribuito su più percorsi aventi metriche differenti. Il traffico viene distribuito in maniera inversamente proporzionale al costo delle route



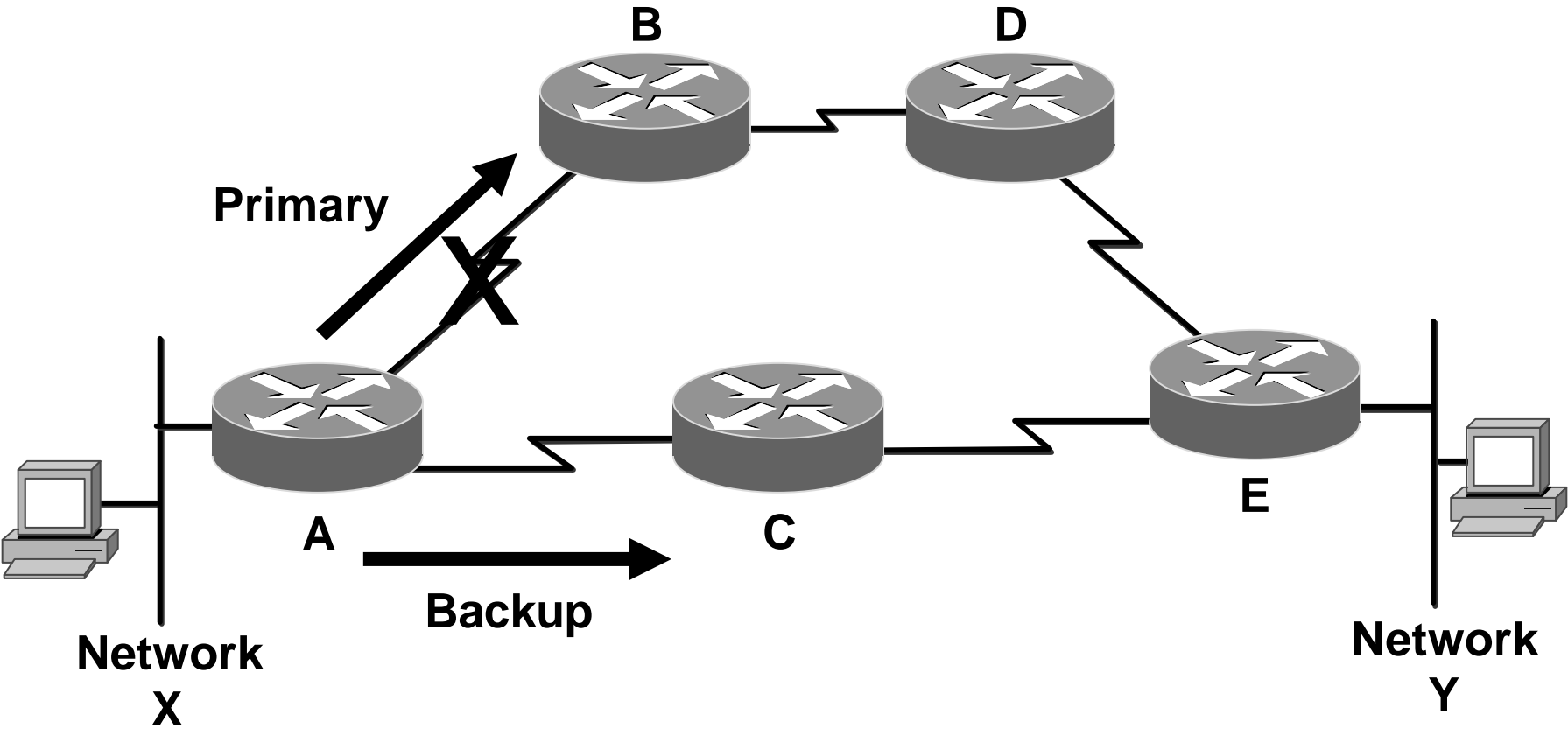
# Load balancing: esempio



# Convergenza

- ⇓ Il processo di portare tutte le tabelle di routing ad uno stato di consistenza viene denominato convergenza
- ⇓ Il tempo che intercorre tra l'istante in cui avviene una modifica nella topologia della rete (ad es., a causa di un guasto) e quello in cui tutti i router hanno aggiornato le proprie tabelle di routing viene definito tempo di convergenza
- ⇓ I protocolli di routing che convergono lentamente possono causare dei routing loops
- ⇓ In generale i protocolli di tipo link state hanno una convergenza più rapida di quelli distance vector

# Convergenza: esempio



# Distance Vector

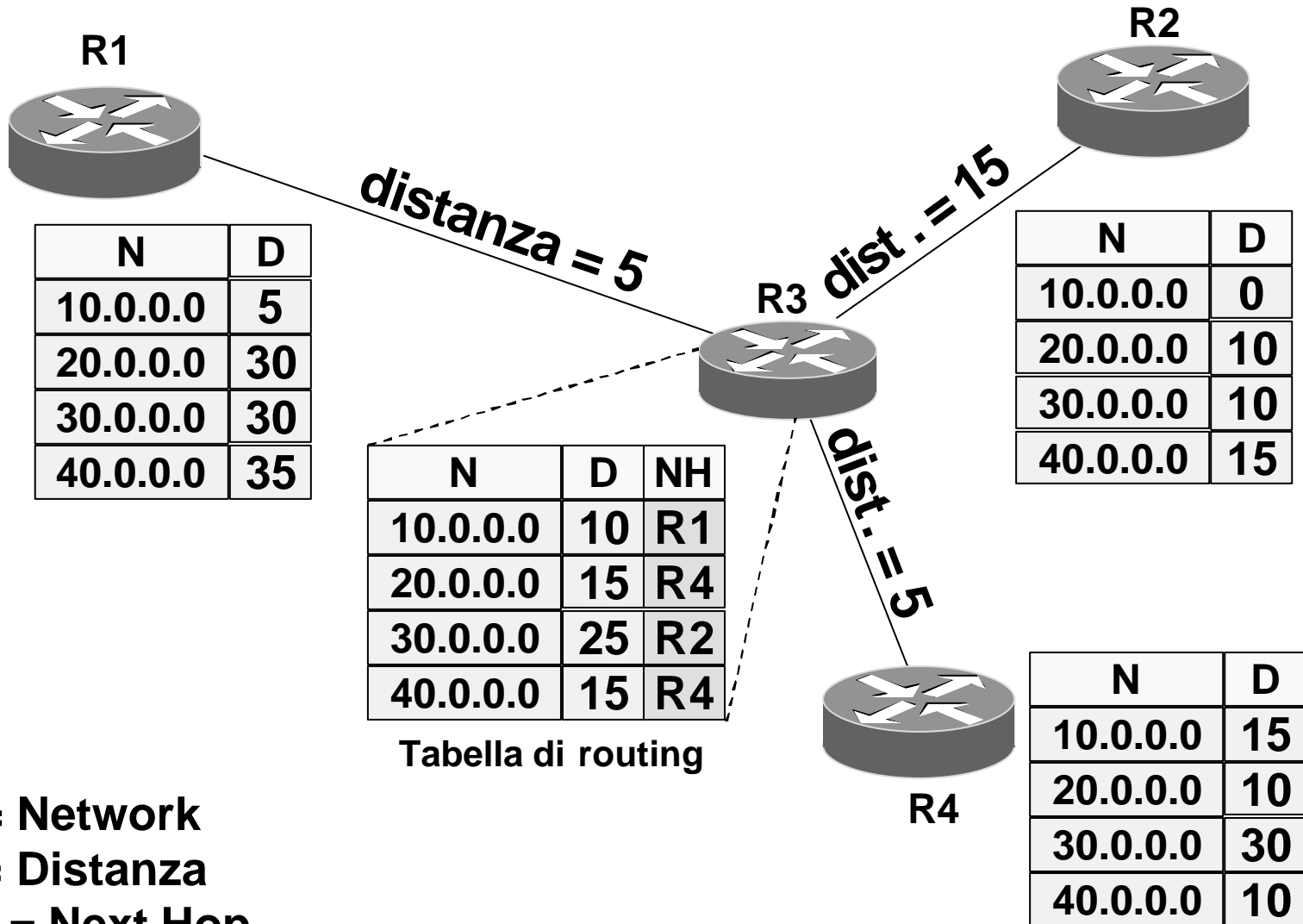
- ↴ Ogni nodo mantiene un database con le distanze minime tra sé stesso e tutte le possibili destinazioni
- ↴ Noto anche come algoritmo di Bellman-Ford
- ↴ Ogni nodo, quando modifica le proprie tabelle di instradamento, invia ai nodi adiacenti un distance vector
- ↴ Il distance vector è un insieme di coppie
  - [indirizzo - distanza]
- ↴ La distanza è espressa tramite metriche classiche quali numero di hop e costo

# Distance Vector

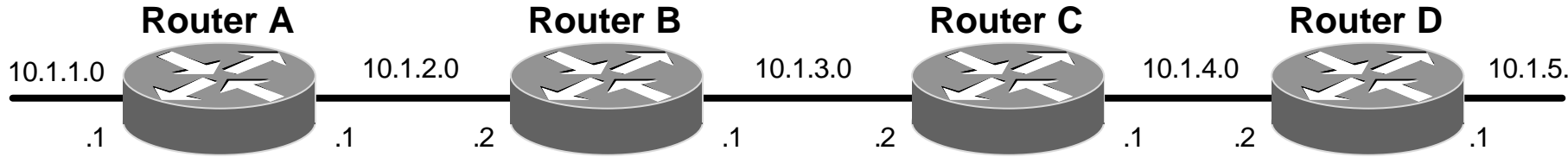
Un router che opera con un protocollo distance vector opera come segue:

- 1) Quando riceve un messaggio da un router adiacente confronta ogni coppia (destinazione, costo) col contenuto della tabella di routing e:
  - a) se la destinazione non è in tabella (e il costo dell'annuncio non è infinito) allora crea una nuova entry per la nuova destinazione e fa partire un timeout timer per la nuova entry
  - b) se la destinazione è in tabella e l'annuncio proviene dal next hop allora aggiorna il costo e fa ripartire il timeout timer
  - c) se la destinazione è in tabella e il costo indica un percorso migliore allora aggiorna il costo e il next hop nella entry e fa ripartire il timeout timer
  - d) altrimenti ignora la coppia (destinazione, costo)
- 2) Quando scatta il timeout timer pone il costo a infinito e fa partire il garbage collection timer
- 3) Quando scatta il garbage collection timer cancella la entry dalla tabella di routing
- 4) Ad intervalli regolari trasmette ai router adiacenti un messaggio che riporta tutte le coppie (destinazione, costo) contenute nella tabella di routing

# Distance Vector

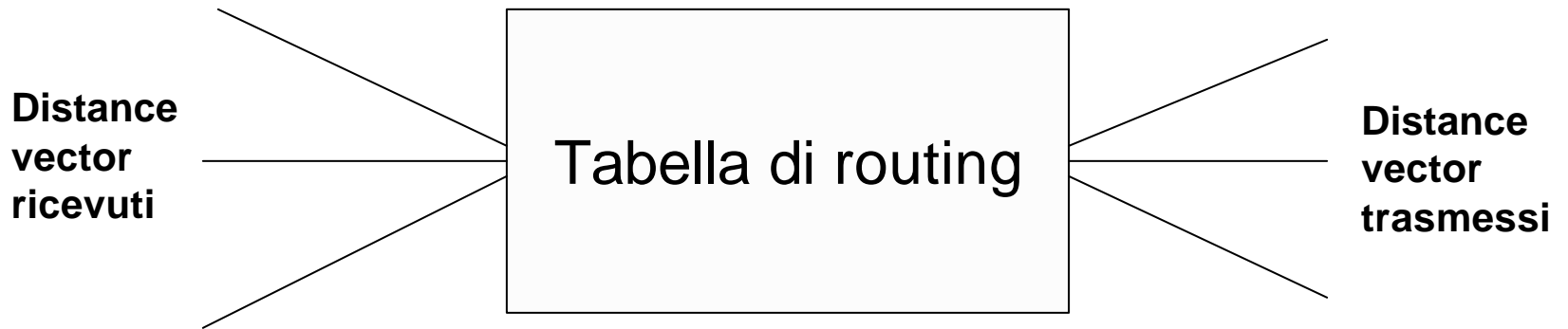


# Distance Vector: esempio



	Router A	Router B	Router C	Router D																																																																								
$t_0$	<table border="1"> <thead> <tr><th>Net</th><th>Via</th><th>Hop</th></tr> </thead> <tbody> <tr><td>10.1.1.0</td><td>---</td><td>0</td></tr> <tr><td>10.1.2.0</td><td>---</td><td>0</td></tr> </tbody> </table>	Net	Via	Hop	10.1.1.0	---	0	10.1.2.0	---	0	<table border="1"> <thead> <tr><th>Net</th><th>Via</th><th>Hop</th></tr> </thead> <tbody> <tr><td>10.1.2.0</td><td>---</td><td>0</td></tr> <tr><td>10.1.3.0</td><td>---</td><td>0</td></tr> </tbody> </table>	Net	Via	Hop	10.1.2.0	---	0	10.1.3.0	---	0	<table border="1"> <thead> <tr><th>Net</th><th>Via</th><th>Hop</th></tr> </thead> <tbody> <tr><td>10.1.3.0</td><td>---</td><td>0</td></tr> <tr><td>10.1.4.0</td><td>---</td><td>0</td></tr> </tbody> </table>	Net	Via	Hop	10.1.3.0	---	0	10.1.4.0	---	0	<table border="1"> <thead> <tr><th>Net</th><th>Via</th><th>Hop</th></tr> </thead> <tbody> <tr><td>10.1.4.0</td><td>---</td><td>0</td></tr> <tr><td>10.1.5.0</td><td>---</td><td>0</td></tr> </tbody> </table>	Net	Via	Hop	10.1.4.0	---	0	10.1.5.0	---	0																																				
Net	Via	Hop																																																																										
10.1.1.0	---	0																																																																										
10.1.2.0	---	0																																																																										
Net	Via	Hop																																																																										
10.1.2.0	---	0																																																																										
10.1.3.0	---	0																																																																										
Net	Via	Hop																																																																										
10.1.3.0	---	0																																																																										
10.1.4.0	---	0																																																																										
Net	Via	Hop																																																																										
10.1.4.0	---	0																																																																										
10.1.5.0	---	0																																																																										
$t_1$	<table border="1"> <thead> <tr><th>Net</th><th>Via</th><th>Hop</th></tr> </thead> <tbody> <tr><td>10.1.1.0</td><td>---</td><td>0</td></tr> <tr><td>10.1.2.0</td><td>---</td><td>0</td></tr> <tr><td>10.1.3.0</td><td>10.1.2.2</td><td>1</td></tr> </tbody> </table>	Net	Via	Hop	10.1.1.0	---	0	10.1.2.0	---	0	10.1.3.0	10.1.2.2	1	<table border="1"> <thead> <tr><th>Net</th><th>Via</th><th>Hop</th></tr> </thead> <tbody> <tr><td>10.1.2.0</td><td>---</td><td>0</td></tr> <tr><td>10.1.3.0</td><td>---</td><td>0</td></tr> <tr><td>10.1.1.0</td><td>10.1.2.1</td><td>1</td></tr> <tr><td>10.1.4.0</td><td>10.1.3.2</td><td>1</td></tr> </tbody> </table>	Net	Via	Hop	10.1.2.0	---	0	10.1.3.0	---	0	10.1.1.0	10.1.2.1	1	10.1.4.0	10.1.3.2	1	<table border="1"> <thead> <tr><th>Net</th><th>Via</th><th>Hop</th></tr> </thead> <tbody> <tr><td>10.1.3.0</td><td>---</td><td>0</td></tr> <tr><td>10.1.4.0</td><td>---</td><td>0</td></tr> <tr><td>10.1.2.0</td><td>10.1.3.1</td><td>1</td></tr> <tr><td>10.1.5.0</td><td>10.1.4.2</td><td>1</td></tr> </tbody> </table>	Net	Via	Hop	10.1.3.0	---	0	10.1.4.0	---	0	10.1.2.0	10.1.3.1	1	10.1.5.0	10.1.4.2	1	<table border="1"> <thead> <tr><th>Net</th><th>Via</th><th>Hop</th></tr> </thead> <tbody> <tr><td>10.1.4.0</td><td>---</td><td>0</td></tr> <tr><td>10.1.5.0</td><td>---</td><td>0</td></tr> <tr><td>10.1.3.0</td><td>10.1.4.1</td><td>1</td></tr> </tbody> </table>	Net	Via	Hop	10.1.4.0	---	0	10.1.5.0	---	0	10.1.3.0	10.1.4.1	1																		
Net	Via	Hop																																																																										
10.1.1.0	---	0																																																																										
10.1.2.0	---	0																																																																										
10.1.3.0	10.1.2.2	1																																																																										
Net	Via	Hop																																																																										
10.1.2.0	---	0																																																																										
10.1.3.0	---	0																																																																										
10.1.1.0	10.1.2.1	1																																																																										
10.1.4.0	10.1.3.2	1																																																																										
Net	Via	Hop																																																																										
10.1.3.0	---	0																																																																										
10.1.4.0	---	0																																																																										
10.1.2.0	10.1.3.1	1																																																																										
10.1.5.0	10.1.4.2	1																																																																										
Net	Via	Hop																																																																										
10.1.4.0	---	0																																																																										
10.1.5.0	---	0																																																																										
10.1.3.0	10.1.4.1	1																																																																										
$t_2$	<table border="1"> <thead> <tr><th>Net</th><th>Via</th><th>Hop</th></tr> </thead> <tbody> <tr><td>10.1.1.0</td><td>---</td><td>0</td></tr> <tr><td>10.1.2.0</td><td>---</td><td>0</td></tr> <tr><td>10.1.3.0</td><td>10.1.2.2</td><td>1</td></tr> <tr><td>10.1.4.0</td><td>10.1.2.2</td><td>2</td></tr> </tbody> </table>	Net	Via	Hop	10.1.1.0	---	0	10.1.2.0	---	0	10.1.3.0	10.1.2.2	1	10.1.4.0	10.1.2.2	2	<table border="1"> <thead> <tr><th>Net</th><th>Via</th><th>Hop</th></tr> </thead> <tbody> <tr><td>10.1.2.0</td><td>---</td><td>0</td></tr> <tr><td>10.1.3.0</td><td>---</td><td>0</td></tr> <tr><td>10.1.1.0</td><td>10.1.2.1</td><td>1</td></tr> <tr><td>10.1.4.0</td><td>10.1.3.2</td><td>1</td></tr> <tr><td>10.1.5.0</td><td>10.1.3.2</td><td>2</td></tr> </tbody> </table>	Net	Via	Hop	10.1.2.0	---	0	10.1.3.0	---	0	10.1.1.0	10.1.2.1	1	10.1.4.0	10.1.3.2	1	10.1.5.0	10.1.3.2	2	<table border="1"> <thead> <tr><th>Net</th><th>Via</th><th>Hop</th></tr> </thead> <tbody> <tr><td>10.1.3.0</td><td>---</td><td>0</td></tr> <tr><td>10.1.4.0</td><td>---</td><td>0</td></tr> <tr><td>10.1.2.0</td><td>10.1.3.1</td><td>1</td></tr> <tr><td>10.1.5.0</td><td>10.1.4.2</td><td>1</td></tr> <tr><td>10.1.1.0</td><td>10.1.3.1</td><td>2</td></tr> </tbody> </table>	Net	Via	Hop	10.1.3.0	---	0	10.1.4.0	---	0	10.1.2.0	10.1.3.1	1	10.1.5.0	10.1.4.2	1	10.1.1.0	10.1.3.1	2	<table border="1"> <thead> <tr><th>Net</th><th>Via</th><th>Hop</th></tr> </thead> <tbody> <tr><td>10.1.4.0</td><td>---</td><td>0</td></tr> <tr><td>10.1.5.0</td><td>---</td><td>0</td></tr> <tr><td>10.1.3.0</td><td>10.1.4.1</td><td>1</td></tr> <tr><td>10.1.2.0</td><td>10.1.4.1</td><td>2</td></tr> </tbody> </table>	Net	Via	Hop	10.1.4.0	---	0	10.1.5.0	---	0	10.1.3.0	10.1.4.1	1	10.1.2.0	10.1.4.1	2						
Net	Via	Hop																																																																										
10.1.1.0	---	0																																																																										
10.1.2.0	---	0																																																																										
10.1.3.0	10.1.2.2	1																																																																										
10.1.4.0	10.1.2.2	2																																																																										
Net	Via	Hop																																																																										
10.1.2.0	---	0																																																																										
10.1.3.0	---	0																																																																										
10.1.1.0	10.1.2.1	1																																																																										
10.1.4.0	10.1.3.2	1																																																																										
10.1.5.0	10.1.3.2	2																																																																										
Net	Via	Hop																																																																										
10.1.3.0	---	0																																																																										
10.1.4.0	---	0																																																																										
10.1.2.0	10.1.3.1	1																																																																										
10.1.5.0	10.1.4.2	1																																																																										
10.1.1.0	10.1.3.1	2																																																																										
Net	Via	Hop																																																																										
10.1.4.0	---	0																																																																										
10.1.5.0	---	0																																																																										
10.1.3.0	10.1.4.1	1																																																																										
10.1.2.0	10.1.4.1	2																																																																										
$t_3$	<table border="1"> <thead> <tr><th>Net</th><th>Via</th><th>Hop</th></tr> </thead> <tbody> <tr><td>10.1.1.0</td><td>---</td><td>0</td></tr> <tr><td>10.1.2.0</td><td>---</td><td>0</td></tr> <tr><td>10.1.3.0</td><td>10.1.2.2</td><td>1</td></tr> <tr><td>10.1.4.0</td><td>10.1.2.2</td><td>2</td></tr> <tr><td>10.1.5.0</td><td>10.1.2.2</td><td>3</td></tr> </tbody> </table>	Net	Via	Hop	10.1.1.0	---	0	10.1.2.0	---	0	10.1.3.0	10.1.2.2	1	10.1.4.0	10.1.2.2	2	10.1.5.0	10.1.2.2	3	<table border="1"> <thead> <tr><th>Net</th><th>Via</th><th>Hop</th></tr> </thead> <tbody> <tr><td>10.1.2.0</td><td>---</td><td>0</td></tr> <tr><td>10.1.3.0</td><td>---</td><td>0</td></tr> <tr><td>10.1.1.0</td><td>10.1.2.1</td><td>1</td></tr> <tr><td>10.1.4.0</td><td>10.1.3.2</td><td>1</td></tr> <tr><td>10.1.5.0</td><td>10.1.3.2</td><td>2</td></tr> </tbody> </table>	Net	Via	Hop	10.1.2.0	---	0	10.1.3.0	---	0	10.1.1.0	10.1.2.1	1	10.1.4.0	10.1.3.2	1	10.1.5.0	10.1.3.2	2	<table border="1"> <thead> <tr><th>Net</th><th>Via</th><th>Hop</th></tr> </thead> <tbody> <tr><td>10.1.3.0</td><td>---</td><td>0</td></tr> <tr><td>10.1.4.0</td><td>---</td><td>0</td></tr> <tr><td>10.1.2.0</td><td>10.1.3.1</td><td>1</td></tr> <tr><td>10.1.5.0</td><td>10.1.4.2</td><td>1</td></tr> <tr><td>10.1.1.0</td><td>10.1.3.1</td><td>2</td></tr> </tbody> </table>	Net	Via	Hop	10.1.3.0	---	0	10.1.4.0	---	0	10.1.2.0	10.1.3.1	1	10.1.5.0	10.1.4.2	1	10.1.1.0	10.1.3.1	2	<table border="1"> <thead> <tr><th>Net</th><th>Via</th><th>Hop</th></tr> </thead> <tbody> <tr><td>10.1.4.0</td><td>---</td><td>0</td></tr> <tr><td>10.1.5.0</td><td>---</td><td>0</td></tr> <tr><td>10.1.3.0</td><td>10.1.4.1</td><td>1</td></tr> <tr><td>10.1.2.0</td><td>10.1.4.1</td><td>2</td></tr> <tr><td>10.1.1.0</td><td>10.1.4.1</td><td>3</td></tr> </tbody> </table>	Net	Via	Hop	10.1.4.0	---	0	10.1.5.0	---	0	10.1.3.0	10.1.4.1	1	10.1.2.0	10.1.4.1	2	10.1.1.0	10.1.4.1	3
Net	Via	Hop																																																																										
10.1.1.0	---	0																																																																										
10.1.2.0	---	0																																																																										
10.1.3.0	10.1.2.2	1																																																																										
10.1.4.0	10.1.2.2	2																																																																										
10.1.5.0	10.1.2.2	3																																																																										
Net	Via	Hop																																																																										
10.1.2.0	---	0																																																																										
10.1.3.0	---	0																																																																										
10.1.1.0	10.1.2.1	1																																																																										
10.1.4.0	10.1.3.2	1																																																																										
10.1.5.0	10.1.3.2	2																																																																										
Net	Via	Hop																																																																										
10.1.3.0	---	0																																																																										
10.1.4.0	---	0																																																																										
10.1.2.0	10.1.3.1	1																																																																										
10.1.5.0	10.1.4.2	1																																																																										
10.1.1.0	10.1.3.1	2																																																																										
Net	Via	Hop																																																																										
10.1.4.0	---	0																																																																										
10.1.5.0	---	0																																																																										
10.1.3.0	10.1.4.1	1																																																																										
10.1.2.0	10.1.4.1	2																																																																										
10.1.1.0	10.1.4.1	3																																																																										

# Distance Vector: operazione di un router





# Distance Vector: caratteristiche

## ↓ Vantaggi:

- Molto semplice da implementare

## ↓ Svantaggi

- Possono innescarsi dei loop a causa di particolari variazioni della topologia
- Difficile capirne e prevederne il comportamento su reti grandi
- L'implementazione di meccanismi migliorativi appesantisce notevolmente il protocollo

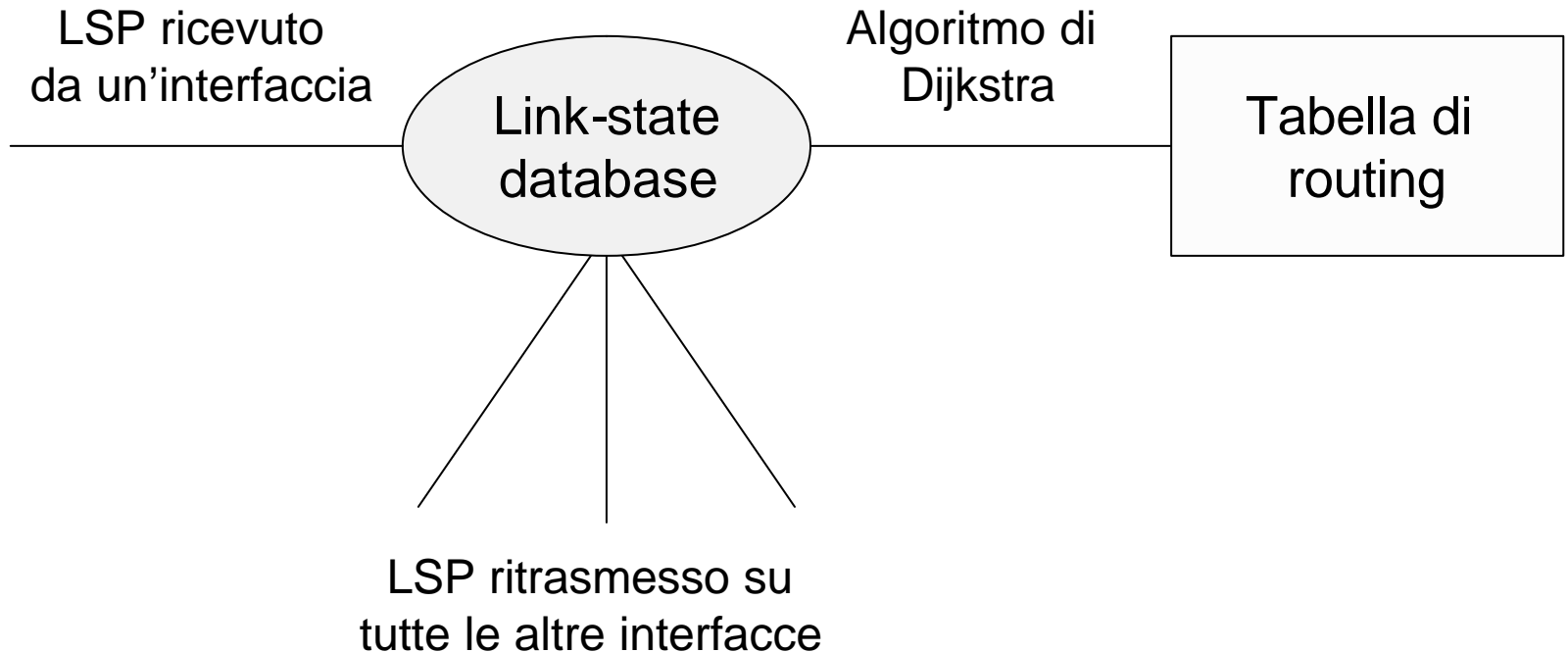
↓ Protocolli Distance Vector sono RIP, IGRP, EIGRP

# Link State

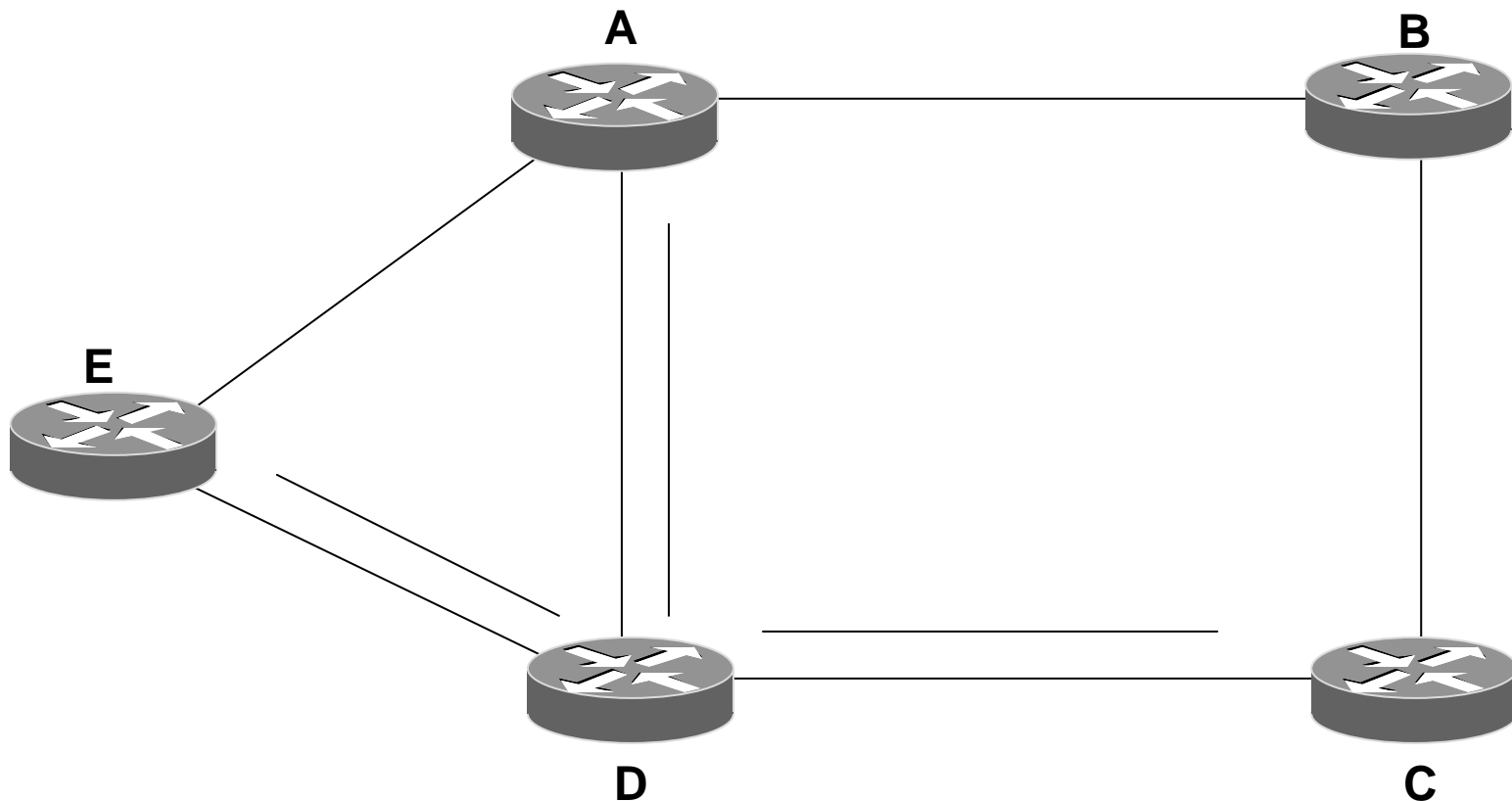
- ↴ Ogni router impara il suo ambito locale: linee e nodi adiacenti
- ↴ Trasmette queste informazioni a tutti gli altri router della rete tramite un Link State Packet (LSP)
- ↴ Tutti i router, memorizzando i LSP trasmessi dagli altri router, si costruiscono una mappa della rete
- ↴ Ogni router calcola indipendentemente la sua tabella di instradamento applicando alla mappa della rete l'algoritmo di Dijkstra o SPF (Shortest Path First)

# Link State: operazione di un router

- ↓ Il LSP è trasmesso in flooding su tutti i link del router
- ↓ I LSP memorizzati formano una mappa completa della rete
  - Link State Database

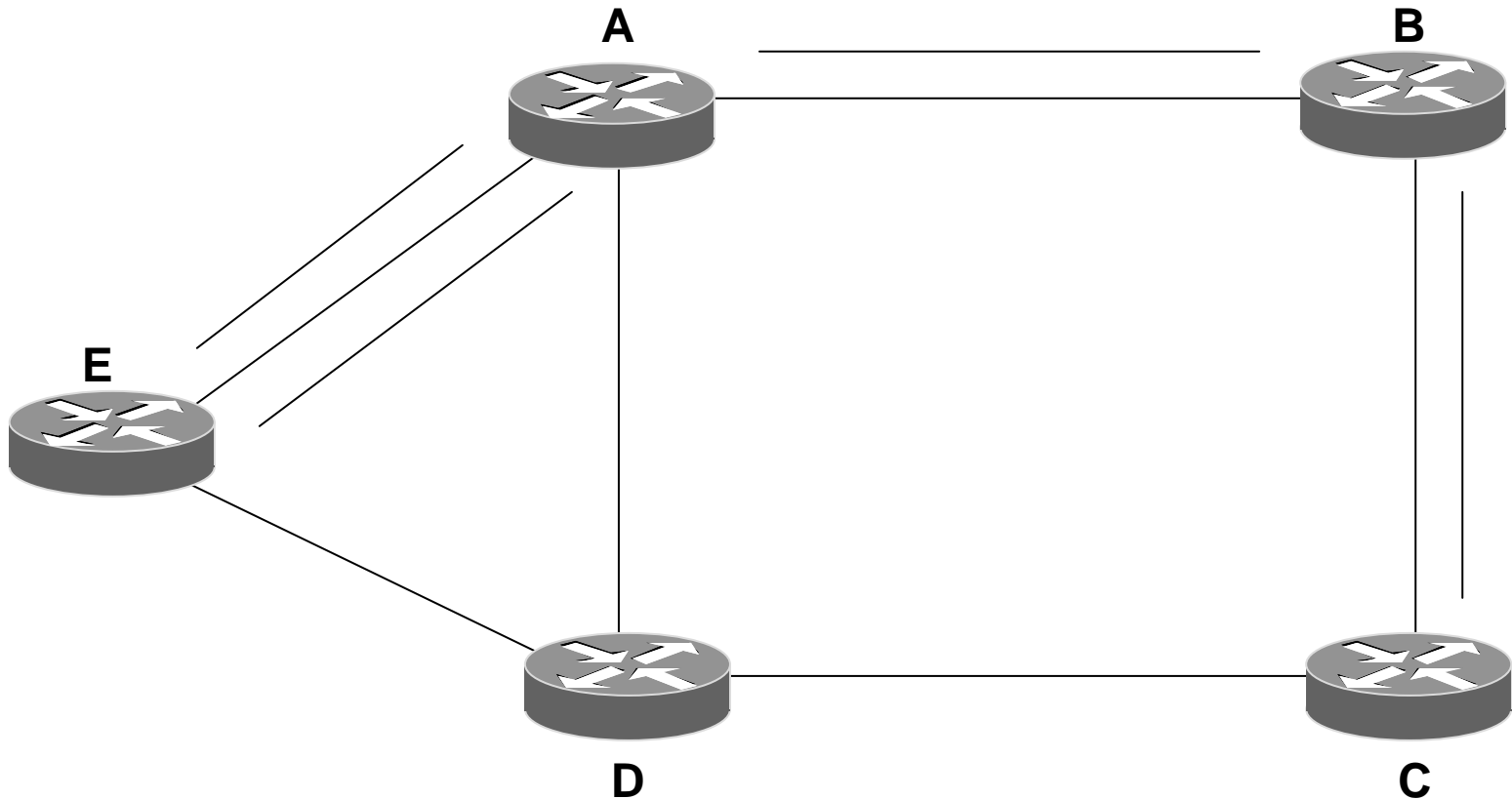


# Flooding (1)



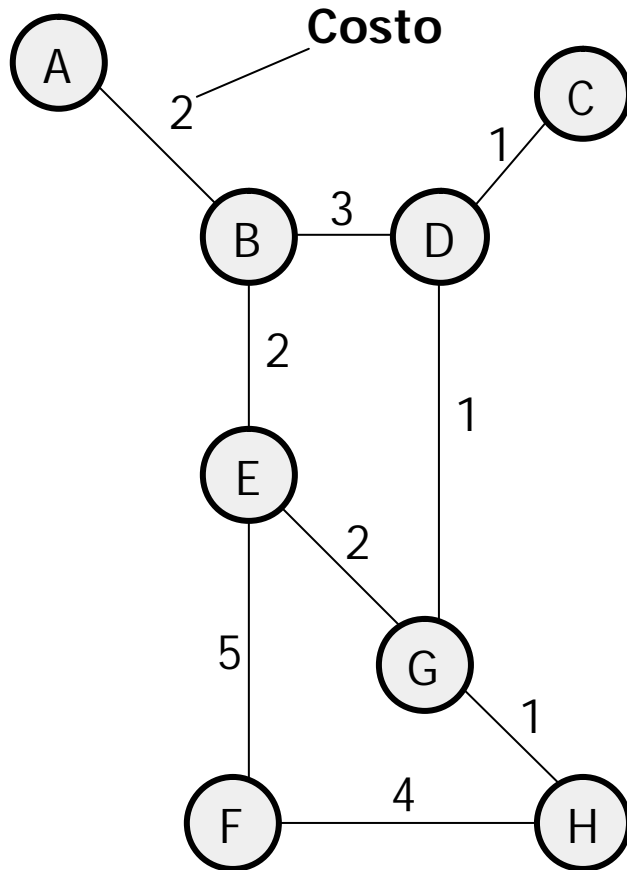
**LSP trasmesso dal router D  
a tutti i router adiacenti**

## Flooding (2)



**I router A, C, E ritrasmettono il LSP di D su tutte le interfacce tranne quella da cui lo hanno ricevuto**

# Link State Database



LSP database

A	B / 2		
B	A / 2	D / 3	E / 2
C	D / 1		
D	B / 3	C / 1	G / 1
E	B / 2	F / 5	G / 2
F	E / 5	H / 4	
G	D / 1	E / 2	H / 1
H	F / 4	G / 1	

(replicato su ogni router)

# Tabella di routing

↓ Ogni router calcola indipendentemente le sue tabelle di routing applicando alla mappa della rete l'algoritmo di Dijkstra o SPF (Shortest Path First)

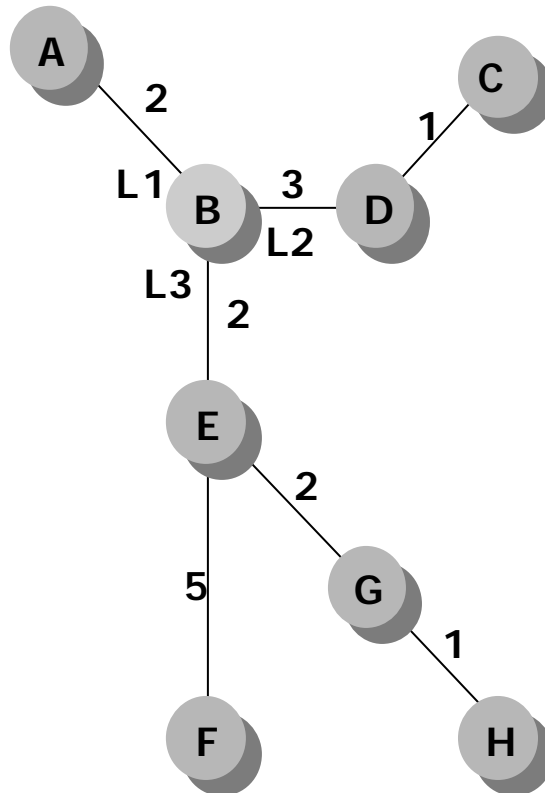


Tabella di routing di B

A	L1
C	L2
D	L2
E	L3
F	L3
G	L3
H	L3

# Link State: caratteristiche

## ↓ Vantaggi:

- Può gestire reti di grandi dimensioni
- Ha una convergenza rapida
- Difficilmente genera loop
- Ogni nodo ha la mappa della rete

## ↓ Svantaggi:

- Molto complesso da realizzare (la prima implementazione ha richiesto a Digital 5 anni)

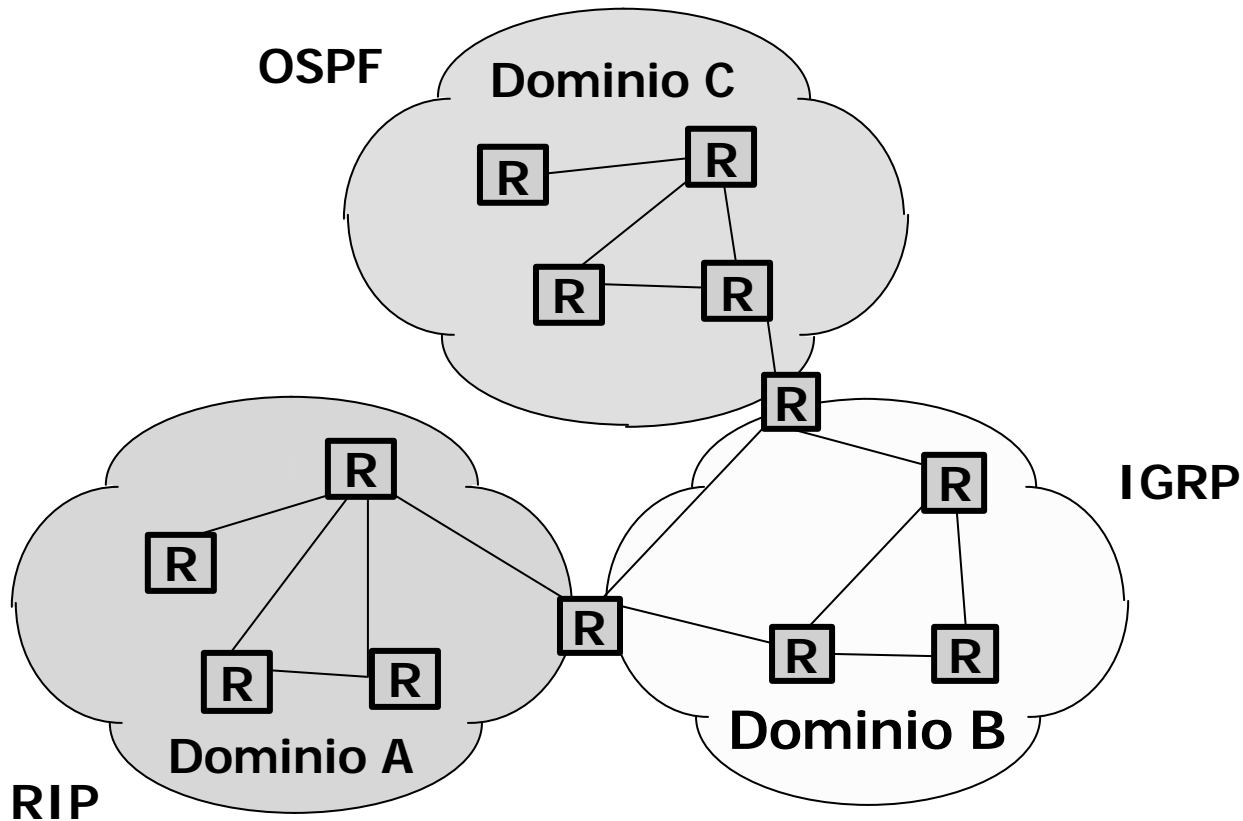
↓ È utilizzato nel protocollo OSPF



# Distance Vector vs. Link State

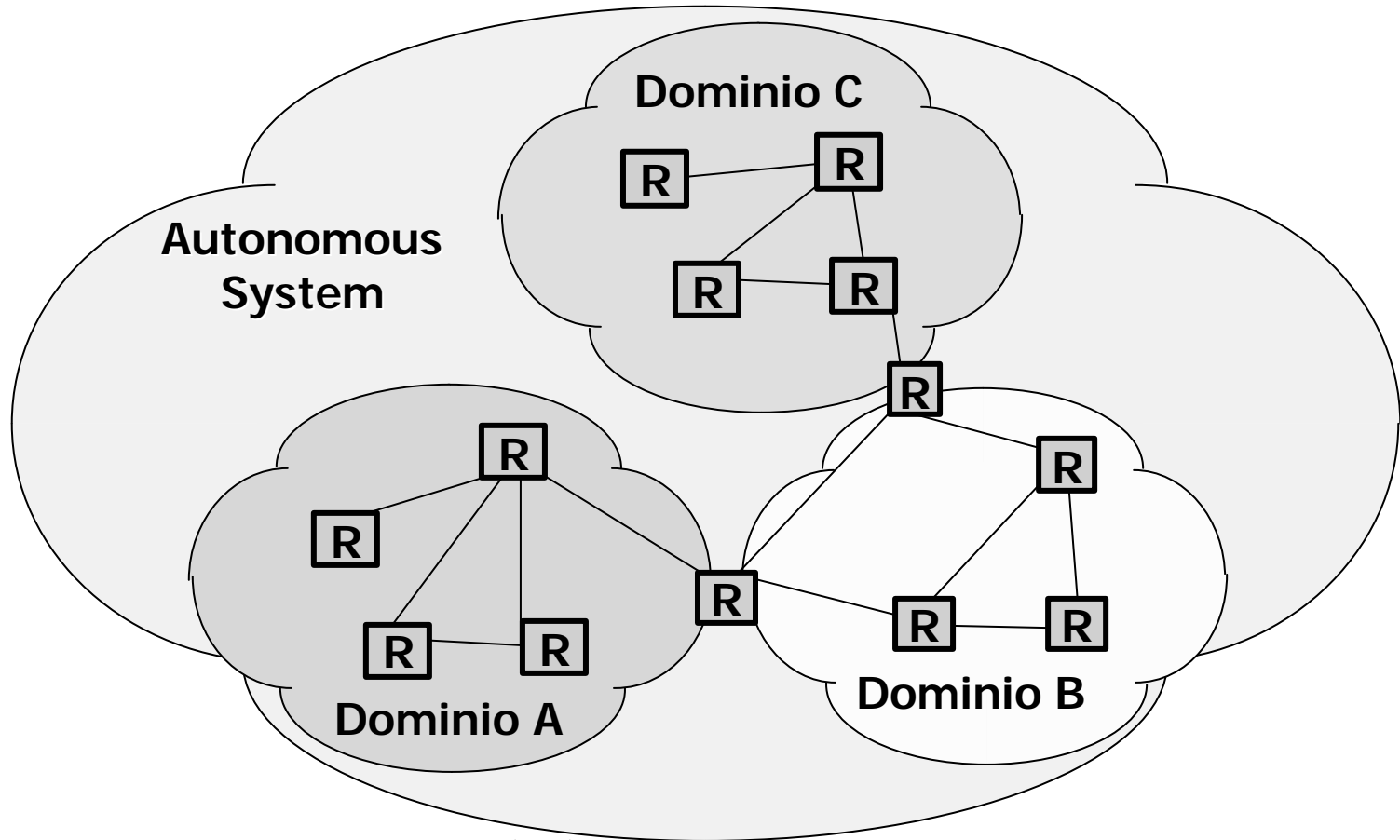
- ↴ Nel LS i router cooperano per mantenere aggiornata la mappa della rete, poi ogni router calcola indipendentemente la sua tabella di instradamento
- ↴ Nel DV i router cooperano per calcolare direttamente le tabelle di instradamento
- ↴ L'algoritmo LS può gestire reti di grandi dimensioni
- ↴ LS ha convergenza rapida e difficilmente genera loop

# Domini di routing



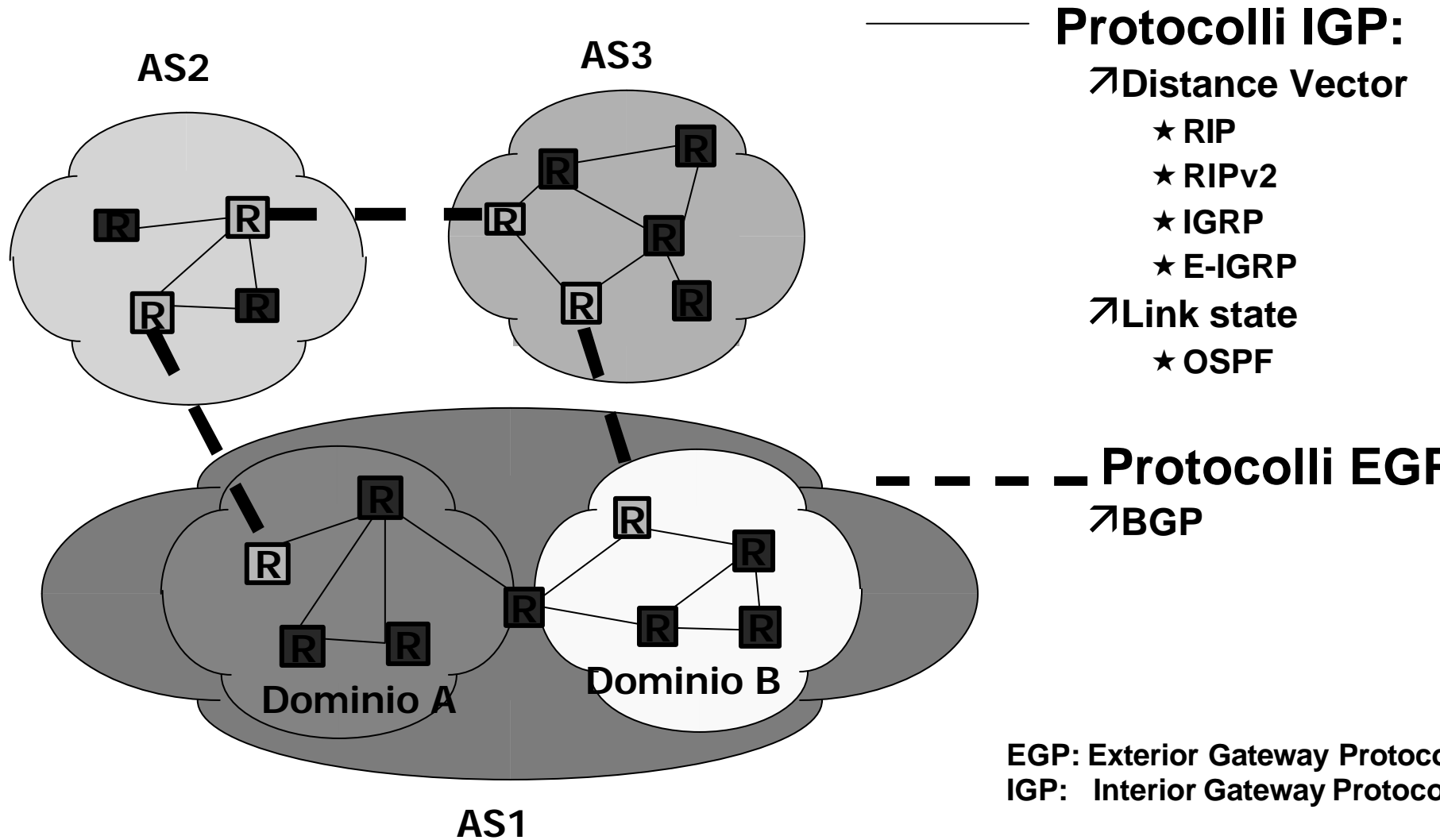
- ↓ Un dominio di routing è un insieme di router che partecipano a determinare il routing dei messaggi IP con uno stesso protocollo di routing

# Autonomous System (AS)



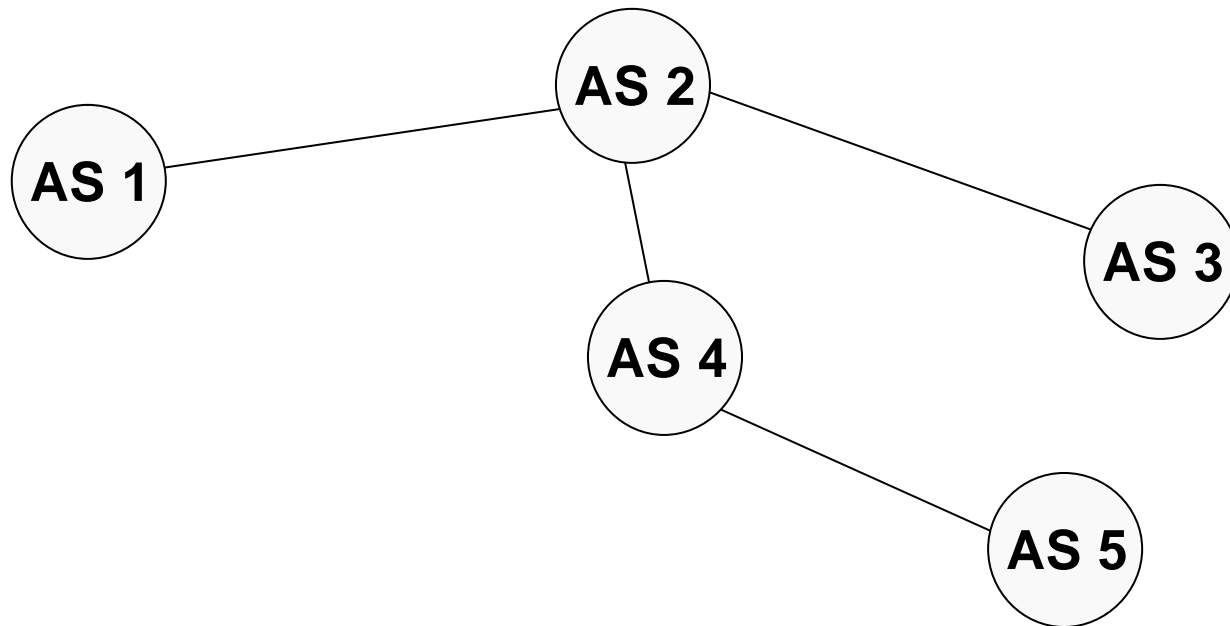
- ⇓ Un Autonomous System (AS) è una porzione della rete in è definita una particolare 'politica di routing' per lo scambio di informazioni con il resto della rete
- ⇓ Gli AS sono identificati da un AS number assegnato da RIPE/InterNIC

# Domini di routing ed AS

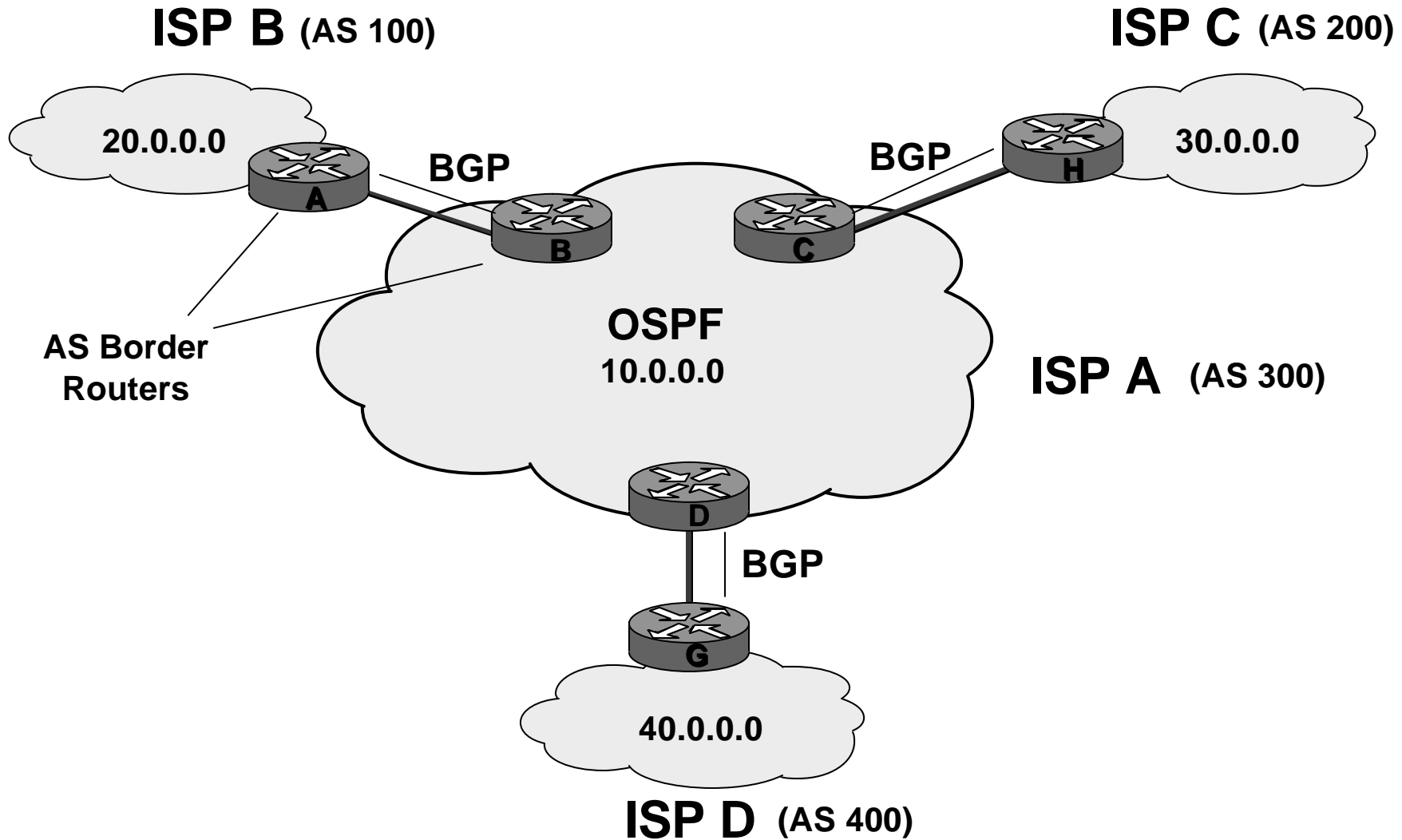


# Border Gateway Protocol

- ↓ Router adiacenti comunicano attraverso una connessione di livello trasporto affidabile
  - TCP
- ↓ Per ogni destinazione è fornita la sequenza di Autonomous System (AS) da attraversare



# ISP: esempio



ICMP e *traceroute*

# ICMP (Internet Control Message Protocol)

- ⇓ Il protocollo ICMP è descritto in RFC 792
- ⇓ Incluso in tutte le implementazioni IP, è un protocollo di basso livello che si appoggia direttamente su IP
- ⇓ Utilizzato per la trasmissione dei messaggi di errore, di messaggi di controllo e misure di prestazioni
- ⇓ I messaggi viaggiano nel campo dati del datagram IP
- ⇓ I messaggi vengono manipolati dal software IP, non dagli applicativi utente



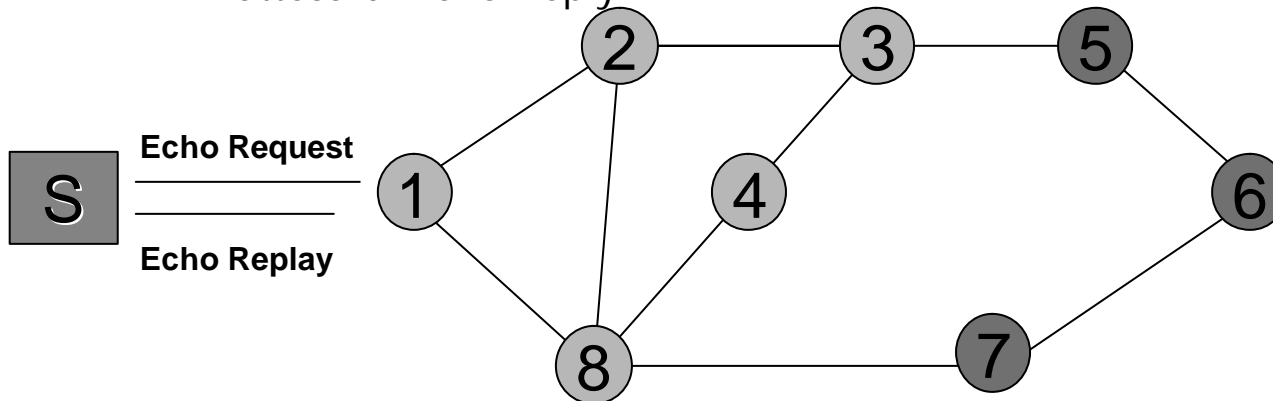
# ICMP: tipi di messaggio

Tipo	Descrizione
0	Echo Reply
3	Destination Unreachable
4	Source Quench
5	Redirect (change a route)
8	Echo Request
11	Time Exceeded for a Datagram
12	Parameter Problem for a Datagram
13	Timestamp Request
14	Timestamp Reply
15	Information Request
16	Information Reply
17	Address Mask Request
18	Address Mask Reply

# Impiego di ICMP nella gestione di rete

## ↓ PING

- diagnosi di raggiungibilità
  - ⊗ generazione di pacchetti di Echo Request verso "Echo Server"
  - ⊗ attesa di Echo Reply



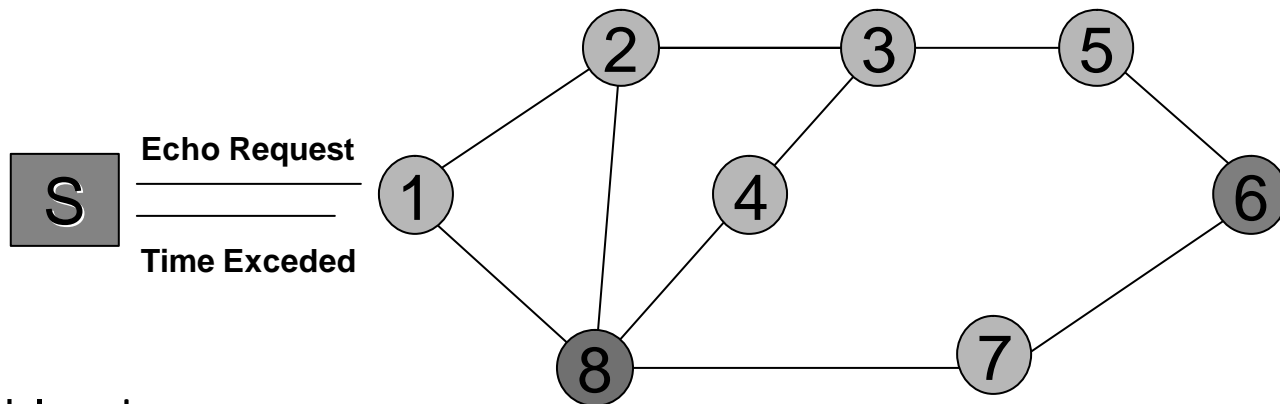
## ↓ Problemi

- scarsa capacità diagnostica
  - ⊗ cosa significa se 5,6 e 7 non rispondono al PING ?
  - ⊗ ci sono decine di possibili cause
  - ⊗ si può migliorare facendo PING da sorgenti diverse

# Impiego di ICMP nella gestione di rete

## ↓ TraceRoute

- identificazione dei percorsi sulla rete
  - ⊗ generazione di più pacchetti successivi di Echo Request
    - TTL inizia da 1 e viene incrementato di 1 ad ogni successivo Echo Request
  - ⊗ ogni pacchetto percorre un passo in più rispetto al precedente
  - ⊗ osservazione dell'indirizzo sorgente dei pacchetti di "Time Exceeded"

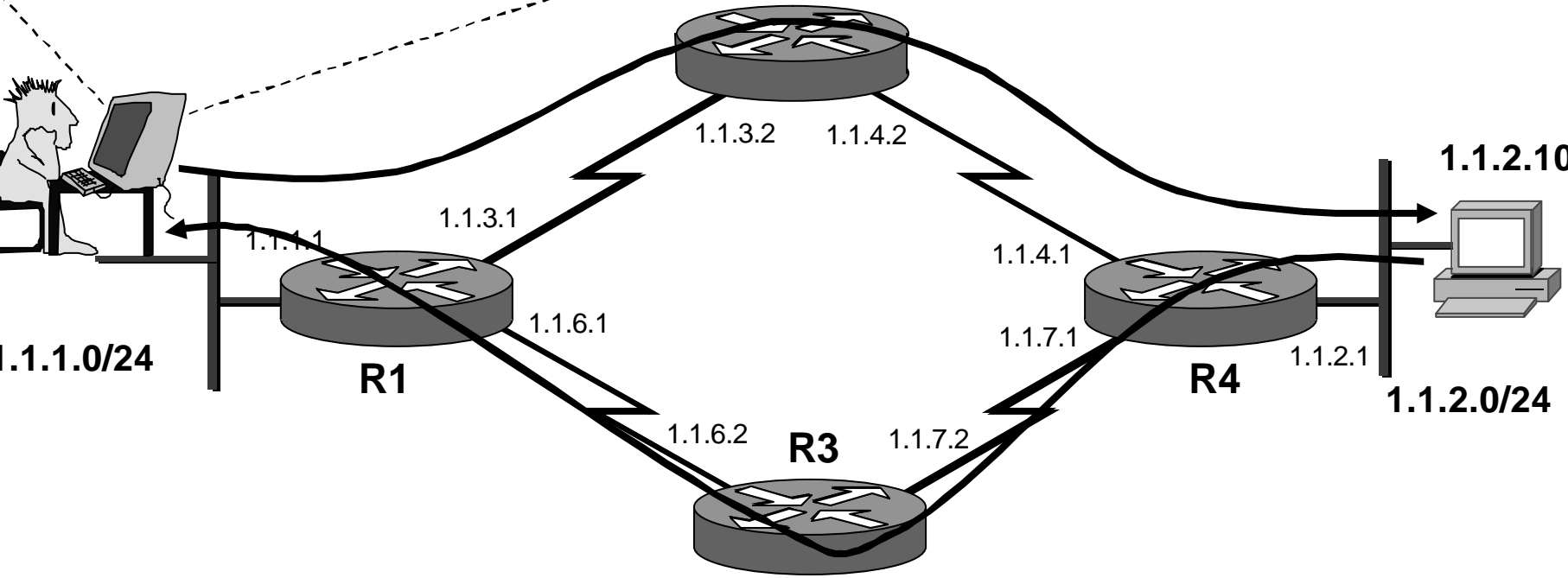


## ↓ Problemi

- come per il ping scarsa capacità diagnostica
  - ⊗ Esempio: cosa succede se il percorso dei pacchetti nelle due direzioni è diverso (Es. 1,2,3,5,6,5,3,4,8,1) ed il nodo 8 è guasto ?

# Traceroute con route asimmetriche

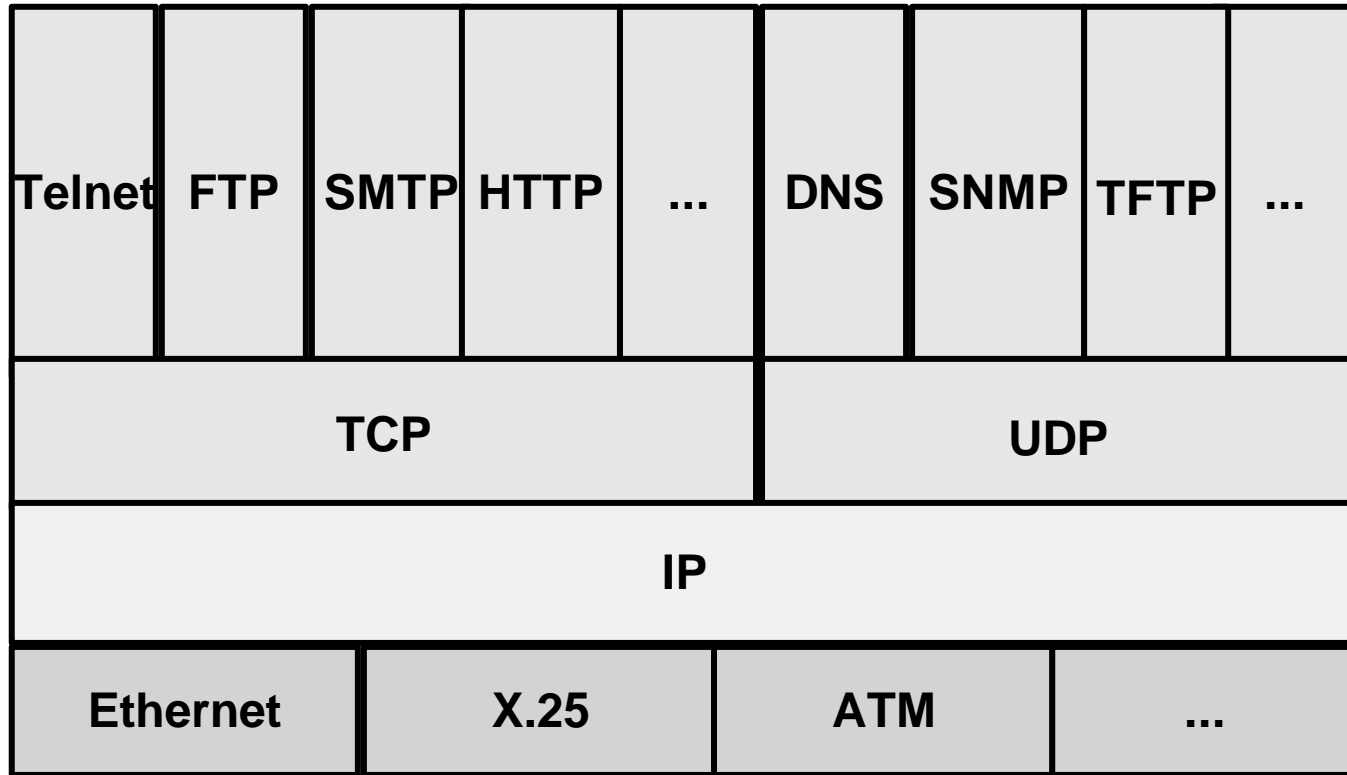
```
C:\tracert 1.1.2.10
Tracing route to 1.1.2.10 over a maximum
of 30 hops:
 1  10 ms  20 ms  10 ms  1.1.1.1
 2  30 ms  50 ms  40 ms  1.1.3.2
 3  40 ms  60 ms  60 ms  1.1.4.1
 4  90 ms  70 ms  80 ms  1.1.2.10
Trace complete.
```



# I protocolli di trasporto: TCP e UDP

# Architettura TCP/IP

↓ Pila dei protocolli Internet



# Modello Client / Server

## ↓ Definizione software

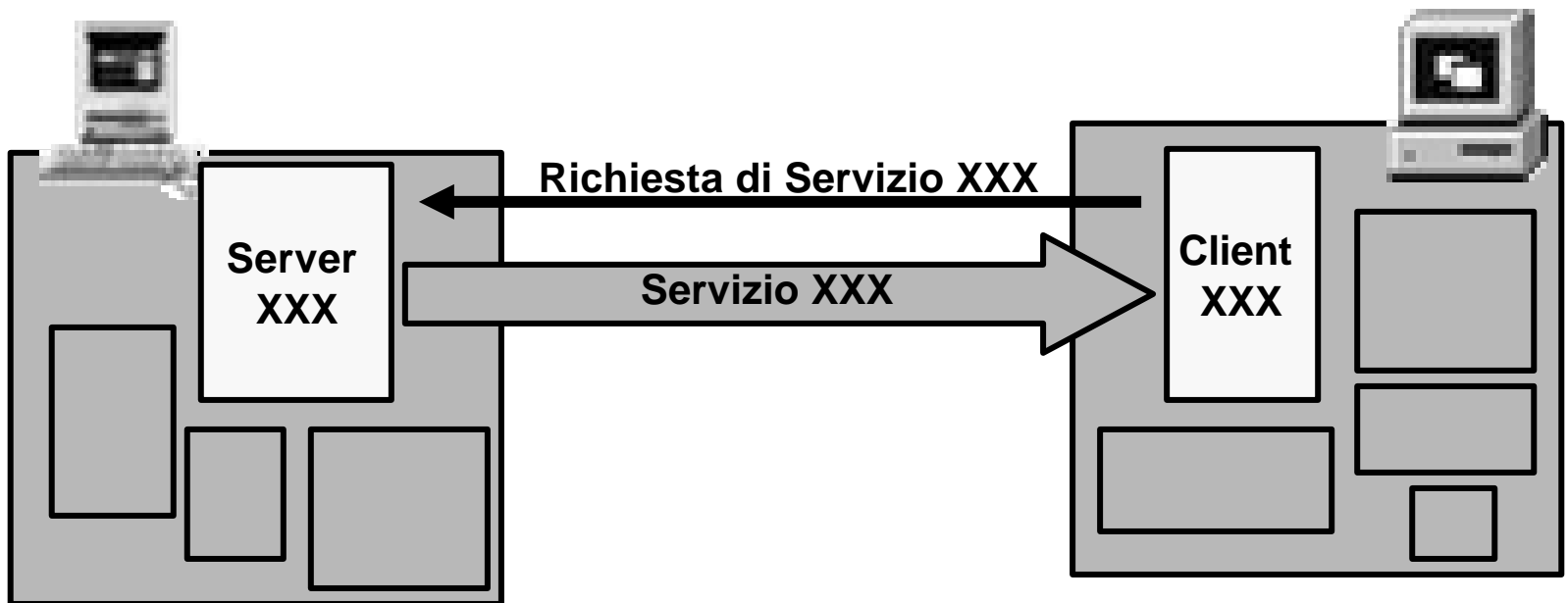
- Processo server
  - ⊗ Programma che mette a disposizione un certo servizio
- Processo client
  - ⊗ Programma che richiede l'esecuzione di un servizio al server

## ↓ Definizione hardware

- Client
  - ⊗ Workstation o PC che per la maggior parte delle applicazioni esegue il lato client e che spesso non ha la possibilità di eseguirne il lato server
- Server
  - ⊗ Host (mini, mainframe, WS, PC) che esegue per la maggior parte delle applicazioni il lato server e che spesso non possiede il lato client dell'applicazione stessa (server dedicato)

# Modello Client / Server

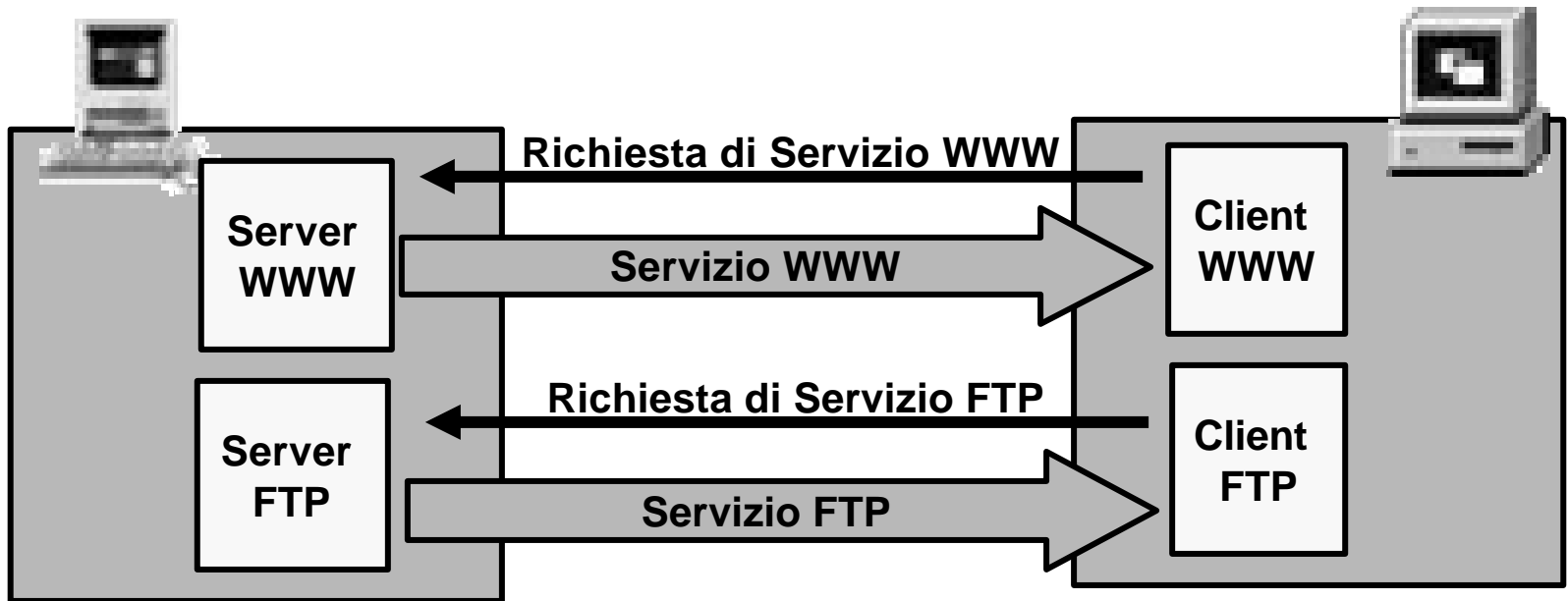
- ↓ Le applicazioni Internet sono basate sul modello Client / Server  
(Nota: gli attributi di Client o Server sono riferiti a componenti logiche dei sistemi, più che ai sistemi *in toto*)





# Modello Client / Server

↓ E' possibile implementare più processi client (server) sullo stesso host



# Strato di Trasporto

- ⇓ Lo scopo di tale strato è quello di fornire una comunicazione end-to-end a livello di processi
- ⇓ Il software di questo strato è responsabile del meccanismo che permette di distinguere, all'interno di uno stesso host, il processo applicativo destinatario (o sorgente) dei dati
- ⇓ Ogni elaboratore contiene un insieme di punti logici di accesso "ports"
- ⇓ Ad ogni servizio è associato un port che consente di indirizzare il processo che realizza il servizio (server)
- ⇓ Nell'architettura Internet esistono due standard principali di protocolli di trasporto:
  - User Datagram Protocol (UDP - RFC 768)
  - Transmission Control Protocol (TCP - RFC 793)

# Strato di Trasporto

- ⇓ La divisione dei compiti fra lo strato di trasporto (UDP, TCP) e IP è la seguente:
  - lo strato IP si occupa del trasferimento dei dati fra elaboratori collegati alle reti interconnesse; quindi l'intestazione IP identifica gli host sorgente e destinazione
  - lo strato UDP (TCP) si occupa dello smistamento dei dati fra sorgenti o destinazioni multiple all'interno dello stesso host tramite il port number
- ⇓ Per richiedere un servizio, fornito da un processo residente su un host remoto, il client deve conoscere il port number associato al servizio stesso

# Porte TCP e UDP

⇩ Sono il mezzo con cui un programma client indirizza un programma server










- un client HTTP per connettersi ad un server HTTP indica:
  - ⊗ l'indirizzo IP dell'elaboratore remoto
  - ⊗ il numero della porta associata al server HTTP

⇩ Caratteristiche

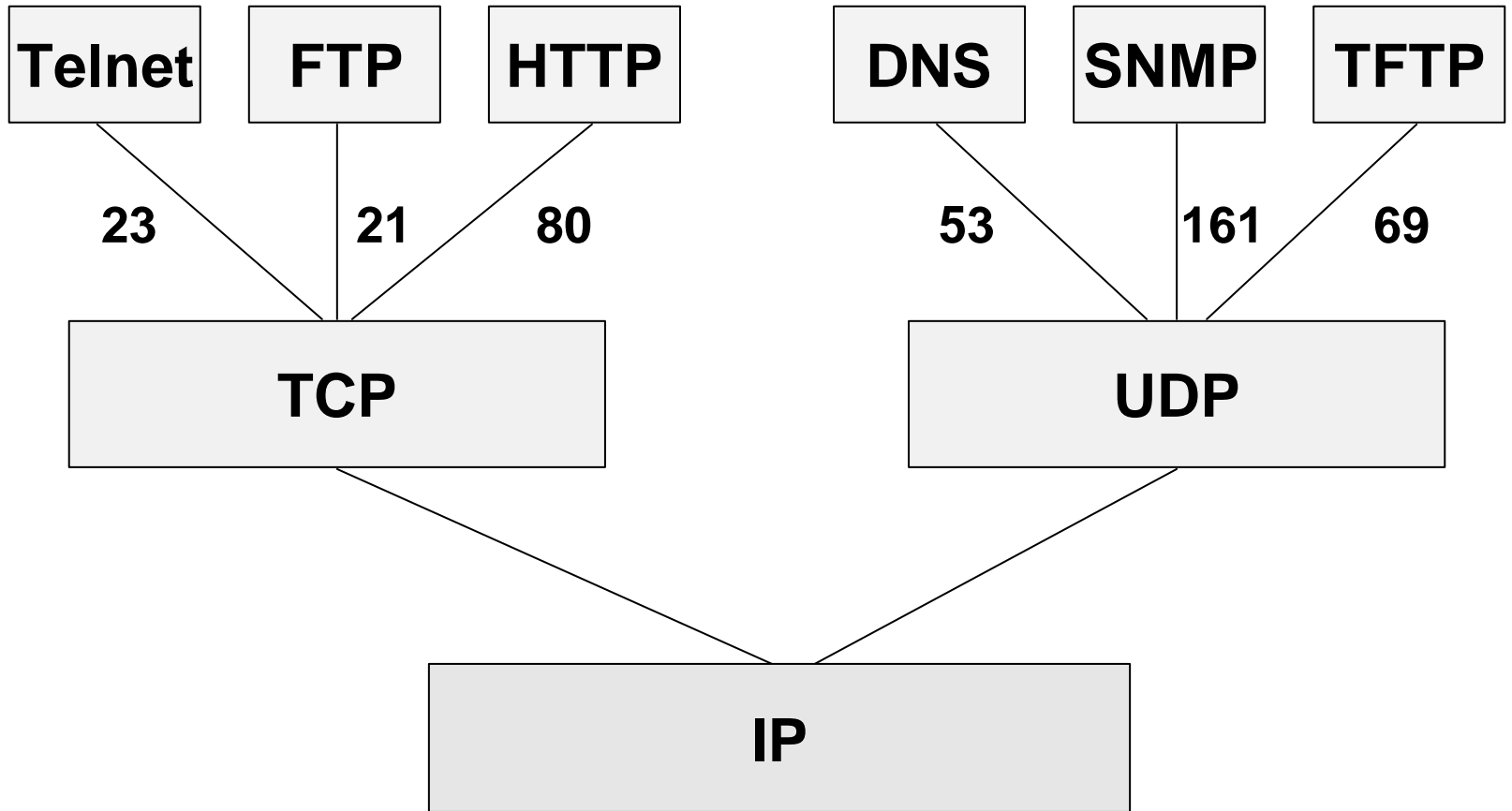
- identificate da un numero naturale su 16 bit
- 0...1023 = porte privilegiate
- 1024...65535 = porte utente

# Well Known Port

↓ Sono associate agli applicativi principali

<b>Servizio</b>	<b>Porta</b>	<b>TCP</b>	<b>UDP</b>
FTP	21		
Telnet	23		
SMTP	25		
TFTP	69		
DNS	53		
HTTP	80		
SNMP	161		
POP3	110		

# Well Known Port

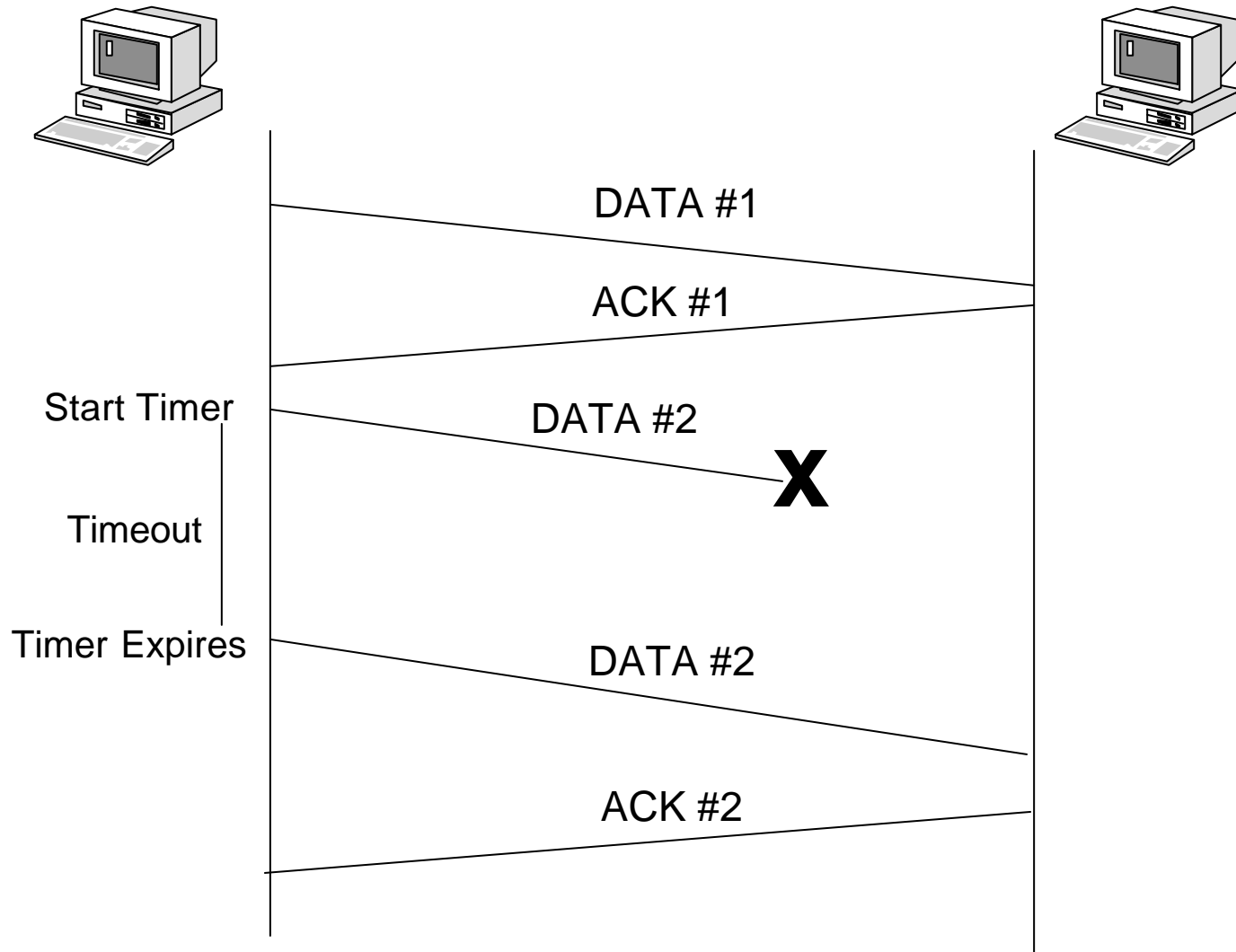


# Transmission Control Protocol (TCP)

↓ Il TCP è un protocollo affidabile in quanto:

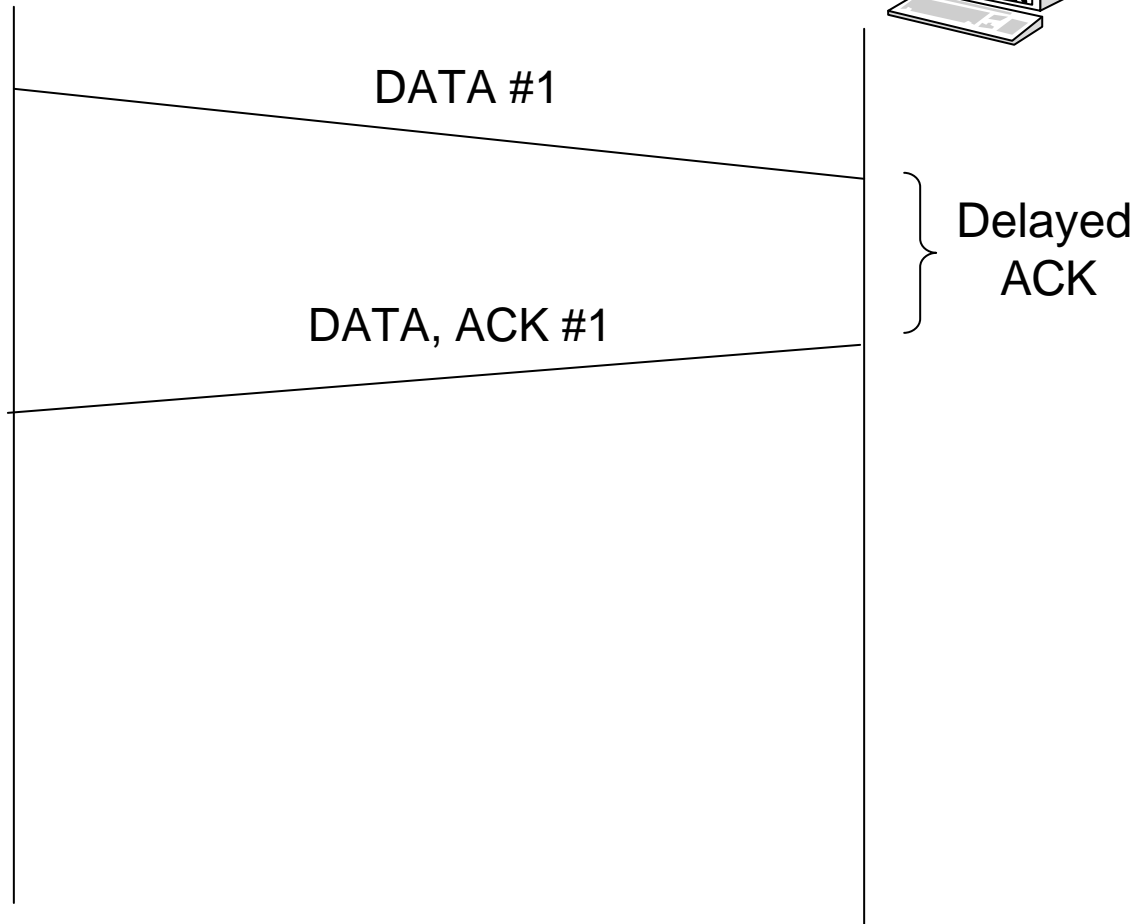
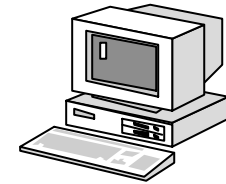
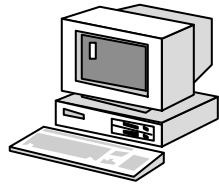
- Quando invia un segmento fa partire un timer, in attesa del riscontro della stazione ricevente. Se un riscontro non viene ricevuto in tempo il segmento viene ritrasmesso
- Quando riceve dei dati dalla stazione mittente invia un riscontro. Il riscontro non viene inviato immediatamente, ma viene ritardato di una frazione di secondo (delayed acknowledgement)
- Se un segmento arriva con un checksum non valido, il TCP scarta il segmento e non invia il riscontro. (Aspetta che il trasmettitore vada in time out e ritrasmetta il segmento)
- Poichè TCP è imbustato su IP, e poichè i pacchetti IP possono arrivare fuori sequenza, i segmenti TCP possono arrivare fuori sequenza. Il TCP se necessario riordina nella corretta sequenza i segmenti, passandoli nell'ordine corretto all'applicativo
- Fornisce il controllo di flusso. Ciascuna estremità di una connessione TCP ha un quantità finita di spazio nel buffer. Il TCP in ricezione non consente alla stazione trasmittente di inviare una quantità di dati che superi lo spazio disponibile nel proprio buffer

# Transmission Control Protocol (TCP)

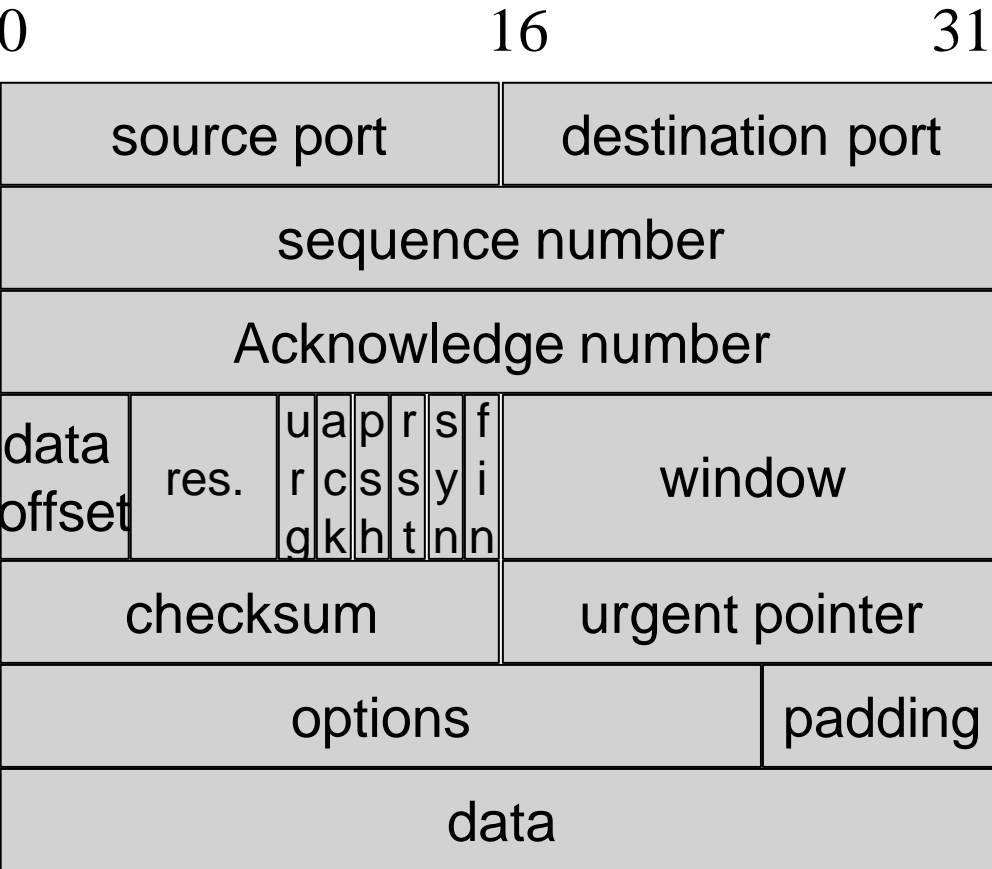




# Transmission Control Protocol (TCP)



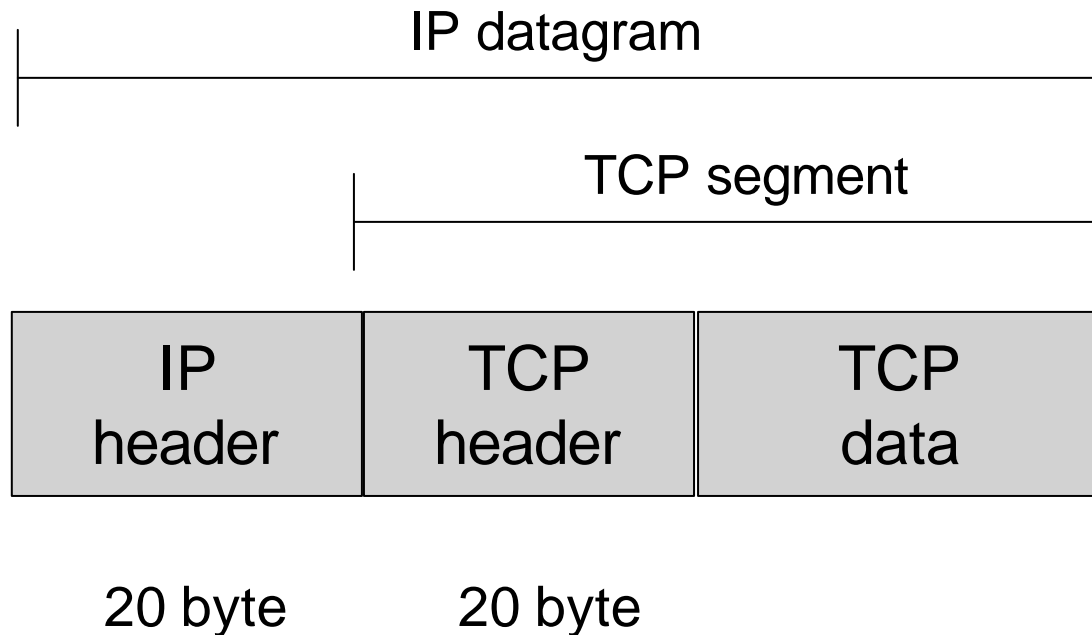
# Header TCP



- urg**: Urgent pointer field significant
- ack**: Acknowledge field significant
- psh**: Push function
- rst**: Reset connection
- syn**: Synchronize sequence number
- fin**: No more data from sender

# TCP segment

- ↓ Il segmento viene incapsulato in un IP Datagram
- ↓ La dimensione dell'header TCP e' 20 byte durante lo scambio dati, in fase di connessione possono essere presenti delle opzioni



# Connessione TCP

↓ TCP identifica un “canale” di comunicazione con il nome di connessione

↓ Sono identificate dalla quadrupla:

↓ < IP client, Port client, IP server, Port server >

↓

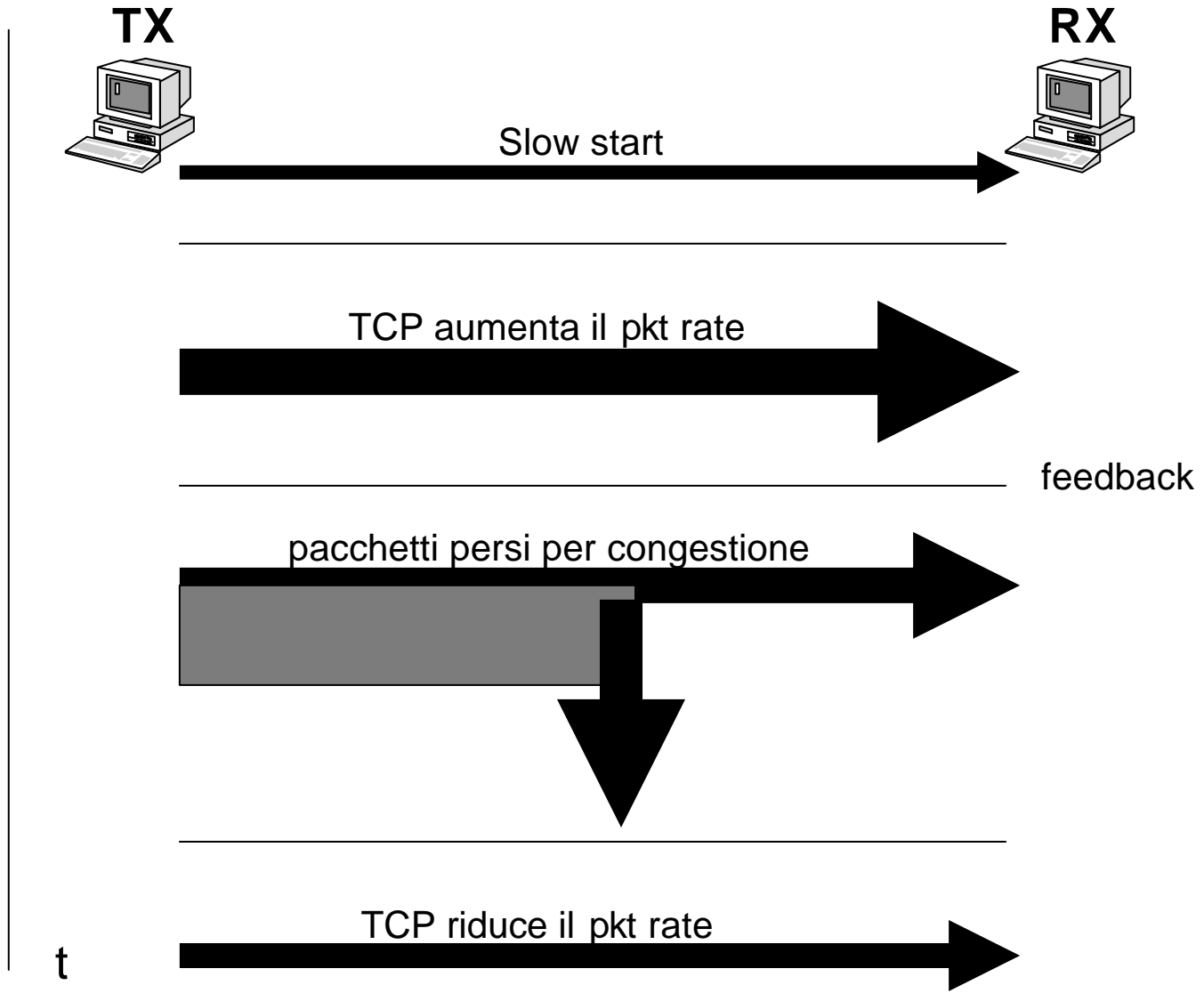
↓ Questa soluzione permette

- A molti client diversi di accedere allo stesso servizio sullo stesso server
- Allo stesso client di attivare più sessioni dello stesso servizio

# Controllo di Flusso e di Congestione

- ⇓ Il controllo di flusso ha lo scopo di limitare il flusso dei dati, prescindendo dal traffico presente nella rete
  - tale meccanismo è indispensabile in Internet dove calcolatori di dimensione e velocità molto diverse comunicano tra loro
- ⇓ Il controllo della congestione ha lo scopo di recuperare situazioni di sovraccarico nella rete

# TCP: comportamento in caso di congestione



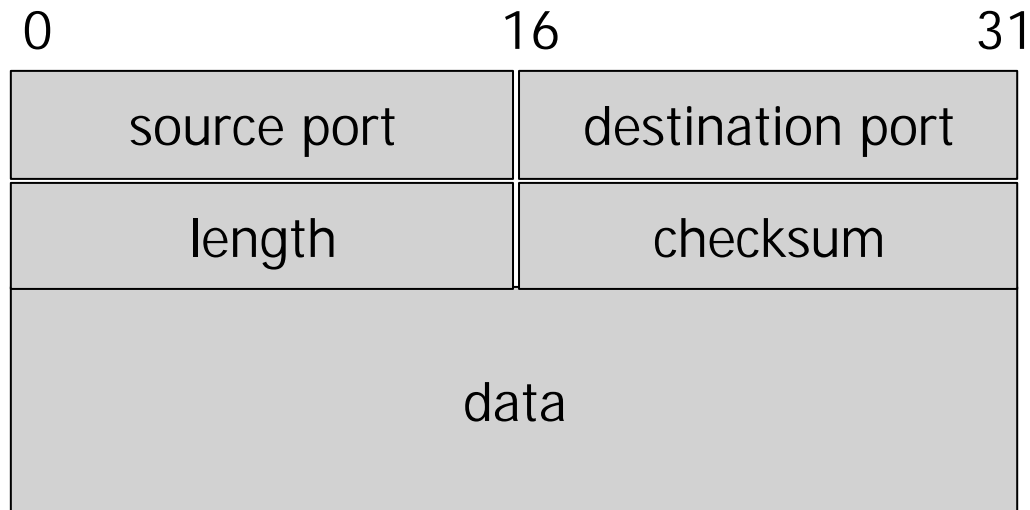
# Strato di Trasporto: UDP

↓ Servizio di trasporto fornito:

- non affidabile
- connectionless

↓ Header UDP

- port sorgente, port destinazione
- campo per eventuale checksum

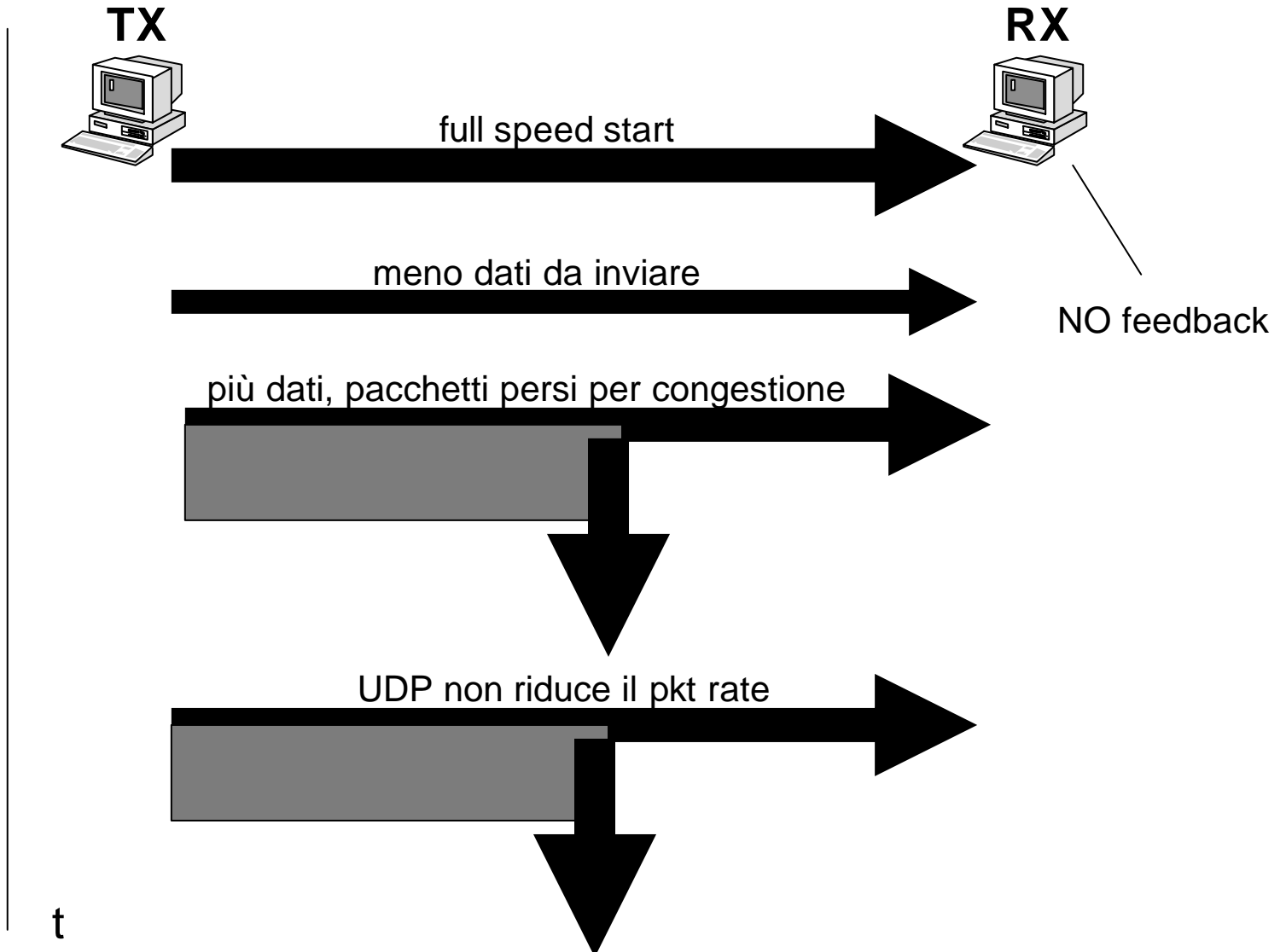


# Protocollo UDP

- ⇩ Protocollo di trasporto di tipo non connesso
- ⇩ Aggiunge due funzionalità a quelle di IP:
  - multiplexing delle informazioni tra le varie applicazioni tramite il concetto di porta
  - checksum (opzionale) per verificare l'integrità dei dati
- ⇩ Non prevede un controllo di flusso
- ⇩ Non è in grado di adattarsi autonomamente a variazioni di traffico
- ⇩ Non prevede meccanismi di ritrasmissione in caso di errori/perdite
  - eventuali meccanismi di ritrasmissione (se necessari) vengono gestiti direttamente dall'applicazione



# UDP: comportamento in caso di congestione



# NAT

(Network Address Translation)

# Indirizzi IP privati

## **IANA-Allocated, Non-Internet Routable, IP Address Schemes**

<b>Class</b>	<b>Network Address Range</b>
<b>A</b>	10.0.0.0-10.255.255.255 (10.0.0.0/8)
<b>B</b>	172.16.0.0-172.31.255.255 (172.16.0.0/12)
<b>C</b>	192.168.0.0-192.168.255.255 (192.168.0.0/16)

# NAT: benefici

- ⇩ Accedere alla Internet pubblica senza richiedere indirizzi IP registrati
- ⇩ Interconnettere reti IP con spazi di indirizzamento sovrapposti
- ⇩ Ridurre il consumo di indirizzi ufficiali (Port Address Translation, PAT)
- ⇩ Migliorare la sicurezza della rete “nascondendo” gli indirizzi reali degli host
- ⇩ Flessibilità
  - si pensi, ad es., alla possibilità di mantenere il proprio schema di indirizzamento anche nel caso di un cambiamento di ISP

# NAT: caratteristiche

- ⇓ Descritto in RFC 1631 (Maggio 1994)
- ⇓ Modifica gli indirizzi IP nell'header IP (e, se serve, nel campo applicativo) secondo la politica definita dal gestore
- ⇓ La traduzione può essere:
  - Statica
    - ⊗ mappatura statica uno-a-uno tra indirizzi interni ed esterni
  - Dinamica
    - ⊗ la corrispondenza tra indirizzo interno ed indirizzo esterno è definita solo all'occorrenza

# NAT: terminologia

## ⇩ Inside Local (IL)

- L'indirizzo IP di un host della rete interna. Questo indirizzo può essere pubblico ed univoco, pubblico ma ufficialmente assegnato ad un'altra organizzazione oppure privato

## ⇩ Inside Global (IG)

- L'indirizzo IP di un host interno così come 'appare' alla rete esterna

## ⇩ Outside Local (OL)

- L'indirizzo IP di un host esterno così come 'appare' alla rete interna

## ⇩ Outside Global (OG)

- L'indirizzo IP di un host della rete esterna

# NAT: tipi di traduzione

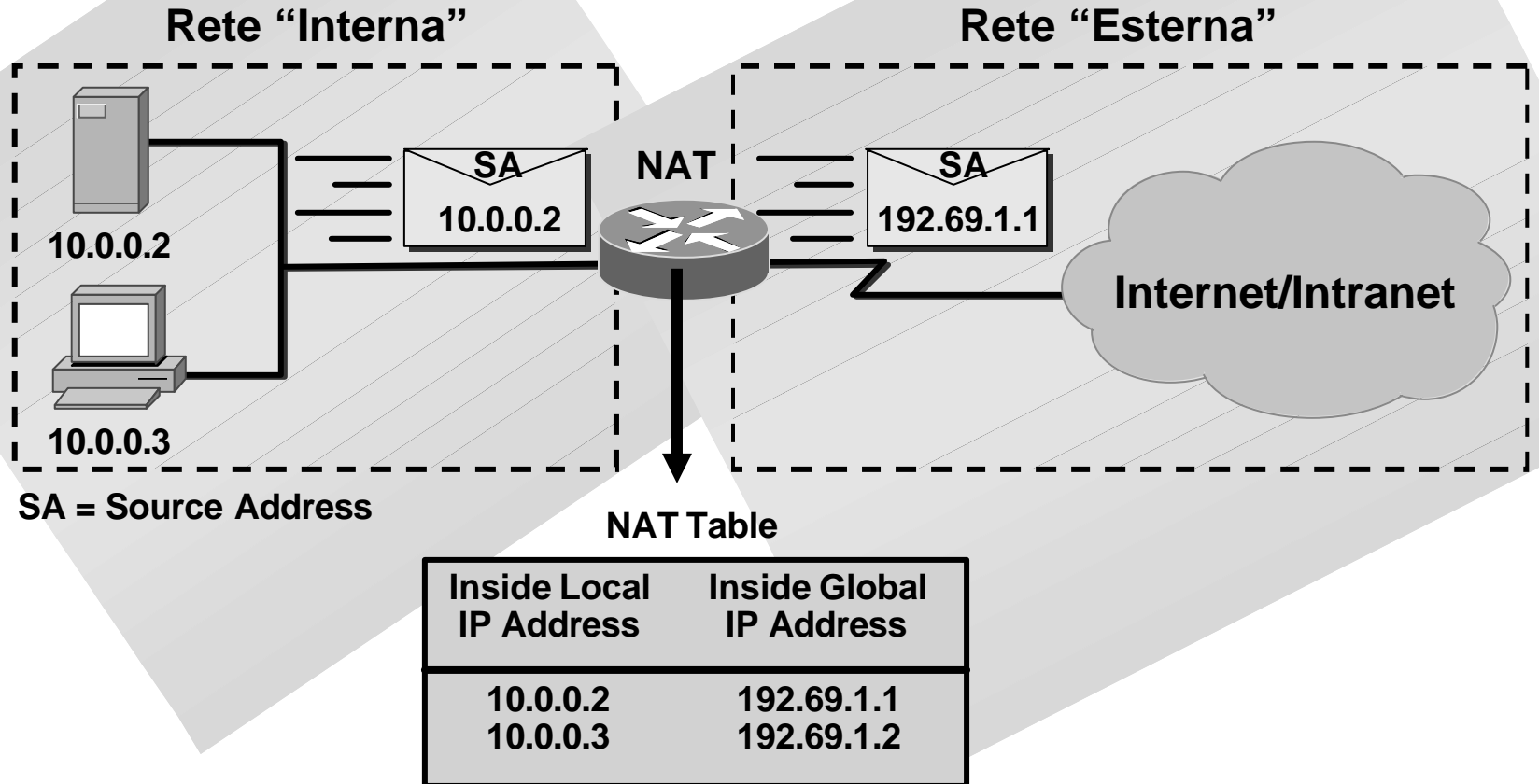
## ⇩ Network Address Translation (NAT)

- Traduce solo gli indirizzi
- Traduzione uno-a-uno, statica o dinamica
- Funzione in ambedue i versi (interno<->esterno)

## ⇩ Port Address Translation (PAT)

- Traduce le coppie Indirizzo/Port
- Traduzione uno-a-N
- Riduce il consumo di indirizzi IP registrati
- Funziona in un solo verso (interno->esterno)

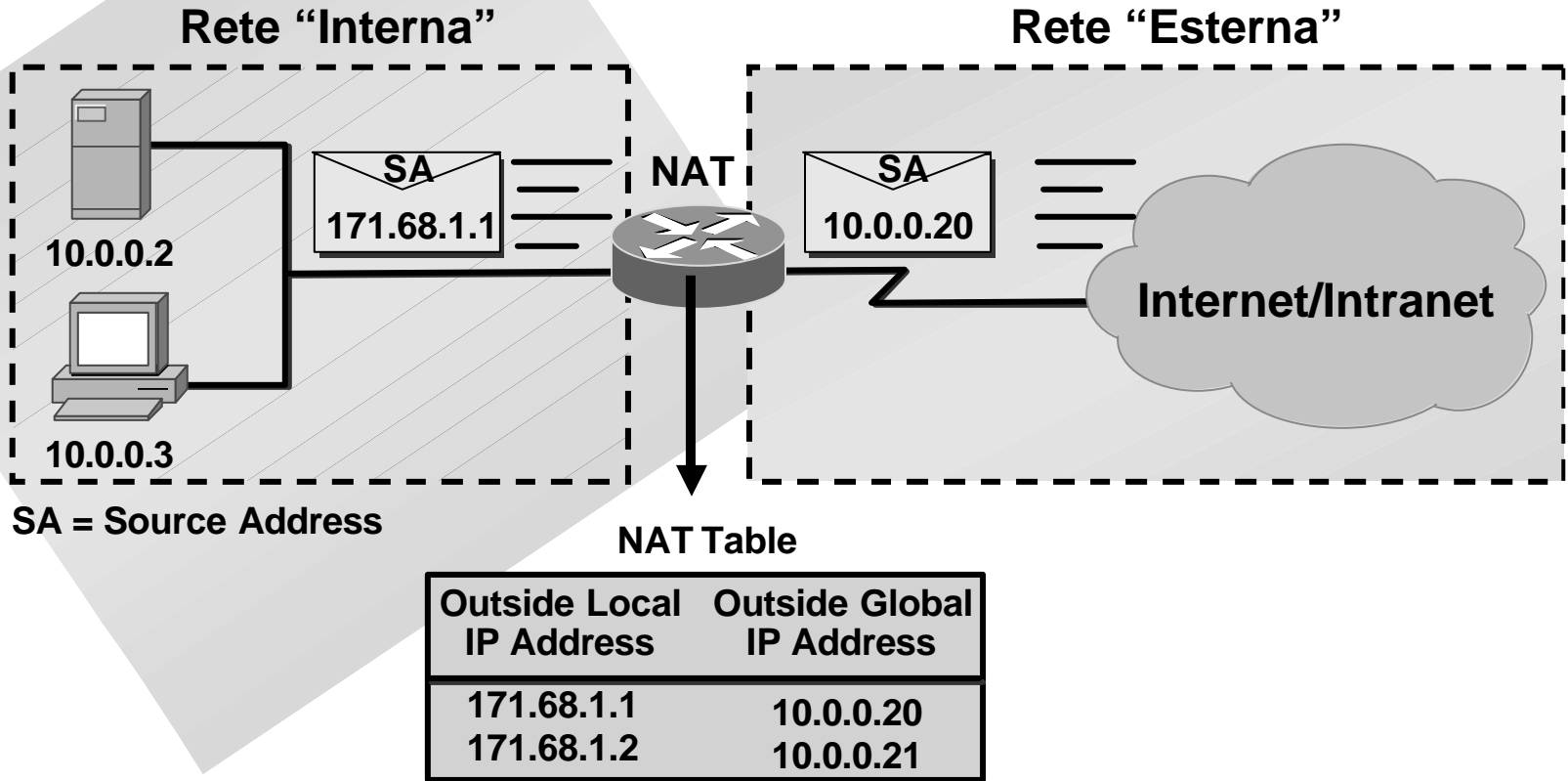
# NAT Traduzione SA interno



**Ovviamente viene anche tradotto il DA da esterno in interno nei messaggi di risposta**

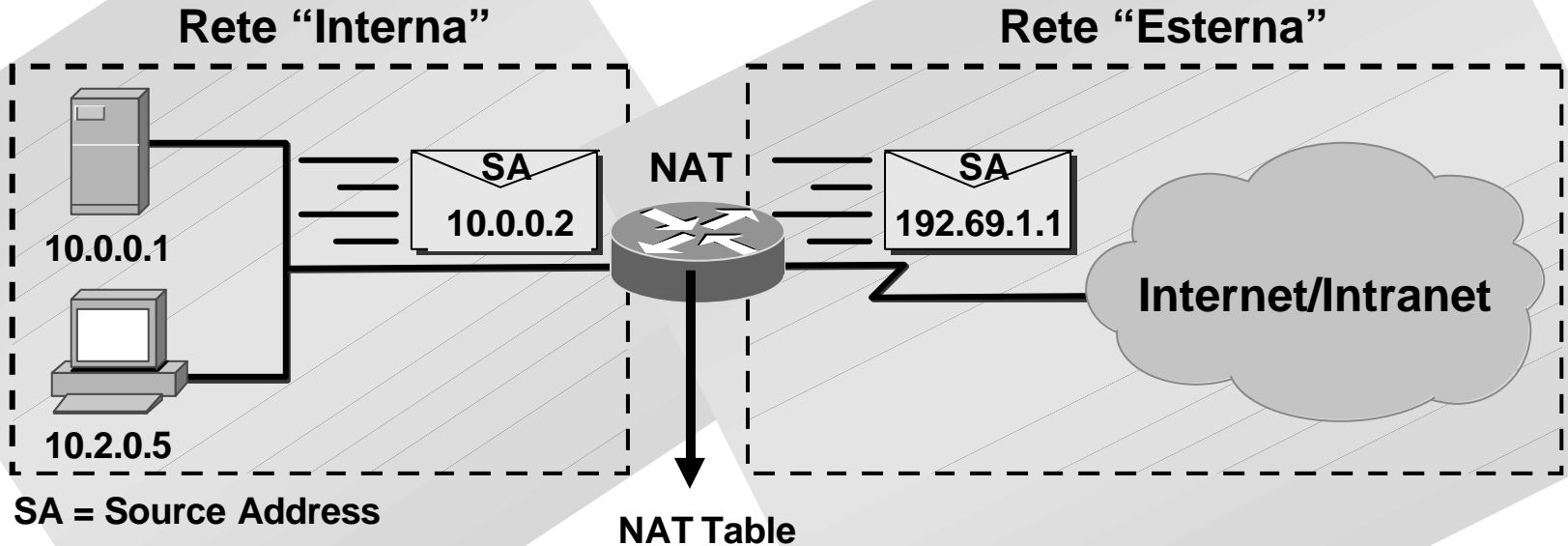


# NAT traduzione SA esterno ↗ interno



Consente di utilizzare indirizzi interni ed esterni che si sovrappongono

# Port Address Translation (PAT)



Inside Local IP Address	Inside Global IP Address
10.0.0.2:1026	192.69.1.1:5001
10.0.0.3:1029	192.69.1.1:5002

- ⇓ Tutti gli host interni utilizzano un singolo indirizzo IP registrato
- ⇓ Vengono utilizzate le porte TCP/UDP per individuare il reale mittente/destinatario del pacchetto

# NAT: considerazioni

↓ Oltre a modificare l'indirizzo IP nell'intestazione del pacchetto devono essere effettuate anche altre operazioni:

- ricalcolo della checksum IP
- ricalcolo della checksum TCP
- modifica del campo dati se questo contiene riferimenti all'indirizzo IP da tradurre

# NAT: applicativi supportati

↓ Gli applicativi supportati:

- HTTP, TFTP, Telnet, NFS
- ICMP\*, FTP\*, DNS\*

↓ Gli applicativi non supportati:

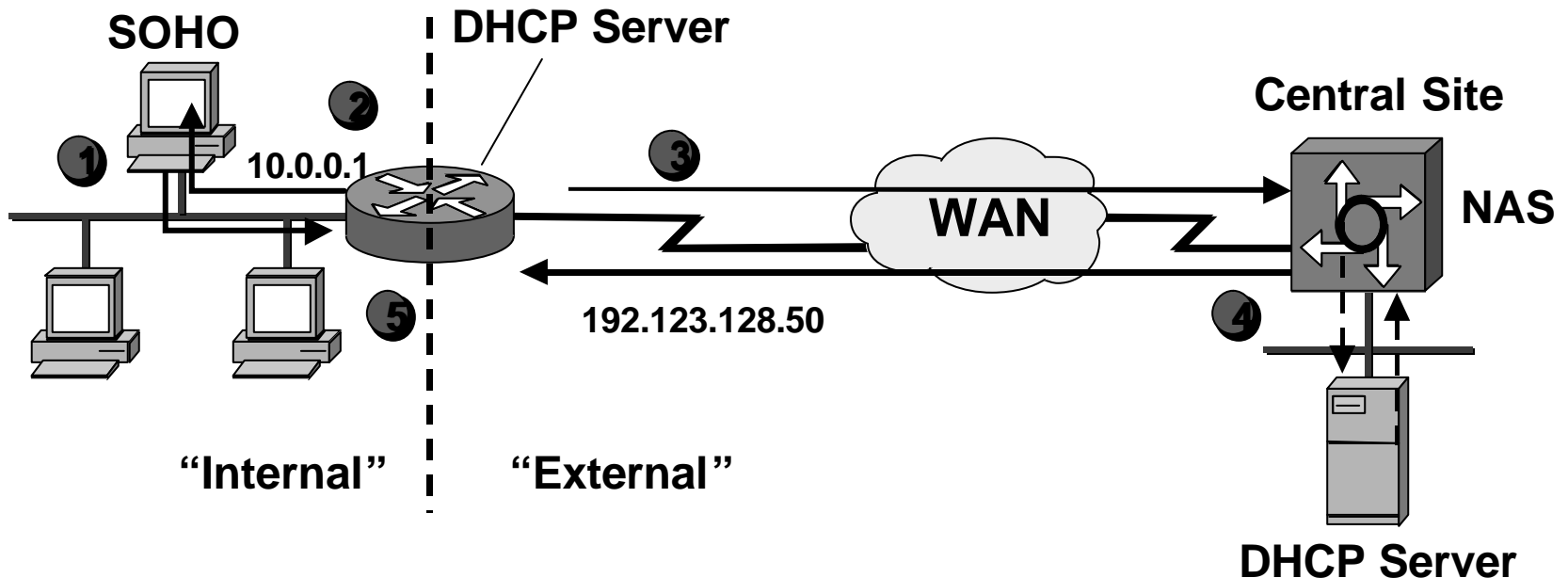
- DHCP
- SNMP
- IP multicast
- Routing table updates

\* non da tutte le implementazioni

# NAT: FAQs

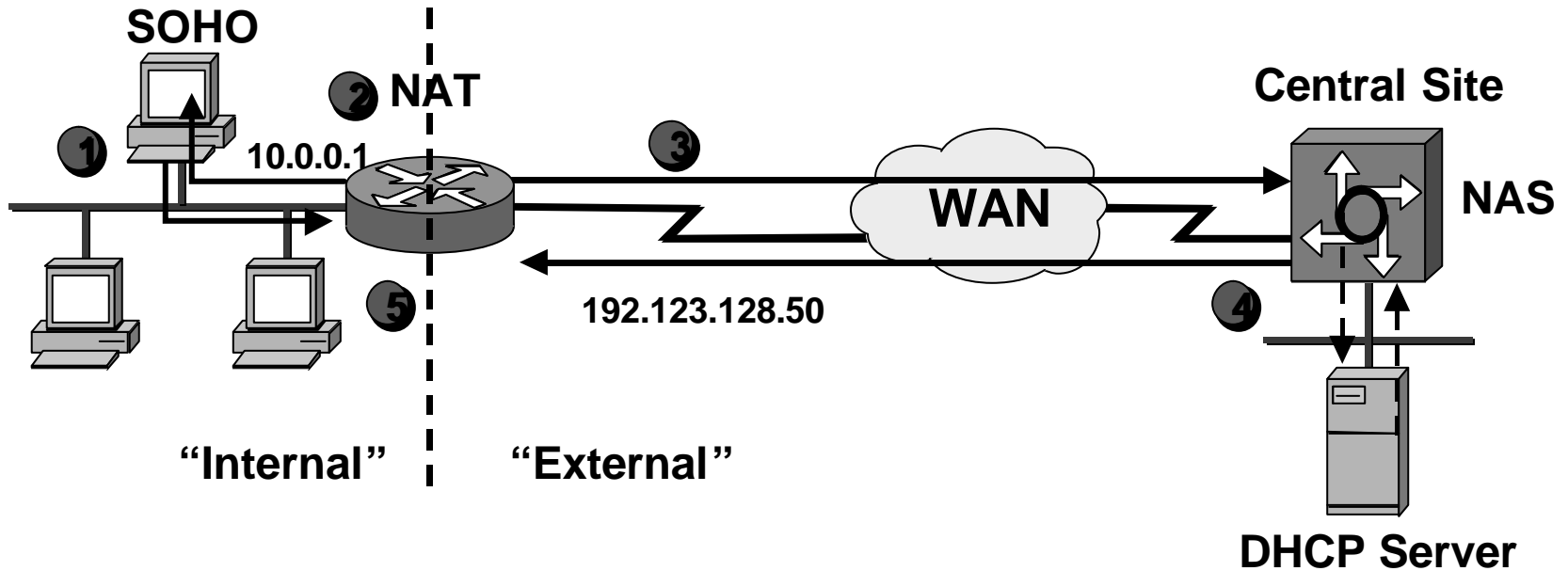
- ⇓ Quante sessioni concorrenti sono supportate dal NAT?
  - Il numero max di sessioni concorrenti supportate dipende dalla quantità di memoria disponibile sull'apparato
- ⇓ E' possibile utilizzare contemporaneamente sia traduzioni statiche che dinamiche?
  - Sì, facendo attenzione che gli indirizzi statici siano esclusi dai pool di indirizzi dinamici
- ⇓ Posso fare in modo che soltanto una parte degli indirizzi della mia rete siano tradotti attraverso il NAT?
  - Sì, configurando una access-list che include l'insieme di host/reti che richiedono la traduzione
- ⇓ Quanti indirizzi interni posso mappare su un unico indirizzo pubblico attraverso il PAT?
  - Poichè il campo port è di 16 bit, fino a 65535 indirizzi (molte implementazioni allocano soltanto le porte non privilegiate da 1024 a 65535)

# Esempio: accesso remoto con DHCP/NAT (1)



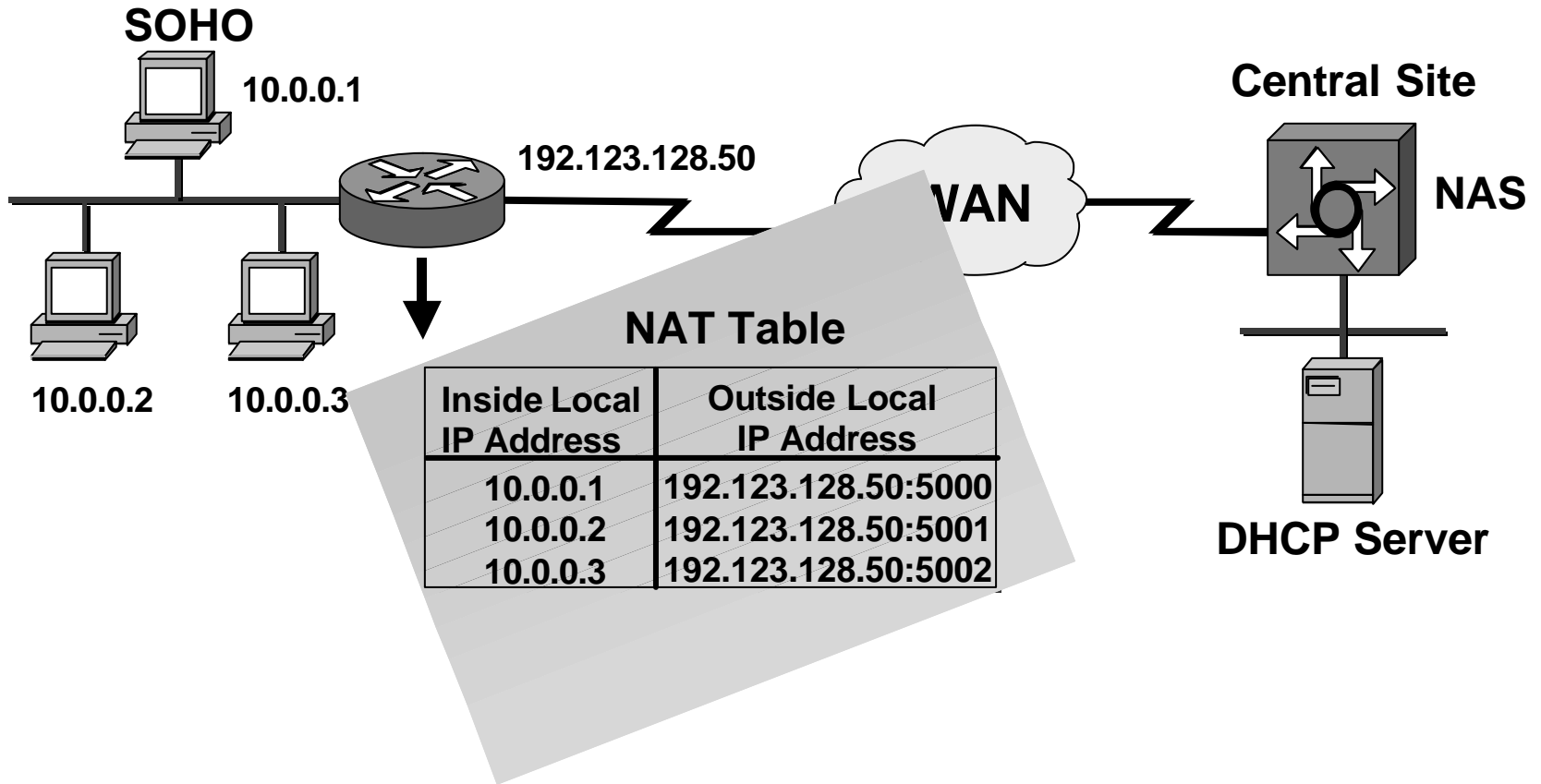
1. Un host del SOHO invia un messaggio di DHCP discover/request per ottenere un indirizzo IP
2. Il router (con funzionalità di DHCP server) risponde alla richiesta DHCP fornendo all'host un indirizzo privato

# Esempio: accesso remoto con DHCP/NAT (2)



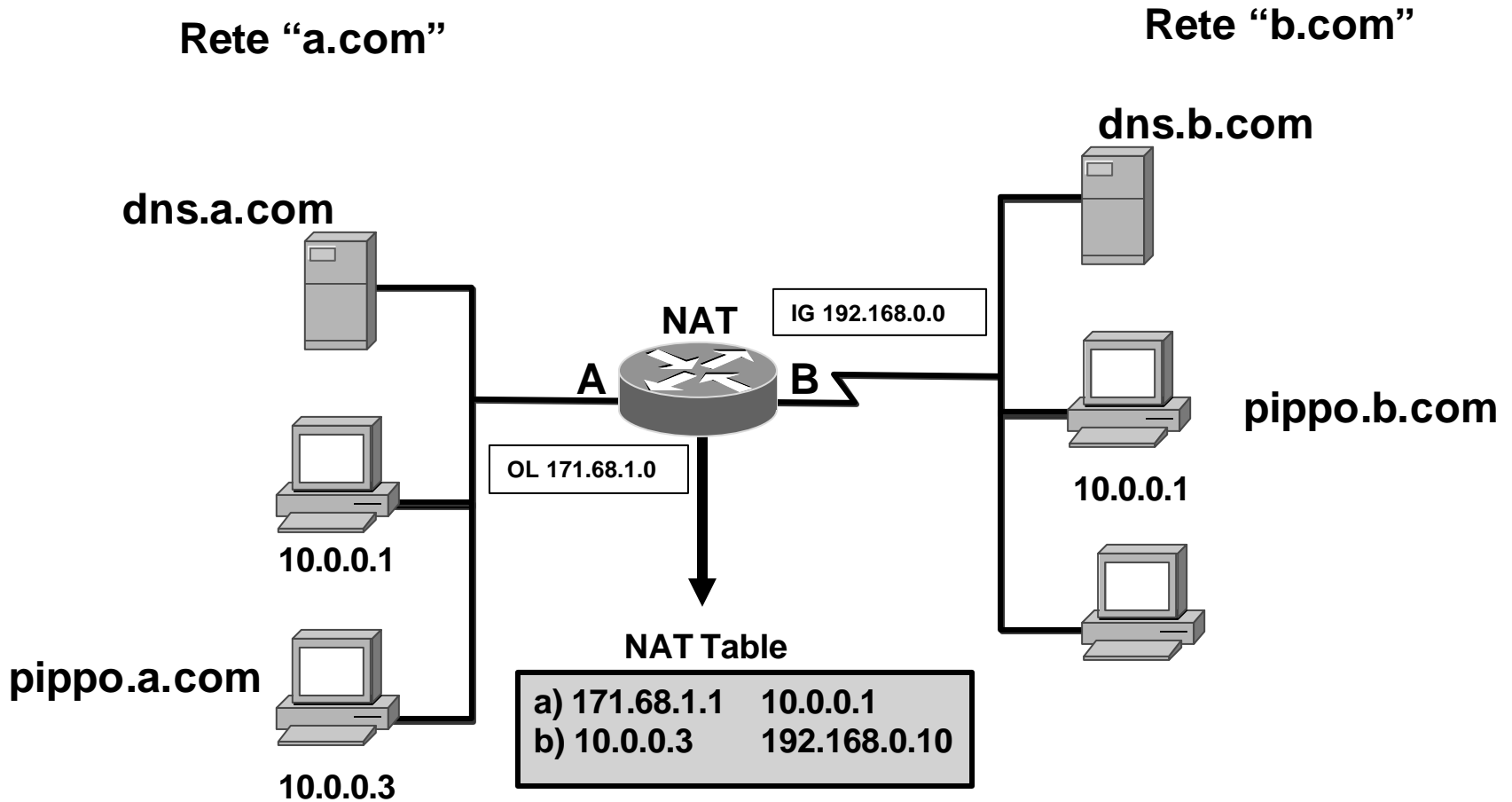
3. Quando il router riceve un pacchetto IP destinato verso la rete esterna invia una richiesta (attraverso PPP/IPCP) di un indirizzo IP pubblico alla sede centrale
4. La sede centrale (attraverso il NAS) risponde fornendo al router un indirizzo IP pubblico (prelevato dal pool locale oppure richiesto al DHCP server)
5. Il router associa tutti gli indirizzi IP interni all'indirizzo IP pubblico assegnatogli dalla sede centrale utilizzando la funzionalità di PAT

# Esempio: accesso remoto con DHCP/NAT (3)





# Esempio: non-sovrapposizione di spazi di indirizzamento mediante NAT

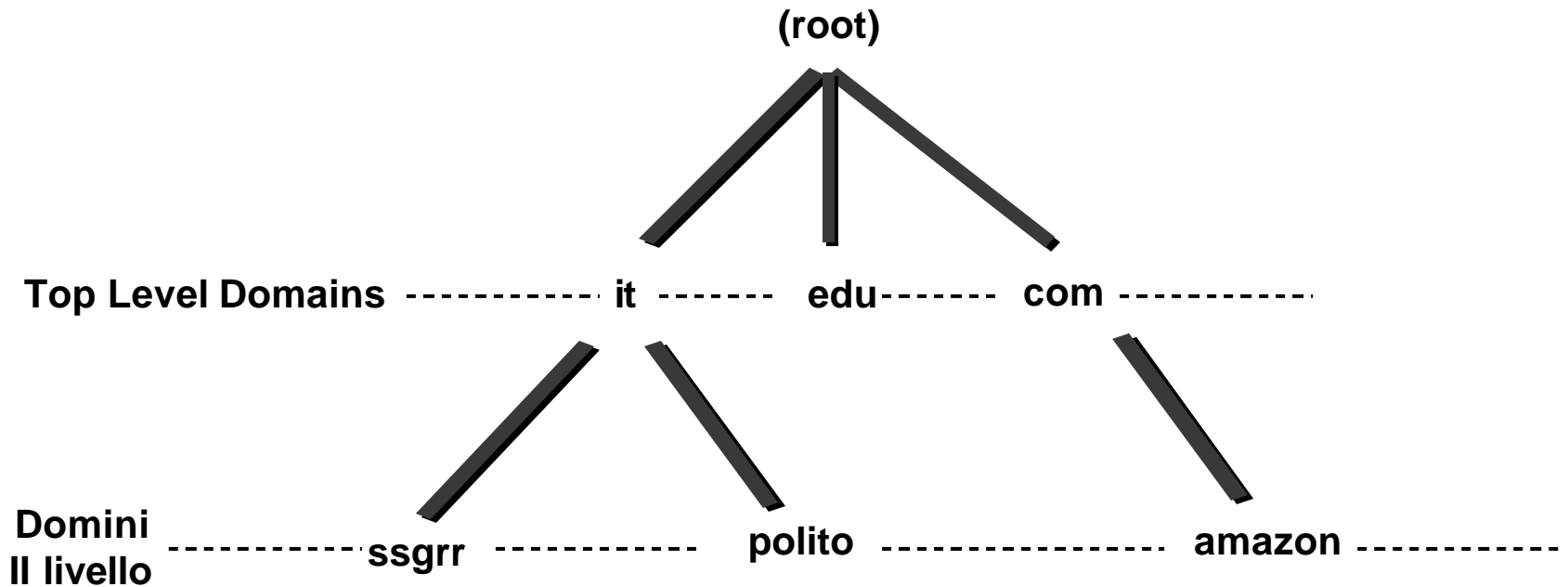


DNS

# Domain Name System (DNS)

↓ In Internet i nomi sono organizzati gerarchicamente in *domini*

- I nomi sono costituiti da stringhe separate da "."
- La parte più significativa è a destra

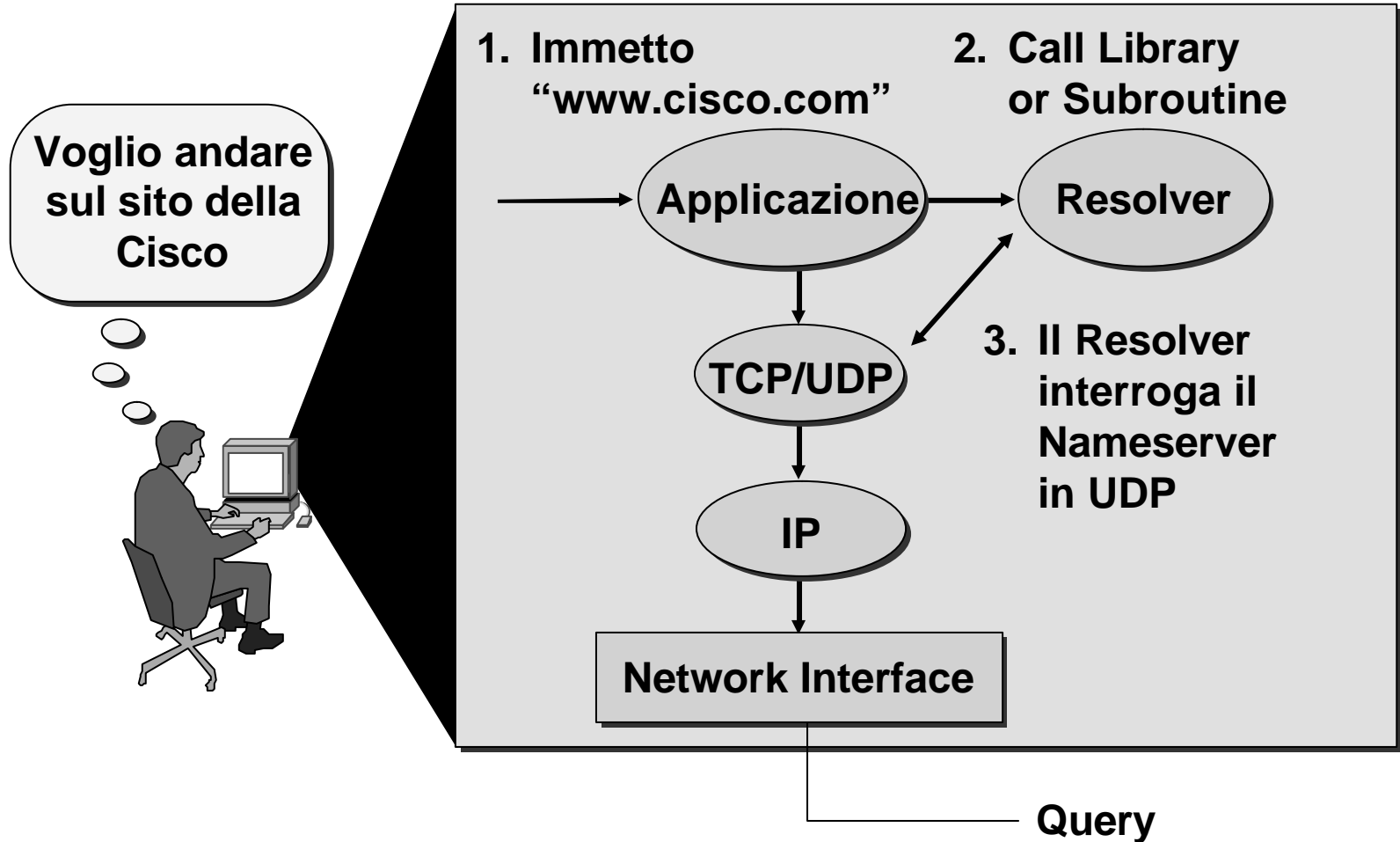


# DNS

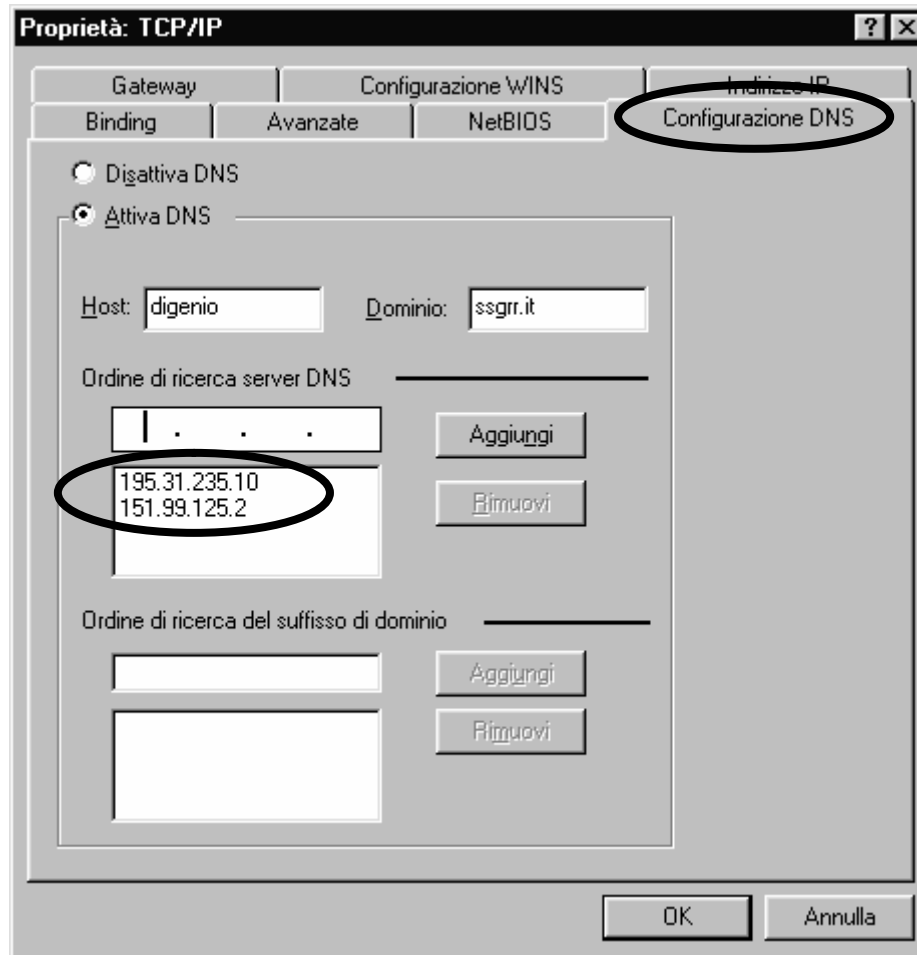
## Top level domains

com	Organizzazioni commerciali (hp.com, sun.com ...)
edu	Organizzazioni educative (berkeley.edu, purdue.edu ...)
gov	Organizzazioni governative (nasa.gov, nsf.gov ...)
mil	Organizzazioni militari (army.mil, navy.mil ...)
net	Organizzazione di gestione reti (nsf.net ...)
org	Organizzazioni non commerciali (eff.org ...)
int	Organizzazioni internazionali (nato.int ...)
<i>country-code</i>	Codice di due caratteri per indicare una nazione

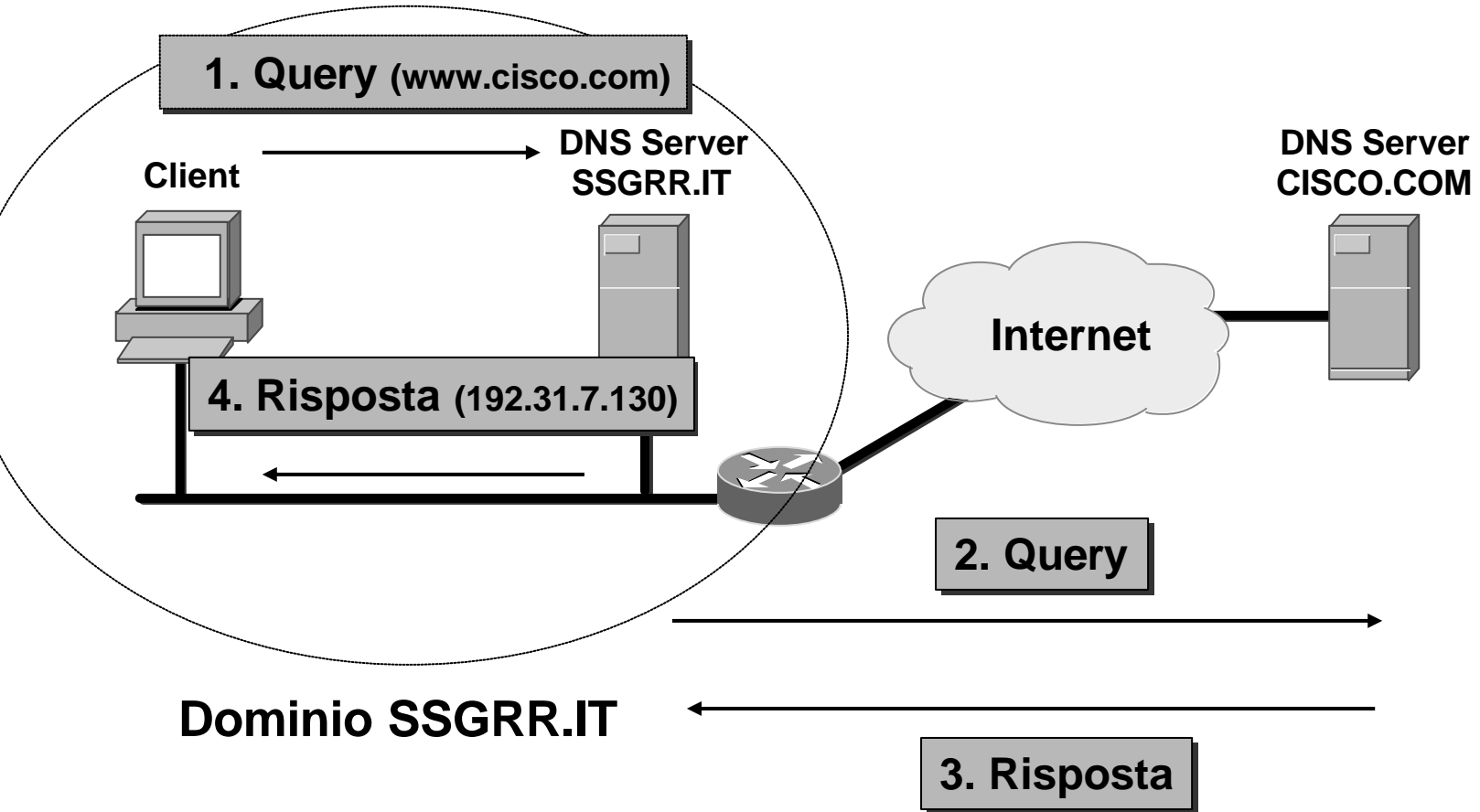
# Resolver



# Configurazione Windows 9x: DNS



# Risoluzione del nome

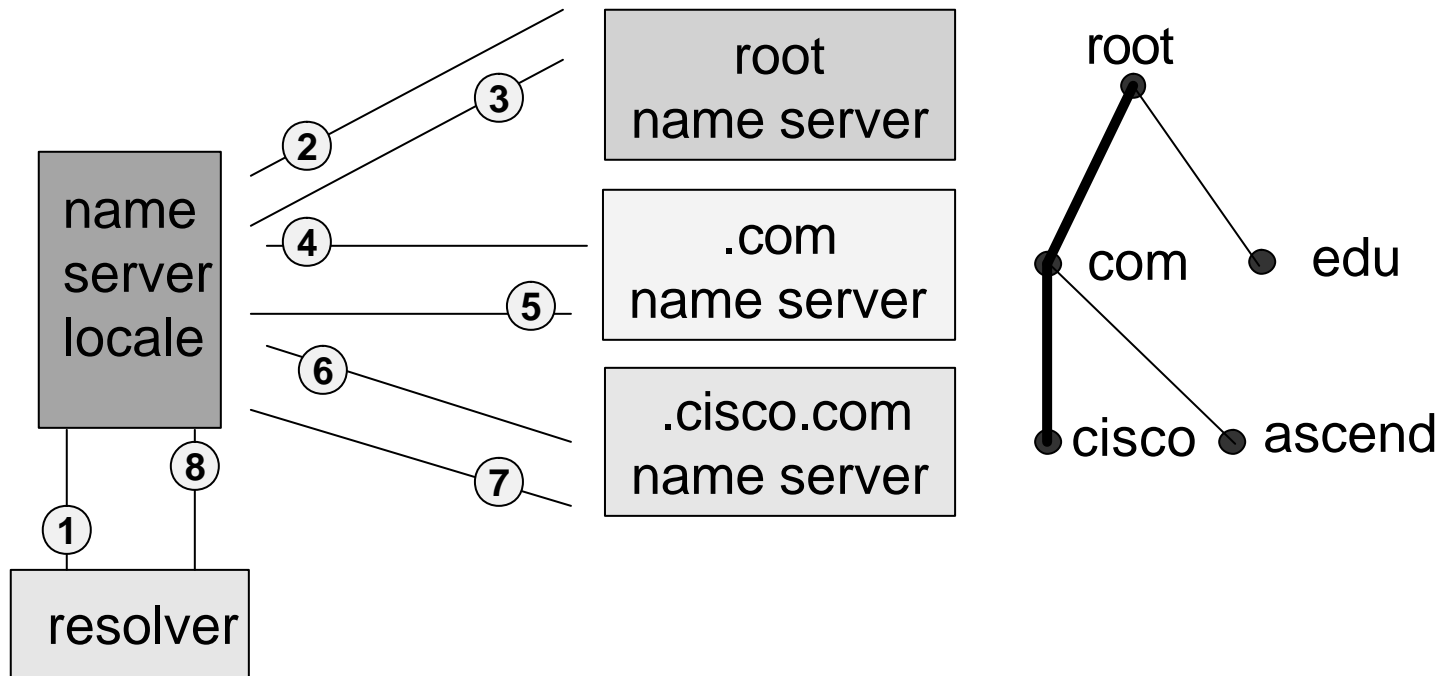


# Root Name Server

- ↓ Quando un server DNS riceve una query per un nome appartenente ad un dominio di cui non ha autorità effettua le seguenti operazioni:
- ☆ verifica nella cache se è presente il nome da risolvere. La cache contiene infatti i record dei nomi risolti più di recente
  - 🕒 invia la query ad uno dei root name server specificati in un file denominato cache file



# Esempio di risoluzione del nome

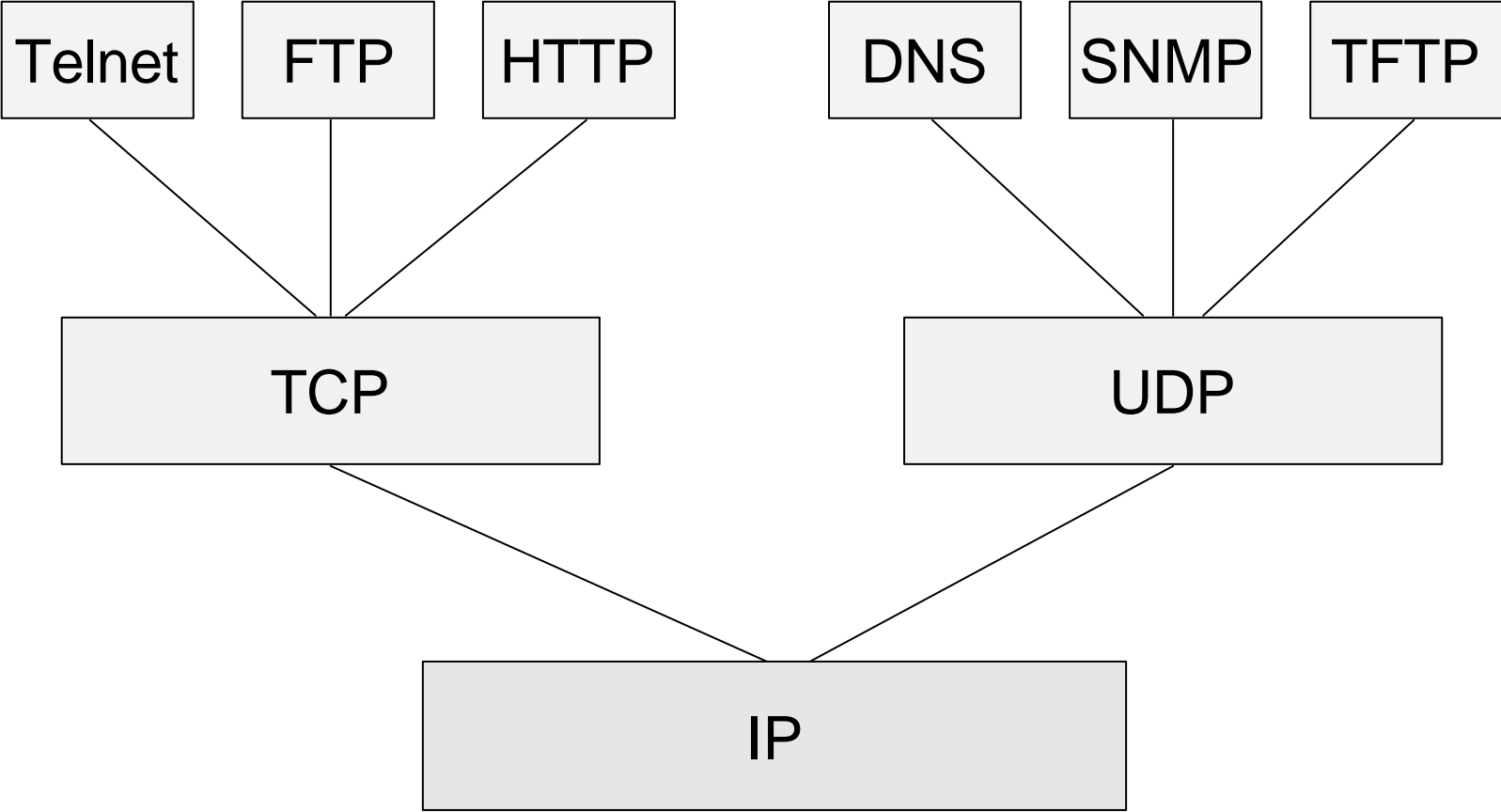


↓ Esempio:

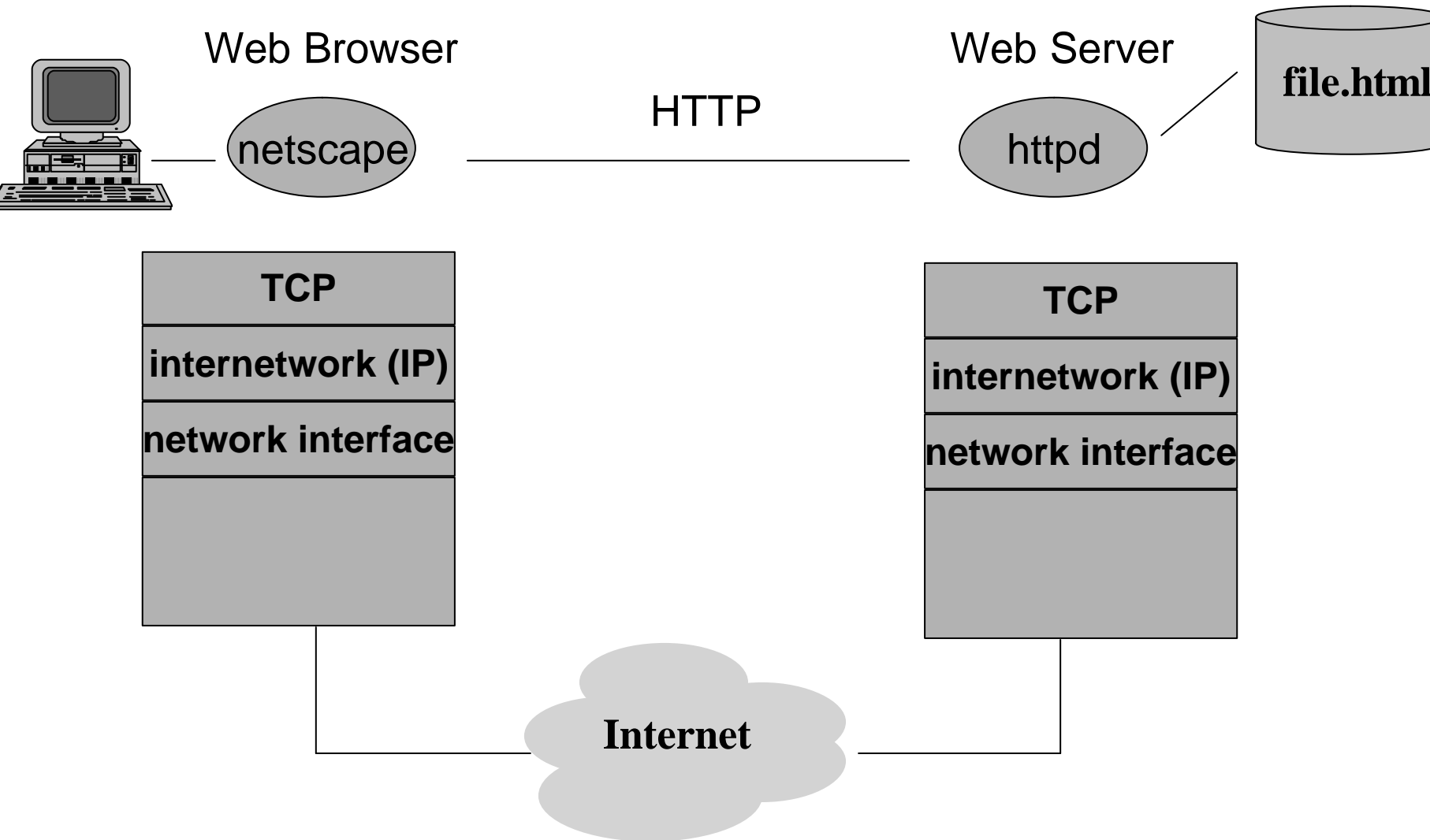
- Risoluzione del nome "www.cisco.com"

# Le applicazioni Internet

# Gli Applicativi

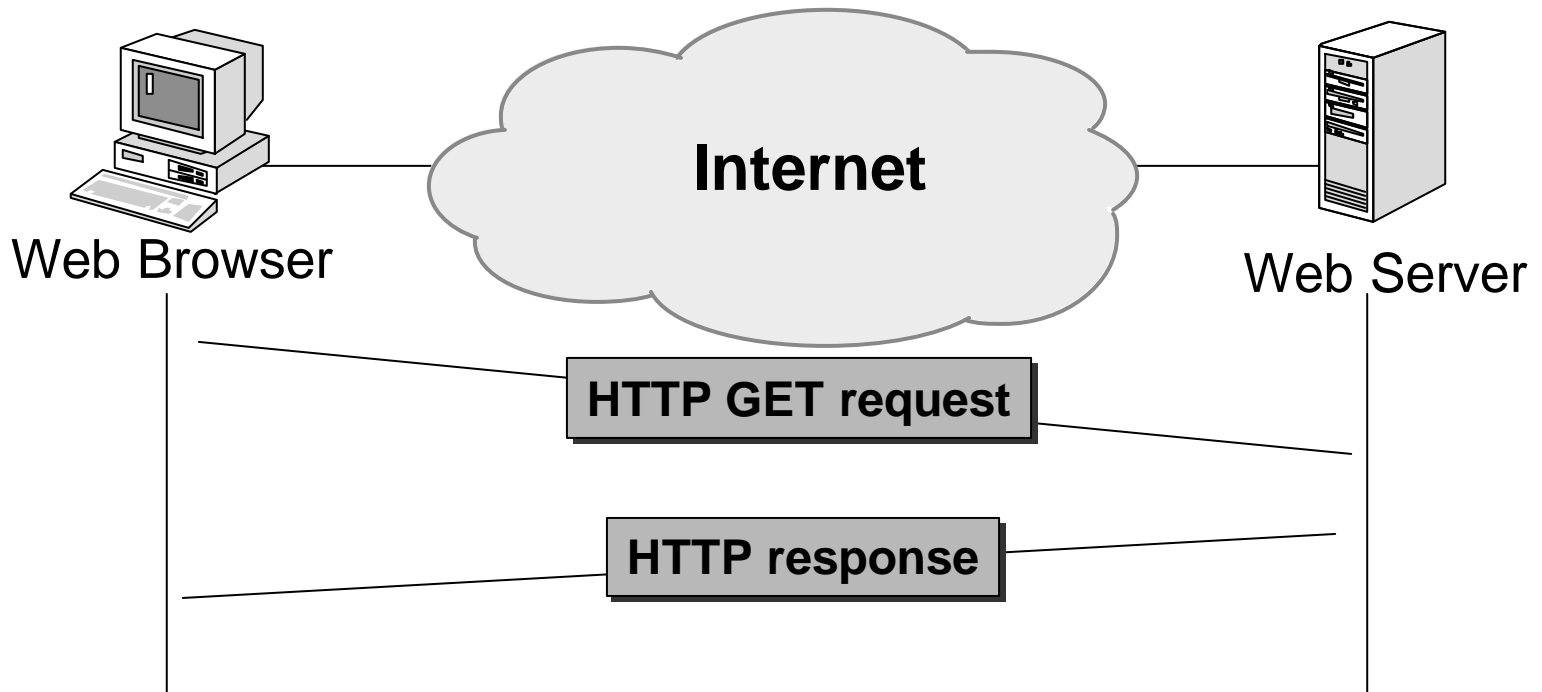


# World Wide Web: il protocollo HTTP



# Protocollo HTTP

- ↓ Il WWW è basato sul protocollo HTTP (HyperText Transfer Protocol)
- ↓ HTTP utilizza principalmente due tipi di messaggio:
  - una richiesta inviata dal client al server (GET)
  - la risposta del server



# HTTP: esempio (1)

- ↓ Un utente normalmente scrive l'URL del sito Web al quale vuole collegarsi
  - ad esempio, `http://www.w3.org/pub/WWW/TheProject.html`
- ↓ Il browser traduce tale URL in un messaggio HTTP:
  - GET `http://www.w3.org/pub/WWW/TheProject.html`  
HTTP/1.0
- ↓ Questo messaggio può essere suddiviso in tre parti:

```
Protocollo: HTTP
Server: www.w3.org
Request: GET /pub/WWW/TheProject.html HTTP/1.0
```

# HTTP: esempio (2)

↓ Il browser effettua le seguenti operazioni:

- risolve il nome `www.w3.org` in indirizzo IP (`172.16.2.3`)
- instaura una connessione TCP (port destinazione 80) con `172.16.12.3`
- invia il messaggio HTTP: `GET /pub/WWW/TheProject.html HTTP/1.0`
- riceve la risposta dal server, che gli invia il file HTML
- processa il file HTML e visualizza la pagina Web

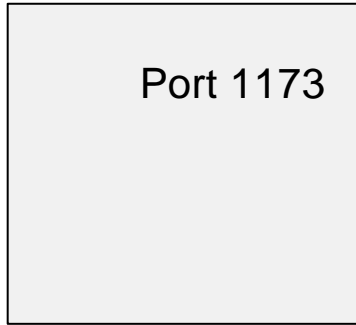
# FTP (File Transfer Protocol)

- ⇓ Il protocollo FTP si distingue dagli altri applicativi perchè utilizza due connessioni TCP per trasferire un file
  - una connessione di controllo
  - una connessione dati
- ⇓ La connessione di controllo viene instaurata dal client utilizzando la porta di destinazione 21. Questa connessione rimane in piedi per tutto il tempo che il client comunica con il server ed è utilizzata dal client per inviare i comandi e dal server per inviare le risposte
- ⇓ La connessione dati viene creata ogni volta che un file è trasferito tra il client ed il server. Questa connessione viene instaurata dal server utilizzando la porta sorgente 20



# FTP: esempio

## FTP client



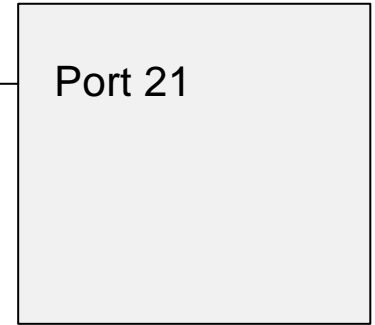
**IP: 195.31.235.5**

(control connection)

PORT 195,31,235,5,4,150

$$256 \times 4 + 150 = 1174$$

## FTP server



## FTP client



**IP: 195.31.235.5**

(control connection)

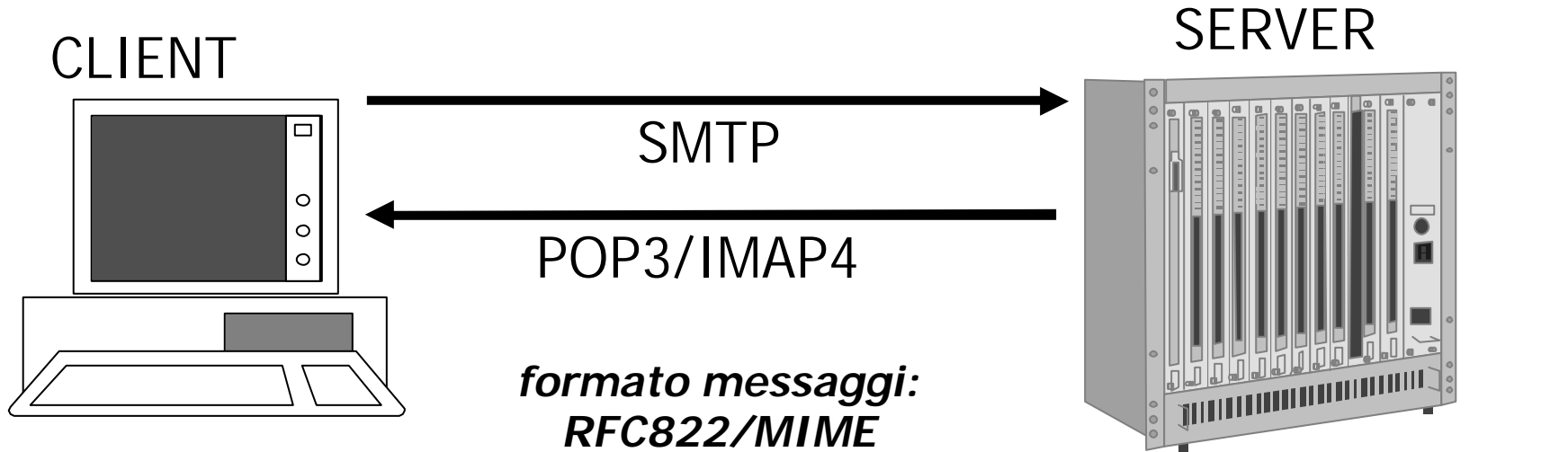
(data connection)

SYN to 195.31.235.5  
port 1174

## FTP server



# La posta elettronica (e-mail)



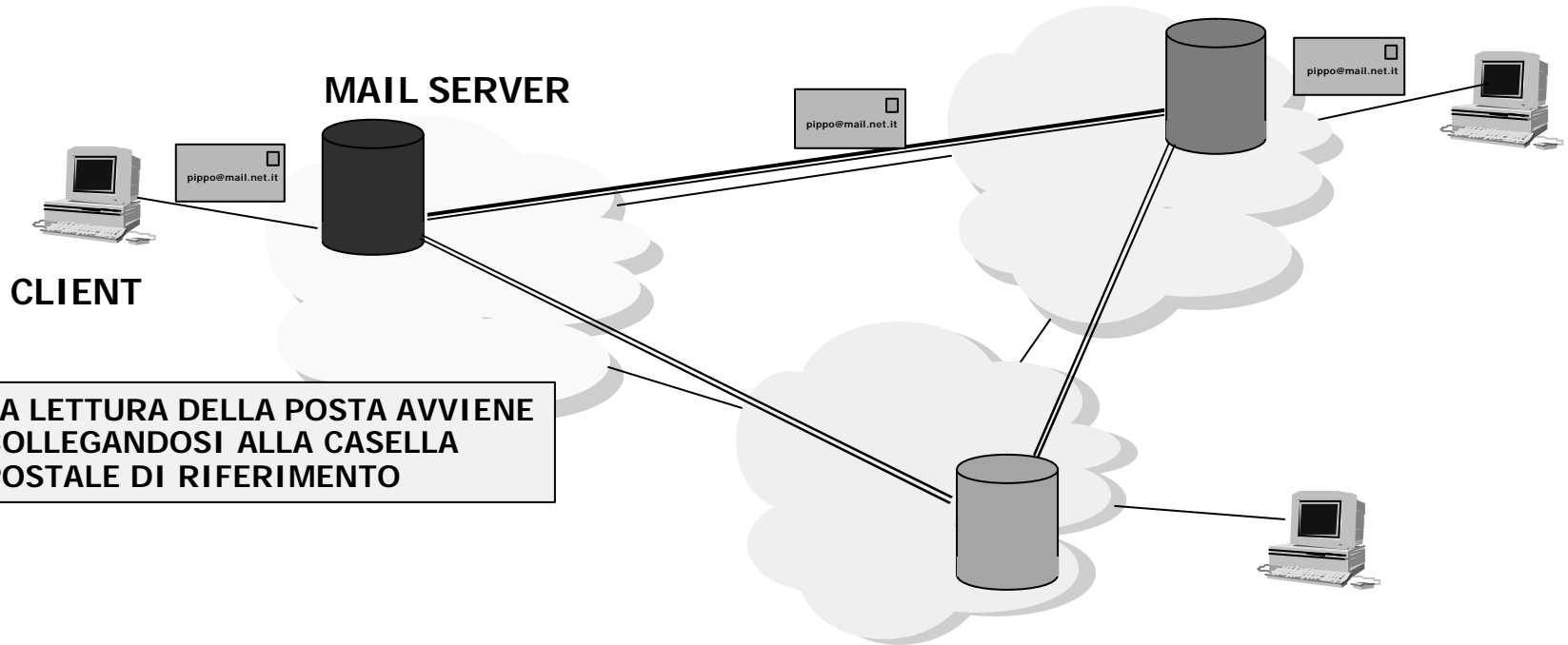
## MAIL USER AGENT (**MUA**):

- ha un'interfaccia utente per l'inserimento dei messaggi
- "conosce" il protocollo per spedire messaggi (SMTP)
- ed il protocollo per riceverli (POP3 o IMAP4)
- "conosce" come comporre i messaggi (RFC822/MIME)

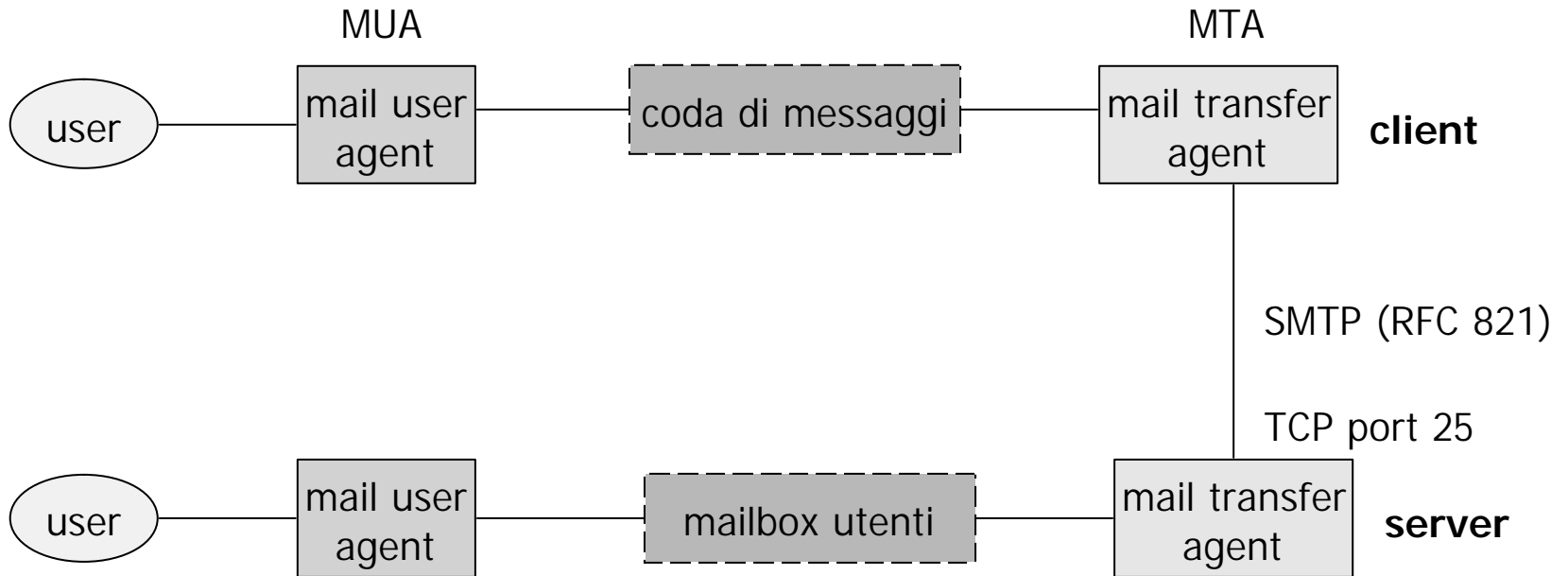
## MAIL TRANSFER AGENT (**MTA**):

- un server SMTP (porta 25): gestisce la spedizione e ricezione dei messaggi verso e da altri server SMTP
- un server POP3 (porta 110) oppure IMAP4: gestisce la spedizione dei messaggi al client

# SMTP (Simple Mail Transfer Protocol)



# SMTP (Simple Mail Transfer Protocol)



# Come ricevere la posta?

- ⇓ Ogni utente che ha accesso ad un sistema di solito ha una sua mailbox (casella di posta elettronica) che corrisponderà ad uno spazio nel file system; in quello spazio il server SMTP locale salverà i messaggi ricevuti per lo specifico utente
- ⇓ Se l'utente accede direttamente al sistema, il programma di email leggerà direttamente dal file system i messaggi
- ⇓ Altrimenti, via rete, sarà necessario un protocollo apposito per la lettura dei messaggi:
  - POP3: permette di scaricare i messaggi come tali, senza funzionalità di gestione
  - IMAP: accesso alla mailbox, i messaggi rimangono sul server e sono quindi accessibili da più sistemi

# E-Mail: SMTP e POP3

