Architetture e Protocolli

IZ3MEZ

Francesco Canova

www.iz3mez.it

francesco@iz3mez.it

Il software di rete (1/2)

Che cosa deve fare il software di rete?

Permettere che un messaggio inviato da una macchina arrivi a destinazione

- Ciò comporta la realizzazione di un software che faccia da ponte tra l'applicazione e il mezzo trasmissivo usato

- Tale software deve prendere in considerazioni aspetti sicuramente non banali, come ad esempio:
  - Indirizzamento:
    - come fa una macchina a dire con quale altra macchina vuole comunicare?
    - necessità di un meccanismo per identificare univocamente le macchine sulla rete
  - Senso di comunicazione:
    - *simplex*: senso unico
    - *half duplex*: a senso unico alternato
    - *full duplex*: a doppio senso

Il software di rete (2/2)

- Ed ancora …

  - Controllo dell'errore:
    - il mezzo trasmissivo non è mai perfetto (è affetto da *rumore*)
    - necessità di meccanismi per rilevare e, ove possibile, correggere gli errori di trasmissione

  - Controllo del flusso:
    - come impedire che un trasmittente veloce "sommerga" un ricevente lento

  - Ordinamento:
    - certi tipi di reti (es. commutazione di pacchetto) non preservano l'ordine dei pacchetti
    - necessità di meccanismi per contrassegnare i pacchetti e ricostruire il messaggio originale

  - Lunghezza dei messaggi:
    - non tutti i processi riescono a gestire messaggi lunghi
    - dividere, trasmettere ed unire i pacchetti

In principio … il software monolitico

- In origine il software di rete era *monolitico*
  - all'interno dello stesso programma si trattava tutto, dall'interazione con l'utente fino alla scelta del voltaggio
- Funziona, ma cosa succede se si cambia ad esempio:
  - il tipo di cavo?
    - con ogni probabilità cambia il voltaggio, il tipo di segnalazione ecc.
  - l'interfaccia con l'utente?
    - ad esempio, si passa da interfaccia a caratteri a interfaccia grafica
  - la tecnologia trasmissiva?
    - ad esempio, si passa da broadcast a point-to-point
- Risposta:
  - occorre manutenere pesantemente il software

Soluzione: architettura a livelli

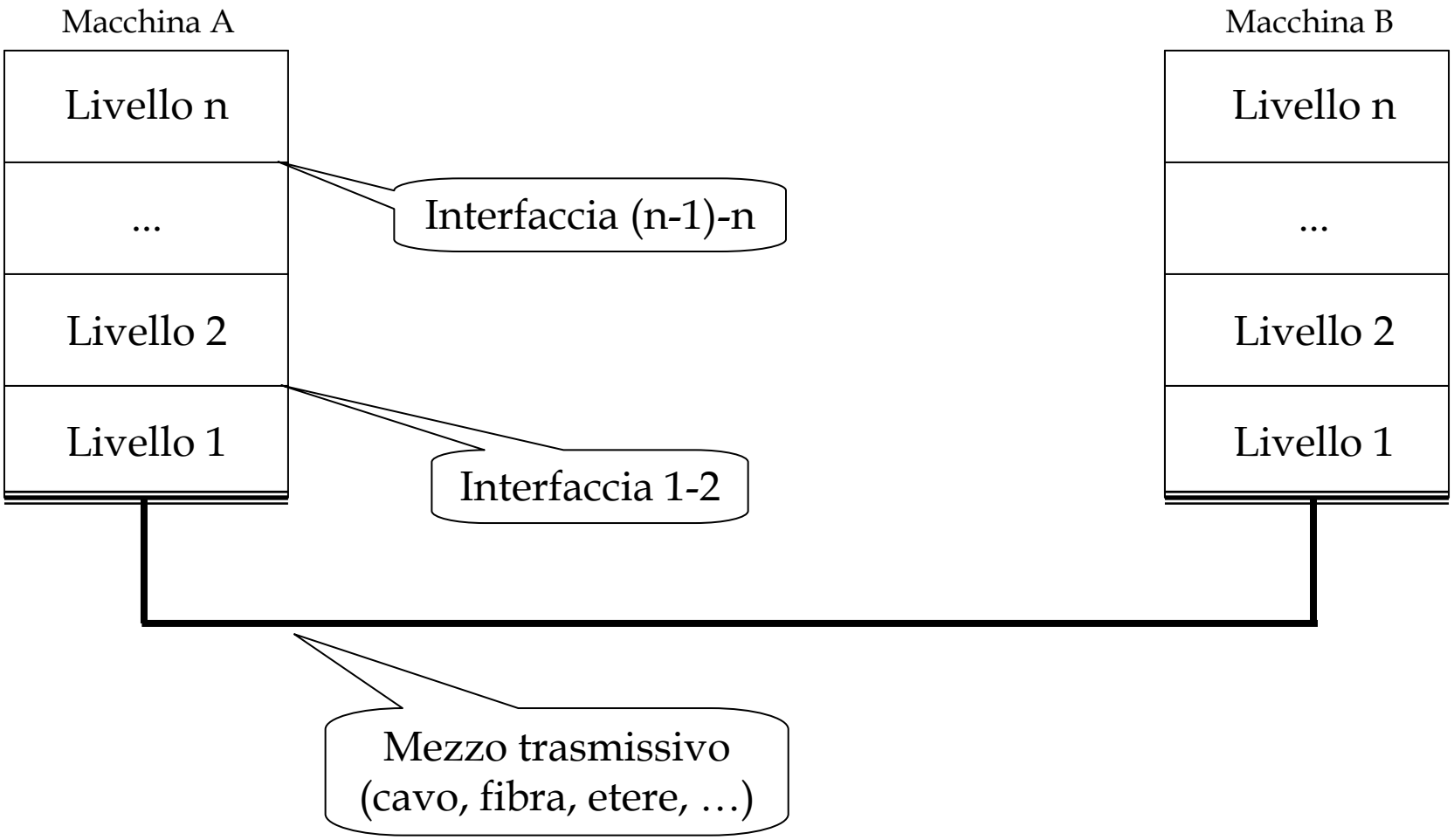
L’architettura a livelli (1/2)

- Il software di rete è suddiviso in moduli separati, detti **livelli** o **strati**

- Ogni livello ha un compito:
  - preciso (stabilito una volta per tutte)
  - costante (non cambia mai)
  - unico (si occupa di una sola cosa)
  - univoco (due livelli diversi hanno sempre compiti diversi)

- Ogni livello comunica solo con i due livelli adiacenti (superiore e inferiore)
  - fornisce un servizio al livello superiore e al livello inferiore nascondendo i dettagli sul come il servizio è implementato; per far ciò, utilizza il servizio fornito dal livello inferiore e dal livello superiore
  - offre una ben precisa **interfaccia** verso il livello superiore,
  - richiede una ben precisa **interfaccia** al livello inferiore
- L’interfaccia specifica quali servizi un livello offre al livello superiore e quali altri può richiede al suo inferiore

L’architettura a livelli (2/2)



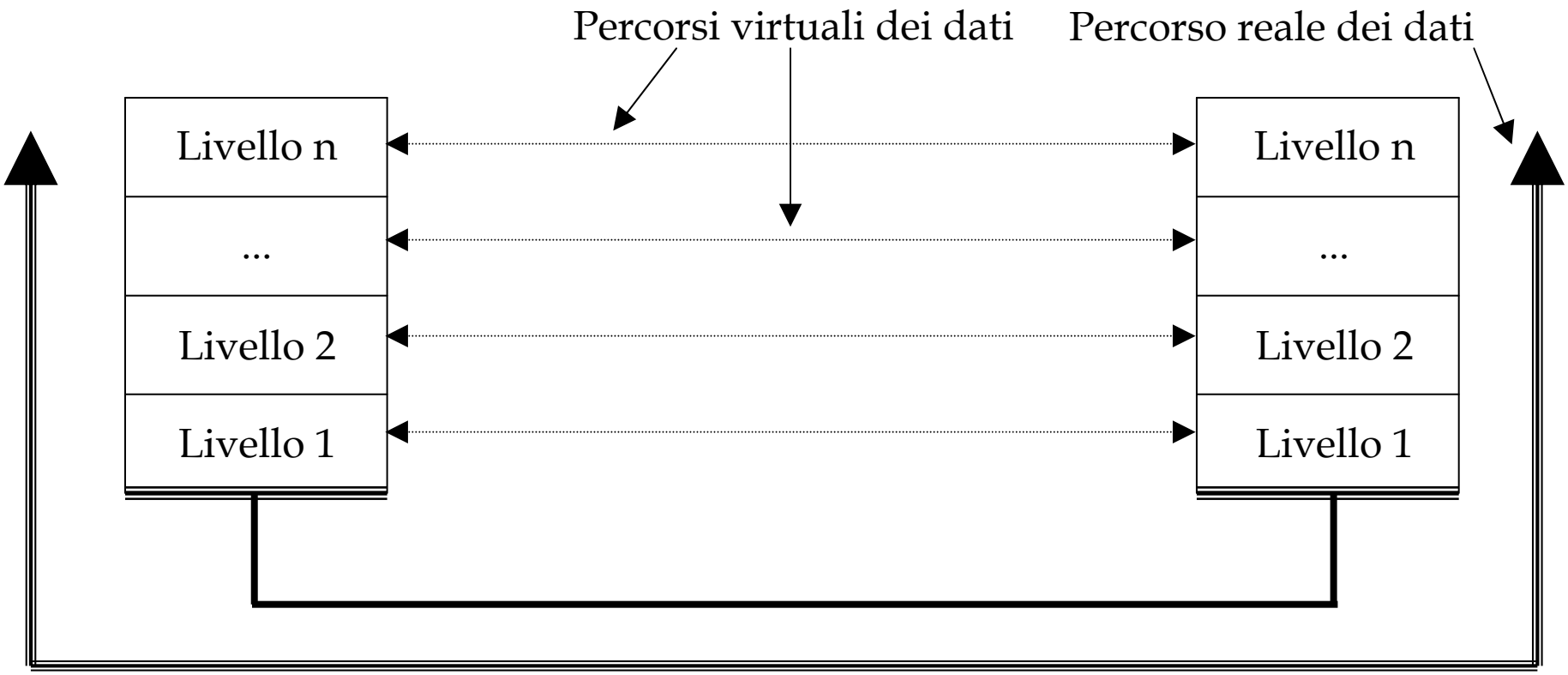
Interfacce

- Un’interfaccia chiara fra due livelli significa:
  - un livello può cambiare liberamente la sua struttura interna purché mantenga le interfacce intatte
  - un livello può essere *sostituito* con un altro che presenti le stesse interfacce

- Un esempio:
  - in una rete esistente si vuol sostituire il mezzo trasmissivo da doppino in rame a fibra ottica
  - è sufficiente sostituire il livello 1, che comunica con l’hardware, e *il resto del sistema non si accorgerà nemmeno del cambiamento*
  - purché ovviamente il nuovo livello fornisca verso l’alto la stessa interfaccia del precedente

La comunicazione fra i livelli

- In un sistema a livelli, ogni livello ha l'*illusione* di comunicare con il suo **pari** (**peer**) della controparte secondo certe regole prestabilite
- In realtà, i dati viaggiano solo verso il livello inferiore e, infine, attraverso il mezzo fisico



I protocolli

- Un **protocollo** è un insieme di regole e convenzioni che governano la comunicazione lungo i percorsi virtuali tra i *peer*
  - si parla quindi di *protocollo di livello 1, 2, ...*
- Il dialogo tra due *peer* di livello *n* viene materialmente realizzato tramite i servizi offerti dal livello *n-1*

- Perché i percorsi virtuali?
- I percorsi virtuali semplificano i protocolli
  - si può prescindere dai dettagli dei livelli inferiori
- L'insieme dei protocolli viene chiamato **pila** (o **stack**) **di protocolli**

I protocolli e i livelli: un’analogia semiseria (1/2)

- John vuole chiedere la mano di Julie, però:
  - John parla solo inglese, mentre Julie parla solo francese
    - (dunque, pur essendo compatibili a livello hardware, non lo sono a livello software)
  - John vive a Timbuctu, Julie a Manaus (niente telefono)

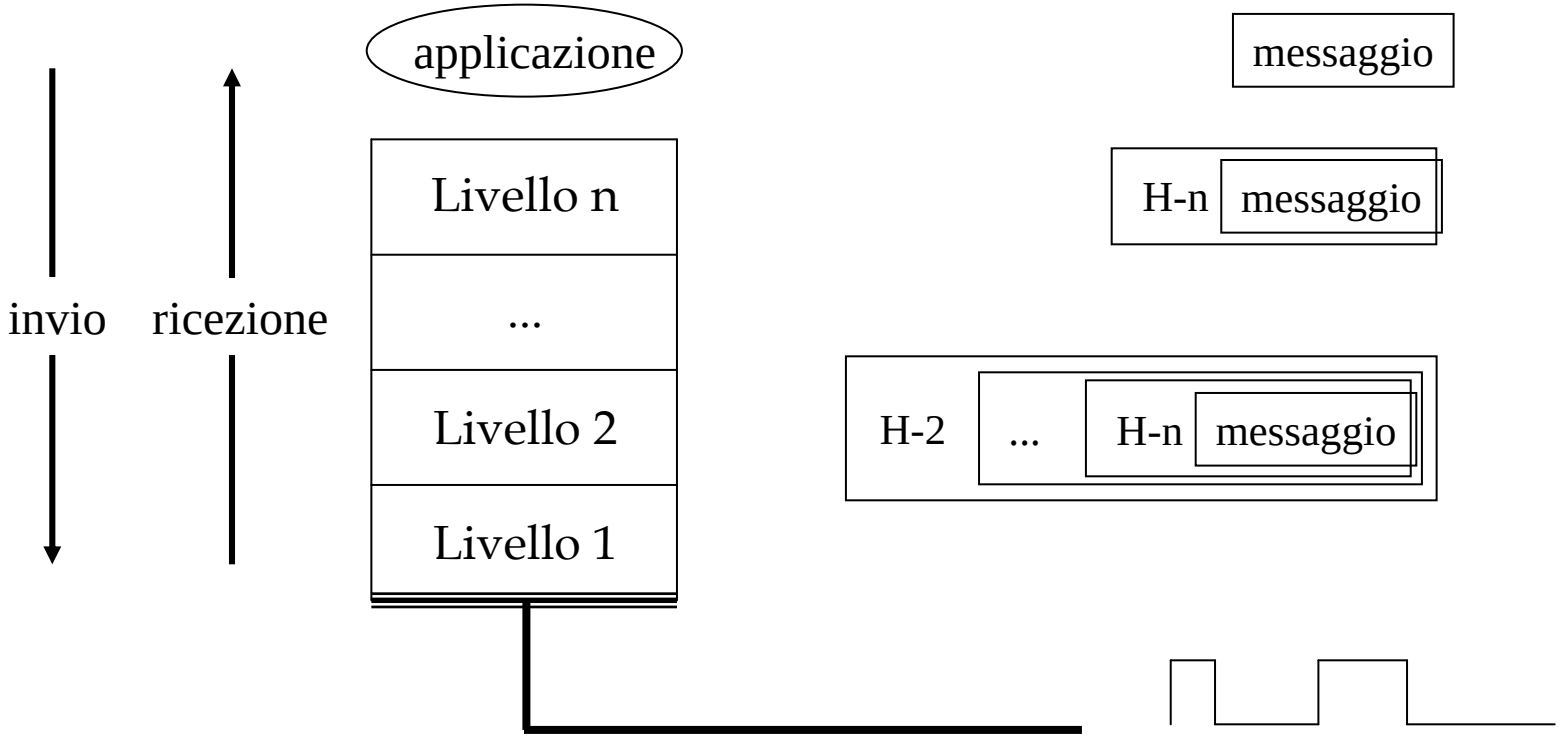
- fortunatamente però:
  - Beth, amica di John, parla inglese e italiano
  - Michel, amico di Julie, parla francese e italiano
  - esiste un servizio di posta fra le due città (il mezzo trasmissivo)

- Le azioni da fare:
  - John deve formulare un insieme di frasi secondo certe regole e convenzioni, ovvero secondo un certo *protocollo*
  - John detta la lettera a Beth che la traduce in italiano (viva la riservatezza)
  - la lettera viene spedita a Manaus
  - Michel la legge e la traduce in francese a Julie
  - il resto riguarda solo loro due...

I protocolli e i livelli: un'analogia semiseria (2/2)

Quello che succede: gli incapsulamenti

- Invio di un messaggio
  - Successivi incapsulamenti del messaggio utilizzando i protocolli del livello di pertinenza
- Ricezione di un messaggio
  - Successivi spacchettamenti del messaggio utilizzando i protocolli del livello di pertinenza



I modelli di riferimento

- Vari modelli di riferimento sono stati proposti negli anni

- Presentano alcune caratteristiche comuni …
  - sono tutti strutturati a livelli
  - tutti supportano i concetti di protocollo e interfaccia

- … e moltissime differenze:
  - il numero dei livelli
  - le funzioni di ogni livello
  - alcuni modelli si occupano di tutto, altri lasciano aperti molti problemi
  - alcuni hanno successo, altri no!
  - …

Il modello OSI (1/2)

- L'ISO (I*nternational Standards* O*rganization*) ha emanato il modello di riferimento mondiale OSI (O*pen Systems* I*nterconnection*)

- Ha lo scopo di fornire:
  - uno standard per la connessione di sistemi aperti
  - un modello rispetto a cui confrontare le varie architetture di rete

- Il modello OSI non è un'architettura di rete:
  - specifica solo cosa devono fare i vari livelli
  - ma non definisce nè i servizi e nè i protocolli

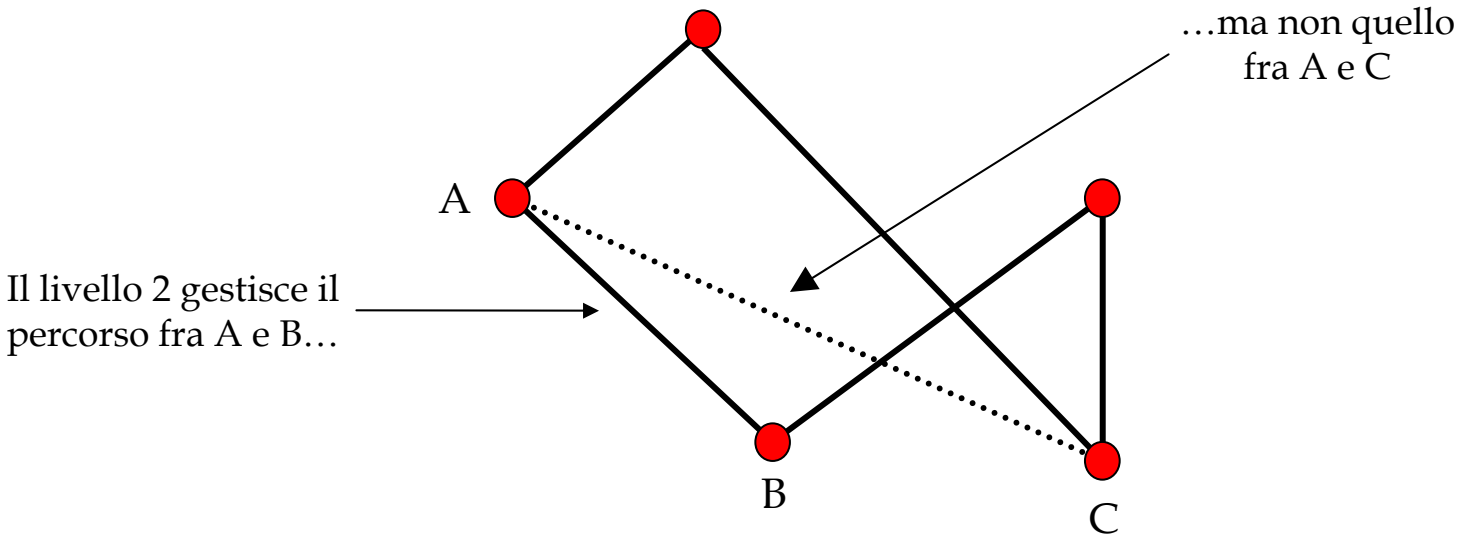
- La definizione dei protocolli venne più tardi (troppo!)

Il modello OSI (2/2)

Scopo: fornire una serie di protocolli da utilizzare per poter usufruire di determinati servizi	**Application**	7
Scopo:convertire i dati in un formato standard prima della comunicazione	**Presentation**	6
Scopo: controllare il dialogo tra i nodi della rete	**Session**	5
Scopo: accettare dati dal livello superiore, dividerli in unità più piccole (se necessario), passarle al livello network e assicurare che tutti i “pezzi” arrivino	**Transport**	4
Scopo: permettere la comunicazione fra macchine non direttamente connesse Introduzione di un sistema di indirizzamento unico a livello della rete La comunicazione avviene grazie a routing e commutazione	**Network**	3
Scopi: 1) occuparsi della trasmissione fra macchine connesse direttamente 2) far sì che il mezzo trasmissivo appaia al livello superiore come una linea di comunicazione priva di errori non rilevati	**Data link**	2
Scopo: inviare bit e riceve bit comunicando direttamente con i vari tipi di infrastrutture e dispositivi di comunicazione	**Physical**	1

Please **D**o **N**ot **T**ouch **S**teve’s **P**et **A**lligator

Livello 2: Cosa realmente gestisce

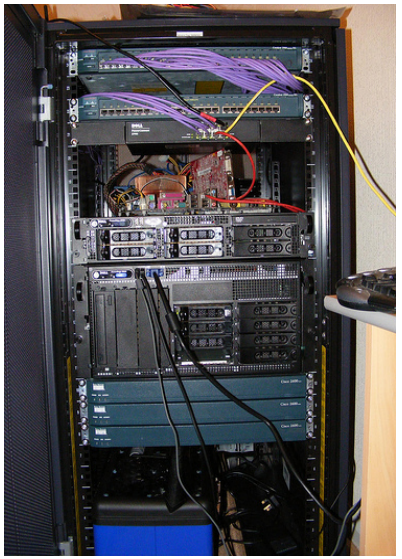


I protocolli di livello 2 utilizzano un proprio sistema di indirizzamento
L'indirizzo MAC (Media Access Control) è un esempio

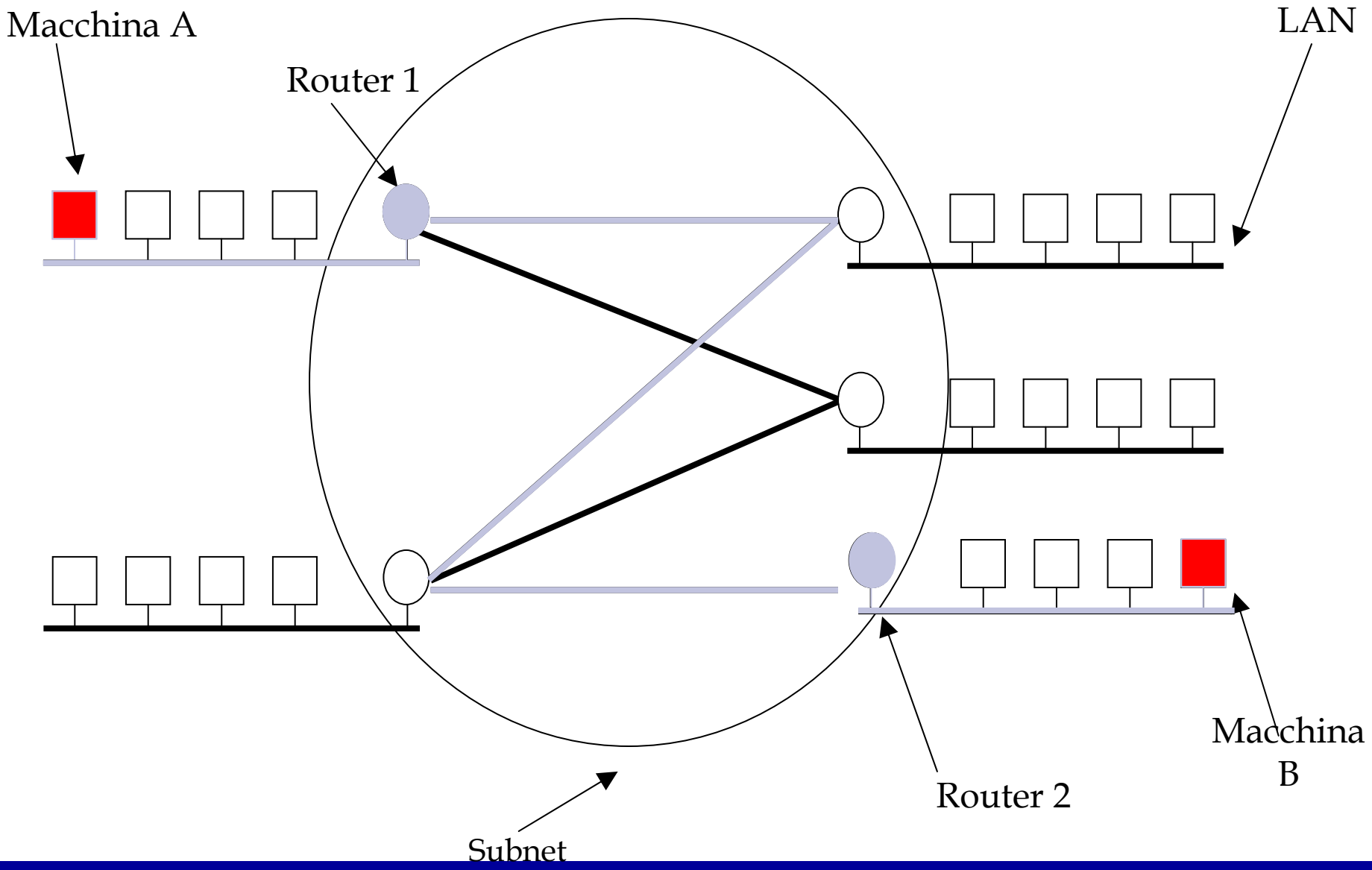
Livello 3: I Router

Un **router** (*instradatore*) è un dispositivo di rete che
si occupa di instradare pacchetti tra reti diverse ed eterogenee

- Per individuare la destinazione utilizzano un indirizzo denominato software o di livello 3 (es. IP address)
- L'algoritmo di instradamento che applicano (algoritmo di routing) può essere:
  - non adattivo (o statico) fondato su tabelle di routing che specificano, una volta per tutte, il percorso da A a B
  - adattivo (o dinamico) cioà cambia a seconda della topologia e del carico della rete
- Gli elementi della tabella di instradamento non sono singoli calcolatori ma reti locali
  - questo permette di interconnettere grandi reti, senza crescite incontrollabili della tabella di instradamento



Livello 3: funzionamento



Il mapping

- Non è detto che ogni software di rete “copra” tutti i livelli del modello ISO/OSI

- Ma ciò è un vantaggio!
  - posso “costruire” la mia rete con il software che voglio
  - l’importante è che su una macchina vi sia la pila completa!

- I software di rete che realizzano reti LAN coprono tutte i livelli bassi della pila
  - vantaggio: costruisco la mia LAN come meglio credo
  - e cosa metto sopra?
    - generalmente il TCP/IP

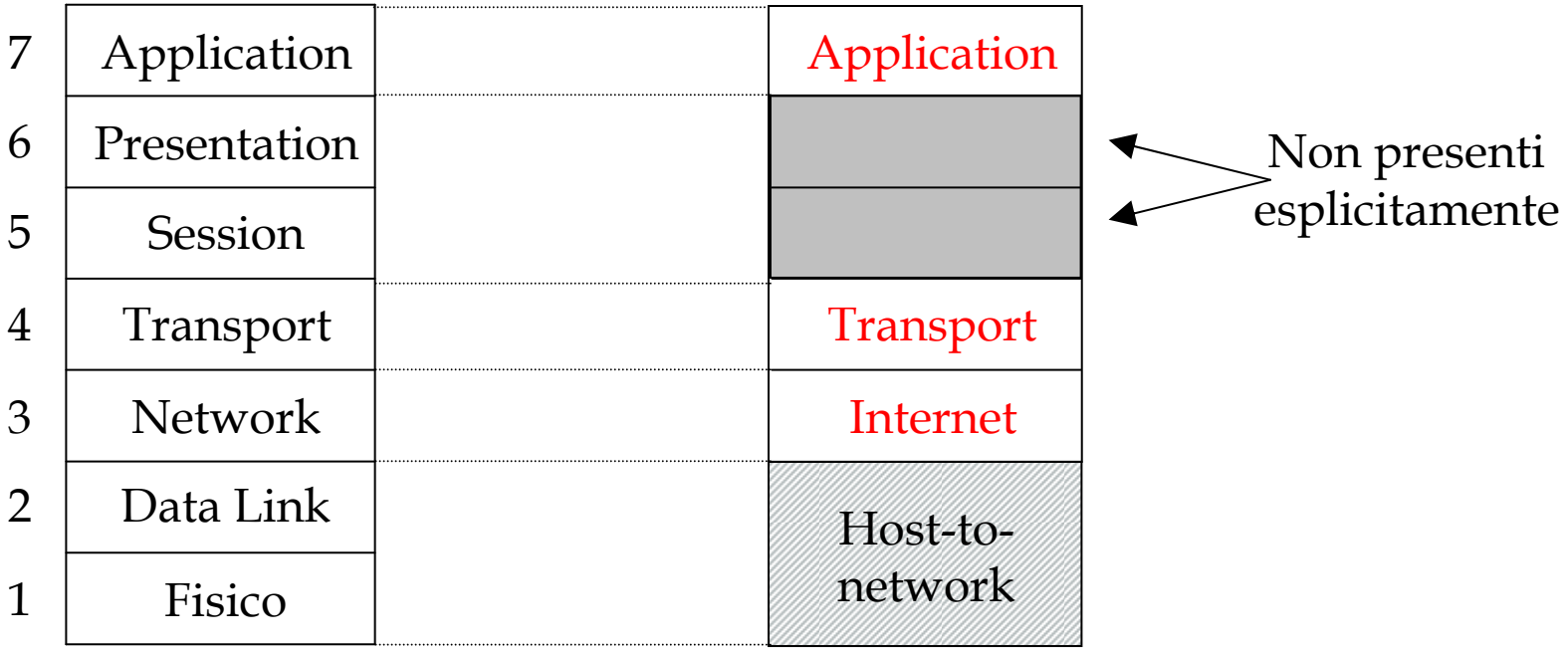
L’architettura TCP/IP

- Lo scopo con cui è nata Internet:
  - assicurare l’integrazione di reti fortemente eterogenee
    - una internet ante litteram
  - garantire la connessione anche a seguito della caduta di una linea o di una macchina
    - progetto del ministero della difesa!
  - architettura flessibile
    - esigenze applicative disparate, da trasmissione file a trasmissione parlato

- Che ha portato alla definizione del software di rete più conosciuto al mondo: **Internet Protocol Suite**, **architettura TCP/IP**
  - dal nome dei suoi 2 protocolli principali

- I protocolli sono specificati per mezzo di documenti detti *RFC* (*Request For Comments*)

Che relazione con il modello ISO/OSI?



Il livello Host-to-network

- Non è coperto dallo standard TCP/IP
  - è lasciato a una “pila bassa” sottostante
- Il motivo è proprio per lasciare libertà nella configurazione della propria rete LAN

- È per questo che TCP/IP gira su moltissime tecnologie diverse
  - Ethernet
  - Token Ring
  - FDDI
  - ATM
  - PPP
  - Frame Relay
  - …
- Ciò che viene richiesto è che l’*host* sia in grado di ricevere pacchetti IP ed effettuare il *framing*

Il livello Internet

È lo strato unificante, che tiene insieme tutta l'architettura
Scopo: permettere ad un host di immettere **pacchetti** su una qualsiasi rete e fare il possibile per farli viaggiare fino a destinazione

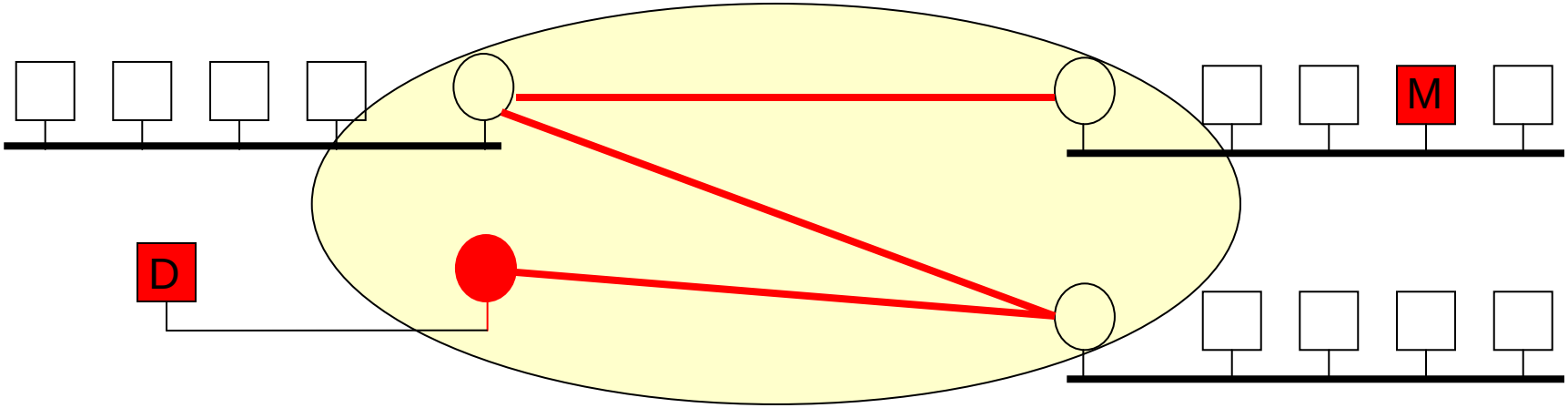
- La richiesta di estrema affidabilità e tolleranza ai guasti ha portato a scegliere una rete **packet-switched connectionless**
  - ciò garantisce il colloquio della macchina sorgente con la macchina destinazione anche in situazioni di "perdita" di parte della subnet
    - connectionless:
      - i dati possono non arrivare nello stesso ordine in cui sono stati spediti
    - connection-oriented:
      - i dati arrivano nello stesso ordine in cui sono stati spediti

Tipiche questioni da risolvere

- Tipiche questioni da risolvere a questo livello:
  - **routing**: determinare il cammino che deve compiere un pacchetto scegliendo tra quelli disponibili il più conveniente
  - **commutazione**: invio del messaggio nel canale scelto
  - **congestione**: se troppi pacchetti sono presenti nella *subnet* allo stesso tempo si intralciano l'un l'altro
  - **accounting**: difficile se i pacchetti attraversano diverse nazioni
  - **conversione di dati** nel passaggio da una rete ad un'altra
    - esempio: indirizzi diversi, pacchetti da frammentare, protocolli diversi

Il protocollo IP

- I protocolli dello strato data link non hanno visione d'insieme dell'intera rete
  - come detto si preoccupano della comunicazione fra macchine direttamente connesse
- Il protocollo **IP** (**Internet Protocol**) è colui che si preoccupa di far recapitare il pacchetto al nodo (router) che è collegato alla rete a cui appartiene la macchina destinataria del pacchetto



- Come individua la macchina? **IP (Internet Protocol) address**

IP Address

- L’indirizzo IP è un numero che identifica in maniera univoca un dispositivo all’interno di una rete
- L'Internet Protocol (IP) utilizza due versioni di indirizzamento IP:
  - IPv4 e IPv6
  - quella a cui ci si riferisce quando si parla di "indirizzo IP" è la IPv4, quella più comunemente utilizzata oggigiorno.

- Gli indirizzi IP (IPv4) sono espressi con i valori decimali corrispondenti separati da un punto
  - notazione “dotted quad”, ovvero quattro numeri separati da punti
  - esempio: **149.132.110.32**
- Tale indirizzo è **univoco**:
  - in Internet non esistono due host con lo stesso *IP Address*
  - identifica un host in Internet in maniera univoca

Indirizzi IP: caratteristiche

- L'indirizzo IP è gerarchico (o strutturato)
  - come i numeri di telefono: prefisso internazionale + prefisso teleselettivo + numero
  - se così non fosse ogni router dovrebbe aver memoria della collocazione di ogni singola macchina
- Sono state progettate 5 classi per distinguere le tipologie di reti connesse ad Internet:
  - grande, media, piccola, indirizzi per applicazioni multicast e indirizzi per usi futuri
- Un indirizzo IP è strutturato almeno in due parti:
  - **Network Address**: l'indirizzo della rete, unico in Internet
  - **Host Address**: l'identificativo di uno specifica macchina all'interno della rete, all'interno di una particolare rete è unico

- I primi bit indicano a quale classe un indirizzo IP appartiene

Indirizzi IP: le classi

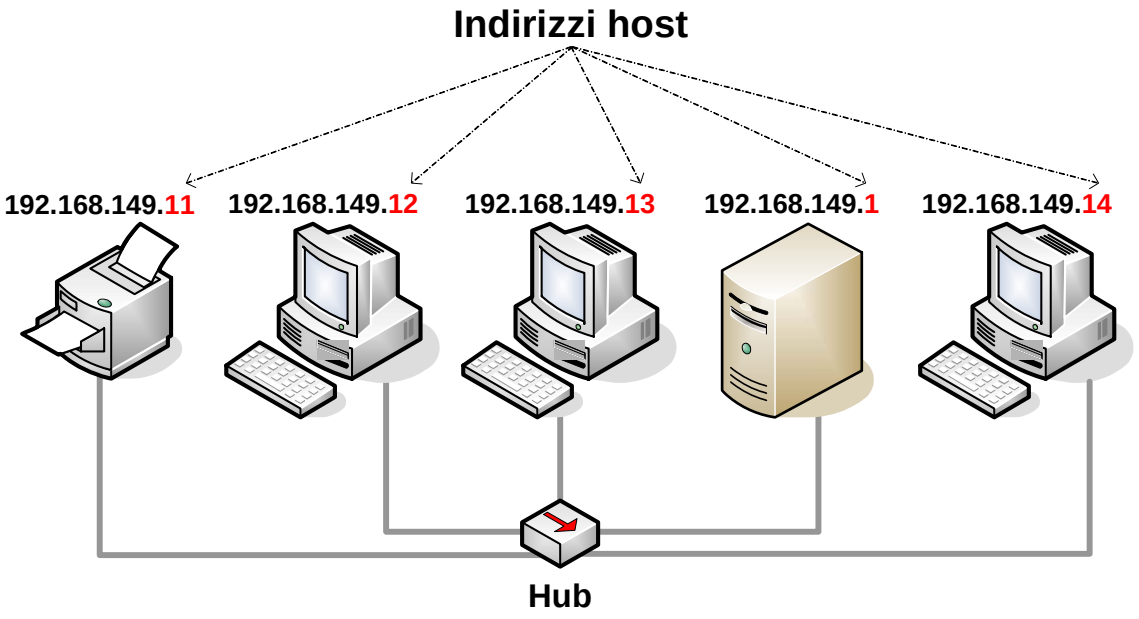
Classe	Utilizzo	Indirizzo	Struttura	Esempio
A	reti di grandi dimensioni (128 reti con ciascuna al massimo 16,7 milioni di macchine connesse)	0-127 . *x* . *x* . *x*	NetAdr: 1° byte	62.101.76.29
B	reti di medie dimensioni (16.384 reti con ciascuna al massimo 65.536 macchine connesse)	128-191 . 0-255 . *x* . *x*	NetAdr: 1° e 2° byte	159.149.53.208
C	reti di piccole dimensioni (2.097.152 reti con ciascuna al massimo 256 macchine connesse)	192-223 . 0-255 . 0-255 . *x*	NetAdr: 1°, 2° e 3° byte	192.6.165.40
D	riservata ad applicazioni di multicast	224-239 . *x* . *x* . *x*	Flat	-
E	riservata ad applicazioni future	240-255 . *x* . *x* . *x*	Non usato	-

Ma chi assegna gli indirizzi IP?

- L’indirizzo IP può essere assegnato:
  - da una persona umana (di norma l’amministratore di rete)
    - si parla in questo caso di **indirizzi IP fissi (o statici)**
    - la macchina avrà sempre l’indirizzo specificato fino a che l’amministratore non lo cambia andando ad operare fisicamente sulla macchina
    - questo vuol dire che la macchina avrà sempre lo stesso indirizzo IP (a meno di modifiche da parte dell’amministratore)
  - da una macchina dedicata che funge da server DHCP (***D**ynamic **H**ost **C**onfiguration **P**rotocol= protocollo di configurazione dinamica degli indirizzi*)
    - si parla in questo caso di **indirizzi IP dinamici**
    - ogni volta che la macchina viene accesa, il DHCP server le assegna un indirizzo
    - questo vuol dire che una macchina può avere nel tempo indirizzi IP differenti

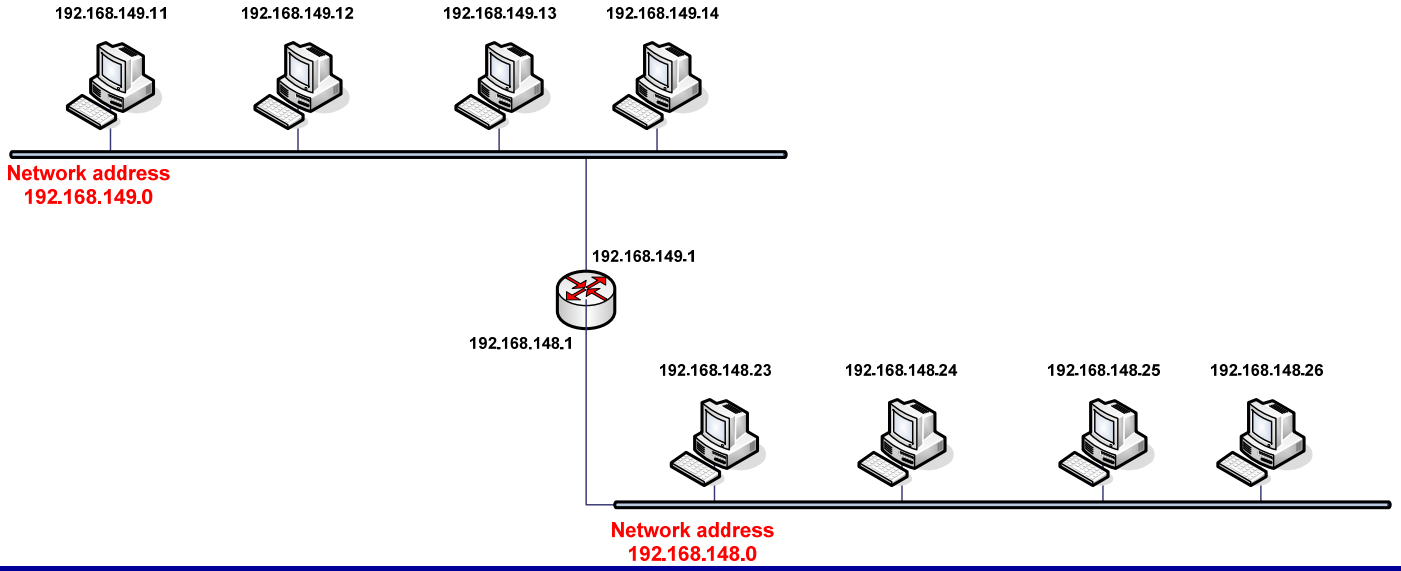
Indirizzi IP: assegnamento

- Le macchine che appartengono alla stessa rete LAN devono avere lo stesso indirizzo di rete e diverso indirizzo host
- Esempio:
  - È stato scelto di assegnare un indirizzo di rete di classe C: 192.168.149.0
  - Gli indirizzi host sono scelti a piacimento tra 1 e 255



Piani di indirizzamento con LAN complesse

- È possibile realizzare reti LAN costituite da più link. Un link identifica una sottorete
  - Ciò viene fatto quando le macchine da interconnettere sono numerose. Infatti, come si vedrà, il carico della rete viene ottimizzato
- Le sottoreti sono interconnesse tramite l'ausilio di apparati di rete denominati router che sono in grado di dirottare opportunamente il traffico di rete.
  - tali apparati sono usati anche per l'interconnesione di una rete LAN ad Internet
- Nell'esempio sotto
  - sono state costruite sue sottoreti con indirizzi rispettivamente 192.168.149.0 e 192.168.148.0
  - un router con uno specifico IP address per ciscuna sottorete le interconnette



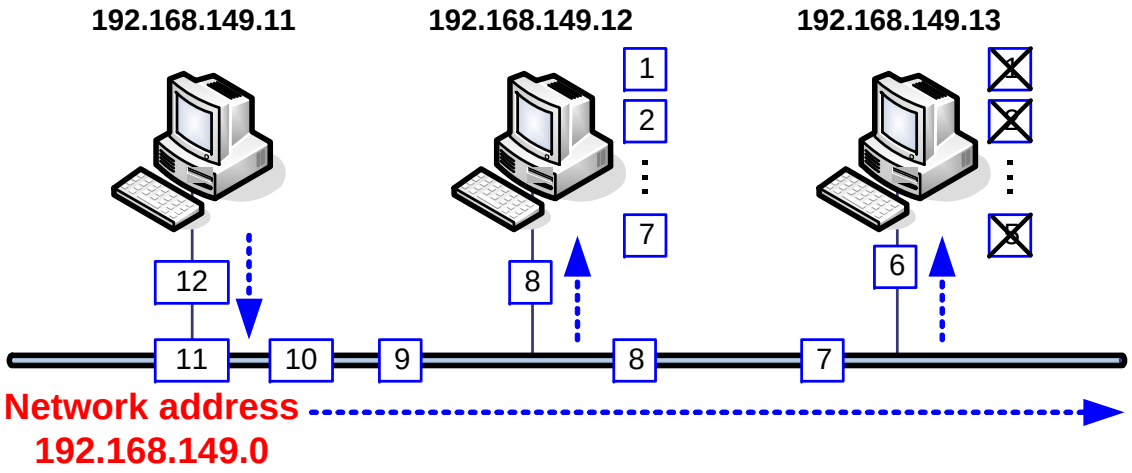
Indirizzi IP: funzionamento

- Quando una macchina deve inviare informazioni ad un'altra macchina
  - suddivide l'informazione in unità più piccole e le inserisce in pacchetti,
  - ciascun pacchetto contiene, oltre all'unità di informazione da trasmettere, anche altre informazioni tra cui l'indirizzo IP della macchina destinataria

- Esistono due casi:
  - Reti semplici: i nodi mittente e destinatario appartengono allo stesso link di una rete LAN
  - Rati complesse: I nodi mittente e destinatario appartengono a reti diverse

Indirizzi IP: funzionamento LAN semplice

- I pacchetti in cui è suddiviso il messaggio raggiungono tutte le macchine connesse al link stesso
  - Ciascuna macchina riceve i pacchetti, ma solo quella destinataria li elabora
- Esempio: supponiamo che la macchina 149.168.149.11 voglia mandare un messaggio alla macchina 149.168.149.12:
  - suddivide il messaggio in tante unità e costruisce i pacchetti con indirizzo 149.168.149.12 e li mette sul canale
  - tutte le macchine NON destinatarie ricevono e aprono i pacchetti; leggono l'indirizzo; poiché non coincide con il loro, buttano via i pacchetti
  - la macchina con IP 149.168.149.12 riceve il pacchetto, lo apre e, visto che è indirizzato a lei, ne legge il contenuto



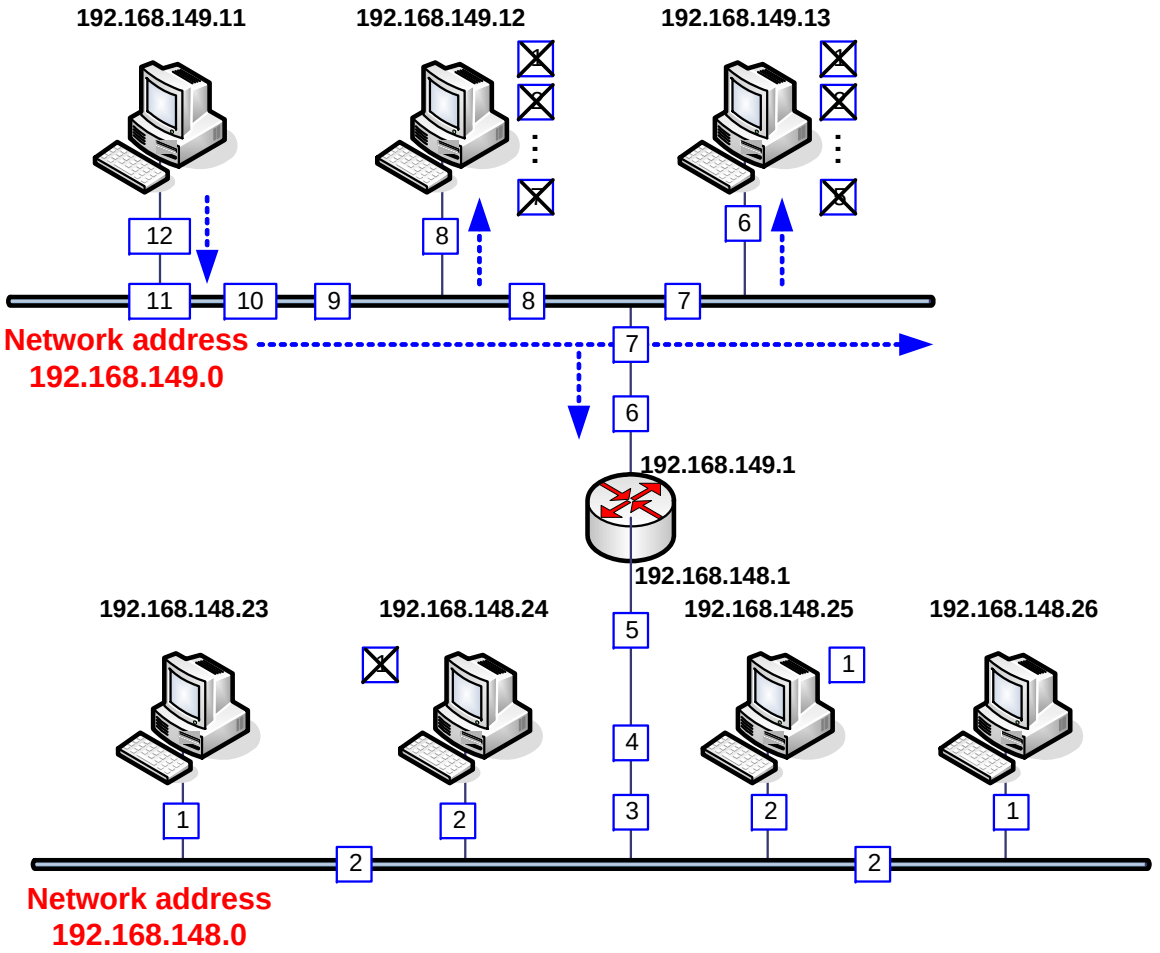
Indirizzi IP: funzionamento in Internet o in LAN complesse

<u>Problema</u>: il mittente e il destinatario non sono connessi, come "farli trovare"?

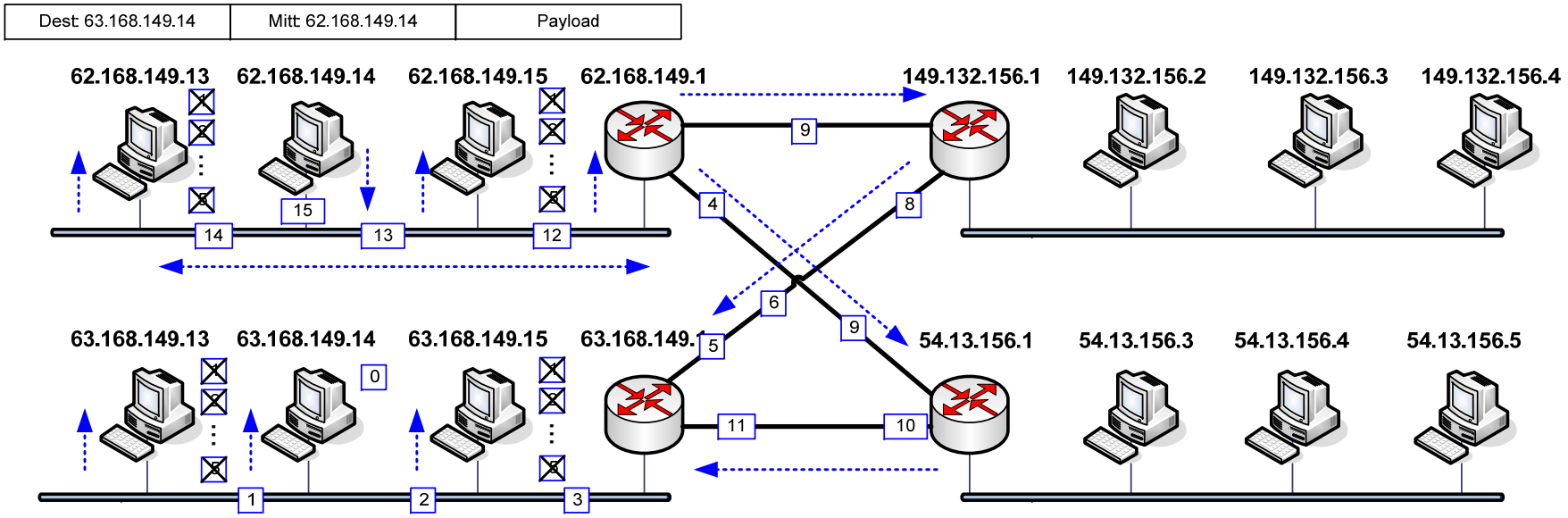
- Occorre:
  - conoscere l'indirizzo IP del destinatario
    - Presente nel pacchetto (o datagram)
  - conoscere il percorso che deve eseguire (routing)
    - Memorizzato in tabelle dette di routing
- I percorsi sono memorizzati in **tabelle di routing**
  - contengono esclusivamente indirizzi di rete
  - grazie al fatto che gli indirizzi sono gerarchici, si estrae la parte di network
- Come avviene la scelta del percorso?
- Esistono 2 tecniche:
  - **Routing statico:** impostazione manuale
  - **Routing dinamico:** le tabelle sono costruite dinamicamente attraverso algoritmi di routing e tenute aggiornate
- Esempi di algoritmi di routing distribuiti:
  - RIP, OSPF, IGRP, EGP, BGP

Esempio

- La macchina 192.168.149.11 deve inviare un messaggio alla macchina 192.168.148.25



Un esempio più complesso



Gli indirizzi privati

- Tutte le macchine che vogliono affacciarsi alla rete Internet devono avere un indirizzo IP univoco
- Però esiste un altro vincolo: tale indirizzo deve essere **pubblico** (i.e., visibile a tutte le macchine collegate alla rete)
- Problema: gli indirizzi IP non sono infiniti
- Soluzione:
  - **indirizzi IP privati**: invece di assegnare un indirizzo IP globalmente unico a ciascuna macchina della rete LAN (che comporterebbe un notevole spreco di indirizzi IP), si assegna un indirizzo IP non globalmente unico
- Gli indirizzi privati sono:
  - da 10.0.0.0 a 10.255.255.255 di classe A,
  - da 172.16.0.0 a 172.31.255.255 di classe B,
  - da 192.168.0.0 a 192.168.255.255 di classe C

Problema …

- Una rete LAN che vuole affacciarsi ad Internet utilizza generalmente un piano di indirizzamento privato
- Ciò sembra in contraddizione con quanto detto nella slide precedente: ogni macchina che si affaccia ad Internet deve avere un indirizzo IP pubblico

… Soluzione

- **NAT** (Network Address Translator) è un software di rete che risiede nel router che opera la traduzione degli indirizzi IP privati in pubblici e viceversa
- Nei pacchetti in uscita sostituisce tutte le occorrenze dell'indirizzo IP privato con il proprio pubblico
- Tiene traccia delle richieste inviate
- In ingresso sostituisce tutte le occorrenze dell'indirizzo pubblico (il proprio) con quello privato della macchina che aveva eseguito la richiesta
- NAT permette un mapping uno-a-uno tra un indirizzo privato e uno pubblico
- Nella configurazione più semplice, NAT opera su un router che connette 2 reti

IPv6

- IPv6 è il nuovo protocollo standard per Internet
- Windows Vista e un considerevole numero di distribuzioni Linux includono un supporto nativo per il protocollo
- Gli indirizzi sono lunghi 128 bits (16 bytes):un numero più che sufficiente per stare tranquilli a lungo!
  - Esempio di indirizzo: 2001:0db8:85a3:08d3:1319:8a2e:0370:7334
- La sostituzione di IPv4 con IPv6 è ostacolata da:
  - Incompatibilità tra IPv6 e IPv4: stesso livello della pila ISO OSI, stesse funzioni
  - Successo straordinario di IPv4
- Per potersi affermare IPv6 deve:
  - Garantire la compatibilità con i dispositivi IPv4 esistenti
  - Fornire strumenti che facilitino il processo di transizione da IPv4 a IPv6

L’approccio e lo scenario

- La necessità di meccanismi di transizione efficaci è stata riconosciuta sin dalle prime fasi di definizione di IPv6
  - Buona parte dello sforzo progettuale è stato dedicato ai meccanismi di transizione
- I meccanismi proposti sono numerosi e variano a seconda dello scenario
- La transizione sarà graduale
  - Non ci sarà un “D-Day” ma un lungo periodo di coesistenza dei due protocolli
  - Il processo non sarà breve (potrebbe durare decenni)
- Possiamo pensare ad un processo in tre fasi:
  - Fase iniziale
    - La rete IPv6 si appoggia all’infrastruttura IPv4
    - I nodi IPv6 utilizzano prevalentemente i servizi IPv4 esistenti
  - Fase intermedia
    - I due protocolli coesistono
  - Fase finale
    - La rete IPv4 si appoggia all’infrastruttura IPv6
    - I nodi legacy IPv4 devono poter utilizzare i servizi IPv6
- In generale le applicazioni devono comunque essere modificate per poter utilizzare IPv6

Gli altri protocolli (1/2)

- Se mittente e destinatario appartengono alla stessa rete ...
- ARP (*Address Resolution Protocol*)
  - al momento di spedire un pacchetto, il protocollo IP viene informato dai livelli superiori dell'indirizzo IP del destinatario
  - tramite ARP rileva l'*indirizzo hardware* per poter demandare il compito di spedizione al livello 2
  - ARP invia un messaggio broadcast in rete chiedendo alla macchina con IP specificato di rispondere
  - Per sapere la tabella dei mapping `arp -a`
- RARP (*Reverse Address Resolution Protocol*)
  - se una macchina non sa il suo indirizzo IP, ma il suo indirizzo hardware, invia un messaggio ad un server RARP con la richiesta informandolo del suo indirizzo hardware

Gli altri protocolli (2/2)

- DHCP
  - DHCP (Dynamic Host Configuration Protocol) Server è un servizio che automaticamente assegna indirizzi IP a client su cui gira il DHCP client software
  - Utilizzando DHCP, è possible centralizzare
    - la gestione degli indirizzi IP,
    - la netmask e
    - altre informazioni di configurazione IP
  - simile al BootP (*Boot Program*)

- ICMP (*Internet Control Message Protocol*)
  - è un protocollo di gestione che fornisce messagistica per IP
  - esempi di messaggi ICMP:
    - destination unreachable
    - buffer full
    - hops

Il livello Transport

- È il primo livello “end-to-end”: permette ad un programma su A di condurre una conversazione con un programma su B senza intermediari
- Il ruolo è quello di suddividere in segmenti le informazioni pervenute dalle applicazioni e mascherare alle stesse la complessità di una rete
- Sono stati definiti 2 protocolli end-to-end:
  - **TCP** (Transmission Control Protocol)
    - connection-oriented: i pacchetti arrivano nello stesso ordine in cui sono stati inviati
    - affidabile:
      - non perde mai i dati: assicura cioè che tutti i dati inviati siano effettivamente giunti a destinazione
      - richiede che il destinatario invii una conferma di ricezione (ack) per ogni pacchetto ricevuto (ciò introduce overhead)
  - **UDP** (User Datagram Protocol)
    - connectionless: i pacchetti possono arrivare in ordine diverso
    - non affidabile: non offre la certezza che tutti i dati siano stati ricevuti

TCP vs. UDP

- A prima vista il confronto è impari:
  - TCP assicura tutto: l'arrivo e l'ordine
  - UDP non assicura niente: né l'arrivo né l'ordine

- …però…

- la connessione e l'affidabilità *costano molto*, per cui:
  - per certi servizi UDP è senz'altro preferibile. Esempi:
    - voce
    - video
    - in genere, per quei servizi il cui destinatario è un essere umano
  - per altri TCP è indubbiamente meglio. Esempio classico:
    - trasferimento di file
    - in genere, tutti quei servizi il cui destinatario è una macchina

Il livello applicativo (1/2)

- I due livelli 5 e 6 del modello ISO/OSI si sono dimostrati *inutili*!
  - le funzionalità che definiscono sono incorporate negli altri livelli (principalmente in quello applicativo)

- Il livello Applicativo contiene tutti i protocolli di alto livello che vengono usati dalle applicazioni

- I protocolli definiti consentono di instaurare comunicazioni fra macchine remote indipendentemente dalla configurazione hardware e dal sistema operativo

- I primi protocolli furono
  - *Telnet*: terminale virtuale
  - *FTP* (*File Transfer Protocol*): trasferimento di file
  - *SMTP* (*Simple Mail Transfer Protocol*) e *POP* (*Post Office Protocol*): posta elettronica

Il livello applicativo (2/2)

- … a cui successivamente se ne aggiunsero altri:
  - *DNS* (*Domain Name System*): mapping fra nomi di host e indirizzi IP
  - *NNTP (Network News Transfer Protocol): articoli nei newsgroup*
  - *HTTP* (*Hyper Text Transport Protocol*) : trasferimeneto di file html
  - *IRC*: chat
  - …
  - e tutto quanto può venire in mente, infatti:

- Il livello applicativo è “aperto”:
  - ogni giorno vengono inventate e sviluppate applicazioni nuove

Il DNS (Domain Name System)

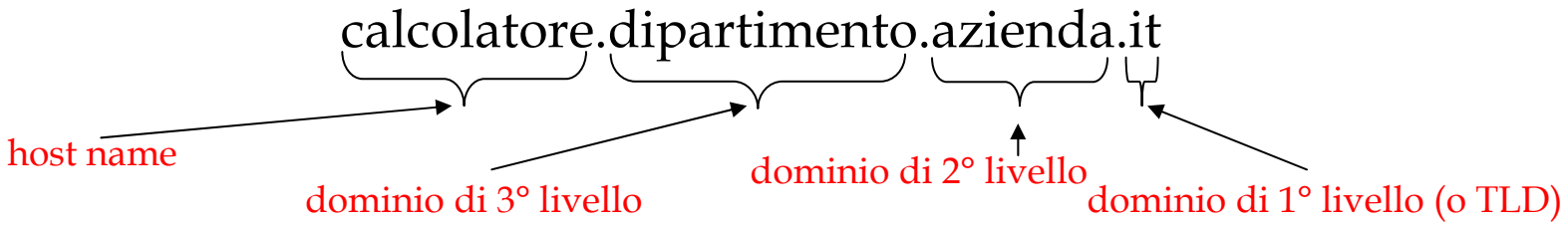
- Ìn una rete TCP/IP (ad esempio Internet):
  - ogni macchina ha un indirizzo IP univoco

- Ad un computer può essere assegnato anche un **indirizzo simbolico** (o *nome simbolico* o *nome di dominio* o *domain name*)

- Perché 2 identificativi?
  - I nomi simbolici sono comodi per gli utenti
  - Gli indirizzi IP sono comodi per i calcolatori
    - tempi di computazione inferiore (es: confronto fra 2 indirizzi)
    - occupazione di memoria inferiore
    - tempi di trasmissione inferiori

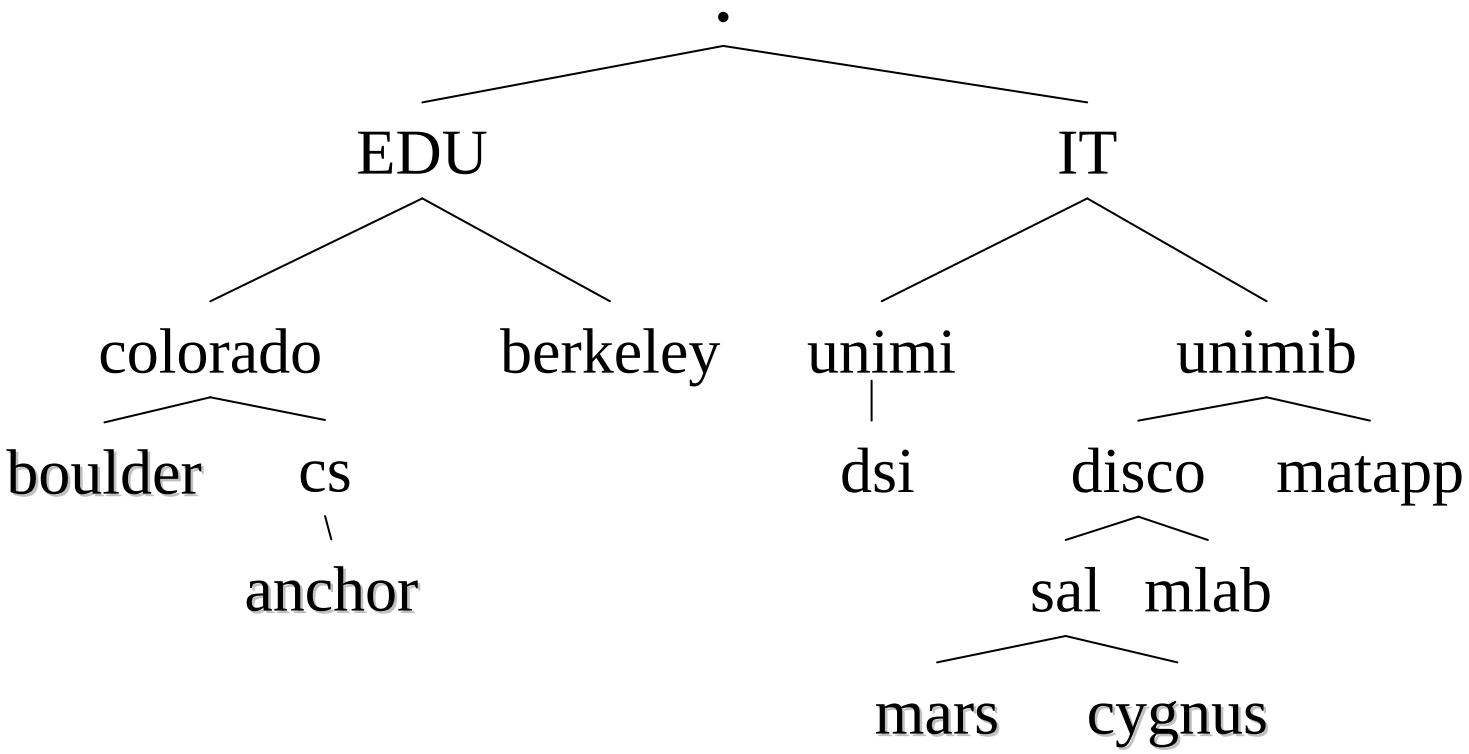
I nomi simbolici

- Un domain name è composto da:
  - un *hostname*, che identifica la macchina all'interno della sua LAN
  - un *dominio*, che identifica la LAN nella rete generale

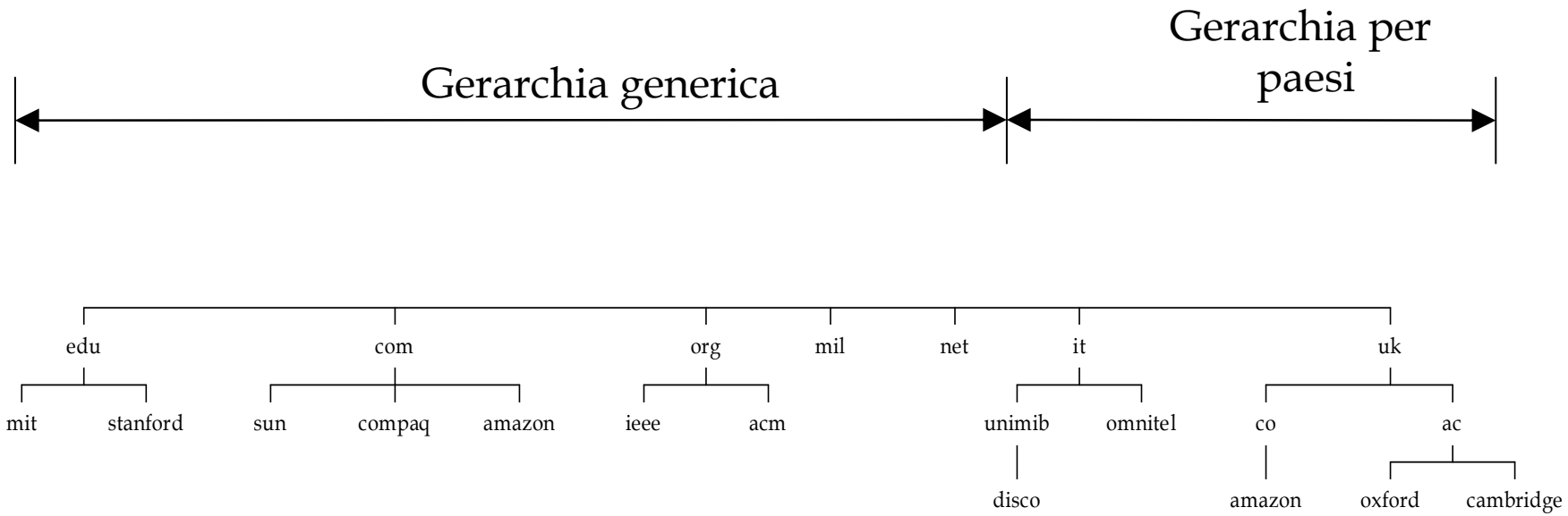


Lo spazio dei nomi

- È l'albero dei domini
- Ogni dominio rappresenta un ramo distinto ed è gestito da una singola entità amministrativa
- La radice è chiamata "." e sotto di essa ci sono i domini denominati TLD



Il quadro (parziale) dei domini



I name server

- Chi tiene traccia delle corrispondenze tra indirizzi IP e domain name?
  - ovvero, chi “risolve” i nomi?
  - un insieme di server *DNS* (Domain Name Server) sparsi per la rete
  - *in generale, ogni dominio ha il suo server DNS che risolve i suoi indirizzi*

- Operano in una delle seguenti modalità:
  - Primary
    - necessario per ogni dominio o sottodominio
    - mantiene i dati originali del dominio sul disco
  - Secondary
    - copia i suoi dati del dominio da un primary server via un’operazione “zone transfer”
  - Caching-only
    - caricano pochi indirizzi da un file di configurazione, il resto aggiornato attraverso risposte a richieste che lui risolve

Tipi di risposte

- Risposte al *resolver* di due tipi
  - Authoritative
    - È garantita di essere accurata
  - Non-authoritative
    - Può essere out of date

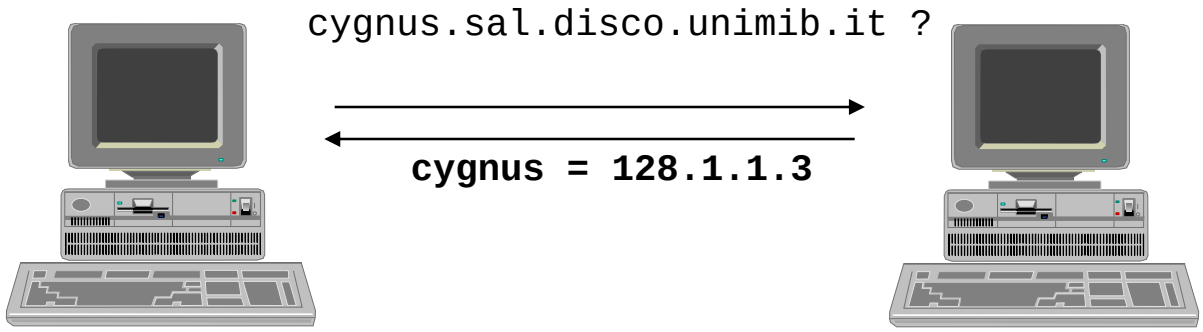
- *Primary* e *secondary* name server sono *authoritative* per i propri domini, ma non per le informazione cached di altri domini
  - anche risposte authoritative possono essere inconsistenti se il sysadmin non aggiorna il secondary name server a fronte di cambiamenti nel primary name server

- *Caching-only* name server non sono mai *authoritative*

Come funziona?

- Name server possono essere
  - *recurvive*
  - *non-recursive*
- Non-recursive
  - Se ha la risposta in cache o è authoritative per il dominio al quale appartiene il nome, fornisce una risposta appropriata
  - In caso contrario, fornisce un referral al server authoritative di un altro dominio che potrebbe sapere la risposta
- Recursive
  - Restituiscono solo la risposta reale o messaggi di errore
  - Pros
    - caching
  - Cons
    - traffico

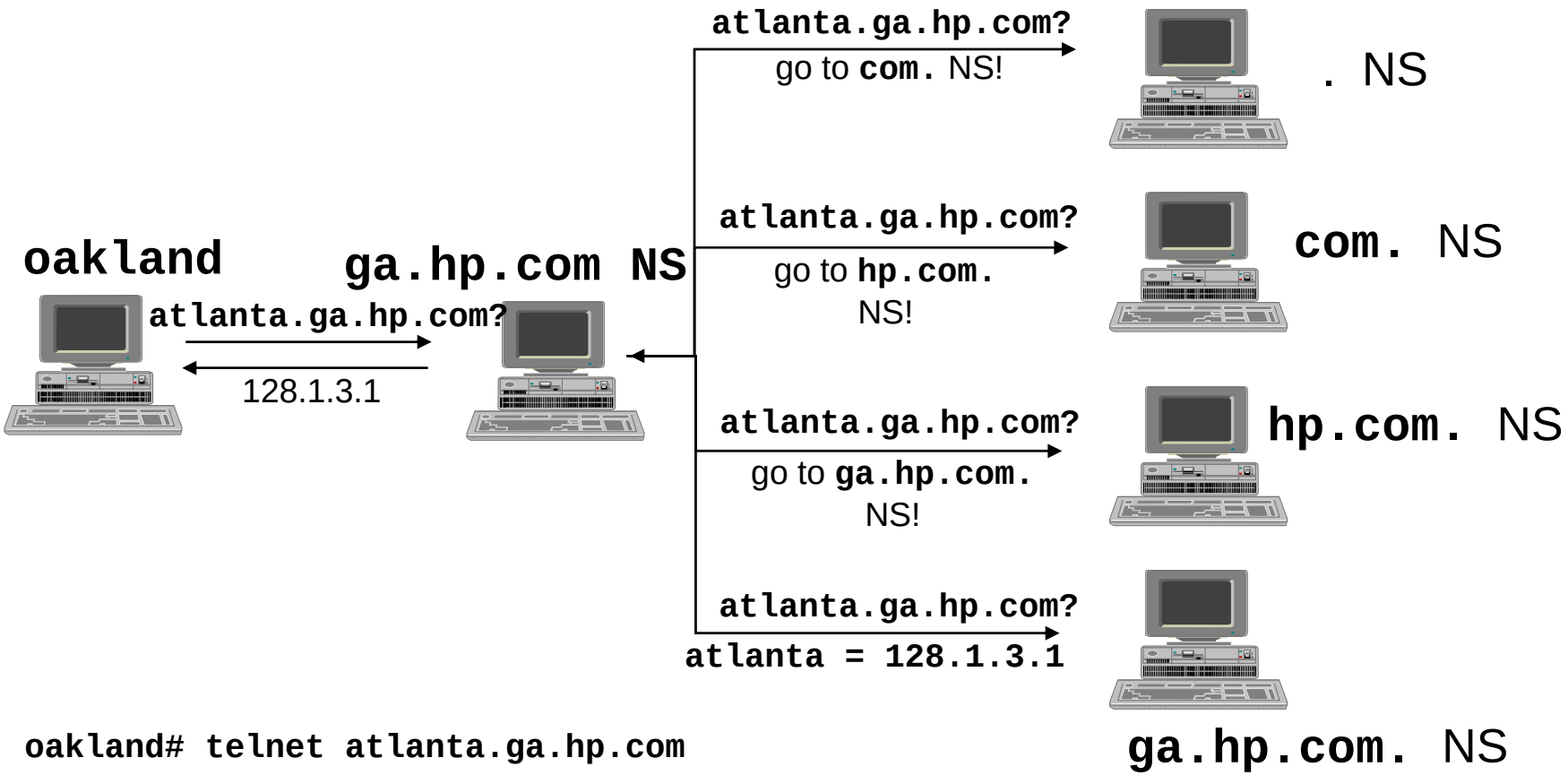
Risoluzione in dominio locale



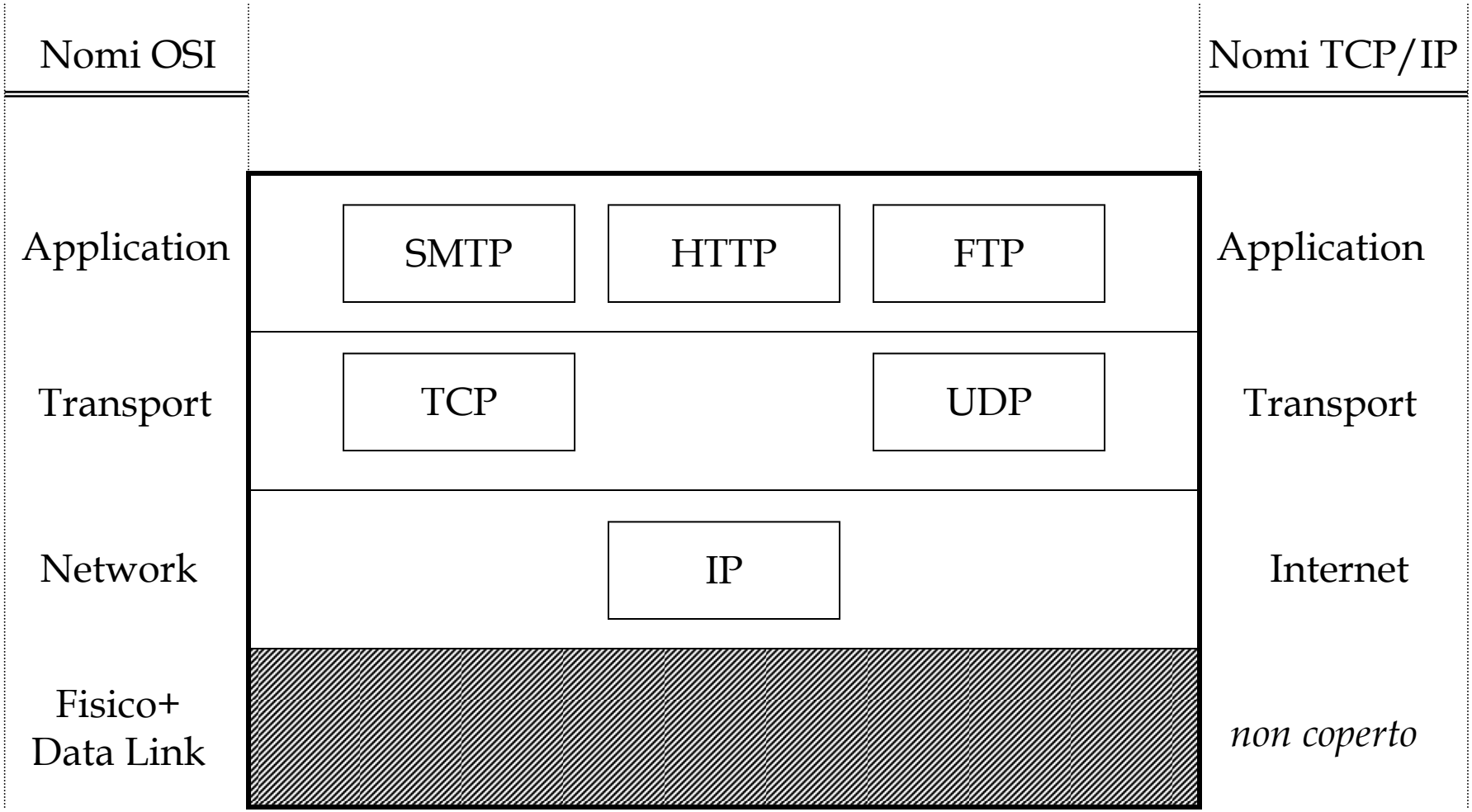

```
# telnet cygnus.sal.disco.unimib.it
```

callisto	128.1.1.1
venus	128.1.1.2
cygnus	128.1.1.3

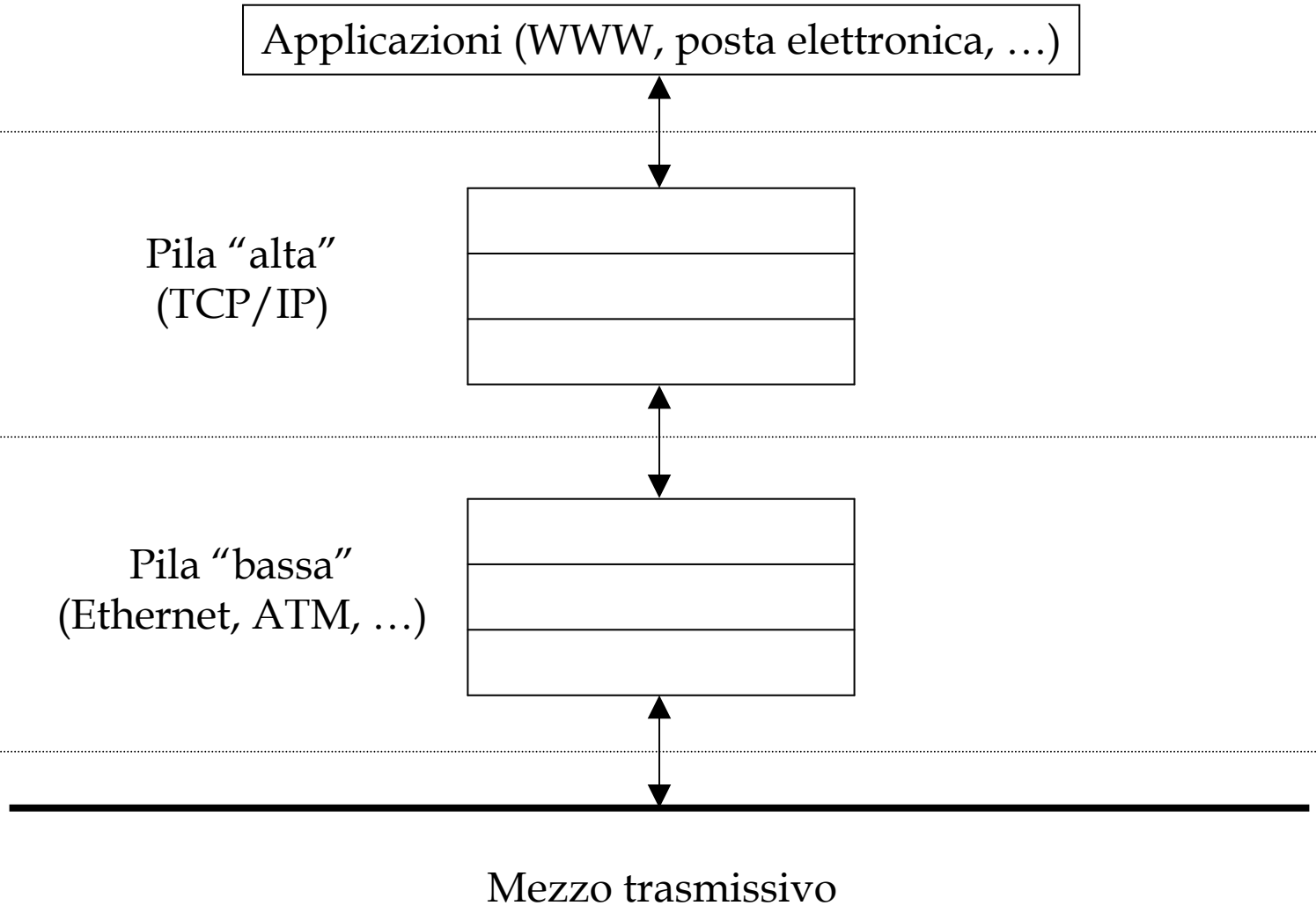
Risoluzione in altri domini (recursive)



Lo scenario conclusivo



Scenario conclusivo



Fine