

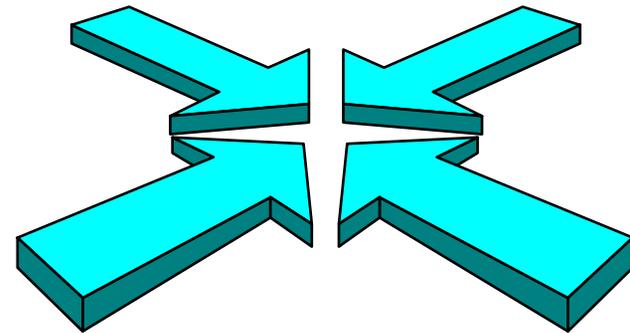
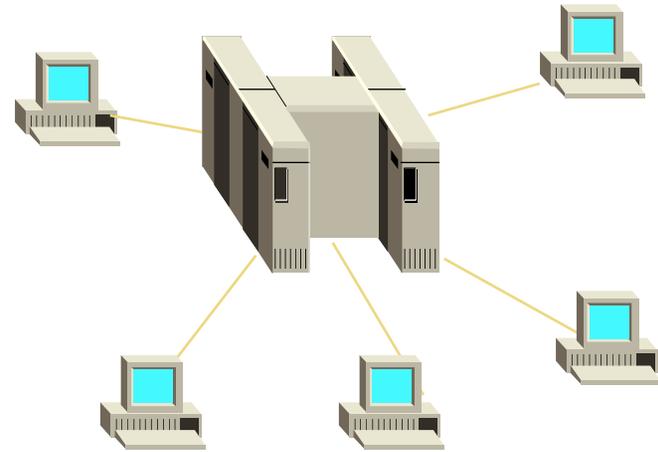
Reti di calcolatori

Introduzione

Slide a cura di Simon Pietro Romano
spromano@unina.it

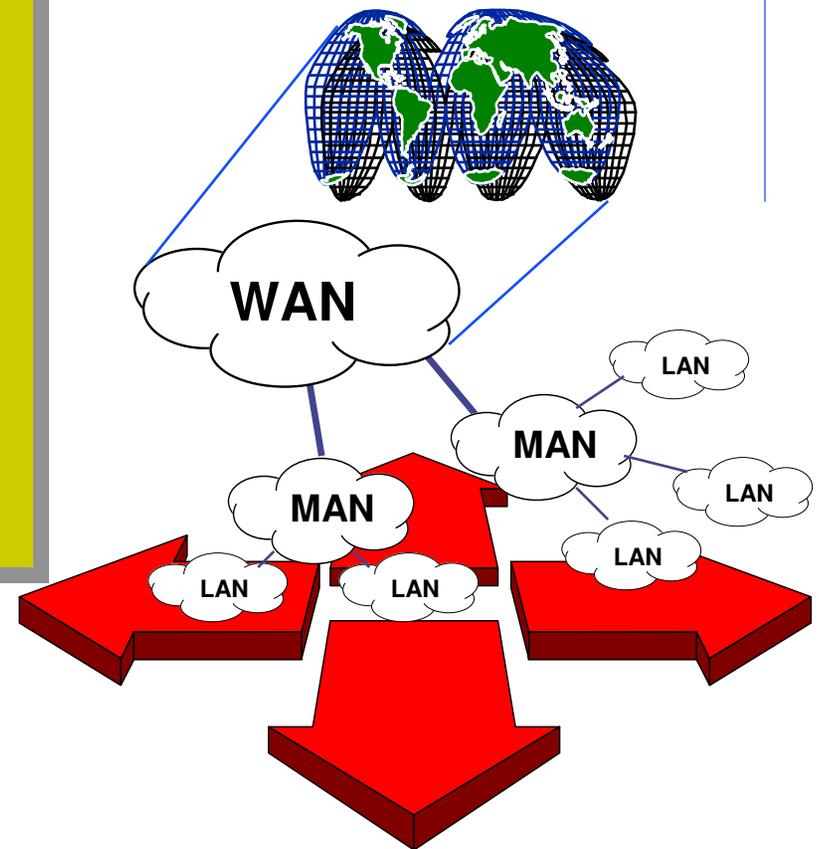
L'evoluzione: dal "computing centralizzato"...

- Sistemi fortemente centralizzati
- Grandi capacità di calcolo
- Rete complessa ma... *omogenea*
- Ambiente periferico:
terminali e stampanti



L'evoluzione: ...al "computing distribuito"

- Avvento del PC
- Nascita e boom delle reti locali
- Evoluzione verso sistemi *aperti*
- Periferia eterogenea ed intelligente
- Nascita degli standard per:
 - Cablaggi strutturati
 - Protocolli di Comunicazione



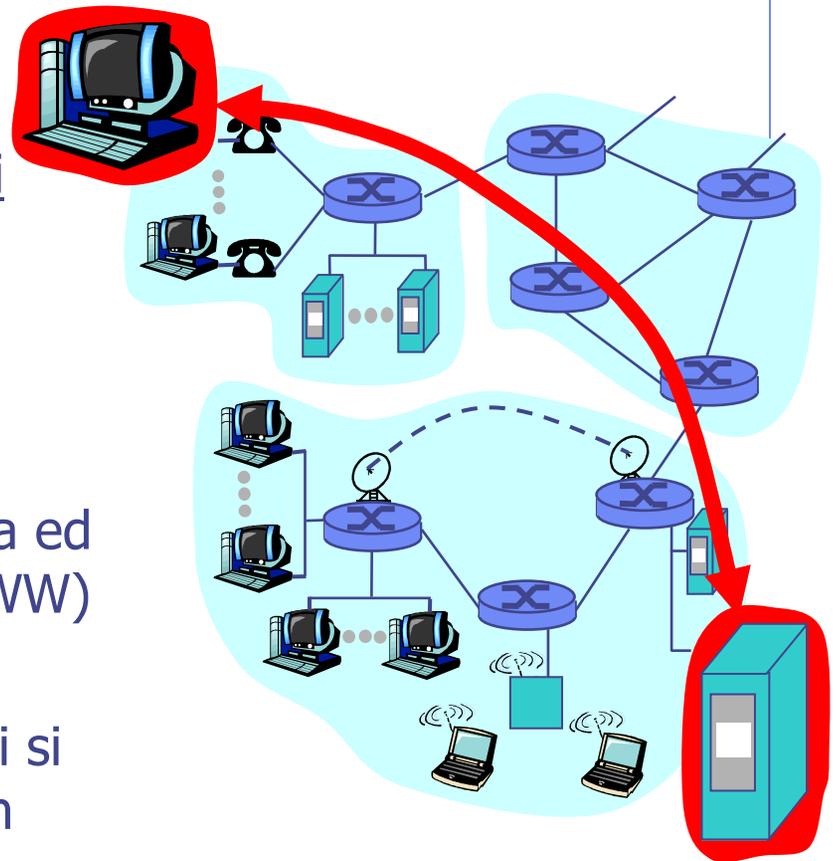
Le reti di calcolatori: scopi

- ◆ **Condivisione dell'informazione**
- ◆ **Condivisione delle risorse**
- ◆ **Accesso a risorse remote**
- ◆ **Convenienza economica**
- ◆ **Crescita graduale**

Elementi costitutivi delle reti di calcolatori (1)

◆ Alle estremità della rete si trovano gli *end-system* o *host*

- sono calcolatori di vario tipo su cui girano i programmi applicativi
- i programmi applicativi possono essere progettati secondo due modelli:
 - ◆ Client-Server
 - Il client invia una richiesta ed il server risponde (es. WWW)
 - ◆ Peer-to-peer
 - Le due entità comunicanti si scambiano informazioni in modo paritetico (es. teleconferenza)



Elementi costitutivi delle reti di calcolatori (2)

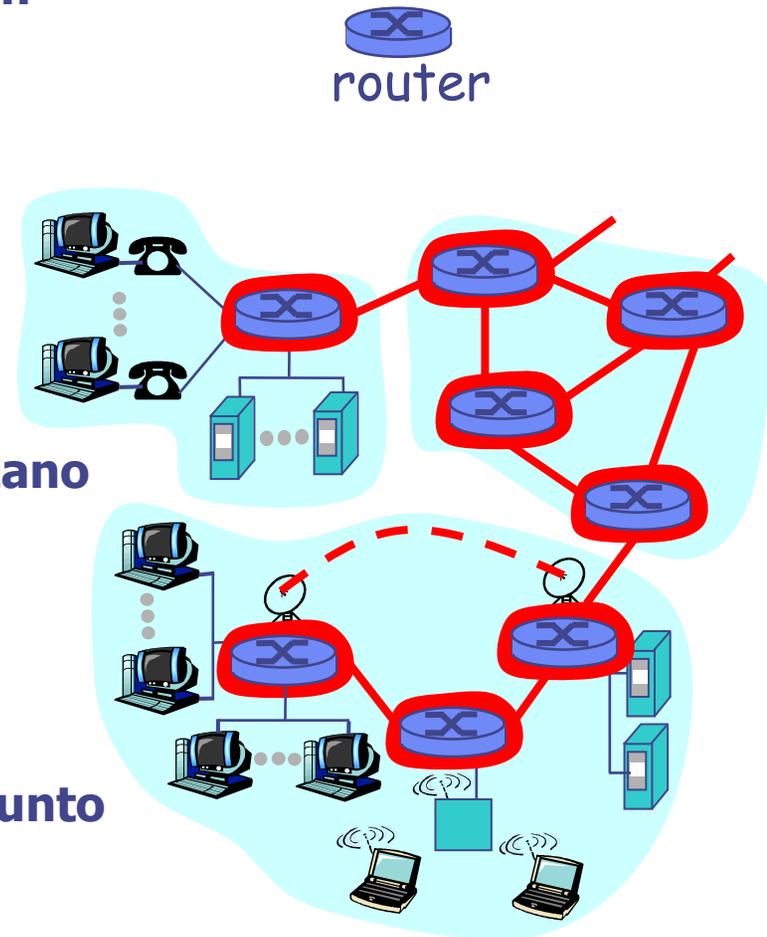
◆ L'infrastruttura della rete è fatta di:

■ apparati tra loro interconnessi

- ◆ hub, switch, bridge
- ◆ modem
- ◆ access point
- ◆ router
- ◆ ...

■ supporti trasmissivi che realizzano le interconnessioni

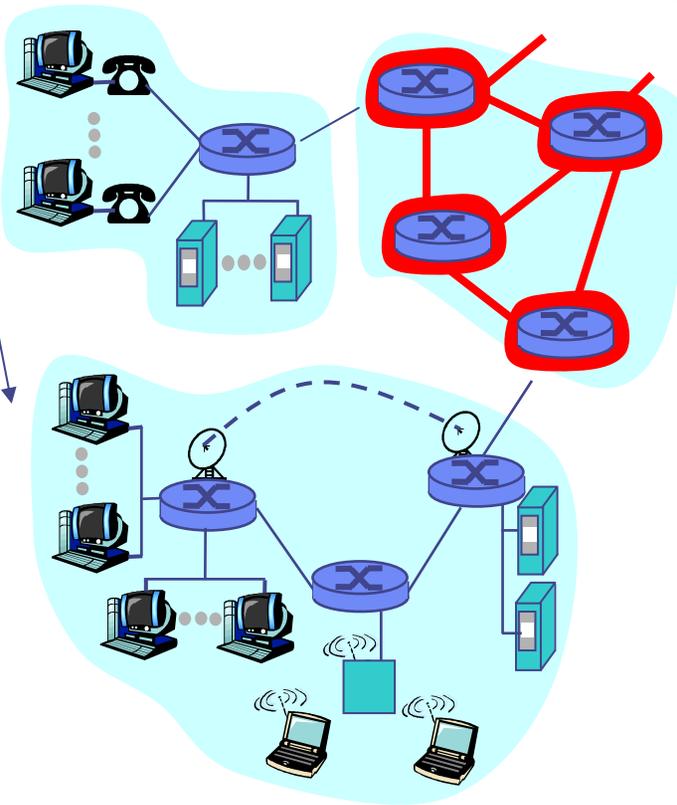
- ◆ doppini in rame
- ◆ cavi coassiali
- ◆ fibre ottiche
- ◆ collegamenti radio punto-punto
- ◆ collegamenti satellitari
- ◆ ...



Struttura delle reti di calcolatori

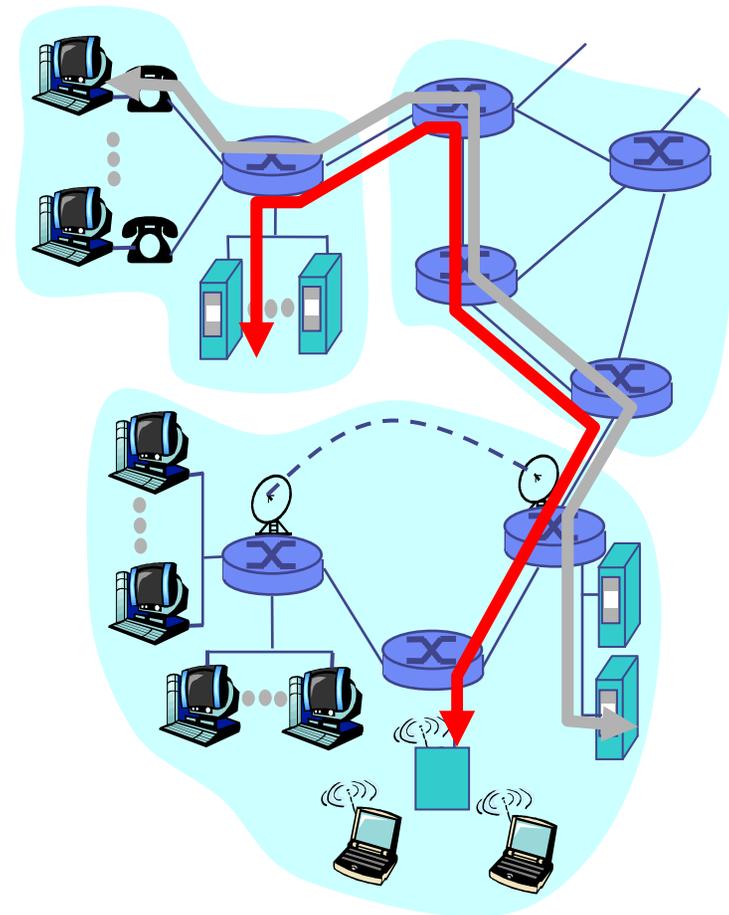
Reti di
accesso

Backbone



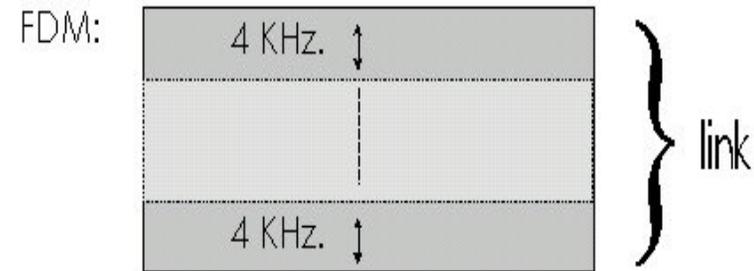
Commutazione di circuito

- ◆ Nelle reti a commutazione di circuito, la capacità trasmissiva all'interno della rete è assegnata per ciascuna "chiamata"
 - E' definita una porzione di capacità trasmissiva che è allocata in modo esclusivo per servire ciascuna comunicazione
 - È il modello dell'attuale rete telefonica



Commutazione di circuito

- ◆ Nella telefonia tradizionale, due coppie di conduttori venivano impegnate per ciascuna conversazione
- ◆ Successivamente, le coppie "fisiche" sono state sostituite da:
 - **"porzioni di banda"** (moltiplicazione a divisione di frequenza, **FDM**)
 - **"porzioni di tempo"** (moltiplicazione a divisione di tempo, **TDM**)



TDM:



All slots labelled **2** are dedicated to a specific sender-receiver pair.

Commutazione di Pacchetto

- ◆ La natura discontinua della trasmissione di dati digitali può essere sfruttata per far sì che flussi di dati differenti possano condividere la stessa connessione, a patto di poterli distinguere
- ◆ Questo principio è alla base della tecnica detta "commutazione di pacchetto" (*packet switching*)

Commutazione di Pacchetto (2)

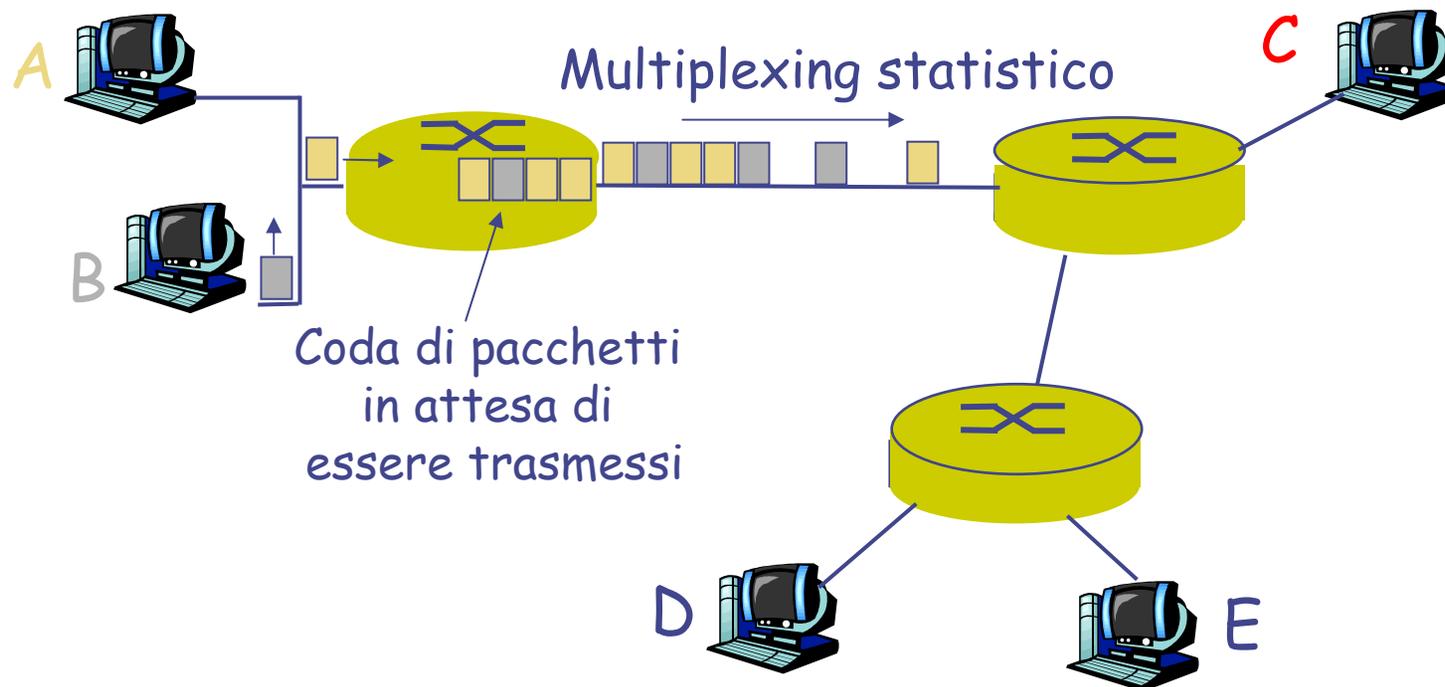
- ◆ Nel *Packet Switching* ciascun flusso di dati è diviso in pacchetti, cioè in entità composte da:
 - un'intestazione (*header*), utilizzata ai fini dell'identificazione e gestione,
 - i dati veri e propri (*payload*)

- ◆ Una rete a commutazione di pacchetto è composta da:
 - sistemi terminali (*End System o host*): producono o ricevono dati
 - apparati che si occupano dell'instradamento dei pacchetti tra sorgente e destinazione, detti nodi della rete (*Network Nodes*)

- ◆ Ogni nodo memorizza i pacchetti in ingresso, per poi instradarli verso il nodo successivo (*store & forward*)

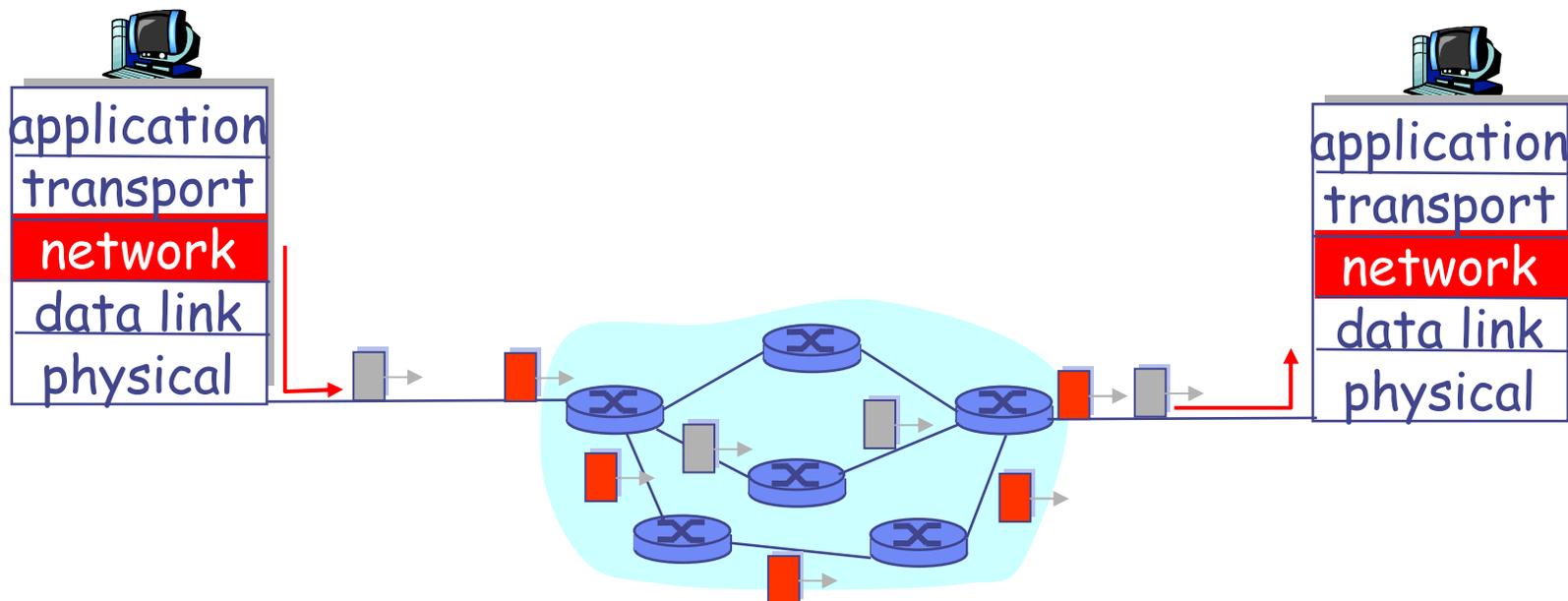
- ◆ I collegamenti fisici tra i nodi sono detti *link*

Commutazione di Pacchetto (3)



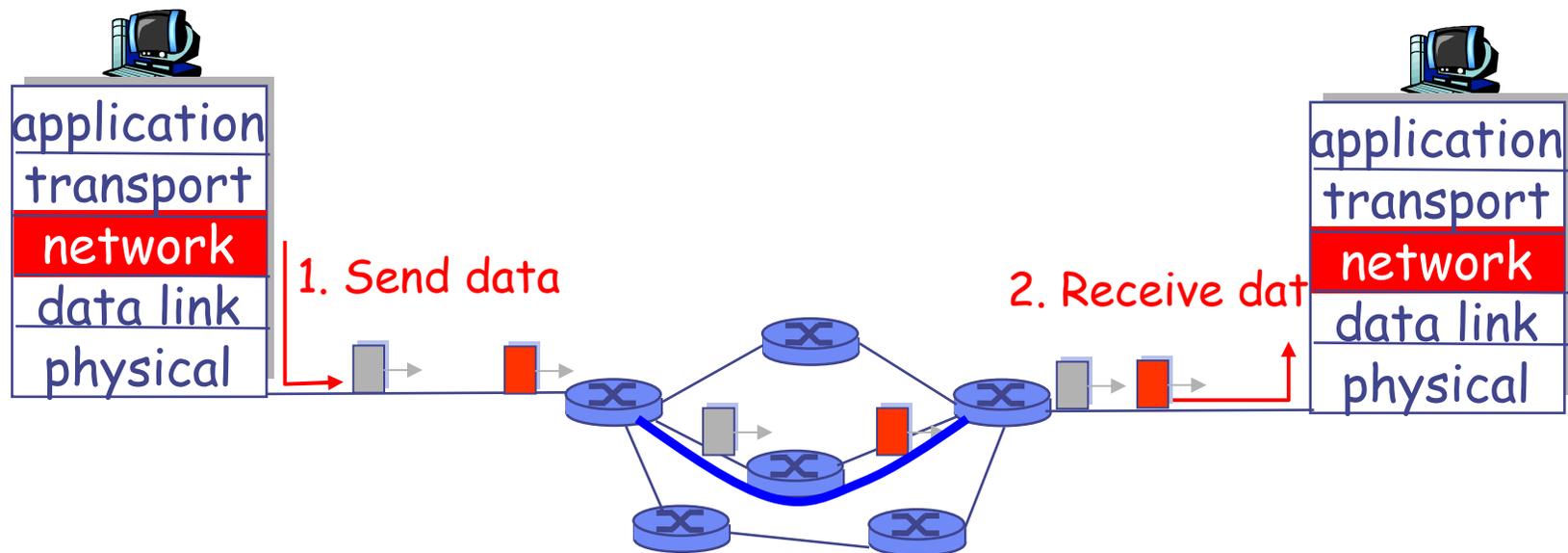
Packet switching: reti a datagrammi

- ◆ Ogni nodo che riceve un pacchetto decide in maniera indipendente a quale altro nodo inoltrarlo, sulla base dell'indirizzo destinazione contenuto nel pacchetto
- ◆ Pacchetti tra la stessa coppia sorgente-destinazione possono seguire percorsi differenti



Packet switching: reti a circuiti virtuali

- ◆ Ogni pacchetto contiene il numero del circuito virtuale
- ◆ Il circuito virtuale è stabilito prima della trasmissione dei dati
- ◆ I nodi devono conservare informazioni sui circuiti virtuali che li attraversano



Datagrammi vs circuiti virtuali

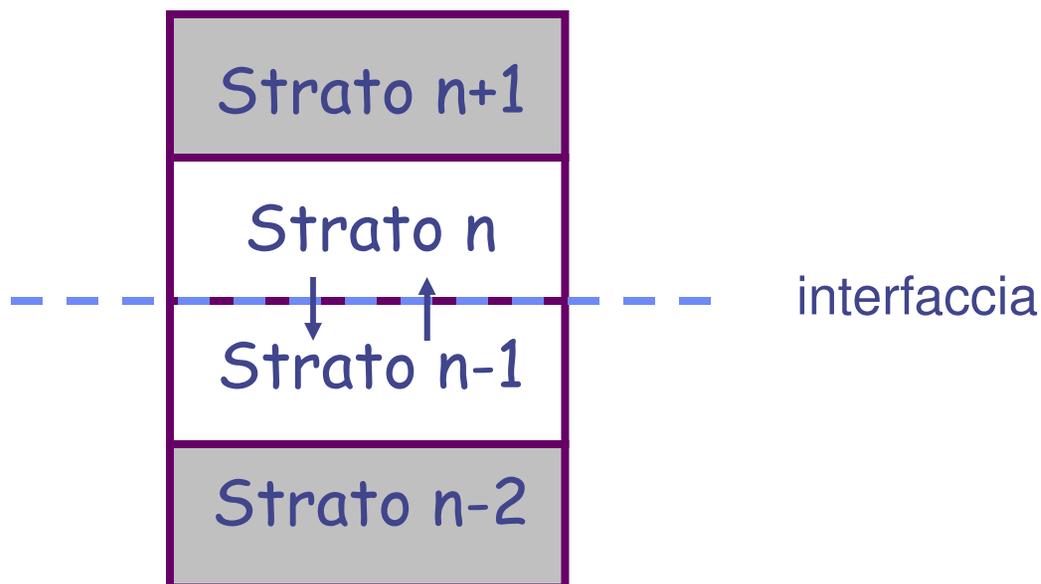
Proprietà	datagrammi	circuiti virtuali
Creazione del circuito	Non richiesta 😊	Richiesta ☹️
Indirizzamento	Ogni pacchetto contiene l'intero indirizzo della sorgente e della destinazione ☹️	Ogni pacchetto contiene un numero di VC 😊
Informazioni sullo stato	I nodi di rete non mantengono informazioni sullo stato 😊	Ogni VC richiede uno spazio di memoria sui nodi ☹️
Instradamento	Ogni pacchetto è instradato indipendentemente ☹️	Percorso pre-calcolato: ogni pacchetto segue questo percorso 😊
Effetti di guasti ai nodi	Nessuno (solo i pacchetti persi durante il guasto) 😊	Tutti i VC che attraversano quel nodo sono chiusi ☹️
Controllo di congestione	Complicato ☹️	Semplice se possiamo allocare spazio sufficiente per ogni VC 😊

Modelli a strati: perché

- ◆ Come vedremo, la suddivisione delle funzionalità secondo un modello a strati agevola la gestione della complessità
- ◆ Ciascuno **strato** (o **livello**):
 - è responsabile di un sottoinsieme definito e limitato di compiti
 - funziona indipendentemente dagli altri
 - interagisce solo con gli strati immediatamente superiore ed inferiore
 - fa affidamento sui "servizi" forniti dallo strato immediatamente inferiore
 - fornisce "servizi" allo strato immediatamente superiore
- ◆ Alcuni strati sono realizzati in software altri in hardware
- ◆ Vantaggi:
 - l'indipendenza tra gli strati consente la sostituzione di uno strato con un altro di pari livello che offra i medesimi servizi allo strato superiore
 - limitare le funzionalità di uno strato ne semplifica la realizzazione
- ◆ Svantaggi:
 - L'eccessivo numero di strati può portare ad inefficienze

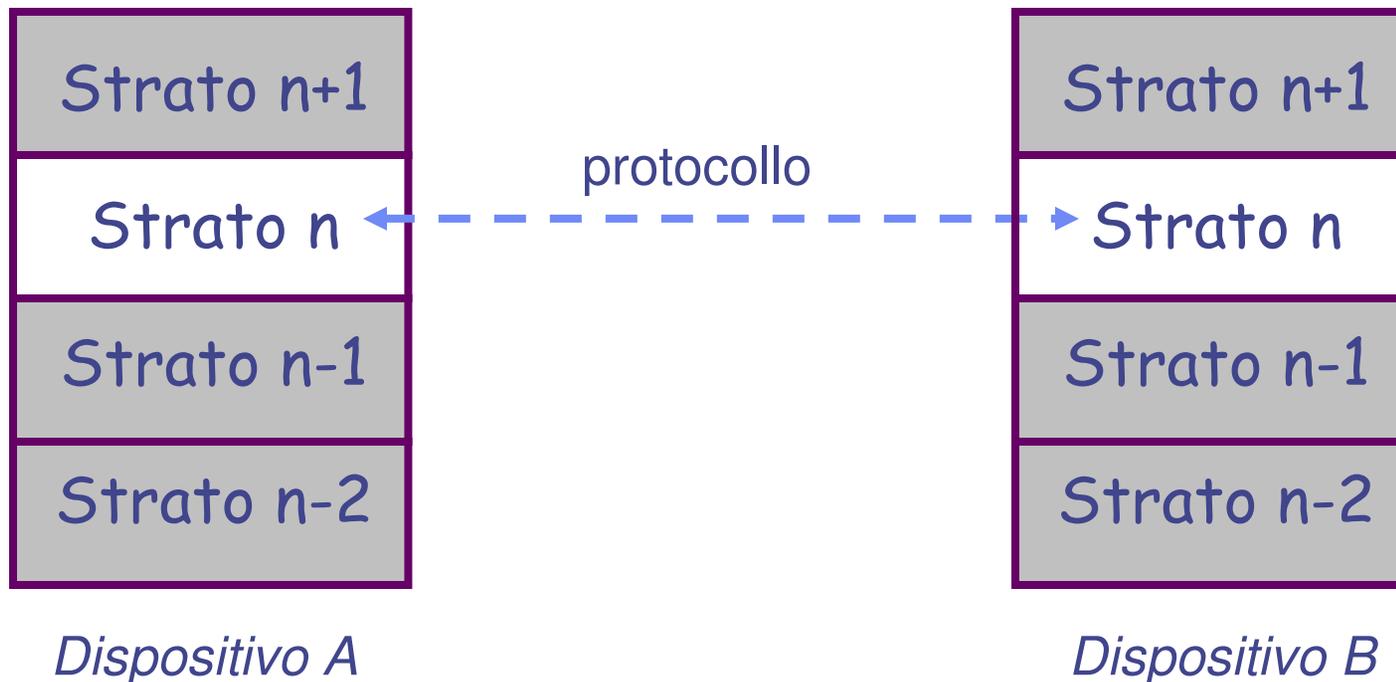
Modelli a strati: interfacce

- ◆ All'interno di ciascun dispositivo di rete, lo scambio di informazioni tra due strati adiacenti avviene attraverso una interfaccia, che definisce i servizi offerti dallo strato inferiore allo strato superiore



Modelli a strati: protocolli

- ◆ Lo strato n-esimo di un dispositivo comunica con lo strato n-esimo di un'altra entità secondo un protocollo assegnato



Protocolli di comunicazione

- Per *protocollo di comunicazione* si intende un insieme di regole che permette la corretta instaurazione, mantenimento e terminazione di una comunicazione di qualsiasi tipo tra due o più entità
- Un protocollo di comunicazione definisce il formato e l'ordine dello scambio di messaggi tra le entità comunicanti
- Nelle reti di calcolatori, un protocollo regola la comunicazione tra entità di pari livello esistenti in due dispositivi della rete tra loro comunicanti
- Nell'ambito delle reti di computer un notevole sforzo è stato compiuto per definire protocolli standard, allo scopo di consentire l'integrazione di reti differenti

Il modello OSI (1)

Negli anni '80 l' *ISO, International Standards Organization*, ha definito un modello di riferimento per reti di calcolatori a commutazione di pacchetto: il modello *OSI, Open System Interconnection*

Il modello OSI è un modello a strati su 7 livelli:

Applicazione

Presentazione

Sessione

Trasporto

Rete

Data link

Fisico

Il modello OSI non è risultato vincente, a causa della sua eccessiva complessità

Il modello OSI (2)



OSI: Open Systems Interconnection

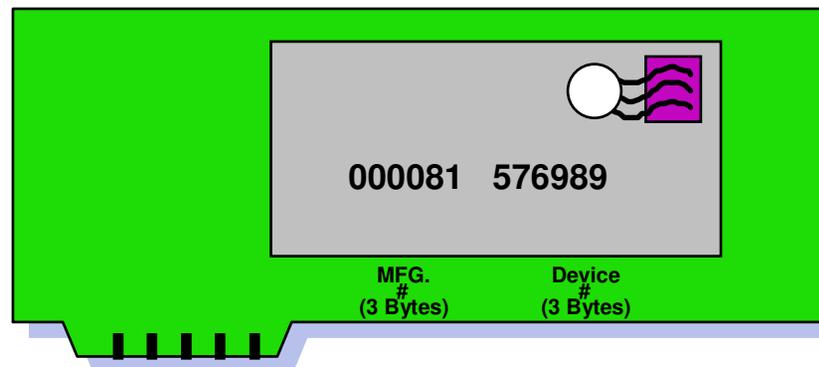
Livello 1: Fisico

- Si occupa di trasmettere sequenze binarie sul canale di comunicazione
- A questo livello si specificano:
 - Caratteristiche elettriche dei segnali
 - Tecniche di codifica/decodifica
 - Caratteristiche dei mezzi trasmissivi
 - Tipi di connettori
- Il livello fisico è nel dominio dell'ingegneria elettronica: descrizione elettrico/meccanica dell'interfaccia

0 1 0 1 0 1 0 1 0 1 0 1 0

Livello 2: Data Link

- ◆ Ha come scopo la trasmissione affidabile di pacchetti di dati (*frame*)
 - Affidabile nel senso di “garanzia di inoltro”
- ◆ Accetta come input i *frame* (tipicamente poche centinaia di byte) e li trasmette sequenzialmente
- ◆ Verifica la presenza di errori di trasmissione aggiungendo delle informazioni aggiuntive di controllo
 - *Frame Control Sequence, FCS*
- ◆ Può gestire meccanismi di correzione di errori tramite ritrasmissione



Livello 3: Rete

- ◆ Questo livello gestisce l'instradamento dei messaggi
- ◆ Determina quali sistemi intermedi devono essere attraversati da un messaggio per giungere a destinazione
- ◆ Il livello 3 gestisce, quindi, delle tabelle di instradamento per ottimizzare il traffico sulla rete

Livello 4: Trasporto

- ◆ Fornisce servizi per il trasferimento dei dati da terminale a terminale (ovvero *end-to-end*), indipendentemente dalla rete sottostante
- ◆ In particolare il livello 4 può:
 - frammentare i pacchetti in modo che abbiano dimensioni idonee al livello 3
 - rilevare/correggere gli errori
 - controllare il flusso
 - controllare le congestioni

Livello 5: Sessione

- ◆ Il livello 5 è responsabile dell'organizzazione del dialogo e della sincronizzazione tra due programmi applicativi e del conseguente scambio di dati
- ◆ Si occupa cioè di stabilire la **sessione**

Livello 6: Presentazione

- Il livello di presentazione gestisce la sintassi dell'informazione da trasferire
- L'informazione è infatti rappresentata in modi diversi su elaboratori diversi (es. ASCII o EBCDIC)



Livello 7: Applicazione

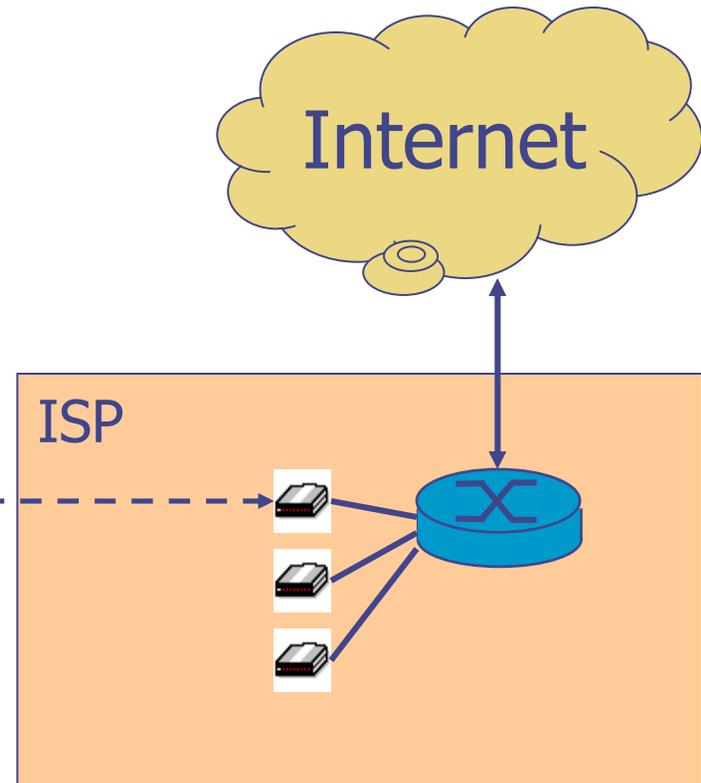
- ◆ È il livello dei programmi applicativi, cioè di quei programmi appartenenti al sistema operativo o scritti dagli utenti, attraverso i quali l'utente finale utilizza la rete
- ◆ Esempi di applicazioni previste dall'OSI sono:
 - VT: Virtual Terminal, connessione interattiva ad un elaboratore remoto
 - FTAM: File Transfer and Access Management
 - X.400: Posta Elettronica
 - X.500: Directory Service
- ◆ Nel mondo Internet, le applicazioni sono: Posta Elettronica, WWW, FTP, ecc...

Internet: architettura della rete

- ◆ La rete è progettata secondo un modello a **datagram**
- ◆ L'informazione viaggia in pacchetti (*datagram*) che vengono trattati dalla rete indipendentemente l'uno dagli altri
- ◆ Ogni terminale è univocamente individuato da un indirizzo associato alla interfaccia che lo collega alla rete
- ◆ Ogni pacchetto contiene l'indirizzo del mittente e l'indirizzo del destinatario
- ◆ L'infrastruttura della rete è costituita dai **router** che hanno il compito di instradare i pacchetti e consegnarli a destinazione
- ◆ Non c'è garanzia che un pacchetto venga realmente consegnato a destinazione
 - I pacchetti possono andare persi nella rete
 - I pacchetti possono seguire percorsi diversi ed arrivare in un ordine diverso da quello con cui sono stati trasmessi

Accesso ad Internet residenziale

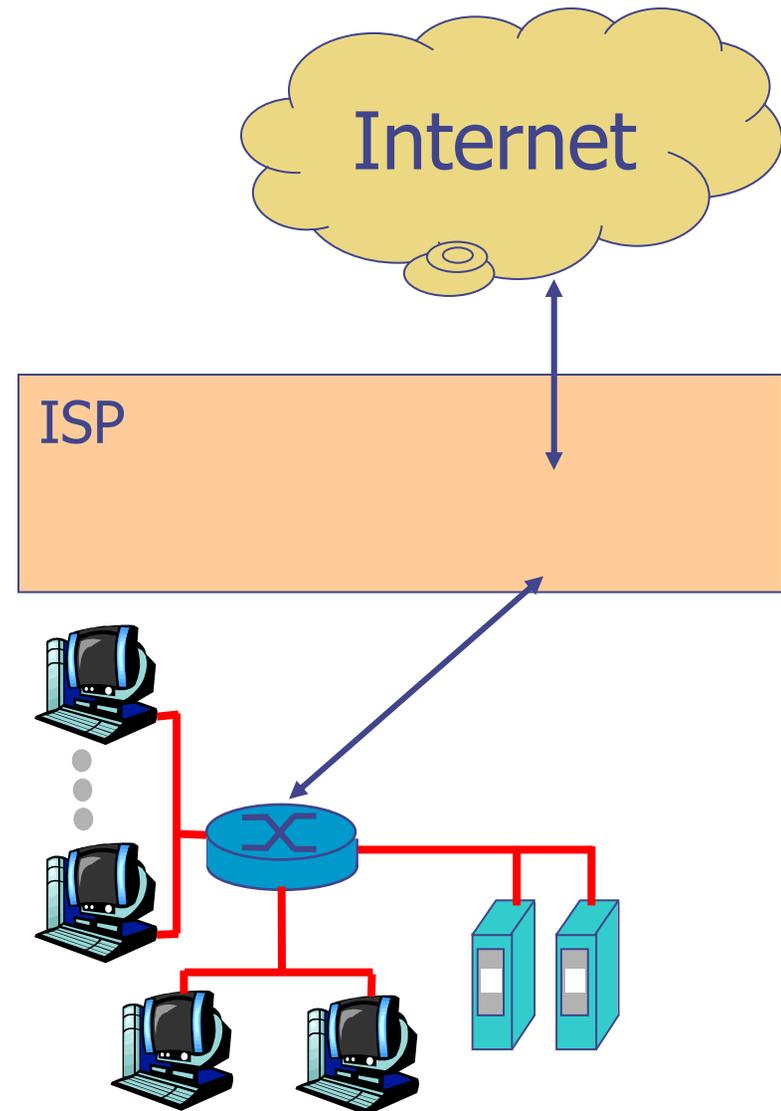
Personal Computer +
Modem



- Modem analogico
 - fino a 56Kbps
- ISDN (*Integrated Services Digital Network*)
 - fino a 128Kbps
- ADSL (*Asymmetric Digital Subscriber Line*)
 - fino ad 1 Mbps upstream e fino ad 8 Mbps downstream

Collegamento ad Internet di LAN aziendali

- ◆ Collegamento mediante:
 - Modem analogico, ISDN, ADSL
 - ◆ Per reti aziendali di piccole dimensioni
 - Linea dedicata con collegamento permanente con l'ISP
 - ◆ Per reti aziendali di medie/grandi dimensioni



Chi regola Internet

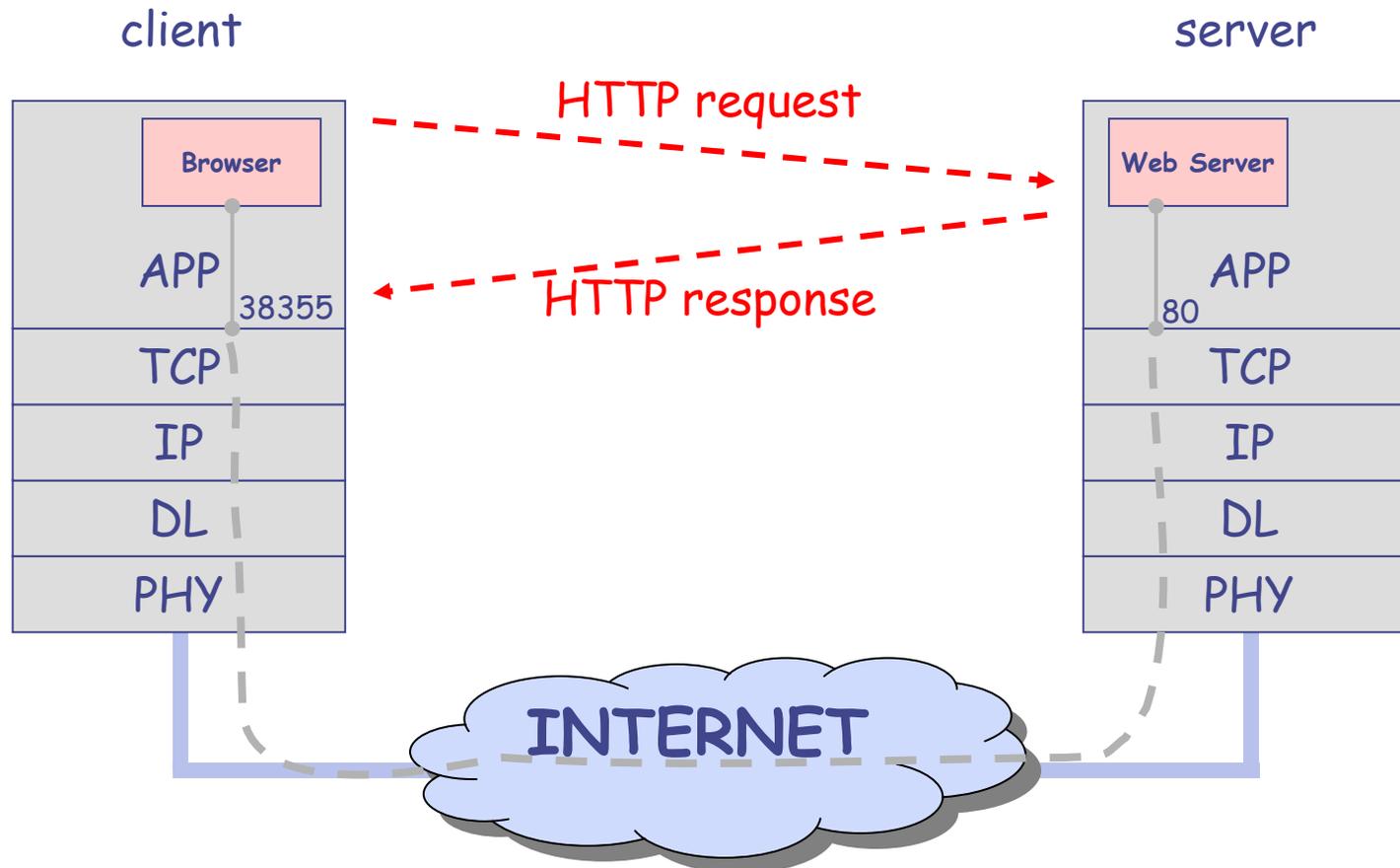
- La standardizzazione dei protocolli in uso su Internet è fatta dall'*Internet Engineering Task Force, IETF*
- L'assegnazione degli indirizzi e dei nomi di dominio è oggi supervisionata dall' *Internet Corporation for Assigned Names and Numbers, ICANN*, che ha preso il posto dell' *Internet Assigned Numbers Authority, IANA*, una authority federale degli USA

Lo stack TCP/IP

- Internet si basa su un modello definito da una collezione di protocolli standardizzati dall'IETF, il modello TCP/IP
- Siccome i protocolli sono organizzati secondo una struttura a pila (stack), si parla dello "stack TCP/IP"
- Il modello prende il nome da due protocolli fondamentali:
 - TCP, Transmission Control Protocol, di livello Trasporto
 - IP, Internet Protocol, di livello Rete



Web: interazione Client→Server



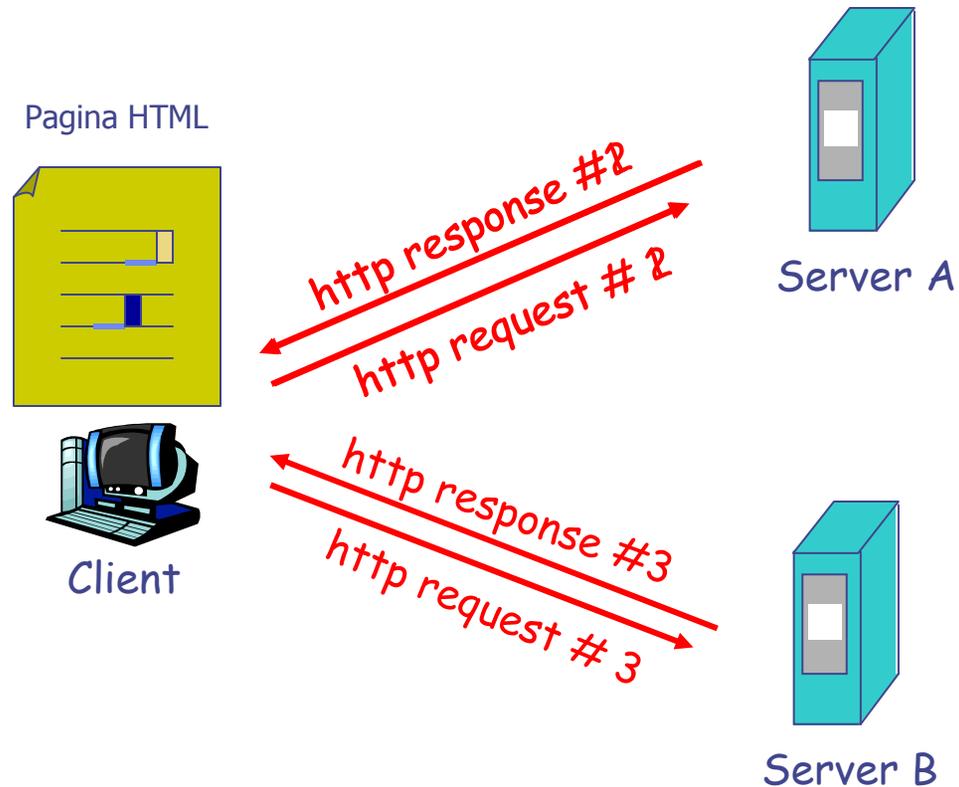
Il protocollo HTTP

- ◆ Per effettuare richieste, specificare cosa si richiede, rispondere alle richieste, si rende necessario un opportuno protocollo.
- ◆ Su Internet si usa il protocollo HTTP
- ◆ E' un protocollo testuale
- ◆ I messaggi sono costituiti da sequenze di byte
- ◆ Ogni byte identifica un carattere secondo la tabella ASCII
- ◆ In certi casi, il payload dei messaggi può essere comunque anche in formato binario.

HTTP per il trasferimento di pagine web

- ◆ Tipicamente, una pagina web è descritta da un file testuale in formato HTML (*Hypertext Markup Language*)
- ◆ La pagina è identificata mediante un indirizzo, detto URL
- ◆ Un file HTML può contenere riferimenti ad altri oggetti che arricchiscono la pagina con elementi grafici
 - Es. sfondo, immagini, ecc.
- ◆ Ciascun oggetto è identificato dal proprio URL
- ◆ Questi oggetti possono trovarsi anche su server web diversi
- ◆ Una volta ricevuta la pagina HTML, il browser estrae i riferimenti agli altri oggetti che devono essere prelevati e li richiede attraverso una serie di connessioni HTTP

HTTP per il trasferimento di pagine web (2)



Connessioni persistenti e non persistenti

non persistente

- ◆ HTTP/1.0
- ◆ Il server analizza una richiesta, la serve e chiude la connessione
- ◆ 2 Round Trip Time (RTT) per ciascuna richiesta

persistente

- ◆ HTTP/1.1
- ◆ Sulla stessa connessione il server analizza tutte le richieste e le serve
- ◆ Il client riceve la pagina iniziale e invia subito tutte le altre richieste
- ◆ Si hanno meno RTTs

URL

- ◆ Un URL HTTP ha la seguente sintassi:

`http://host[:port]/path[#fragment][?query]`

- ◆ Host identifica il server

- Può essere sia un nome simbolico che un indirizzo IP in notazione dotted decimal

- ◆ Port è opzionale; di default è 80

- ◆ Path identifica la risorsa sul server

- es: `images/sfondo.gif`

- ◆ #fragment identifica un punto preciso all'interno di un oggetto

- es: `#paragrafo1`

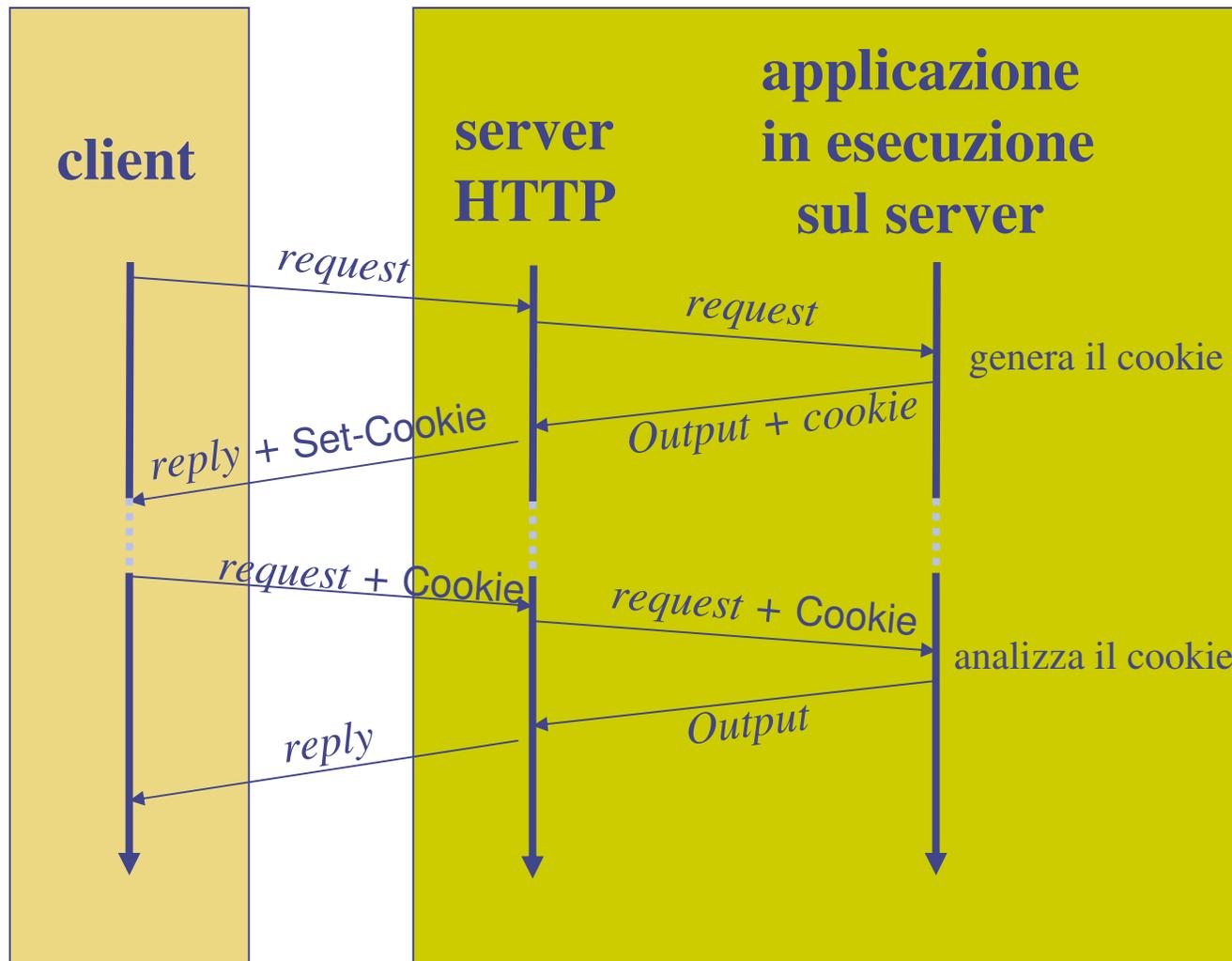
- ◆ ?query è usato per passare informazioni dal client al server

- es: dati inseriti nei campi di una form

I cookies

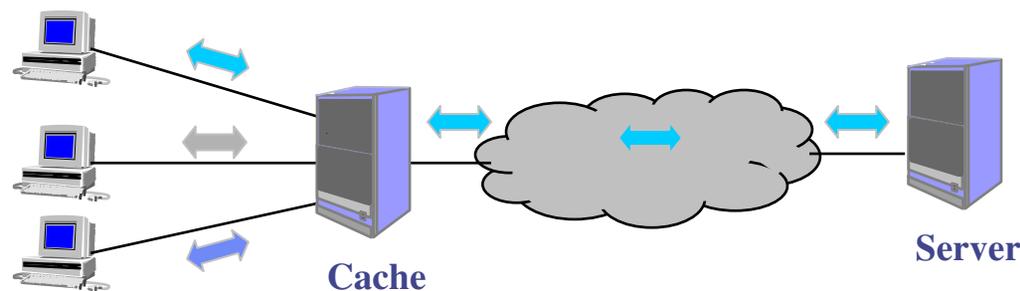
- ◆ **HTTP è stateless:** il server non è tenuto a mantenere informazioni su connessioni precedenti
- ◆ Un **cookie** è una breve informazione scambiata tra il server ed il client
- ◆ Tramite un cookie il client mantiene lo stato di precedenti connessioni, e lo manda al server di pertinenza ogni volta che richiede un documento
- ◆ Esempio: tramite un cookie si viene rediretti sulla pagina in Italiano tutte le volte che ci si ricollega allo stesso server
(es. www.google.com)
- ◆ I cookies sono definiti in RFC 2108 (su proposta di Netscape)

I cookies (2)



Web caching

- ◆ Si parla genericamente di Web caching quando le richieste di un determinato client non raggiungono il Web Server, ma vengono intercettate da una cache.
- ◆ Tipicamente, un certo numero di client di una stessa rete condivide una stessa cache web, posta nelle loro prossimità (es. nella stessa LAN).
- ◆ Se l'oggetto richiesto non è presente nella cache, questa lo richiede *invece* del client conservandone una copia per eventuali richieste successive.
- ◆ Richieste successive alla prima sono servite più rapidamente.
- ◆ Due tipi di interazione HTTP: client-cache e cache-server.



Il protocollo SMTP

- ◆ Una volta che una e-mail è stata scritta attraverso l'uso di un programma su un personal computer, è necessario inviarla al destinatario
- ◆ Come è noto, il destinatario potrebbe non essere in quel momento disponibile ad accettare messaggi di posta:
 - utente impegnato
 - computer spento
- ◆ La posta elettronica sfrutta degli intermediari per il trasferimento delle e-mail tra le parti, alla stregua degli uffici postali che ospitano pacchi nell'attesa che i destinatari passino a ritirarli
- ◆ Per trasferire messaggi di posta elettronica tra gli intermediari si utilizza un apposito protocollo
- ◆ Si chiama Simple Mail Transfer Protocol, definito in RFC821

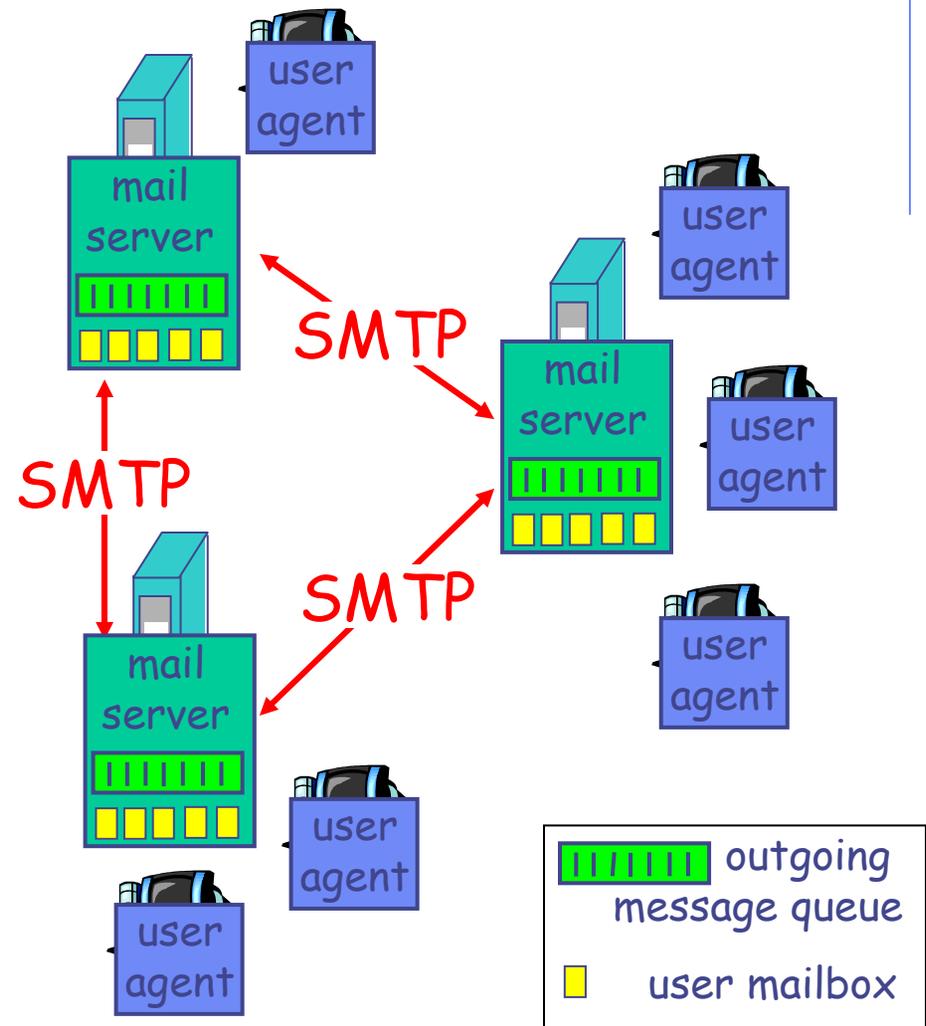
Le entità in gioco

Tre entità principali:

- ◆ user agents
- ◆ mail servers
- ◆ protocollo SMTP

User Agent

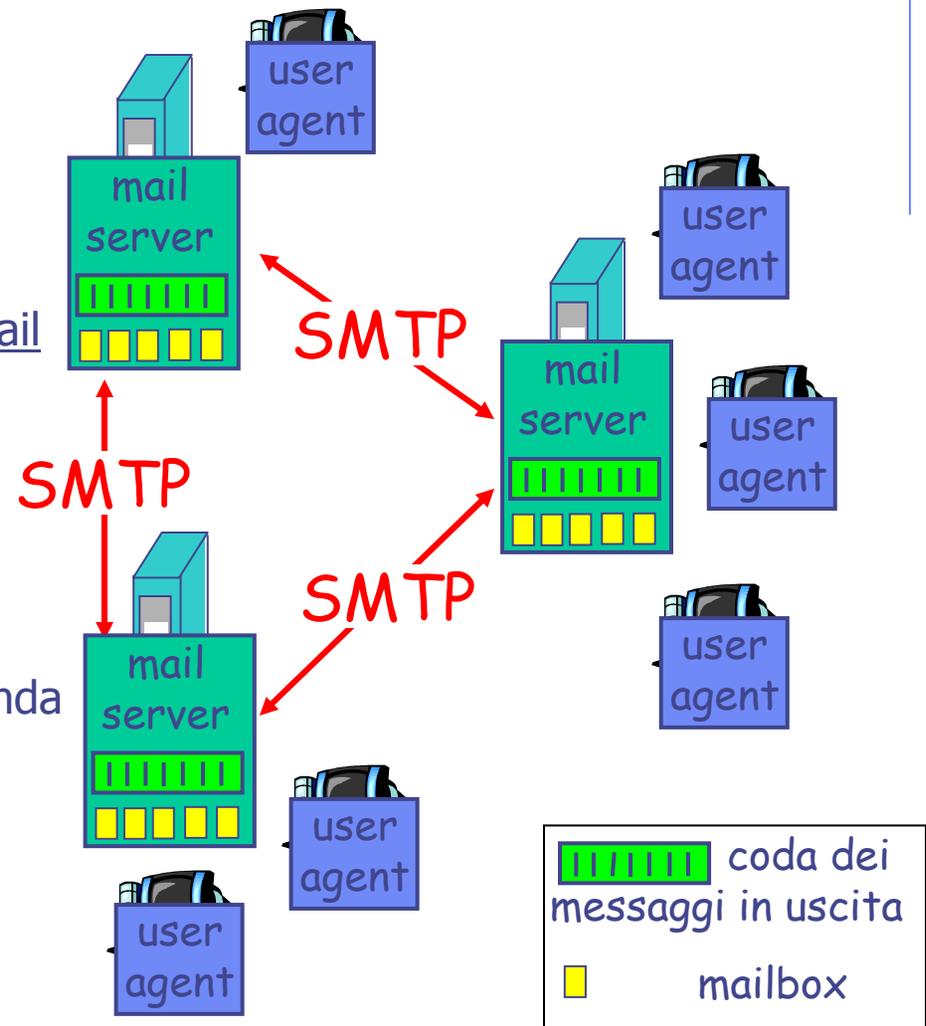
- ◆ anche detto mail reader
- ◆ composizione, modifica, lettura di messaggi
- ◆ es.: Eudora, Outlook, Netscape Messenger
- ◆ messaggi in uscita ed in entrata immagazzinati sul server



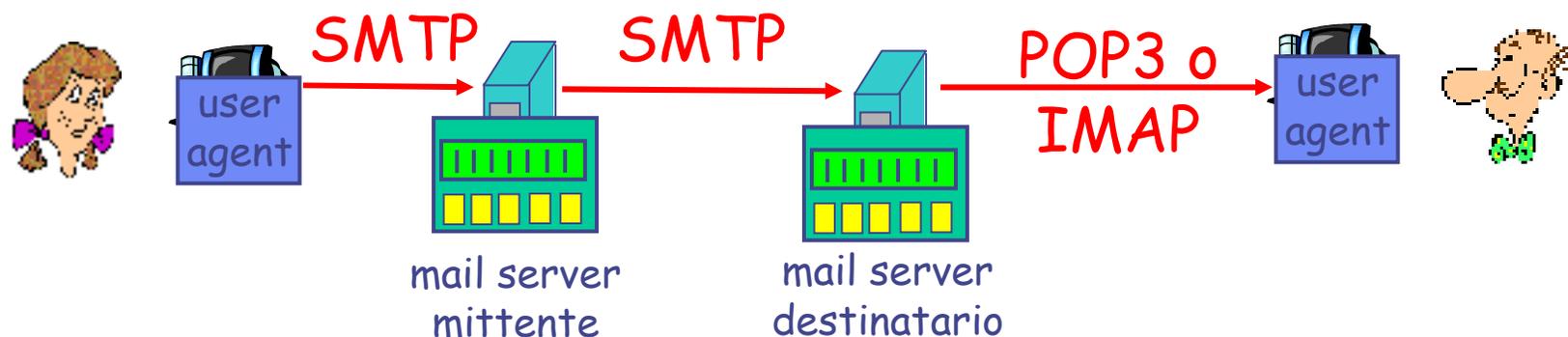
I mail server

Mail Server

- ◆ **mailbox** contenente messaggi in entrata (non letti) per l'utente
- ◆ **coda dei messaggi in uscita** contenente i messaggi non ancora recapitati
- ◆ **protocollo SMTP** a cavallo di due mail server.
 - "client": mail server mittente
 - "server": mail server destinatario
- ◆ Un "mail server" funge in momenti diversi da client o da server a seconda del ruolo che ricopre nello scambio del messaggio



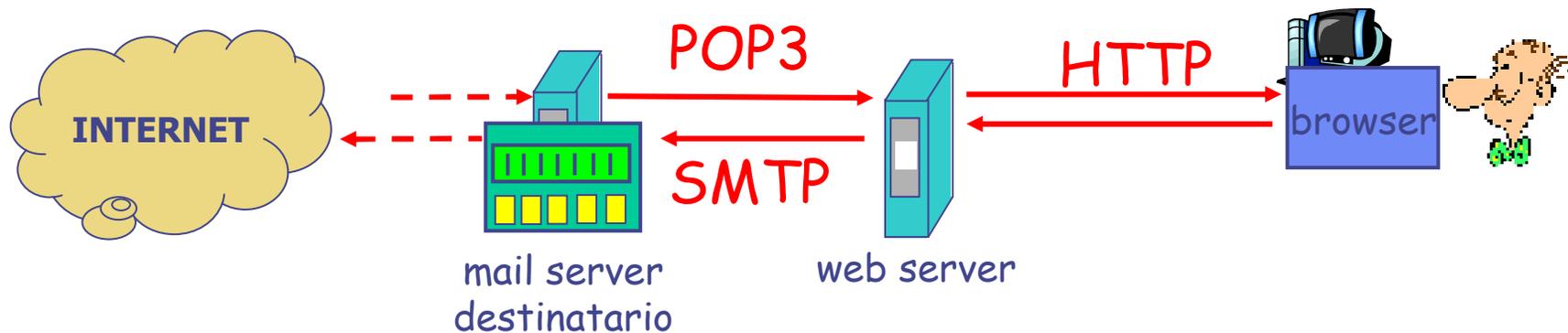
La catena dei protocolli per la posta



- ◆ SMTP: consegna di messaggi
- ◆ Protocolli di accesso alla mail: recupero dei messaggi dai server
 - POP: Post Office Protocol
 - ◆ autorizzazione (agent \leftrightarrow server) e download
 - IMAP: Internet Mail Access Protocol [RFC 1730]
 - ◆ più complicato e potente
 - ◆ manipolazione dei messaggi sul server

L'accesso alla posta via WEB

- ◆ Molti siti web forniscono accesso alle proprie caselle di posta (libero, inwind, tin, hotmail, Yahoo!, ecc.)
- ◆ In questo caso non serve avere uno user agent installato e correttamente configurato per ricevere ed inviare posta.
- ◆ È sufficiente disporre di un qualsiasi browser
 - HTTP: Hotmail , Yahoo! Mail, ecc.



Domain Name System (DNS)

- ◆ Tutti noi siamo oggi abituati a raggiungere un servizio (e quindi il calcolatore che lo offre) utilizzando nomi simbolici di facile memorizzazione:
 - www.google.com
 - www.rai.it
 - pippo@unina.it
- ◆ Questi nomi non sono immediatamente adatti ad essere compresi dai dispositivi che costituiscono la rete Internet.
- ◆ Un nome di questo tipo, infatti, non dà informazioni esatte sulla dislocazione sul territorio della macchina che si desidera contattare.
- ◆ I router, di conseguenza, non saprebbero come instradare i dati in maniera tale da raggiungere la destinazione.

Nomi simbolici vs Indirizzi IP

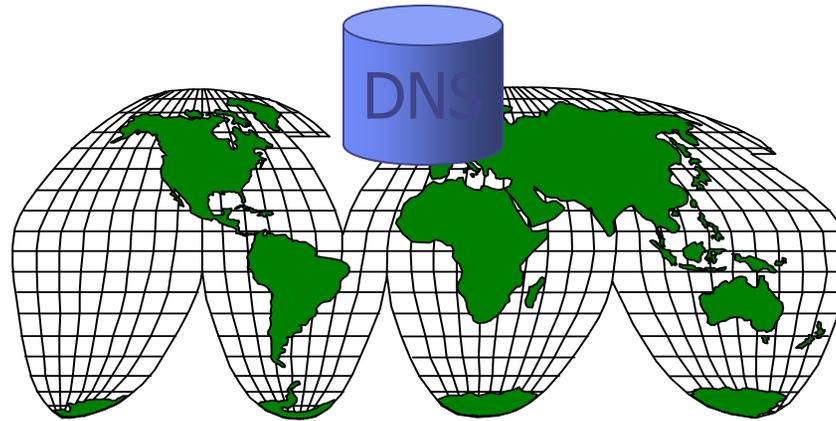
- ◆ La rete Internet è stata progettata invece per lavorare con indirizzi di diversa natura. Per es.:
 - 143.225.229.3
 - 217.9.64.225
- ◆ Questi indirizzi, detti indirizzi IP, sono formati da 4 numeri che vanno da 0 a 255 separati da un punto.
- ◆ Ogni dispositivo nella rete Internet ha un tale indirizzo; esso permette l'identificazione univoca a livello globale e la localizzazione.
- ◆ A differenza dei nomi simbolici, essendo gli indirizzi IP di lunghezza fissa, sono più facilmente gestibili dalle macchine
- ◆ Data la loro natura gerarchica forniscono inoltre precise informazioni sulla loro dislocazione sul territorio:
 - leggendo l'indirizzo da sinistra verso destra si ottengono informazioni sempre più dettagliate circa la rete di appartenenza (all'interno di tutte le reti di Internet)

Il servizio DNS

- ◆ Non volendo rinunciare alla comodità di lavorare con nomi simbolici, è stato necessario progettare un servizio di risoluzione dei nomi simbolici in indirizzi IP.
- ◆ Tale servizio associa ad un nome simbolico univoco (www.grid.unina.it) un indirizzo IP (143.225.229.3) permettendo così di raggiungere la macchina.
- ◆ Questo servizio si chiama **Domain Name System (DNS)** ed è definito in RFC1034 e RFC1035.

DNS centralizzato?

- ◆ Si potrebbe pensare di risolvere il problema piazzando in un unico punto della terra una macchina che realizzi la risoluzione di tutti i nomi.



- ◆ Questa soluzione, sebbene teoricamente realizzabile, ha così tanti svantaggi da risultare impraticabile:
 - Single Point of Failure
 - Volume di traffico
 - Database distante
 - Manutenzione

DNS distribuito!

- ◆ Quello che si fa è distribuire le informazioni sul territorio
- ◆ Ciascuno ha la responsabilità di raccogliere, gestire, aggiornare e divulgare le informazioni che lo riguardano.
- ◆ In particolare l'approccio è di tipo gerarchico:
 - gli elementi più alti nella gerarchia contengono molte informazioni non dettagliate
 - gli elementi più bassi nella gerarchia contengono poche informazioni dettagliate
- ◆ Attraverso un colloquio concertato tra le entità (di cui gli utenti non hanno percezione) si riesce a fornire il servizio di risoluzione.

Tipologie di server DNS (Local)

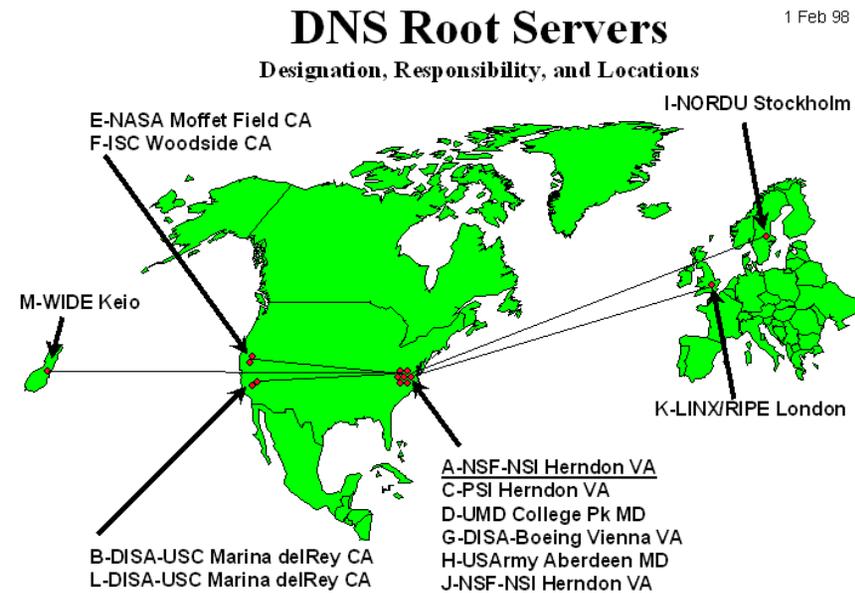
◆ Local Name Server (Locale)

- ciascun ente (università, società, etc...) ne installa uno nel proprio dominio.
- Tutti gli host nel dominio inviano a questo server le proprie richieste di risoluzione.
- Ciascun host deve essere configurato con l'indirizzo del DNS server locale per il dominio. Questa configurazione spesso avviene manualmente, ma in certi casi può avvenire anche in maniera automatica.

Tipologie di server DNS (Root)

◆ Root Name Server (Fondamentale)

- Ne esistono varie decine in Internet e i loro indirizzi sono ben noti alla comunità.
- Ad essi si riferiscono i Local Name Server che non possono soddisfare immediatamente una richiesta di risoluzione.
- In questo caso il Local Name Server si comporta come client DNS ed invia una richiesta di risoluzione al Root Name Server.



Tipologie di server DNS (Authoritative)

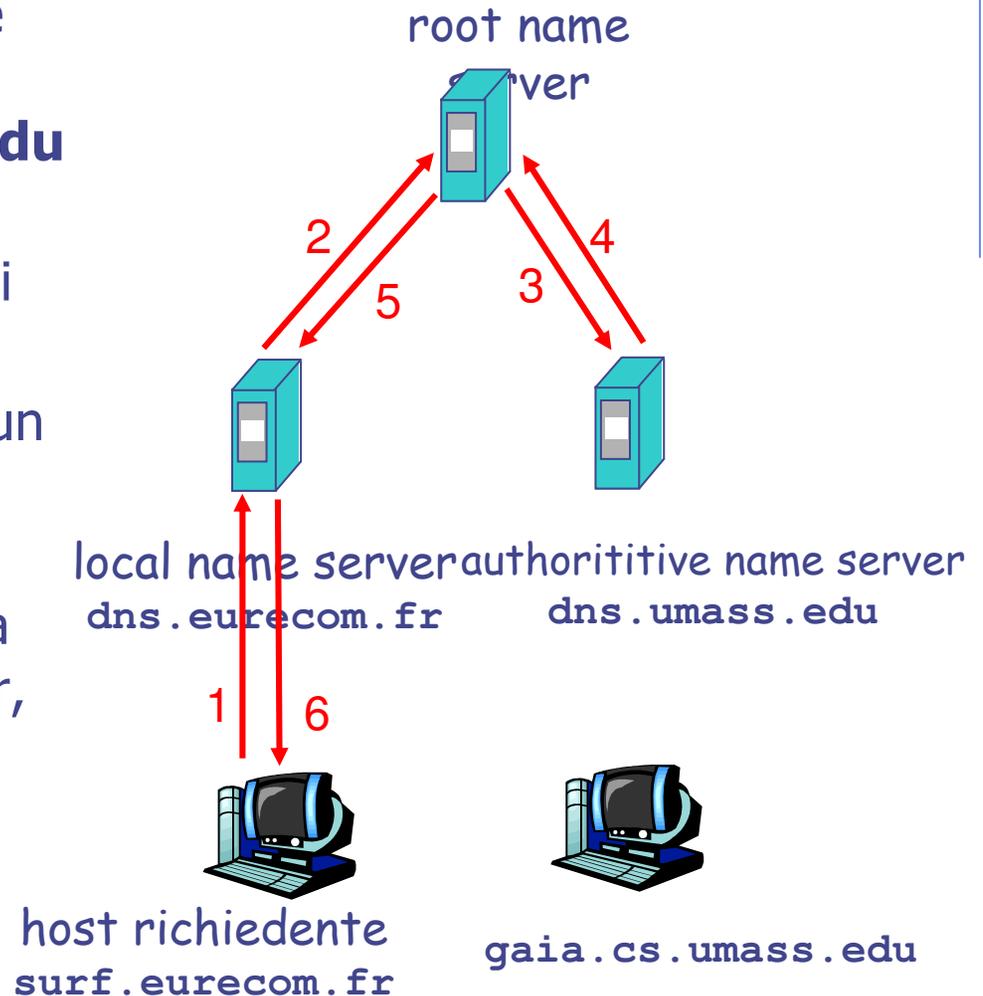
◆ Authoritative Name Server (Assoluto)

- È un server dei nomi capace di risolvere tutti i nomi all'interno di un dominio.
 - ◆ P.es.: un server dei nomi assoluto per il dominio unina.it deve essere capace di risolvere tutti i nomi del tipo xyz.unina.it
- Ad essi si riferiscono i Root Name Server quando, interpellati dai Local Name Server, devono risolvere un indirizzo.

Un semplice esempio

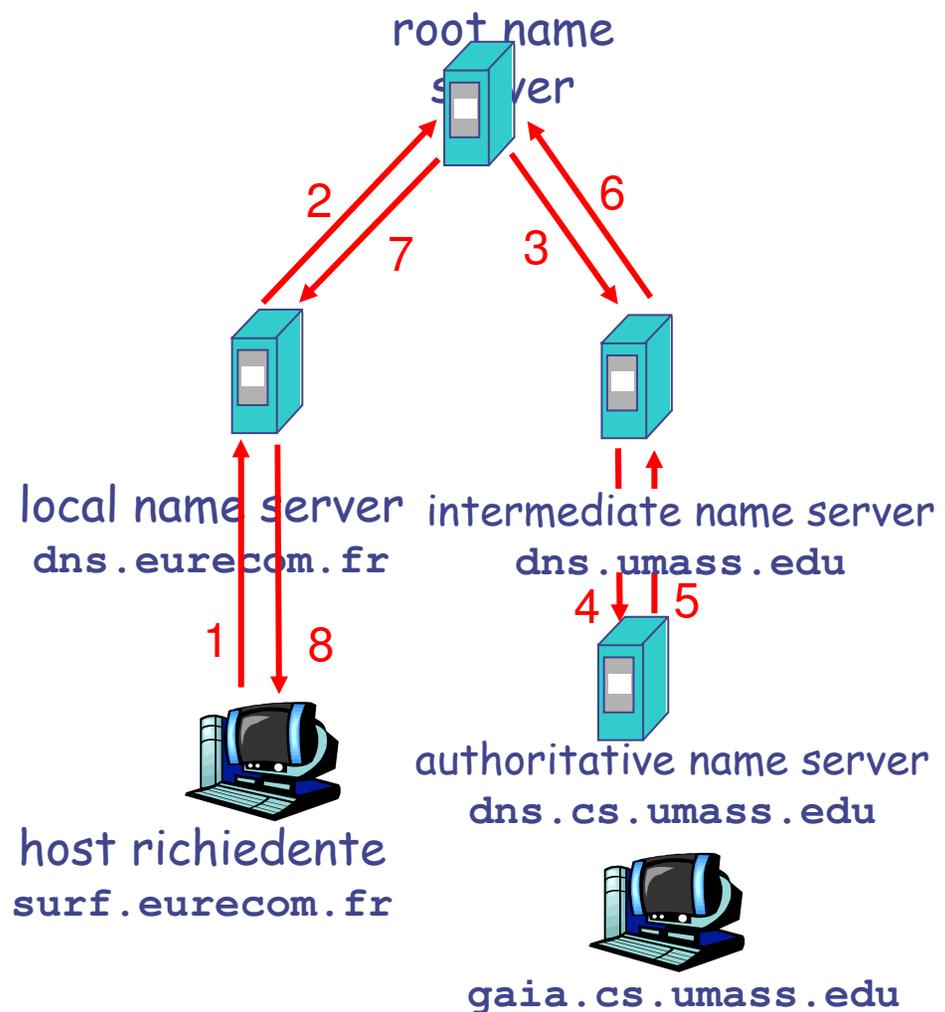
L'host **surf.eurecom.fr** vuole conoscere l'indirizzo IP dell'host **gaia.cs.umass.edu**

1. Contatta il suo Local Name Server: **dns.eurecom.fr** di cui possiede l'indirizzo IP
2. **dns.eurecom.fr** contatta un root name server, se necessario
3. il root name server contatta l'Authoritative Name Server, **dns.umass.edu**, se necessario.



Un esempio a più livelli

Il Root Name Server potrebbe non contattare necessariamente l'Authoritative Name Server finale, ma un Authoritative Name Server intermedio



Richieste iterative e ricorsive

Richiesta Ricorsiva:

- ◆ delega al NS contattato la responsabilità di risolvere l'indirizzo
- ◆ troppo carico? (p.es.: su un root NS)

Richiesta Iterativa:

- ◆ il server contattato risponde con il nome del server da contattare
- ◆ "Non so risolvere, ma chiedi a quest'altro server"