

Introduzione alle reti di calcolatori: Internet

Rossano Gaeta
Dipartimento di Informatica
Università di Torino

Dati docente

- Rossano Gaeta
- E-mail: rossano@di.unito.it
- www.di.unito.it/~rossano/DIDATTICA/bioinfo

Testo consigliato

- ❑ Computer networking
A Top-Down approach featuring the Internet
- ❑ James F.Kurose - Keith W.Ross
- ❑ Addison-Wesley

- ❑ Dovrebbe esistere anche una versione in italiano
- ❑ Esiste la versione elettronica

Obiettivi del seminario

- Introduzione alle reti di calcolatori usando Internet come esempio

Introduzione

Obiettivo:

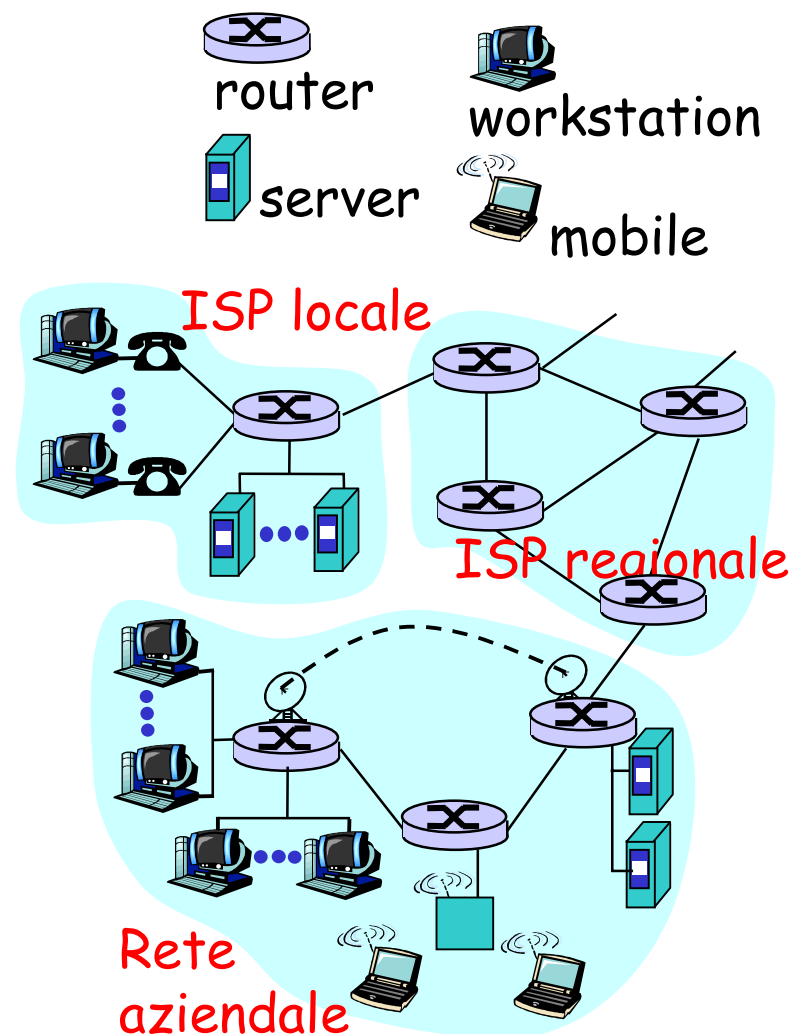
- ❑ Introduzione alle reti di telecomunicazioni
- ❑ approccio:
 - descrittivo
 - uso di Internet come esempio

Sommario:

- ❑ Cos'è Internet
- ❑ Cos'è un protocollo?
- ❑ network edge
- ❑ network core
- ❑ Reti di accesso, mezzi trasmissivi
- ❑ Protocolli a livelli
- ❑ backbones, NAP, ISP
- ❑ storia

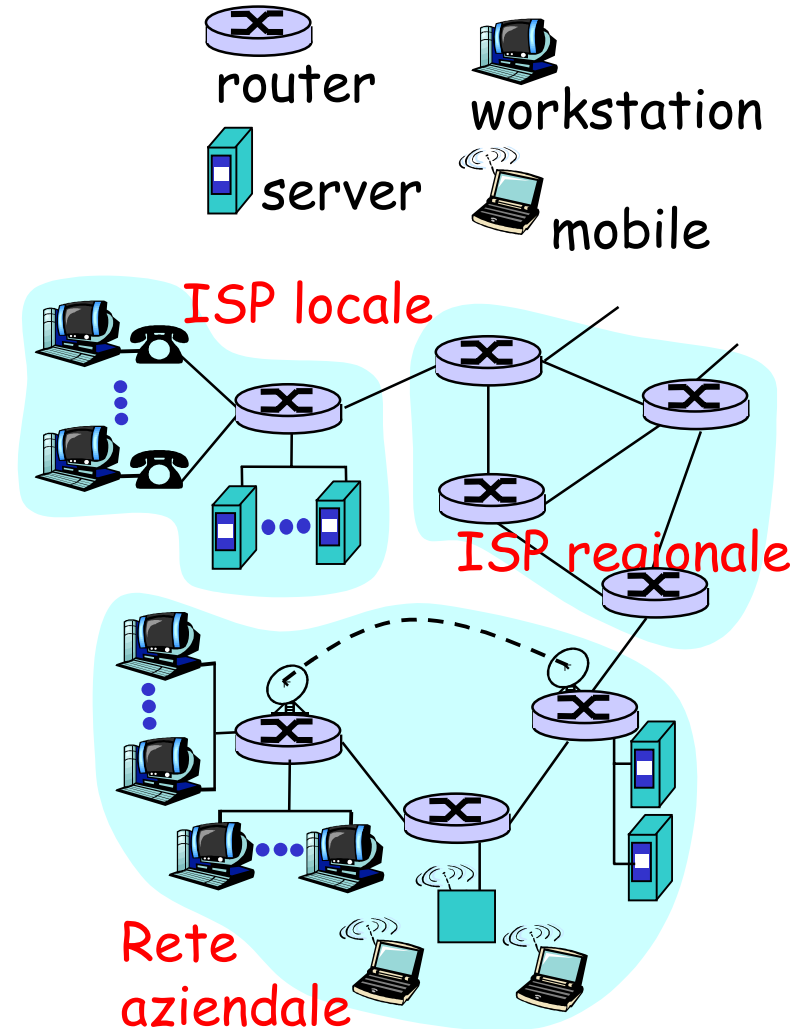
Cos'è Internet?

- ❑ Milioni di dispositivi di calcolo tra loro interconnessi: *host, end-systems*
 - Pc, workstation, server
 - PDA's phones, toasters
- Che eseguono *applicazioni di rete*
- ❑ *Canali di comunicazione*
 - fibra, rame, radio, satellite
- ❑ *Router*: instradano pacchetti di dati attraverso la rete



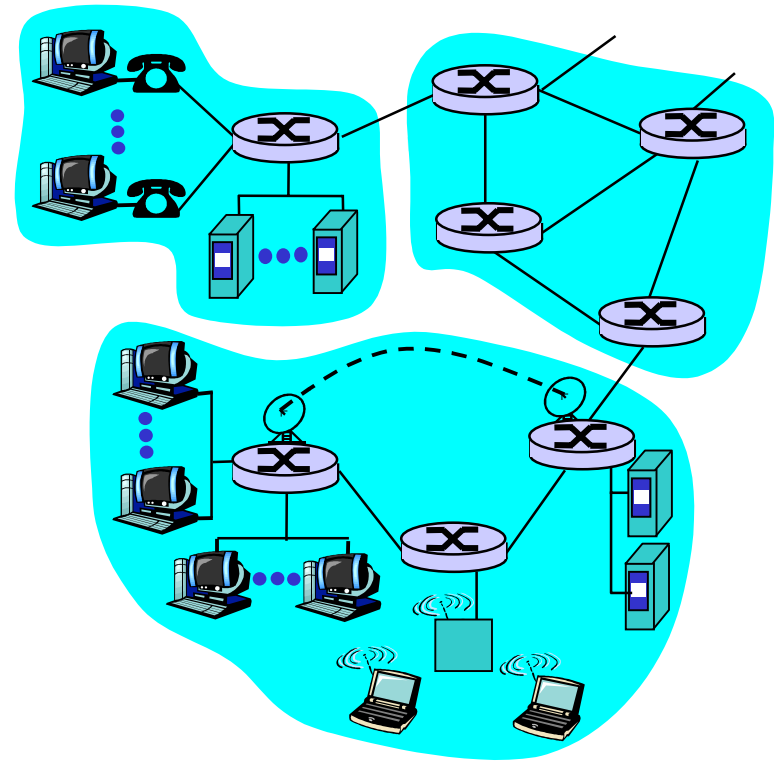
Cos'è Internet?

- **protocolli:** controllano la spedizione e la ricezione di messaggi
 - e.g., TCP, IP, HTTP, FTP, PPP
- **Internet: "rete di reti"**
 - Debolmente gerarchica
 - Internet pubblica vs intranet private
- **Standard di Internet**
 - RFC: Request for comments
 - IETF: Internet Engineering Task Force



Cos'è Internet: accento sui servizi

- **Infrastruttura di comunicazione** che consente ad applicazioni distribuite lo scambio di dati:
 - WWW, email, giochi, e-commerce, database, file (MP3) sharing
- **Servizi di comunicazione** forniti:
 - connectionless
 - connection-oriented



Cos'è un protocollo?

Protocolli umani:

- ❑ "Che ora è?"
- ❑ "Ho una domanda"
- ❑ Presentazioni...

... messaggi specifici vengono spediti

... azioni specifiche sono compiute quando i messaggi sono ricevuti, o in seguito ad altri eventi

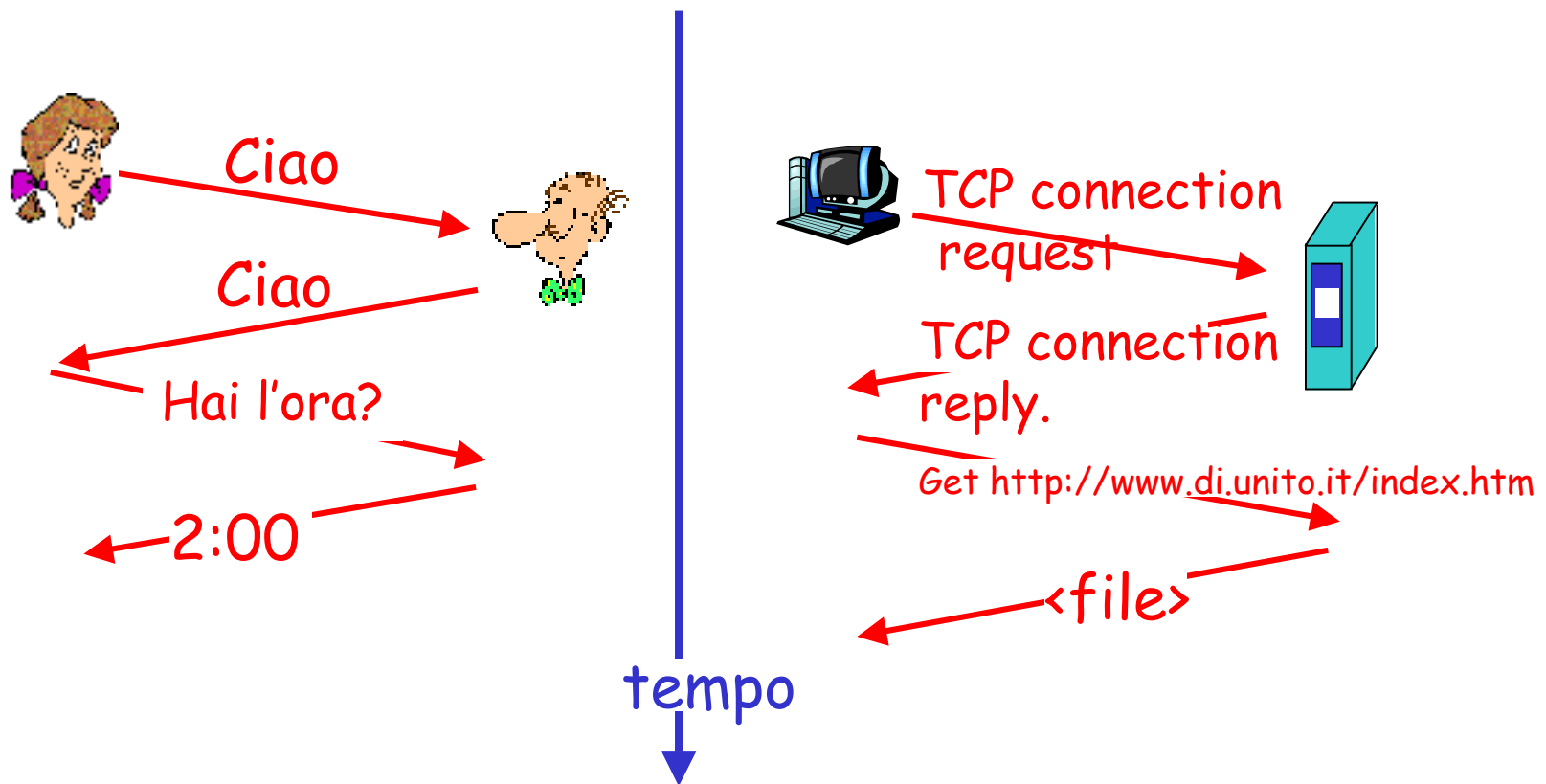
Protocolli di rete:

- ❑ macchine invece di esseri umani
- ❑ Tutte le attività di comunicazione in Internet sono governate da protocolli

I protocolli definiscono formato e ordine dei messaggi spediti e ricevuti tra entità della rete, e le azioni da compiere in seguito alla ricezione e/o trasmissione dei messaggi o di altri eventi

Cos'è un protocollo?

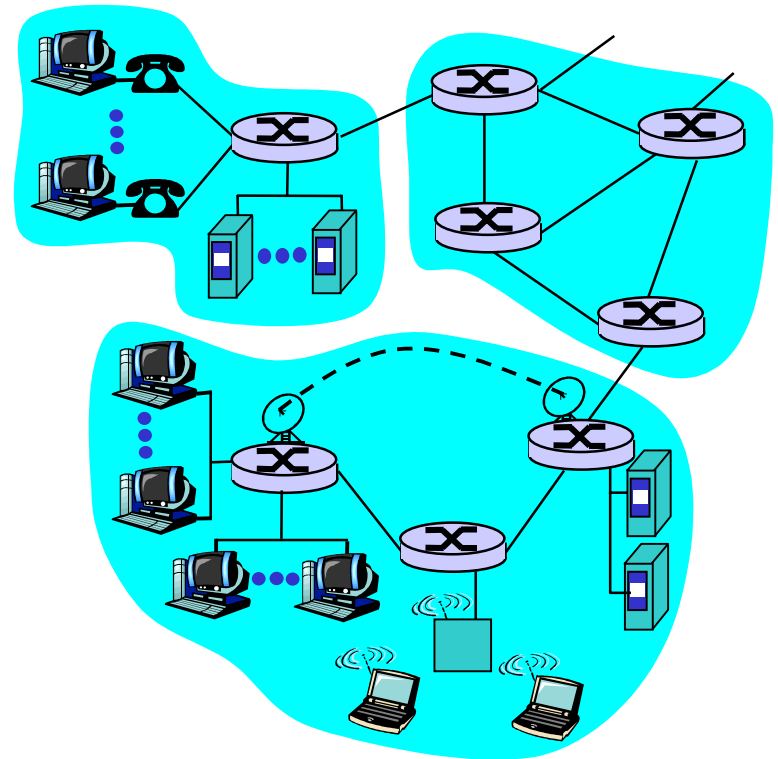
Un protocollo umano e un protocollo di reti di computer:



Domanda: Altri protocolli umani?

Struttura della rete:

- ❑ **network edge:** applicazioni ed host
- ❑ **network core:**
 - router
 - rete di reti
- ❑ **reti di accesso, mezzi trasmissivi:** canali di comunicazione



La edge network:

□ end systems (host):

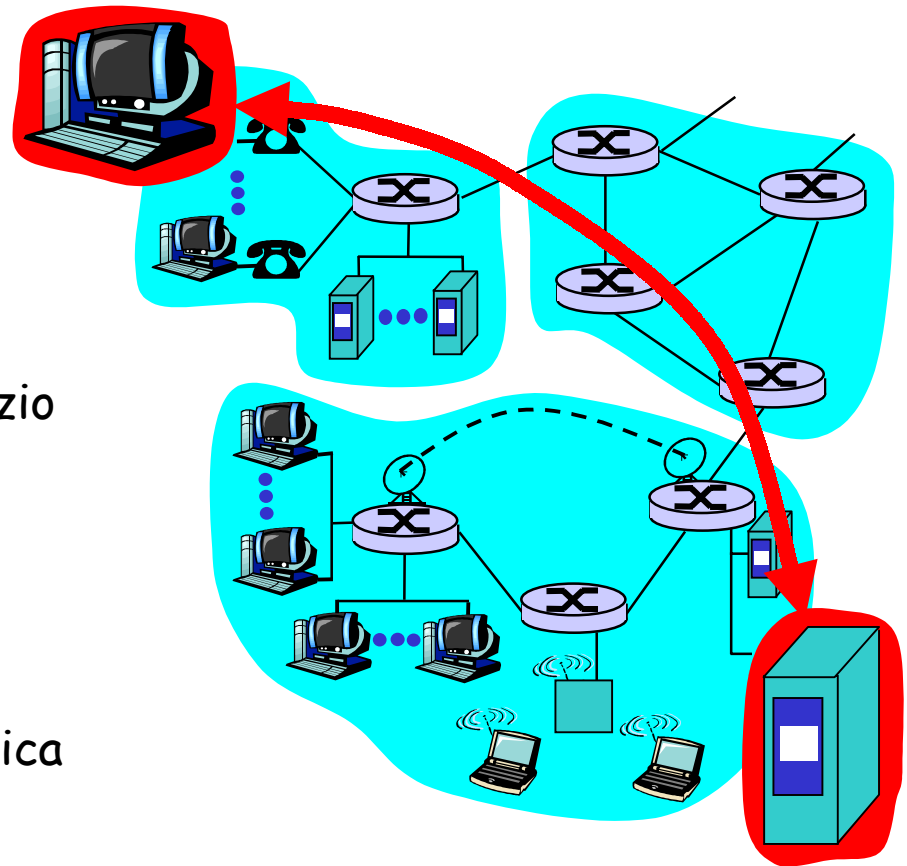
- Eseguono programmi applicativi
- e.g., WWW, email
- al "bordo della rete"

□ modello client/server

- il client richiede, riceve servizio dal server
- e.g., WWW client (browser)/server; email client/server

□ modello peer-peer:

- interazione tra host simmetrica
- e.g.: Gnutella, KaZaA



La edge Network: servizi connection-oriented

Obiettivo: trasferimento dati tra host

- *handshaking*: fase di preparazione antecedente al trasferimento dati
 - Ciao - Ciao nel protocollo umano
 - *Stabilire uno "stato"* nei due host comunicanti
- TCP - Transmission Control Protocol
 - Servizio di scambio dati di tipo connection-oriented di Internet

Servizio TCP [RFC 793]

- *Trasferimento affidabile ed ordinato di byte di un flusso dati*
 - perdite: conferma di ricezione (acknowledgement) e ritrasmissione
- *Controllo di flusso*
 - Il mittente non sovraccaricherà il ricevitore
- *Controllo di congestione:*
 - I mittenti diminuiscono la loro velocità di spedizione quando la rete si congestiona

La edge Network: servizi connectionless

Obiettivo: trasferimento dati tra host

- o Esattamente lo stesso!
- **UDP** - User Datagram Protocol [RFC 768]: Servizio connectionless di Internet
 - o Senza handshaking
 - o Trasferimento dati non-affidabile
 - o senza controllo di flusso
 - o senza controllo congestione

Applicazioni che usano TCP:

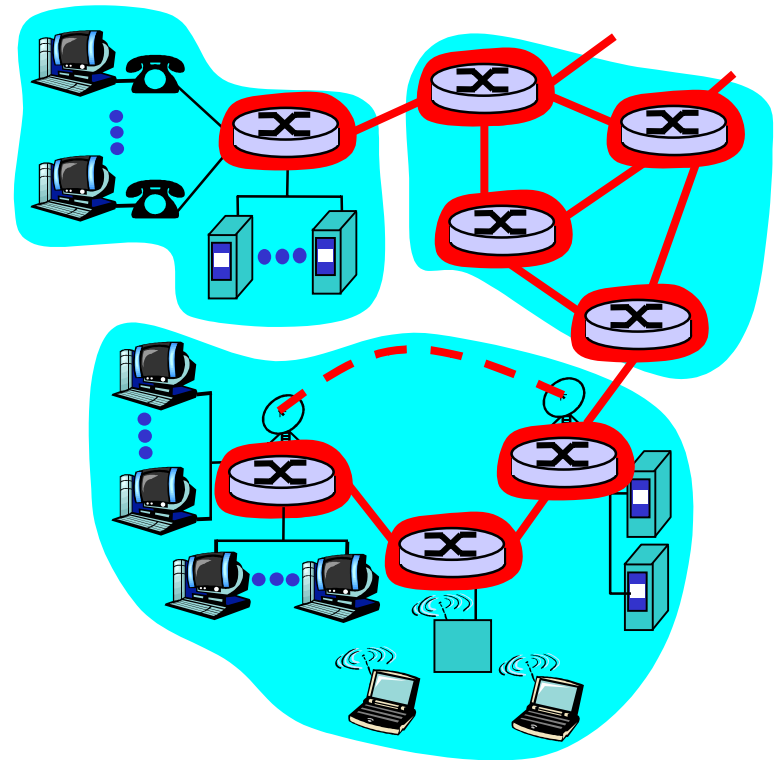
- HTTP (WWW), FTP (trasferimento file), Telnet (login remoto), SMTP (email)

Applicazioni che usano UDP:

- streaming media, teleconferencing, Internet telephony

La Core Network

- ❑ Maglia di router interconnessi
- ❑ Domanda fondamentale: come vengono trasferiti i dati attraverso la rete?
 - **Commutazione di pacchetto**: i dati sono spediti attraverso la rete in quantità discrete chiamate **pacchetti**



La Core Network: commutazione di Pacchetto

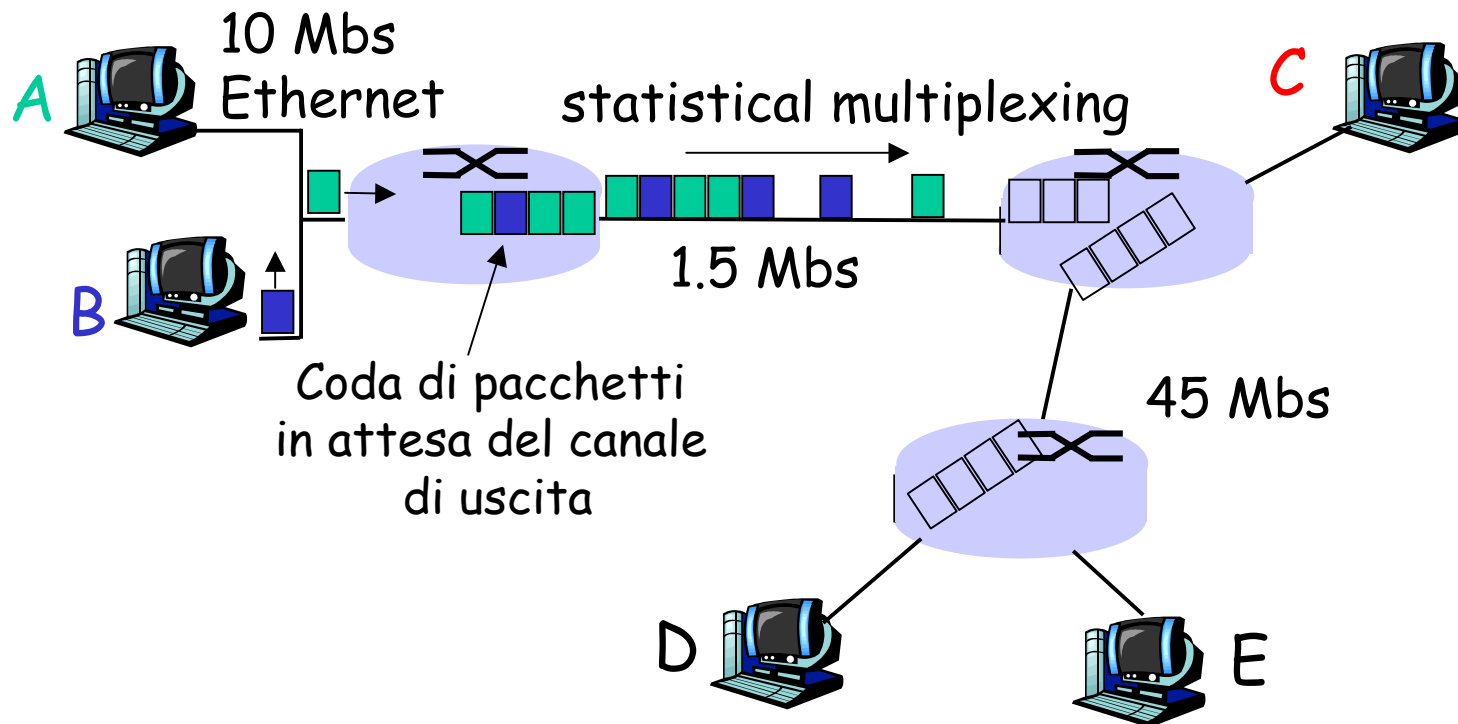
Ogni flusso dati end-end diviso in *pacchetti*

- ❑ I pacchetti degli utenti A e B *condividono* risorse di rete
- ❑ Ogni pacchetto usa tutta la larghezza di banda del canale
- ❑ Risorse usate quando sono necessarie

Contesa delle risorse:

- ❑ La richiesta aggregata di risorse può eccedere l'ammontare disponibile
- ❑ congestione: i pacchetti si accodano ed attendono l'uso del canale
- ❑ store and forward: pacchetti ricevuti interamente prima di essere spediti

La core Network: commutazione di pacchetto



Reti a commutazione di pacchetto: routing

- Obiettivo: spostare pacchetti tra router, dal host sorgente all' host destinatario
- **Reti datagram**:
 - *L'indirizzo destinazione* determina il prossimo passo
 - Le strade (route) possono variare durante le sessioni
 - I router NON mantengono informazioni sullo stato delle connessioni

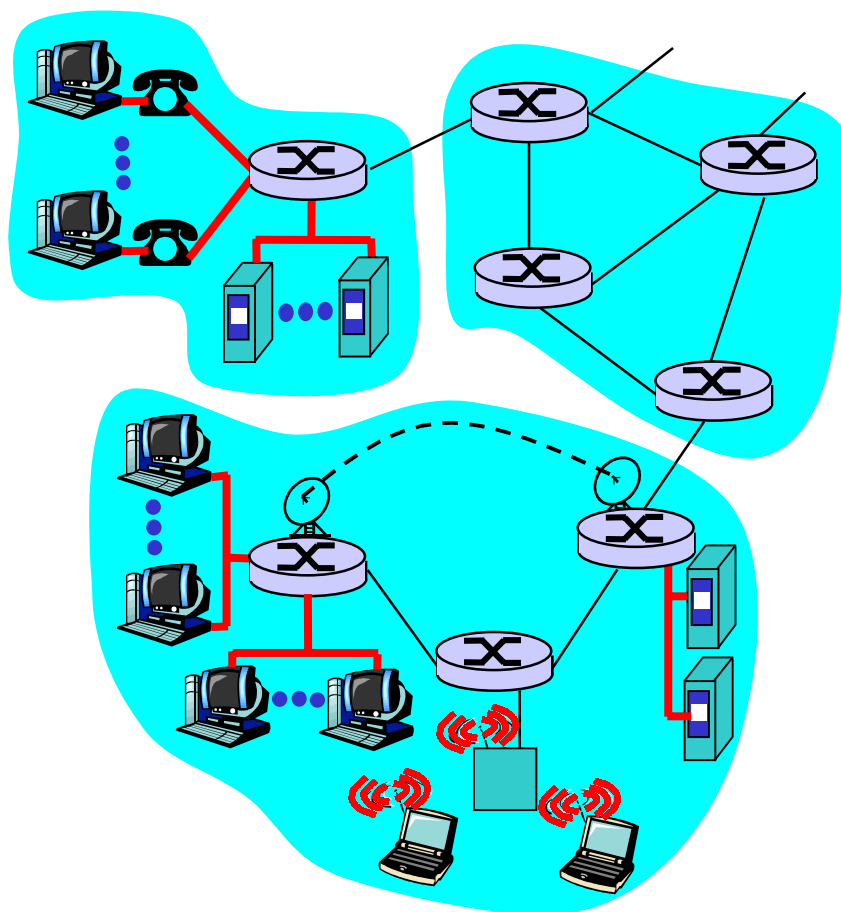
Reti di accesso e mezzi trasmissivi

Domanda: come si connettono gli host agli edge router?

- ❑ Reti di accesso residenziale (da casa)
- ❑ Reti di accesso istituzionali (scuole, università, aziende)
- ❑ Reti di accesso mobili

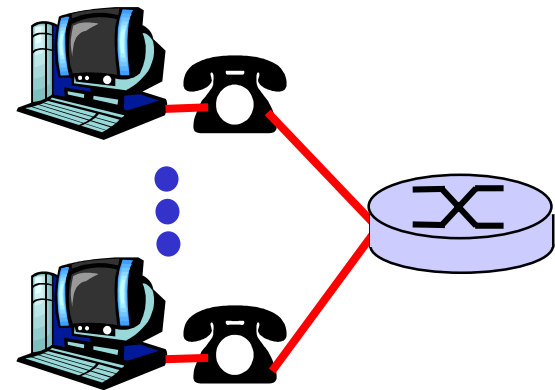
Caratteristiche:

- ❑ Larghezza di banda (bit al secondo) delle reti di accesso
- ❑ Condivise o dedicate?



Accesso Residenziale: accesso point to point

- **Connessione telefonica via modem**
 - Fino a 56Kbps di accesso diretto ad un router (in teoria)
- **ISDN: integrated services digital network: connessione completamente digitale a 128Kbps verso un router**
- **ADSL: asymmetric digital subscriber line**
 - Fino a 1 Mbps casa-router
 - Fino a 8 Mbps router-casa
 - Diffusione ADSL: **in corso**

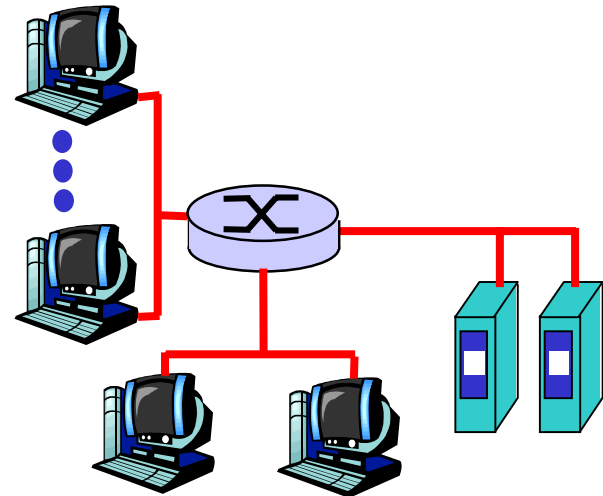


Accesso Residenziale: cable modems

- ❑ **HFC: hybrid fiber coax**
 - asimmetrico: fino a 10Mbps router-casa, 1 Mbps casa-router
- ❑ **rete** di cavi and fibre connettono abitazioni ai router di ISP
 - Accesso condiviso tra le abitazioni al router
 - problemi: congestione, dimensionamento
- ❑ diffusione: disponibile, in USA, dalle compagnie di TV via cavo

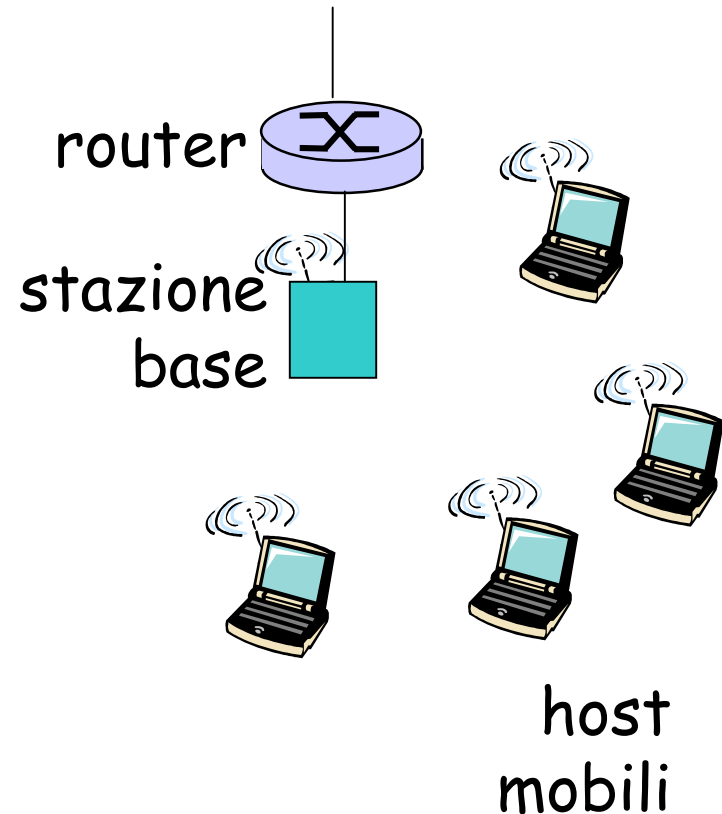
Accesso Istituzionale: local area networks

- ❑ **La local area network** (LAN) di aziende, università, connette host ad un edge router
- ❑ **Ethernet:**
 - Cavo condiviso o dedicato connette gli host ed il router
 - 10 Mbs, 100Mbps, Gigabit Ethernet
- ❑ **diffusione:** istituzioni, LAN casalinghe, attuale



Reti di accesso Wireless

- ❑ Una rete condivisa di accesso *wireless* connette host a router
- ❑ **wireless LAN:**
 - Spettro radio sostituisce il cavo
 - e.g., Lucent Wavelan 11 Mbps
- ❑ **Accessi wireless in area geografica**
 - CDPD: accesso wireless al router di un ISP attraverso la rete cellulare



Mezzi trasmissivi

- ❑ **Canali fisici:** bit di dati trasmessi si propagano lungo il canale
- ❑ **Mezzi guidati:**
 - segnali si propagano in mezzi solidi: rame, fibra
- ❑ **Mezzi non guidati:**
 - Segnali si propagano liberamente, e.g., radio

Twisted Pair (TP)

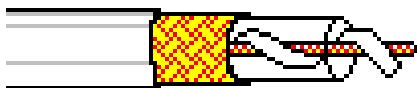
- ❑ Due cavi di rame isolati ed intrecciati
 - Categoria 3: doppino telefonico, 10 Mbps Ethernet
 - Categoria 5 TP: 100Mbps Ethernet



Mezzi trasmissivi: cavi coassiali, fibra

Cavo coassiale:

- ❑ Conduttore rame (portante segnale)
- ❑ Strato di plastica isola il conduttore da uno schermo di metallo intrecciato (per bloccare interferenze esterne)
- ❑ bi-direzionale
- ❑ Uso tipico per 10Mbps Ethernet



Cavo in fibra ottica:

- ❑ Fibra di vetro che trasporta impulsi ottici
- ❑ Operazioni ad alta velocità:
 - 100Mbps Ethernet
 - Alta velocità di trasmissione punto-punto (e.g., 5 Gps)
- ❑ Bassa probabilità di errore



Mezzi trasmissivi: radio

- ❑ Segnale trasportato nello spettro elettromagnetico
- ❑ Nessun cavo fisico
- ❑ bi-direzionale
- ❑ Effetti dell'ambiente sulla propagazione:
 - riflessione
 - ostruzione (oggetti ostacolo)
 - interferenza

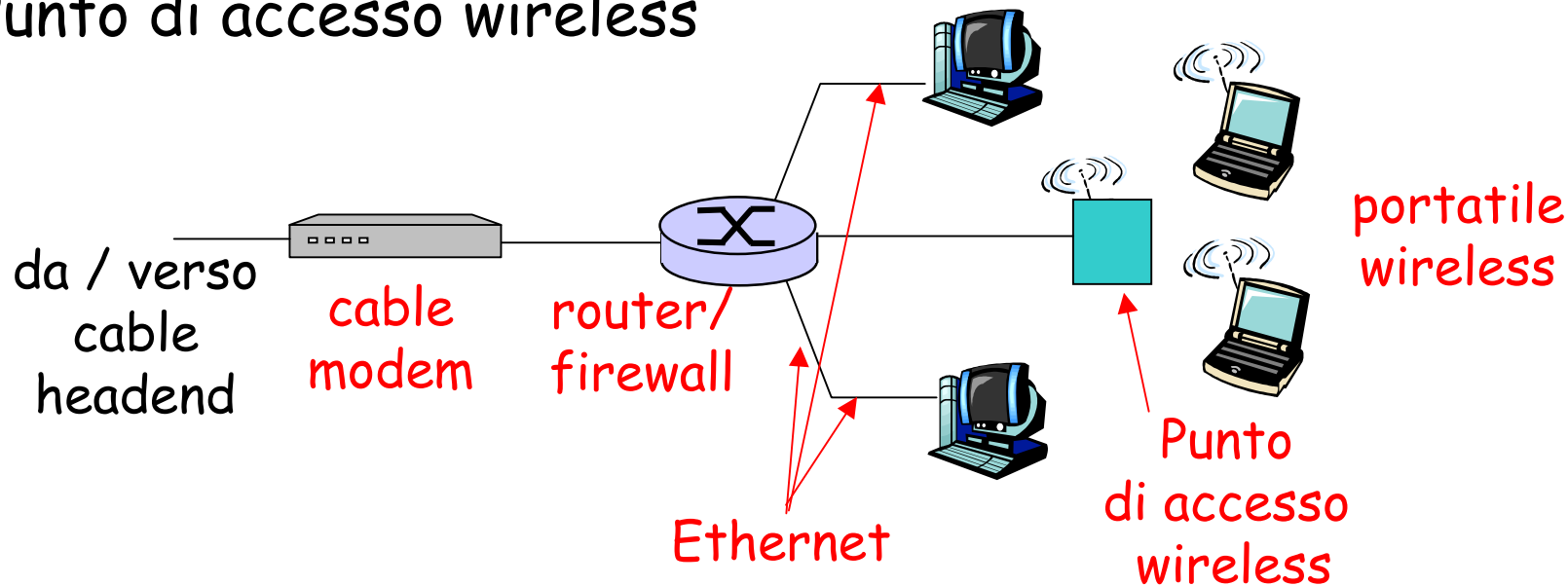
Tipi di canali radio:

- ❑ **microonde**
 - e.g. fino a 45 Mbps
- ❑ **LAN** (e.g., WaveLAN)
 - 2Mbps, 11Mbps
- ❑ **Area geografica** (e.g., cellulare)
 - e.g. CDPD, 10 Kbps
- ❑ **satellite**
 - fino a 50Mbps

Reti residenziali: il futuro?

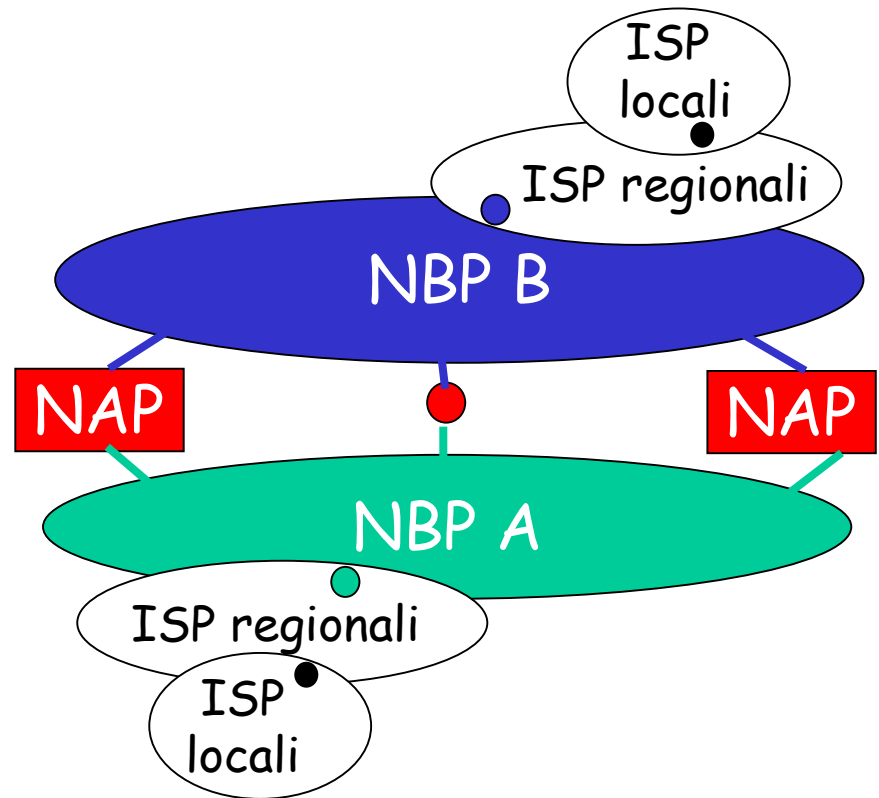
Componenti tipiche:

- ❑ Modem ADSL o per cavo
- ❑ router/firewall
- ❑ Ethernet
- ❑ Punto di accesso wireless



Struttura di Internet: rete di reti

- a grandi linee gerarchica
- **national/international backbone providers (NBP)**
 - e.g. BBN/GTE, Sprint, AT&T, IBM, UUNet
 - si inter-connettono direttamente, o tramite Network Access Point (NAP)
- **ISP regionali**
 - connettono ai NBP
- **ISP locali**, privati, istituzioni
 - connettono agli ISP regionali



Livelli di un protocollo

Le reti di telecomunicazione sono complesse!

- molte componenti:
 - host
 - router
 - canali di comunicazione (diversi mezzi trasmissivi)
 - applicazioni
 - protocolli
 - hardware, software

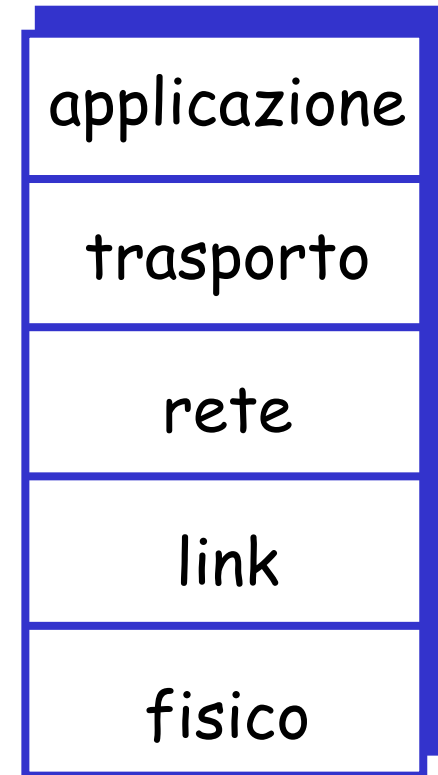
Domanda:

Si può organizzare la struttura di una rete?

O, almeno, la discussione?

Livelli di protocolli Internet

- ❑ **applicazione:** supporto per applicazioni di rete
 - ftp, smtp, http
- ❑ **trasporto:** trasferimento dati da host a host
 - tcp, udp
- ❑ **rete:** instradamento di pacchetti in cui sono suddivisi i dati da mittente a destinazione
 - ip, protocolli di routing
- ❑ **link:** trasferimento dati tra due elementi vicini (connessi) delle rete
 - ppp, ethernet
- ❑ **fisico:** trasferimento di bit lungo i canali di comunicazione fisici



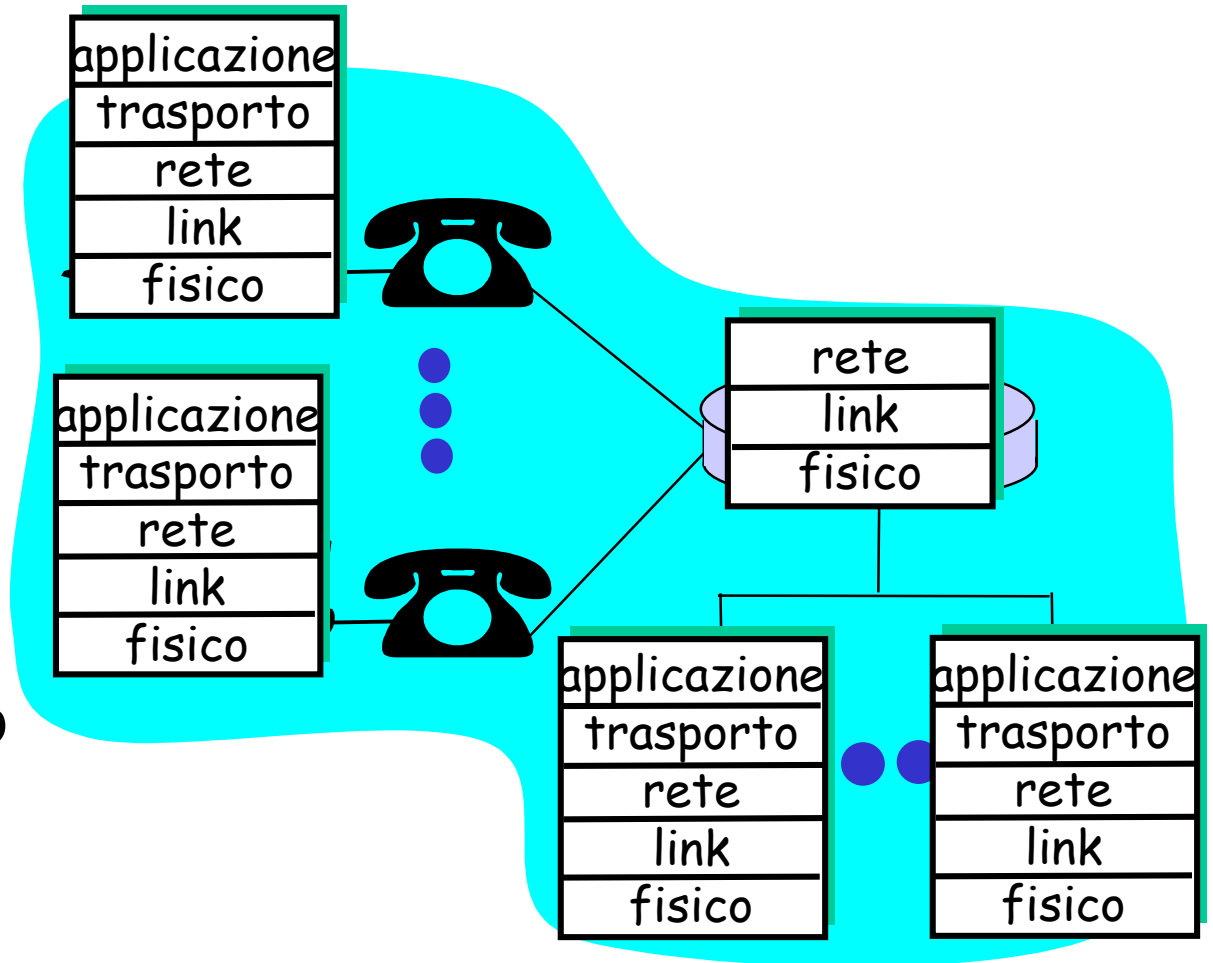
Funzionalità dei livelli

- Ogni livello può eseguire uno, o più, dei seguenti tipi di compito
 - controllo degli errori
 - controllo di flusso
 - segmentazione e ricostruzione
 - multiplexing/demultiplexing
 - "setup" di connessione

Struttura a livelli: comunicazione logica

Ogni livello:

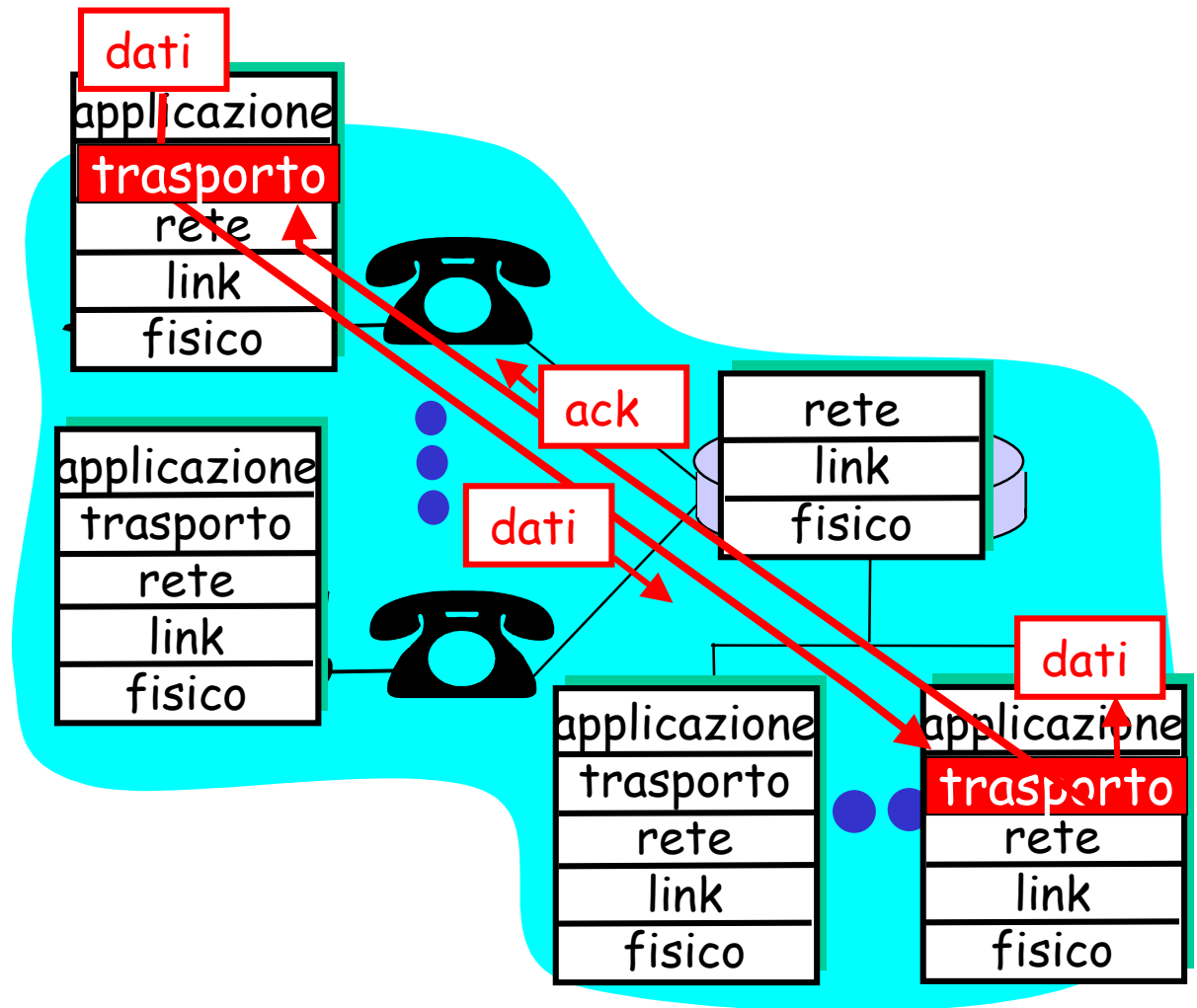
- distribuito
- "entità" implementano funzioni di un livello ad ogni nodo
- entità eseguono azioni, scambiano messaggi con i propri "pari"



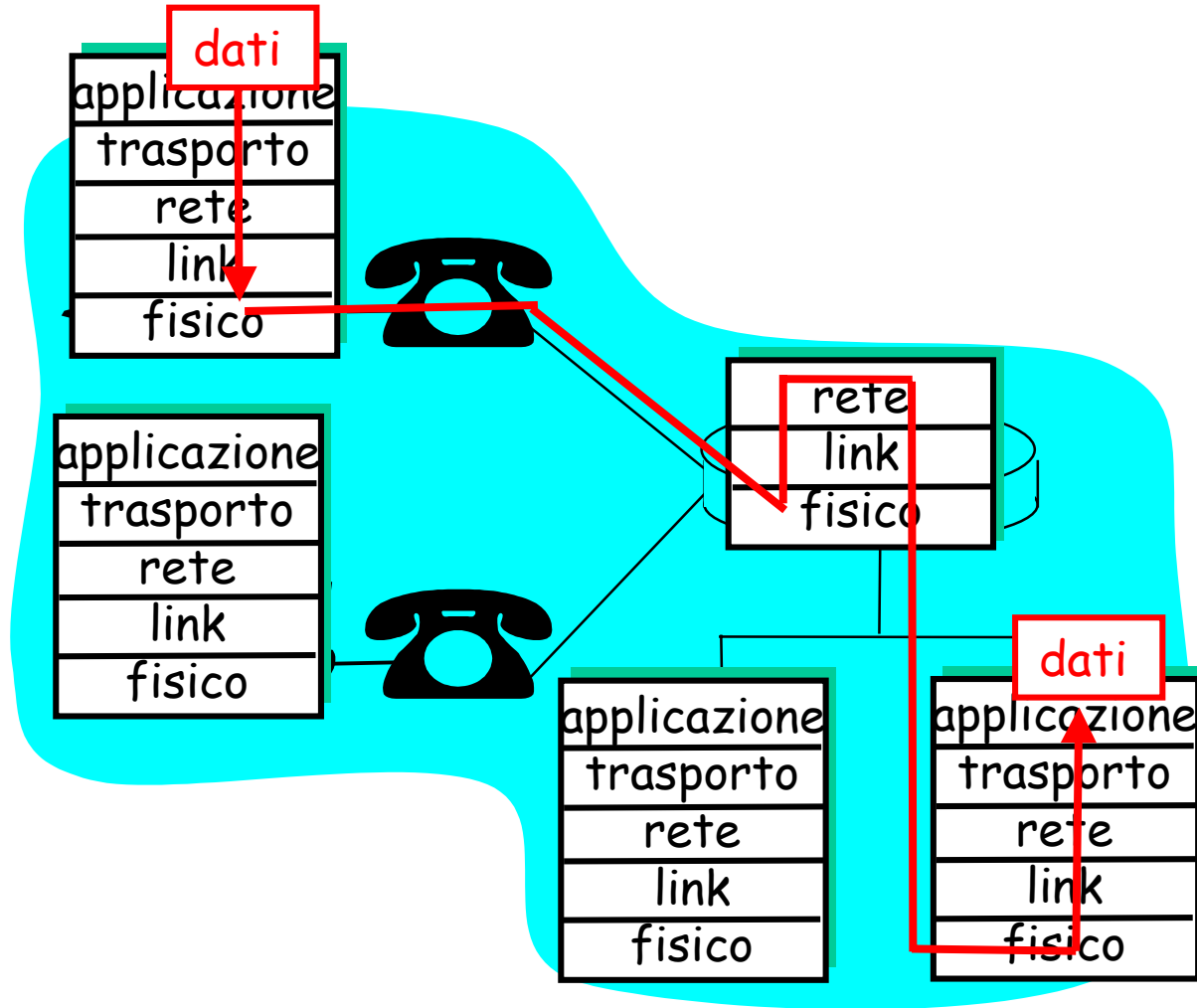
Struttura a livelli: comunicazione logica

E.g.: trasporto

- ❑ Riceve dati dall'applicazione
- ❑ Aggiunge indirizzamento, informazioni sul controllo di affidabilità per formare un segmento
- ❑ spedisce il segmento al proprio "pari"
- ❑ Attende che il "pari" confermi la ricezione (con un "ack")



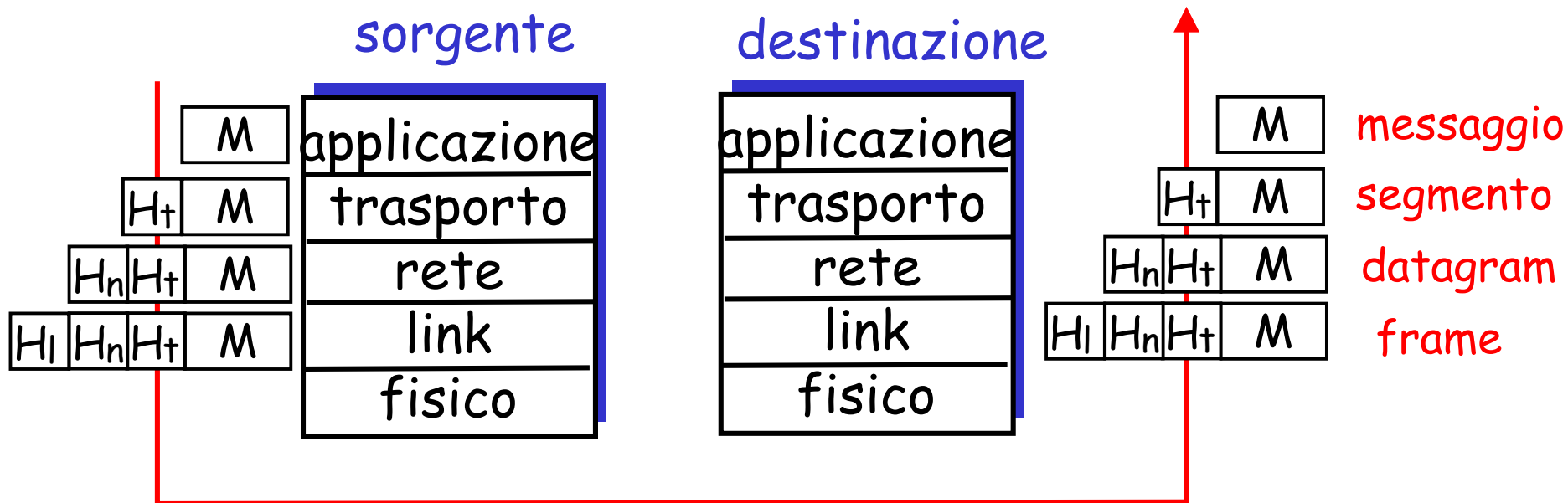
Struttura a livelli: comunicazione fisica



Protocolli a livelli e dati

Ogni livello prende dati dal livello superiore

- ❑ Aggiunge informazioni in un header (intestazione) per creare una nuova unità di dati
- ❑ Passa la nuova unità di dati al livello sottostante



Storia di Internet

1961-1972: primi principi della commutazione di pacchetti

- 1961: Kleinrock - teoria delle code dimostra l'efficacia della commutazione di pacchetto
- 1964: Baran - commutazione di pacchetto nelle reti militari
- 1967: ARPAnet concepita dal Advanced Research Projects Agency
- 1969: primo nodo ARPAnet operativo
- 1972:
 - Dimostrazione pubblica di ARPAnet
 - NCP (Network Control Protocol) primo protocollo host-host
 - Primo programma di e-mail
 - ARPAnet ha 15 nodi

Storia di Internet

1972-1980: Internetworking, nuove reti e reti proprietarie

- ❑ 1970: rete satellitare ALOHAnet nelle Hawaii
- ❑ 1973: Metcalfe nella sua tesi di PhD propone Ethernet
- ❑ 1974: architettura proposta da Cerf e Kahn per l'interconnessione di reti
- ❑ fine anni 70: architetture proprietarie: DECnet, SNA, XNA
- ❑ fine anni 70: commutazione di pacchetti di lunghezza fissa (precursore di ATM)
- ❑ 1979: ARPAnet ha 200 nodi

Principi di internetworking di Cerf e Kahn:

- minimalismo, autonomia - nessuna modifica interna richiesta per interconnettere reti
- modello di servizio best effort (al meglio)
- router che non mantengono lo stato delle connessioni
- controllo decentralizzato

Definiscono l'architettura dell'attuale Internet

Storia di Internet

1980-1990: nuovi protocolli, proliferazione di reti

- ❑ 1983: diffusione di TCP/IP
- ❑ 1982: definizione del protocollo smtp per l'e-mail
- ❑ 1983: definizione del DNS per la traduzione da nome ad indirizzo IP
- ❑ 1985: definizione del protocollo ftp
- ❑ 1988: controllo di congestione in TCP
- ❑ Nuove reti nazionali: Csnet, BITnet, NSFnet, Minitel
- ❑ 100,000 host connessi alla confederazione di reti

Storia di Internet

Anni 90: commercializzazione, il WWW

- ❑ Primi anni 90: disarmo di ARPAnet
- ❑ 1991: l'NSF allenta le restrizioni sull'uso commerciale di NSFnet (in disarmo nel 1995)
- ❑ Primi anni 90: WWW
 - ipertesti [Bush 1945, Nelson 1960's]
 - HTML, http: Berners-Lee
 - 1994: Mosaic, più tardi Netscape
 - fine anni 90: commercializzazione del WWW

Fine anni 90:

- ❑ 50 milioni di computers on Internet (stime)
- ❑ più di 100 milioni di utenti (stime)
- ❑ canali di comunicazione dei backbone operanti a 1 Gbps

Introduzione: sommario

Molto materiale

- ❑ panoramica su Internet
- ❑ cos'è un protocollo?
- ❑ edge network, core network, rete d'accesso
- ❑ strutturazione a livelli e modelli di servizio
- ❑ backbone, NAP, ISP
- ❑ storia

Parte 2: livello Applicazione (http)

Obiettivi:

- ❑ Aspetti concettuali ed implementativi di protocolli di applicazioni di rete
 - Paradigma client-server
 - Modello dei servizi
- ❑ Imparare concetti sui protocolli esaminando un protocollo a livello applicazione molto diffuso

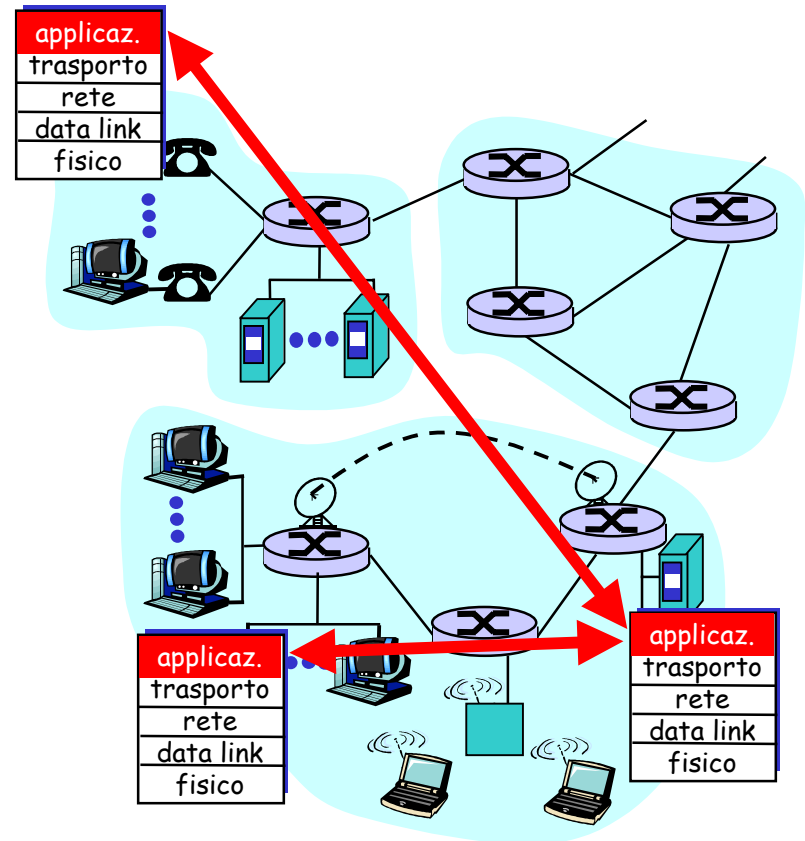
Applicazioni e protocolli a livello applicazione

Applicazione: processi distribuiti comunicanti

- vengono eseguiti sugli host di rete come processi utente
- scambio di messaggi per implementare l'applicazione
- e.g., email, ftp, Web

Protocolli a livello Applicazione

- una parte di un'applicazione
- definiscono i messaggi scambiati dall'applicazione e le azioni intraprese
- Usano i servizi di comunicazione forniti da protocolli a livello sottostante (TCP, UDP)



Applicazioni di rete: terminologia

Processo: programma in esecuzione in un host.

- processi in esecuzione su host diversi (distanti) comunicano con un **protocollo a livello applicazione**

- **user agent:** processo software, che si interfaccia con l'utente "verso l'alto" e con la rete "verso il basso".
 - implementa il protocollo a livello applicazione
 - **Web:** browser
 - **E-mail:** mail reader
 - **streaming audio/video:** media player

Paradigma Client-server

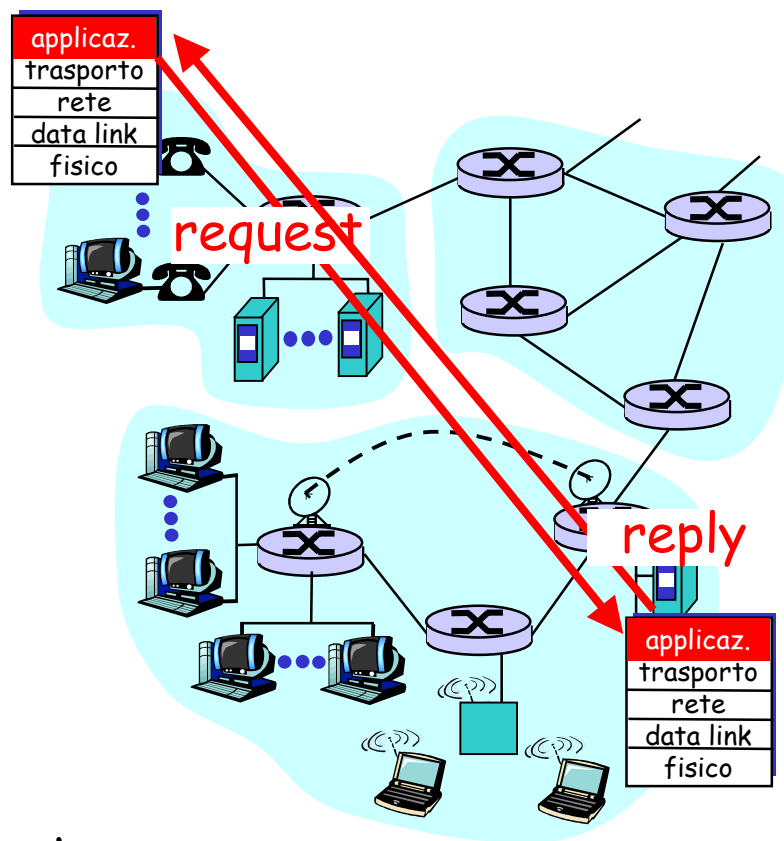
Una tipica applicazione di rete si compone di due parti: *client* e *server*

Client:

- ❑ Avvia il contatto con il server ("parla per primo")
- ❑ Solitamente, richiede un servizio al server
- ❑ **Web**: il client è implementato nel browser; **e-mail**: in mail reader

Server:

- ❑ Fornisce il servizio richiesto al client
- ❑ e.g., il Web server spedisce la pagina Web richiesta, il mail server recapita l'e-mail



Di quale servizio di trasporto necessita un'applicazione?

Perdita di dati

- ❑ Alcune applicazioni (e.g., audio) possono tollerare perdite
- ❑ Altre applicazioni (e.g., trasferimento file, telnet) richiedono un trasferimento dati affidabile al 100%

Time-sensitive

- ❑ Alcune applicazioni (e.g., telefonia su Internet, giochi interattivi) richiedono piccoli ritardi

Larghezza di banda

- ❑ Alcune applicazioni (e.g., multimediali) richiedono un ammontare minimo di larghezza di banda per essere "efficaci"
- ❑ Altre applicazioni ("applicazioni elastiche") fanno uso di qualunque larghezza di banda riescono ad ottenere

Requisiti del servizio di Trasporto di applicazioni comuni

<u>Applicazione</u>	<u>Perdite dati</u>	<u>Larghezza di banda</u>	<u>Time Sensitive</u>
trasferimento file	senza	elastica	no
e-mail	senza	elastica	no
documenti Web	tollerante	elastica	no
real-time audio/video	tollerante	audio: 5Kb-1Mb video: 10Kb-5Mb	si, alcuni 100 msec
stored audio/video	tollerante	come sopra	si, pochi secs
giochi interattivi	tollerante	alcuni Kbps	si, alcuni 100 msec
applicazioni finanziarie	senza	elastica	si e no

Servizi di trasporto in Internet

Servizio TCP:

- ❑ *connection-oriented*: fase iniziale di "setup" necessaria tra client e server
- ❑ *trasporto affidabile* tra processo mittente e destinatario
- ❑ *controllo di flusso*: il mittente non sovraccaricherà il ricevitore
- ❑ *controllo di congestione*: regolazione della velocità del mittente quando la rete è sovraccarica
- ❑ *non fornisce*: tempi, garanzie su larghezza di banda minima

Servizio UDP:

- ❑ trasferimento dati non affidabile tra processo mittente e processo destinatario
- ❑ non fornisce: setup della connessione, affidabilità, controllo di flusso, controllo di congestione, tempi o larghezza di banda garantiti

Domanda: perché mai esiste UDP?

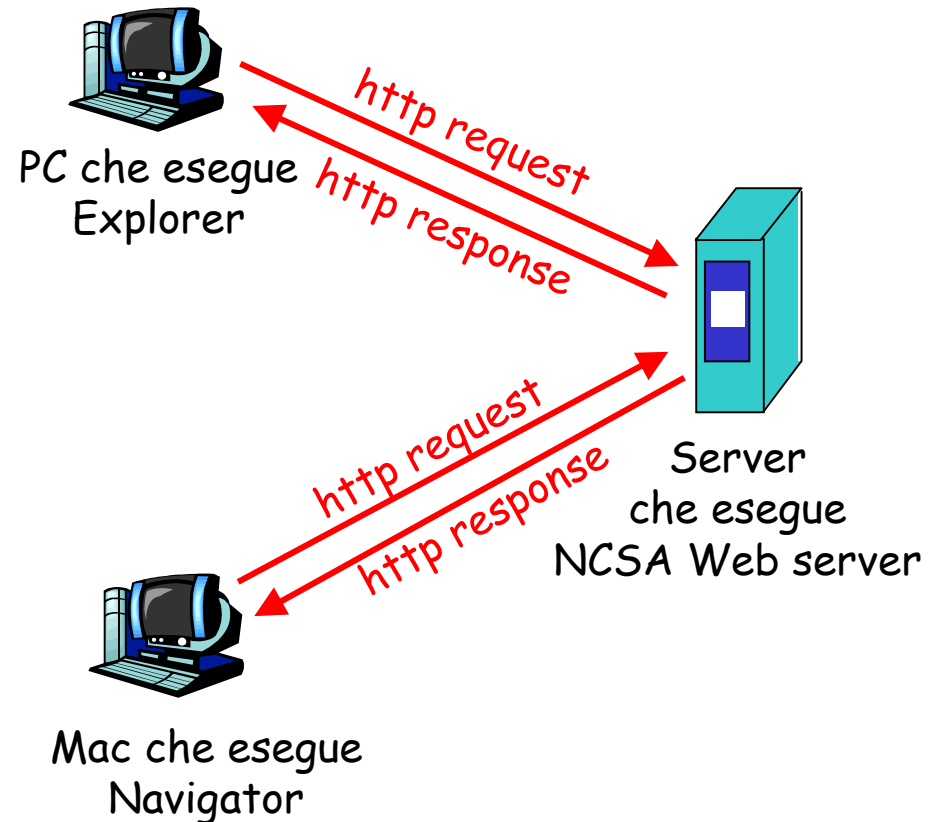
Applicazioni Internet: protocolli a livello applicazione e trasporto

Applicazione	Protocollo a livello applicazione	Protocollo a livello trasporto sottostante
e-mail	smtp [RFC 821]	TCP
accesso a terminale remoto	telnet [RFC 854]	TCP
Web	http [RFC 2068]	TCP
trasferimento file	ftp [RFC 959]	TCP
streaming multimedia	proprietario (e.g. RealNetworks)	TCP or UDP
file server remoto	NSF	TCP or UDP
telefonia su Internet	proprietario (e.g., Vocaltec)	solitamente UDP

Il Web: il protocollo http

http: hypertext transfer protocol

- ❑ Protocollo a livello applicazione per il Web
- ❑ Modello client/server
 - *client*: il browser che richiede, riceve e mostra oggetti Web
 - *server*: Web server che spedisce oggetti in risposta ad una richiesta
- ❑ http1.0: RFC 1945
- ❑ http1.1: RFC 2068-2616



Il WEB: terminologia

- ❑ **pagina WEB (documento):** collezione di oggetti
- ❑ **oggetto:** un file (HTML, JPEG, ...)
- ❑ **file HTML base:** con direttive e riferimenti ad altri oggetti
- ❑ **URL (Uniform Resource Locator):** meccanismo di identificazione risorse. Si compone del nome del host sul quale risiede l'oggetto e il path-name dell'oggetto

o www.di.unito.it/various/presentation_en.html

nome host

path-name

Il protocollo http

http: usa servizio TCP:

- ❑ il client avvia una connessione TCP (crea un socket) con il server, porta 80
- ❑ il server accetta la connessione TCP dal client
- ❑ vengono scambiati messaggi http (messaggi del protocollo di livello applicazione) tra il browser (client http) ed il Web server (server http)
- ❑ la connessione TCP viene chiusa

http è "stateless"

- ❑ il server non mantiene alcuna informazione sulle richieste passate dei client

nota

I protocolli che mantengono lo stato sono complessi!

- ❑ Tutta la storia passata della connessione (stato) deve essere mantenuta, memorizzata
- ❑ se server o client subiscono un crash, la loro conoscenza dello stato può essere inconsistente e deve essere ricostruita

http: esempio

Supponiamo l'utente digiti l'URL

www.someSchool.edu/someDepartment/home.index

(contiene testo e
10 riferimenti ad
immagini jpeg)

1a. il client http inizia una
connessione TCP al server http
(che è un processo)
all'indirizzo
www.someSchool.edu. La porta
80 è il default per i server
http.

1b. il server http sull' host
www.someSchool.edu in attesa
di connessioni TCP alla porta
80. "accetta" la connessione,
notificandola al client


2. il client http spedisce il
messaggio http *request
message* (contenente l'URL)
usando la connessione TCP

3. il server http riceve il messaggio
di richiesta, forma un messaggio
http *response message*
contenente l'oggetto richiesto
(someDepartment/home.index), e
spedisce un messaggio usando la
connessione TCP

tempo
↓

http: esempio (continuazione)

4. il server http chiude la connessione TCP.



5. il client http riceve il messaggio di risposta contenente il file html e lo mostra. Parsifica (analizza) il file html, trova i riferimenti a 10 oggetti jpeg.

6. passi 1-5 si ripetono per ognuno dei 10 oggetti jpeg.

tempo



Connessioni persistenti e non persistenti

Non persistenti

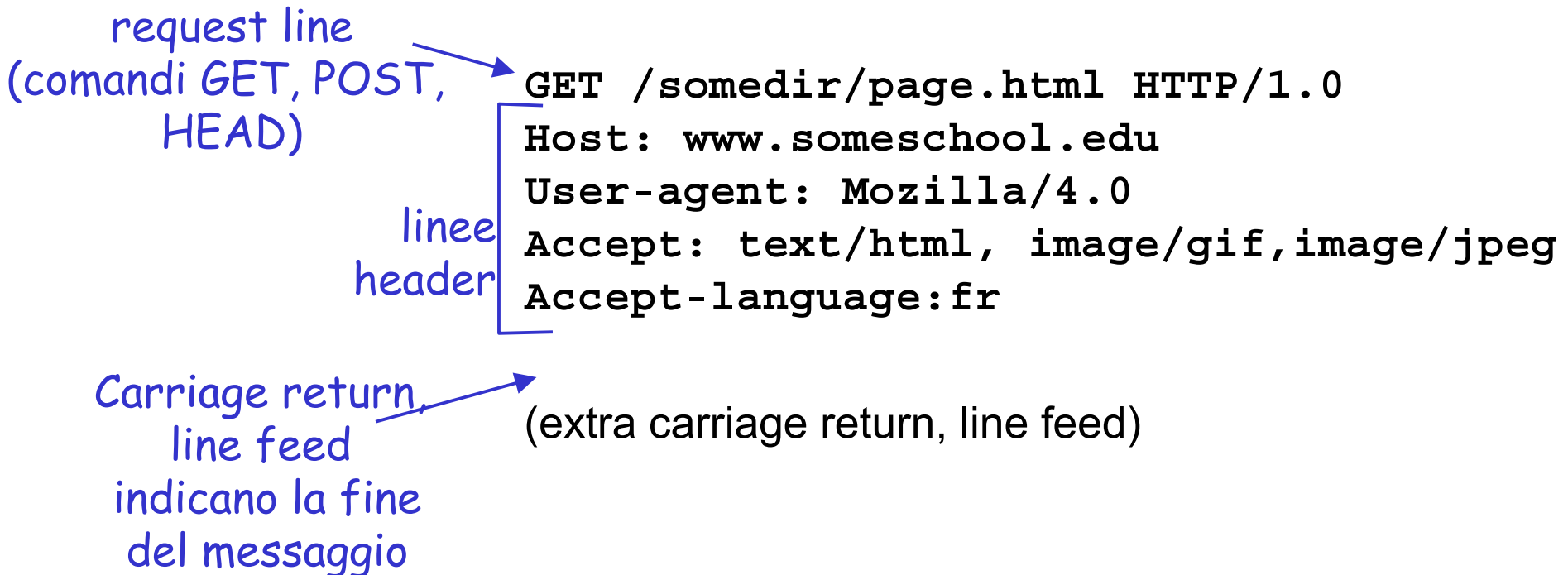
- ❑ http/1.0: il server parsifica le richieste, risponde, chiude la connessione TCP
- ❑ 2 RTT (round trip time) per ottenere l'oggetto
 - connessione TCP
 - richiesta/trasferimento oggetto
- ❑ Ogni trasferimento risente della bassa velocità iniziale di trasferimento di TCP
- ❑ molti browser aprono connessioni multiple in parallelo

Persistenti

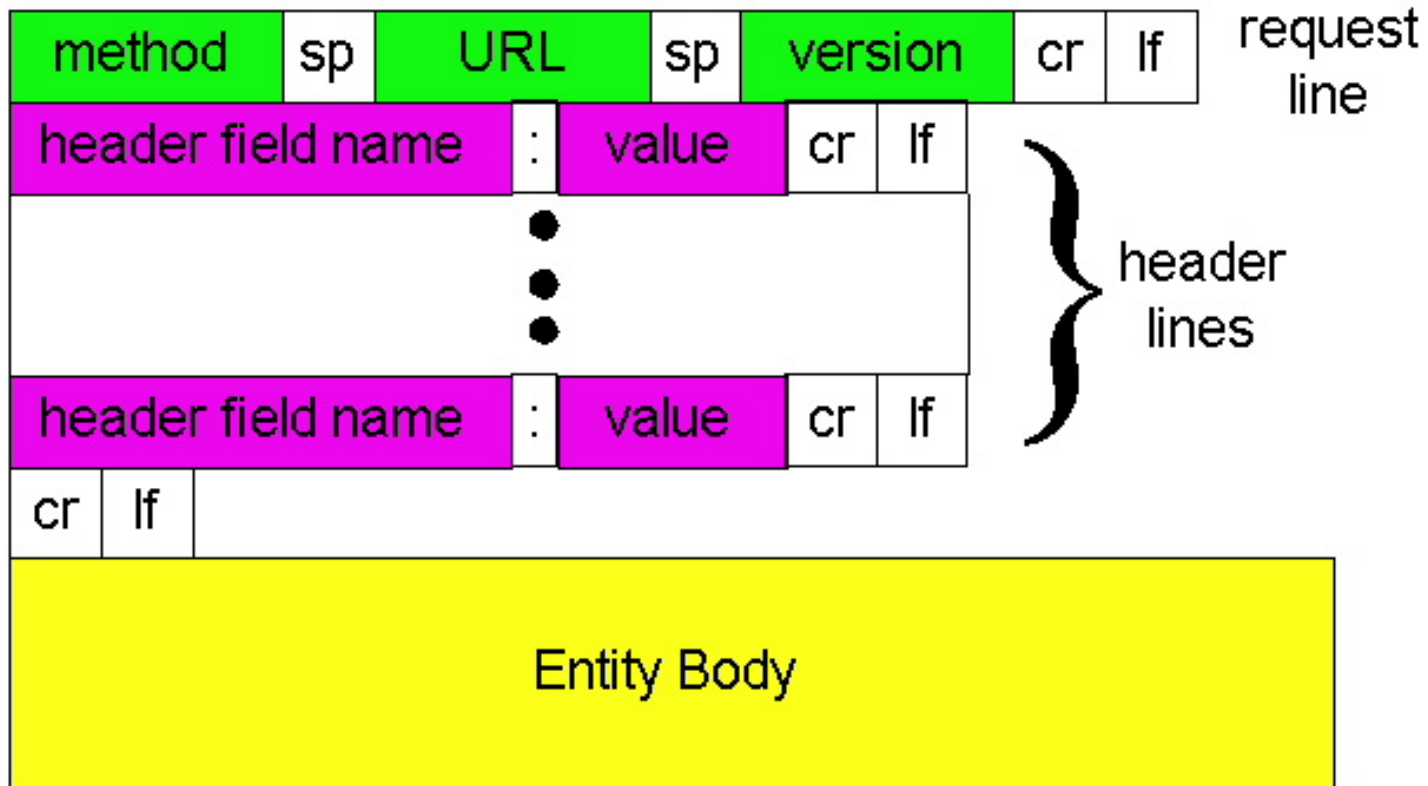
- ❑ default per http/1.1
- ❑ usando la stessa connessione TCP: il server, parsifica richieste, risponde, parsifica nuove richieste.....
- ❑ il client spedisce le richieste per tutti gli oggetti a cui si fa riferimento non appena riceve il file HTML base.
- ❑ Con o senza pipelining

Formato dei messaggi http: request

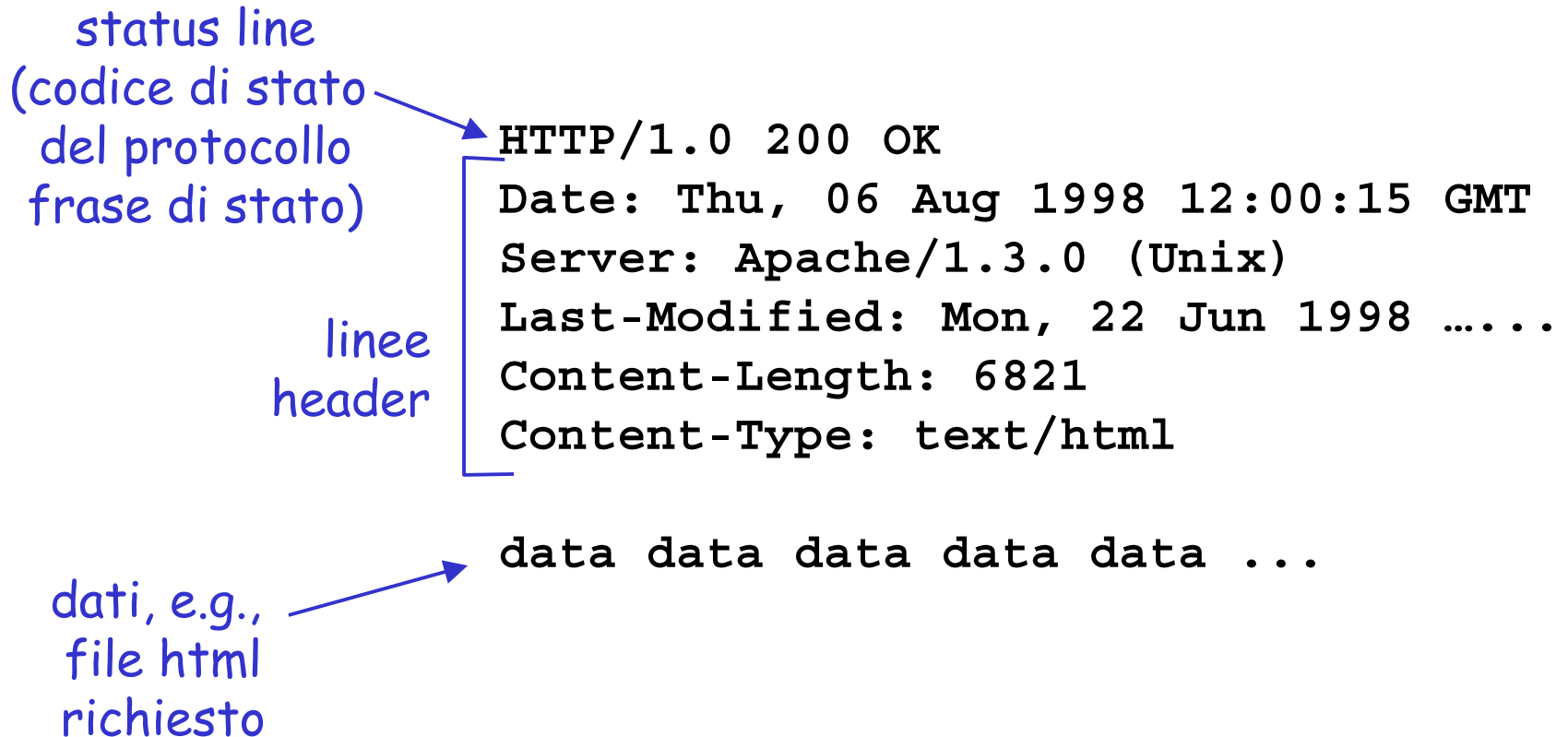
- ❑ due tipi di messaggi http: *request, response*
- ❑ **http request message:**
 - ASCII (formato human-readable)



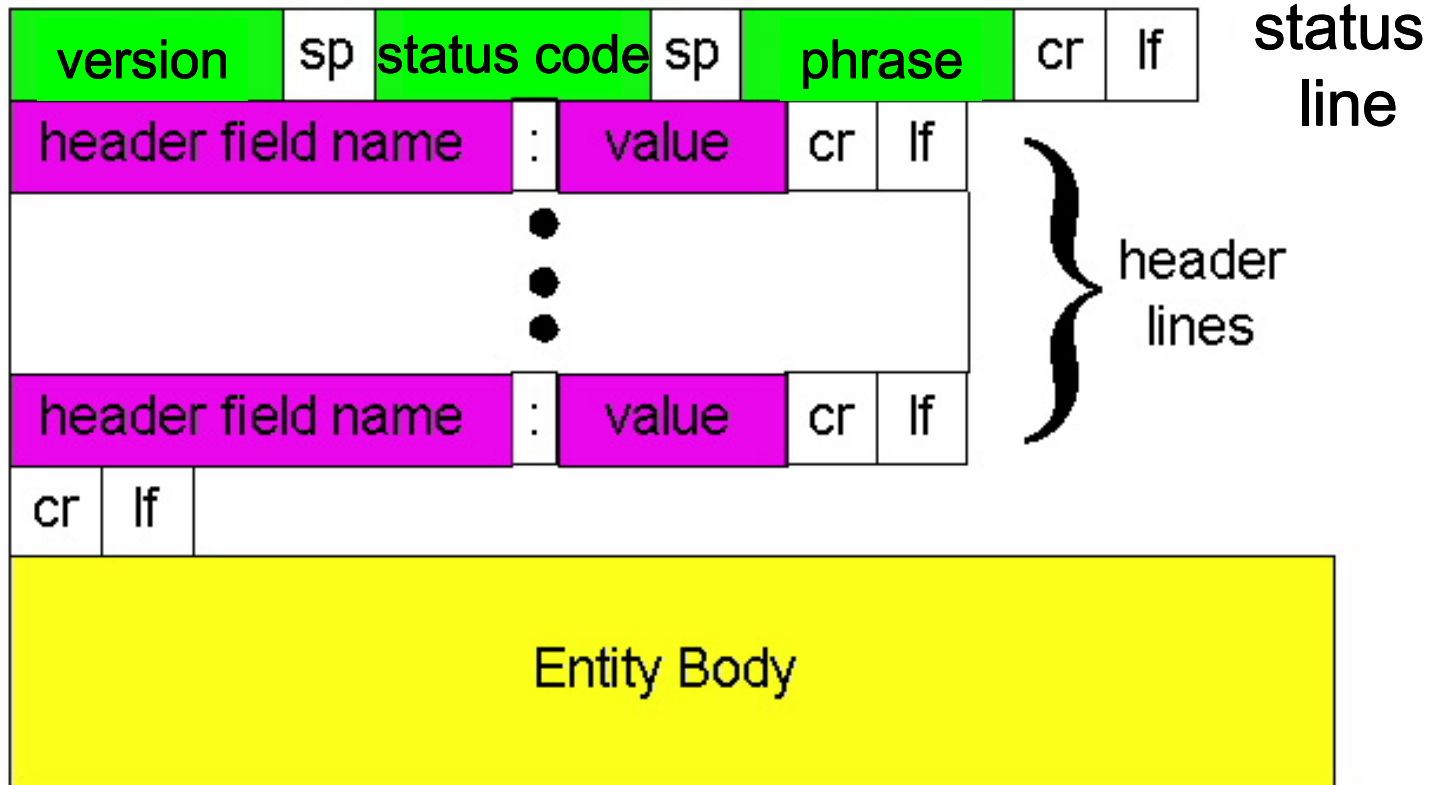
http request message: formato generale



Formato dei messaggi http: response



http response message: formato generale



Codici di stato per http response

Nella prima linea del response message server->client.

Alcuni codici d'esempio:

200 OK

- o richiesta con successo, l'oggetto richiesto segue in questo messaggio

301 Moved Permanently

- o L'oggetto richiesto è stato spostato, la nuova locazione è specificata dopo in questo messaggio (Location:)

400 Bad Request

- o request message non compreso dal server

404 Not Found

- o Documento richiesto non trovato su questo server

505 HTTP Version Not Supported

Provate http (lato client)

1. Collegatevi con telnet ad un Web server:

```
telnet www.eurecom.fr 80
```

Apri una connessione TCP sulla porta 80 (porta di default per un http server) su www.eurecom.fr. Qualunque cosa si digiti viene spedita sulla porta 80 a www.eurecom.fr

2. Digitate un http request GET:

```
GET /~ross/index.html HTTP/1.0
```

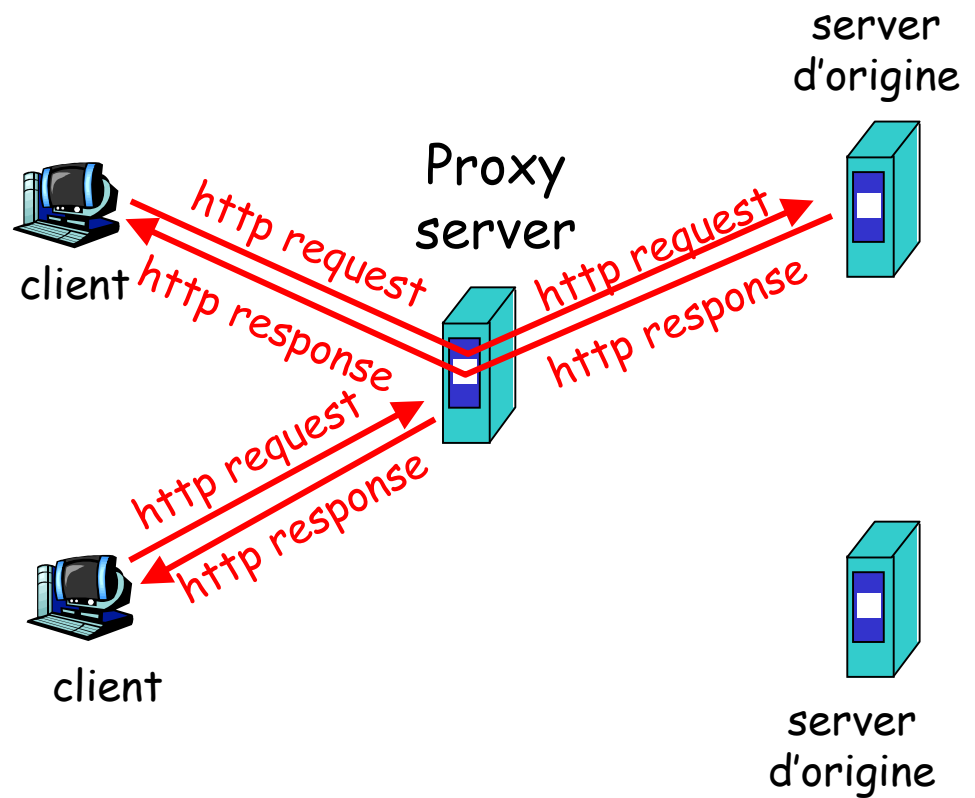
Digitando questo (digitate due volte carriage return), spedite un minimale (ma completo) GET request al server http

3. Guardate il response message spedito dal server http!

Web Cache (proxy server)

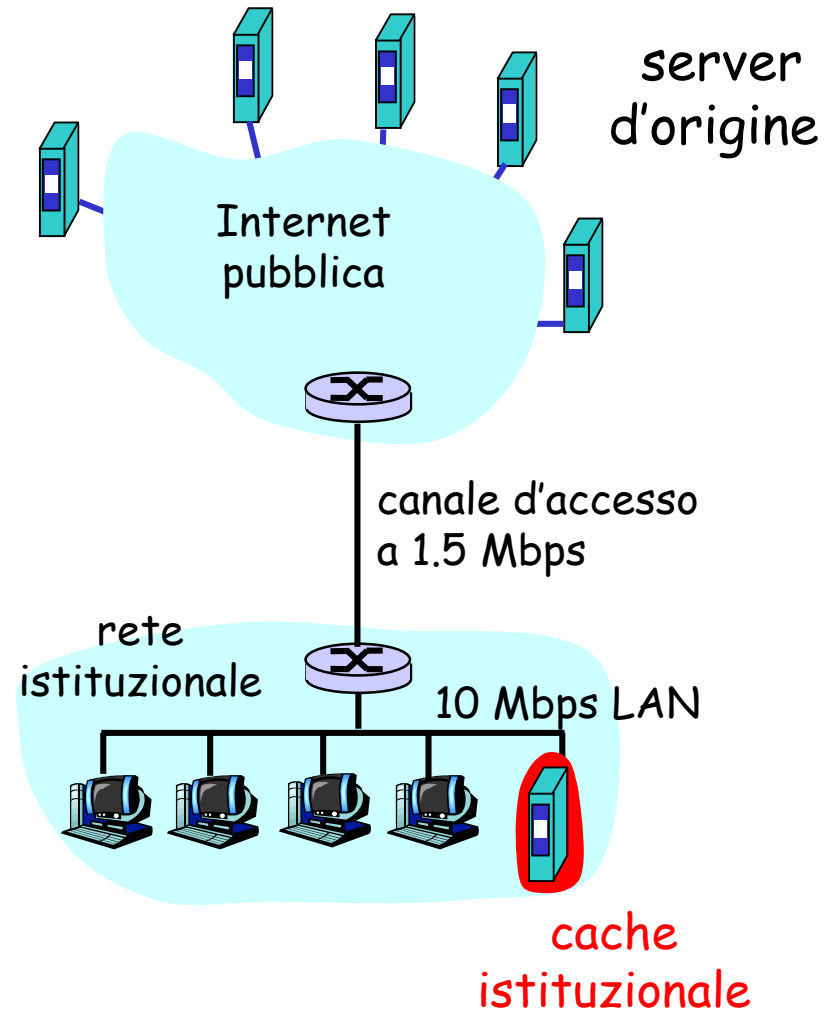
Obiettivo: soddisfare le richieste del client senza coinvolgere il server originale

- utente configura il browser: accesso Web via web cache
- Il client manda tutti gli http request al web cache
 - oggetto nel web cache: il web cache spedisce l'oggetto
 - altrimenti il web cache richiede l'oggetto dal server d'origine, quindi lo spedisce al client



Perché il Web Caching?

- Se:** il web cache è "vicino" al client (e.g., nella stessa rete)
- tempi di risposta inferiori: il web cache è "più vicino" al client
 - Diminuzione del traffico verso server distanti
 - i canali esterni alla rete del ISP locale/istituzionale sono spesso colli di bottiglia



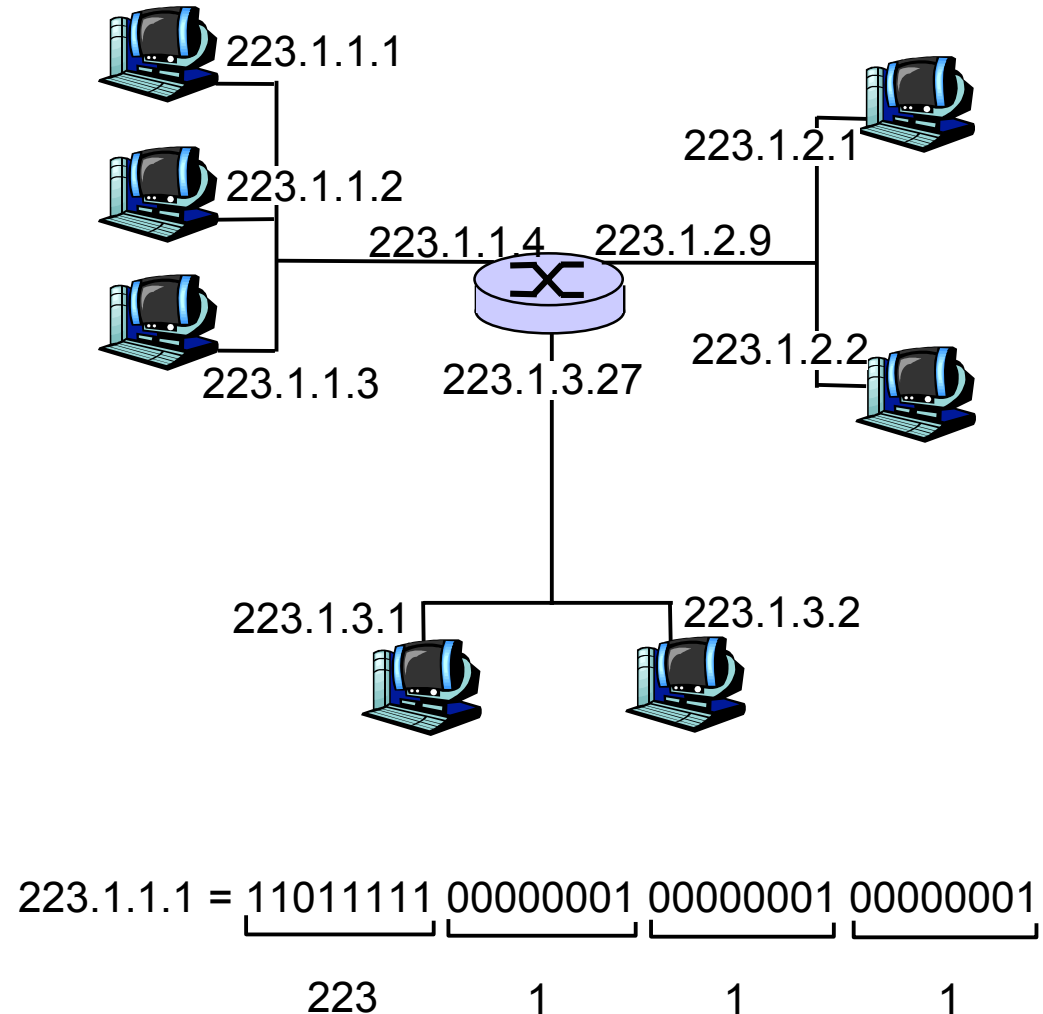
Protocolli a livello applicazione

domanda: come fa un processo ad identificare l'altro processo con il quale vuole comunicare?

- o "numero di porta" - permette all' host che riceve di determinare a quale dei processi che sta eseguendo (locali) debba essere recapitato il messaggio
- o **Indirizzo IP** dell' host sul quale è in esecuzione l'altro processo

Indirizzi IP: introduzione

- indirizzo IP:
identificatore di 32-bit
per l'interfaccia di rete di
host e router
- *interfaccia di rete*:
connessione tra host,
router ed il canale fisico
 - i router, tipicamente
hanno interfacce multiple
 - gli host possono avere
interfacce multiple
 - gli indirizzi IP sono
associati alle interfacce e
NON agli host o ai router



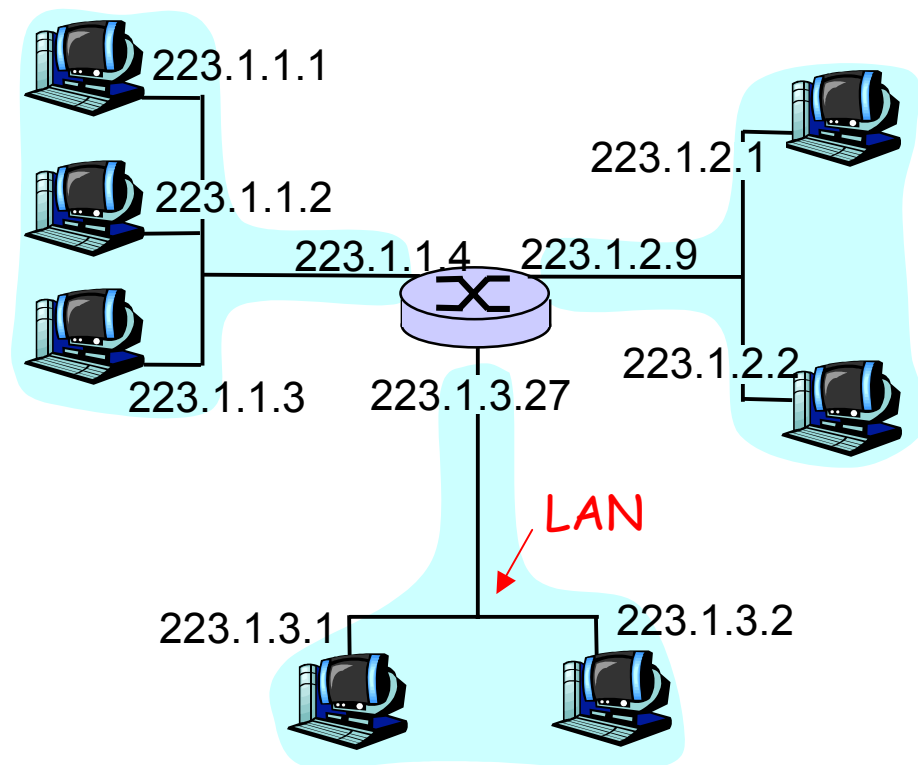
Indirizzi IP

□ indirizzo IP:

- parte rete (bit più a sinistra)
- parte host (bit più a destra)

□ *Cos'è una rete ?* (dal punto di vista dell'indirizzo IP)

- dispositivi d'interfaccia con la stessa parte rete dell'indirizzo IP
- possono fisicamente raggiungere l'un l'altra senza l'intervento di router



rete di 3 reti IP
(per gli indirizzi IP che iniziano con 223,
i primi 24 bit sono l'indirizzo della rete)

Indirizzi IP: come averne uno?

Host (porzione host dell'indirizzo):

- codificato dall'amministratore del sistema in un file
- **DHCP: Dynamic Host Configuration Protocol**: ottiene dinamicamente un indirizzo:
 - l'host manda un messaggio broadcast "DHCP discover"
 - il server DHCP risponde con un messaggio "DHCP offer"
 - l'host richiede un indirizzo IP con un "DHCP request"
 - il server DHCP spedisce un indirizzo IP con un "DHCP ack"

Indirizzi IP: come averne uno?

Rete (porzione rete dell'indirizzo):

- si ottiene una porzione allocata dello spazio di indirizzi del ISP:

blocco ISP	<u>11001000</u>	<u>00010111</u>	<u>00010000</u>	00000000	200.23.16.0/20
Organizzaz. 0	<u>11001000</u>	<u>00010111</u>	<u>00010000</u>	00000000	200.23.16.0/23
Organizzaz. 1	<u>11001000</u>	<u>00010111</u>	<u>00010010</u>	00000000	200.23.18.0/23
Organizzaz. 2	<u>11001000</u>	<u>00010111</u>	<u>00010100</u>	00000000	200.23.20.0/23
...
Organizzaz. 7	<u>11001000</u>	<u>00010111</u>	<u>00011110</u>	00000000	200.23.30.0/23

Indirizzi IP: l'ultimo pezzo...

Domanda: Come ottiene un ISP un blocco d'indirizzi?

Risposta: **ICANN:** Internet Corporation for Assigned Names and Numbers

- o alloca gli indirizzi
- o gestisce il DNS
- o assegna i nomi dei domini, risolve le dispute

DNS: Domain Name System

Persone: molti identificativi:

- # CF, nome, # passaporto

Host e router in Internet:

- indirizzo IP (32 bit) - usato per indirizzare datagrams
- "nome", e.g., pianeta.di.unito.it - usato dagli esseri umani

Domanda: corrispondenza tra indirizzo IP e nome ?

Domain Name System:

- *database distribuito* implementato con una gerarchia di *name server*
- *protocollo di livello applicazione* host, router, e name servers comunicano per *risolvere* nomi (traduzione indirizzo/nome)
 - nota: funzione chiave in Internet, implementata come protocollo a livello applicazione
 - complessità nella edge network

Servizi offerti dal DNS

- ❑ Traduzione di indirizzi mnemonico -> IP
- ❑ Host aliasing
 - indirizzi mnemonici "complicati" possono avere alias più semplici
- ❑ Mail server aliasing
- ❑ Distribuzione di carico
 - web server replicati su host con stesso indirizzo mnemonico (e.g., cnn.com) ma diversi indirizzi IP
 - rotazione degli indirizzi nelle risposte (l'ordine determina chi contattare per primo da parte del client HTTP)

DNS name servers

- nessun server ha tutte le corrispondenze nome-indirizzo IP

name server locali:

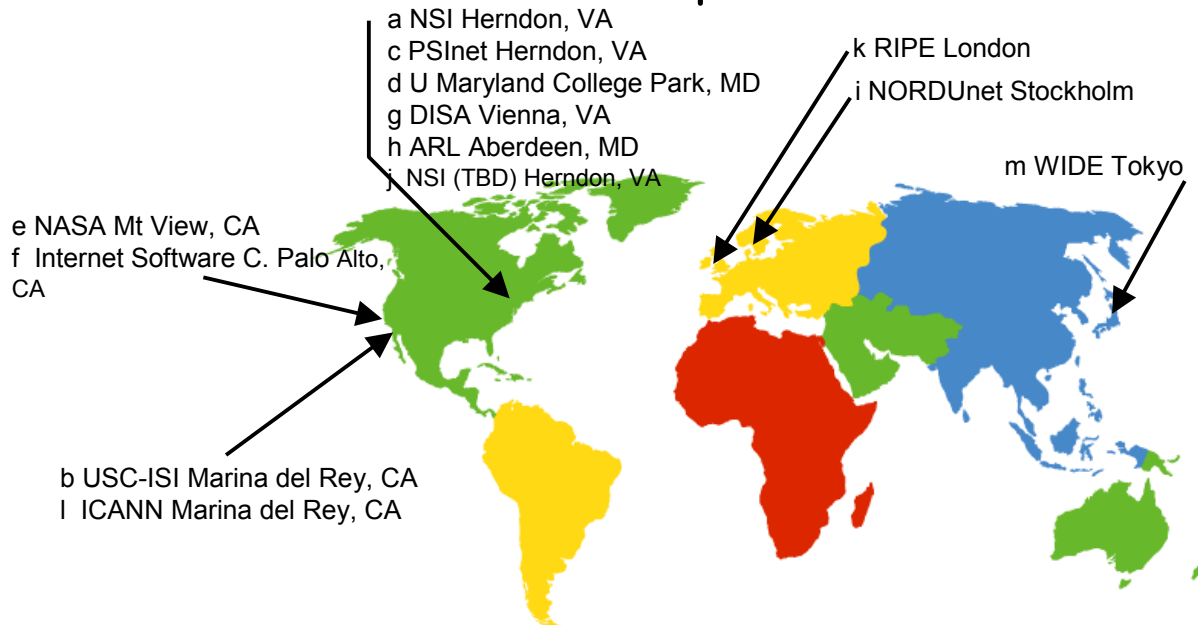
- ogni ISP o azienda ha dei *name server locali (default)*
- la query DNS di un host è innanzitutto diretta al name server locale

name server autoritativo:

- per un host: memorizza il nome e l'indirizzo IP di quell' host
- può eseguire la traduzione nome/indirizzo per il nome di quell' host

DNS: Root name servers

- ❑ contattati dai name server locali che non sanno risolvere un nome
- ❑ root name server:
 - contatta il name server autoritativo se la corrispondenza per il nome non è conosciuta
 - ottiene la corrispondenza
 - restituisce la corrispondenza al name server locale



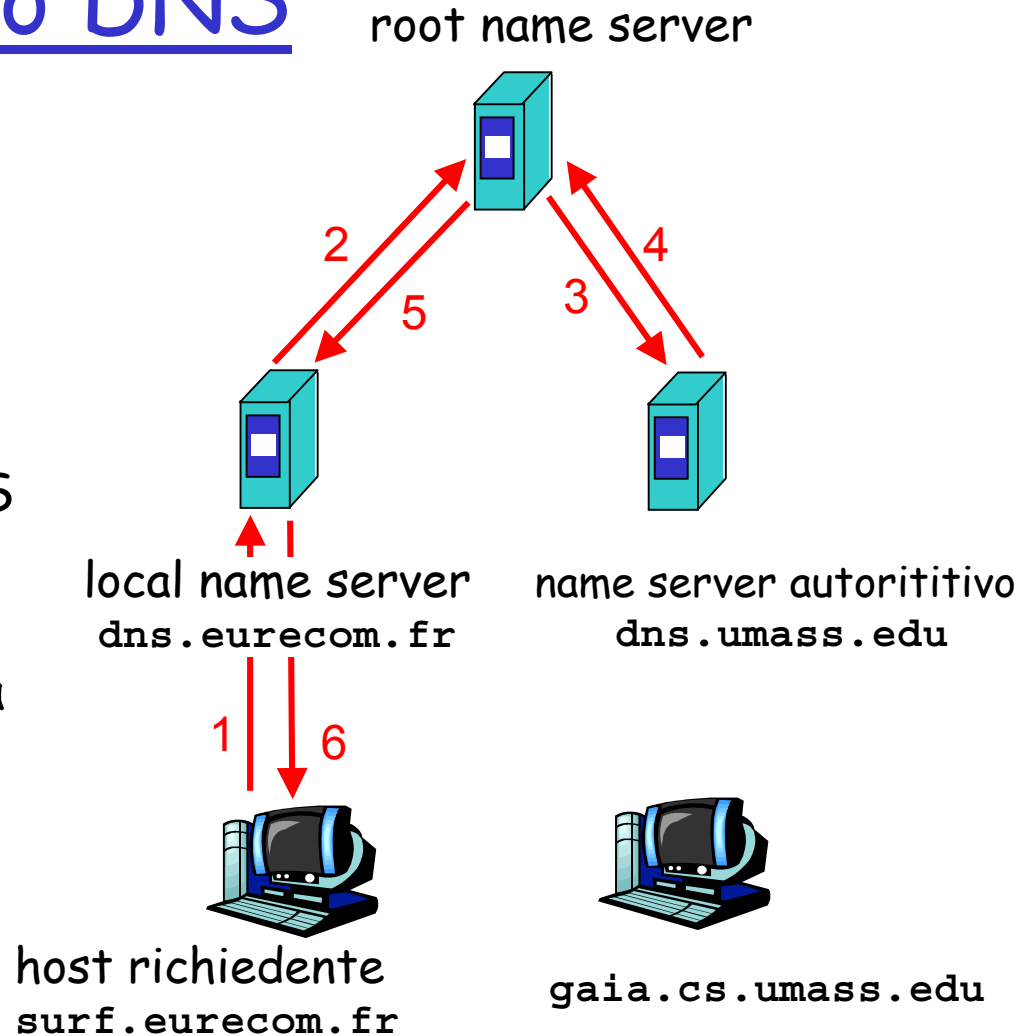
13 root name
server nel mondo

Semplice esempio DNS

l' host

`surf.eurecom.fr`
vuole l'indirizzo IP di
`gaia.cs.umass.edu`

1. contatta il suo server DNS locale,
`dns.eurecom.fr`
2. `dns.eurecom.fr` contatta il root name server, se necessario
3. il root name server contatta il name server autoritativo, `dns.umass.edu`, se necessario



Esempio DNS

Root name server:

- può non conoscere il name server autoritativo
- può conoscere un *name server intermedio*: chi contattare per trovare il name server autoritativo

