

Internet

La rete delle reti: collega fra loro reti locali, metropolitane, geografiche e singoli computer di tutto il mondo



Internet

- 1990: 3000 reti e 200.000 calcolatori (detti host)
- 1992: viene collegato il milionesimo host
- Agli esordi il numero di host cresce in modo esponenziale mentre in questi anni si osserva un rallentamento, con un incremento annuo del 6% (sondaggio Nielsen)
- Maggio 2002: hanno accesso ad Internet 457 milioni di persone, di cui 174 milioni negli Stati Uniti (sondaggio Nielsen)

Cosa vuol dire essere in Internet?

- Una macchina è in Internet se utilizza la famiglia di protocolli **TCP/IP** (ufficiale dal 1 gennaio 1983), ha un suo indirizzo IP, ed ha la capacità di spedire pacchetti IP a tutte le altre macchine su Internet

- E' possibile essere in Internet anche in modo temporaneo chiamando un fornitore di servizi Internet (per esempio mediante un modem)

Cosa vuol dire protocollo?

- Abbiamo già visto che la comunicazione tra due calcolatori in una rete è governata da un insieme di regole che prendono il nome di **protocollo** e che forniscono funzionalità per
 - indirizzamento (addressing)
 - instradamento (routing)
 - gestione di eventuali errori di trasmissione (error detection, error recovery, sequence control)
 - gestione della velocità di comunicazione (flow control)

Comunicazione multilivello

- Per ogni coppia di livelli adiacenti esiste una **interfaccia**
- Le convenzioni usate nella conversazione sono il protocollo
 - si tratta di un accordo tra i partecipanti su come deve avvenire la comunicazione
- Al di sotto del livello più basso c'è il mezzo fisico che serve per il trasferimento dei dati

Comunicazione multilivello: i livelli di Internet



Comunicazione multilivello: i livelli di Internet



Il calcolatore A produce un messaggio che deve essere inviato al calcolatore B.

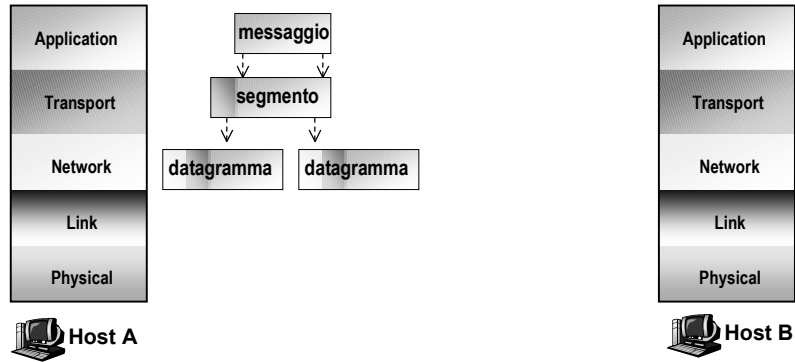
Il messaggio viene prodotto da un programma applicativo nel livello più alto della gerarchia (Application).

Comunicazione multilivello: i livelli di Internet



Il livello Application passa il messaggio al livello Transport dove viene inserita un'intestazione (**header**) che permette di identificarlo.

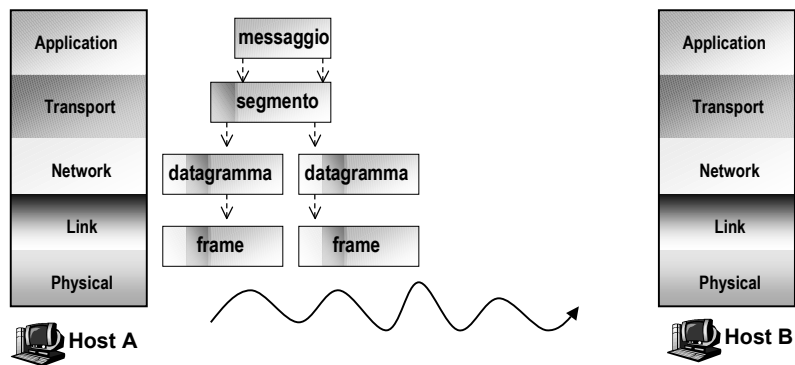
Comunicazione multilivello: i livelli di Internet



Dal livello Transport il segmento viene passato al livello Network dove viene suddiviso in parti più piccole (**pacchetti** o **datagram**).

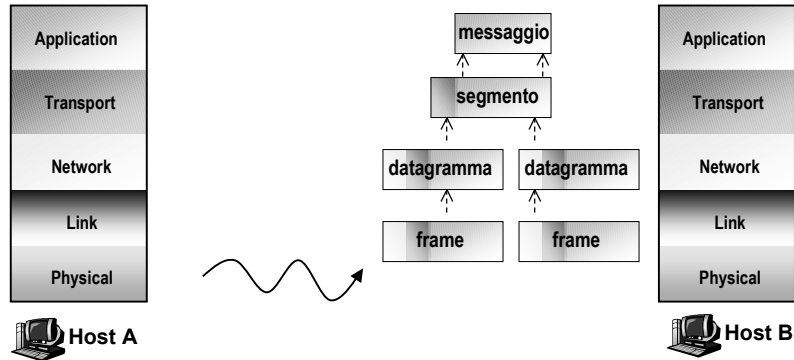
Ad ogni pacchetto viene aggiunto un **header**.

Comunicazione multilivello: i livelli di Internet



Finalmente si raggiungono i livelli più bassi: il livello Data link (**frame**) e il livello Physical dove avviene il trasferimento fisico dei dati verso il nodo destinatario B.

Comunicazione multilivello: i livelli di Internet



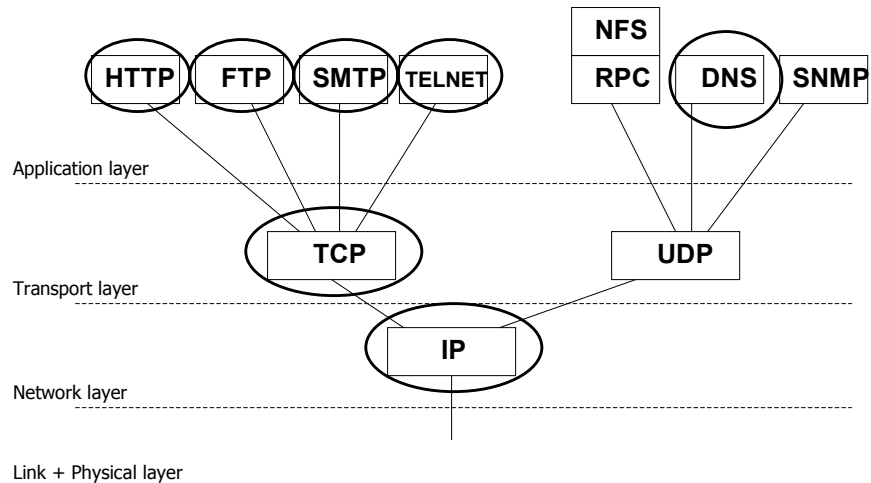
Sul nodo B i pacchetti arrivano al livello più basso e risalgono tutti i livelli, con le intestazioni che vengono eliminate di volta in volta
 Il messaggio originale viene così ricostruito e ricevuto dal destinatario.

La famiglia di protocolli TCP/IP

- Come abbiamo visto si tratta di protocolli organizzati in **livelli concettuali**, e ad ogni livello corrispondono determinate funzioni

Application	Servizi per l'utilizzo della rete (trasferimento file, email, login remoto, WWW)
Transport	Comunicazione end-to-end
Network	Indirizzamento Routing tra reti
Link + Physical	

La famiglia di protocolli TCP/IP

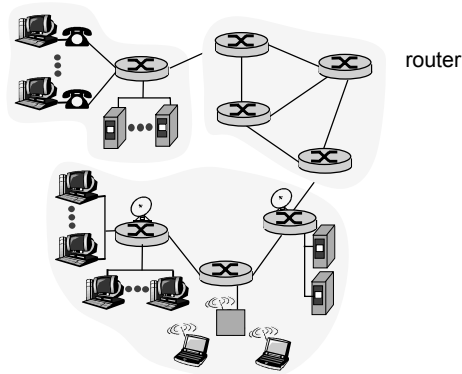


Network layer: IP

- Internet può essere vista come una collezione di sottoreti diverse (**eterogenee**) connesse tra loro (**internetworking**)
- La “colla” che tiene insieme le varie sottoreti è l'**Internet Protocol (IP)**
- Permette di trasportare i dati dalla sorgente alla destinazione, sfruttando la presenza di reti intermedie lungo il percorso

Network layer: IP

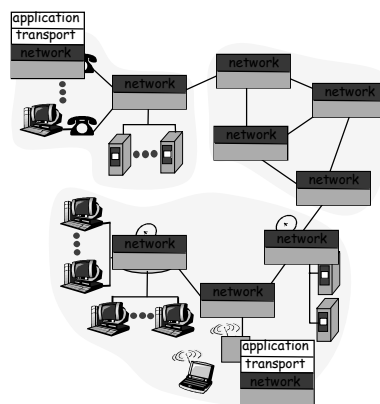
- Una **entità** di livello Network è presente su tutti i dispositivi collegati ad Internet



NB. Una entità è una parte del sistema operativo che implementa le funzionalità previste da un certo livello

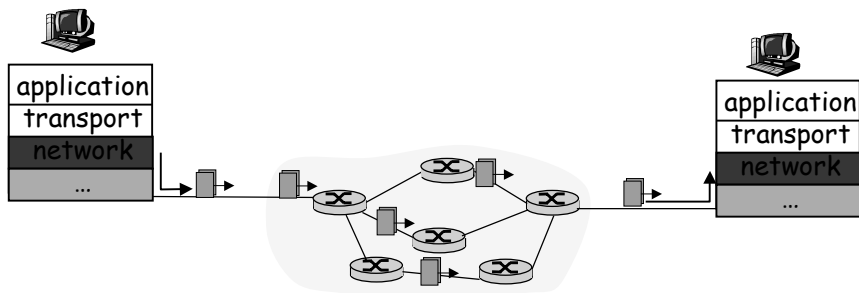
Network layer: IP

- Una entità di livello Network è presente su tutti i dispositivi collegati in Internet



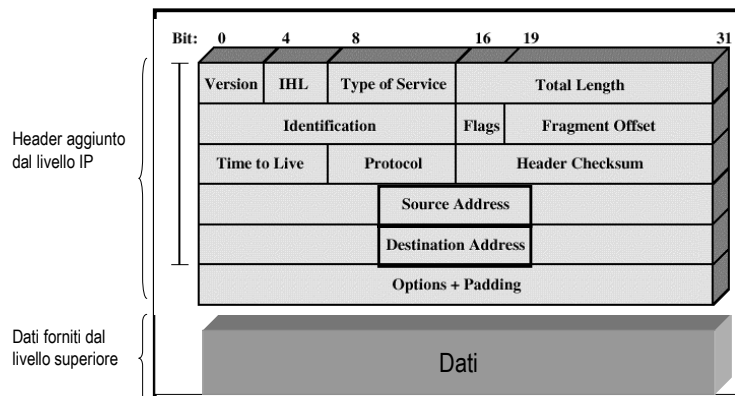
Network layer: IP

- Trasmissione di tipo **packet switching**
I dati, suddivisi in pacchetti, possono seguire percorsi diversi
- I **router** non mantengono informazioni sullo stato delle comunicazioni tra il mittente e il destinatario

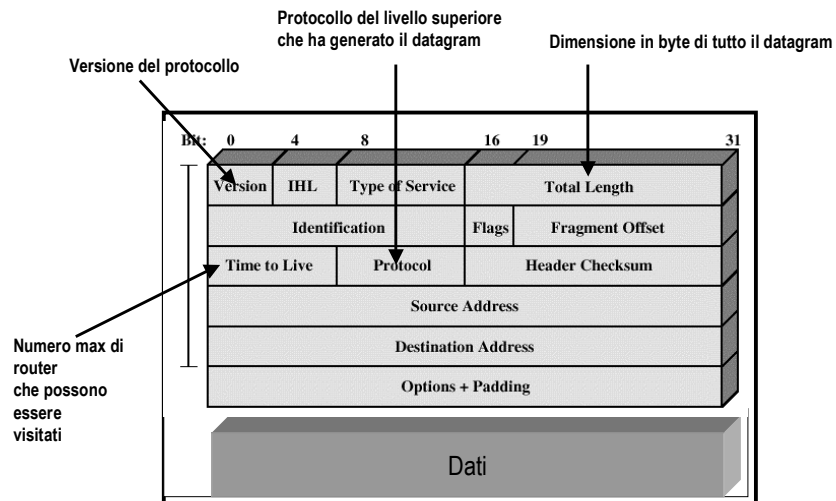


IP: datagram

- Specifica il formato dei pacchetti (**datagram**) spediti sulla rete Internet (lunghezza max 64 KB)
- L'header IP è lungo 20 byte



IP: datagram



IP: datagram

- Il servizio fornito da IP è **connectionless**
 - ogni datagram è gestito indipendentemente da tutti gli altri e IP non si preoccupa di verificare la corretta ricezione dei datagram
- Il servizio è **unreliable**
 - i datagram possono arrivare fuori sequenza oppure possono essere persi

IP: indirizzi

- Ogni calcolatore collegato ad Internet possiede un indirizzo univoco detto indirizzo IP (**32 bit**)
 - **NetId**: identifica la **rete** cui il calcolatore è fisicamente collegato
 - **HostId**: identifica il **calcolatore** all'interno della rete

IP: indirizzi

- I 32 bit di un indirizzo IP sono suddivisi in 4 campi da 8 bit ciascuno

10000000000010100000001000011110

- Di solito si usa una rappresentazione formata da quattro numeri decimali separati da un punto

128.10.2.30

NB: l'indirizzo 127.0.0.1 indica il localhost

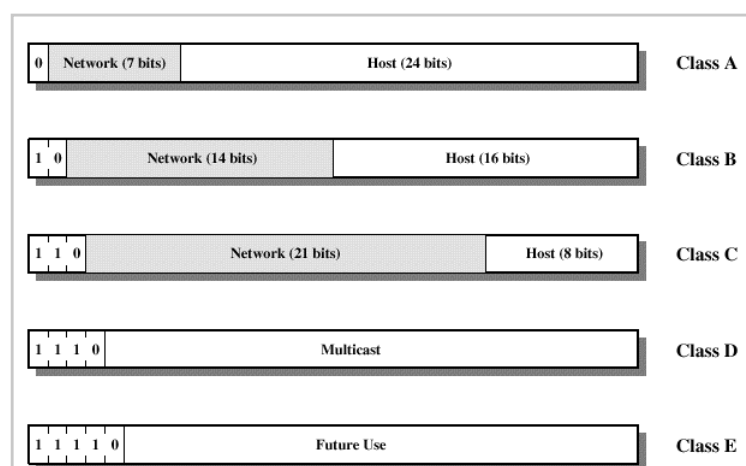
IP: indirizzi

- Gli indirizzi IP devono essere **univoci**
 - per questo motivo è stata istituita una organizzazione, **Internet Assigned Number Authority**, preposta ad assegnare gli indirizzi IP garantendone l'univocità

- Quando vi collegate ad Internet da casa è il provider che vi assegna un indirizzo IP scegliendolo tra quelli che ha acquistato

- In SW1 ogni PC, oltre al nome logico, ha un indirizzo IP

IP: classi di reti



IP: classi di reti

- Numero di reti e di host per ciascuna classe

Classe	Reti	Host
A	$2^7=128$	$2^{24}=16.777.216$
B	$2^{14}=16.384$	$2^{16}=65.536$
C	$2^{21}=2.097.152$	$2^8=256$

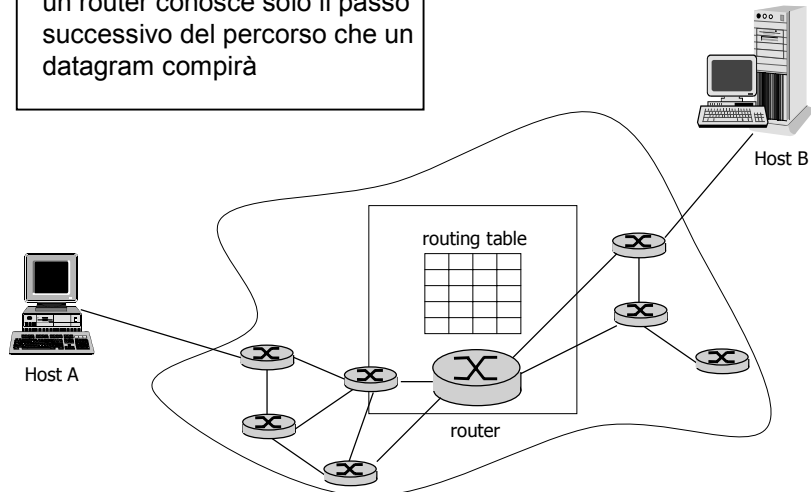
Gli indirizzi a 32 bit si stanno "saturando". Una nuova versione di IP usa indirizzi a 128 bit.

IP: routing

- IP fornisce anche l'instradamento (**routing**) dei pacchetti tra mittente e destinatario
- Nei router viene mantenuta una tabella di routing che viene usata per trovare il prossimo router o host

IP: routing

Routing di tipo next-hop:
un router conosce solo il passo
successivo del percorso che un
datagram compirà



IP: routing

Protocollo di routing

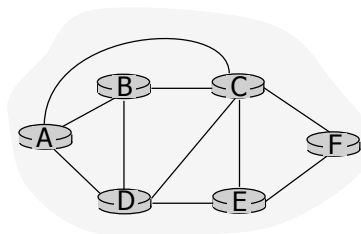
Scopo: determinare un "buon"
percorso (sequenza di router)
nella rete tra sorgente e
destinazione

Percorso "buono"

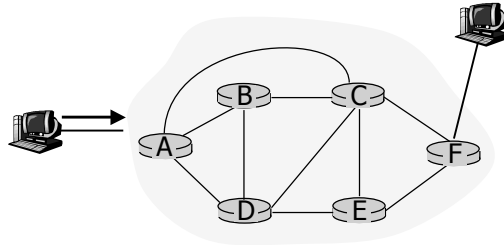
in genere significa "più corto"
(con meno router da attraversare)

La "topologia" può cambiare

qualche router o link si può guastare

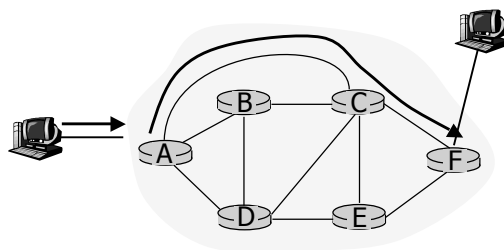


IP: routing



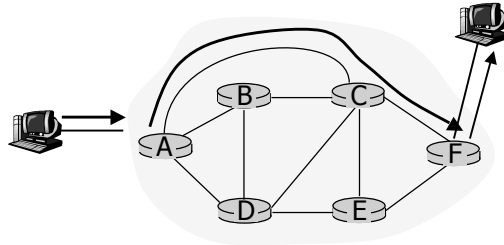
il calcolatore sorgente invia il datagram ad un router collegato alla sua stessa rete ...

IP: routing



... che provvederà ad inviarlo ad un altro router, e così via fino a quando il datagram non giungerà ad un router appartenente alla stessa rete del destinatario ...

IP: routing

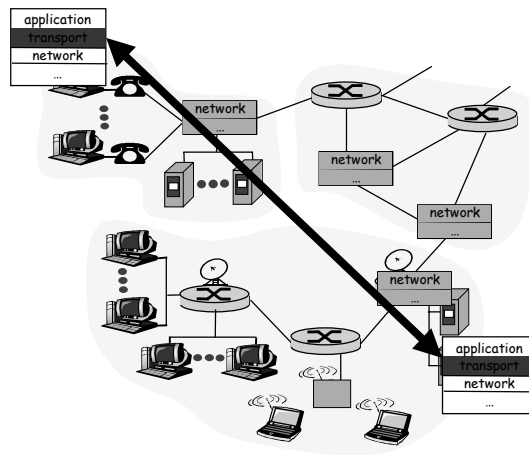


... ed il router finale invierà il datagram al destinatario mediante recapito diretto

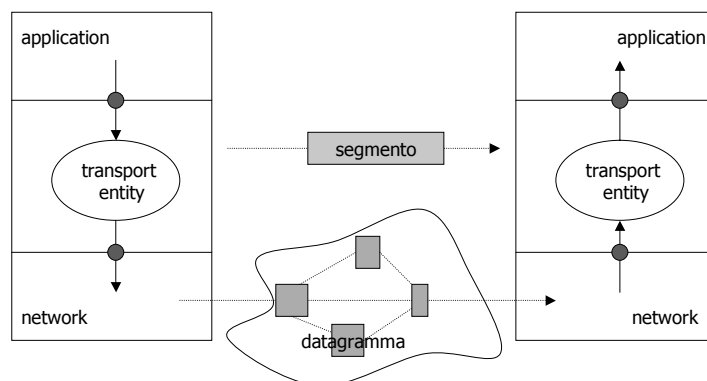
Transport layer

- Il livello Transport è il cuore di tutta la gerarchia di protocolli
- Il suo compito è quello di fornire un trasporto **affidabile** dall'host di origine a quello di destinazione, indipendentemente dalla rete utilizzata
- In Internet il protocollo di questo livello è chiamato **Transmission Control Protocol (TCP)**

Transport layer



Transport layer



I servizi transport sono basati sui servizi network

Transport layer

- Può offrire servizi **affidabili** orientati alla connessione (**TCP**) oppure servizi datagram (**UDP**)
- Si deve specificare la **destinazione finale**, cioè si deve decidere come è fatto **l'indirizzo del livello di trasporto**

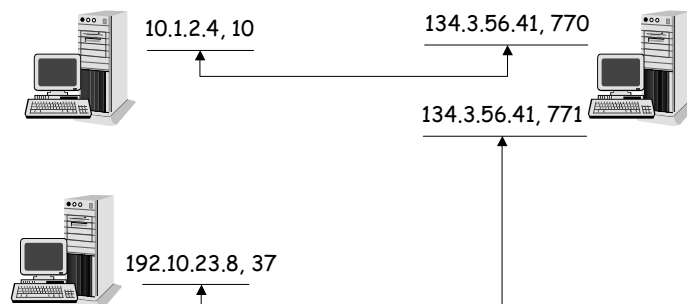
TSAP = <NSAP address, informazione supplementare>

TSAP: Transport Service Access Point

NSAP: Network Service Access Point

TSAP per TCP

<indirizzo IP mittente, **porta mittente**>



TCP: porte

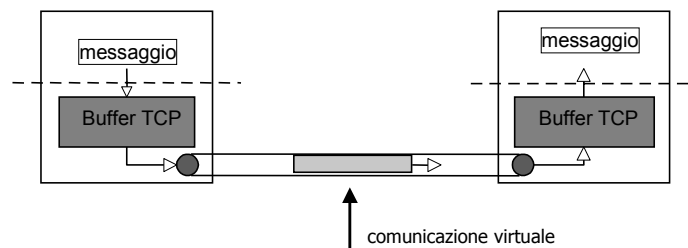
- I servizi più comuni usano sempre le stesse porte
 - **21** ftp
 - **22** ssh
 - **23** telnet
 - **25** mail
 - **80** http

- Sono dette porte **well-known**
- Ricordate il proxy del laboratorio SW1?

Su Linux guardate il file /etc/services

TCP: dettagli nei prossimi corsi ...

- Un messaggio spedito da un'applicazione viene frammentato in **segmenti** che sono vengono spediti consecutivamente



- Anche a livello TCP ci sono delle intestazioni (**header**) da associare ad ogni segmento
- La **connessione è affidabile**

Application layer

- Si colloca al di sopra del livello Transport ed è il livello nel quale viene svolto il “lavoro utile” per l’utente

- In questo livello si trovano diversi protocolli, alcuni relativi alle applicazioni che usiamo abitualmente in Internet
 - SMTP (Simple Mail Transfer Protocol)
 - FTP (File Transfer Protocol)
 - TELNET
 - HTTP (HyperText Transfer Protocol)

Application layer

- I protocolli del livello Application sono basati sul modello di interazione **client/server**

- Per usare i servizi messi a disposizione mediante questi protocolli bisogna contattare un server, **ma come?**
 - Tutte le volte che usate il browser e richiedete delle pagine HTML di un sito, di fatto state contattando un web server remoto

 - Tutte le volte che inviate una e-mail di fatto il mail server del vostro provider contatta il mail server del provider del vostro destinatario

Domain Name System - DNS

- Gli indirizzi IP numerici sono difficili da ricordare
- Si usano quindi degli **indirizzi simbolici** che sono più significativi per l'essere umano
elios.disi.unige.it, samphrey.dcs.ed.ac.uk,
developer.netscape.com
- Questi nomi vengono tradotti in indirizzi IP numerici mediante il **Domain Name System**

Domain Name System - DNS

- Gli indirizzi simbolici hanno un formato come quello seguente

...nome5.nome4.nome3.nome2.nome1

Domain Name System - DNS

- Sono costruiti a partire da uno **schema gerarchico** di nomi basato sul concetto di dominio



Domain Name System - DNS

- Domini di primo livello (**top level**)

com	aziende
edu	università americane
gov	istituzioni governative
mil	istituzioni militari
net	fornitori d'accesso
org	organizzazioni non-profit
au	Australia
ch	Svizzera
fr	Francia
it	Italia
jp	Giappone
uk	Inghilterra
.....	

Domain Name System - DNS

- Ogni dominio deve essere in grado di “risolvere i nomi” dei calcolatori di sua competenza
- Si usano i **name server** che gestiscono la corrispondenza tra nomi simbolici e indirizzi IP numerici
- Quando un'applicazione deve collegarsi ad una risorsa di cui conosce il nome logico (ad es. albert.unige.it), invia una richiesta al suo name server locale

Domain Name System - DNS

- Il name server locale, se conosce la risposta, la invia direttamente al richiedente. Altrimenti interroga il **name server di top level**. Questi può conoscere l'indirizzo oppure inoltrare l'interrogazione ai suoi figli nella gerarchia
- Si continua con le interrogazioni fino a quando non si ottiene l'indirizzo IP numerico della risorsa
- Quando l'applicazione riceve la risposta crea una connessione TCP con la destinazione, usando l'indirizzo IP appena ricevuto

Posta elettronica

- Si basa sul protocollo SMTP (Simple Mail Transfer Protocol) e permette lo scambio dei messaggi tra gli utenti collegati alla rete

- È necessario fornire
 - l'indirizzo del mittente
 - l'indirizzo del destinatario
 - il corpo del messaggio

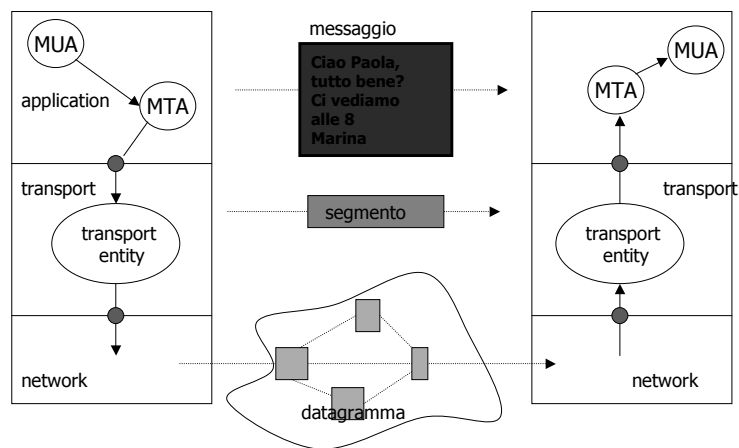
- Gli indirizzi devono avere un formato ben preciso
Esempio: `ribaudo@disi.unige.it`

Posta elettronica

- Viene implementata in Internet attraverso la cooperazione di **due sottosistemi**
 - Mail User Agent (**MUA**) che permette all'utente di comporre il proprio messaggio, di leggere i messaggi in arrivo, ...

 - Mail Transport Agent (**MTA**) che si occupa di trasportare i messaggi sulla rete fino alla consegna al Mail Transport Agent di destinazione

Posta elettronica



I servizi application sono basati sui servizi transport a loro volta basati sui servizi network

Simple Mail Transfer Protocol - SMTP

- Si occupa del **trasporto dei messaggi** in internet
 - quando l'utente, dopo aver composto il messaggio preme il pulsante **Send**, il suo client di posta contatta il **server SMTP**
 - Il **server SMTP** chiede al **DNS** l'indirizzo del **server SMTP** che si trova nel dominio del **destinatario**
 - Si apre una **connessione TCP**, poi una conversazione tra i due server e viene consegnato il messaggio

Consegna del messaggio

- Quando il destinatario usa il pulsante **Get Mail** del suo programma di posta elettronica di fatto il suo **MUA** contatta il **MTA** per scaricare la posta in arrivo

- Si possono usare due protocolli diversi
 - **POP3** (Post Office Protocol)
 - **IMAP** (Internet Mail Access Protocol)

- Qualcosa che avete già visto in SW1 ????

Trasferimento file

- Si basa sul **File Transfer Protocol (FTP)**

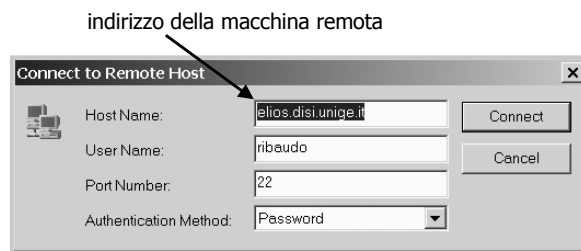
- Permette di collegarsi a siti remoti per prendere (download) / salvare (upload) file

- L'accesso può essere riservato (tramite login e password) oppure aperto a tutti (si parla di anonymous ftp)

Collegamento remoto

- **Telnet**

- permette di collegarsi a calcolatori che si trovano in località remote per lavorare interattivamente su di essi



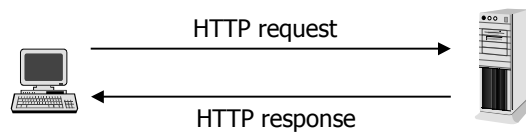
- sono necessari login e password

World Wide Web (WWW)

- Si basa sul protocollo HTTP (HyperText Transfer Protocol) che gestisce l'interazione tra un client e un server web
- Client e server si scambiano dei messaggi
 - **richieste** da parte del client
 - **risposte** da parte del server

HyperText Transfer Protocol - HTTP

- L'utente richiede una pagina residente su un server e il suo browser richiede una connessione TCP con il server
- Il server accetta la connessione iniziata dal browser
- Il browser ed il server si scambiano messaggi
- La connessione viene chiusa



HTTP

- HTTP 1.0 è **stateless**
 - il server non mantiene alcuna informazione circa le richieste già inviate da un particolare browser
 - se una pagina web contiene 10 oggetti, ci saranno 10 richieste distinte da parte del browser
- HTTP 1.1 permette di stabilire una connessione persistente

HTTP: esempio di request

- Messaggio in codice ASCII

header { **GET** nomefile.html HTTP/1.0
User-agent: Mozilla/4.0
Accept: text/html, image/gif, image/jpeg
Host: elios.disi.unige.it
Accept-language: fr
...

Oltre a GET si possono usare POST, HEAD, PUT ...

HTTP: esempio di response

header { HTTP/1.0 **200 OK**
Date: Friday, 13 Nov 2002, 12:00:12 GMT
Server: Apache/1.3.0 (Unix)
Last-Modified: ...
Content-Type: text/html
...

body { <HTML>
<HEAD>
...
</HEAD>
...

HTTP: esempio di response

- I codici di risposta notificano al browser le informazioni relative al successo o al fallimento della connessione
 - 200 ok
 - 400 bad request
 - 401 unauthorized
 - 403 forbidden
 - 404 not found
 - 500 internal server error
 - 503 service unavailable
 - 505 HTTP version not support

Uso dei servizi di rete

- Quando vi collegate ad Internet, pensate a tutto quello che succede ...
- Maggiori dettagli? Ai prossimi anni