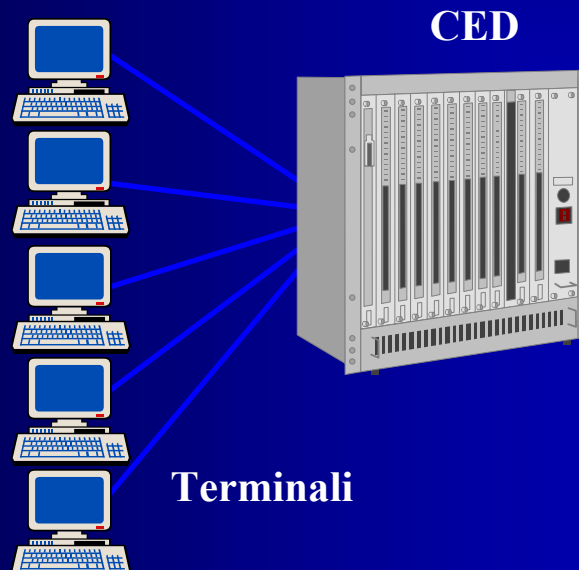


# Le reti di computer e Internet

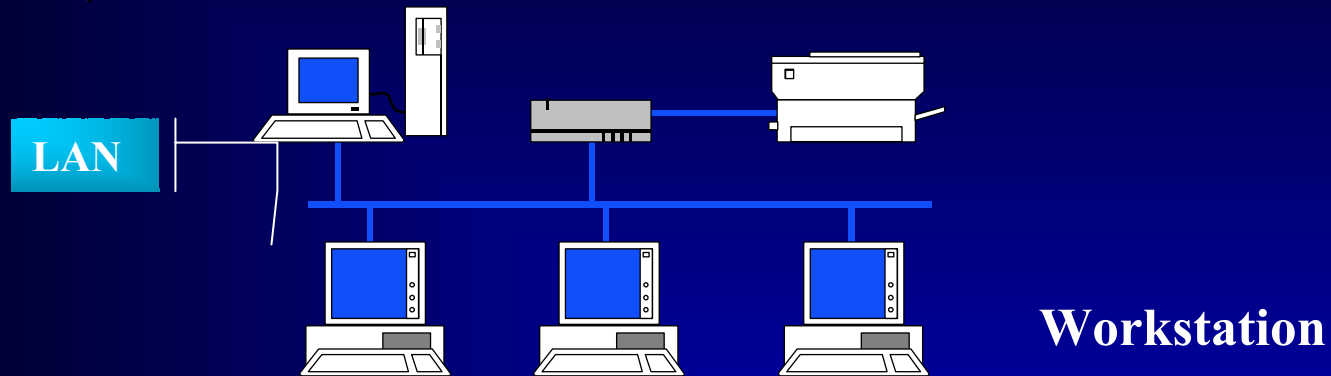
# Introduzione

Negli ultimi anni le tecnologie telefonica, radiotelevisiva e computer stanno rapidamente convergendo in unico sistema di comunicazione. In particolare, la combinazione di elaboratori e sistemi di telecomunicazione ha avuto una profonda influenza sull'organizzazione dei sistemi di calcolo. Si è passati dal vecchio modello **mainframe - terminali**, in cui la potenza di calcolo è concentrata in un unico grande elaboratore a cui si accede per mezzo di un certo numero di terminali, ...



# Introduzione

... a quello attuale in cui vi è un grande numero di elaboratori **autonomi, interconnessi** fra loro:



---

**autonomi:** significa che non deve esserci fra loro una relazione tipo master/slave (ad es., l'uno non può forzare lo spegnimento dell'altro)

---

**Interconnessi:** significa che devono essere capaci di scambiare informazioni (sfruttando un opportuno mezzo fisico).

---

# Usi delle reti

## Usi delle reti

Moltissimi sono gli usi delle reti di elaboratori, sia per le organizzazioni che per i singoli individui.

**condivisione risorse** si possono rendere disponibili a chiunque programmi e informazioni anche distanti migliaia di km

## Usi delle reti

Moltissimi sono gli usi delle reti di elaboratori, sia per le organizzazioni che per i singoli individui.

**condivisione risorse** si possono rendere disponibili a chiunque programmi e informazioni anche distanti migliaia di km

**affidabilità** si ottiene mettendo in rete sorgenti alternative delle risorse (ad es. duplicando le applicazioni e i dati su più computer). E' importante in sistemi che devono funzionare a tutti i costi (traffico aereo, centrali nucleari, sistemi militari, ecc.)

## Usi delle reti

Moltissimi sono gli usi delle reti di elaboratori, sia per le organizzazioni che per i singoli individui.

**diminuzione dei costi** una rete di personal computer costa molto meno di un mainframe. A volte alcuni elaboratori sono più potenti ed offrono agli altri dei servizi (modello client-server, vedi figura sotto)



## Usi delle reti

Moltissimi sono gli usi delle reti di elaboratori, sia per le organizzazioni che per i singoli individui.

**diminuzione dei costi** una rete di personal computer costa molto meno di un mainframe. A volte alcuni elaboratori sono più potenti ed offrono agli altri dei servizi (modello client-server, vedi figura sotto)

**scalabilità** si possono aumentare le prestazioni del sistema aumentando il numero di elaboratori (entro certi limiti)



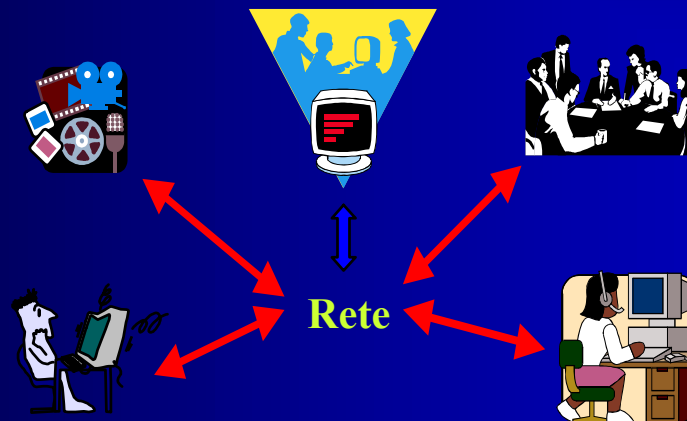


## Usi delle reti

Moltissimi sono gli usi delle reti di elaboratori, sia per le organizzazioni che per i singoli individui.

**comunicazione  
fra persone**

è possibile inviare messaggi, scambiarsi file,  
ecc.



# Usi delle reti

**Per i singoli individui:**

**accesso ad informazioni remote, ad es.:**

**accesso a servizi bancari**

**acquisti da casa**

**navigazione sul World Wide Web**

**comunicazioni fra persone:**

**posta elettronica**

**videoconferenza**

**gruppi di discussione**

**divertimento:**

**video on demand (selezione e ricezione via rete di un qualunque spettacolo tratto da un catalogo)**

**giochi interattivi (contro macchine o avversari umani)**

# Aspetti hardware delle reti

## Aspetti hardware delle reti: Local Area Network

Le reti locali (Local Area Network, LAN), in genere:

- sono possedute da una organizzazione (reti private)
- hanno un'estensione che arriva fino a qualche km
- si distendono nell'ambito di un singolo edificio o campus (non si possono, di norma, posare cavi sul suolo pubblico)
- sono usatissime per connettere PC o workstation.

## Aspetti hardware delle reti: Metropolitan Area Network

Le reti metropolitane (*Metropolitan Area Network, MAN*) hanno un'estensione tipicamente urbana

Fino a qualche anno fa erano basate essenzialmente sulle tecnologie delle reti geografiche, utilizzate su scala urbana

Recentemente è stato introdotto uno standard apposito: IEEE 802.6 o DQDB

## Aspetti hardware delle reti: Wide Area Network

Le reti geografiche (Wide Area Network, WAN) si estendono a livello di una nazione, di un continente o dell'intero pianeta. Una WAN è tipicamente costituita di due componenti distinte:

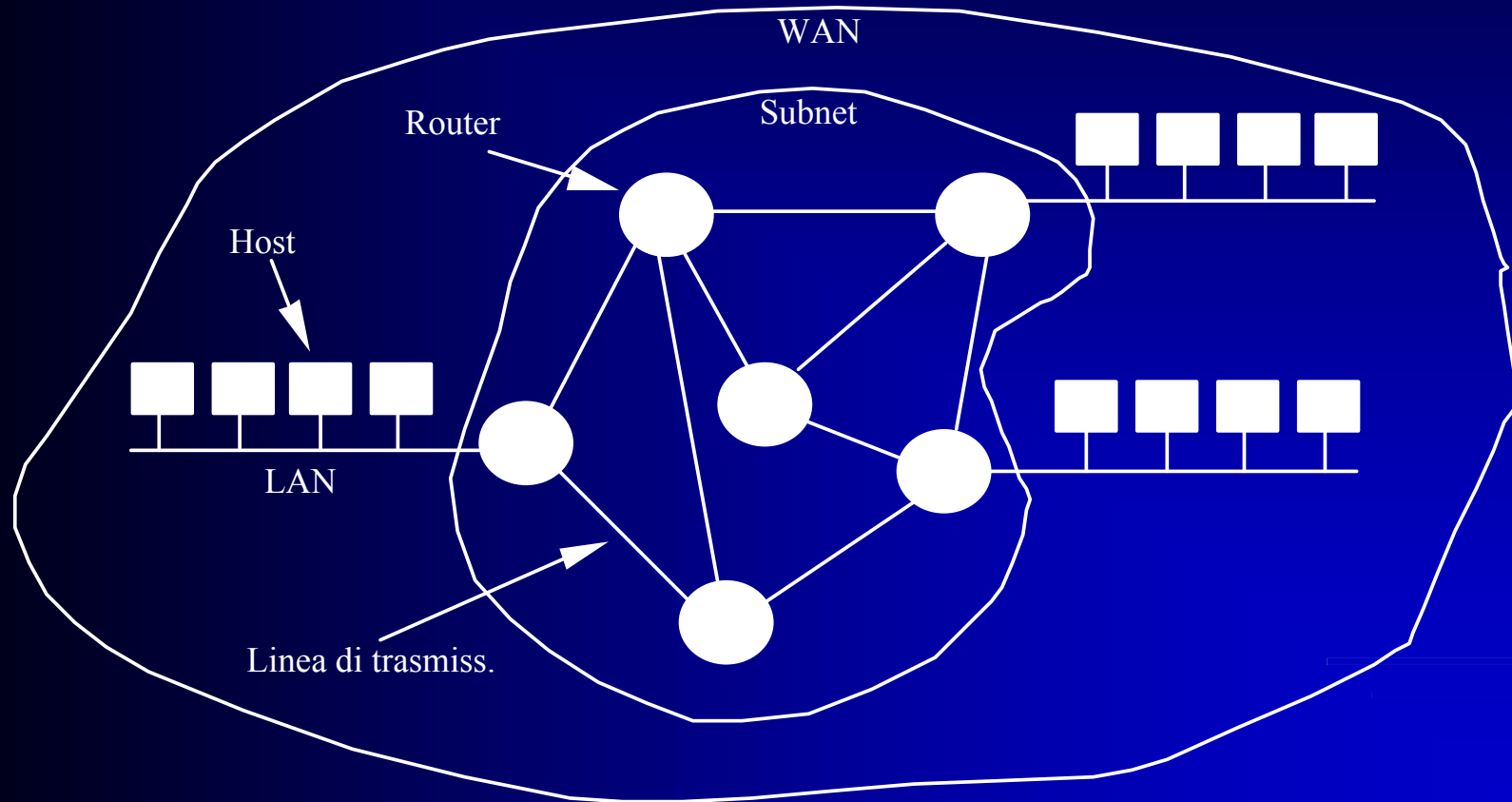
- un **insieme di elaboratori** (host oppure end system)
- una **communication subnet** (o subnet)

Di norma la subnet consiste, a sua volta, di due componenti:

- **linee di trasmissione** (dette anche circuiti, canali, trunk)
- **elementi di commutazione** (switching element, router)

## Aspetti hardware delle reti: Wide Area Network

Una tipica WAN è utilizzata per connettere più LAN fra loro



## Aspetti hardware delle reti: interconnessione di reti

- **internet** come sinonimo di internetwork, cioè la interconnessione di più reti generiche
- **Internet** (con la I maiuscola) per riferirci alla specifica internetwork, basata su TCP/IP, che ormai tutti conoscono.

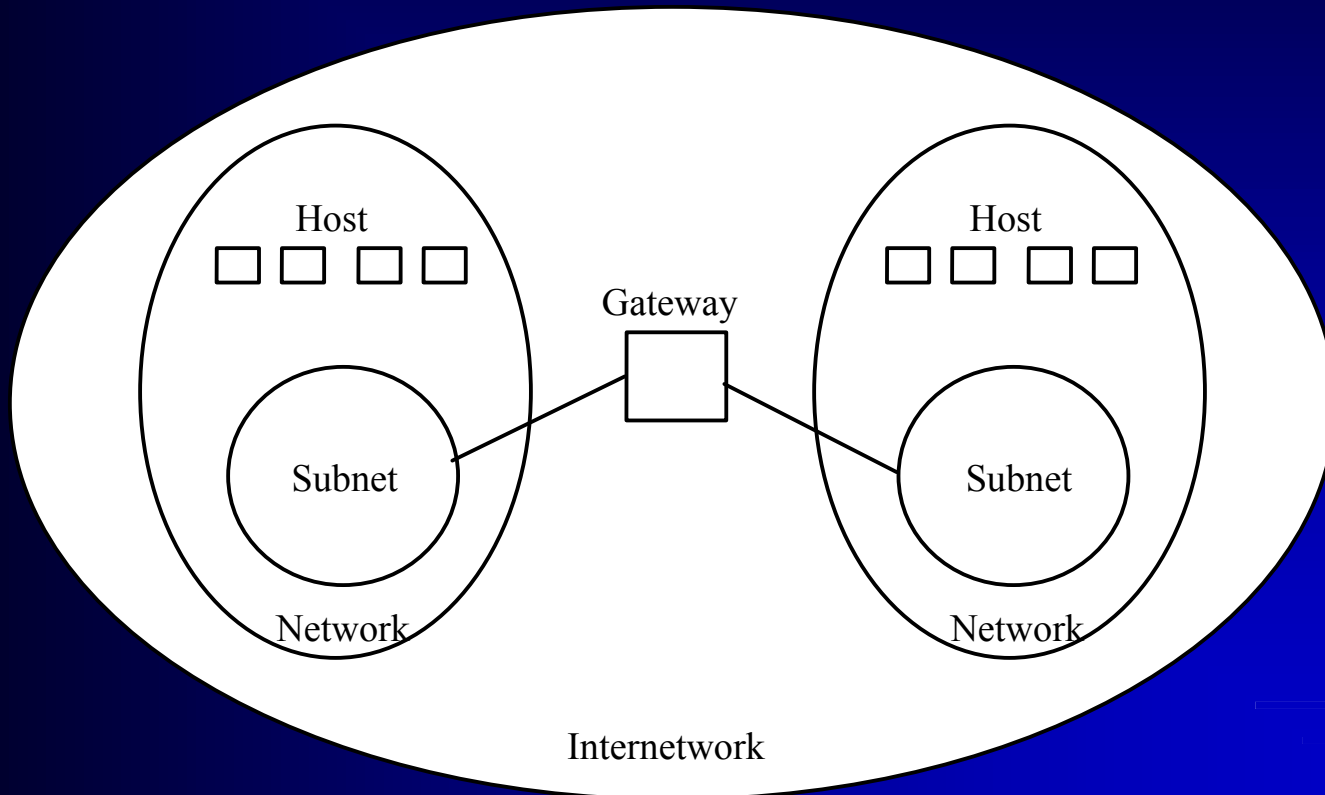
C'è molta confusione sui termini:

- **sottorete** (subnet), che nel contesto di una WAN è l'insieme dei router e delle linee di trasmissione
- **rete** (network), che altro non è che una subnet più tutti gli host collegati
- **internetwork**, che è una collezione di più network, anche non omogenee, collegate per mezzo di gateway(router).



# Aspetti hardware delle reti: interconnessione di reti

## Relazioni fra subnet, network e internetwork



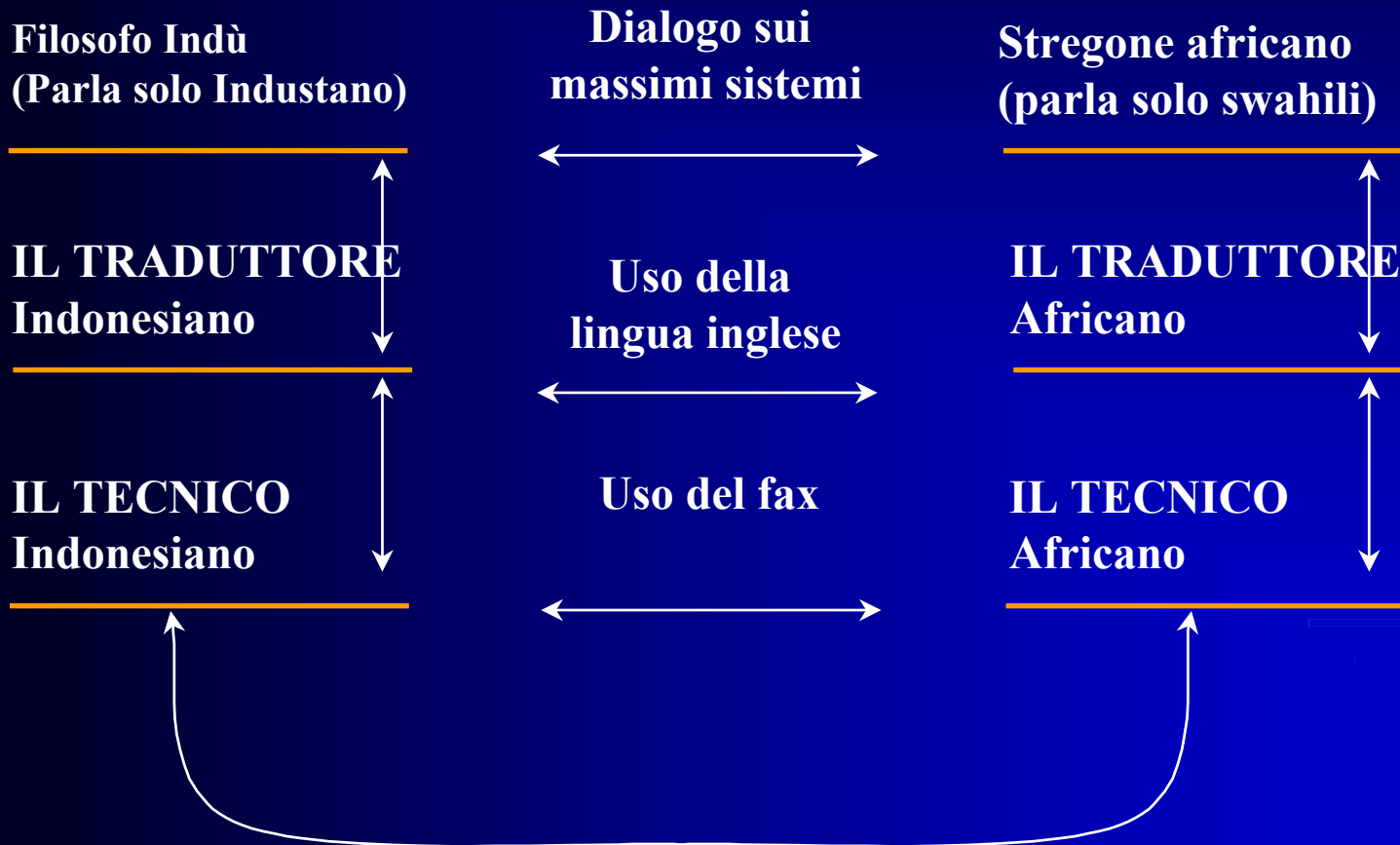
# Aspetti software delle reti: gerarchie di protocollo

Per ridurre la complessità di progetto, le reti sono in generale organizzate a livelli, ciascuno costruito sopra il precedente.



# Aspetti software delle reti: funzionamento del software di rete

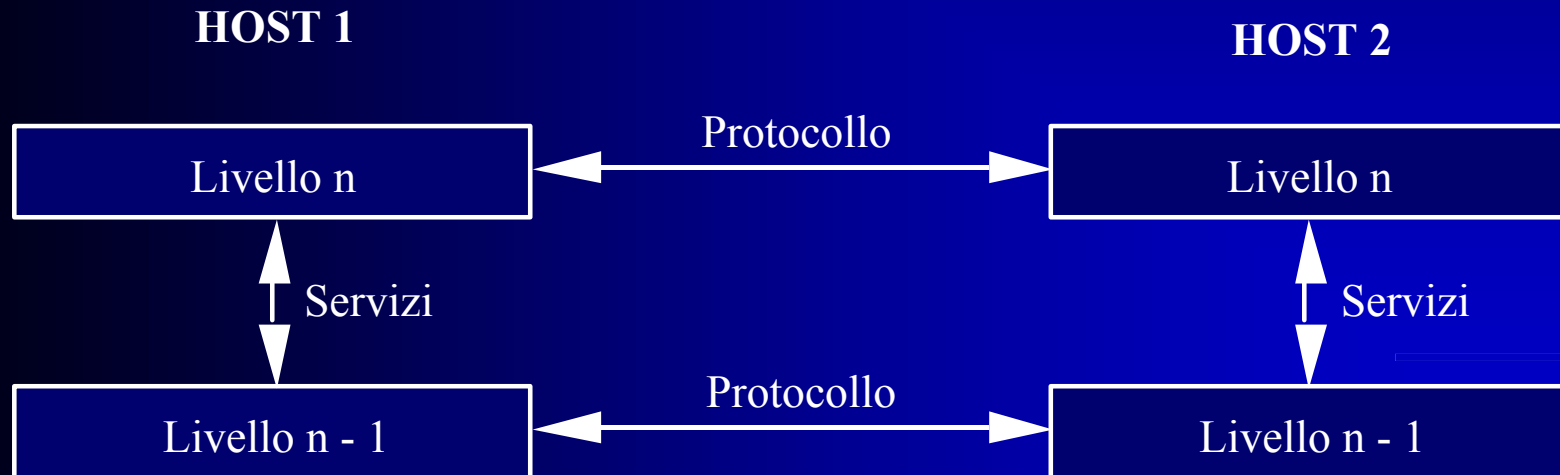
Per comprendere i meccanismi basilari di funzionamento del software di rete si può pensare alla seguente analogia umana, nella quale un filosofo indiano vuole conversare con uno stregone africano:



# Aspetti software delle reti: servizi vs. protocolli

Servizi e protocolli sono spesso confusi, ma sono concetti ben distinti.

<u><i>Servizio</i></u>	insieme di operazioni primitive che un livello offre al livello superiore. Come tali operazioni siano implementate non riguarda il livello superiore
<u><i>Protocollo</i></u>	insieme di regole che governano il formato ed il significato delle informazioni (messaggi, frame, pacchetti) che le peer entity si scambiano fra loro. Le entità usano i protocolli per implementare i propri servizi.

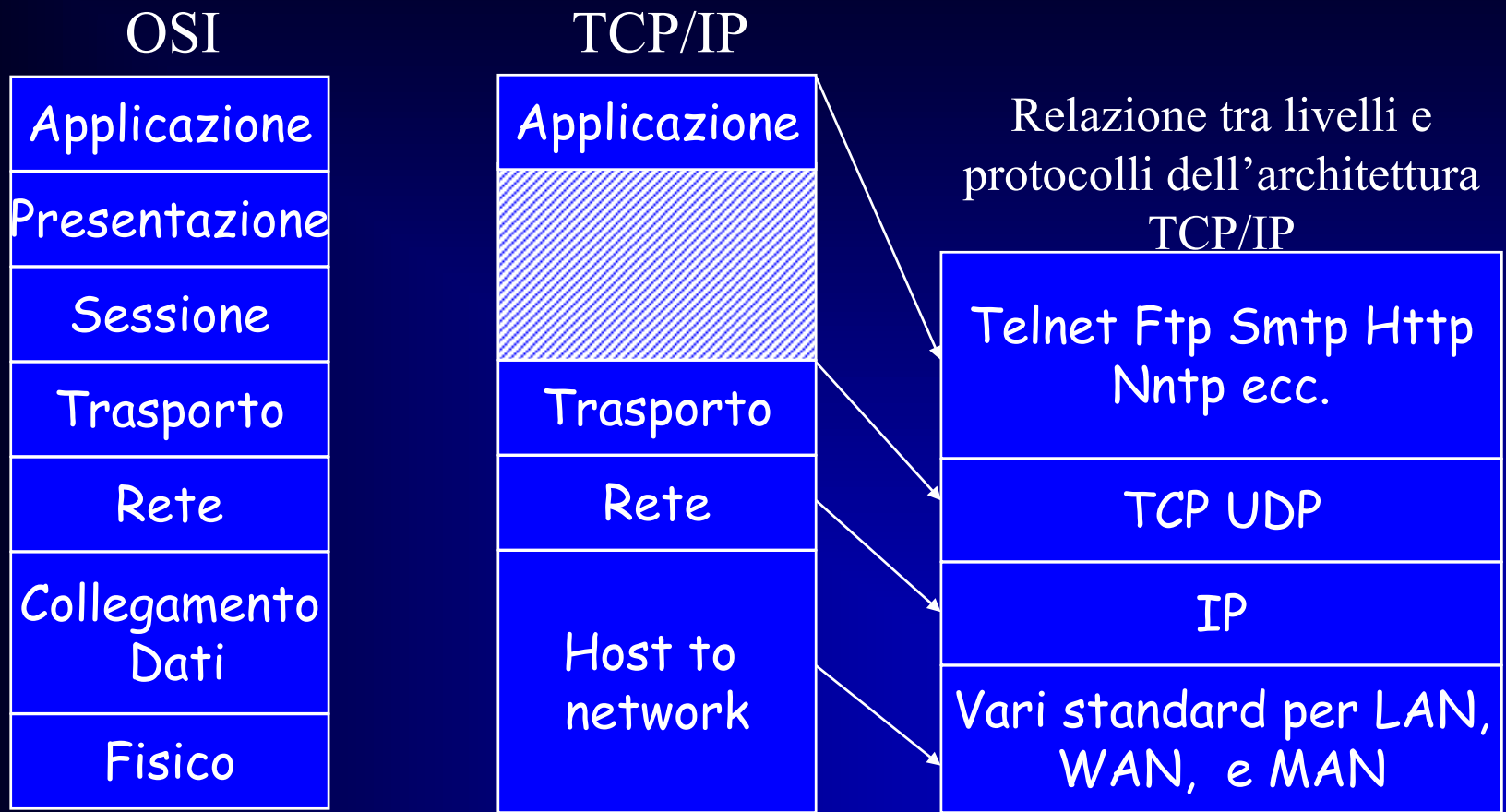


# La realtà nel mondo delle reti: modello OSI

## Il modello OSI



# La realtà nel mondo delle reti: Internet protocol suite



Relazione tra i livelli OSI e TCP/IP

## Il livello cinque (Application)

# Il livello cinque (Application)

Sopra il livello transport c'è il livello application, nel quale viene effettivamente svolto il lavoro utile per l'utente

In questo livello si trovano diverse tipologie di oggetti:

- protocolli di supporto a tutte le applicazioni, come per esempio il *DNS (Domain Name System, RFC 1034 e 1035)*;
- protocolli di supporto ad applicazioni di tipo standardizzato, come ad esempio:
  - *SNMP (Simple Network Management Protocol, RFC 1157) per la gestione della rete;*
  - *FTP (File Transfer Protocol, RFC 959) per il trasferimento di file;*
  - *SMTP e POP3 (Simple Mail Transfer Protocol, RFC 821, e Post Office Protocol, RFC 1225) per la posta elettronica;*
  - *HTTP (HyperText Transfer Protocol, RFC 1945) alla base del World Wide Web (WWW);*



## II DNS

## II DNS

- riferirsi a una risorsa (sia essa un host oppure l'indirizzo di posta elettronica di un utente) utilizzando un indirizzo IP numerico (della forma x.y.z.w) è estremamente scomodo
- si è creato un meccanismo tramite il quale tali risorse possono essere identificate tramite un *nome logico*, cioè una stringa di caratteri (molto più comprensibile per un essere umano) quale ad esempio:
  - **sparc1.unimi.it** (riferimento ad un host);
  - **john@cern.ch** (indirizzo di posta elettronica)

## II DNS

Il funzionamento, in breve, è il seguente:

- quando un'applicazione deve collegarsi ad una risorsa di cui conosce il nome logico (ad es. `sparc1.unimi.it`), invia una richiesta al *DNS server* locale;
- il *DNS server* locale,
  - se conosce la risposta, la invia direttamente al richiedente
  - altrimenti interroga a sua volta un *DNS server* di livello superiore, e così via. Quando finalmente arriva la risposta, il *DNS server* locale la passa al richiedente;
- quando l'applicazione riceve la risposta (costituita del numero IP della risorsa in questione) crea una connessione TCP con la destinazione, usando l'indirizzo IP testé ricevuto.

## II DNS

- Nel caso di un host, la forma del nome logico è costituita da un certo numero di sottostringhe separate da punti, come nell'esempio seguente:

`host.subdomain3.subdomain2.subdomain1.topleveldomain`

dove:

- la prima sottostringa (quella più a sinistra) identifica il nome dell'host;
- le altre sottostringhe (tranne quella più a destra) identificano ciascuna un sottodominio del dominio di cui alla sottostringa seguente;
- l'ultima sottostringa (quella più a destra) identifica il top-level domain di appartenenza.

## II DNS

Per gli USA sono definiti, fra gli altri, i seguenti top-level domain:

<i>com</i>	aziende
<i>edu</i>	università
<i>gov</i>	istituzioni governative
<i>mil</i>	istituzioni militari
<i>net</i>	fornitori d'accesso
<i>org</i>	organizzazioni non-profit

## II DNS

Fuori degli USA, ogni nazione ha un suo top-level domain:

<i>au</i>	<b>Australia</b>
<i>ch</i>	<b>Svizzera</b>
<i>fr</i>	<b>Francia</b>
<i>it</i>	<b>Italia</b>
<i>jp</i>	<b>Giappone</b>
<i>uk</i>	<b>Inghilterra</b>

## II DNS

- **L'estensione di un dominio è del tutto indipendente da quella delle reti e sottoreti IP.**
- **Ogni dominio ha la responsabilità di fornire il servizio DNS per quanto di propria competenza. Ossia, deve poter rispondere a interrogazioni riguardanti tutti gli host contenuti nel dominio stesso.**

## II DNS

- Ciò si ottiene predisponendo un *name server* (o più di uno), che è un processo in grado di gestire le seguenti informazioni:
  - informazioni di corrispondenza fra nomi simbolici e indirizzi IP. Per ogni host del dominio esiste un *resource record* che contiene tali informazioni; tale record è detto *authoritative record*, in quanto è gestito dal DNS server responsabile del dominio (che è supposto fornire sempre informazioni corrette e aggiornate);
  - l'identità dei name server responsabili dei sottodomini inclusi nel dominio, così da poter inviare loro le richieste che gli pervengono dall'alto della gerarchia;
  - l'identità del name server responsabile del dominio di livello immediatamente superiore, così da potergli inviare le richieste che gli pervengono dal basso della gerarchia.



## II DNS

- Quando una risposta ritorna indietro, essa viene tenuta dal server in una sua cache per un certo periodo;
- qui costituisce un nuovo record, detto *cached record* perché contiene della informazione che potrebbe anche divenire, col passare del tempo, obsoleta e non più corretta.

## II World Wide Web

## Il World Wide Web

- Il *World Wide Web* (detto anche *Web*, *WWW* o *W3*) è nato al Cern nel 1989 per consentire una agevole cooperazione fra i gruppi di ricerca di fisica sparsi nel mondo.
- E' un'architettura software volta a fornire l'accesso e la navigazione a un enorme insieme di documenti collegati fra loro e sparsi su milioni di elaboratori.
- Tale insieme di documenti forma un *ipertesto* (*hypertext*), cioè un testo che viene percorso in modo non lineare.

## Il World Wide Web

I documenti che costituiscono l'ipertesto gestito dal Web sono detti *pagine web*, e possono contenere, oltre a normale testo formattato, anche:

- rimandi (detti *link* o *hyperlink*) ad altre pagine web;
- immagini fisse o in movimento;
- suoni;
- scenari tridimensionali interattivi;
- codice eseguibile localmente.

## Architettura client-server del Web

**Il Web è una architettura software di tipo *client-server*, nella quale sono previste due tipologie di componenti software, ciascuno avente compiti ben definiti:**

- *client*
- *server*

# Client

- **Il client (o user agent) è lo strumento a disposizione dell'utente che gli permette l'accesso e la navigazione nell'ipertesto del Web.**
- **Esso ha varie competenze:**
  - **trasmettere all'opportuno server le richieste di reperimento dati che derivano dalle azioni dell'utente;**
  - **ricevere dal server le informazioni richieste;**
  - **visualizzare il contenuto della pagina Web richiesta dall'utente, gestendo in modo appropriato tutte le tipologie di informazioni in esse contenute;**
  - **consentire operazioni locali sulle informazioni ricevute (ad esempio salvarle su disco, stamparle).**

# Client

I client vengono comunemente chiamati *browser* (sfogliatori).

Gli esempi più noti sono:

- NCSA Mosaic (il primo);
- Netscape Navigator;
- Microsoft Internet Explorer.

# Client

- In generale è troppo complicato e costoso (sarebbero necessari aggiornamenti troppo frequenti) sviluppare un browser che sappia gestire direttamente tutti i tipi di informazioni presenti sul Web, poiché essi sono in continuo e rapido aumento.
- Per questa ragione, di norma i browser gestiscono direttamente solo alcune tipologie di informazioni, quali:
  - testo formattato;
  - immagini fisse;
  - codice eseguibile.



## Client

- **Viceversa, di norma gli altri tipi di informazioni vengono gestiti in uno (o entrambi) dei seguenti modi:**
  - **consegnandoli a un programma esterno (*helper*) che provvederà alla corretta gestione (ad esempio, un file contenente un filmato verrà consegnato a un programma per il playback di filmati);**
  - **se il browser ha un'architettura modulare le sue funzionalità possono essere estese per mezzo di *plug-in*, ossia librerie di codice eseguibile specializzato che possono essere caricate in memoria secondo le necessità. In questa situazione, se il necessario plug-in è installato, il browser provvede a caricarlo e gli affida la gestione delle informazioni da trattare.**

# Server

- **Il server è tipicamente un processo in esecuzione su un elaboratore. Esso, di norma, è sempre in esecuzione (tranne che in situazioni eccezionali) ed ha delle incombenze molto semplici, almeno in linea di principio. Infatti deve:**
  - **rimanere in ascolto di richieste da parte dei client;**
  - **fare del suo meglio per soddisfare ogni richiesta che arriva:**
    - **se possibile, consegnare il documento richiesto;**
    - **altrimenti, spedire un messaggio di notifica di errore (documento non esistente, documento protetto, ecc.).**

# Server

- **Nonostante la apparente semplicità di tale compito, la realizzazione di un server non è banale, perché:**
  - **deve fare il suo lavoro nel modo più efficiente possibile, dunque deve essere implementato con un occhio di riguardo alle prestazioni;**
  - **deve essere in grado di gestire molte richieste contemporaneamente, e mentre fa questo deve continuare a rimanere in ascolto di nuove richieste.**
- **Il secondo requisito in particolare implica una qualche forma di concorrenza nel lavoro del server. Essa si può ottenere in vari modi, anche in funzione delle caratteristiche del sistema operativo sottostante.**

## Standard utilizzati nel Web

**Ci sono tre standard principali che, nel loro insieme, costituiscono l'architettura software del Web:**

- **sistema di indirizzamento basato su Uniform Resource Locator (URL): è un meccanismo standard per fare riferimento alle entità indirizzabili (risorse) nel Web, che possono essere:**
  - documenti (testo, immagini, suoni, ecc.);
  - programmi eseguibili (vedremo poi);
- **linguaggio HTML (HyperText Markup Language): è il linguaggio per la definizione delle pagine Web;**
- **protocollo HTTP (HyperText Transfer Protocol): è il protocollo che i client e i server utilizzano per comunicare.**

# URL

- **Una URL costituisce un riferimento a una qualunque risorsa accessibile nel Web.**
- **Tale risorsa ovviamente risiede da qualche parte, ed è in generale possibile accedervi in vari modi.**
- **Dunque, una URL deve essere in grado di indicare:**
  - **come si vuole accedere alla risorsa;**
  - **dove è fisicamente localizzata la risorsa;**
  - **come è identificata la risorsa.**

# URL

- Per queste ragioni, una URL è fatta di 3 parti, che specificano:
  - il *metodo di accesso*;
  - l'*host* che detiene la risorsa;
  - l'*identità* della risorsa.
- Un tipico esempio di una URL è:

**<http://somewhere.net/products/index.html>**

nella quale:

<code>http://</code>	è il metodo di accesso
<code>somewhere.net</code>	è il nome dell'host
<code>/products/index.html</code>	è l'identità della risorsa

# URL

## Metodo di accesso

- Indica il modo di accedere alla risorsa, cioè che tipo di protocollo bisogna usare per colloquiare col server che controlla la risorsa.
- I metodi di accesso più comuni sono:

<i>http</i>	protocollo nativo del Web
<i>ftp</i>	file transfer protocol
<i>news</i>	protocollo per l'accesso ai gruppi di discussione
<i>gopher</i>	vecchio protocollo per il reperimento di informazioni; concettualmente simile al Web, gestisce solo testo
<i>mailto</i>	usato per spedire posta
<i>telnet</i>	protocollo di terminale virtuale, per effettuare login remoti
<i>file</i>	accesso a documenti locali

# URL

## Nome dell'host

- Può essere l'*indirizzo IP* numerico o, più comunemente, il *nome DNS* dell'host a cui si vuole chiedere la risorsa.
- Dopo il nome dell'host può essere incluso anche un *numero di port*. Se non c'è, si intende il port 80 (che è il default). Ad esempio:

**<http://somewhere.net:8000/products/index.html>**

- In questo modo si possono avere, sullo stesso host, diversi server Web in ascolto su diverse porte.



# URL

Identità della risorsa

- **Consiste, nella sua forma più completa, della specifica del nome di un file e del cammino che porta al direttorio in cui si trova.**

**Ad esempio, la URL:**

**<http://somewhere.net/products/toasters/index.html>**

**specifica il file *index.html* contenuto nel direttorio *toasters*, a sua volta contenuto nel direttorio *products* il quale si trova nel direttorio radice dell'host *somewhere.net*.**

# URL

## Identità della risorsa

- **Esistono alcune regole per il completamento di URL non interamente specificate:**
  - **se manca il nome del direttorio, si assume quello della pagina precedente;**
  - **se manca il nome del file (ma c'è quello del direttorio), a seconda del server:**
    - **si restituisce un file prefissato del direttorio specificato (index.html, default.html oppure welcome.html);**
    - **se tale file non esiste, talvolta si restituisce un elenco dei file nel direttorio.**

# Linguaggio HTML

- Il linguaggio per la formattazione di testo HTML è una specializzazione del linguaggio *SGML* (*Standard Generalized Markup Language*) definito nello standard ISO 8879.
- HTML è specializzato nel senso che è stato progettato appositamente per un utilizzo nell'ambito del Web.

# Linguaggio HTML

- Un *markup language* si chiama così perché i comandi (*tag*) per la formattazione sono inseriti in modo esplicito nel testo, a differenza di quanto avviene in un word processor *WYSIWYG* (*What You See Is What You Get*), nel quale il testo appare visivamente dotato dei suoi formati, come fosse stampato. TROFF e TeX sono altri markup language, mentre ad esempio Microsoft Word è WYSIWYG.

Per esempio in HTML il testo:

...questo è **<B>grassetto</B>** e questo no...

indica che la parola grassetto deve essere visualizzata in *grassetto* (*bold*). Quindi il testo in questione dovrà apparire come segue:

...questo è grassetto e questo no...

# Linguaggio HTML

- Il ruolo di HTML è quindi quello di definire il modo in cui deve essere visualizzata una pagina Web (detta anche *pagina HTML*), che tipicamente è un documento di tipo testuale contenente opportuni tag di HTML.

# Linguaggio HTML

**Il client, quando riceve una pagina compie le seguenti operazioni:**

- **interpreta i tag presenti nella pagina;**
- **formatta la pagina di conseguenza, provvedendo automaticamente ad adattarla alle condizioni locali (risoluzione dello schermo, dimensione della finestra, profondità di colore, ecc.);**
- **mostra la pagina formattata sullo schermo.**

## Il protocollo HTTP

- Il protocollo HTTP sovrintende al dialogo fra un client e un server web, ed è il linguaggio nativo del Web.
- HTTP non è ancora uno standard ufficiale. Infatti, HTTP 1.0 (rfc 1945) è *informational*, mentre HTTP 1.1 (rfc 2068) è ancora in fase di proposta.
- HTTP è un *protocollo ASCII*, cioè i messaggi scambiati fra client e server sono costituiti da sequenze di caratteri ASCII (e questo è un problema se è necessaria la riservatezza delle comunicazioni).

# Il protocollo HTTP

**Il protocollo prevede che ogni singola interazione fra client e server si svolga secondo il seguente schema:**

- **apertura di una connessione di livello transport fra client e server (TCP è lo standard di fatto, ma qualunque altro può essere usato);**
- **invio di una singola richiesta da parte del client, che specifica la URL desiderata;**
- **invio di una risposta da parte del server e dei dati di cui alla URL richiesta;**
- **chiusura della connessione di livello transport.**

**Il protocollo è di tipo *stateless*, cioè non è previsto il concetto di *sessione* all'interno della quale ci si ricorda dello stato dell'interazione fra client e server. Ogni singola interazione è storia a se ed è del tutto indipendente dalle altre.**



# Il protocollo HTTP

La richiesta del client

- **Quando un client effettua una richiesta invia diverse informazioni:**
- **il metodo (cioè il comando) che si chiede al server di eseguire;**
- **il numero di versione del protocollo HTTP in uso;**
- **l'indicazione dell'oggetto al quale applicare il comando;**
- **varie altre informazioni, fra cui:**
  - **il tipo di client;**
  - **i tipi di dati che il client può accettare.**

# Il protocollo HTTP

La risposta del server

- **La risposta del server è articolata in più parti, perché c'è un problema di fondo: come farà il client a sapere in che modo dovrà gestire le informazioni che gli arriveranno?**
- **Ovviamente, non si può mostrare sotto forma di testo un'immagine o un file sonoro! Dunque, si deve informare il client sulla natura dei dati che gli arriveranno prima di iniziare a spedirglieli.**
- **Per questo motivo la risposta consiste di 3 parti:**
  - **una *riga di stato*, che indica quale esito ha avuto la richiesta (tutto ok, errore, ecc.);**
  - **delle *metainformazioni* che descrivono la natura delle informazioni che seguono;**
  - **le *informazioni vere e proprie* (ossia, l'oggetto richiesto).**