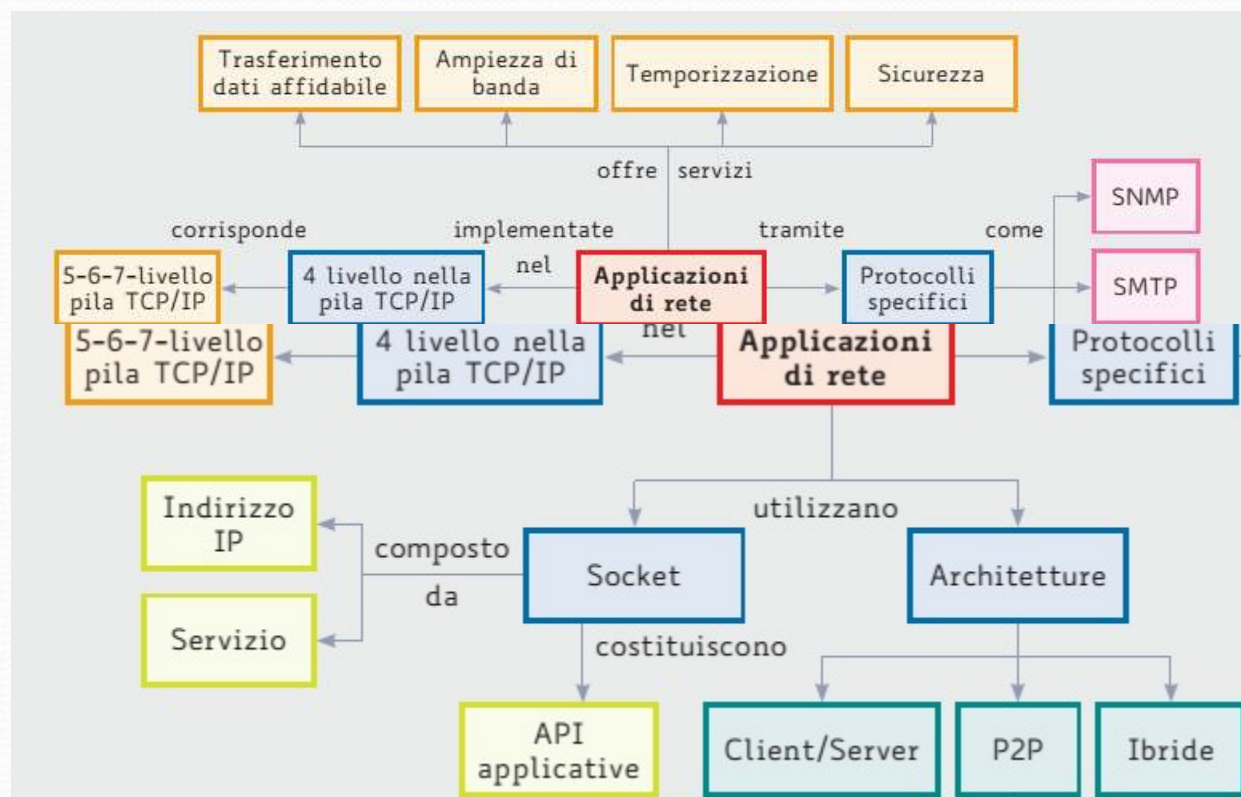


# Architetture e Protocolli di rete

# Indice

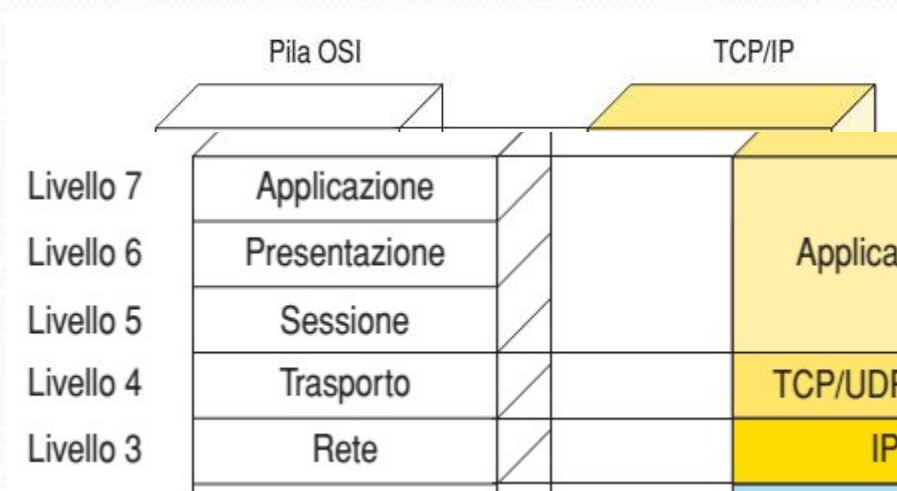
- concetto di applicazione di rete
- tipologie di applicazione
- i protocolli per le applicazioni di rete

# Mappa



# Il modello ISO/OSI e le applicazioni

- Il principale scopo delle reti è quello di condividere dati mediante applicazioni.



- Nel modello **ISO/OSI** e **TCP** il livello delle applicazioni si occupa di implementare le applicazioni di rete che vengono utilizzate dall'utente finale
- Il programmatore non si deve preoccupare dei livelli inferiori ma soltanto utilizzare le **primitive di comunicazione**

# Il modello ISO/OSI e le applicazioni

- Il livello applicazione implementa i vari protocolli, tra cui:
  - SNMP: Simple Network Management Protocol;
  - SMTP: Simple Mail Transfer Protocol;
  - POP<sub>3</sub>: Post Office Protocol;
  - FTP: File Transfer Protocol;
  - HTTP: HyperText Transfer Protocol;
  - DNS: Domain Name System.

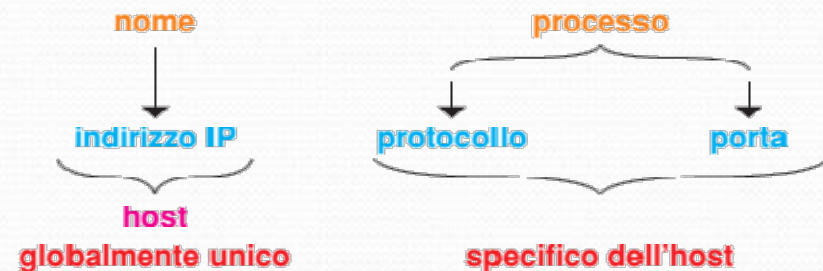


# Applicazioni di rete

- L'applicazione di rete prende anche il nome di applicazione distribuita dato che non viene eseguita su di un solo elaboratore (concentrata).

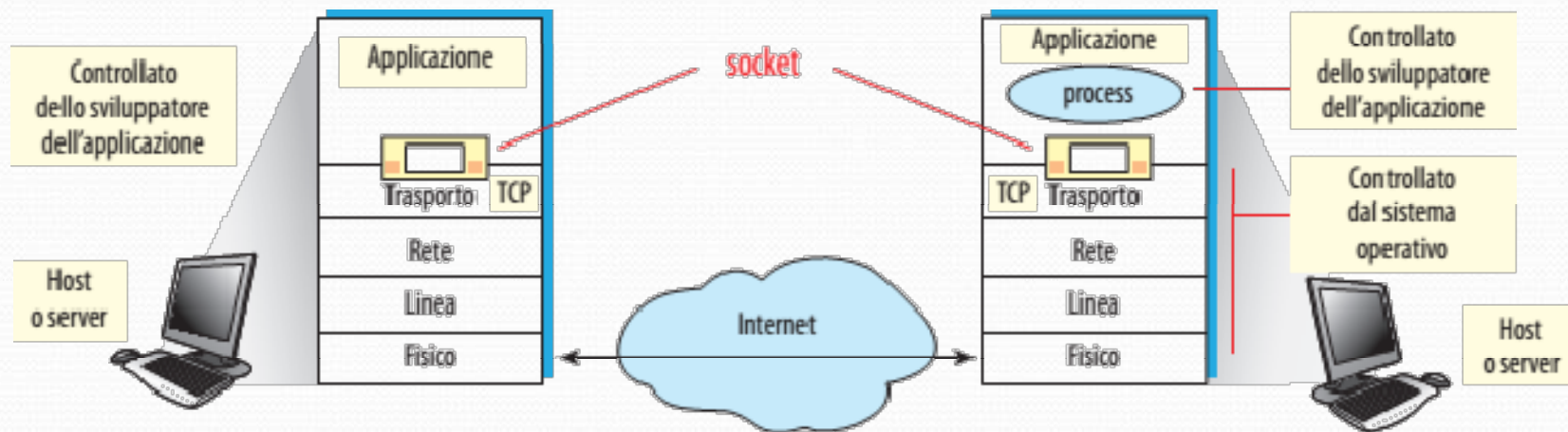
# Identificazione mediante socket

- L'identificazione deve tenere conto di due informazioni, l'indirizzo IP e il processo appartenente a quel determinato host:
  - un'identificazione del nodo su cui opera il processo con cui si desidera comunicare;
  - un'identificazione del particolare processo all'interno di quel nodo.



# Identificazione mediante socket

- l'identificazione univoca avviene conoscendo sia l'indirizzo IP che il numero di porta associato al processo in esecuzione su un host
- questo meccanismo prende il nome di meccanismo dei socket





# Identificazione mediante socket

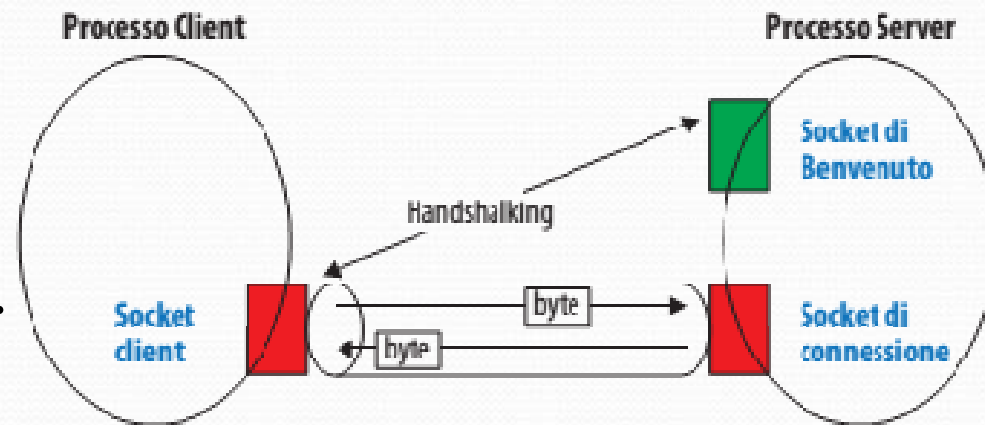
- un socket consente di comunicare attraverso la rete utilizzando la pila TCP/IP ed è quindi parte integrante del protocollo
- le API mettono a disposizione del programmatore gli strumenti necessari a codificare la connessione e l'utilizzo del protocollo di comunicazione.

# Architettura per l'applicazione di rete

- Il primo passo che il programmatore deve effettuare per progettare una applicazione di rete è la scelta della architettura dell'applicazione
- le principali caratteristiche delle architetture attualmente utilizzate sono:
  - client-server
  - peer-to-peer (P2P)
  - architetture ibride (dove convivono client-server e P2P)

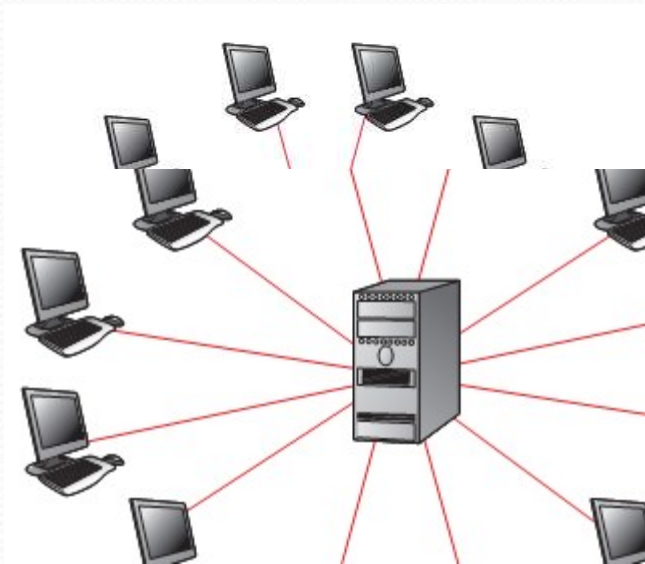
# Architettura Client-Server

- Nell'architettura client-server la caratteristica principale è che deve sempre esserci un server attivo che offre un servizio restando in attesa che uno o più client si connettano a esso per poter rispondere alle richieste che gli vengono effettuate.



# Architettura Client-Server

- Un tipico esempio di questa architettura è il WWW

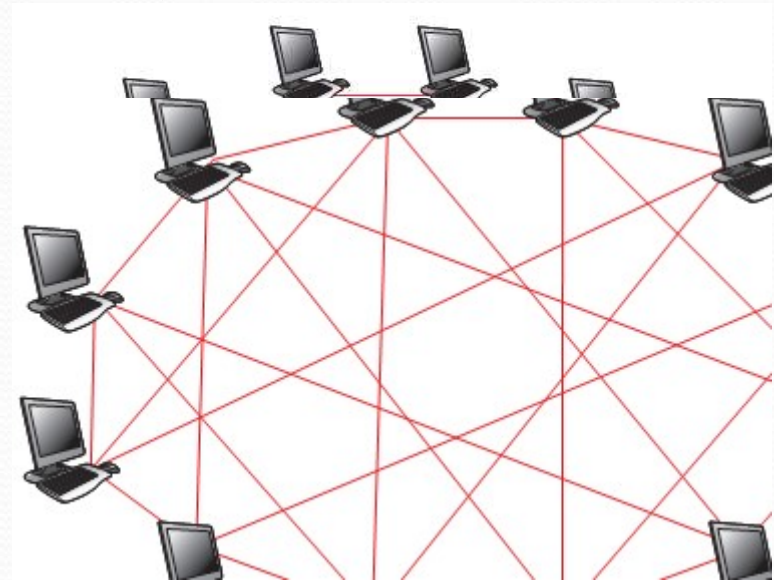


- Il server deve sempre essere attivo e deve possedere un indirizzo IP fisso dove può essere raggiunto dagli host client
- l'indirizzo IP deve essere statico, contrariamente a quello dei client che generalmente è dinamico



# P2P decentralizzato

- Nella architettura completamente decentralizzata un peer ha sia funzionalità di client che di server (hanno funzionalità simmetrica e sono anche chiamati servent)
- è impossibile localizzare una risorsa mediante un indirizzo IP statico
- vengono effettuati nuovi meccanismi di indirizzamento, definiti a livello superiore rispetto al livello IP.



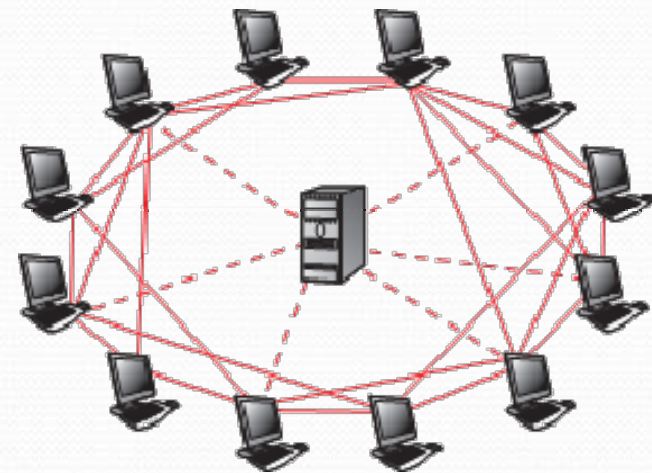


# P2P decentralizzato

- I peer sono responsabili:
  - di conservare i dati e le informazioni (il server centrale non memorizza file)
  - di informare il server del contenuto dei file che intendono condividere
  - di permettere ai peer che lo richiedono di scaricare le risorse condivise.

# P2P centralizzato

Il P2P centralizzato è un compromesso tra il determinismo del modello client-server e la scalabilità del sistema puro: ha un server centrale (directory server) che conserva informazioni sui peer (index, cioè il mapping risorse-peer) e risponde alle richieste su quelle informazioni effettuando quindi la ricerca in modalità centralizzata chiamati leaf



# Strato di trasporto

- Le applicazioni richiedono allo strato di trasporto un insieme di servizi specifici oltre ai protocolli necessari per realizzarli, che possono essere standard o realizzati ad hoc.
- Tutti i protocolli, sia standard che specifici, hanno in comune una particolarità: trasferire dei messaggi da un punto a un altro della rete.

# Servizi offerti dallo strato di trasporto

- Ogni applicazione deve scegliere tra i protocolli di trasporto alle sue specifiche esigenze e cioè:
  - trasferimento dati affidabile
  - ampiezza di banda
  - temporizzazione
  - sicurezza



# Trasferimento dati affidabile

- A tale scopo il livello di trasporto mette a disposizione due protocolli:
  - UDP User Datagram Protocol: il protocollo di trasporto senza connessione
  - TCP Transmission Control Protocol: il protocollo orientato alla Connessione



# Ampiezza di banda (Bandwidth)

- Alcune per poter “funzionare” hanno bisogno di avere una garanzia sulla larghezza di banda minima disponibile
- possono richiedere un throughput garantito di  $r$  bsp
  - si pensi alla trasmissione di un evento in diretta in una Web-TV.

# Temporizzazione

- Alcune applicazioni, come la telefonia VoIP, i giochi interattivi, gli ambienti virtuali, per essere “realistiche” ammettono solo piccoli ritardi  
Si utilizza un protocollo di trasporto in tempo reale, come RTP (Real Time Protocol)
- E' in grado di studiare i ritardi di rete e calibrare gli apparati e i collegamenti per garantire di restare nei limiti di tempo prefissati

# Sicurezza

- Il problema della sicurezza nelle reti riveste una grande importanza dato che le reti per loro natura non sono sicure
- molteplici sono le minacce e i pericoli per i dati che sono presenti nei diversi host e che circolano sulla rete.
- Garantire la sicurezza di un sistema informativo significa impedire a potenziali soggetti attaccanti l'accesso o l'uso non autorizzato di informazioni e risorse.