

Università di Roma Tor Vergata  
Corso di Laurea triennale in Informatica  
**Sistemi operativi e reti**

A.A. 2022-2023

Pietro Frasca

Parte II

Lezione 5 (29)

Giovedì 23-03-2023

# Gli strati di Internet e i servizi che forniscono

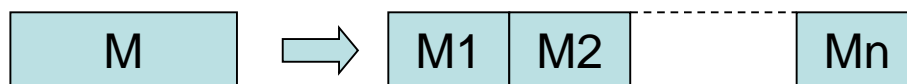
## Strato di applicazione

- Lo strato di applicazione consiste in applicazioni di rete.
- Molte applicazioni di rete sono diventate protocolli standard, come l'HTTP utilizzato nel Web, l'SMTP per la posta elettronica e l'FTP per il trasferimento dei file.
- Un'applicazione di rete può usare, oltre i servizi dello strato di trasporto, anche i servizi offerti dai protocolli di questo livello. Ad esempio, spesso le applicazioni peer to peer, utilizzano l'HTTP o l'FTP per il trasferimento di file.



# Strato di trasporto

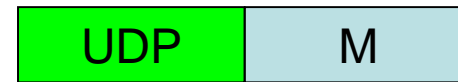
- Lo strato di trasporto fornisce il servizio di trasporto dei messaggi dello strato di applicazione fra le estremità di un'applicazione (comunicazione tra processi remoti).
- I protocolli di trasporto più usati in Internet sono il **TCP**, l'**UDP** e l'**SCTP** (che non tratteremo).
- Il TCP fornisce alle applicazioni un servizio orientato alla connessione **affidabile** che garantisce il corretto trasferimento dei messaggi dello strato di applicazione. Fornisce il **controllo del flusso** e il **controllo della congestione**. Il TCP, inoltre, frammenta i messaggi più grandi di una determinata dimensione, in segmenti più piccoli nel lato mittente e li riassume nella destinazione.



- Il protocollo UDP fornisce alle applicazioni un **servizio senza connessione**, trasmette i dati senza alcuna garanzia. Se il messaggio da spedire è troppo grande, deve essere frammentato a livello di applicazione.



**Segmento TCP**



**Segmento UDP**

## Strato di rete

Lo strato di rete è responsabile dell'instradamento dei datagram da un host mittente ad un host destinazione.

- Lo strato di rete ha due componenti principali.
  - **Il protocollo IP** che definisce i campi nel datagram IP e le operazioni che host e router eseguono su questi campi.
  - **vari protocolli di instradamento** che decidono il percorso che i datagram devono seguire fra sorgente e destinazione.
- Come descritto, Internet è una rete di reti, e pertanto in ciascuna rete potrà funzionare un protocollo di instradamento piuttosto che un altro.

- I protocolli dello strato di trasporto di Internet (TCP e UDP) in un host sorgente passano un segmento dello strato di trasporto allo strato di rete.
- Lo strato di rete fornisce quindi il servizio di instradare il segmento verso la sua destinazione. Quando il pacchetto arriva a destinazione, IP passa il segmento allo strato di trasporto all'interno della destinazione.



**Datagram IP**



**Datagram IP**

## **Strato di collegamento**

- Per trasferire un pacchetto da un nodo (host o router) al successivo sul percorso, lo strato di rete utilizza i servizi dello strato di collegamento. In particolare, in ciascun nodo, IP passa il datagramm allo strato di collegamento, che lo invia al nodo successivo lungo il percorso.

- Arrivato al nodo successivo, lo strato di collegamento passa il datagram allo strato di rete.
- I servizi forniti dallo strato di collegamento dipendono dallo specifico protocollo dello strato di collegamento utilizzato sul link. Per esempio, alcuni protocolli, come Wi-Fi, forniscono un trasferimento affidabile dal nodo trasmittente al nodo ricevente mentre altri, come Ethernet, non implementano un servizio di trasferimento affidabile.
- Esempi di tecnologie dello strato di collegamento sono Wi-Fi, Ethernet e PPP. Poiché spesso i datagram devono attraversare molti link per passare dalla sorgente alla destinazione, un datagram può essere gestito da diversi protocolli di collegamento lungo il suo percorso. Per esempio, un datagram può essere gestito da Ethernet su un link e da PPP sul link successivo.

- IP riceverà servizi differenti da ciascuno dei diversi protocolli dello strato di collegamento.



**Frame Ethernet**

## Strato fisico

Mentre il compito dello strato di collegamento è di trasferire frame da un nodo della rete a quello successivo, il compito dello strato fisico è di trasmettere realmente i segnali fisici corrispondenti ai *bit del frame* da un nodo al successivo. I protocolli in questo strato dipendono dalla tecnologia di trasmissione e dal mezzo trasmissivo del link come ad esempio, doppino e fibre ottiche. Ethernet, ad esempio, ha molti protocolli dello strato fisico: uno per il cavo UTP, un altro per la fibra ottica, uno per il wireless e così via.

**Descriveremo i livelli della pila protocollare di Internet seguendo un approccio dall'alto verso il basso, iniziando dallo strato di applicazione e procedendo fino allo strato di collegamento.**

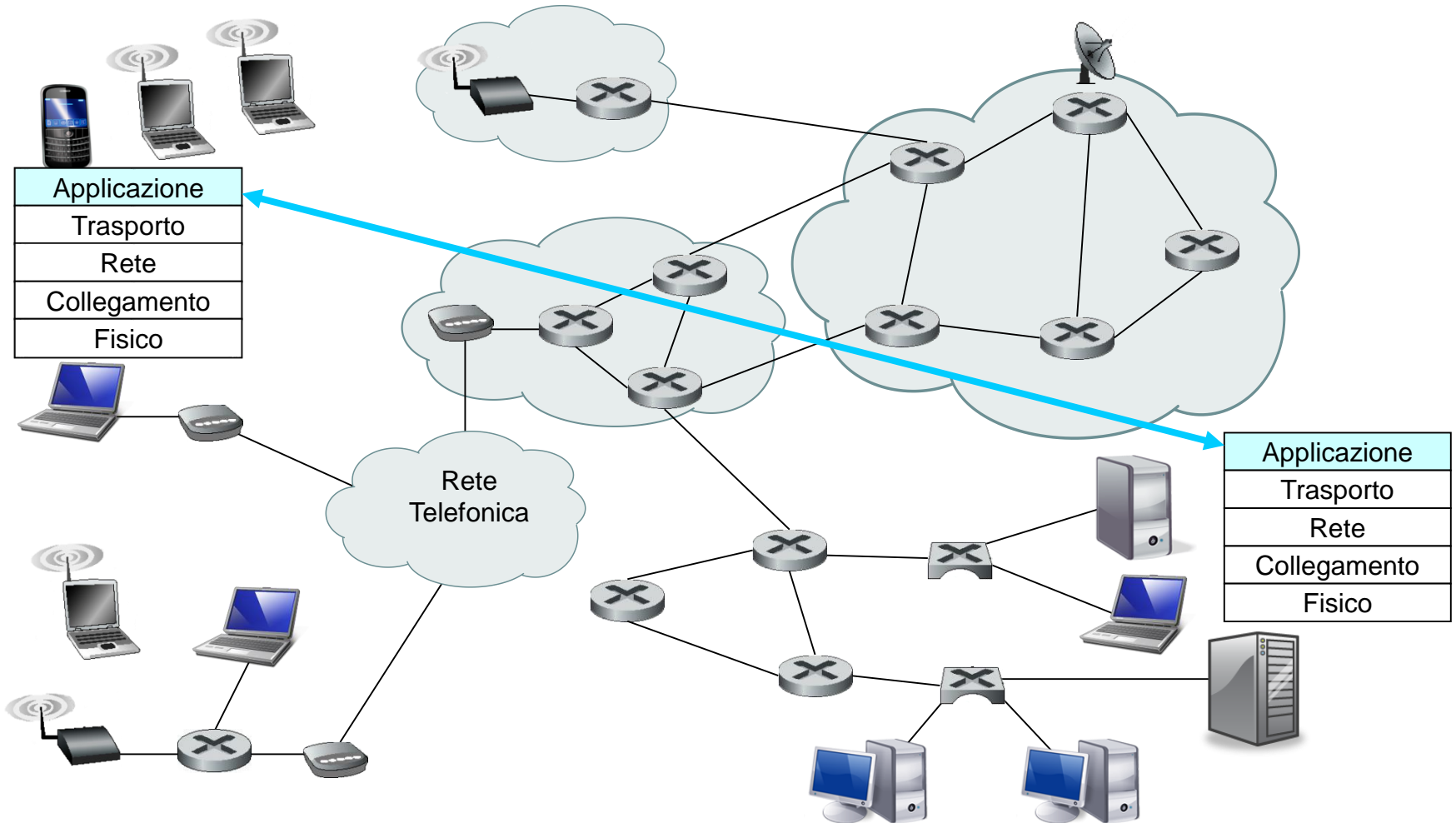


# Livello di applicazione

## Architetture e protocolli dello strato di applicazione

- L'architettura stratificata dello stack TCP/IP consente a processi su diversi host di comunicare tra loro **scambiandosi messaggi** attraverso la rete. Un processo mittente crea e invia messaggi e un processo destinatario riceve questi messaggi ed eventualmente rinvia messaggi di risposta al mittente.
- Le applicazioni di rete per comunicare devono utilizzare **protocolli** che definiscono: **il formato dei messaggi e l'ordine in cui essi sono scambiati, e la definizione delle operazioni da svolgere nella fase di trasmissione e alla ricezione dei messaggi.**
- Vari protocolli dello strato di applicazione sono di pubblico dominio e sono definiti in documenti detti **RFC** (Reference For Comment).

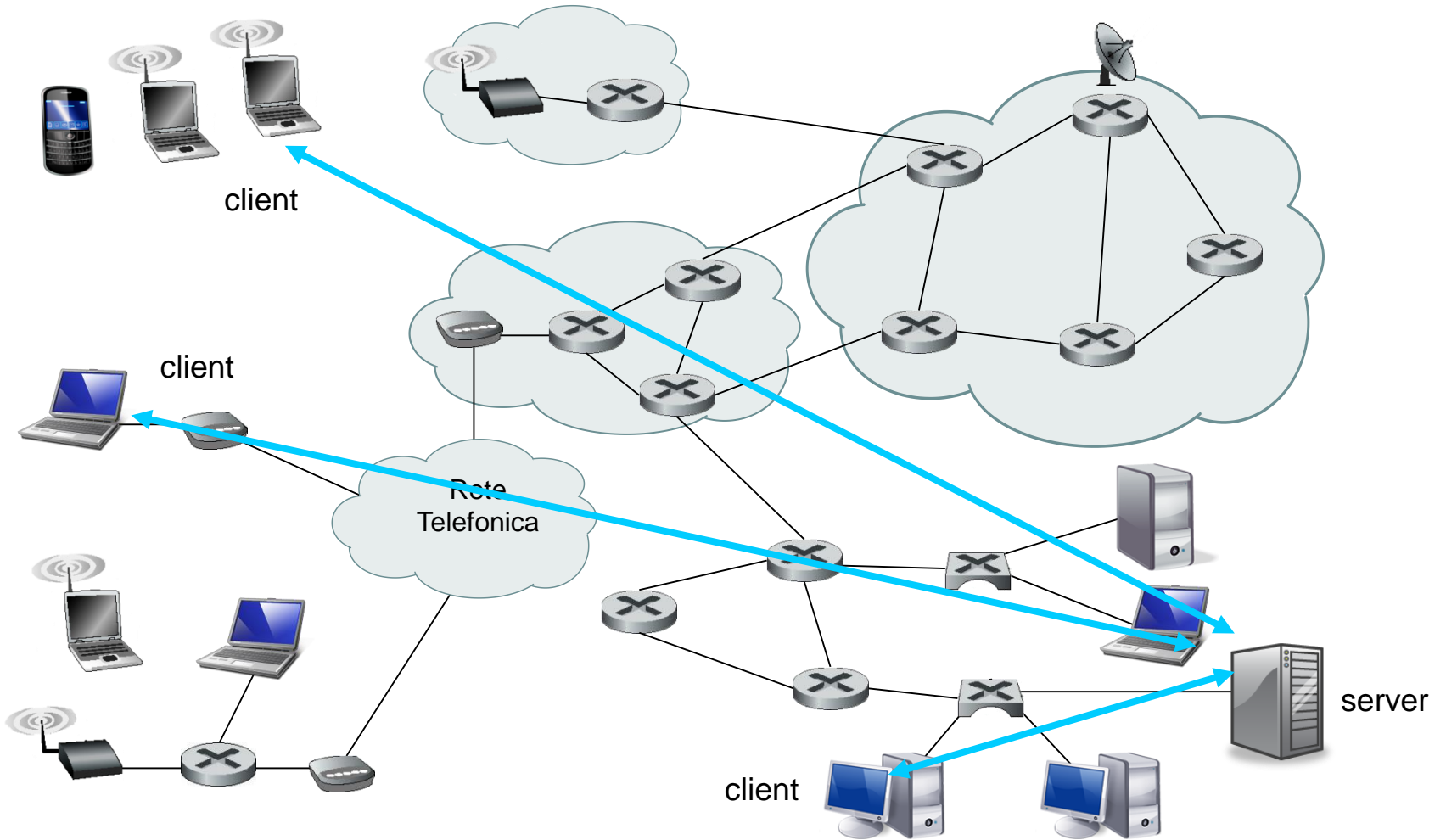
Nella figura è mostrato come i processi comunicano tra loro usando lo strato di applicazione della pila protocollare Internet.



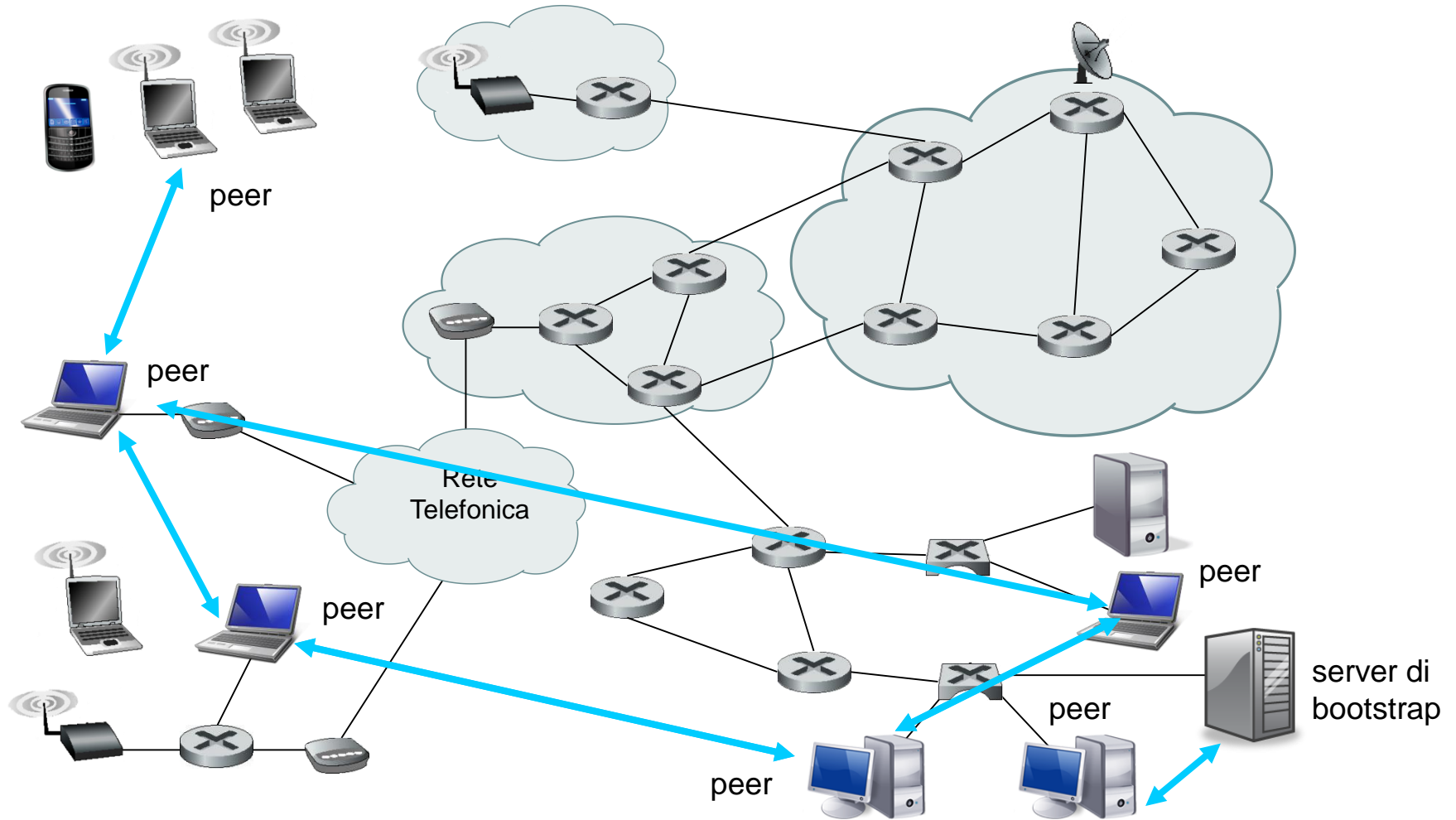
- Le principali architetture delle applicazioni di rete sono: **client/server** e **Peer To Peer** (P2P).
- Molte applicazioni di rete hanno un'architettura client/server e pertanto si realizzano in due parti, **un lato client e un lato server**. Il lato client che gira su un host comunica con il lato server che è in esecuzione su un altro host. Ad esempio, in un browser è implementato il lato client del protocollo HTTP e in un server Web è implementato il lato server del protocollo. Il browser invia messaggi per richiedere pagine web e il server risponde inviandole.
- Con l'architettura client/server i client non comunicano direttamente tra loro.
- Le applicazioni di rete possono anche implementare sia il lato client che il lato server sullo stesso host. In questo caso, per convenzione, si indica come client l'host che richiede l'instaurazione della comunicazione.

- Nell'architettura P2P, gli host connessi ad una rete sono chiamati "pari" (*peer*) e possono scambiarsi dati direttamente tra di loro.
- Ogni pari si comporta sia da client che da server, è cioè sia un distributore sia un fruitore di contenuti.
- In un sistema P2P, in ogni istante, possono essere connessi un elevato numero di pari, e ogni pari generalmente ha molti file da condividere. Se un pari **P** vuole ottenere un particolare file, allora il pari **P** deve avere funzionalità per determinare gli indirizzi IP dei pari connessi che hanno il file desiderato.
- L'architettura P2P è quindi molto più complessa della client/server.

La figura seguente mostra l'architettura client/server.



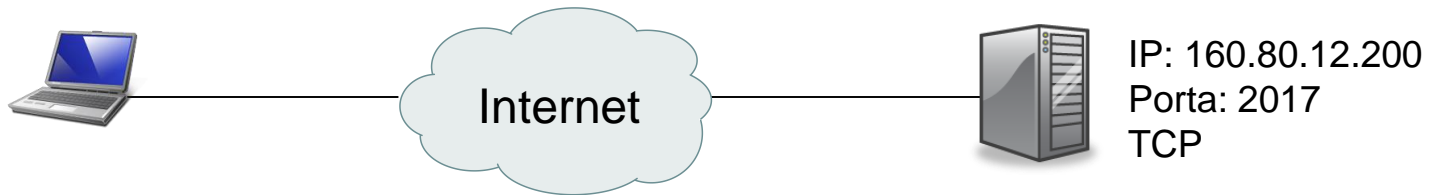
Nella figura è mostrata l'architettura P2P.



# Indirizzamento dei processi

- Per consentire a processi remoti di comunicare tra loro è necessario che essi siano identificati nella rete in modo univoco. Un processo è specificato da due informazioni:

1. **il nome o l'indirizzo dell'host su cui il processo è attivo**
2. **un identificatore del processo sull'host.**



- In Internet, un host (o meglio un suo adattatore di rete) è identificato univocamente da un indirizzo di 32 bit (IPv4) e/o di 128 bit (IPv6) detto **indirizzo (o numero) IP**.
- Il processo del host è identificato da un **numero di porta** di 16 bit che deve essere univoco per uno stesso protocollo di trasporto, nel caso siano presenti più processi di rete sullo stesso host.

- Ai protocolli dello strato di applicazione che sono stati standardizzati sono stati assegnati numeri di porta specifici. Per esempio, il protocollo HTTP, utilizzato nel Web è identificato dal numero di porta 80; il protocollo SMTP utilizzato nella posta elettronica è identificato dal numero di porta 25. Questi numeri di porta assegnati sono chiamati **numeri di porta ben conosciuti (well known ports)**.
- I numeri di porta ben conosciuti sono assegnati dallo **IANA (Internet Assigned Numbers Authority)** e sono compresi tra 0 e 1023.
- Quando si realizzano nuove applicazioni di rete, all'applicazione non devono essere assegnati i numeri di porta ben conosciuti.



Server web: porta 80

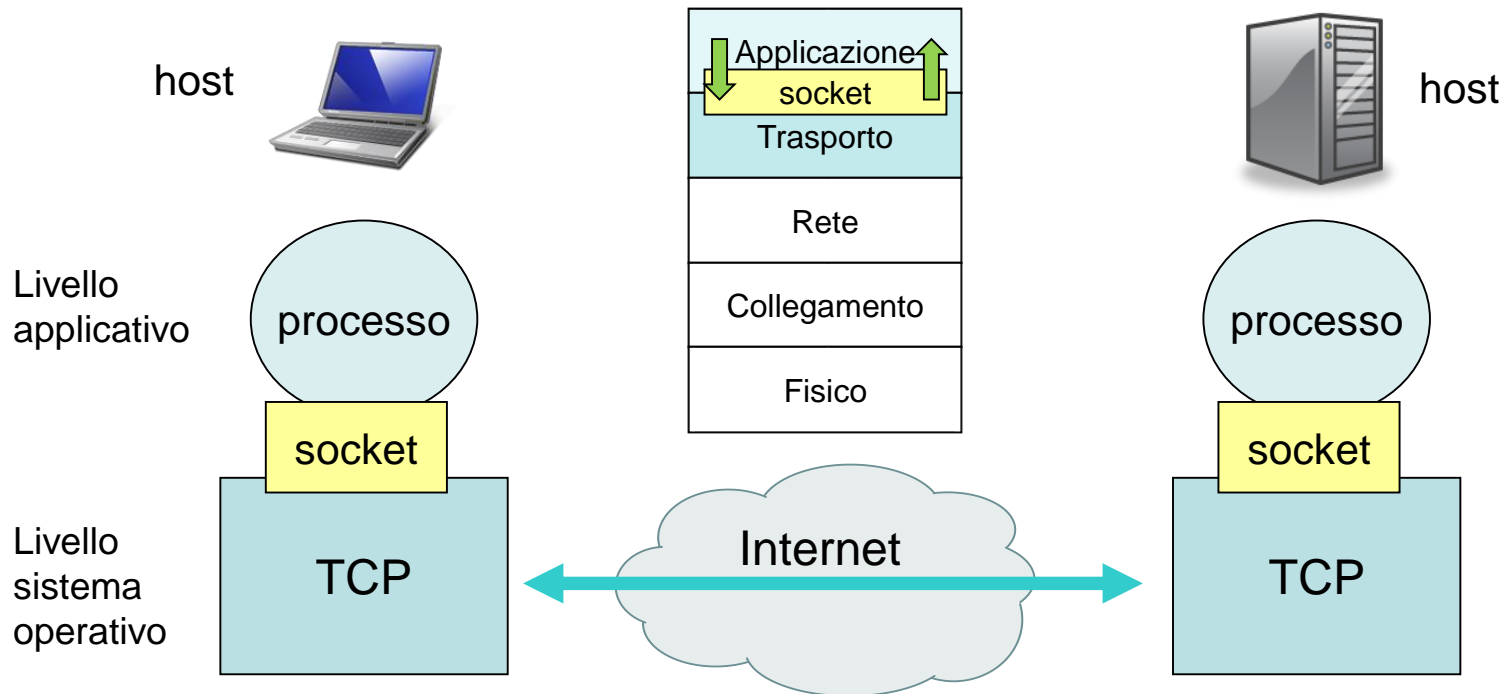


Server SMTP: porta 25



# Processi di comunicazione nella rete

- Un processo invia messaggi e riceve messaggi dalla rete attraverso un'interfaccia software con lo strato di trasporto detta **socket**. Una volta che il messaggio è arrivato all'host di destinazione, il messaggio arriva alla socket del processo ricevente. Possiamo pensare a una socket come ad una porta o ad una presa.



# Servizi di trasporto per le applicazioni

- Quando si progetta e sviluppa un'applicazione di rete è necessario stabilire quali requisiti di servizio debba possedere e quindi scegliere il protocollo di trasporto più adeguato.
- A grandi linee, possiamo classificare i requisiti di servizio di un'applicazione in quattro proprietà:
  - **Trasferimento affidabile dei dati,**
  - **larghezza di banda (throughput),**
  - **timing (temporizzazione)**
  - **sicurezza.**

# Trasferimento affidabile dei dati

- Alcune applicazioni, come posta elettronica, trasferimento dei file, web etc, richiedono un trasferimento di dati affidabile, cioè **senza perdita di dati**.
- Altre applicazioni dette ***perdite-tolleranti***, come ad esempio le applicazioni multimediali audio/video in soft real-time possono tollerare qualche perdita di dati. In queste applicazioni multimediali, la perdita dei dati, se non eccessiva, produce accettabili difetti di riproduzione audio/video.

# Larghezza di banda

- Alcune applicazioni sono dette a **larghezza di banda dipendenti** poiché devono essere in grado di trasmettere i dati a una determinata velocità. Per esempio, se un'applicazione audio multimediale per funzionare correttamente deve trasmettere dati a 32 kbit/s, è necessario che tale larghezza di banda sia disponibile, altrimenti dovrebbe generare un'eccezione in quanto la trasmissione audio a una velocità troppo bassa è insufficiente.
- Le **applicazioni elastiche** (*elastic application*), invece, possono funzionare correttamente sia con una larghezza di banda grande sia piccola. Ad esempio, posta elettronica e trasferimento di file, sono applicazioni elastiche.

# Timing

- Le applicazioni interattive in tempo reale, richiedono brevi ritardi di trasmissione dei dati. Ad esempio, la telefonia Internet richiede ritardi inferiori a circa 150 millisecondi. Se i ritardi sono compresi tra 150 e 400 millisecondi la comunicazione può essere accettabile ma ritardi superiori ai 400 millisecondi producono pause innaturali nella conversazione che diventa quindi incomprensibile.
- Per applicazioni non in tempo reale non c'è alcuna limitazione sui ritardi punto-punto, anche se ovviamente un piccolo ritardo è meglio di un grande ritardo.
- Nelle reti, ed in particolare in internet, si usa il termine **jitter** per indicare la variazione media del ritardo di ricezione dei pacchetti trasmessi, causata dalle code dei router congestionati.

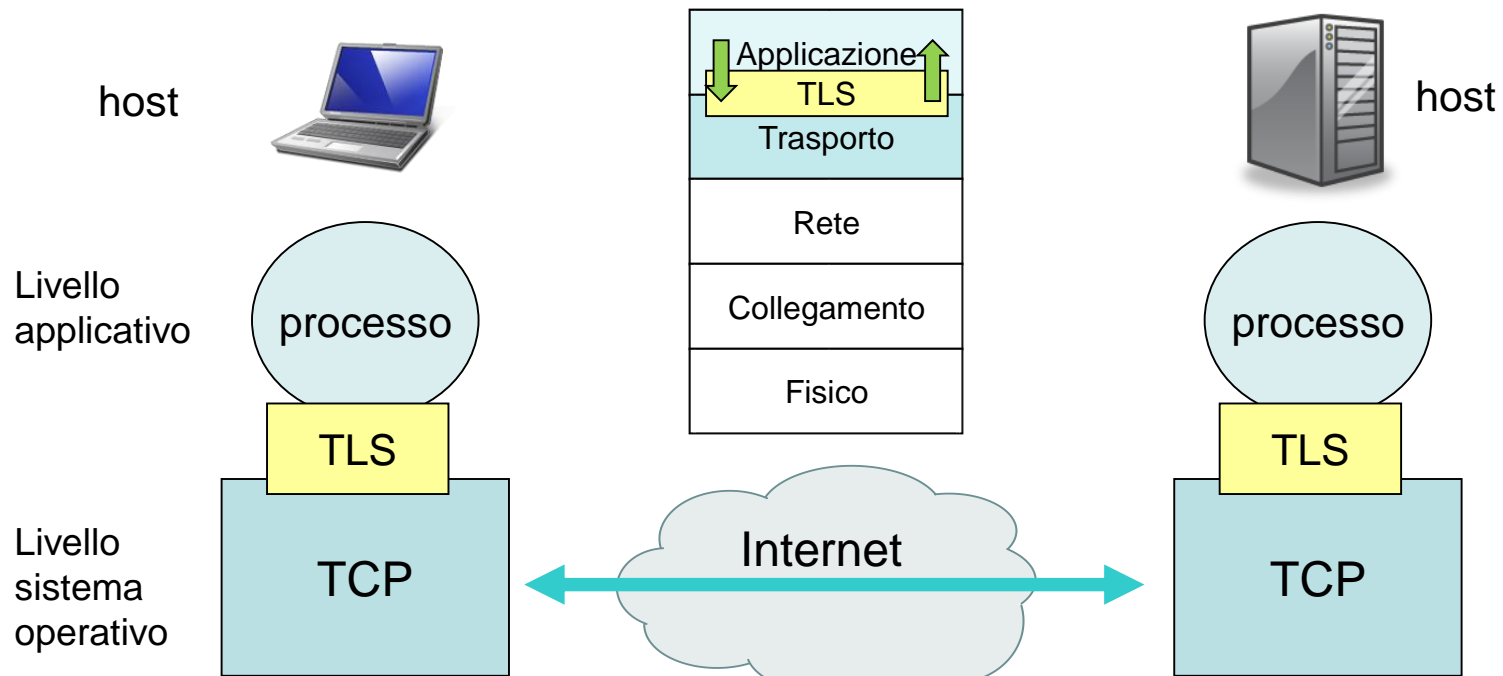
Più in generale, in elettronica e telecomunicazioni con il termine **jitter** si indica la variazione di una o più caratteristiche di un segnale come, ad esempio, la variazione di ampiezza, di frequenza, di fase.

La figura seguente riassume affidabilità, larghezza di banda, e timing richiesti da alcune applicazioni Internet.

Applicazione	Perdita dei dati	Larghezza di banda	Sensibile al tempo
Trasferimento di file	No	Elastica	No
E-mail	No	Elastica	No
Documenti Web	No	Elastica (pochi kbit/s)	No
Audio/Video in tempo reale	Tollerabile	Audio: pochi kbit/s-1 Mbit Video: 10 kb-5 Mbit	Sì, centinaia di ms
Audio/Video memorizzati	Tollerabile	Come sopra	Sì, pochi secondi
Giochi interattivi	Tollerabile	Pochi kbit/s- 10 kbit	Sì: centinaia di ms
Messaggi istantanei	No	Elastica	Sì e no

# Sicurezza

- Sia TCP che l'UDP non forniscono servizi di cifratura dei dati.
- Per ovviare a queste limitazioni sono stati sviluppati **SSL (Secure Socket Layer)** sostituito da **TLS (Transport Layer Security)** un miglioramento, a livello di sicurezza, delle socket del TCP.
- Quando un'applicazione utilizza TLS i dati vengono cifrati nel lato mittente e vengono decifrate nella destinazione.



# Servizi forniti dai protocolli di trasporto

- Lo strato di trasporto di Internet ha due protocolli (dal 2000 è presente anche **SCTP**):

**UDP (*User Datagram Protocol*)**

**TCP (*Transmission Control Protocol*).**

- Per realizzare un'applicazione di rete, è necessario decidere dapprima se scegliere di usare UDP o TCP. I due protocolli offrono un modello di servizio molto diverso alle applicazioni.



# Servizi TCP

- Il TCP fornisce alle applicazioni un servizio orientato alla connessione e un servizio di trasferimento affidabile dei dati.
- ***Servizio orientato alla connessione:*** Prima che due processi applicativi inizino a comunicare tra loro, il TCP effettua una procedura iniziale detta **handshake** ("**stretta di mano**"), terminata la quale si instaura una connessione TCP fra i **socket** dei due processi.
- La comunicazione è di tipo **full-duplex** che consente ai due processi di inviare contemporaneamente messaggi in entrambe le direzioni. La comunicazione può essere chiusa sia dal client che dal server.
- ***Servizio di trasporto affidabile:*** Il TCP garantisce ai processi che tutti i dati trasmessi arrivano a destinazione senza errori e nello stesso ordine di partenza.

- Il TCP fornisce anche un servizio di **controllo della congestione**, un servizio per regolare il traffico sulla rete. Il **controllo della congestione** del TCP modula la velocità di trasmissione dei dati di un processo (client o server) quando la rete è congestionata.
- Con il TCP la velocità di trasmissione è regolata dal servizio di controllo della congestione, che riduce la velocità media di trasmissione del mittente quando la rete è congestionata. Pertanto, il TCP non garantisce una velocità minima di trasmissione dei dati e quindi non garantisce anche un ritardo minimo.

# Servizi UDP

- L'UDP è un protocollo di trasporto semplice.
- L'UDP è un protocollo *senza connessione*, quindi non c'è la fase di handshake prima che i due processi inizino a comunicare.
- L'UDP fornisce un servizio di trasferimento dati non affidabile. Pertanto, i messaggi che arrivano alla socket ricevente possono non arrivare in ordine o non arrivare per niente.
- L'UDP non implementa il servizio di controllo della congestione, così un processo può inviare dati nella socket fino alla massima velocità consentita.
- L'UDP è più adeguato per applicazioni soft real-time le quali generalmente possono tollerare qualche perdita di dati, ma richiedono che la velocità di trasmissione non scenda sotto una determinata soglia.
- Come il TCP, l'UDP non dà garanzie sul ritardo.

<b>Applicazioni</b>	<b>Protocollo dello strato di applicazione</b>	<b>Protocollo di trasporto sottostante</b>
Posta elettronica	SMTP (RFC 821)	TCP
Accesso a terminale remoto	Telnet (RFC 854)	TCP
Web	HTTP (RFC 2616)	TCP
Trasferimento di file	FTP (RFC 959)	TCP
File server remoto	NFS (McKusik 1996)	UDP o TCP
Streaming multimediale	Spesso proprietario (per esempio, Real Networks)	UDP o TCP
Telefonia Internet	Spesso proprietario (per esempio, Dialpad)	Tipicamente UDP