

## Indice

I protocolli principali .....	2
Architetture di rete. ....	3
Architettura Client-Server. ....	3
Architettura Peer to Peer (P2P). ....	3
Il World Wide Web (WWW). ....	4
Architettura del Web. ....	4
Il protocollo HyperText Transfer Protocol (HTTP). ....	5
Formato dei messaggi http. ....	6
Metodi GET e POST. ....	6
Posta elettronica .....	7
Trasferimento di File: Ftp.....	12
Domain Name System (DNS). ....	19


## LE APPLICAZIONI DI RETE

Il livello di applicazione del modello ISO/OSI (livello 7) e del modello TCP/IP, come sappiamo implementa le applicazioni ed i servizi di rete che vengono utilizzati dagli utenti. Questi servizi sono corredati, come tutto ciò che è gestito nella rete, da protocolli che ne consentono l'utilizzo ai cosiddetti sistemi aperti, ossia ai sistemi elaborativi che, pur avendo caratteristiche diverse, grazie a questi protocolli, riescono a comunicare tra di loro.

### I protocolli principali

I principali protocolli del livello di applicazione sono:

<b>SNMP</b>	<b>Simple Network Management Protocol</b> è un protocollo di monitoraggio, per la raccolta e l'organizzazione di informazioni sui dispositivi gestiti su reti IP e per la modifica di tali informazioni per cambiare il comportamento del dispositivo. I dispositivi che in genere supportano SNMP includono modem via cavo, router, switch, server, workstation, stampanti ecc..
<b>SMTP</b>	<b>Simple Mail Transfert Protocol</b> è un protocollo utilizzato dai server di posta elettronica per la trasmissione di email. I client mail a livello utente utilizzano SMTP solo per inviare il messaggio al server mail, il quale si occupa dell'invio del messaggio stesso (porta 25)
<b>POP3</b>	<b>Post Office Protocol</b> è un protocollo di livello applicativo di tipo client-server che ha il compito di permettere, mediante autenticazione, l'accesso da parte del client ad un account di posta elettronica presente su di un host server e scaricare le e-mail dell'account stesso.
<b>FTP</b>	<b>File Transfer Protocol</b> è un protocollo utilizzato per il trasferimento di dati basato su un sistema client-server. Consente di caricare, spostare e scaricare file all'interno di un sistema di directory in hosting. L'FTP usa il Transmission Control Protocol (TCP) per il trasferimento dati, e richiede autenticazione del client attraverso nome utente e password. (generalmente usa porta 21)
<b>HTTP</b>	<b>HyperText Transfer Protocol</b> è un protocollo utilizzato dal WWW per il trasferimento di dati ipertestuali (pagine web), utilizza un'architettura client-server. (porta 80)
<b>DNS</b>	<b>Domain Name System</b> è un protocollo che associa un nome ad un indirizzo IP. E' una sorta di server (locale) o di rete internet che raccoglie e associa i nomi degli indirizzi web agli indirizzi IP corrispondenti. Quando un utente inserisce una URL nel proprio browser, il DNS si mette al lavoro per connettere quella URL all'indirizzo IP del server. Generalmente offre e gestisce spazio web (porta 53).
<b>SSH</b>	<b>Secure Shell</b> è un protocollo che permette di stabilire una sessione remota cifrata tramite interfaccia a riga di comando con un altro host di una rete informatica. È il protocollo che ha sostituito l'analogo, ma insicuro, Telnet.
<b>TELNET</b>	è un protocollo che permette di stabilire una sessione di login remota tramite interfaccia a riga di comando con un altro host di una rete informatica.

 Un server può offrire più servizi agli utenti della rete (ad esempio: email, sito web, DNS,...). Quando questo riceve la richiesta di uno di questi servizi da parte di un client, se la richiesta inviata dal client contiene il solo indirizzo IP del server, questi ovviamente, non saprebbe quali dei servizi disponibili deve fornire al richiedente. Per risolvere questo problema il client, oltre l'indirizzo del server deve aggiungere un'ulteriore informazione: la porta logica. La porta logica identifica il tipo di servizio desiderato. L'accoppiata indirizzo IP + porta logica prende il nome di socket (socket = presa).

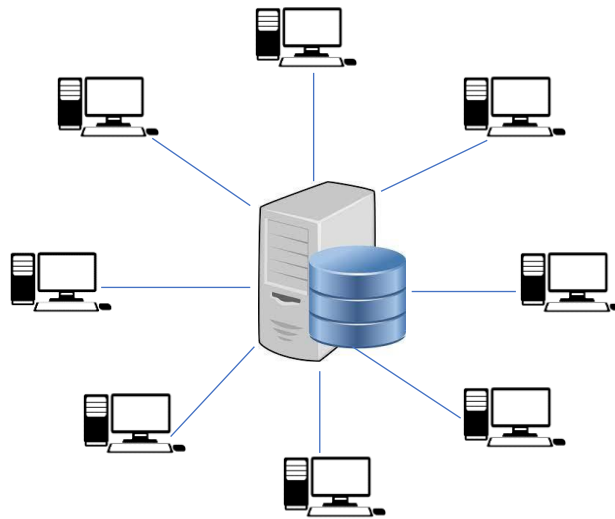
## Architetture di rete.

Le principali architetture di rete sono:

- Client – server
- Peer to peer (P2P)
- Ibride (client-server + P2P).

### Architettura Client-Server.

In questa architettura deve essere presente almeno un server che offre un servizio (almeno uno) e che resta in attesa che uno o più client si connettano e glielo richiedano.



### Architettura Peer to Peer (P2P).

In queste architetture coppie di host (peer – pari grado) si scambiano informazioni assumendo il ruolo di 'server' o di 'client' nel momento in cui devono fornire o ricevere informazioni dal peer collegato. Le architetture P2P necessitano di software appositi che devono essere installati su ogni host della rete (eMule, Gnutella, bBitTorrent,...). In questa architettura, un host può essere attivo (collegato alla rete) o meno. I problemi principali per chi ricerca delle informazioni nell'architettura P2P, sono quindi:

- ✓ chi possiede le informazioni
- ✓ chi le possiede è attivo nel momento in cui viene effettuata la ricerca?

Il primo problema viene risolto con la ridondanza delle informazioni: la stessa informazione è presente su più host della rete P2P.

Le architetture P2P si suddividono in tre tipologie:

**Decentralizzate (pure):** ciascun nodo della rete ha la stessa importanza gerarchica e la medesima funzione. In questo caso se un nodo necessita di una risorsa deve chiedere a tutti i nodi chi la possiede.

**Peer-to-Peer con Discovery Server:** questo modello Peer-to-Peer è dotato di 1 server centralizzato (Discovery). Quando un peer entra nella rete, comunica la sua esistenza al server centrale che, in risposta, restituisce l'elenco completo dei nodi già connessi a quella rete, questo gli permetterà di aprire le comunicazioni con tutti gli altri. Un nodo che necessita di una risorsa chiede chi la possiede ai solo ai nodi connessi.

**Peer-to-Peer con Lookup Server:** è un'evoluzione dell'architettura Discovery. I nodi comunicano, periodicamente, a tutti gli altri quali sono i propri contenuti. Così, quando un nodo dovrà effettuare una ricerca, invierà la richiesta al Discovery server che, essendo a conoscenza delle informazioni contenute da ciascun peer, lo indirizzerà in maniera più efficiente.

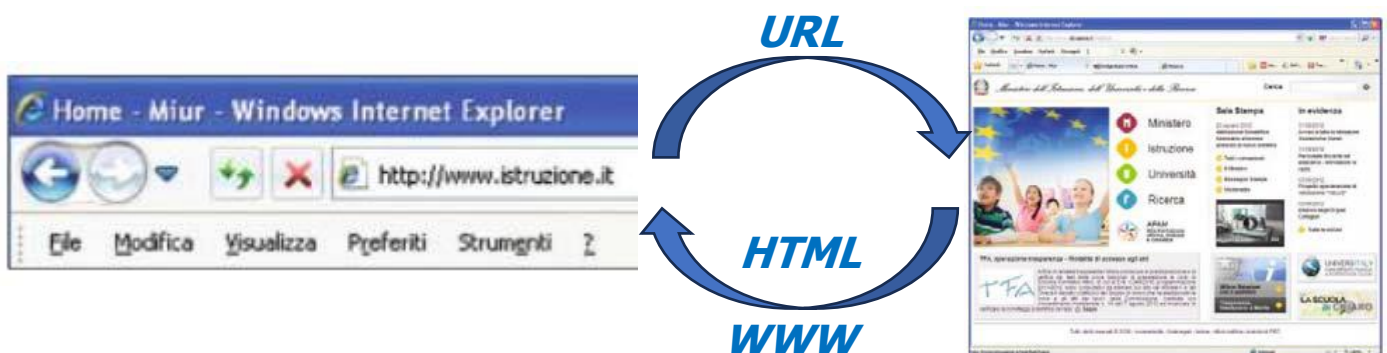
## Il World Wide Web (WWW).

Il WWW (chiamato semplicemente Web) è un sistema ipermediale che ci consente di accedere a delle informazioni sotto forma di ipertesti ossia di documenti che non contengono solo informazioni di tipo testuale ma anche immagini, animazioni ed audio.

Per utilizzare il Web abbiamo bisogno di programmi appositi, detti browser (sfogliatori), che trasformano le informazioni scritte con un apposito linguaggio (HTML) in informazioni intelleggibili agli utenti (testo, immagini, filmati, suoni...). I browser più conosciuti sono: **Chrome, Microsoft Edge, Mozilla, Opera...**

## Architettura del Web.

All'interno della rete ciascuna risorsa (pagina web) è identificata univocamente dall'URL (Universal Resource Locator), ossia l'indirizzo della risorsa nel web; tramite questo indirizzo si può richiedere l'invio della risorsa (della pagina web). Ad esempio, supponiamo che ci si voglia collegare ad un certa pagina web, si deve digitare nella barra in alto del **browser** l'URL della pagina, ad esempio l'indirizzo **http://www.istruzione.it:**



L'URL contiene tutte le informazioni per localizzare il file (pagina web) così che la richiesta arrivi al server che contiene il file il quale risponde inviando al client la pagina richiesta

(scritta in HTML). Il browser trasformerà la pagina HTML ricevuta in un testo ipermediale a noi comprensibile.

In definitiva l'URL è formato da:

- il protocollo per la connessione (http nel ns caso)
- il nome simbolico (dominio) o l'indirizzo IP del server
- il pathname del file sul server.

Il formato dell'URL in definitiva è:

**protocollo://sottodominio"n".sottodominio"n-1"...sottodominio"1".dominio/directory/file**

ad esempio:

**http://www.cs.provincia.it/imu/2020/aliquote.html**

**http** è il protocollo usato

**cs** è il sottodominio (nel nostro esempio ce n'è uno solo)

**provincia.it** è il dominio (ossia il server ove è collocata la risorsa)

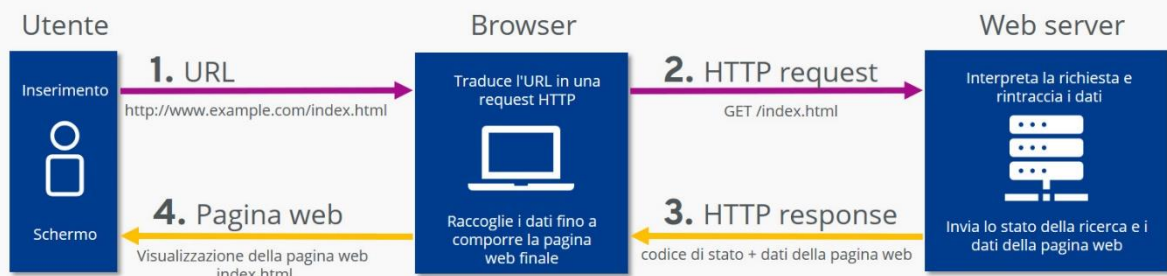
**imu/2020** la directory e sottodirectory all'interno del server ove è collocato...

**aliquote.html** il file restituito al client.

## Il protocollo HyperText Transfer Protocol (HTTP).

Il trasferimento dei dati ipertestuali nel web, tra un client ed un server (architettura client-server), si basa sul *protocollo applicativo http*, all'interno del protocollo TCP/IP, utilizzando gli indirizzi IP dei due dispositivi. I messaggi che si scambiano client e server, tramite il protocollo **http**, possono essere di due categorie: **request** (richiesta) e **response** (risposta) e sono un insieme di righe di caratteri ASCII terminati da CRLF (*Carriage Return Line Feed*)

### Processo di comunicazione secondo HTTP



1. L'utente inserisce nella barra degli indirizzi del browser l'URL della risorsa cercata.

- Il browser estrae il dominio dall'url e apre una connessione TCP sulla porta 80 tra client e server creando il socket di comunicazione
2. Il browser, attraverso il socket, invia al server con una **http request**, la richiesta della risorsa cercata
  3. Il server, ricevuta la richiesta, tramite una **http response** invia al browser del client, prima una comunicazione se la risorsa cercata è stata trovata e, successivamente (se è stata trovata), una serie di **http response**, contenenti la risorsa.
  4. Il browser visualizza la risorsa e chiude la comunicazione.

## Formato dei messaggi http.

Un messaggio http è formato da tre parti: **start-line**, **header**, **body**

Nella **start-line** di richiesta (browser) sono presenti un metodo (GET/POST i più usati), la risorsa richiesta (URL), la versione http

Nella **start-line** di risposta (Server), sono presenti la versione dell'http ed un codice che indica lo stato della richiesta (se è in fase di elaborazione, se la risorsa è stata trovata o meno, ecc....).

Nell'**header** sono presenti dei *meta-dati* che si scambiano client e server quali: il tipo del documento inviato dal server, il tipo di browser che ha fatto la richiesta, la data di invio e/o dell'ultima modifica del documento... questi meta-dati non sono visualizzati dal browser.

Nel **body** il server inserisce il documento (parte di esso: il server invia tanti messaggi http response contenenti vari 'pezzi' del documento finché il documento non è stato inviato completamente) richiesto dal browser.

## Metodi GET e POST.

Nella comunicazione tra Client e Server, il client ha spesso la necessità di inviare dei parametri (ossia dei dati) al server. Come abbiamo visto, nella *start-line* del http si possono definire dei metodi che indicano la modalità di trasmissione di questi dati: **GET** e **POST**. Nel metodo **GET** i parametri vengono inseriti nel protocollo http dopo l'URL, in questo modo sono visibili nella barra degli indirizzi quando vengono trasmessi. Bè se questi parametri non sono sensibili, poco male (ad esempio: mi collego ad un sito di articoli sportivi e voglio sapere se sono presenti articoli della marca [Xwztf](#) inserendo [Xwztf](#) nel campo **cerca marche**: con il metodo GET questo parametro viene trasmesso in chiaro nel http al server); se invece sono 'sensibili' (UserID e password, coordinate bancarie...) farli viaggiare in 'chiaro' nella rete... non mi sembra una buona idea. Con il metodo **POST**, i parametri vengono inseriti nel BODY del protocollo http e risultano invisibili. Se su usa, inoltre, il protocollo Https (che vedremo in seguito) i dati vengono cifrati: con la cifratura si aumenta la sicurezza durante la trasmissione.

## Posta elettronica

Il successo di INTERNET è legato anche alla posta elettronica **email** (electronic mail), usata per inviare messaggi ad amici, usata dalle aziende per normali transazioni commerciali o per messaggi commerciali **spam**, dato che il loro costo è praticamente nullo.

Gli indirizzi email hanno la forma:

**nomeutente@dominio (@ -> AT (presso))**

dove **nomeutente** identifica l'interlocutore (sia mittente che destinatario) univoco per il dominio ed è scelto dall'amministratore del **server** e **dominio** è rappresentato dal nome del provider e nel caso seguente dalla nazionalità oppure da un IP address:

**mariorossi@tiscali.it**

Può contenere qualsiasi carattere anche **\_** e **.** **escluse le accentate.**

L'indirizzo di posta non è associato alla persona bensì ad una casella di posta elettronica, ogni utente può avere più caselle di posta e più nomi utenti (alias) possono essere associati alla stessa casella di posta.

La posta elettronica si differenzia in funzione della modalità di accesso che può essere di due tipi:

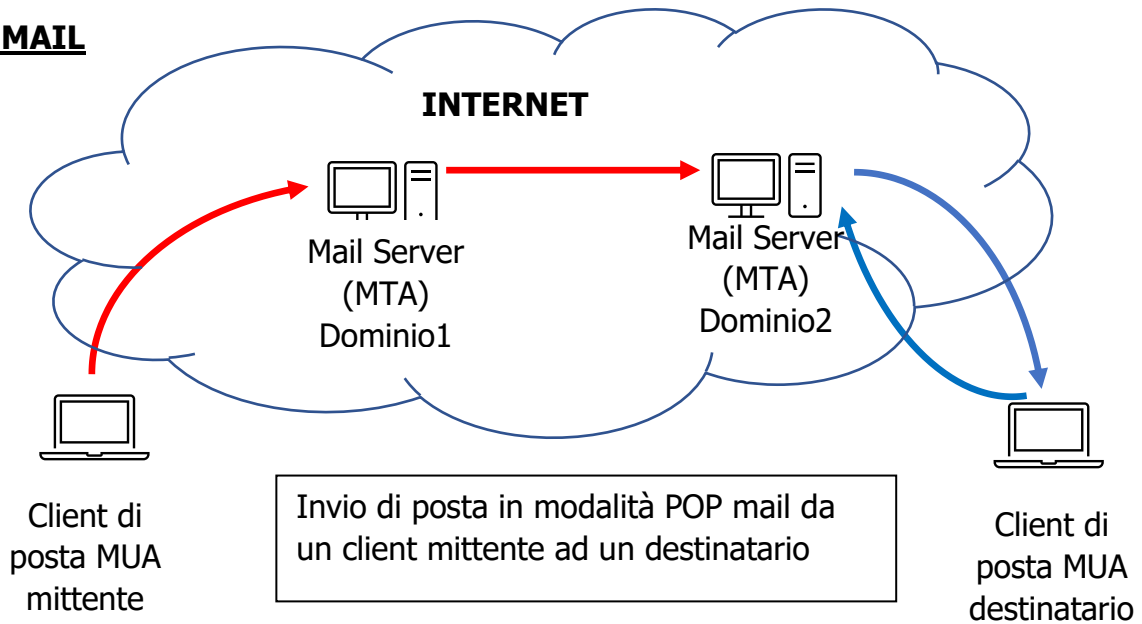
- **POP mail**
- **Web mail**

La **POP mail** permette di **leggere ed inviare i messaggi dal computer** sul quale è installato il programma di posta elettronica, che prende il nome di *client di posta*, come **Eudora, Mozilla Thunderbird, Outlook Express, IncrediMail.**

Il **client di posta** è, quindi, un programma che consente di inviare/ricevere posta elettronica direttamente sul computer sul quale è installato e configurato per la trasmissione/invio di una email; è dotato di un'interfaccia grafica **MUA** (Mail User Agent) che ci consente di comporre/leggere i messaggi di posta elettronica da e verso il server di posta elettronica **ISP** al quale siamo registrati. (**ISP Internet Service Provider**).

La **Web mail** è un'applicazione web a cui accediamo tramite il browser: tale applicazione ci consente di inviare/ricevere mail non dal/sul nostro computer ma da un server di posta del provider cui siamo abbonati e su cui conserviamo/riceviamo/inviemo le nostre mail ed a cui accediamo tramite l'inserimento dell'indirizzo di posta elettronica ed una password. MUA in questo caso è un servizio di webmail cui accediamo sempre tramite il browser.

**POP MAIL**

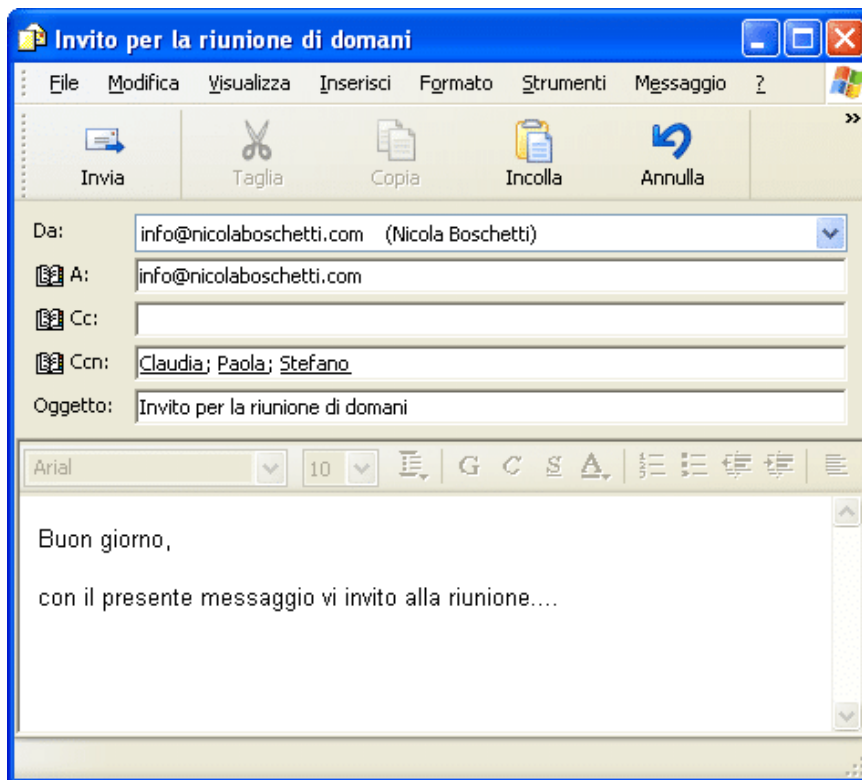


Protocollo SMTP ———

Protocollo POP3/IMAP4 ———

Vediamo come avviene l'invio/ricezione di un messaggio tramite **POP Mail**. Sul nostro PC è installato un programma **client di posta**, che, una volta avviato, ci presenta una maschera dalla quale possiamo scegliere l'operazione da compiere: limitiamoci alle sole operazioni di invio e ricezione di una email.

**Invio SMTP (Simple Mail Transfer Protocol)**





Tramite la maschera in figura (MUA) il programma costruisce una stringa costituita da un **header** e da un **body** con una linea bianca che li separa.

**Header** contiene le informazioni per il trasporto:

from (da): <mittente>

to (a): <lista destinatari>

CC (Carbon Copy): <lista destinatari per conoscenza>

Bcc (CCn): <lista destinatari per conoscenza nascosti>

Date: <data spedizione>

Reply to (Rispondi): <indirizzo diverso dal mittente>

Subject (Oggetto): <titolo oggetto dei messaggi>

**Body (Corpo del messaggio)**: messaggio vero e proprio, formato testo cioè ASCII a 7 bit 128 caratteri.

Con il **MIME (Multipurpose Internet Mail Extensions – applicativo che 'potenzia' SMTP per l'invio di dati non solo ASCII)** è possibile inserire anche immagini, segnali audio, video.

L'**invio del messaggio** da parte del **client** viene fatta con il protocollo **SMTP**: il messaggio arriva quindi al Mail Server del ISP che lo ritrasmette sulla rete, da mail server a mail server sino ad arrivare al mail server dell'ISP del destinatario.

Quello che noi indichiamo come **mail server** tecnicamente prende il nome di **MTA** (Mail Transfer Agent) ed offre i seguenti servizi:

- **server SMTP** (porta 25) gestisce la spedizione e la ricezione dei messaggi tra i **server SMTP**;
- **server POP3** (porta 110) gestisce la spedizione dei messaggi al **client**;
- **server IMAP4** (porta 143) permette la gestione dei messaggi sul **server** dal **client**.

Il MTA è un'applicazione in esecuzione su un server dell'**ISP** dell'utente.

I **mail server** hanno al loro interno la mailbox "fermo posta" contenente i messaggi in entrata che non sono stati letti e scaricati sui client nel caso di modalità **POP mail**. In questa modalità, quando viene avviato il client di posta questo si connette al Mail Server e se ci sono messaggi fermi nella mailbox, tramite il protocollo POP3, vengono scaricati sul PC e quindi possono essere letti tramite il MUA del client di posta installato sul PC.

### **Prelievo posta: Post Office Protocol (POP3)**

Esaminiamo come un utente può accedere alla sua casella per leggere i propri messaggi e scaricarli sul proprio PC.

Usa il protocollo **POP (POP3)** che permette al client di posta di accedere al server e di trasferire i messaggi dalla propria mailbox ("fermoposta") al proprio PC, con il programma client di posta installato sul PC stesso.

Con POP3 il client effettua il log-on al mail server del destinatario e scarica tutti i messaggi ad esso pervenuti dai mail server mittenti dall'ultima connessione stabilita con allegati.

Il protocollo **POP** è un comune protocollo client/server dove lo **user agent (MUA)** ha il ruolo di **client POP** ed il **mail server** ha il ruolo di **server POP**.

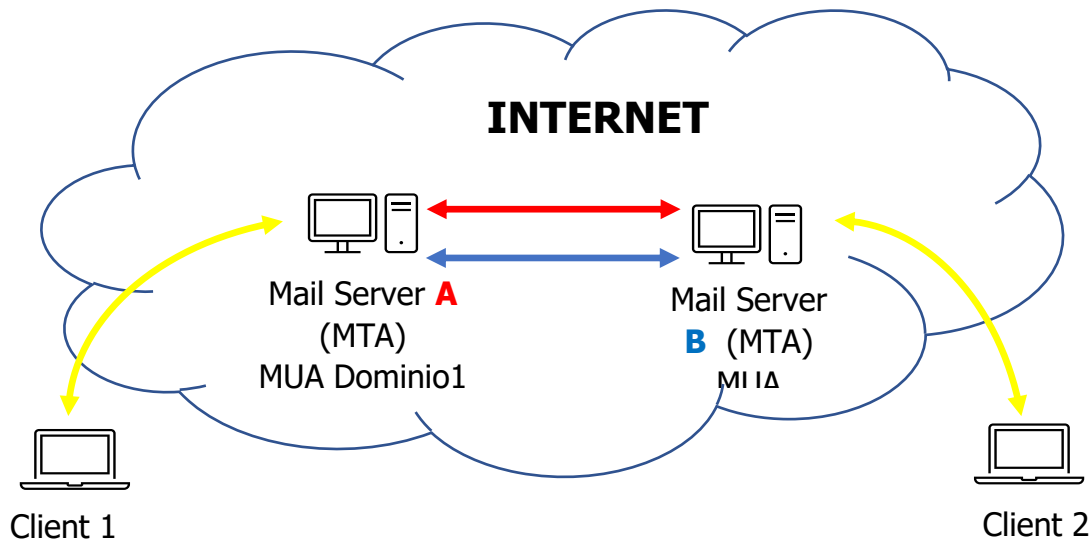
La conversazione POP3 utilizza la porta 100 e avviene in tre fasi:

- **Autorizzazione**: il client si identifica e il server verifica che abbia le autorizzazioni

- **Transazione:** fase di scarico e quit del client
- **Aggiornamento:** il server elimina tutti i messaggi scaricati e chiude la connessione

## **WEB MAIL**

Nel caso di **Web mail** al **MTA** vengono aggiunte tutte le funzionalità del **MUA** dato che l'utente effettua operazioni direttamente sul server.



Protocollo HTTP —————

Protocollo SMTP —————

Protocollo IMAP4/POP3 —————

In questo caso, come è intuibile dalla figura, i client si collegano a delle applicazioni web di posta elettronica sulle Mail Server degli ISP tramite http, ossia tramite il browser.

Tramite queste applicazioni web ci interfacciamo con il MUA presente sul server e inviamo e riceviamo le mail (che non vengono scaricate sul nostro PC).

Ovvio che gli esempi sopra descritti sono limite, nel senso che i client o sono entrambi POP Mail o entrambi gestiscono la loro posta elettronica tramite Web Mail: in realtà i casi sono solitamente misti, anche se ormai i client di posta stanno cedendo il posto alle Web Mail.

## **Protocollo IMAP**

Un protocollo alternativo al **POP3** è **Internet Message Access Protocol**

Il principale vantaggio è quello di consentire alcune manipolazioni sulla posta in entrata ancora prima di prelevarla dal server.

Il protocollo IMAP, che opera sulla porta 143, consente di:

- rinominare la propria casella elettronica;
- cancellare singoli messaggi senza prelevarli;
- leggere le intestazioni dei messaggi senza doverli prelevare interamente;
- prelevare porzioni dei messaggi;
- effettuare l'accesso simultaneo alla stessa casella di posta.

### **IMAP4 è la versione più recente e come POP3 si occupa della posta in arrivo**

È candidata a divenire lo standard di riferimento su Internet e permette di archiviare i messaggi in cartelle direttamente sul server, di accedere contemporaneamente a più mail server e di condividere la mailbox con altri mail server, offre una migliore integrazione con la tecnologia MIME.

### **Trasferimento SMTP**

Il protocollo **SMTP** usa **TCP porta 25** per consegnare in modo affidabile i messaggi dal client al server.

È un protocollo *furbo*: infatti verifica l'esistenza del destinatario prima di inviare il messaggio, in caso di destinatari multipli cerca di evitare di inviare più volte.

L'invio è composto da tre fasi (vedi Figura):

1. Il programma di posta elettronica usato dall'utente invia il messaggio al server A usando il protocollo SMTP.
2. Il server trasferisce il messaggio al server B del destinatario utilizzando lo stesso protocollo:

**A sulla base dell'indirizzo email identifica B e apre una connessione**

**B identifica A e accetta la connessione**

**A comunica username del destinatario**

**B verifica la validità dell'indirizzo e autorizza la trasmissione del messaggio**

**A invia e chiude la connessione**

**B memorizza il messaggio e attende che il destinatario si colleghi e ritiri il messaggio con apposito protocollo (POP3 O IMAP)**

3. Il destinatario preleva il messaggio dal proprio server



*N.B. i punti 1. e 3. vengono eseguiti in caso di POP Mail*

## Trasferimento di File: Ftp

Il protocollo utilizzato a livello applicativo per trasferire i file è il **File Transfer Protocol**, chiamato **FTP**.

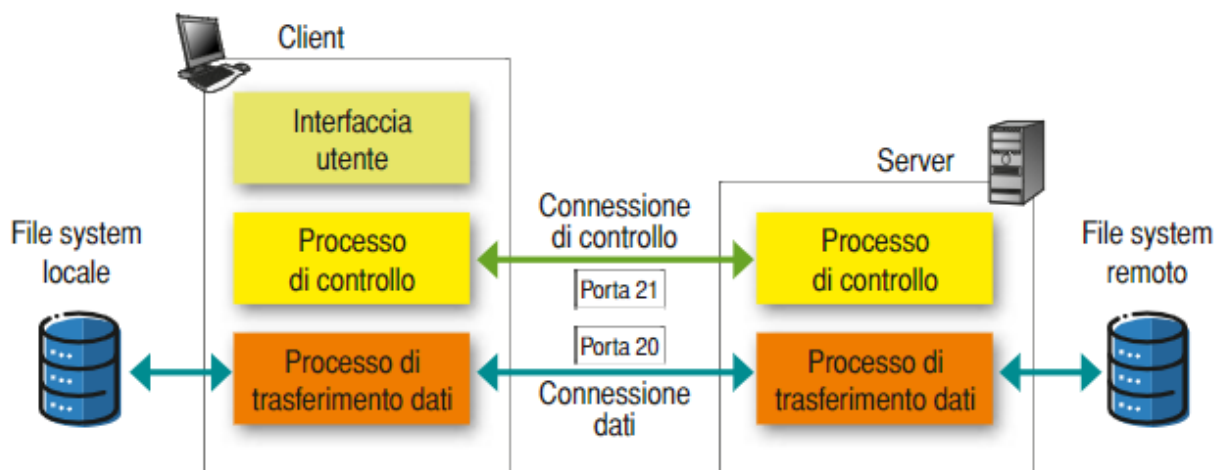
Gli obiettivi principali dell'**FTP** sono:

- Promuovere la condivisione di file
- Incoraggiare l'uso di computer remoti
- Risolvere in maniera trasparente incompatibilità tra differenti sistemi di deposito file;
- Trasferire dati in maniera affidabile ed efficiente

**FTP** utilizza due canali **TCP** separati che agiscono in parallelo:

**1.** una connessione di **controllo** usata per spedire le informazioni di controllo tra client e server, come user, password, comandi per la variazione della directory remota; viene chiamata **control connection** o **command channel**;

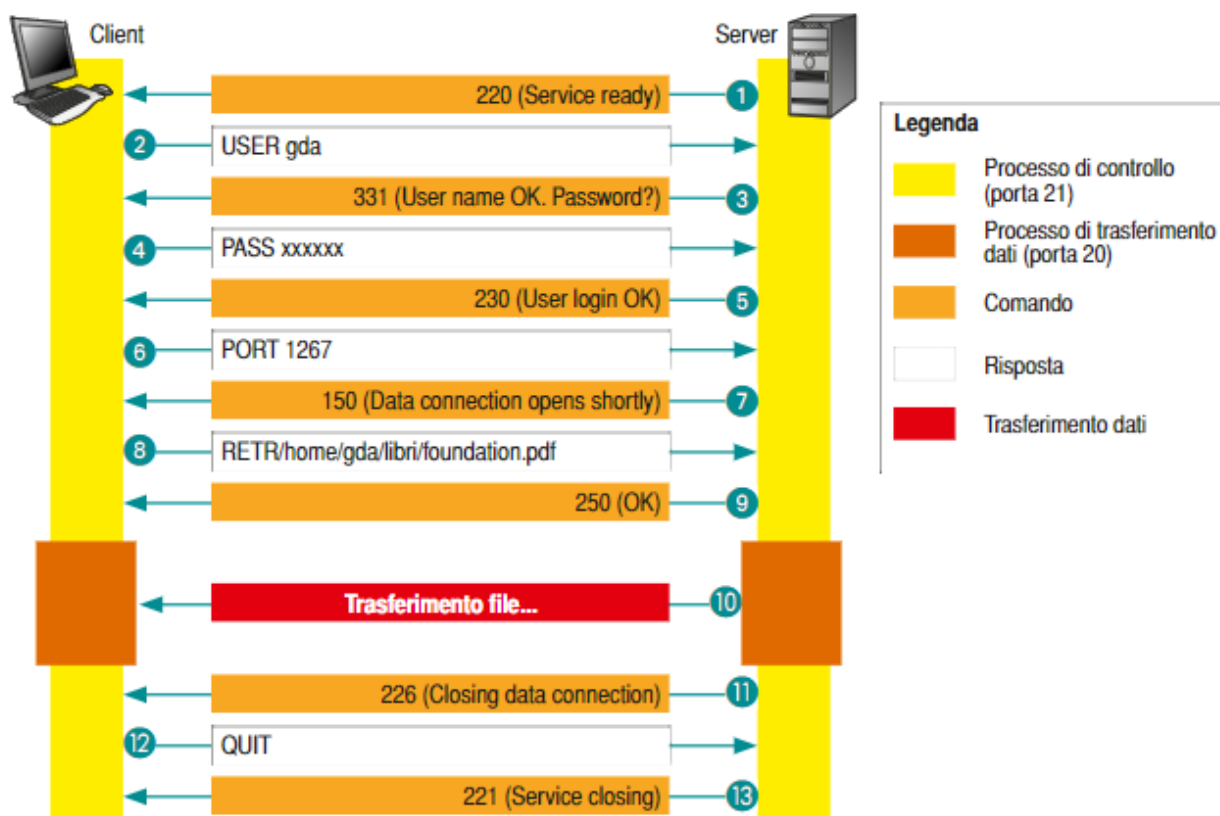
**2.** una connessione **dati** utilizzata per il trasferimento dei file; viene chiamata **data connection** o **data channel**;



### **Il server ed il client FTP**

Il protocollo **FTP** si riferisce ad un modello client/server dove la macchina **server** ha installato uno specifico programma che può essere fornito con il **SO** o installato successivamente.

Entrambe le macchine devono avere installato un software FTP: FTP client ed FTP server.



### FTP server

Esistono parecchi software in distribuzione, sia gratuiti con licenza che a pagamento:

- *Filezilla Server;*
- *Bulletproof Ftp Server;*
- *Golder Ftp Server;*
- *Globalscape Secure Ftp Server;*
- *Cerberus Ftp Server;*
- *Gene6 Ftp Server;*
- *Serv-U Ftp Server;*
- *Wftpd;*
- *Surgeftp;*

Il software FTP server mette a disposizione dei clienti molteplici opzioni per interagire con i file condivisi nel suo file system;

- download/upload di file;
- recupero (resume) di trasferimenti interrotti;
- rimozione e rinomina dei file;
- creazione di directory;
- navigazione tra directory.

L'accesso viene effettuato mediante autenticazione Utente e password a cui vengono assegnati privilegi

Se si entra come anonymous senza password si assegna solo modalità lettura.

## **FTP client**

Anche la connessione client usa un apposito software per eseguire **upload** o **download** di file di un server ad un determinato indirizzo ip.

Generalmente un FTP client ha due componenti:

- **parte di comunicazione** che implementa FTP;
- **interfaccia utente** che agevola l'utente;

I differenti programmi si differenziano dalle opzioni offerte all'utente, upload multipli, velocità, componenti grafici, interruzione di trasferimenti e ripresa senza file corrotti ecc. Le funzioni comuni sono le seguenti:

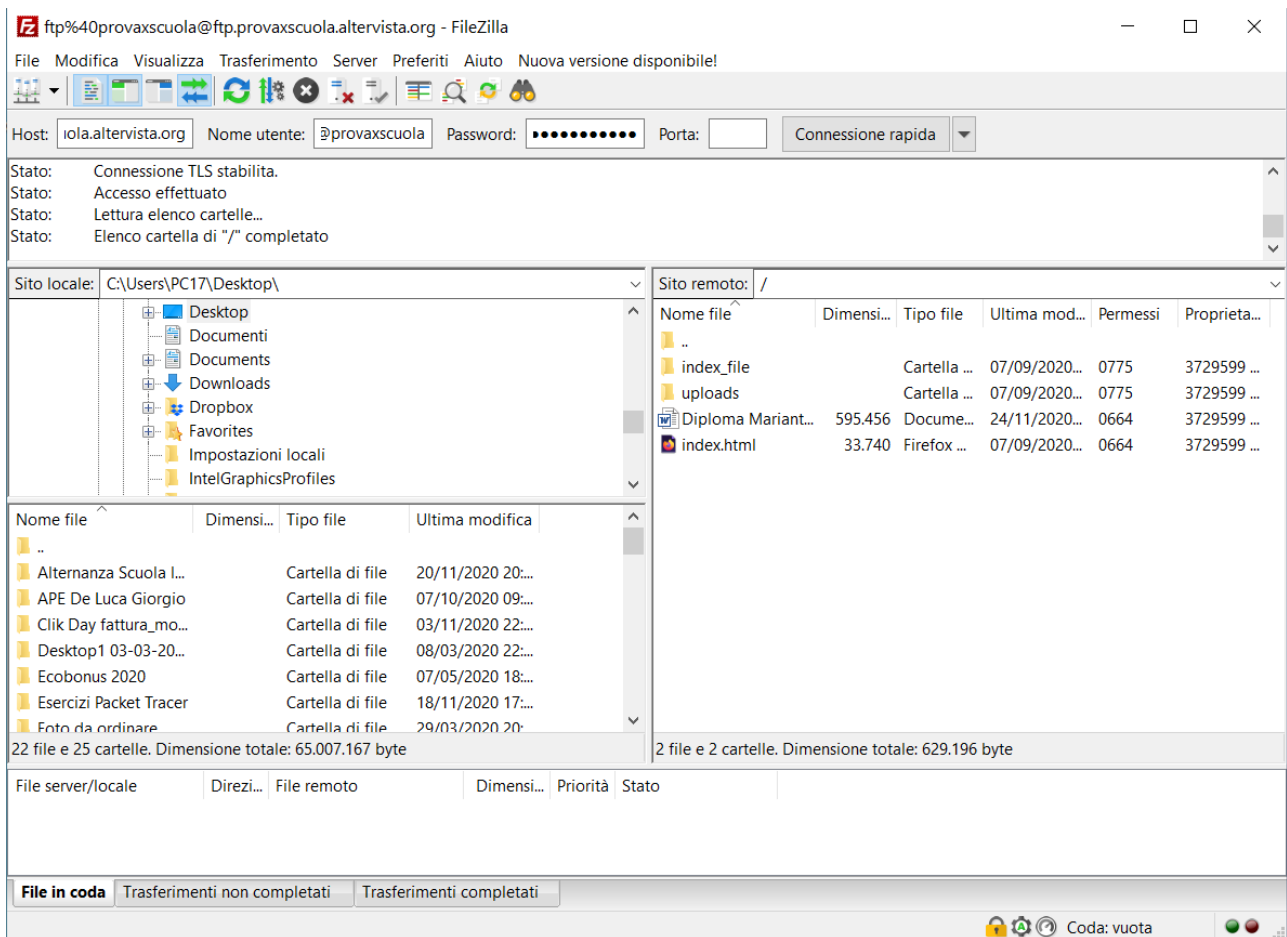
- connessione al server remoto;
- trasferimento di un file dal client al server (put);
- trasferimento di un file dal server al client (get);
- lettura dei file presenti nella directory corrente del server (dir);
- cambio directory (cd);
- disconnessione (bye).

put = upload  
get = download

Elenco di alcuni **client FTP**:

- *Ace FTP freeware*;
- *Directory uploader*;
- *Racing turtle FTP*;
- *Filezilla*;
- *Smart FTP*;
- *Cesar FTP*;
- *TRELLIAN*;
- *Cute FTP*;
- *WSFTP*;
- *Fastrean netfile FTP*;
- *Blazeware FTP*;

Alcuni prodotti possono fare da client e server.

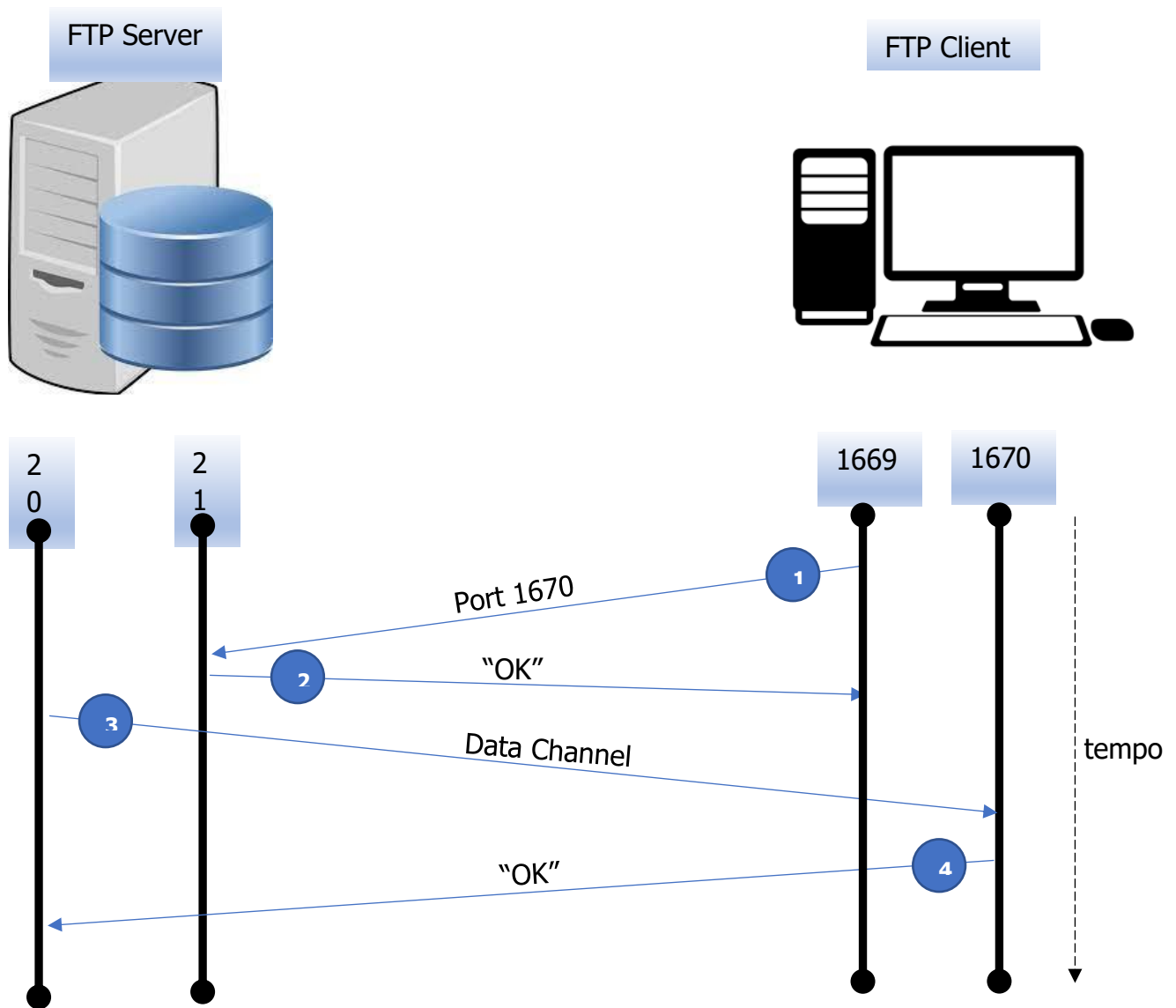


## La comunicazione FTP

Abbiamo due possibili situazioni:

### ➤ Collegamento normale (normal mode)

- Utente avvia una sessione FTP
- FTP client apre due porte superiori alla porta 1023 (1669 - 1670)
- Chiede connessione al server sulla **porta 21** del server indicando il protocollo TCP e la porta dati del client **port 1670**
- Server risponde ok sulla porta del client 1669
- Scambio di messaggi di controllo
- Il server avvia il **trasferimento dati** tra la sua porta 20 e 1670 del client

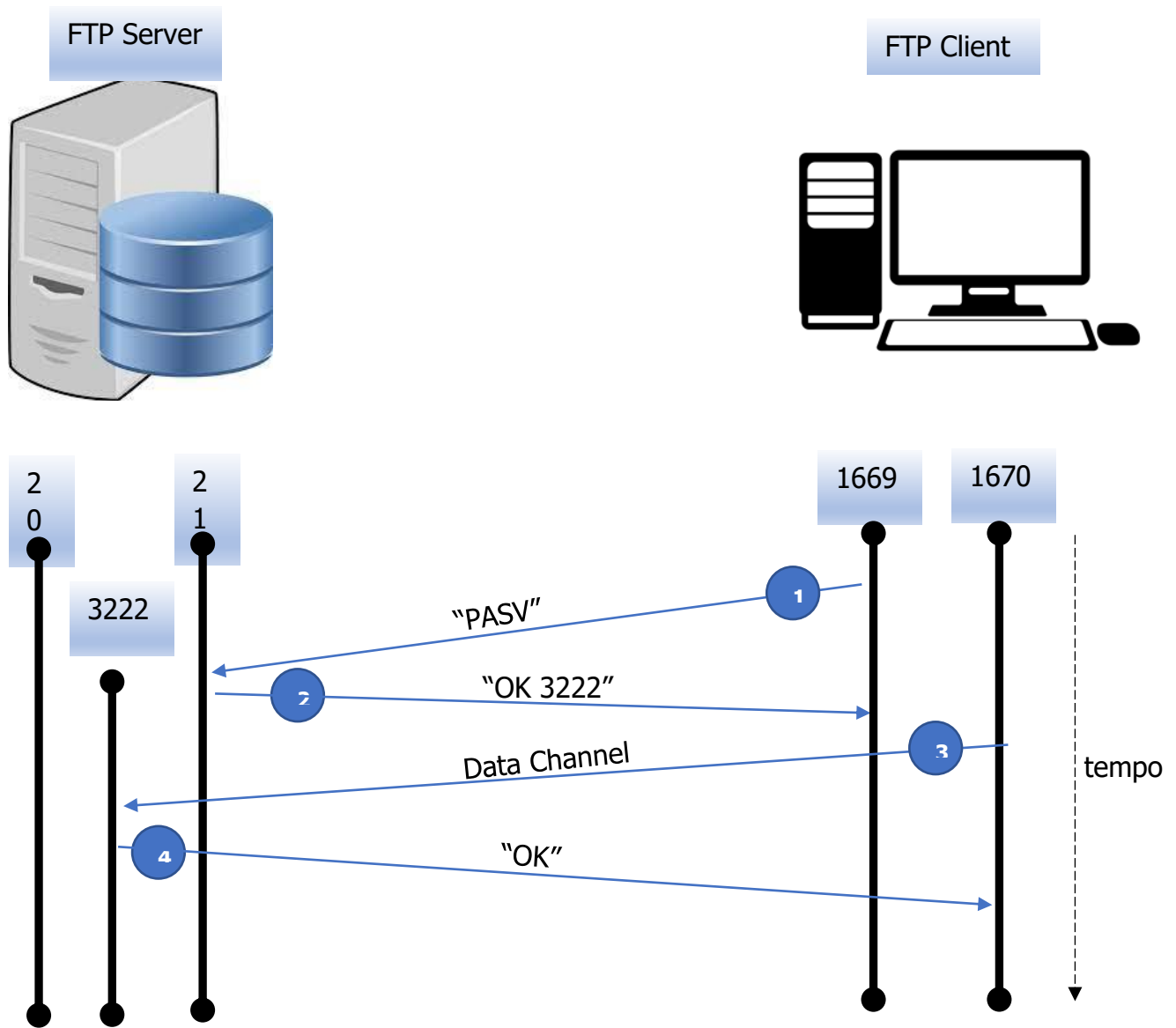


La connessione di controllo è **persistente** resta aperta anche quando termina lo scambio dei dati per una successiva trasmissione dati  
 Se le sessioni aperte sono troppe dopo un certo tempo si chiude anche la connessione di controllo.

➤ **Collegamento passivo (PSV mode).**

- Il client alloca due porte ma non comunica al server il suo numero di porta per la connessione dati
- richiede la connessione di controllo in cui il client indica il **PASV mode** invece della porta del server la 21
- il server risponde ok alla connessione e invia il suo numero di porta su cui avviare una connessione dati ad esempio **porta data 3322**
- Il client **attiva** una connessione dati (**Data Channel 1670-3322**)





Questa modalità è più sicura per attacchi hacker

Il client invia **comandi** al server in **ASCII**:

USER <nome utente>

PASS <password>

LIST

RETR <filename>

STOR <filename>

CWD <directory>

QUIT

Anche il server invia risposte al client in **ASCII**:

331 Username ok, password required

125 data connection already open, transfer starting

425 can't open data connection

452 Error writing

1 azione è stata iniziata

2 azione completata con successo

3 comando accettato ma in sospeso mancano altre informazioni

4 comando non accettato per errore provvisorio

5 comando non accettato per errore irrecuperabile

## FTPS

Sempre in termini di sicurezza, è doveroso ricordare che il protocollo **FTP** non prevede alcuna cifratura per i dati scambiati tra client e server e tra questi vengono scambiati nomi utenti, password, comandi, codici di risposta e file trasferiti che possono essere agevolmente “sniffati” da malintenzionati. Per ovviare a questo problema è stata definita una nuova specifica, la **RFC 4217**, che aggiunge al protocollo **FTP** originale un layer di cifratura **SSL** più una nuova serie di comandi e codici di risposta: tale protocollo prende il nome di **FTPS**.

Tutti i trasferimenti FTPS sono cifrati e l'algoritmo di cifratura è negoziato con il server: se viene impostato, la finestra del trasferimento mostrerà il simbolo di un “lucchetto” a indicare il trasferimento sicuro.

## Domain Name System (DNS).

Quando vogliamo accedere ad un sito, di solito digitiamo nella barra degli indirizzi del browser un nome (dominio) del tipo **www.peresempio.it**, in questo modo accediamo alla pagina web ed alle informazioni memorizzate all'interno di un Server collegato ad internet. Sappiamo, però, che i dispositivi sulla rete sono individuati da un insieme di numeri, gli indirizzi IP. Da qualche parte deve esserci, quindi, un 'archivio' che associa **www.peresempio.it** con l'indirizzo IP, o meglio, un archivio che associ tutti i domini pubblici con i corrispondenti indirizzi IP così che, tramite l'indirizzo IP, poter individuare sulla rete il server che contiene il sito **www.peresempio.it**.

Come si può notare il nome di un dominio è formato da un insieme di stringhe intervallate da punti: la parte più importante è quella più a destra (.it) detto *dominio di primo livello*; la parola successiva (peresempio) è detta *dominio di secondo livello* quindi www indica che siamo in presenza di un IP relativo ad una pagina web.

L'operazione di tradurre un dominio in un indirizzo IP è affidata ad un insieme di server detti Server DNS. Vediamo quali sono questi server e quali sono le operazioni che svolgono.

### Resolver DNS

Il *resolver DNS* o *resolver ricorsivo* funge da intermediario principale tra un computer e gli altri server DNS. Il suo scopo è quello di inoltrare una richiesta ad altri server del sistema dei nomi di dominio per poi rispedirla indietro una volta soddisfatta.

Quando il *resolver DNS* riceve una richiesta, per prima cosa cerca nella sua memoria cache un indirizzo IP corrispondente al nome di dominio presente nella richiesta. Infatti, se al Resolver DNS è stata già fatta la medesima richiesta da un utente, il *resolver DNS*, una volta soddisfatta la richiesta, memorizza il dominio e l'indirizzo IP associato, nella sua memoria cache. Quindi se il dominio (con associato l'indirizzo IP) viene trovato nella sua cache, la richiesta inviata ai server DNS termina qui: il resolver DNS invia l'IP corrispondente al client e l'utente si collegherà immediatamente il sito che desidera visitare.

Tuttavia, se non viene trovata alcuna corrispondenza nella sua cache, il *resolver DNS* invierà la richiesta al server DNS successivo: il **root nameserver**.

### Root nameserver

Il *root nameserver* o *server root DNS* si trova in cima alla gerarchia DNS.

Non conserva l'informazione che si sta cercando, ovvero l'indirizzo IP che corrisponde al nome di dominio, ma fornisce indicazioni su dove è possibile trovarlo.

Quando il root nameserver riceve una richiesta dal resolver DNS ricorsivo, identifica il dominio di *primo livello* del nome di dominio. Quindi, indicherà al resolver ricorsivo di andare al *nameserver TLD* corretto (ogni primo livello del dominio viene gestito da un server TLD dedicato).

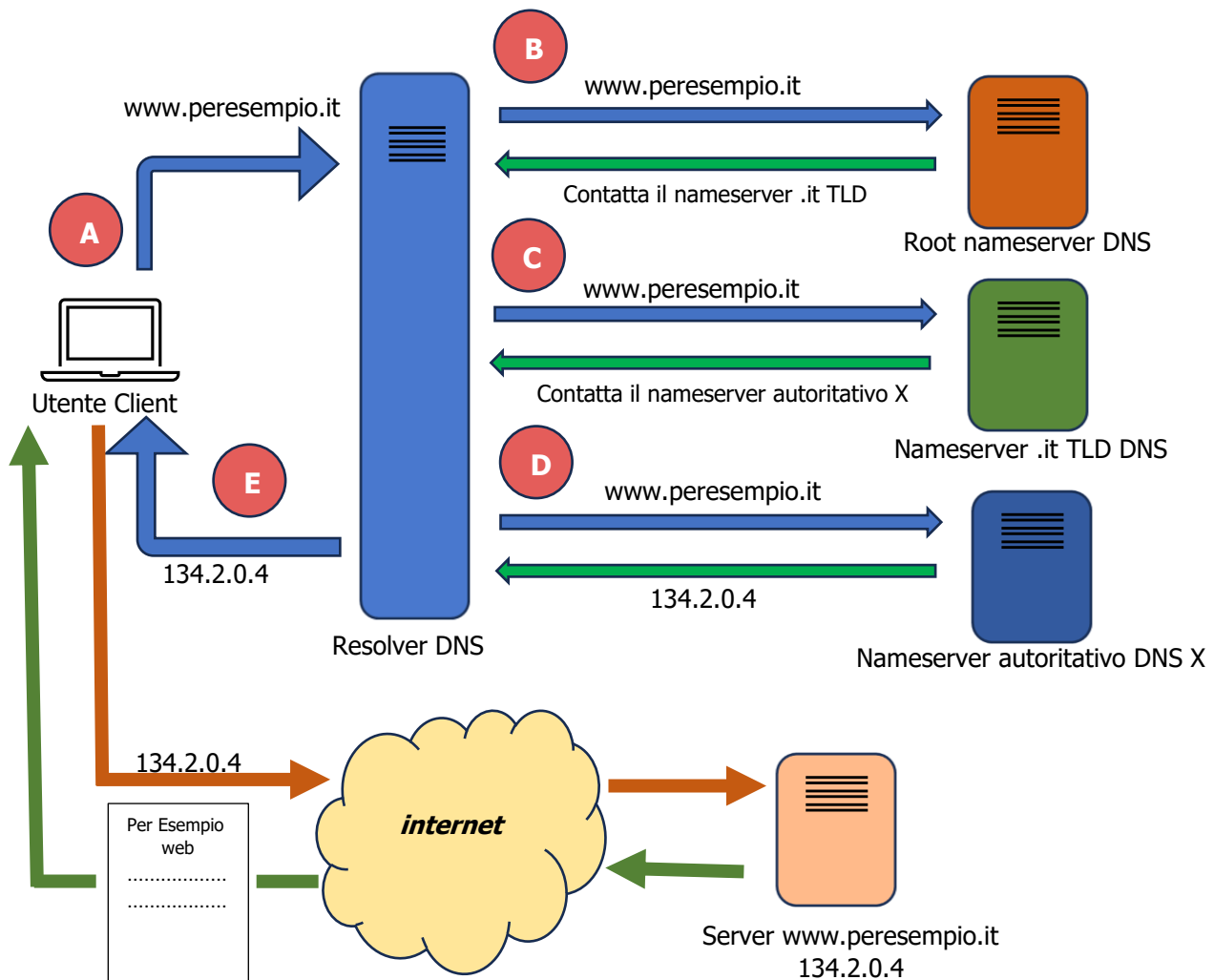
**Nameserver TLD**

Il nameserver TLD (Top Level Domain) è una funzione del server DNS responsabile della memorizzazione e della gestione delle informazioni sui nomi che utilizzano uno specifico dominio di primo livello (TLD). Un TLD è l'estremo di un nome di dominio, come .it, .com, .org, .online, .net...

Se la richiesta è di trovare l'indirizzo IP di **www.pereseempio.it**, il root nameserver reindirizzerà il resolver DNS ricorsivo al nameserver TLD .it. Successivamente, il nameserver TLD informerà il resolver sulla posizione dell'indirizzo IP corrispondente, presso uno specifico **nameserver autoritativo**.

**Nameserver autoritativo**

Il **nameserver autoritativo** o **server DNS autoritativo** è l'autorità finale nel processo di risoluzione DNS. Memorizza tutte le informazioni relative al nome di dominio che si desidera visitare, compreso il suo indirizzo IP. Il resolver ricorsivo ottiene l'indirizzo IP e lo rimanda al computer del client che ha effettuato la richiesta, indirizzandolo al sito.



**A** Il **client** invia al **resolver DNS** il dominio: se il **resolver DNS** ha già memorizzato nella sua cache l'indirizzo IP relativo al dominio, lo invia al **client** e stop. Altrimenti...

- B** Invia la richiesta al **server root DNS** il quale esamina il primo livello del dominio e indica al *resolver DNS* il *server TLD* a cui collegarsi.
- C** Il *resolver DNS* si collega al **server TLD**, indicato dal *server root DNS*, inviandogli la richiesta. Nei *server TLD* sono conservate tutte le informazioni relative ai domini che hanno lo stesso primo livello: ci saranno quindi *server TLD .it* (i domini italiani)... *TLD .com* (i domini commerciali) e così via. Il *server TLD* risponde al *resolver DNS* inviandogli l'indirizzo del **server autoritativo** che contiene l'indirizzo IP del dominio.
- D** Il *resolver DNS* contatta il **server autoritativo**, indicatogli dal *server TLD*, che gli restituisce l'IP associato al dominio.
- E** Il *resolver DNS* conserva nella sua cache le informazioni ricevute ed invia l'IP al *client* che, conoscendo l'indirizzo IP, può collegarsi al server del sito **www.peresempio.it**.