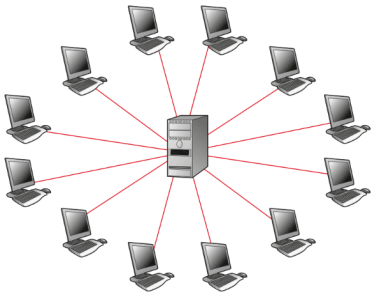
Architetture rete

Architettura client-server

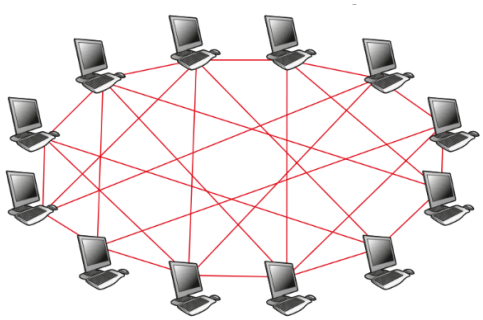


Architettura dove tutti i dispositivi si collegano ad un dispositivo centrale, detto **server attivo**.

Architetture P2P (*peer-to-peer*)

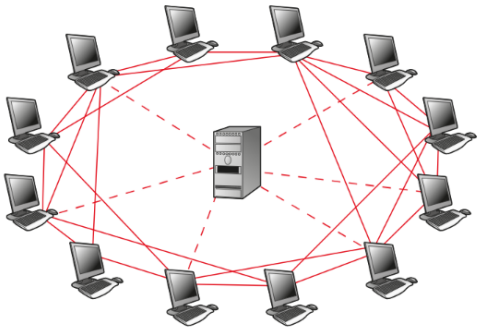
Detta **peer**, cioè formata da *entità di pari grado*, che dialogano direttamente tra loro.
Ci sono più tipi:

P2P decentralizzato



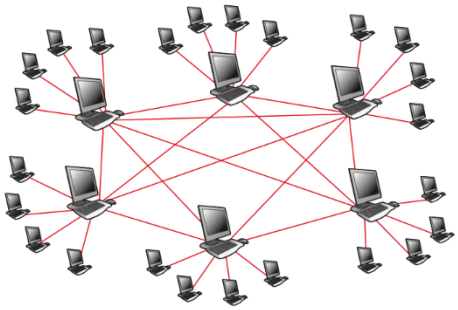
I peer funzionano sia come *server* che come *client*. (funzionalità simmetrica detta **servent**).
Le risorse sono identificate tramite *indirizzi IP dinamici*.
Saranno usati metodi di indirizzamento alternativi, superiori al livello IP.
Queste reti sono capaci di adattarsi ad un continuo cambiamento di peer, mantenendo la connettività e le prestazioni senza necessitare un'unità centrale.

P2P centralizzato



I peer sono coloro che richiedono e contengono i file.
Un server centrale, **directory server**, conserva gli **index**, cioè le informazioni su dove si possono trovare le risorse. (*mapping risorse dei peer*)

P2P ibrido



Alcuni peer, eletti tramite specifici algoritmi, lavorano come **supernodi/super-peer/ultra-peer**, il cui scopo è di indicizzare.
Gli altri nodi vengono chiamati **leaf peer**.

Protocolli livello applicativo

Protocollo FTP

Protocollo usato per il trasferimento di file.
Lavora sui *socket* **20** e **21**.
È formato da **2 canali**, che lavorano in *parallelo*:

- **Connessione di controllo**, dove si trasmettono le *informazioni di controllo*, come *credenziali*, i *comandi da eseguire*.
  (Anche chiamato *control connection/command channel*)
- **Trasferimento di file**
  (anche chiamato *data connection/data channel*)

Viene usato tramite **2 software**:

- **FTP Server**
  Eseguito sul server, gestisce:
  - *Download/upload di file*

    - Recupero di trasferimenti interrotti
    - rimozione/rinomina di file
    - creazione di directory
    - navigazione tra directory

L'accesso ad esso viene eseguito tramite *credenziali*, dove più utenti possono avere **privilegi diversi**.
Alcuni offrono anche l'*accesso anonimo*, con generalmente il solo privilegio di lettura.

- **FTP Client**
  Eseguito sul client, è generalmente composto da 2 componenti:
    - **Parte di comunicazione**, *back-end* che implementa il protocollo
    - **Interfaccia utente**, *generalmente grafica*
      Le funzioni comuni sono:
    - connessione al server remoto
    - **get**, trasferimento di un file da server a client
    - **put**, trasferimento di un file dal client al server
    - **dir**, lettura dei file nella directory selezionata
    - **cd**, cambiamento directory
    - **bye**, disconnessione dal server

Questo protocollo, però, non sfrutta algoritmi di crittografia, scambiando messaggi in chiaro.

Ci sono 2 protocolli che usano cifrature:

- **FTPS**, che aggiunge un **layer di cifratura SSL** (21 e il socket TLS)
- **SFTP**, che sfrutta **SSH** come crittografia (22)

HTTP

HTTP, usato per trasferire i file web, come le pagine dei siti.
HyperText Transfer Protocol. client-server.
Permette il trasferimento dei siti web.

Il Client fa la richiesta al server.
Il server invia la risposta al client.

URL, rappresenta l'indirizzo alla risorsa.
Uniform Resource Locator
È formato dal:

- *Protocollo*, http o https.

- *Dominio*, nome dell'host. Corrisponde all'indirizzo IP del server.
- *Percorso relativo del file*.
  L'**URI** è l'*Uniform Resource Indicator*, che è solo il percorso relativo al file senza dominio.

HTTP usa **socket 80**.
HTTPS usa **socket 443**.

Una pagina web è formata da tanti elementi.
Titolo, immagini, testo...
Si fa una richiesta per ogni elemento richiesto.

Ci sono 2 tipi di connessioni per il trasferimento delle risorse:

- **Persistente**
  Più elementi possono essere trasmessi tramite una sola connessione. Viene chiusa dopo una quantità configurata di tempo. (*timeout*)
- **Non persistente**
  La connessione deve essere creata per ogni elemento da trasmettere.
  Più lento. Bisogna eseguire l'handshake.

Caratteristiche connessioni non persistenti:

- Bisogna eseguire 3-way handshake ogni volta
- Overhead per ogni connessione TCP
- I browser aprono spesso connessioni parallele
  Caratteristiche connessioni persistenti:
- La connessione TCP rimane aperta per un tempo configurato dopo l'invio della risposta.
- I successivi messaggi possono usare la connessione già stabilita.
- Invia le richieste quando incontra un oggetto referenziato. (*link, immagini...*)
  Quando trova altri file che devono essere aggiunti alla pagina
- Serve eseguire un solo 3-way handshake per trasmettere tutti gli oggetti

Il **parsing** è l'interpretazione del codice html.

Le richieste http vengono eseguite con porte casuali, maggiori di 1024.
La porta 80 viene usata solo per ricevere le richieste.
Il server mittente può poi inviare i dati su porte diverse dalla 80.

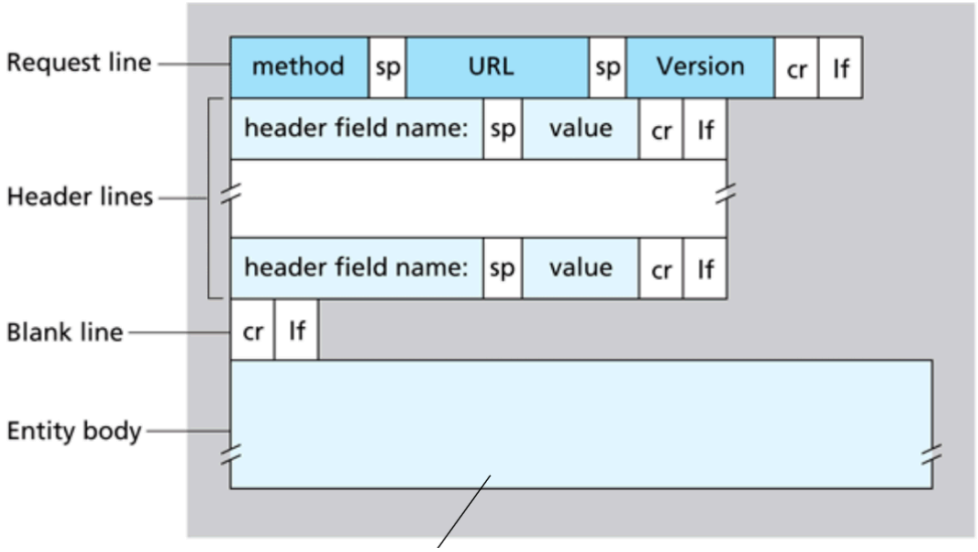
All'inizio il server manda la struttura della risorsa, che indica quali oggetti bisogna richiedere.

Tra le tecniche per velocizzare/ottimizzare la CPU si usa la tecnica delle **pipeline**.
I processi vengono divisi i pezzi, e vengono mandati in parallelo, scalati nel tempo.

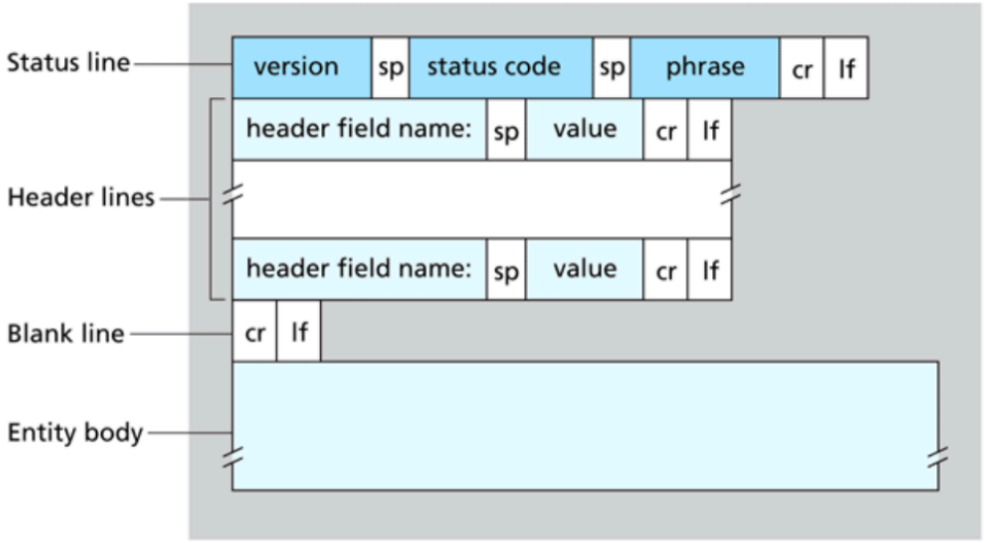
Il messaggio di richiesta http è formato da:

- **Metodo** usato, *GET* o *PUT*
- **URL**
- **Versione HTTP**
- *header file name*, per modellare le risposte
  Con informazioni come l'User Agent, necessarie per inviare una pagina adatta, ad esempio, al Browser, e la lingua.
  Viene anche inviato il parametro *if-modified-since*, per verificare se c'è una versione nuova durante la ricarica della pagina.
- *Riga vuota*
- **Body del messaggio**, usato per inviare file al server
  sono tutti formati con spazi separatori

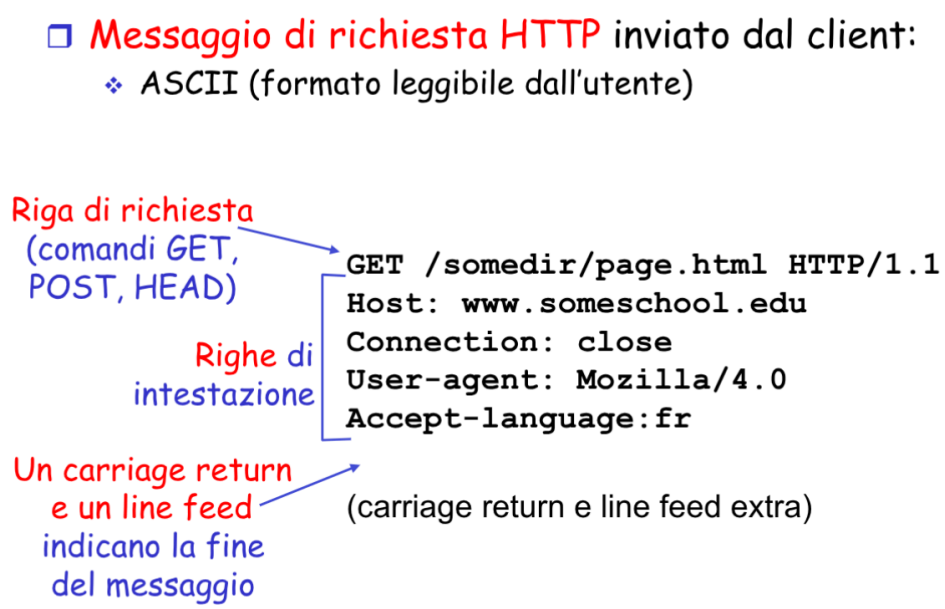
Formato di richiesta messaggi HTTP



Formato di risposta messaggi HTTP
Cambia solo la *Status line*

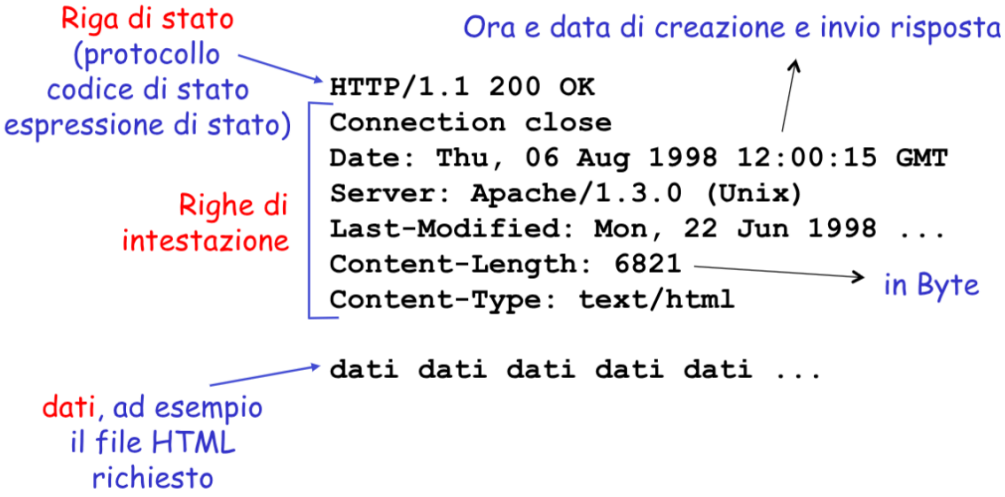


Domanda HTTP:



close cioè che la connessione deve essere chiusa dopo la trasmissione dei dati, *keep-alive* per connessione consistente.
Il parametro *Keep-Alive* stabilisce invece per quanto tempo deve rimanere aperta la connessione.
Il comando GET richiede l'*URI* (*Uniform Resource Indicator*), che indica la posizione del file, integrandolo poi con l'host

Risposta HTTP:



Last-Modified usato per quando viene ricaricata la pagina.

Abbiamo 5 **metodi**:

- **GET**, usato quando il client richiede una risorsa/documento al server
  - *GET-Condizionale*, usato ad esempio, quando si ricarica la pagina

- **HEAD**, usato quando il client non vuole scaricare il documento, ma ottenere l'header, cioè le informazioni del file.
- **POST**, usato per fornire input al server, arrivando nel *body* del messaggio http.
  Può essere usato in combinazione con la richiesta *GET*, ad esempio, nei motori di ricerca.
- **PUT**, usato per inviare un file al server.
  La posizione viene scritta nell'*URL*, e il documento viene inserito nel *body*.
- **DELETE**, usato per eliminare il documento specificato nell'*URL*.

Quando si ricarica la pagina, viene inviata la richiesta *HEAD*, verificando se c'è sul server una versione più recente del documento locale sul client.

Codice	espressione	Significato
200	OK	La trasmissione si è completata correttamente
301	*Moved Permanently*	La posizione della risorsa è cambiata. La nuova posizione sarà scritta nell'header *Location* della risposta
400	*Bad Request*	Messaggio di richiesta non compreso dal server
403	*Forbidden*	La risorsa è raggiungibile ma non si hanno i privilegi necessari per accedere a essa
404	*Not Found*	La risorsa non è stata trovata sul server
505	*HTTP Version Not Supported*	Il server non supporta/ non ha la versione HTTP usata dal client

Cookie

HTTP è un protocollo *senza-stato*, non viene memorizzato nulla dopo il trasferimento della risorsa.
Tengono traccia delle risposte/richieste i **cookie**, usati anche per memorizzare alcune informazioni come le *credenziali* e il *session-id*

Caching

Il **caching** viene usato per velocizzare il trasferimento/caricamento delle pagine web. Ci sono 2 tipi:

- **Browser**, una *cache locale*, personalizzabile dall'utente. (che può impostare dopo quanto tempo svuotarla automaticamente)
  Con il parametro *last-modified* viene verificato che sia la versione più recente della pagina.
- **Proxy**, che sono dei piccoli server usati per memorizzare le pagine web e alleviare il carico del server centrale.
  È un piccolo server in mezzo tra l'utente e il server principale.

E-Mail

Ci sono 2 tipi:

- **Web mail**, accessibili con qualunque computer con accesso ad internet, con la necessità di ricordare la password (se si accede da computer diversi). Le mail non vengono salvate sul computer.
- **Pop mail**, accessibili solo dal computer che ha il client installato, la password viene memorizzata nel momento di creazione dell'account.

Ci sono 2 programmi principali usati nei servizi e-mail:

- **MUA**, *Mail User Agent*, il programma che viene usato dall'utente per la inviare, ricevere e leggere email. L'interfaccia tra l'utente e il sistema.
  Crea e legge i messaggi gestione di cartelle, allegati, contatti, firme...
- **MTA**, *Mail Transfer Agent*, il programma che si occupa di trasportare le email da un server all'altro. Riceve, smista e consegna i messaggi.

POP3 (*Post Office Protocol*) è un protocollo usato per scaricare le email dai server di posta sui propri dispositivi.
Le email dopo il download vengono eliminate sul server.
Usa la porta *110* per connessioni non cifrate e *995* per connessioni sicure.
La connessione avviene in 3 fasi:

- **AUTORIZZAZIONE**, dove vengono verificate le credenziali
- **TRANSAZIONE**, in cui viene scaricata la posta, e al termine del download il cliente termina la connessione con il comando *QUIT*
- **AGGIORNAMENTO**, in cui il server elimina tutti i messaggi dalla casella postale e chiude la connessione

Un alternativa a questo protocollo è **IMAP** (*Internet Message Access Protocol*), più evoluto ma meno diffuso.
Oltre alle funzionalità di *POP3*, supporta anche:

- Rinominare la propria casella elettronica
- Cancellare singoli messaggi senza doverli prelevare
- Leggere le intestazioni senza prelevare l'email intera
- Prelevare solo porzioni dei messaggi

C'è anche il protocollo **SMPT** (*Simple Mail Transfer Protocol*), usato per inviare email di testo, senza immagini o stile.
Il body è fatto di caratteri ASCII, e con la MIME è possibile anche inserire altri effetti.
L'utente invia l'email al server SMTP, che la invia ad un server con POP/IMAP, che la invia poi

all'utente destinatario.
MIME, *Multipurpose Internet Mail Extensions*, viene usato per definire il formato dei messaggi di posta elettronica.

DNS

Il *Domain Name System* viene usato per tradurre i *nomi di dominio* (*matteoc.cloud*) in *indirizzi IP* (*51.254.146.120*), e viceversa.
Il Browser mantiene una cache DNS locale in modo da velocizzare la navigazione, richiedendo l'indirizzo solo se non viene trovato nel DNS locale.
I 2 principali sono:

- *Google* **8.8.8.8** (il cui sito include una piccola pagina per visualizzare informazioni sui domini)
- *CloudFlare* **1.1.1.1** (pagina che fornisce informazioni sul DNS)

Telnet

Telnet è un vecchio protocollo di rete che permette di accedere in modo remoto ad un computer tramite la riga di comando.
Usa la porta **23**.
Non fornisce una connessione sicura, i dati viaggiano in chiaro.
Oggi è spesso sostituito con **SSH** (*Secure Shell*), dove i dati viaggiano in modo cifrato.

VLAN

pg.110-113
Le *Virtual LAN* vengono usate per creare sotto-reti in modo più economico, scalabile e ottimizzato.
Si necessita di uno **switch di livello 3**.

Sono identificate con un **VID** (*Virtual Identificator*), valore 8bit, compreso tra *1 e 4094*
Abbiamo 2 tipi:

- **VLAN Untagged**, dove le VLAN sono assegnate sulle porte dello switch.
  Le operazioni che vengono svoltee sono 3:
  - **ingress**, dove abbiamo i frame che entrano nella VLAN assegnata alla porta.
  - **forwarding**, il frame può essere inoltrato solo ai membri della VLAN.
  - **egress**, il frame viene inoltrato a tutte le porte valide.
- **VLAN Tagged**, viene definito dal protocollo **802.1Q**.
  Nei frame vengono aggiunti **4 byte di tag**.
  Le porte tagged vengono anche chiamate **porte trunk**.
  I **primi 2 byte** sono il **TPI** (*Tag Protocol Identifier*), che contiene l'**EtherType**, che evidenzia

il formato del frame.
Gli **ultimi 2 byte** sono il **TCI**/*VLAN Tag* (*Tag Control Information*), strutturato in:
user_priority, 3 bit, che indica la priorità del frame.
CFI, 1 bit, che indica se il *MAC address* è in forma canonica.
VID, 12 bit.

Il *VID 1* è usato dalla *VLAN Nativa*, che include tutti i dispositivi.
Poco sicura, spesso usata durante gli attacchi.
VLAN su cui naviga il traffico di controllo, come i ping.

Le porte possono essere distinte in **porte trunk** e **untagged**:

- Se la porta è di tipo *untagged*, i frame header non avranno informazioni aggiuntive sul tag.
  Queste porte sono dette **porte d'accesso**, e i link associati ad esse **access link**.
- Se la porta è di tipo *tagged*, i frame header avranno i tag aggiuntivi sulla VLAN.
  Queste porte sono dette **porte trunk**, e i link associati ad esse **trunk link**.

Le *porte trunk* vengono usate per collegare più switch tra di loro.

Alcuni switch supportano le **porte ibride**, su cui può viaggiare sia traffico untagged che tagged contemporaneamente.
Le operazioni che devono svolgere gli switch sono:

- **ingress**, dove viene identificato il tipo di frame, e identificata la VLAN di appartenenza
  - Se il frame è untagged, la *VLAN di appartenenza* viene identificata con quella associata alla porta
  - Se il frame è tagged, la *VLAN di appartenenza* viene identificata dai tag
- **forwarding**, dopo l'identificazione della VLAN di appartenenza, vengono applicate le regole di forwarding e identificata la porta di uscita
- **egress**: se necessario viene effettuato l'inserimento o rimozione dei tag:
  - Se il frame in ingresso è di tipo **802.1Q** e la porta in uscita è di tipo **tagged**, il frame viene inoltrato senza modifiche
  - Se il frame in ingresso è **untagged** e la porta in uscita è di tipo **untagged**, il frame viene inoltrato senza modifiche
  - Se il frame in ingresso è di tipo **802.1Q** e la porta di uscita è di tipo **untagged**, bisogna **rimuovere i tag**
  - Se il frame in ingresso è di tipo **802.3** (no vlan) e la porta in uscita è di tipo **tagged** è necessario **inserire i tag**

Interfacciamento tra 2 VLAN

Ci sono 2 metodi:

- **Inter-VLAN tradizionale**, dove lo switch viene collegato al router con più cavi, ognuno rappresentante una delle VLAN. (le porte vengono impostate in modalità **access**)
- **Router-On-A-Stick**, dove viene usato un solo cavo, dove le VLAN vengono separate al livello logico tramite il protocollo *802.1q* (le porte vengono impostate in modalità **trunk**)
  Le VLAN vengono impostate sulla porta interfacciandosi sulla stessa porta, ma in vlan diverse.
  es. *interface FA 0/0.10* permette di interfacciarsi sulla porta 0/0 in una sottointerfaccia, a cui il traffico sarà poi etichettato come una VLAN specifica con il comando **encapsulation dot1q *VID***, e l'impostazione del default gateway.