

ESERCIZI – argomento: PROTOCOLLO HTTP

prof. Leonardo Sportiello

Esercizio 1

Analisi di una richiesta HTTP

Comprendere come funziona il protocollo HTTP e vedere una richiesta HTTP reale attraverso gli strumenti di sviluppo di Google Chrome.

Istruzioni:

1. Apri Google Chrome e vai su un sito web che supporti **HTTPS**;
2. Apri gli **Strumenti di Sviluppo (F12)** oppure **CTRL+Shift+I**;
3. Naviga alla scheda “**Network/Rete**”;
(in questa scheda sono presenti tutte le richieste HTTP inviate dal browser al server)
4. Non chiudere la scheda “Network/Rete” e ricarica la pagina con **F5**;
(vedrai che molte richieste vengono elencate in tempo reale man mano che il browser carica la pagina)
5. Esamina la prima richiesta HTTP in elenco, **visualizza i dettagli**;
6. **Analisi delle sezioni della richiesta:**
 - **URL di richiesta:** Mostra l’URL completo della risorsa richiesta;
 - **Metodo:** Indica il metodo HTTP utilizzato (come **GET** o **POST**);
 - **Status code/Codice di stato:** Mostra il codice di stato restituito dal server;
(come 200 OK o 404 Not Found).
 - **Headers/Intestazioni:** Clicca sulla sezione Headers per vedere gli header della richiesta.
 - **Generale:** Contiene informazioni generali come l’URL, il metodo e lo status code.
 - **Request Headers/Intestazione richiesta:** Mostra gli header inviati dal browser al server, come Host, User-Agent, Accept, Connection, ecc.
 - **Response Headers/Intestazione risposta:** Mostra gli header inviati dal server in risposta, come Content-Type, Content-Length, Set-Cookie, ecc.

Rispondi alle seguenti domande:

1. Qual è il metodo HTTP utilizzato per la richiesta principale?
2. Quale status code è stato restituito dal server? Cosa indica questo codice?
3. Quali header di richiesta e di risposta sono presenti?
4. Il browser ha richiesto risorse aggiuntive (CSS, immagini, script)? Quali?
5. Cosa succede se clicchi su un link o ricarichi la pagina? Noti cambiamenti nelle richieste?
6. C'è qualche differenza tra le richieste GET e POST che riesci a identificare?

Esercizio 2

Risposte HTTP e Proxy.

1. GET http://cs.unibg.it/index.html HTTP/1.1
2. User-Agent: Mozilla/4.0
3. Accept: text/html, image/gif, image/jpeg
4. If-Modified-Since: 27 Feb 2017 08:10:00

1. Scrivi due risposte plausibili del server HTTP (solo la status line).
2. Supponendo che il messaggio sia inviato attraverso un Proxy, specifica il comportamento del Proxy stesso.

FORMULE GENERICHE

Calcolo delle prestazioni protocollo HTTP

Formulario utile alla risoluzione dei prossimi quesiti proposti.

1. Calcolo del tempo di download

- **Tempo di trasferimento di un oggetto (connessione non persistente)**

- Quando devi calcolare il tempo necessario per trasferire un oggetto attraverso una connessione TCP:

$$t_{\text{oggetto}} = \frac{L}{r} \qquad t_{\text{oggetto}} = \frac{L}{C}$$

oppure

- **t_{oggetto}** : tempo di trasferimento di un singolo oggetto (secondi)
 - **L** : dimensione dell'oggetto (kbit)
 - **C** : capacità totale del collegamento (kbit/s)
 - **r** : velocità di trasmissione del collegamento (kbit/s)
-
- **Capacità del collegamento condiviso tra più connessioni parallele**
 - Quando il collegamento viene condiviso tra più connessioni parallele, la velocità effettiva per ogni connessione si calcola così:

$$r = \frac{C}{N}$$

- **r** : velocità di trasmissione di una singola connessione (kbit/s)
 - **C** : capacità totale del collegamento (kbit/s)
 - **N** : numero totale di connessioni aperte in parallelo
-
- **Tempo totale di download per connessioni in parallelo**
 - Poiché le connessioni sono parallele, il tempo totale sarà determinato dal tempo necessario per trasferire un singolo oggetto:

$$t_{\text{totale}} = t_{\text{oggetto}}$$

- **t_{totale}** : tempo totale per scaricare tutti gli oggetti (secondi)
 - **t_{oggetto}** : tempo di trasferimento di un singolo oggetto (secondi)
-
- **Tempo totale di download per connessioni in serie**
 - Se le connessioni sono aperte in serie, il tempo totale per trasferire più oggetti è la somma dei tempi di trasferimento di ciascun oggetto:

$$t_{\text{totale}} = N \times t_{\text{oggetto}} = N \times \frac{L}{C}$$

- **t_{totale}** : tempo totale per scaricare tutti gli oggetti (secondi)
- **C** : capacità totale del collegamento (kbit/s)
- **N** : numero di oggetti da scaricare
- **L** : dimensione dell'oggetto (kbit)

2. Condivisione del collegamento e connessioni TCP

▪ Capacità condivisa tra più flussi

- Quando il collegamento viene condiviso tra più flussi TCP (ad esempio con più trasferimenti di file in parallelo), la capacità del collegamento viene divisa equamente tra i flussi:

$$r = \frac{C}{n}$$

- ***n***: numero totale di flussi sul collegamento
- ***C***: capacità totale del collegamento (kbit/s)
- ***r***: velocità effettiva di trasferimento per il flusso HTTP del client (kbit/s)

▪ Tempo di trasferimento di un oggetto per flusso

- Il tempo di trasferimento per ogni oggetto dipende dalla velocità di trasferimento effettiva:

$$t_{\text{oggetto}} = \frac{L}{r}$$

- ***r***: velocità di trasferimento effettiva (kbit/s)
- ***L***: dimensione dell'oggetto (kbit)

▪ Tempo totale con connessione persistente

- Se il client apre una connessione persistente per scaricare tutti gli oggetti, il tempo totale è:

$$t_{\text{totale}} = N \times t_{\text{oggetto}}$$

↖
al valore sommario RTT

- ***N***: numero di oggetti (compreso il file HTML)
- ***t_{oggetto}***: tempo di trasferimento per ciascun oggetto (secondi)

▪ Tempo totale con connessioni parallele non persistenti

- Se il client apre tutte le connessioni in parallelo, il tempo totale sarà pari al tempo necessario per scaricare il singolo oggetto più lungo:

$$t_{\text{totale}} = t_{\text{oggetto}}$$

↖
al valore sommario RTT

- ***t_{oggetto}***: tempo di trasferimento per ciascun oggetto (secondi)

3. Caching e ritardo medio

- **Tempo di risposta nel caso di Cache Hit**

- Se la risorsa è già presente nella cache del proxy (cache hit), il tempo di risposta sarà:

$$T_{\text{hit}} = \frac{L}{C}$$

- T_{hit} : tempo per scaricare la pagina dal proxy (secondi)
- L : dimensione della pagina (kbit)
- C : capacità del collegamento client-proxy (kbit/s)

- **Tempo di risposta nel caso di Cache Miss**

- Se la risorsa non è presente nella cache e deve essere recuperata dal server, il tempo di risposta sarà:

$$T_{\text{miss}} = \frac{L}{c}$$

- T_{miss} : tempo per scaricare la pagina dal server (secondi)
- L : dimensione della pagina (kbit)
- c : capacità del collegamento proxy-server (kbit/s)

- **Tempo medio complessivo**

- Il tempo medio complessivo per ottenere una pagina dipende dalla probabilità che la risorsa sia disponibile nella cache:

$$T_{\text{medio}} = P \times T_{\text{hit}} + Q \times T_{\text{miss}}$$

- T_{medio} : tempo medio complessivo per ottenere una pagina (secondi)
- P : probabilità che la pagina sia disponibile nella cache (cache hit rate)
- Q : probabilità che la pagina non sia disponibile nella cache (cache miss rate, $Q = 1 - P$)
- T_{hit} : tempo per una cache hit (secondi)
- T_{miss} : tempo per una cache miss (secondi)

Esercizio 3

un oggetto base + 10 altri
oggetti -> **totale N=11**

Calcolo del tempo di download

Un client HTTP richiede a un server una pagina web composta da un oggetto base (un file HTML) e da 10 altri oggetti. Ogni oggetto ha dimensione $L=200$ kbit. Il collegamento bidirezionale tra il client e il server ha una capacità $C=100$ kbit/s. I messaggi di controllo utilizzati per aprire la connessione TCP tra il client e il server, così come il messaggio HTTP "GET", hanno dimensione $m=100$ bit. Il ritardo di propagazione del collegamento è $\tau=100$ ms.

Calcola il tempo totale necessario perché il client riceva la pagina web e i 10 oggetti nei due seguenti casi:

1. Il client HTTP apre in **parallelo** e in modalità **non persistente** tutte le connessioni TCP necessarie per scaricare la pagina web e i suoi oggetti collegati (si ipotizzi che il ritmo di trasmissione di ogni connessione TCP sia uguale a $r = C/N$, dove N è il numero di connessioni aperte in parallelo).
2. Il client HTTP apre in **serie** 11 connessioni TCP in modalità **non persistente**.

Dati:

- Numero di oggetti: **11** (1 file HTML + 10 oggetti)
- Dimensione di ogni oggetto: $L = 200$ kbit
- Capacità del collegamento: $C = 100$ kbit/s
- Ritardo di propagazione $\tau = 100$ ms

Esercizio 4 – Link utile <https://convertlive.com/it/u/convertire/megabit/a/kilobit>

HTTP (Condivisione del collegamento e connessioni TCP)

Un client HTTP richiede a un server HTTP una pagina web costituita da un oggetto base (file HTML) e 11 altri oggetti. Ogni oggetto ha una dimensione $L=50$ kB. Il collegamento tra client e server HTTP è in grado di trasferire informazioni a una velocità $C=1$ Mbit/s in entrambe le direzioni. Il tempo di andata e ritorno (RTT) è pari a **150 ms**. Il collegamento è condiviso con altri 9 trasferimenti file di lunga durata verso altri server.

Assumendo che tutti i trasferimenti condividano equamente la capacità del collegamento ottenendo una velocità media pari a C/n , dove n è il numero di trasferimenti paralleli, calcola il tempo totale per ricevere interamente la pagina web richiesta nei due seguenti casi:

1. Il client HTTP apre una **singola connessione TCP persistente** per scaricare tutti gli oggetti della pagina web.
2. Il client HTTP apre in **parallelo in modalità non persistente** tutte le connessioni TCP necessarie per scaricare la pagina web.

Dati:

- Numero di oggetti: **12** (1 file HTML + 11 oggetti)
- Dimensione di ogni oggetto: $L = 50$ kB
- Capacità del collegamento: $C = 1$ Mbit/s
- RTT: 150 ms = 0.15 secondi
- Numero di flussi interferenti: 9
- Numero totale di flussi: $n = 9$ flussi interferenti + 1 flusso del client = 10

Esercizio 5

HTTP (Caching)

Un'azienda possiede una rete locale con un proxy HTTP con cache locale (vedi figura 1). I client sono collegati al proxy HTTP tramite collegamenti dedicati con capacità $C=1\text{ Gb/s}$. La probabilità che il contenuto (pagina web) richiesto dal generico client sia presente nella cache del proxy locale (cache hit rate) è $P=0.4$ (la probabilità che la pagina web richiesta non sia presente nella cache del proxy locale è $Q=0.6$).

Trova il ritardo medio sperimentato dal generico client da quando invia la richiesta HTTP per una pagina web a quando ottiene la pagina web richiesta.

Assumiamo che:

- I messaggi di richiesta HTTP siano di **100 [byte]**,
- La pagina web richiesta sia di **100 [kbyte]**,
- Il proxy HTTP abbia un canale di comunicazione con capacità equivalente a $c=100\text{ Mb/s}$ verso il server web che ospita le pagine web richieste,
- Il tempo di apertura delle connessioni TCP tra client e proxy e tra proxy e web server sia trascurabile

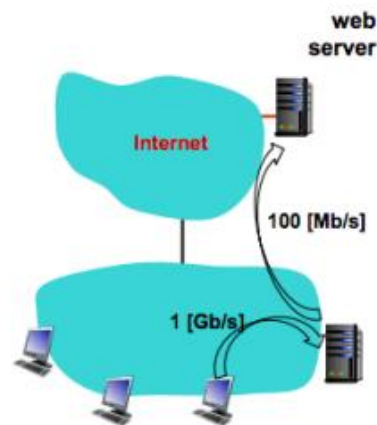


Figura 1 Topologia di riferimento

Dati:

- Cache hit rate: $P = 0.4$
- Cache miss rate: $Q = 0.6$
- Dimensione pagina web: $L = 100\text{ kB}$
- Capacità del collegamento client-proxy: $C = 1\text{ Gb/s}$
- Capacità del collegamento proxy-server: $c = 100\text{ Mb/s}$

NOTA BENE

Concetti approfonditi utili allo svolgimento degli esercizi

➤ Connessioni persistenti (o permanenti)

- Una connessione persistente (detta anche permanente) si riferisce al fatto che una singola connessione TCP viene mantenuta aperta per più richieste HTTP. Ciò significa che, dopo che una richiesta è stata completata, la connessione non viene chiusa, ma rimane aperta per gestire altre richieste senza dover riaprire una nuova connessione TCP.
- Esempio: Un client scarica una pagina web composta da un file HTML e 10 immagini. Con una connessione persistente, il client apre una singola connessione TCP, e tutti gli oggetti vengono trasferiti attraverso quella connessione senza che sia necessario aprirne di nuove.

➤ Connessioni non persistenti (o non permanenti)

- Una connessione non persistente (detta anche non permanente) si riferisce al fatto che una connessione TCP viene aperta per ciascuna richiesta HTTP, e viene chiusa immediatamente dopo il completamento del trasferimento di un oggetto. Questo significa che, per ogni oggetto richiesto, si deve stabilire una nuova connessione TCP.
- Esempio: Lo stesso client vuole scaricare una pagina web composta da un file HTML e 10 immagini. Con una connessione non persistente, il client deve aprire e chiudere una connessione TCP separata per ogni oggetto, con conseguente overhead per ogni trasferimento.

➤ Relazione tra serie/parallelo e persistente/non persistente

- **Connessioni in serie vs connessioni in parallelo**
 - **Connessioni in serie:**
 - Le connessioni in serie possono essere sia **persistenti** che **non persistenti**.
 - **Persistente in serie:** Una singola connessione TCP viene utilizzata per trasferire tutti gli oggetti, uno dopo l'altro, senza chiudere la connessione. La connessione rimane aperta per tutto il tempo.
 - **Non persistente in serie:** Una nuova connessione TCP viene aperta e chiusa per ogni oggetto, con conseguente overhead per ogni oggetto.
 - **Connessioni in parallelo:**
 - Anche le connessioni in parallelo possono essere sia **persistenti** che **non persistenti**.
 - **Persistente in parallelo:** Si apre una **connessione TCP persistente** ed è possibile gestire più richieste contemporaneamente tramite la stessa connessione.
 - **Non persistente in parallelo:** Ogni oggetto viene trasferito utilizzando una connessione TCP separata, e tutte le connessioni vengono aperte contemporaneamente. Questo significa che ci saranno più connessioni simultanee, ma ciascuna si chiuderà al termine del trasferimento.

Il **RTT** (*Round-Trip Time*), in italiano “tempo di andata e ritorno”, è il tempo totale impiegato da un pacchetto di dati per viaggiare dal client al server e tornare indietro al client con una risposta; misura il tempo necessario per completare un ciclo di comunicazione.