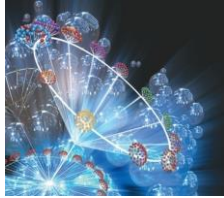


Corso di Laurea in Informatica (I Anno Magistrale)  
Reti geografiche: struttura, analisi e prestazioni  
A.A. 2019/2020



## Il protocollo HTTP e l'interazione fra HTTP e TCP

Delfina Malandrino  
<https://docenti.unisa.it/delfina.malandrino>  
[dmalandrino@unisa.it](mailto:dmalandrino@unisa.it)

## Organizzazione della lezione

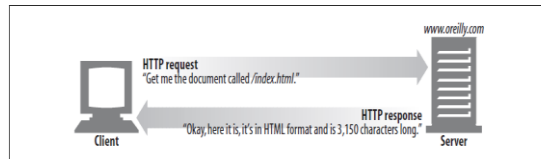
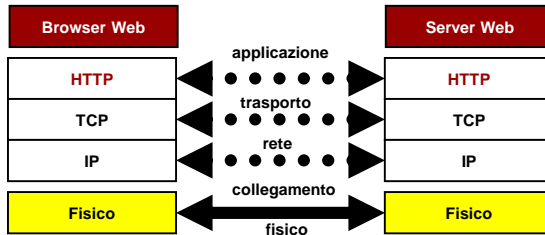
2

- HyperText Transfer Protocol (HTTP) - cenni
- Interazione fra HTTP e TCP
- Conclusioni

## HTTP: Hypertext Transfer Protocol

3

- The world's Web browsers, servers, and related Web applications all talk to each other through HTTP



## HTTP: Hypertext Transfer Protocol

4

- Protocollo per trasmettere informazioni in
  - ▣ formati multipli
  - ▣ linguaggi multipli
  - ▣ e diversi set di caratteri
- La sintassi di HTTP è basata su MIME: Multipurpose Internet Mail Extensions
  - ▣ il cui scopo era quello di definire messaggi che contenessero diversi tipi di dati, presenti anche contemporaneamente

## Storia ed evoluzione del protocollo

5

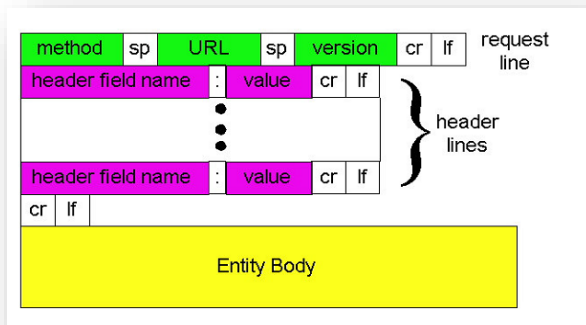
- Diverse versioni:
  - ▣ HTTP 1.0 è basato su RFC 1945
  - ▣ HTTP 1.1 è basato su RFC 2616
- Uno dei protocolli di più lenta ma significativa evoluzione
  - ▣ HTTP 0.9 a HTTP 1.0 richiese 4 anni
  - ▣ HTTP 1.0 a HTTP 1.1 richiese 4 anni (dal 2001 è stabile)
- Proposto (1990) da **Tim Berners Lee**

## Elementi del linguaggio

6

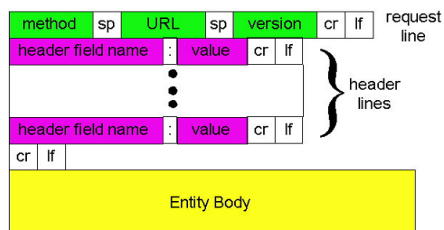
- Una richiesta HTTP ha il seguente formato:

Header lines {  
     Request-line  
     General-Headers  
     Request-Headers  
     Entity-Headers  
     CRLF  
     Body del messaggio (opzionale)



## Elementi de

7



(a) Request message

```
GET /test/hi-there.txt HTTP/1.1
Accept: text/*
Host: www.joes-hardware.com
```

*Start line**Headers**Body*

(b) Response message

```
HTTP/1.0 200 OK
Content-type: text/plain
Content-length: 19
```

```
Hi! I'm a message!
```

## Classi di risposta

8

- Presa in prestito da SMTP:
  - ▣ organizzazione in classi
  - ▣ caratterizzate numericamente:
    - 1xx: informative
    - 2xx: successo
    - 3xx: redirectione
    - 4xx: errore client
    - 5xx: errore server
- In aggiunta, una stringa di spiegazione viene fornita

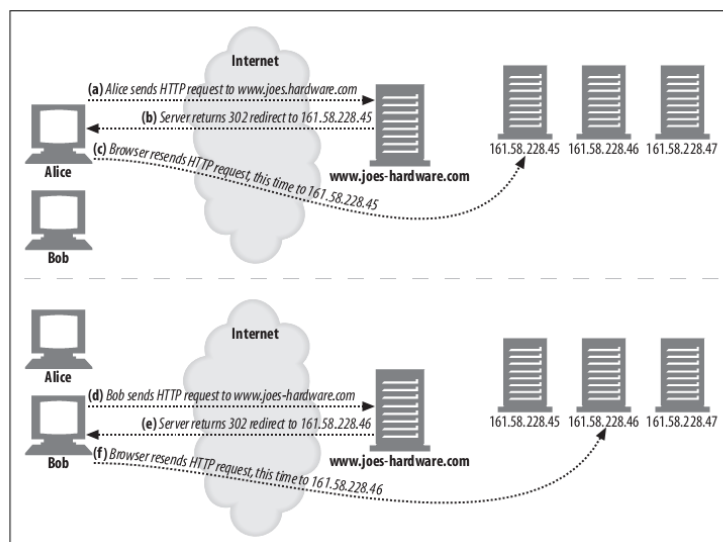
## 3xx: classe di response con redirectione

9

- Informa che altre azioni devono essere intraprese dal client
- Codici:
  - ▣ 300 Multiple Choices
    - non usato in pratica (server solo per il default)
  - ▣ 301 Moved Permanently
    - risorsa spostata definitivamente
    - utile per GET e HEAD
  - ▣ 302 Moved Temporarily
    - variante di 301
    - client possono continuare a usare la URL in futuro
    - 301 è cacheable mentre 302 no
  - ▣ 304 Not Modified
    - se la risorsa non è stata modificata dalla data indicata

## 302 Moved Temporarily

10



## Gli obiettivi di HTTP 1.1

11

- Nuovi concetti:
  - ▣ Meccanismo hop-by-hop
  - ▣ Ottimizzazione della banda
  - ▣ Gestione delle connessioni
  - ▣ Trasmissione del messaggio in chunk
  - ▣ Virtual hosting
  - ▣ Supporto per le varianti di una risorsa
  - ▣ Autenticazione crittografata
  - ▣ Miglioramento dei meccanismi di caching

## Gli obiettivi di HTTP 1.1

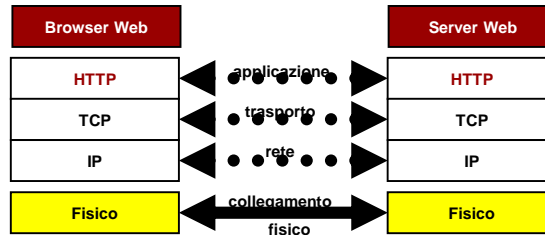
12

- Nuovi concetti:
  - ▣ Meccanismo hop-by-hop
  - ▣ Ottimizzazione della banda
  - ▣ **Gestione delle connessioni**
  - ▣ Trasmissione del messaggio in chunk
  - ▣ Virtual hosting
  - ▣ Supporto per le varianti di una risorsa
  - ▣ Autenticazione crittografata
  - ▣ Miglioramento dei meccanismi di caching

## Gestione delle connessioni

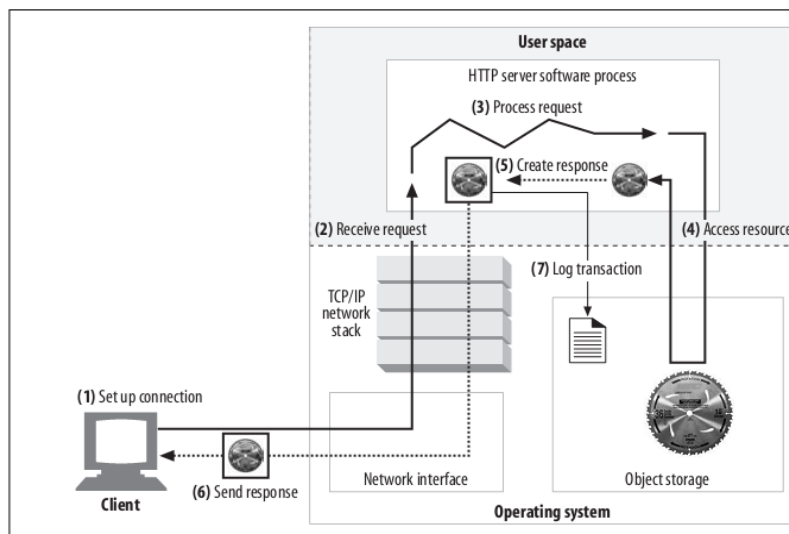
13

- Poiché HTTP è stratificato su TCP, le performance delle transazioni HTTP dipendono dalle performance di TCP



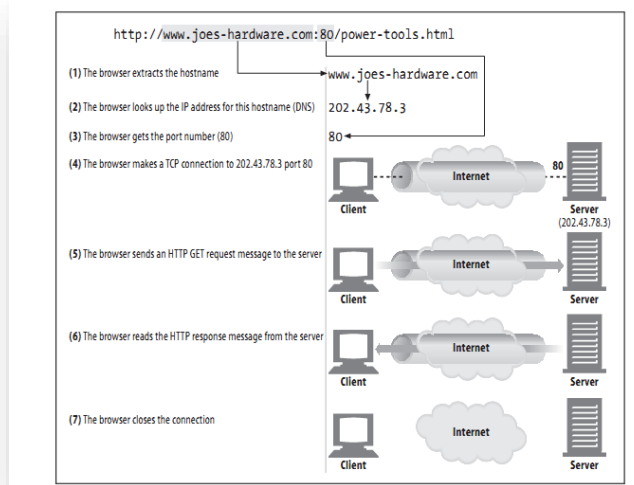
## Gestione delle connessioni

14



## Gestione delle connessioni

15



## Gestione delle connessioni

16

- I ritardi possono dipendere da:
  - ▣ Caratteristiche hardware
  - ▣ Carico della rete
  - ▣ Carico del Web server
  - ▣ Taglia dei messaggi di richiesta e di risposta
  - ▣ Distanza fra client e server
  - ▣ Dettagli puramente **intrinseci del protocollo TCP**



## Gestione delle connessioni

17

- Fattori da analizzare nel progetto di *high-performance HTTP software*
  - ▣ TCP connection setup handshake
  - ▣ TCP slow-start congestion control
  - ▣ Nagle's algorithm for data aggregation
  - ▣ TCP's delayed acknowledgment algorithm for piggybacked acknowledgments
  - ▣ TIME\_WAIT delays and port exhaustion

## Gestione delle connessioni

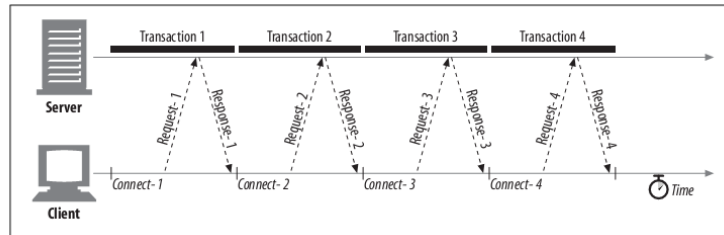
18

- Il problema: HTTP usa praticamente esclusivamente TCP come protocollo di trasporto
  - ▣ che non è ottimizzato per le tipiche connessioni di HTTP frequenti, di breve durata, con pochi dati da scambiare
- Il three-way handshake di TCP e i 4 pacchetti di chiusura
  - ▣ portano una richiesta HTTP (tipicamente in 10 pacchetti) a diventare di 17 pacchetti (41% di overhead!!)
- L'idea: sfruttare il fatto che un documento HTML è spesso un *contenitore* di altri documenti
  - ▣ riferiti da URI

## Gestione delle connessioni

19

- Pagina Web con tre embedded images



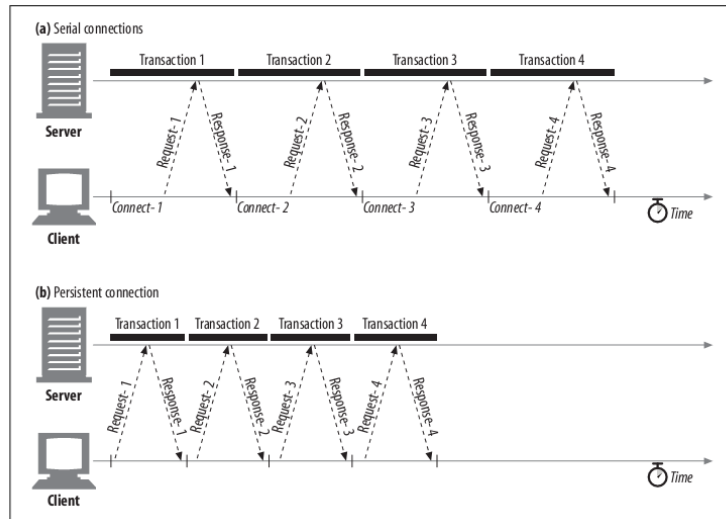
## Connessioni persistenti

20

- Possibilità di trasferire coppie multiple di richiesta e risposta in una stessa connessione TCP
- Vantaggi
  - ▣ Riduzione costi delle connessioni TCP
  - ▣ Riduzione della latenza
    - Aumenta la dimensione della finestra di congestione di TCP
  - ▣ Svantaggi
    - Quando chiudere le connessioni?

## Connessioni seriali vs persistenti

21



## Pipelining

22

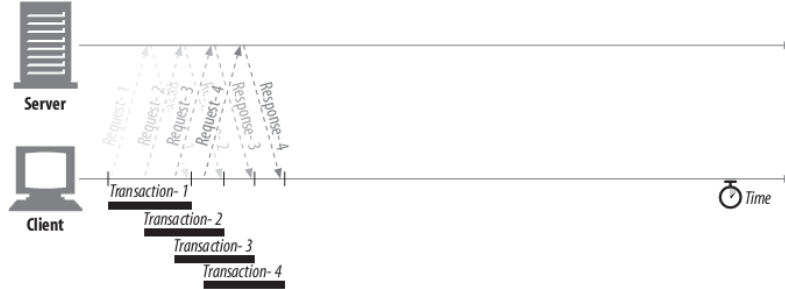
- Trasmissione di più richieste senza attendere l'arrivo della risposta alle richieste precedenti
- Le risposte devono essere fornite dal server nello stesso ordine in cui sono state fatte le richieste
- Vantaggi
  - ▣ Ulteriore riduzione della latenza e ottimizzazione del traffico di rete

## Pipelining

23

- Web page con tre embedded images

(c) Pipelined, persistent connection



## La chiusura delle connessioni persistenti

24

- Header generale usato per segnalare la chiusura della connessione persistente: **Connection: close**
- Il server può chiudere la connessione
  - ▣ Allo scadere di un timeout, applicato sul tempo totale di connessione o sul tempo di inattività di una connessione
  - ▣ Allo scadere di un numero massimo di richieste da servire sulla stessa connessione