Corso di Laurea in Ingegneria Informatica

Corso di Reti di Calcolatori
(a.a. 2011/12)

Roberto Canonico (roberto.canonico@unina.it)
Giorgio Ventre (giorgio.ventre@unina.it)

Protocolli applicativi: HTTP

Seconda parte

4 ottobre 2011

I lucidi presentati al corso sono uno strumento didattico
che NON sostituisce i testi indicati nel programma del corso

Nota di copyright per le slide COMICS

Nota di Copyright

Connessioni persistenti e non persistenti

non persistente

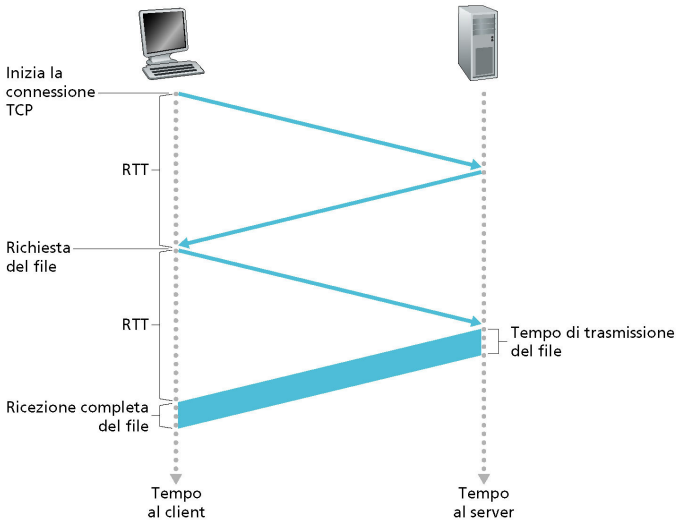
- HTTP/1.0 (RFC 1945)
- Il server analizza una richiesta, la serve e chiude la connessione
- 2 Round Trip Time (RTT) per ciascuna richiesta
- Ogni richiesta subisce lo slow-start TCP

persistente

- HTTP/1.1 (RFC 2068, RFC 2116)
- Sulla stessa connessione il server analizza tutte le richieste e le serve
- Il client riceve la pagina iniziale e invia subito tutte le altre richieste
- Si hanno meno RTT ed un solo slow-start
- Esiste anche una versione con parallelismo (with pipelining)

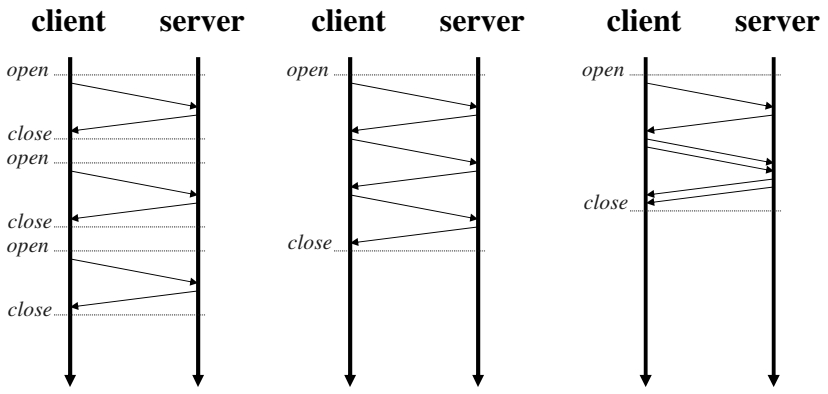
3

Round Trip Time e connessioni HTTP



4

La connessione HTTP



Non persistente

Persistente

Persistente con pipelining

5

Gli header generali

- Gli header generali si applicano solo al messaggio trasmesso e si applicano sia ad una richiesta che ad una risposta, ma non necessariamente alla risorsa trasmessa
- **Date**: data ed ora della trasmissione
- **MIME-Version**: la versione MIME usata per la trasmissione (sempre 1.0)
- **Transfer-Encoding**: il tipo di formato di codifica usato per la trasmissione
- **Cache-Control**: il tipo di meccanismo di caching richiesto o suggerito per la risorsa
- **Connection**: il tipo di connessione da usare
  - Connection: Keep-Alive → tenere attiva dopo la risposta
  - Connection: Close → chiudere dopo la risposta
- **Via**: usato da proxy e gateway

* MIME, Multipurpose Internet Mail Extensions - RFC 2045-2056

6

Gli header di risposta

- Gli header della risposta sono posti dal server per specificare informazioni sulla risposta e su se stesso al client
  - **Server**: una stringa che descrive il server: tipo, sistema operativo e versione
  - **Accept-ranges**: specifica che tipo di range può accettare (valori previsti: byte e none)

7

Gli header dell'entità

- Gli header dell'entità danno informazioni sul body del messaggio, o, se non vi è body, sulla risorsa specificata
- **Content-Type**: oggetto/formato
  - Ogni coppia oggetto/formato costituisce un tipo MIME dell'entità acclusa
  - Specifica se è un testo, se un'immagine GIF, un'immagine JPG, un suono WAV, un filmato MPG, ecc…
  - Obbligatorio in ogni messaggio che abbia un body
- **Content-Length**: la lunghezza in byte del body
  - Obbligatorio, soprattutto se la connessione è persistente
- **Content-Base, Content-Encoding, Content-Language, Content-Location, Content-MD5, Content-Range**: l'URL di base, la codifica, il linguaggio, l'URL della risorsa specifica, il valore di digest MD5 e il range richiesto della risorsa
- **Expires**: una data dopo la quale la risorsa è considerata non più valida (e quindi va richiesta o cancellata dalla cache)
- **Last-Modified**: la data e l'ora dell'ultima modifica
  - Serve per decidere se la copia posseduta (es. in cache) è ancora valida o no
  - Obbligatorio se possibile

8

Una prova con telnet

> telnet www.unina.it 80
GET / HTTP/1.0

Request-line

HTTP/1.1 200 OK
Date: Mon, 25 Oct 2010 23:02:38 GMT
Server: Apache/2.2.14 (Unix) mod_jk/1.2.28 DAV/2
ETag: W/"2097-1213082454000"
Last-Modified: Tue, 10 Jun 2008 07:20:54 GMT
Content-Length: 2097
Connection: close
Content-Type: text/html

Response

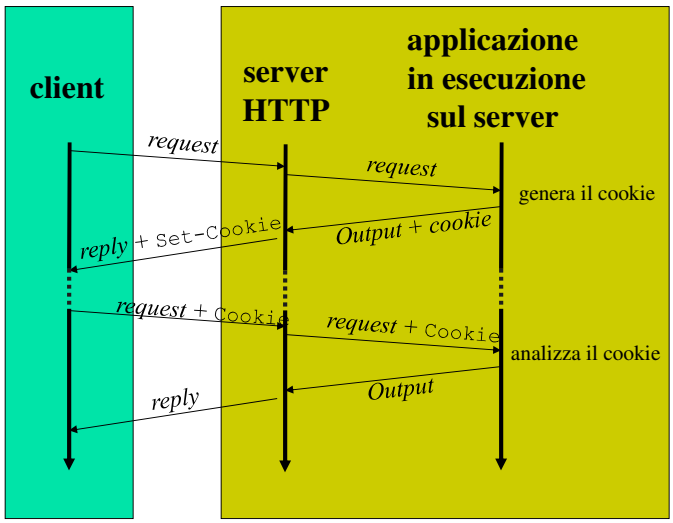
9

I cookies

- **HTTP è stateless**: il server non è tenuto a mantenere informazioni su connessioni precedenti
- Un cookie è una breve informazione scambiata tra il server ed il client
- Tramite un cookie il client mantiene lo stato di precedenti connessioni, e lo manda al server di pertinenza ogni volta che richiede un documento
- Esempio: tramite un cookie viene riproposta la propria username all'atto dell'accesso ad un sito per la posta
- I cookies sono definiti in RFC2109(su proposta di Netscape)

10

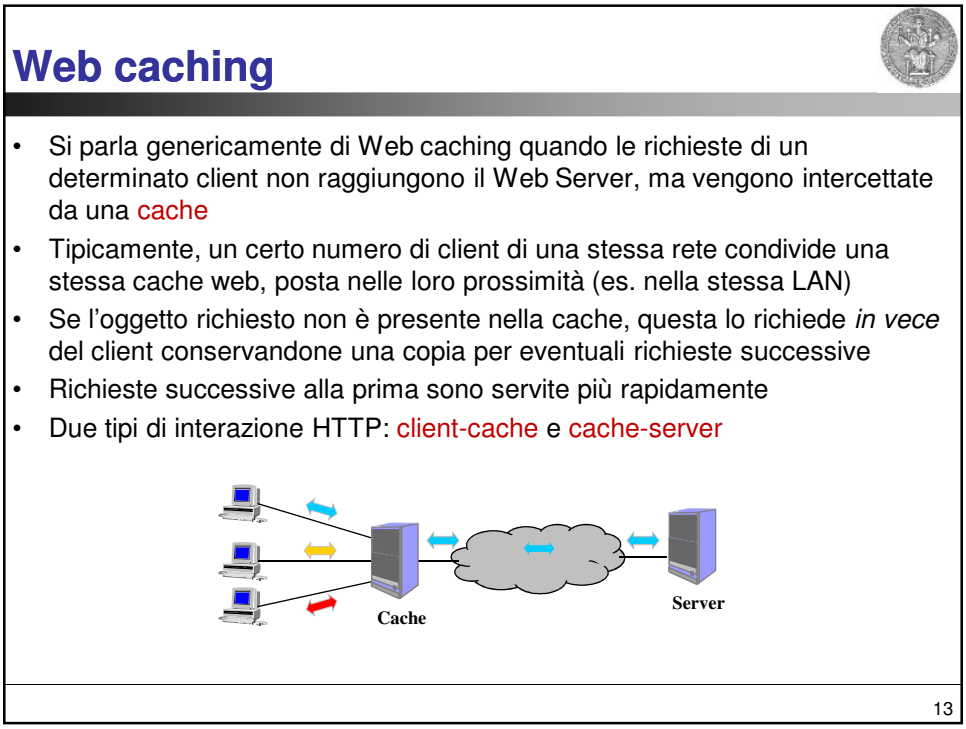
Cookies (2)



Cookies: header specifici

- Il meccanismo dei cookies dunque definisce due nuovi possibili header: uno per la risposta, ed uno per le richieste successive
  - **Set-Cookie**: header della risposta
    - il client può memorizzarlo (se vuole) e rispedirlo alla prossima richiesta
  - **Cookie**: header della richiesta
    - il client decide se spedirlo sulla base del nome del documento, dell'indirizzo IP del server, e dell'età del cookie
- Un browser può essere configurato per accettare o rifiutare i cookies
- Alcuni siti web richiedono necessariamente la capacità del browser di accettare i cookies

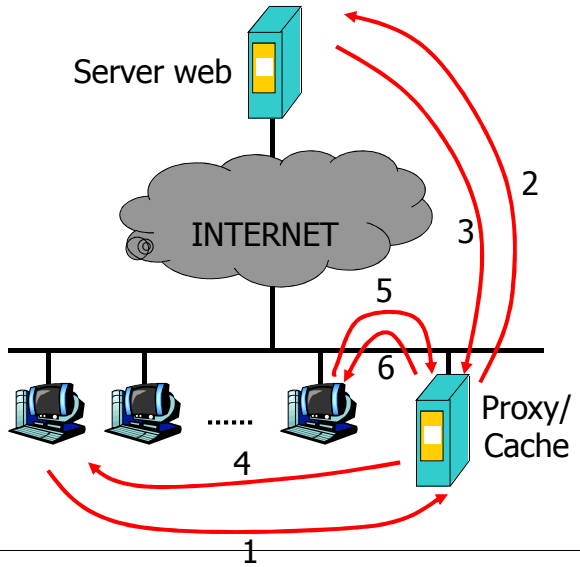
Web caching



- Si parla genericamente di Web caching quando le richieste di un determinato client non raggiungono il Web Server, ma vengono intercettate da una cache
- Tipicamente, un certo numero di client di una stessa rete condivide una stessa cache web, posta nelle loro prossimità (es. nella stessa LAN)
- Se l'oggetto richiesto non è presente nella cache, questa lo richiede *in vece* del client conservandone una copia per eventuali richieste successive
- Richieste successive alla prima sono servite più rapidamente
- Due tipi di interazione HTTP: client-cache e cache-server

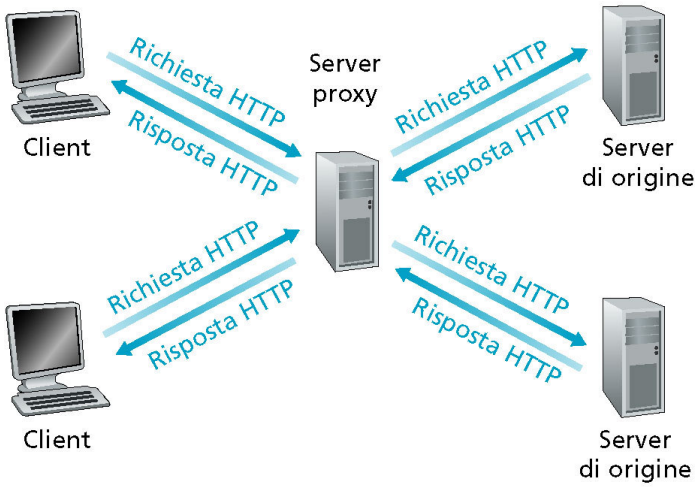
13

Transazioni web con caching



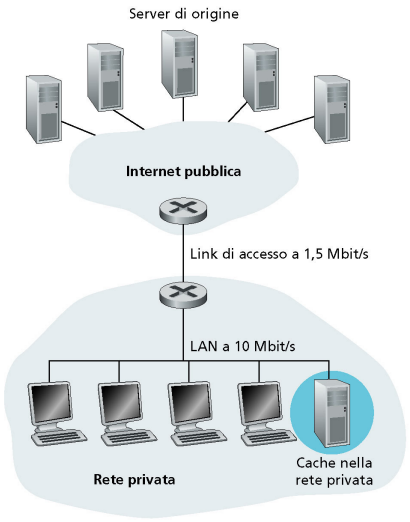
14

Server proxy: schema logico



15

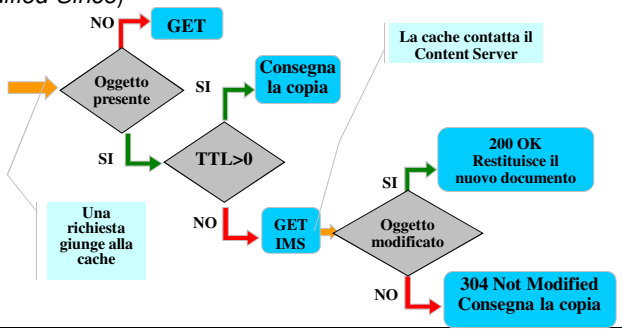
Server proxy in una rete di accesso



16

Gestione della coerenza

- Problema: cosa succede se l'oggetto presente nel server è aggiornato ?
- La copia in cache deve essere aggiornata per mantenersi uguale all'originale
- HTTP fornisce due meccanismi per la gestione della coerenza:
  - TTL (Time To Live) : il server quando fornisce un oggetto dice anche quando quell'oggetto "scade" (header *Expires*)
    - Quando TTL diventa < 0, non è detto in realtà che l'oggetto sia stato realmente modificato
  - Il client può fare un ulteriore controllo mediante una GET condizionale (*If-Modified-Since*)



Non solo browsing…

- HTTP non è utilizzato solo per il Web
- Ad esempio:
  - XML e VoiceXML trasportati su HTTP
  - Web Services e SOA (Service Oriented Architecture)
  - Video streaming
  - Peer-to-peer