Azzolini Riccardo 2020-11-17

HTTP e FTP

1 World Wide Web

Il **World Wide Web** (**WWW**) è una collezione distribuita di documenti *iper-testuali* e *iper-mediali*, cioè documenti multimediali tra loro collegati, che permettono una lettura dinamica, non lineare, "a n dimensioni".

Il contenuto di un documento accessibile sul Web può cambiare nel tempo. In particolare, da questo punto di vista, un documento può essere:

statico: non muta nel tempo;

dinamico: viene creato dal server al momento della richiesta;

attivo: viene eseguito presso il client.

2 HTML

HTML, **HyperText Markup Language**, è un linguaggio di *markup* per la creazione di pagine Web. Esso fornisce appositi **tag**, etichette, che permettono di definire i contenuti di un documento iper-testuale o iper-mediale.

La struttura di base di una pagina HTML è mostrata nel seguente esempio:

```
<html>
  <head>
    <title>Esempio</title>
  </head>
  <body>
    Contenuto dell'esempio...
  </body>
</html>
```

3 Browser

Il **browser** è l'applicazione che interpreta e visualizza una pagina Web. Esso è costituito da tre moduli:

- il **controller**, che seleziona i protocolli applicativi necessari per accedere ai contenuti delle pagine Web richieste in base agli input dell'utente;
- i **protocolli applicativi**;
- l'**interprete**, che interpreta il contenuto delle pagine Web e lo rappresenta a video.

4 Uniform Resource Locator

Ogni documento accessibile sul Web è identificato da un **URL**, **Uniform Resource Locator**, che è composto da quattro parti:

$$\textit{protocollo}\texttt{://}\textit{host}\texttt{:}\textit{porta}\texttt{/}\textit{path}$$

1. il *protocollo applicativo*;
2. l'*host* remoto su cui risiede il documento;
3. il *numero di porta* su cui è in ascolto il protocollo applicativo remoto (se omesso, viene usata di default la porta well-known corrispondente al protocollo applicativo specificato);
4. il *path* (percorso) della pagina nel filesystem remoto.

5 HyperText Transfer Protocol

Il protocollo usato per lo scambio di pagine Web è **HTTP**, **HyperText Transfer Protocol**. Esso usa il protocollo di trasporto TCP, poiché nel caso del Web l'affidabilità è più importante rispetto alla banda e ai ritardi (non è un'applicazione real-time).

L'interazione tra client e server avviene tramite messaggi di richiesta e risposta:

1. il client invia al server una **HTTP request**, con la quale richiede un oggetto (documento HTML, immagine, ecc.) residente sul server;
2. in risposta, il server manda al client una **HTTP response** contenente l'oggetto richiesto.

Il server non mantiene informazioni riguardanti la storia della sessione, cioè tratta ogni richiesta in modo indipendente dalle altre: perciò, si dice che HTTP è un protocollo **stateless**. Il vantaggio è che, non dovendo memorizzare informazioni sui client connessi, il server può gestire molte connessioni contemporaneamente.

HTTP supporta diversi tipi di connessione:

- **non persistente**: viene creata una nuova connessione TCP per ogni oggetto trasferito;
- **persistente**: una singola connessione TCP può essere usata per trasferire più oggetti, e ciò può avvenire:
  - **senza parallelismo**: si invia una nuova richiesta solo dopo aver ricevuto la risposta alla richiesta precedente;
  - **con parallelismo**: si possono inviare più richieste alla volta.

Tra questi, il tipo di connessione che permette i minori ritardi è quello persistente con parallelismo.

5.1 HTTP request

I campi più importanti presenti nella richiesta HTTP sono l'*URL*, che identifica un oggetto sul server, e il **metodo**, che indica l'azione che il client vuole compiere su tale oggetto. Oltre a questi, sono presenti anche dei campi, chiamati *header*, che specificano varie informazioni sulla connessione e sul client. Infine, il client può specificare dei dati da inviare al server (ad esempio i contenuti di un nuovo oggetto da caricare sul server).

I principali metodi sono:

GET: richiede il contenuto di un oggetto;

HEAD: richiede solo le informazioni (header) relative a un oggetto, ma non il suo contenuto (questo metodo viene usato ad esempio per il debugging);

POST: aggiunge un contenuto in coda a una pagina;

PUT: aggiunge o aggiorna un oggetto;

DELETE: elimina un oggetto.

5.2 HTTP response

Nella risposta HTTP è presente uno **status code** numerico, che indica l'esito della richiesta (ad esempio, 200 significa "OK", mentre 404 indica "Not found", cioè che l'oggetto richiesto non è stata trovato). Ci sono poi una serie di *header* contenenti informazioni sulla connessione, sul server e sull'oggetto restituito. Infine, c'è il campo contenente i dati applicativi veri e propri, che prende il nome di *entity body*: se non ci sono stati errori, esso contiene l'oggetto richiesto dal client.

5.3 Cookie technology

Siccome HTTP è un protocollo stateless, per permettere al server di tenere traccia del comportamento dell'utente è stato necessario introdurre la **cookie technology**, un meccanismo mediante il quale l'utente si identifica ogni volta che fa una richiesta al server.

Al momento del primo accesso a un sito, il server crea nel proprio database un identificativo univoco associato all'utente, chiamato **UID**, e lo comunica al client, che lo scrive in un **cookie file** gestito dal browser. Successivamente, durante la consultazione del sito, il client comunica ogni volta al server il proprio UID (tramite un apposito header inserito in ciascuna richiesta), e così il server può tenere traccia dell'attività dell'utente.

6 File Transfer Protocol

FTP, **File Transfer Protocol**, permette di accedere a un filesystem remoto (di un server) e scambiare file con il filesystem locale (della propria macchina). Esso usa due connessioni TCP:

- una **connessione di controllo**, per trasferire informazioni di controllo tra client e server;
- una **connessione dati**, per trasferire un file tra client e server, o inviare al client un elenco di file/directory presenti sul server.

La connessione dati *non è persistente*: ne viene creata una nuova per ogni file trasferito. Alle connessioni di controllo e dati FTP sono assegnati rispettivamente i numeri di porta well-known 20 e 21.

FTP richiede l'**autenticazione** dell'utente remoto, che deve possedere un account sul server. Siccome le connessioni TCP devono essere associata all'account dell'utente, FTP *non è stateless*, quindi un server è in grado di gestire contemporaneamente solo un numero limitato di utenti.

6.1 Rappresentazione dei dati

Il trasferimento di un file può avvenire in una di tre diverse modalità:

- **stream mode** (quella usata di default): il file è trasferito come un flusso di byte contigui;
- **block mode**: il file è trasferito come flusso di blocchi, ciascuno dei quali è preceduto da un'intestazione;
- **compressed mode**: il file è trasferito in forma compressa.

6.2 Comandi e risposte

Per interagire con il server, il client invia sulla connessione di controllo una serie di comandi, ciascuno dei quali ha una particolare funzione. Ad esempio, ci sono comandi per l'autenticazione, la navigazione del filesystem remoto, l'apertura della connessione dati, ecc.

Il server risponde ai comandi del client con dei codici numerici.

6.3 Creazione della connessione dati

La creazione della connessione dati è sotto il controllo del client:

1. il client esegue una **open passiva**, mettendosi in ascolto su un numero di porta a propria scelta;
2. il client comunica al server il numero di porta scelto, tramite l'apposito comando `PORT` sulla connessione di controllo;
3. il server esegue una **open attiva**, stabilendo una connessione tra la propria porta 20 e la porta del client ricevuta con il comando `PORT`.