

### Esercizio 8.1

Usando Wireshark, si osservi e si analizzi il traffico HTTP e DNS scambiato da un browser Web con un server DNS e un server Web per il recupero di una pagina html.

### Esercizio 8.2

Si implementi in Java un server Web (http) sequenziale. Il server dovrà consentire ad un browser esistente di recuperare (usando il metodo GET) file di tipo testo (plain o HTML) e file contenenti immagini, attraverso messaggi di richiesta e di risposta HTTP, veicolati mediante lo strato di trasporto fornito dal protocollo TCP. Questi file dovranno essere resi disponibili al server Web collocandoli in una directory designata sulla macchina su cui il server è in esecuzione. Allo scopo, il server deve essere configurato mediante le seguenti opzioni, fornite attraverso un file di configurazione dal nome *webserver.conf*:

#### **port NNN**

NNN è una stringa ASCII che rappresenta il numero di porta TCP su cui il server si pone in ascolto. Se questa opzione non è specificata, sarà usato il numero di porta di default 80.

#### **root /path**

/path è il percorso che indica la posizione della radice del file system gestito dal server rispetto all'intero file system del sistema su cui il server è in esecuzione. Se tale opzione non viene specificata, si assume come default la directory corrente.

Si ricorda che il formato di un messaggio HTTP di richiesta è:

```
Request = Request-Line
          *(( general-header
             | request-header
             | entity-header ) CRLF)
          CRLF
          [ message-body ]
```

La request line è così organizzata:

```
Request-Line = Method SP Request-URI SP HTTP-Version CRLF
```

Il formato di un messaggio HTTP di risposta è:

```
Response = Status-Line
          *(( general-header
             | response-header
             | entity-header ) CRLF)
          CRLF
          [ message-body ]
```

La status line è così organizzata:

```
Status-Line = HTTP-Version SP Status-Code SP Reason-Phrase CRLF
```

**Suggerimenti per l'implementazione**

Si trascuri inizialmente la lettura del contenuto del file di configurazione e si considerino i valori di default per il numero di porta e per la directory radice.

Si implementi una prima versione semplificata del server Web che, ricevuto il messaggio di richiesta dal client, provveda a stampare su standard output il contenuto del messaggio.

Si proceda quindi con l'implementazione dell'elaborazione del messaggio di richiesta per produrre il relativo messaggio di risposta.

Si consideri inizialmente un'implementazione basata su connessioni non permanenti chiudendo la connessione alla fine del messaggio di risposta. Si usi allo scopo la linea di intestazione "Connection: close" per segnalare lo specifico comportamento al browser.

**Materiale di riferimento**

Come per la maggior parte dei protocolli per Internet, le specifiche di HTTP sono ben descritte nei "Request for Comments": HTTP/1.0 è definito dal RFC 1945 mentre HTTP/1.1 è definito dal RFC 2616. Gli RFC possono essere scaricati dal sito [www.rfc-editor.org](http://www.rfc-editor.org)