

Prova in itinere – Esempio d’esame 1

Cognome	
Nome	
Matricola	

Tempo complessivo a disposizione per lo svolgimento: 1h30m

Usare lo spazio dopo ogni Esercizio/Quesito per la risposta.

Es1 (9pt)	Es2 (9 pt)	Ques (9 pt)	Lab (6pt)

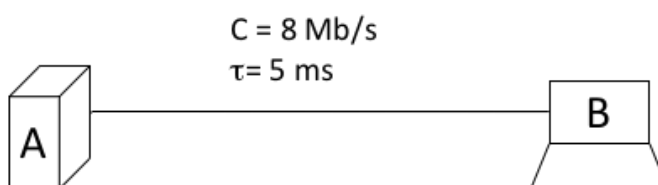
1 - Esercizio (9 punti)

Una connessione TCP tra l’host A e l’host B è caratterizzata dai seguenti parametri:

- Lunghezze di *header* e *ack* trascurabili;
- *link* bidirezionali;
- RCWND = 16 MSS
- SSTHRESH = 8 MSS
- MSS = 1 kB
- Ritardo di propagazione, $\tau = 5$ ms
- Valore iniziale del *Time-Out* = 3 s

Si risponda ai seguenti quesiti:

- a) Dire se la trasmissione sul link diventa mai continua; in caso affermativo, trovare il tempo oltre cui la trasmissione sul link diventa continua;
- b) Trovare il tempo di trasferimento di un file di 80 kB da A a B;
- c) Ipotizzando che l’ultimo segmento in trasmissione venga perso, ripetere il punto b)



Soluzione

$$T = \frac{MSS}{C} = \frac{8 \cdot 10^3}{8 \cdot 10^6} = 1 \text{ ms}$$

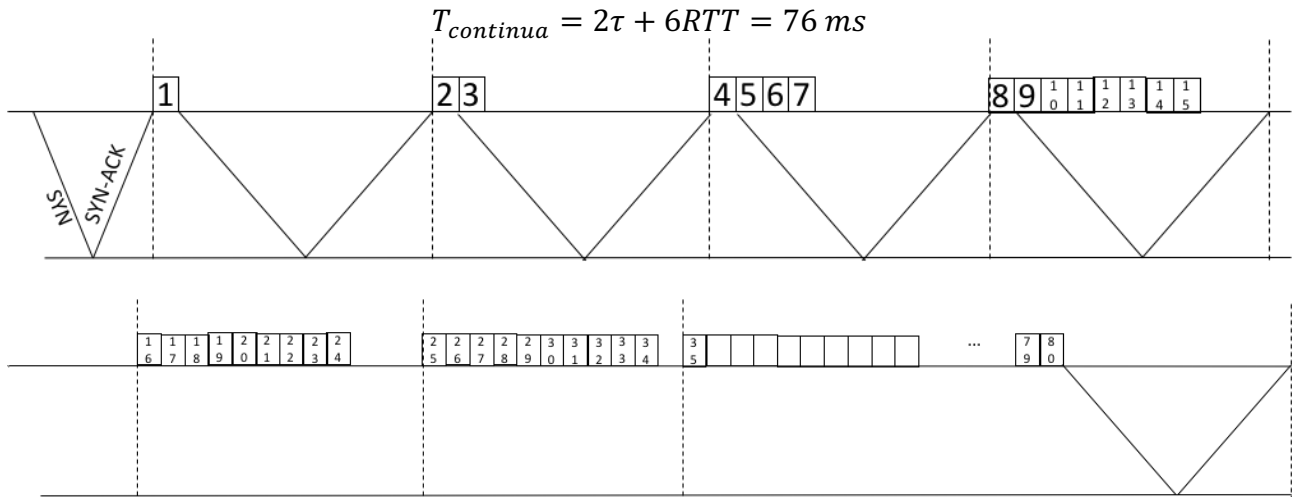
$$RTT = T + 2\tau = 11 \text{ ms}$$

$$\text{Finestra tx continua: } W_c = \frac{RTT}{T} = 11 \text{ MSS}$$

a)

- Apertura connessione: 2τ
- Slow start: (1) – (2) – (4) – (8): $4RTT$
- Congestion avoidance: (9) – (10): $2RTT$

Poi trasmissione continua



b)

File = 80 MSS

- Apertura connessione: 2τ
- Slow start: (1) – (2) – (4) – (8): $4RTT$
- Congestion avoidance: (9) – (10): $2RTT$

In totale 34 segmenti

- Trasmissione continua dei restanti 46 segmenti: $46T + 2\tau$

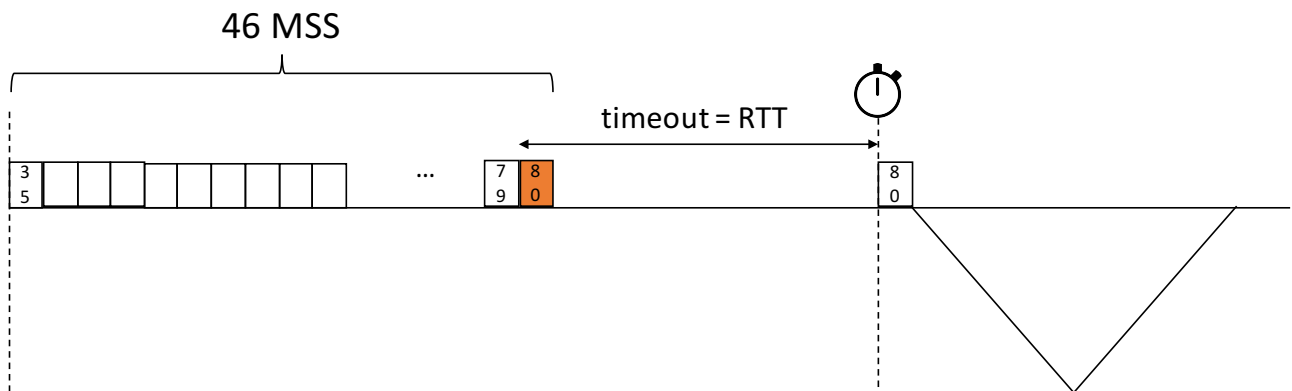
Tempo totale:

$$T_{tot} = 2\tau + 6RTT + 46T + 2\tau = 132\text{ ms}$$

c)

L'ultimo segmento viene ritrasmesso dopo lo scadere del timeout. Il timeout viene settato sulla base dell'algorithmo di stima del RTT che essendo costante porta ad avere:

$$T_{out} = RTT$$

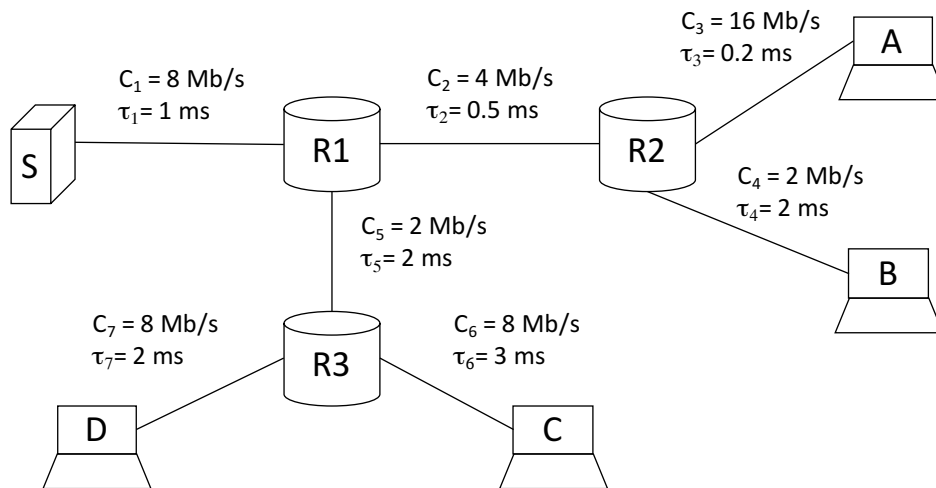


e quindi

$$T_{tot} = 2\tau + 6RTT + 45T + T_{out} + T + 2\tau = 143\text{ ms}$$

2 - Esercizio (9 punti)

- a) In una rete a commutazione di pacchetto al tempo $t=0$ sono presenti 8 pacchetti in S diretti rispettivamente alle seguenti destinazioni: A, A, B, A, C, C, D, D. Calcolare il tempo di ricezione di ciascuno dei pacchetti assumendo che i pacchetti abbiano le seguenti dimensioni: pacchetti verso A, $L_A=1000B$; pacchetti verso B, $L_B=2000B$; pacchetti verso C, $L_C=500B$; pacchetti verso D, $L_D=1000B$.
- b) Si assuma che un client http in A voglia scaricare una pagina web contenuta del server in S. La capacità del collegamento tra S ed A è limitata dal collegamento che costituisce il collo di bottiglia della rete, e che è condiviso con due flussi interferenti di lunga durata (file transfer) tra D e B e tra C e B. La pagina web è composta da un documento base (html) di 100B e da 8 immagini di 1MB. Si calcoli il tempo di scaricamento della pagina web nel caso di connessione http persistente per il documento base e le immagini, e nel caso di connessione non persistente (prima il documento html e poi le 8 immagini con connessioni in parallelo).



Soluzione

a)

$$T_1^A = \frac{L_A}{C_1} = \frac{8 \cdot 10^3}{8 \cdot 10^6} = 1 \text{ ms}$$

$$T_2^A = \frac{L_A}{C_2} = 2 \text{ ms}$$

$$T_3^A = \frac{L_A}{C_3} = 0.5 \text{ ms}$$

$$T_1^B = \frac{L_B}{C_1} = 2 \text{ ms}$$

$$T_2^B = \frac{L_B}{C_2} = 4 \text{ ms}$$

$$T_4^B = \frac{L_B}{C_4} = 8 \text{ ms}$$

$$T_1^C = \frac{L_C}{C_1} = 0.5 \text{ ms}$$

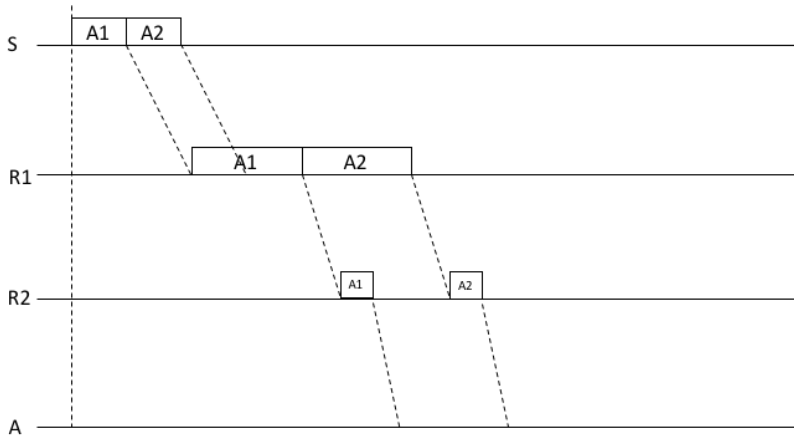
$$T_5^C = \frac{L_C}{C_5} = 2 \text{ ms}$$

$$T_6^C = \frac{L_C}{C_6} = 0.5 \text{ ms}$$

$$T_1^D = \frac{L_D}{C_1} = 1 \text{ ms}$$

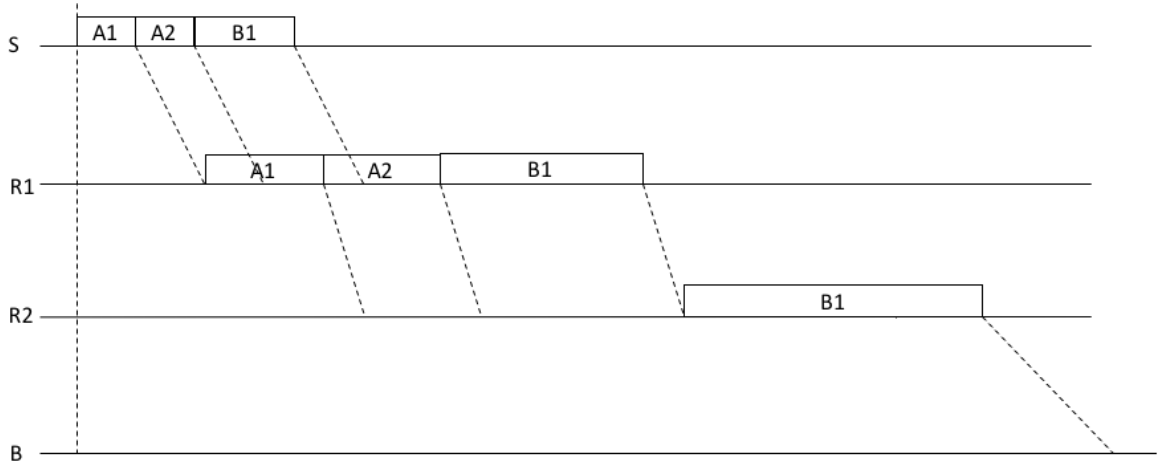
$$T_5^D = \frac{L_D}{C_5} = 4 \text{ ms}$$

$$T_7^D = \frac{L_{CD}}{C_7} = 1 \text{ ms}$$

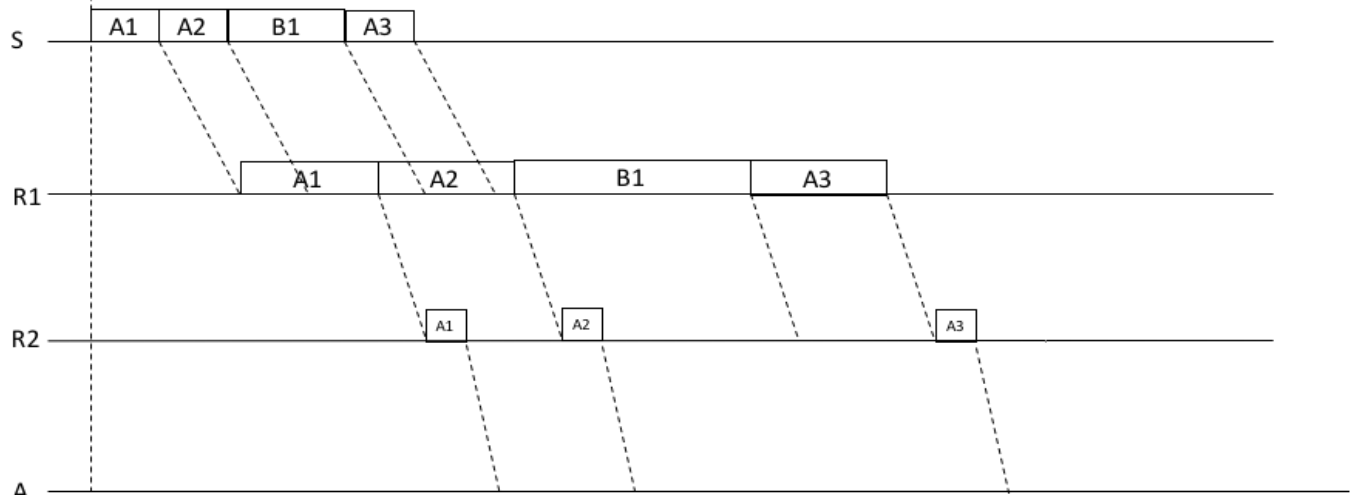


$$T_{A1} = T_1^A + \tau_1 + T_2^A + \tau_2 + T_3^A + \tau_3 = 5.2 \text{ ms}$$

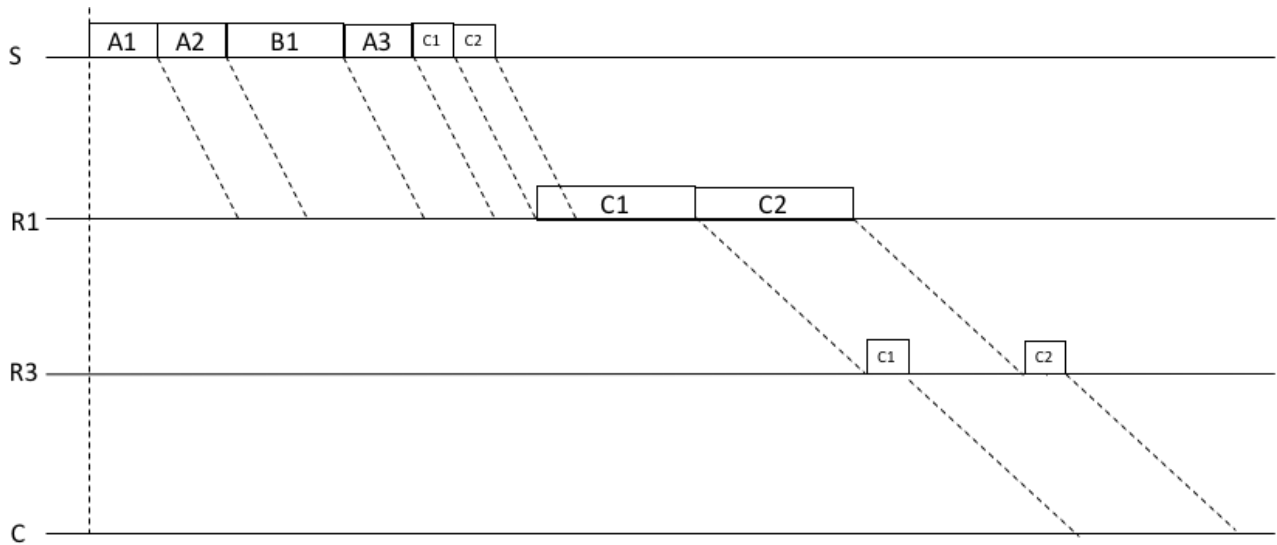
$$T_{A2} = T_{A1} + T_2^A = 7.2 \text{ ms}$$



$$T_{B1} = T_1^A + \tau_1 + 2T_2^A + T_2^B + \tau_2 + T_4^B + \tau_4 = 1 + 1 + 2 \times 2 + 4 + 0.5 + 8 + 2 = 20.5 \text{ ms}$$

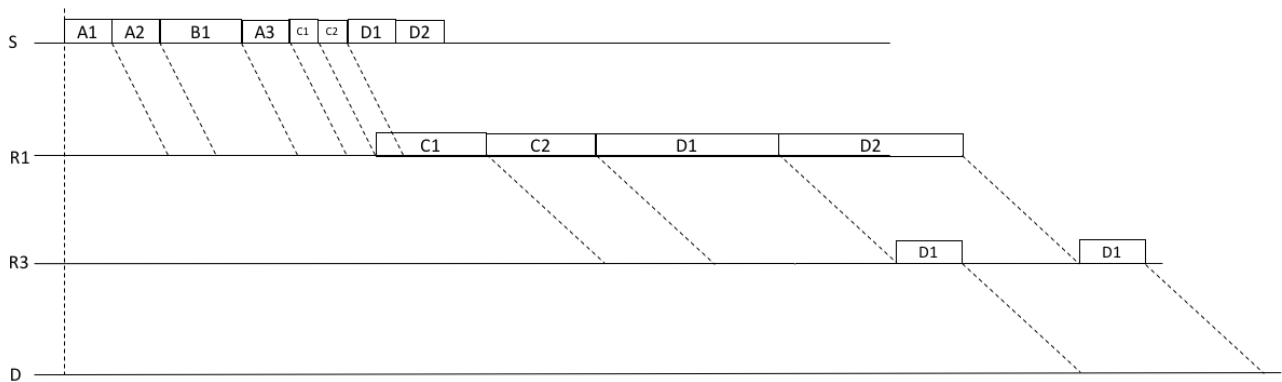


$$T_{A3} = T_1^A + \tau_1 + 3T_2^A + T_2^B + \tau_2 + T_3^A + \tau_3 = 1 + 1 + 3 \times 2 + 4 + 0.5 + 0.5 + 0.2 = 13.2 \text{ ms}$$



$$T_{C1} = 3 T_1^A + T_1^B + T_1^C + \tau_1 + T_5^C + \tau_5 + T_6^C + \tau_6 = 14 \text{ ms}$$

$$T_{C2} = T_{C1} + T_5^C = 16 \text{ ms}$$



$$T_{D1} = 3 T_1^A + T_1^B + T_1^C + \tau_1 + 2T_5^C + T_5^D + \tau_5 + T_7^D + \tau_7 = 19.5 \text{ ms}$$

$$T_{D1} = T_{D1} + T_5^D = 23.5 \text{ ms}$$

b)

Nel caso di una connessione persistente la condivisione equa tra i due flussi interferenti e il flusso http tra S e A porta quest'ultimo ad un rate $R=2$ Mb/s.

$$RTT = 2(\tau_1 + \tau_2 + \tau_3) = 3.4 \text{ ms}$$

$$T_{html} = \frac{100 * 8 \text{ bit}}{2 \text{ Mb/s}} = 0.4 \text{ ms}$$

$$T_{obj} = \frac{8 * 10^6 \text{ bit}}{2 \text{ Mb/s}} = 4 \text{ s}$$

$$T_{tot} = RTT + RTT + T_{html} + 8(RTT + T_{obj}) = 32.0344 \text{ s}$$

Nel caso di connessione non persistente e oggetti in parallelo, la parte del file html non cambia, per gli oggetti invece il rate di condivisione equa risulta 0.4 Mb/s (4Mb/s condivisi da 10 flussi).

$$T_{obj} = \frac{8 * 10^6 \text{ bit}}{0.4 \text{ Mb/s}} = 20 \text{ s}$$

$$T_{tot} = 2RTT + T_{html} + 2RTT + T_{obj} = 20.014 \text{ s}$$

Quesiti (9 punti)

Q1

Come viene settato il TIMEOUT in TCP e perché?

$$SRTT^{(i)} = (1-\alpha) SRTT^{(i-1)} + \alpha RTT^{(i)}$$

$$DEV = |RTT^{(i)} - SRTT^{(i-1)}|$$

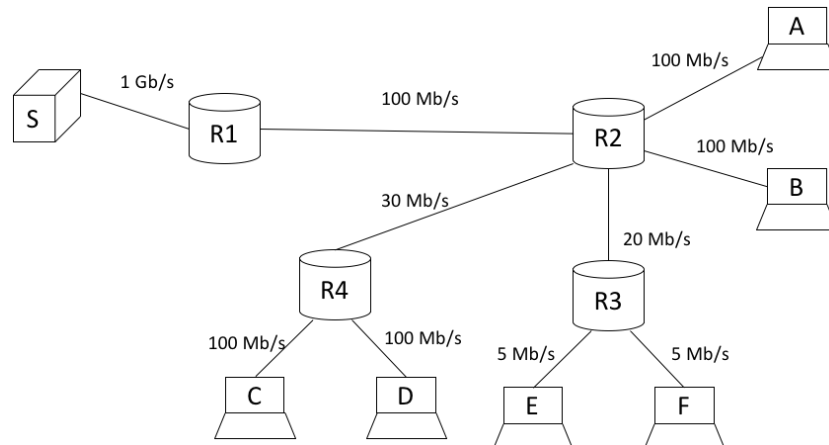
$$SDEV^{(i)} = 3/4 SDEV^{(i-1)} + 1/4 DEV$$

$$TIMEOUT = SRTT + 4 SDEV$$

Il valore del timeout deve tenere conto della variazione del RTT in base alle condizioni della rete, da una parte, per poter individuare il prima possibile un segmento perso, dall'altra, per non essere troppo breve ed impedire il funzionamento corretto di TCP.

Q2

Si consideri la rete sotto e si assuma siano presenti contemporaneamente 2 trasferimenti file con TCP da ciascun host (A, B, C, D, E, F) verso il server S. Assumendo condivisione equa delle risorse, qual è la velocità di ciascuna connessione?



S-E: 2.5 Mb/s

S-F: 2.5 Mb/s

S-C: 7.5 Mb/s

S-D: 7.5 Mb/s

S-A: 15 Mb/s

S-B: 15 Mb/s

Q3

Si consideri la risposta al comando dig riportata sotto.

- Secondo te, perché sono presenti più nomi DNS nei record MX del dominio fastwebnet.it e più indirizzi IP per mx1.fastwebnet.it? Cosa ne farà chi legge questa risposta?
- a quale server (nome e indirizzo IP) viene inviata una mail con destinatario mioamico@fastwebnet.i?
- che ruolo svolge il server dns2.fastweb.it?

```
;<>> DiG 9.8.3-P1 <>> fastwebnet.it any ANY
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 14581
;; flags: qr rd ra; QUERY: 1, ANSWER: 7, AUTHORITY: 2, ADDITIONAL: 12

;; QUESTION SECTION:
;fastwebnet.it.          IN      ANY

;; ANSWER SECTION:
fastwebnet.it.          242 IN     SOA    dns1.fastweb.it. dnsmaster.fastweb.it. 2015123000
10800 900 604800 86400
fastwebnet.it.          43394 IN     MX     10 mx2.fastwebnet.it.
fastwebnet.it.          43394 IN     MX     10 mx1.fastwebnet.it.
fastwebnet.it.          43394 IN     MX     10 mx4.fastwebnet.it.
fastwebnet.it.          43394 IN     MX     10 mx3.fastwebnet.it.
fastwebnet.it.          7235 IN     NS     dns2.fastweb.it.
fastwebnet.it.          7235 IN     NS     dns1.fastweb.it.

;; AUTHORITY SECTION:
fastwebnet.it.          7235 IN     NS     dns2.fastweb.it.
fastwebnet.it.          7235 IN     NS     dns1.fastweb.it.

;; ADDITIONAL SECTION:
```

```
mx1.fastwebnet.it. 46232 IN A 85.18.95.26
mx1.fastwebnet.it. 46232 IN A 85.18.95.27
mx1.fastwebnet.it. 46232 IN A 85.18.95.28
mx1.fastwebnet.it. 46232 IN A 85.18.95.15
mx1.fastwebnet.it. 46232 IN A 85.18.95.16
mx1.fastwebnet.it. 46232 IN A 85.18.95.25
mx2.fastwebnet.it. 62714 IN A 85.18.95.15
mx2.fastwebnet.it. 62714 IN A 85.18.95.16
mx2.fastwebnet.it. 62714 IN A 85.18.95.25
mx2.fastwebnet.it. 62714 IN A 85.18.95.26
mx2.fastwebnet.it. 62714 IN A 85.18.95.27
mx2.fastwebnet.it. 62714 IN A 85.18.95.28
```

- a) Si tratta di un meccanismo di bilanciamento di carico tra più server
- b) Ad uno dei server MX, ad esempio `mx2.fastwebnet.it`, `85.18.95.15`
- c) E' un server authoritative per il dominio `fastwebnet.it`

Laboratorio (6 punti)

Q1

Si consideri la seguente sessione HTTP. Si indichi se essa è corretta oppure no, e, in caso negativo, si indichino possibili correzioni.

```
GET /upload/napoli/test.html HTTP/1.1
User-Agent: Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.8.1.3)
Accept: text/html

HTTP/1.1 200 OK
Date: Fri, 27 Ago 2007 04:15:42 GMT
Server: Apache/2.0.52 (Red Hat)
Last-Modified: Fri, 20 Ago 2007 04:15:20 GMT
Accept-Ranges: bytes
Content-Length: 264
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
Content-Type: text/html
<html lang="it">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
  <title>Test Page</title>
</head>
<body>
<h1>Pagina di test</h1>
</body>
</html>

GET /upload/napoli/test.html HTTP/1.1
User-Agent: Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.8.1.3)
Accept: text/html
If-Modified-Since: Thu, 23 Ago 2007 04:15:10 GMT

HTTP/1.1 304 Not Modified
Date: Wed, 27 Ago 2007 04:15:54 GMT
Server: Apache/2.0.52 (Red Hat)
```

Non è corretta perché nei due GET manca l'header Host obbligatorio per HTTP 1.1

Occorre aggiungere ad entrambe i GET la riga Host: <nome DNS del server>

Q2

Si consideri il seguente codice Python

```
from socket import *
sPort = 12000

sSocket = socket(AF_INET, SOCK_DGRAM)
sSocket.bind(('', YYYYYY))

while 1:
    message, cAddress = sSocket.recvfrom(2048)
    if message=='Come ti chiami?':
        sSocket.sendto('Luca', XXXXXX)
    elif message=='Quanti anni hai?':
        sSocket.sendto('20', XXXXXX)
    else:
        sSocket.sendto('Non capisco :(', XXXXXX)
```

Si indichi:

- se il codice implementa un client o un server
- se viene usato TCP o UDP
- cosa inserire al posto di XXXXXX e YYYYYY
- cosa fa l'applicazione implementata

a) Un server
b) UDP
c)
YYYYYY=sPort
XXXXXX=cAddress

d) Risponde a domanda “Come ti chiami?” con “Luca”, a domanda “Quanti anni hai?” con “20”, e a tutte le altre con “Non capisco”.

Q3

Scrivere un server TCP (completando il codice Python sottostante) con connessioni persistente che riceve messaggi e risponde con lo stesso messaggio fino a che non riceve la stringa “FINE” e risponde con “CIAO”.

```
from socket import *

serverPort = 12000
serverSocket = socket(AF_INET, SOCK_STREAM)

serverSocket.bind(('', serverPort))
serverSocket.listen(1)

while True:
    print 'The server is ready to receive'
    connectionSocket, clientAddress = serverSocket.accept()
    print "Connection form: ", clientAddress
    while True:
        sentence = connectionSocket.recv(1024)
```

```
    if sentence == 'FINE':
        connectionSocket.send('CIAO')
        break
    else:
        connectionSocket.send(sentence)
connectionSocket.close()
```