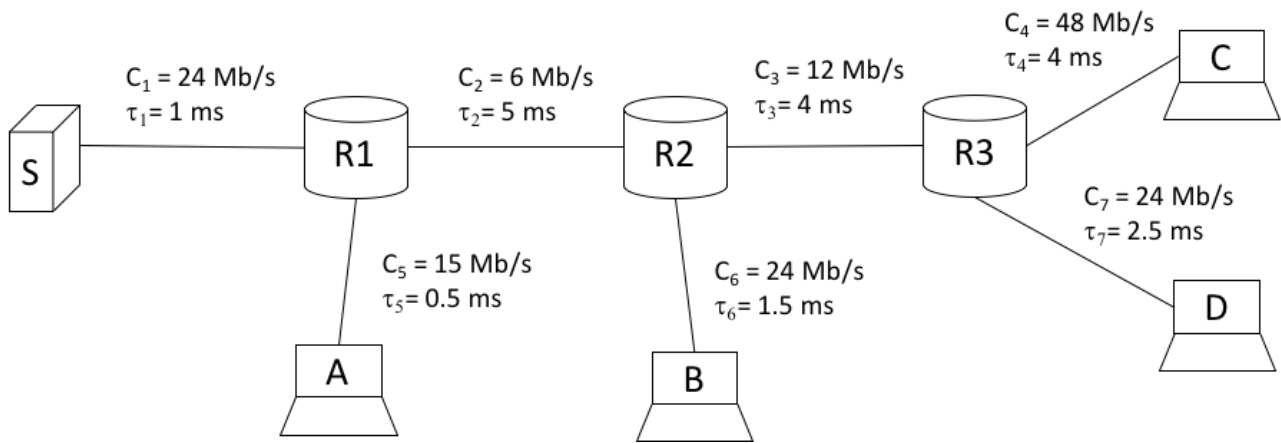


Prova in itinere – 2 Maggio 2018

Cognome	
Nome	
Matricola	

Tempo complessivo a disposizione per lo svolgimento: 2h
Usare lo spazio dopo ogni Esercizio/Quesito per la risposta.

Es1 (6 pt)	Es2 (6 pt)	Es3 (5 pt)	Ques (9 pt)	Lab (6pt)



Compito

1- Esercizio (6 punti)

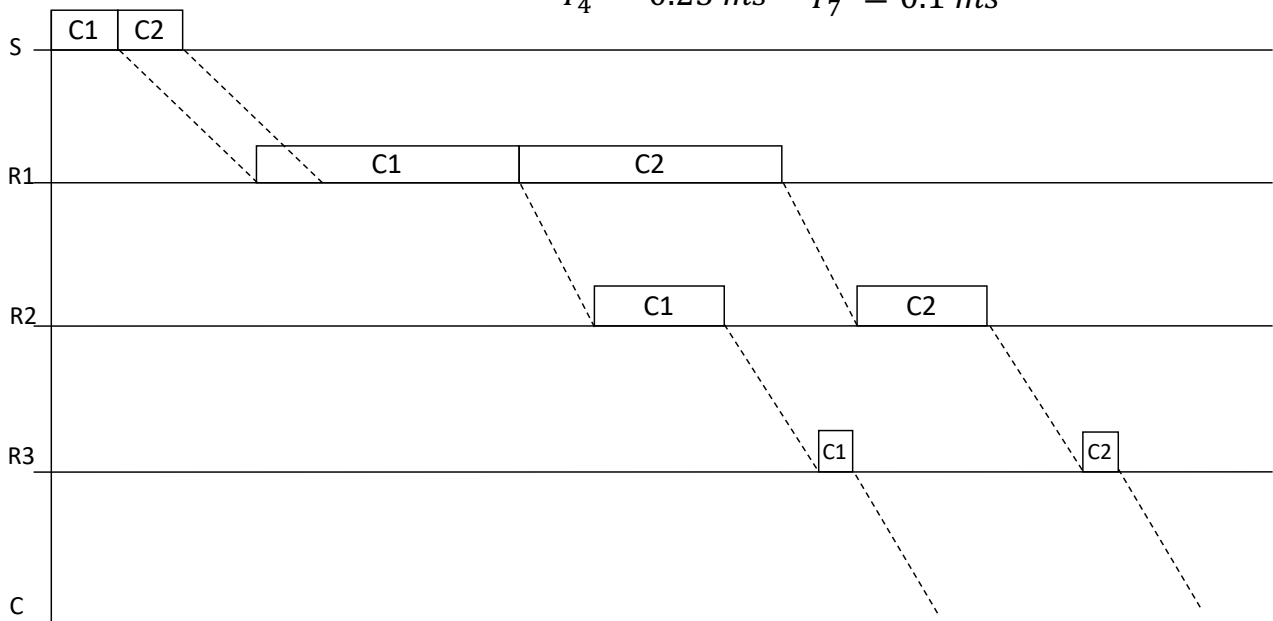
Si consideri la rete nella figura della prima pagina che utilizza la commutazione di pacchetto di tipo store-and-forward.

- Al tempo $t=0$ sono presenti 8 pacchetti in S diretti rispettivamente alle seguenti destinazioni: C,C,D,D,A,B,A,B. I pacchetti abbiano le seguenti dimensioni: $L_A=1200$ Byte, $L_B=1500$ Byte, $L_C=1500$ Byte, $L_D=300$ Byte. Si calcoli l'istante di ricezione dei soli pacchetti diretti a C, D e B.
- Al tempo $t=0$ sono presenti 3 pacchetti in S diretti a C con dimensioni $L_C=1500$ Byte, e 3 pacchetti in A e diretti a B con dimensione $L_B=1500$ Byte. Si calcoli l'istante di ricezione dei soli pacchetti diretti a B.

SOLUZIONE

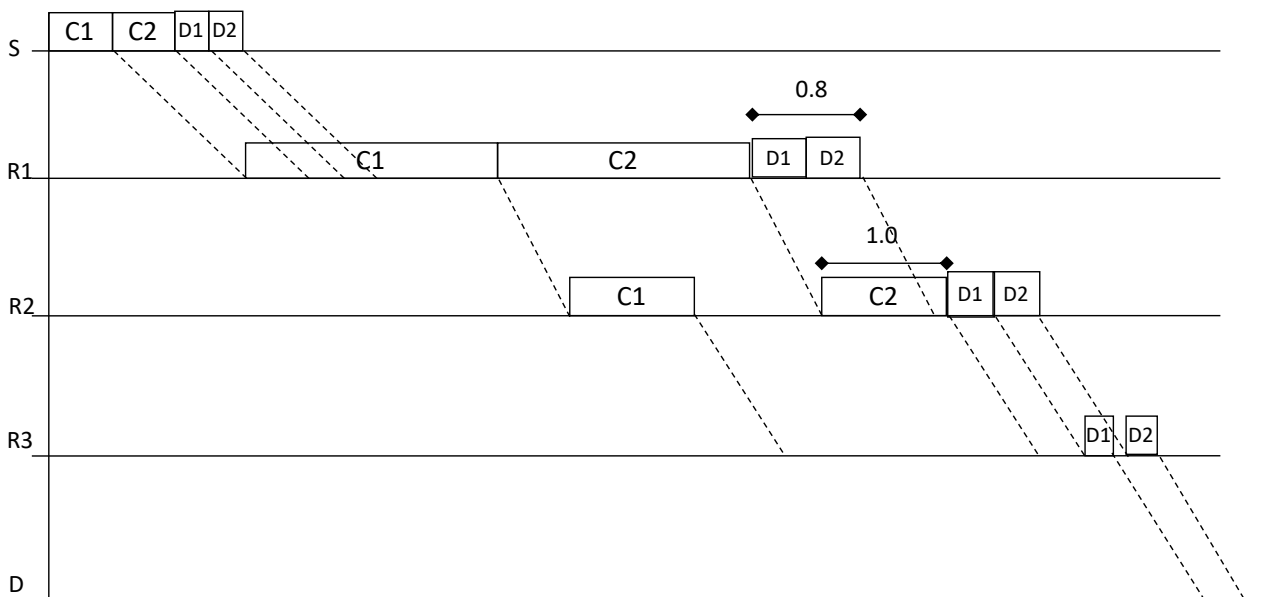
a)

$$\begin{array}{cccc}
 T_1^A = 0.4 \text{ ms} & T_1^B = 0.5 \text{ ms} & T_1^C = 0.5 \text{ ms} & T_1^D = 0.1 \text{ ms} \\
 T_5^A = 0.64 \text{ ms} & T_2^B = 2 \text{ ms} & T_2^C = 2 \text{ ms} & T_2^D = 0.4 \text{ ms} \\
 & T_6^B = 0.5 \text{ ms} & T_3^C = 1 \text{ ms} & T_3^D = 0.2 \text{ ms} \\
 & & T_4^C = 0.25 \text{ ms} & T_7^D = 0.1 \text{ ms}
 \end{array}$$



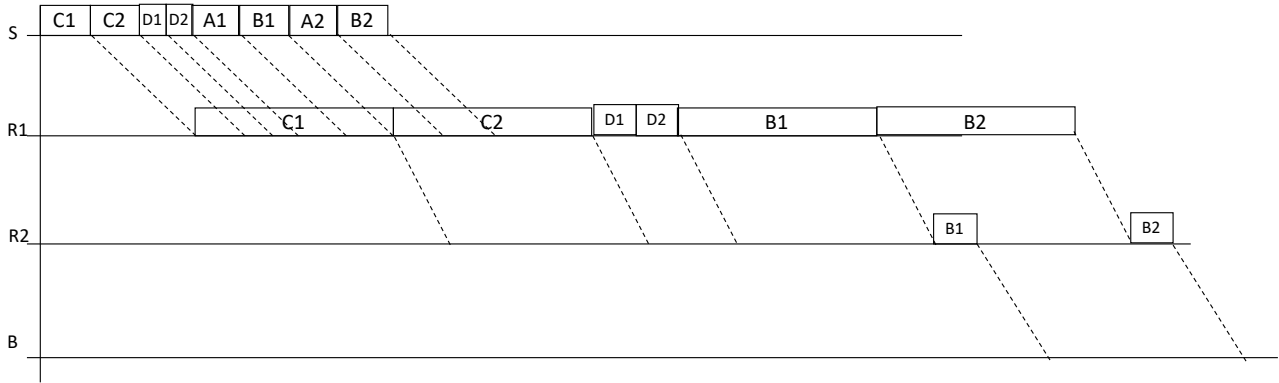
$$T_{C1} = T_1^C + \tau_1 + T_2^C + \tau_2 + T_3^C + \tau_3 + T_4^C + \tau_4 = 0,5 + 1 + 2 + 5 + 1 + 4 + 0,25 + 4 = 17,75 \text{ ms}$$

$$T_{C2} = T_{C1} + T_2^C = 17,75 + 2 = 19,75 \text{ ms}$$



$$T_{D1} = T_1^C + \tau_1 + 2T_2^C + \tau_2 + T_3^C + T_3^D + \tau_3 + T_7^D + \tau_7 = 0,5 + 1 + 2 * 2 + 5 + 1 + 0,2 + 4 + 0,1 + 2,5 = 18,3 \text{ ms}$$

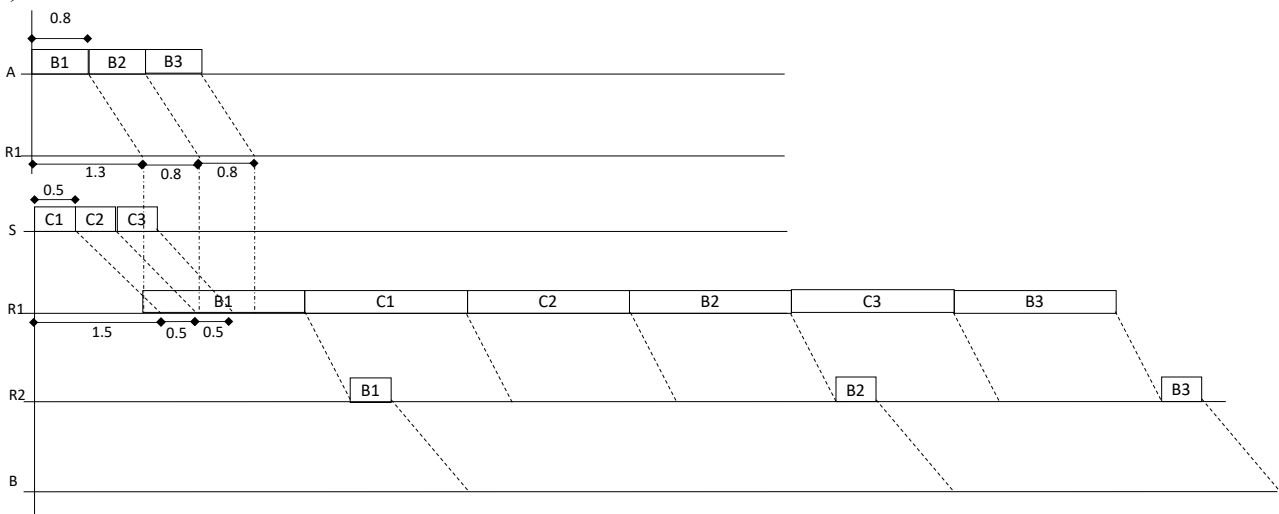
$$T_{D2} = T_{D1} + T_3^D = 18,3 + 0,2 = 18,5 \text{ ms}$$



$$T_{B1} = T_1^C + \tau_1 + 2T_2^C + 2T_2^D + T_2^B + \tau_2 + T_6^B + \tau_6 = 0,5 + 1 + 2 * 2 + 2 * 0,4 + 2 + 5 + 0,5 + 1,5 = 15,3 \text{ ms}$$

$$T_{B2} = T_{B1} + T_2^B = 15,3 + 2 = 17,3 \text{ ms}$$

b)



$$T_{B1} = T_5^B + \tau_5 + T_2^B + \tau_2 + T_6^B + \tau_6 = 0,8 + 0,5 + 2 + 5 + 0,5 + 1,5 = 10,3 \text{ ms}$$

$$T_{B2} = T_{B1} + 2T_2^C + T_2^B = 10,3 + 2 * 2 + 2 = 16,3 \text{ ms}$$

$$T_{B3} = T_{B2} + T_2^C + T_2^B = 16,3 + 2 + 2 = 20,3 \text{ ms}$$

2 - Esercizio (6 punti)

Si consideri la rete nella figura della prima pagina. Una connessione TCP tra l'host S e l'host C è caratterizzata dai seguenti parametri: lunghezze di *header* e *ack* trascurabili, *link* bidirezionali simmetrici, $MSS = 1500$ Byte, $RCWND = 200$ kByte, $SSTHRESH = 12$ kByte.

- Si calcoli il tempo necessario a trasferire un file di 0.525 MByte.
- Si ripeta il calcolo assumendo che il 25° pacchetto vada perso e il timeout corrispondente sia $T_{out} = 200$ ms (si assuma i pacchetti fuori sequenza siano memorizzati).

SOLUZIONE

a)

$$F = 0.525 \text{ MB} = \frac{0.525 \cdot 10^6}{1500} = 350 \text{ MSS}$$

$$T_i = L * 8 / C_i$$

$$RTT = T_1 + T_2 + T_3 + T_4 + 2(\tau_1 + \tau_2 + \tau_3 + \tau_4) = 31.75 \text{ ms}$$

$$T_{open} = 2(\tau_1 + \tau_2 + \tau_3 + \tau_4) = 28 \text{ ms}$$

$$SSTHRESH = 12 \text{ kB} = 8 \text{ MSS}$$

$$W_c = \left\lfloor \frac{RTT}{T_2} \right\rfloor = 16 \text{ MSS}$$

(1)(2)(4)(8)(9)(10)(11)(12)(13)(14)(15)(251 in continua)

$$T_{tot}^a = T_{open} + 11RTT + 250T_2 + RTT = 909 \text{ ms}$$

b)

(1)(2)(4)(8)(9)(T_{out} [9 MSS memorizzati])(1 [$SSTHRESH = \frac{10}{2} = 5$])(2)(4)(5)(6)(7)(8)(9)(10)(11)(12)(13)(14)(15)(200 in continua)

$$T_{tot}^b = T_{open} + 5RTT + T_{out} + 14RTT + 199T_2 + RTT = 1.261 \text{ s}$$

- Esercizio (5 punti)

Si consideri la rete nella figura della prima pagina. Si assuma S sia un server http e C un client http. Si assuma, inoltre, tempo di apertura della connessione dovuto al solo tempo di propagazione e RTT pari a 31.75 [ms], indipendente dal rate della connessione. Occorre trasferire un documento base di 15 KByte e 5 immagini di 0.525 MByte.

- Si calcoli il tempo necessario assumendo 1 flusso interferente tra A e B e 2 flussi tra B e D, e che la connessione sia unica e persistente.
- Come in a), ma con connessioni non-persistenti e trasferimento in parallelo delle immagini.

SOLUZIONE

a)

$$T_{open} = 2(\tau_1 + \tau_2 + \tau_3 + \tau_4) = 28 \text{ ms}$$

Link	1	2	3	4
Capacità	24	6	12	48
Flussi	1	2	3	1
Rate flusso	24	3	4	48

$$T_{html}^a = \frac{L_{html}}{C_2/2} = 40 \text{ ms}$$

$$T_{obj}^a = \frac{L_{obj}}{C_2/2} = 1.4 \text{ s}$$

$$T_{tot}^a = T_{open} + RTT + T_{html}^a + 5(RTT + T_{obj}^a) = 7.2585 \text{ s}$$

b)

Link	1	2	3	4
Capacità	24	6	12	48
Flussi	5	6	7	5
Rate flusso	4,8	1	1,7	9,6

$$T_{html}^b = \frac{L_{html}}{C_2/2} = 40 \text{ ms}$$

$$T_{obj}^b = \frac{L_{obj}}{C_2/6} = 4.2 \text{ s}$$

$$T_{tot}^b = T_{open} + RTT + T_{html}^b + T_{open} + RTT + T_{obj}^b = 4.3595 \text{ s}$$

Quesiti (9 punti)

Q1

Da un host viene eseguito il comando dig. Rispondere alle domande sotto.

```
$ dig -t ANY polimi.it

; <<> DiG 9.10.6 <<> -t ANY polimi.it
;; global options: +cmd
;; Got answer:
;; ->HEADER<<- opcode: QUERY, status: NOERROR, id: 16498
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 11, AUTHORITY: 0, ADDITIONAL: 7

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;polimi.it.                IN      ANY

;; ANSWER SECTION:
polimi.it.                86400  IN     SOA    ns.polimi.it. root.ns.polimi.it. 2017083034 11600 3600 604800 86400
polimi.it.                86400  IN     NS     dns.cineca.it.
polimi.it.                86400  IN     NS     ns.polimi.it.
polimi.it.                86400  IN     NS     ns2.polimi.it.
polimi.it.                86400  IN     A      131.175.187.72
polimi.it.                900    IN     MX     0 polimi-it.mail.protection.outlook.com.

;; ADDITIONAL SECTION:
ns.polimi.it.            86400  IN     A      131.175.12.1
ns2.polimi.it.          86400  IN     A      131.175.12.2
dns.cineca.it.          1543   IN     A      130.186.1.70
polimi-it.mail.protection.outlook.com. 9 IN A    213.199.154.234
polimi-it.mail.protection.outlook.com. 9 IN A    213.199.154.170
dns.cineca.it.          2      IN     AAAA   2001:760:2e0a:8002::20

;; Query time: 4 msec
;; SERVER: 131.175.12.1#53(131.175.12.1)
;; WHEN: Thu Apr 26 10:38:03 CEST 2018
;; MSG SIZE rcvd: 657
```

Il server da cui arriva la risposta è authoritative? Quali sono i server authoritative?

I server authoritative sono dns.cineca.it (130.186.1.70), ns.polimi.it (131.175.12.1), ns2.polimi.it (131.175.12.2).

La risposta arriva dall'IP 131.175.12.1, che è un NS authoritative

A cosa è associato l'indirizzo 131.175.187.72?

Ad una macchina del dominio polimi.it, la macchina di default.

A quali server (nome ed indirizzo IP) verrebbe spedita una mail indirizzata a rettore@polimi.it?

Nome: polimi-it.mail.protection.outlook.com
IP: 213.199.154.234 o 213.199.154.170

A che scopo gli indirizzi 213.199.154.234 e 213.199.154.170 sono associati allo stesso nome?

A fare load balacing sulle richiesta di connessione al nome DNS a cui si riferiscono, vale a dire il mail server del dominio polimi.it

Q2

Un segnale audio è caratterizzato da una banda di 22 kHz. Viene campionato alla minima frequenza di campionamento usando 24 bit per campione. Successivamente viene suddiviso in pacchetti di 1460 Byte di dati e 140 Byte di overhead dei vari protocolli. Calcolare:

Il rate del segnale campionato e quantizzato (in bit a secondo):

$$F_c = 2 * B = 44 \text{ kHz}$$

$$R = F_c * 24 = 1,056 \text{ Mb/s}$$

Il numero di pacchetti generati ogni secondo:

Il flusso di R [bit/s] deve essere accomodato in pacchetti che trasportano 1460 [byte], quindi devo generare ogni secondo

$$F_p = \frac{R}{1460 * 8} = 90,41 \text{ pkt/s}$$

Il rate dovuto alla trasmissione dei pacchetti generati (in bit al secondo):

I pacchetti generati trasportano i bit del segnale a cui viene aggiunto in header per un rate complessivo di:

$$R_p = F_p * (1460 + 140) * 8 = 1,157 \text{ Mb/s}$$

Q3

Durante una sessione TCP, l'algoritmo di Jacobson stima valor medio e deviazione standard del RTT come $SRTT^0 = 160 \text{ ms}$ e $SDEV^0 = 40 \text{ ms}$. I due segmenti successivi registrano un RTT di $RTT^1 = 160 \text{ ms}$ e $RTT^2 = 90 \text{ ms}$. Si indichino nella tabella i valori di SRTT, SDEV, DEV e del Timeout alla ricezione di ciascuno dei due segmenti considerando $(1 - \alpha) = 7/8$ come peso della stima precedente di RTT e $(1 - \beta) = 3/4$ come peso della stima precedente di SDEV. Si usi la tabella per indicare i risultati finali e lo spazio sottostante per mostrare i conti fatti.

	RTT	SRTT	DEV	SDEV	Timeout
$SRTT^0 = 100$ $SDEV^0 = 40$	$RTT^1 = 160$	$SRTT^1 =$	$DEV^1 =$	$SDEV^1 =$	$T^1 =$
	$RTT^2 = 90$	$SRTT^2 =$	$DEV^2 =$	$SDEV^2 =$	$T^2 =$

$$SRTT^1 = \frac{7}{8} SRTT^0 + \frac{1}{8} RTT^1 = 107,5 \text{ ms}$$

$$SRTT^2 = \frac{7}{8} SRTT^1 + \frac{1}{8} RTT^2 = 105,3 \text{ ms}$$

$$DEV^1 = |RTT^1 - SRTT^0| = 60 \text{ ms}$$

$$DEV^2 = |RTT^2 - SRTT^1| = 17,5 \text{ ms}$$

$$SDEV^1 = \frac{3}{4} SDEV^0 + \frac{1}{4} DEV^1 = 45 \text{ ms}$$

$$SDEV^2 = \frac{3}{4} SDEV^1 + \frac{1}{4} DEV^2 = 38,1 \text{ ms}$$

$$T^1 = SRTT^1 + 4SDEV^1 = 287,5 \text{ ms}$$

$$T^2 = SRTT^2 + 4SDEV^2 = 257,7 \text{ ms}$$

Laboratorio (6 punti)

Nome Cognome	
Matricola	

Client

```
from socket import *
clientSocket = socket(AF_INET, SOCK_DGRAM)
clientSocket.settimeout(1)
try:
    while 1:
        message = raw_input('Input a Country name, or Close to finish this client:')
        if message == 'Close':
            break
        clientSocket.sendto(message, ('localhost', 2018))
        modifiedMessage, serverAddress = clientSocket.recvfrom(2048)
        if modifiedMessage == '1':
            print message + ' is qualified to the World Cup'
        else:
            print message + ' is not qualified to the World Cup'
finally:
    clientSocket.close()
```

Server

```
from socket import *
import requests
import json
serverSocket = socket(AF_INET, SOCK_STREAM)
serverSocket.bind(('localhost', 2018))
while 1:
    try:
        message, clientAddress = serverSocket.recvfrom(2048)
        modifiedMessage = message.title()
        #title() ritorna una copia della stringa in cui il primo carattere di tutte le
        # parole è maiuscolo
        r = requests.get('http://api.football-data.org/v1/competitions/467/teams')
        teams_qualificati = parseRisposta(r.json())
        t = '0'
        for team in teams_qualificati:
            if team == modifiedMessage:
                t = '1'
        serverSocket.sendto(t, clientAddress)
    finally:
        serverSocket.close()
```

Q1. Indicare gli errori che ci sono nello script Server (2 punti)

1. Modificare serverSocket per utilizzare UDP: serverSocket = socket(AF_INET, SOCK_DGRAM).
2. Cambiare il blocco finally e togliere serverSocket.close() dal ciclo while. Inserire il blocco except, ad esempio:
Except error, e:
print e