# Project 1 - MDRS

Universidade de Aveiro(*UA*)

João Afonso Pereira Ferreira
Rafael Luís Ferreira Curado

VERSION 1

# Report - Project 1

*Modelação e Desempenho de Redes e Serviços (MDRS)*

Amaro Sousa

DETI - Departamento de Electrónica, Telecomunicações e Informática

Engenharia de Computadores e Telemática (ECT)

João Afonso Pereira Ferreira
Rafael Luís Ferreira Curado

(103037) ferreiraafonsojoao@ua.pt
(103199) rafael.curado@ua.pt

25/10/2023

# Table of Contents

# List of Figures

# Chapter 1

# Introduction

In this report will be discussed and analysed the impact of various parameters on the performance of simulated systems using combination of simulation results with theoretical modeling technique predictions. In order to accomplish this, it was used *MATLAB*.

# Chapter 2

# Development

## 2.1 Task 1

### 2.1.1 1 (a)

**Goal:** Present the average packet delay results in bar charts with the confidence intervals in error bars. Justify the results. Draw conclusions concerning the impact of the link capacity in the obtained results.

**Code**

```
1
2  %% 1.a)
3  lambda = 1800;           % pps
4  f = 1000000;             % Bytes
5  N = 20;                  % times to simulate
6  P = 100000;              % stopping criteria
7  alfa = 0.1;              % 90% confidence interval
8  C = [10, 20, 30, 40];    % 10, 20, 30, 40 Mbps
9
10 APD = zeros(length(C), N);
11 media = zeros(length(C), 1);
12 term = zeros(length(C), 1);
13 for i = 1:length(C)
14     for n = 1:N
15         [~, APD(i, n), ~, ~] = Simulator1(lambda, C(i)
               , f, P); % it captures only the second
               value ignoring the others
16     end
17
18     media(i, 1) = mean(APD(i,:)); % side is 1-by-20
```

```
    term(i, 1) = norminv(1-alfa/2) * sqrt(var(APD(i,:))/N);

    fprintf("APD C%d (ms) = %.2e +- %.2e\n", C(i), media(i), term(i)); % check for each C
end

figure(1)
bar(C, APD(:, 1))
hold on
grid on
er = errorbar(C, APD(:, 1), term(:, 1) * -1, term(:, 1));
xlabel('C, Link Bandwidth (Mbps)')
ylabel('APD, Average Packet Delay (ms)')
title('Average Packet Delay (ms) vs Link Bandwidth (Mbps)');
hold off;
hold off
```

**Results**

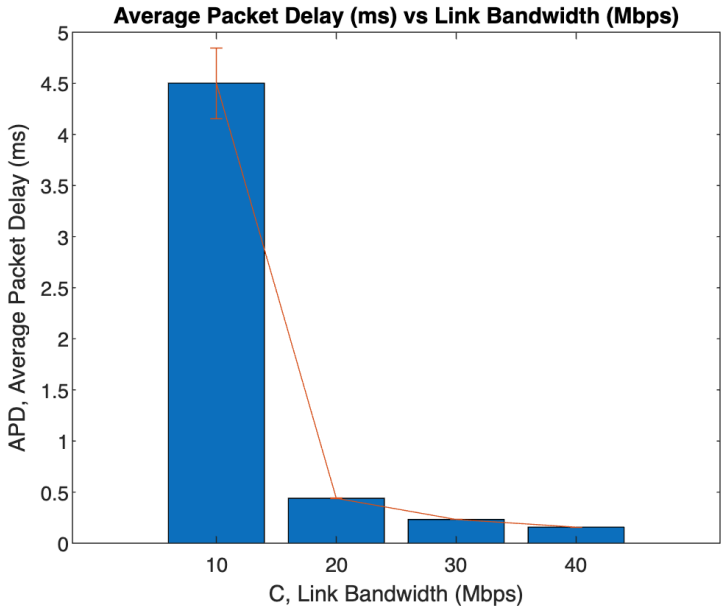



Figure 2.1: Average Packet Delay Results

As observed in Fig. 2.1, when increasing link capacity (C) the Average Packet Delay (APD) reduces. The link capacity increases from 10 Mbps to 40 Mbps, the average packet delay decreases. This result can be explain since a larger link capacity allows the network to handle a bigger traffic load as well as reduced network congestion results in faster packet transfer, transmission of packets and shorter queuing waits.

In conclusion, these simulation results give a demonstration of how the link capacity (C) is a critical factor in determining the average packet delay in a network. Higher link capacities lead to reduced delays.

### 2.1.2 1 (b)

**Goal:** Consider the system modelled by a M/G/1 queuing model. Determine the theoretical values of the average packet delay for all cases of 1 (a) (2.1.1). Compare the theoretical values with the simulation results and draw conclusions.

**Code**

```
1
2  % 1 b)
3  lambda = 1800;           % pps
4  f = 1000000;             % Bytes
5  C = [10, 20, 30, 40];    % 10, 20, 30, 40 Mbps
6
7  % Calculation of the probability associated with each
       packet size
8  prob = (1 - (0.19 + 0.23 + 0.17)) / ((109 - 65 + 1) +
       (1517 - 111 + 1)); % calculating the probability
       associated with each packet size that is not 64,
       110, or 1518 bytes
9
10 bytes = 64:1518;
11
12 % Loop through each link capacity to calculate E[S] &
       E[S^2]
13 for i = 1:length(C)
14
15     S = (bytes .* 8)./(C(i)*10^6);  % E[S]
16     S2 = (bytes .* 8)./(C(i)*10^6); % E[S^2]
17
18     for j = 1:length(bytes)
19         if j == 1
20             S(j) = S(j) * 0.19;
21             S2(j) = S2(j)^2 * 0.19;
22         elseif j == 110 - 64 + 1
23             S(j) = S(j) * 0.23;
```

```
            S2(j) = S2(j)^2 * 0.23;
        elseif j == 1518 - 64 + 1
            S(j) = S(j) * 0.17;
            S2(j) = S2(j)^2 * 0.17;
        else
            S(j) = S(j) * prob;
            S2(j) = S2(j)^2 * prob;
        end
    end

    ES = sum(S);
    ES2 = sum(S2);

    w = lambda * ES2 / (2 * (1 - lambda * ES)) + ES;

    fprintf("For C = %2d Mbps: -->", C(i));
    fprintf(" Av. Packet Delay %.2e (ms)\n", w * 10^3)
        ;
end
```

**Results**

When running the code, this is the output we get:

```
For C = 10 Mbps:  -> Av.  Packet Delay 4.39e+00 (ms)
For C = 20 Mbps:  -> Av.  Packet Delay 4.36e-01 (ms)
For C = 30 Mbps:  -> Av.  Packet Delay 2.31e-01 (ms)
For C = 40 Mbps:  -> Av.  Packet Delay 1.58e-01 (ms)
```

From the theoretic lessons and slides, if we are considering a system modeled by an M/G/1 queuing model:

1. The arrival of customers is a Poisson process with rate $\lambda$.
2. The service time S of the server has a generic distribution and is independent of customer arrivals.
3. The system has one server.
4. The system accommodates an infinite number of customers.

In order to obtain the average delay, which corresponds to the sum of the average delay in the waiting queue + the average service time, we need to use the formula:

$$W = \frac{\lambda E[S^2]}{2(1 - \lambda E[S])} + E[S], \tag{2.1}$$

in which we only have knowledge of $\lambda$, which is 1800 packets per second. Then, the missing variables $E[S]$ and $E[S^2]$ must be calculated, and finally, the results will be obtained.

As the link capacity (C) increases, the average packet delay decreases, which is in line with the expected behavior in a queuing system since the APD is decreasing while increasing the link capacity.

### 2.1.3 1 (c)

**Goal:** Present the results in 2 different figures: (i) the average packet delay results and (ii) the average throughput results (in both cases, in bar charts with the confidence intervals in error bars). Justify the obtained results. Draw conclusions concerning the impact of the packet arrival rate in the obtained results

**Code**

```
1
2  % 1 c)
3
4  lambda = [1000, 1300, 1600, 1900];  % Packet arrival
       lambda (pps)
5  C = 10;                             % Link capacity (
       Mbps)
6  f = 1000000;                        % Bytes
7  N = 20;                             % Number of
       simulations
8  P = 100000;                         % Stopping
       criteria
9  alfa = 0.1;                         % 90% confidence
       interval
10
11 % Initialize variables to store simulation results for
       all lambda
12 APD_values = zeros(1, length(lambda));
13 APD_terms = zeros(1, length(lambda));
14 TT_values = zeros(1, length(lambda));
15 TT_terms = zeros(1, length(lambda));
16
17 for i = 1:length(lambda)
18     % Variables to store the simulation results
19     PL = zeros(1, N);
20     APD = zeros(1, N);
21     MPD = zeros(1, N);
22     TT = zeros(1, N);
23
```

```
    for it = 1:N
        [~, APD(it), ~, TT(it)] = Simulator1(lambda(i)
            , C, f, P);
    end

    % Calculate Avg. Packet Delay
    media = mean(APD);
    term = norminv(1 - alfa/2) * sqrt(var(APD) / N);
    APD_values(i) = media;
    APD_terms(i) = term;

    % Calculate Avg. Packet Loss
    media = mean(TT);
    term = norminv(1 - alfa/2) * sqrt(var(TT) / N);
    TT_values(i) = media;
    TT_terms(i) = term;
end

% i)
figure(1);
hold on;
grid on;
bar(lambda, APD_values);
errorbar(lambda, APD_values, APD_terms);
title('Average Packet Delay');
xlabel('Packet Rate (pps)');
ylabel('Avg. Packet Delay (ms)');
hold off;

% ii)
figure(2);
hold on;
grid on;
bar(lambda, TT_values);
errorbar(lambda, TT_values, TT_terms);
title('Average Throughput');
xlabel('Packet Rate (pps)');
ylabel('Average Throughput (Mbps)');
hold off;
```

**Results**

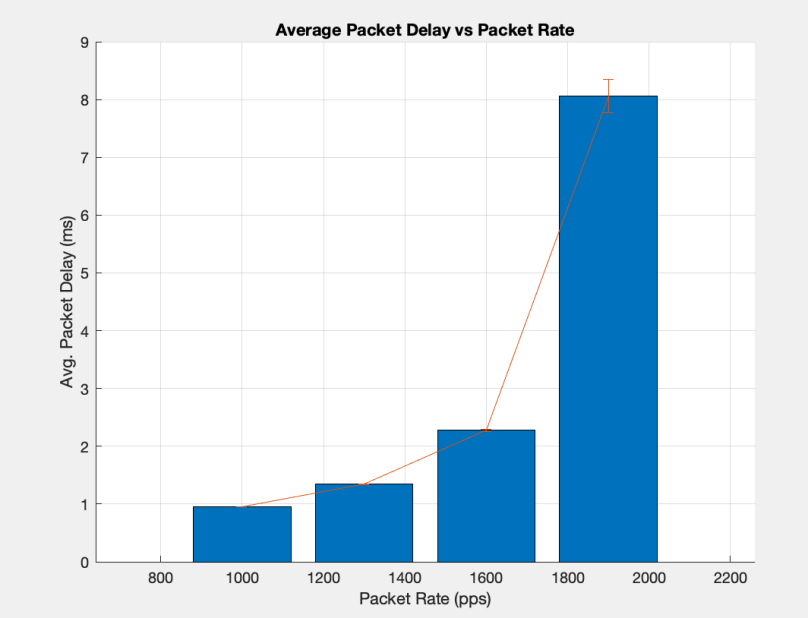



Figure 2.2: Average Packet Delay

APD: When the packet arrival rate ($\lambda$) is high, it indicates that more packets are arriving in the queue in a shorter amount of time. This increased traffic might cause congestion in a network or a system. This congestion in packets leads to increased delay because they have to wait in a queue before actually being served. This explains why it is possible to observe an increase in APD as packet rate ($\lambda$) increases.

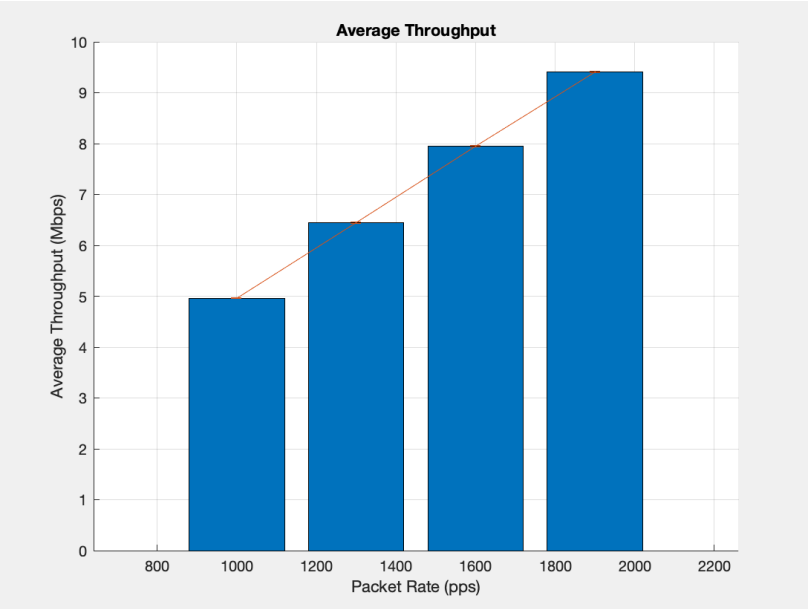



Figure 2.3: Average Throughput

Average Throughput (TT): When the packet arrival rate ($\lambda$) increases, it implies that more packets are being transmitted over the network. This can result in increased throughput because more data is transmitted per unit of time.

Each additional packet arrival rate of 1000 pps results in an approximately 5 Mbps increase in throughput. When the packet arrival rate is high, it indicates that the network is capable of delivering more data.

In summary, the observed increase in APD with an increasing packet rate ($\lambda$) is might cause congestion, which might cause greater packet delays. The increase of Average Throughput (TT) with increasing packet rate, on the other hand, demonstrates the network's ability to transmit more data when dealing with greater packet arrival rates.

### 2.1.4 1 (d)

**Goal:** Repeat experiments 1.c but now using Simulator2 with b = $10^-5$. Justify the obtained results. Compare these results with the previous ones and draw conclusions concerning the impact of the BER in the obtained results.

**Code**

```
lambda = [1000, 1300, 1600, 1900];  % Packet arrival lambda (pps)
C = 10;                             % Link capacity (Mbps)
f = 1000000;                        % Bytes
N = 20;                             % Number of simulations
P = 100000;                         % Stopping criteria
b = 10^-5                           % ber
alfa = 0.1;                         % 90% confidence interval

% Initialize variables to store simulation results
APD_values = zeros(1, length(lambda));
APD_terms = zeros(1, length(lambda));
TT_values = zeros(1, length(lambda));
TT_terms = zeros(1, length(lambda));

for i = 1:length(lambda)
    % Variables to store the simulation results
    PL = zeros(1, N);
    APD = zeros(1, N);
    MPD = zeros(1, N);
    TT = zeros(1, N);

    for it = 1:N
        [~, APD(it), ~, TT(it)] = Simulator2(lambda(i)
            , C, f, P, b);
    end

    % Calculate Avg. Packet Delay
    media = mean(APD);
    term = norminv(1 - alfa/2) * sqrt(var(APD) / N);
    APD_values(i) = media;
    APD_terms(i) = term;

    % Calculate Avg. Packet Loss
    media = mean(TT);
    term = norminv(1 - alfa/2) * sqrt(var(TT) / N);
    TT_values(i) = media;
    TT_terms(i) = term;
end

```

```
% i)
figure(1);
hold on;
grid on;
bar(lambda, APD_values);
errorbar(lambda, APD_values, APD_terms);
title('Average Packet Delay');
xlabel('Packet Rate (pps)');
ylabel('Avg. Packet Delay (ms)');
hold off;

% ii)
figure(2);
hold on;
grid on;
bar(lambda, TT_values);
errorbar(lambda, TT_values, TT_terms);
title('Average Throughput');
xlabel('Packet Rate (pps)');
ylabel('Average Throughput (Mbps)');
hold off;
```

**Results**

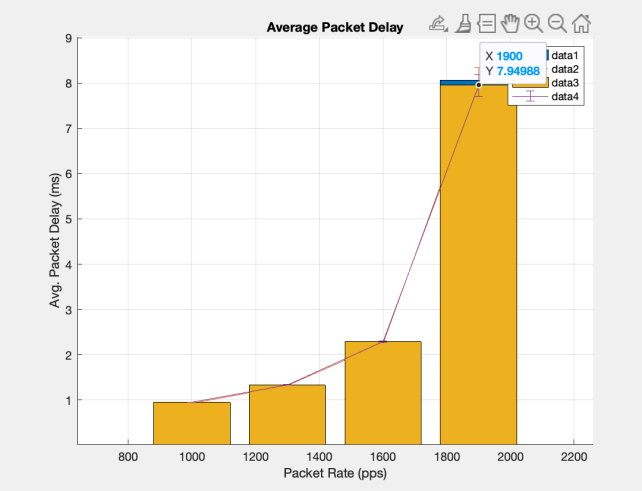



Figure 2.4: Average Packet Delay w/ BER (YELLOW, BLUE IS FROM 1 c)

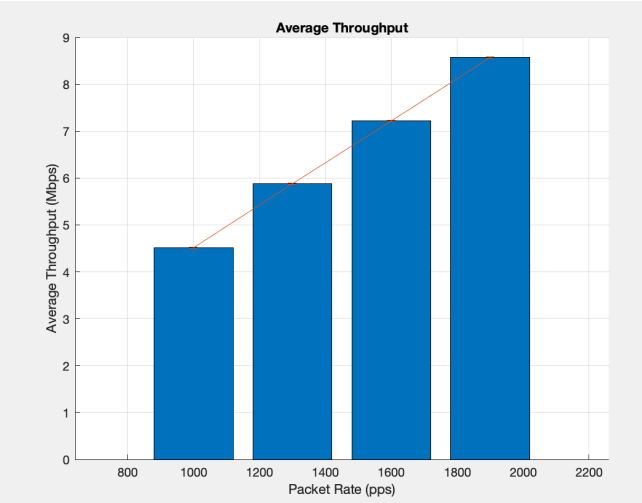



Figure 2.5: Average Throughput w/ BER

In exercise 1 c) without BER, it's expected that APD is relatively lower because there are no errors introduced in the simulation. The packets are assumed to be transmitted perfectly without errors. This results in relatively lower delays. On the other hand, in this exercise 1 d) with BER, the retransmission of packets that experience errors can contribute to higher APD. The time taken for error detection and recovery processes can increase the overall delay for packets. Reduced Throughput: Packet errors can result in a lower Transmitted Throughput (TT). When packets are corrupted and need retransmission, they consume additional bandwidth and reduce the effective throughput of the network.

### 2.1.5 1 (e)

**Goal:** Change both simulators to consider that the packet size is between 64 and 1518 bytes with the following new probabilities: 25% for 64 bytes, 17% for 110 bytes, 11% for 1518 bytes and an equal probability for all other values (i.e., from 65 to 109 and from 111 to 1517). Repeat experiments 1.c and 1.d with the new simulators.Justify the obtained results

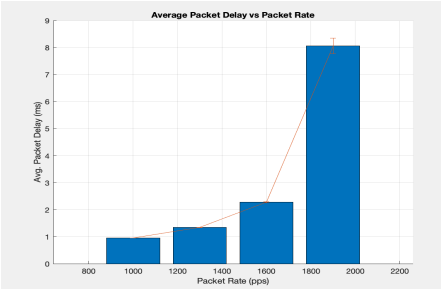
In order to change the packet size with the probabilities described before it had to be changed Simulator 1 and 2. To accomplish this, the function *GeneratePacketSize()* had to be changed for both simulators as we can see bellow:

### Code For the New Simulator 1 and 2

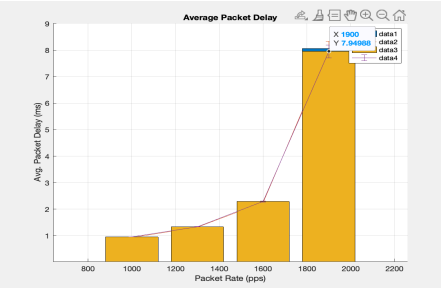
```
1
2  % BEFORE:
3  function out= GeneratePacketSize()
4      aux= rand();
5      aux2= [65:109 111:1517];
6      if aux <= 0.19
7          out= 64;
8      elseif aux <= 0.19 + 0.23
9          out= 110;
10     elseif aux <= 0.19 + 0.23 + 0.17
11         out= 1518;
12     else
13         out = aux2(randi(length(aux2)));
14     end
15 end
16
17
18 % AFTER:
19
20 function out= GeneratePacketSize()
21     aux= rand();
22     aux2= [65:109 111:1517];
23     if aux <= 0.25
24         out = 64;
25     elseif aux <= 0.25 + 0.17
26         out = 110;
27     elseif aux <= 0.25 + 0.17 + 0.11
28         out = 1518;
29     else
30         out = aux2(randi(length(aux2)));
31     end
32 end
```

### Results

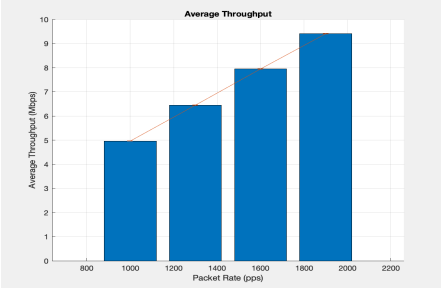
In order to establish a comparison between the results obtained in this exercise with the ones of 1.c and 1.d, will be present the previous exercises results again.



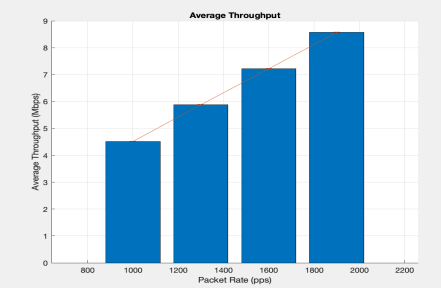

(a) Average Packet Delay w/out BER




(b) Average Packet Delay w/ BER

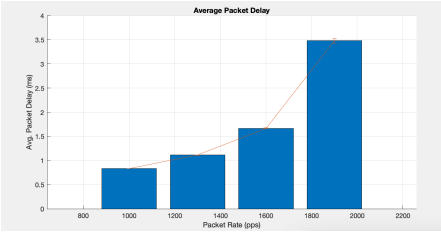



(c) Average Throughput w/out BER

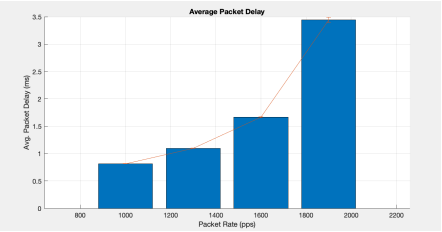



(d) Average Throughput w/ BER

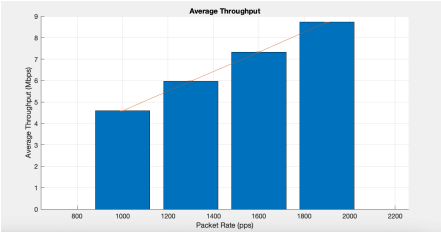
Figure 2.6: Subfigures (a) and (c) are related to Exercise 1c), the others to 1d)



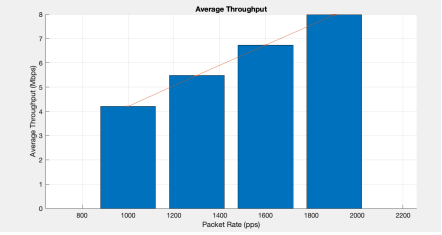

(a) Average Packet Delay Simulator 1 V2




(b) Average Packet Delay Simulator 2 V2




(c) Average Throughput Simulator 1 V2




(d) Average Throughput Simulator 2 V2

Figure 2.7: Results for 1 e)

Now Comparing the results from 1 e) with 1 c) and d) it was possible to notice:

- **LOWER Average Packet Delay**:
  In exercise 1 e) it was introduced a packet size distribution with a percentage of **smaller packets** (64 bytes). Smaller packets often lead to less transmitting time, which explains **lower delays**.

  In exercise 1 d) using Simulator2, it includes **BER** although if is being discussed smaller packets, due to the probability of a bit error occurring in a smaller packet is lower compared to a larger one they will be less affected by these bit error, which leads to **lower delays**.

  In terms of efficiency, we are talking about a smaller packet size distribution and from what it was previously described, **lower delays** can increase **network efficiency**, since the packets will be transmitted way more faster, no network congestion.

- **SLIGHTLY LOWER Average Throughput**:
  As mentioned before, it was introduced a packet size distribution with a percentage of **smaller packets** (64 bytes). Smaller packets often lead to less transmitting time, which explains **lower delays** as they may lead to slightly lower average throughput since it is a measure of how many units of information a system can process in a given amount of time [1].

  BER impact: introducing smaller packets that are less affected by bit errors, may still experience retransmissions and, thus, it may result a slight reduction in throughput as we can observe in Fig. 2.7.

  Smaller packets enable faster transmission and, as a result, less time required to reach full network capacity, resulting in a small decrease in throughput.

## 2.2 Task 2

For this task the team started by changing both simulators to estimate a new performance parameter: the average packet queuing delay as well as the calculation of it for data packets and VoIP packets. Following, it will only be present the modified code for both simulators that the team found the best to achieve this problem solution.

**Code For the New Simulator 3**

```
function [PLd, PLv , APDd , APDv , MPDd, MPDv , APQDd, APQDv, TT] = Simulator3V2(lambda,C,f,P,n)
% INPUT PARAMETERS:
%  lambda - packet rate (packets/sec)
%  C      - link bandwidth (Mbps)
%  f      - queue size (Bytes)
%  P      - number of packets (stopping criterion)
%  n      - number of VoIP packet flows
% OUTPUT PARAMETERS:
%  PLd    - Packet Loss of data packets (%)
%  PLv    - Packet Loss of VoIP packets (%)
%  APDd   - Average Delay of data packets (milliseconds)
%  APDv   - Average Delay of VoIP packets (milliseconds)
%  MPDd   - Maximum Delay of data packets (milliseconds)
%  MPDv   - Maximum Delay of VoIP packets (milliseconds)
%  APQDd  - Average Packet Queuing Delay of data packets (milliseconds)
%  APQDv  - Average Packet Queuing Delay of VoIP packets (milliseconds)
%  TT     - Transmitted Throughput (data + VoIP) (Mbps)

% Events:
ARRIVAL= 0;          % Arrival of a packet
DEPARTURE= 1;        % Departure of a packet

%Packet type
DATA = 0;
VOIP = 1;

% Queuing Delay Variables
TotalQueuingDelayD = 0;
TotalQueuingDelayV = 0;

%State variables:
STATE = 0;               % 0 - connection free; 1 - connection bysy
QUEUEOCCUPATION= 0;      % Occupation of the queue (in Bytes)
```

```
34 QUEUE= [];                % Size and arriving time
       instant of each packet in the queue
35
36 ...
37
38 % Similation loop:
39 while (TRANSMITTEDPACKETSD+TRANSMITTEDPACKETSV) < P %
       Stopping criterium
40     ...
41     if QUEUEOCCUPATION > 0
42             Event_List = [Event_List; DEPARTURE, Clock
                   + 8*QUEUE(1,1)/(C*10^6), QUEUE(1,1),
                  QUEUE(1,2), QUEUE(1,3)];
43             QUEUEOCCUPATION= QUEUEOCCUPATION - QUEUE
                   (1,1);
44
45             if (TRANSMITTEDPACKETSD +
                   TRANSMITTEDPACKETSV) < P
46                 if QUEUE(1,3) == DATA
47                     TotalQueuingDelayD =
                           TotalQueuingDelayD + (Clock -
                           QUEUE(1,2));
48                 else
49                     TotalQueuingDelayV =
                           TotalQueuingDelayV + (Clock -
                           QUEUE(1,2));
50                 end
51             end
52
53             QUEUE(1,:)= [];
54
55         else
56             STATE= 0;
57         end
58
59 ...
60
61 APQDd = 1000*TotalQueuingDelayD / TRANSMITTEDPACKETSD;
62 APQDv = 1000*TotalQueuingDelayV / TRANSMITTEDPACKETSD;
```

**Code For the New Simulator 4**

```
function [PLd, PLv, APDd, APDv, MPDd, MPDv, APQDd,
    APQDv, TT] = Simulator4V2(lambda,C,f,P,n)
%Events:
ARRIVAL= 0;         % Arrival of a packet
DEPARTURE= 1;       % Departure of a packet
%Packet type
DATA = 0;
VOIP = 1;
% Queuing Delay Variables
TotalQueuingDelayD = 0;
TotalQueuingDelayV = 0;
%State variables:
STATE = 0;
QUEUEOCCUPATION= 0;
QUEUE= [];
...
% Similation loop:
while (TRANSMITTEDPACKETSD + TRANSMITTEDPACKETSV) < P
            if QUEUEOCCUPATION > 0
                QUEUE = sortrows(QUEUE, 3, "descend");
                EventList = [EventList; DEPARTURE,
                    Clock + 8*QUEUE(1,1)/(C*10^6),
                    QUEUE(1,1), QUEUE(1,2), QUEUE(1,3)
                    ]; % Add PacketType
                QUEUEOCCUPATION= QUEUEOCCUPATION -
                    QUEUE(1,1);

               if (TRANSMITTEDPACKETSD +
                   TRANSMITTEDPACKETSV) < P
                    if QUEUE(1,3) == DATA
                        TotalQueuingDelayD =
                            TotalQueuingDelayD + (Clock
                             - QUEUE(1,2));
                    else
                        TotalQueuingDelayV =
                            TotalQueuingDelayV + (Clock
                             - QUEUE(1,2));
                    end
                end
                QUEUE(1,:)= [];
            else
                STATE= 0;
            end
...
```

```
APQDd = 1000*TotalQueuingDelayD / TRANSMITTEDPACKETSD;
APQDv = 1000*TotalQueuingDelayV / TRANSMITTEDPACKETSD;
```

### 2.2.1 2 (a)

**Goal:** Consider the case of $\lambda$ = 1500 pps, C = 10 Mbps and f =1.000.000 Bytes. Run 20 times Simulator3 with a stopping criterion of P = 100.000 at each run and compute the estimated values and the 90% confidence intervals of the average delay and average queuing delay of data packets and VoIP packets when n = 10, 20, 30 and 40 VoIP flows. Present the results in 4 different figures: (i) the average delay of data packets, (ii) the average delay of VoIP packets, (iii) the average queuing delay of data packets and (iv) the average queuing delay of VoIP packets

**Code**

```
% Exercicio 2
%% 2 a)
lambda = 1500;  % pps
C = 10;         % Mbps
f = 1000000;    % Bytes
P = 10000;      % stopping criterion
num_runs = 20;

% Number of VoIP flows
voip_flows = [10, 20, 30, 40];

% Confidence level
alfa = 0.1;  % 90% confidence level

% Initialize arrays to store results
APD_data = zeros(length(voip_flows), num_runs);
APD_voip = zeros(length(voip_flows), num_runs);
APQD_data = zeros(length(voip_flows), num_runs);
APQD_voip = zeros(length(voip_flows), num_runs);

% Calculate mean and confidence intervals for APD_data
    and APQD_data
APD_data_values = zeros(length(voip_flows), 1);
APD_data_terms = zeros(length(voip_flows), 1);
APD_voip_values = zeros(length(voip_flows), 1);
APD_voip_terms = zeros(length(voip_flows), 1);
APQD_data_values = zeros(length(voip_flows), 1);
APQD_data_terms = zeros(length(voip_flows), 1);
APQD_voip_values = zeros(length(voip_flows), 1);
```

```
APQD_voip_terms = zeros(length(voip_flows), 1);

for n_idx = 1:length(voip_flows)
    n = voip_flows(n_idx);
    for run = 1:num_runs
        [PLd, PLv , APDd , APDv , MPDd, MPDv , APQDd,
           APQDv, TT] = Simulator3V2(lambda, C, f, P,
           n);

        % Store the results for each run
        APD_data(n_idx, run) = APDd;
        APD_voip(n_idx, run) = APDv;
        APQD_data(n_idx, run) = APQDd;
        APQD_voip(n_idx, run) = APQDv;
    end

    % Calculate Avg. Packet Delay for Data Packets
    mean_APD_data = mean(APD_data(n_idx, :));
    term_APD_data = norminv(1 - alfa/2) * sqrt(var(
       APD_data(n_idx, :)) / num_runs);
    APD_data_values(n_idx) = mean_APD_data;
    APD_data_terms(n_idx) = term_APD_data;

    % Calculate Avg. Packet Delay for VoIP Packets
    mean_APD_voip = mean(APD_voip(n_idx, :));
    term_APD_voip = norminv(1 - alfa/2) * sqrt(var(
       APD_voip(n_idx, :)) / num_runs);
    APD_voip_values(n_idx) = mean_APD_voip;
    APD_voip_terms(n_idx) = term_APD_voip;

    % Calculate Avg. Queuing Delay for Data Packets
    mean_APQD_data = mean(APQD_data(n_idx, :));
    term_APQD_data = norminv(1 - alfa/2) * sqrt(var(
       APQD_data(n_idx, :)) / num_runs);
    APQD_data_values(n_idx) = mean_APQD_data;
    APQD_data_terms(n_idx) = term_APQD_data;

    % Calculate Avg. Queuing Delay for VoIP Packets
    mean_APQD_voip = mean(APQD_voip(n_idx, :));
    term_APQD_voip = norminv(1 - alfa/2) * sqrt(var(
       APQD_voip(n_idx, :)) / num_runs);
    APQD_voip_values(n_idx) = mean_APQD_voip;
    APQD_voip_terms(n_idx) = term_APQD_voip;
end

% Average Delay of Data Packets
```

```
figure(1);
grid on;
hold on;
bar(voip_flows, APD_data_values);
errorbar(voip_flows, APD_data_values, APD_data_terms,
    'r.', 'CapSize', 10);
title('Average Delay of Data Packets');
xlabel('Number of VoIP Flows');
ylabel('Average Delay (ms)');
legend('Average Delay', 'Confidence Interval');

% Average Delay of VoIP Packets
figure(2);
grid on;
hold on;
bar(voip_flows, APD_voip_values);
errorbar(voip_flows, APD_voip_values, APD_voip_terms);
title('Average Delay of VoIP Packets');
xlabel('Number of VoIP Flows');
ylabel('Average Delay (ms)');
legend('Average Delay', 'Confidence Interval');

% Create a new figure for Average Queuing Delay of
    Data Packets
figure(3);
grid on;
hold on;
bar(voip_flows, APQD_data_values);
errorbar(voip_flows,APQD_data_values,APQD_data_terms);
title('Average Queuing Delay of Data Packets');
xlabel('Number of VoIP Flows');
ylabel('Average Queuing Delay (ms)');
legend('Average Queuing Delay','Confidence Interval');

% Create a new figure for Average Queuing Delay of
    VoIP Packets
figure(4);
grid on;
hold on;
bar(voip_flows, APQD_voip_values);
errorbar(voip_flows,APQD_voip_values,APQD_voip_terms);
title('Average Queuing Delay of VoIP Packets');
xlabel('Number of VoIP Flows');
ylabel('Average Queuing Delay (ms)');
legend('Average Queuing Delay','Confidence Interval');
```

## Results

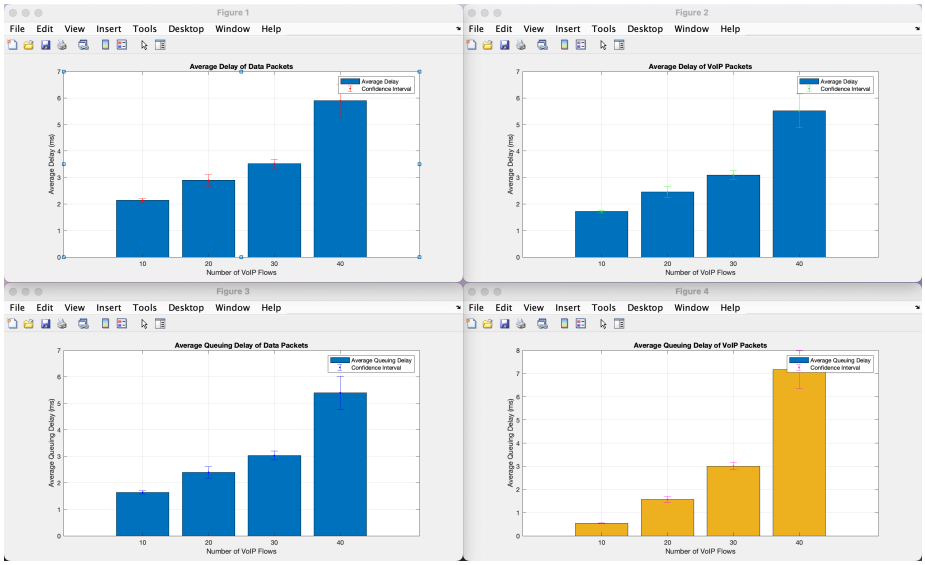



Figure 2.8: (i) the average delay of data packets, (ii) the average delay of VoIP packets, (iii) the average queuing delay of data packets and (iv) the average queuing delay of VoIP packets

**Method:**

Using the new Simulator3, it was simulated a system with shared *First In First Out (FIFO) queue* where both data and VoIP packets are statistically multiplexed without any specific priority for VoIP packets.

After collecting data on average delay and average queuing delay for both data packets and VoIP packets for each number of VoIP flows it was calculated the average delay of data packets, the average delay of VoIP packets, the average queuing delay of data packets and the average queuing delay of VoIP packets as well as their confidence Intervals and then presented the results in four different figures as we can see in Fig. 2.8.

**Justifications:**

Since both *data and VoIP packets* are equally treated in terms of queuing and service, this lead to some interesting observations:

1. If the number of VoIP flows increases, this means that the network congestion will also increase, leading to a higher rivalry between these packets for network resources. Since this competition exist, there will be more delay in the queue as an outcome of it and also the limited capacity of these resources may also result in longer queuing delays.

2. As the number of VoIP flows rises, there may be an increase in the average queuing latency of data packets. Data packets may encounter longer queuing delays when VoIP packets occupy the queue as additional VoIP traffic is added.

3. With an increase in VoIP flows, there should also be an increase in the average queuing delay of VoIP packets. VoIP packets are handled equitably in the shared queue; but, as their quantity grows, they can have to wait longer in queue, leading to increased queuing times.

**Conclusions:**

In conclusion, on this exercise it was observed that as the number of VoIP flows increases, both data and VoIP packets may experience increased queuing delays, affecting their average delays as expected.

### 2.2.2 2 (b)

**Goal:** Repeat experiments 2.a but now with Simulator4. Justify the obtained results. Compare these results with the results of 2.a and draw conclusions concerning the impact on the results when the VoIP service is supported with higher priority than the data service.

#### Code

The code is exactly the same as 2 (a) only with a small difference:

```
1 [PLd, PLv , APDd , APDv , MPDd, MPDv , APQDd, APQDv,
     TT] = Simulator4V2(lambda, C, f, P, n);
```

## Results

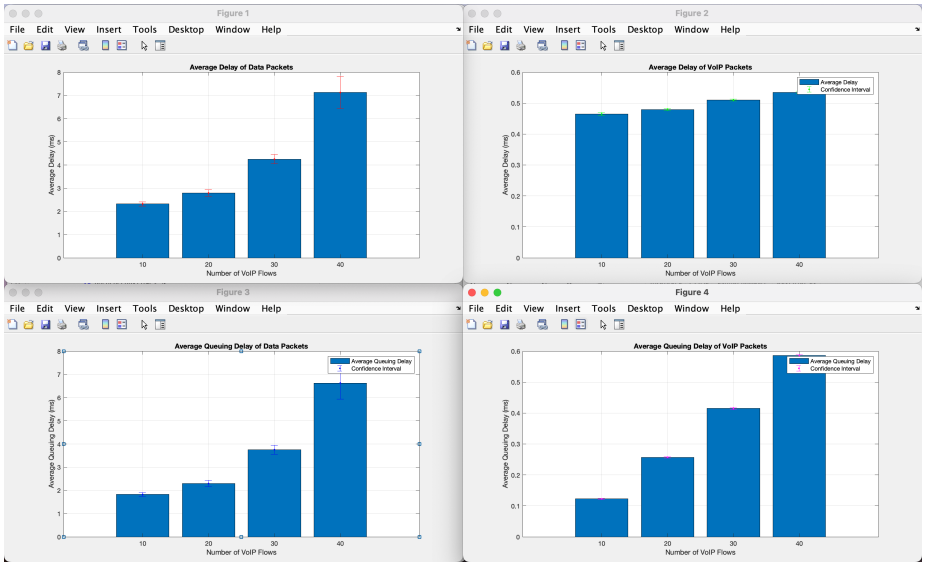



Figure 2.9: (i) the average delay of data packets, (ii) the average delay of VoIP packets, (iii) the average queuing delay of data packets and (iv) the average queuing delay of VoIP packets

**Method:**

Using the new Simulator4 it was simulated a system where the VoIP service is given higher priority.

After collecting data on average delay and average queuing delay for both data packets and VoIP packets for each number of VoIP flows it was calculated the average delay of data packets, the average delay of VoIP packets, the average queuing delay of data packets and the average queuing delay of VoIP packets as well as their confidence Intervals and then presented the results in four different figures as we can see in Fig.

**Expected Results and Justification:**

In Simulator4V2, VoIP packets have higher priority in the queue management when compared to data packets, which means that VoIP packets will experience shorter queuing delays, resulting in lower average queuing delay for VoIP packets compared to Simulator3. However, data packets may experience higher queuing delays when the queue is occupied with VoIP traffic.

The **average delay of data packets** in Simulator4V2 may be higher than in Simulator3, primarily due to the higher priority given to VoIP packets. Data packets may experience more queuing delays when the queue is occupied with VoIP packets.

The **average delay of VoIP packets** in Simulator4V2 is expected to be lower than in Simulator3 because of the higher priority. VoIP packets will experience shorter queuing delays.

The **average queuing delay of data packets** in Simulator4V2 may be higher than in Simulator3 because data packets are less prioritized, and they may have to wait longer in the queue when VoIP packets are present.

The **average queuing delay of VoIP packets** in Simulator4V2 should be lower than in Simulator3 because they have higher priority.

**Conclusions:**

In this exercise we observed that by giving higher priority to VoIP packets they will have reduced delays. However, data packets, experienced increased delays. This shows the importance of prioritization in network design and quality of service.

### 2.2.3 2 (c)

**Goal:** Consider the system modelled by a M/G/1 queueing model with priorities. Determine the theoretical values of the average packet delay for data packets and VoIP packets in all cases of 2.b. Compare the theoretical values with the simulation results of experiments 2.b and draw conclusions.

This was definitely the most challenging exercise, since the results weren't being the ones expected after applying different ways to solve and different formulas.

**Code**

```
% Same parameters as 1 b)

% Calculation of the probability associated with each
    packet size
prob = (1 - (0.19 + 0.23 + 0.17)) / ((109 - 65 + 1) +
    (1517 - 111 + 1));

bytes = 110:130;
bytes_times_8 = bytes * 8;

for n = voip_flows
    S_data = zeros(1, length(bytes));  % E[S] for data
        packets
    S2_data = zeros(1, length(bytes)); % E[S^2] for
        data packets
```

```
12
13      S_voip = zeros(1, length(bytes));  % E[S] for VoIP
            packets
14      S2_voip = zeros(1, length(bytes)); % E[S^2] for
            VoIP packets
15
16      for j = 1:length(bytes)
17          if j == 1
18              S_data(j) = (bytes_times_8(j)) / (C *
                    10^6) * 0.19;
19              S2_data(j) = ((bytes_times_8(j)) / (C *
                    10^6))^2 * 0.19;
20          elseif j == 110 - 64 + 1
21              S_data(j) = (bytes_times_8(j)) / (C *
                    10^6) * 0.23;
22              S2_data(j) = ((bytes_times_8(j)) / (C *
                    10^6))^2 * 0.23;
23          elseif j == 1518 - 64 + 1
24              S_data(j) = (bytes_times_8(j)) / (C *
                    10^6) * 0.17;
25              S2_data(j) = ((bytes_times_8(j)) / (C *
                    10^6))^2 * 0.17;
26          else
27              S_data(j) = (bytes_times_8(j)) / (C *
                    10^6) * prob;
28              S2_data(j) = ((bytes_times_8(j)) / (C *
                    10^6))^2 * prob;
29          end
30
31          % Adjust weights based on the number of VoIP
                flows
32          S_voip(j) = S_data(j) * (1 - 1/n);
33          S2_voip(j) = S2_data(j) * (1 - 1/n);
34
35      end
36
37      ES_data = sum(S_data);
38      ES2_data = sum(S2_data);
39      ES_voip = sum(S_voip);
40      ES2_voip = sum(S2_voip);
41
42      % Calculate traffic intensity for data and VoIP
43      rho_data = lambda * ES_data;
44      rho_voip = lambda * ES_voip;
45
```

```
46      % Calculate Average Delay and Average Queuing
            Delay
47      W_data = (lambda * ES2_data + lambda * ES_data^2)
            / (2 * (1 - rho_data)) + ES_data;
48      Wq_voip = (lambda * ES2_voip + lambda * ES_voip^2)
            / (2 * (1 - rho_voip)) + ES_voip;
49
50      fprintf('For n = %d VoIP Flows:\n', n);
51      fprintf('APD data n%d (milliseconds) = %.4f\n', n,
            W_data * 10^3);
52      fprintf('AQD voip n%d (milliseconds) = %.4f\n', n,
            Wq_voip * 10^3);
53      fprintf('\n');
54  end
```

**Results**

Even though the team could not produced the wished results, theoretically the average packet delay results in exercise 2.c is expected to be lower compared to the simulation results in exercise 2.b. In exercise 2.b, it is being simulated a system where the VoIP packets have priority comparing to the data ones, and as previously mentioned, VoIP packets will experience less delay compared to data packets in the simulation. Now in the exercise 2 c), it is being used a theoretical model of a prioritized M/G/1 queueing system which means that "on paper", this is the ideal system, where there is a perfect prioritization, resulting indeed in even lower packet delay for VoIP packets compared to the simulations previously performed.

Concluding, it should be expected that the theoretical values of the average packet delay for VoIP packets would be lower in 2 c) compared to the simulation results in 2.b, confirming the impact of prioritization on reducing packet delay for VoIP packets and simulating this "ideal system".

# Acrónimos

**ECT** Engenharia de Computadores e Telemática

**MDRS** Modelação e Desempenho de Redes e Serviços

**APD** Average Packet Delay

**BER** Bit Error Rate

**FIFO** First In First Out

# Bibliography

[1] Tech Target, *Throughput*, `https://www.techtarget.com/searchnetworking/definition/throughput`, Accessed: [10/2023], 2023.