MINI-PROJECT 1:

PERFORMANCE EVALUATION OF POINT-TO-POINT LINKS SUPPORTING PACKET SERVICES

Pedro Caio (105127) and Miguel Ferreira (93419)

Auto-evaluation: 50 % / 50 %

29/10/2024

Task 1

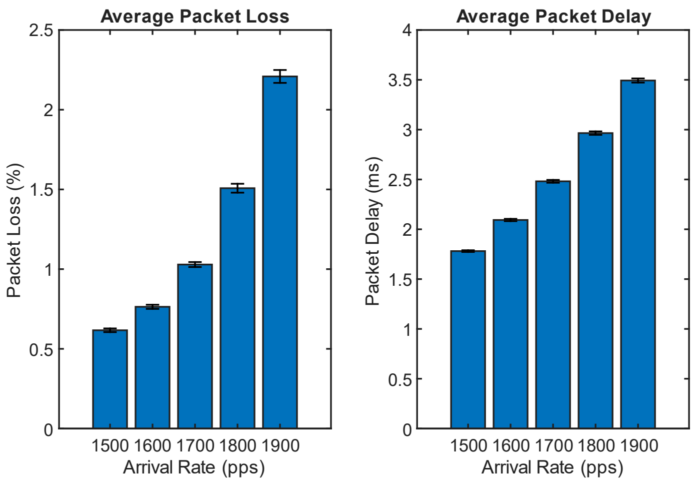
In this task, use *Sim2* developed in Task 6 of the Practical Guide. Consider always the capacity of the link $C = 10$ Mbps and the size of the queue $f = 10.000$ Bytes. When performance parameters are estimated by simulation, present the results based on 20 runs of the simulator (with a stopping criterion of $P = 100.000$ on each run) and with 90% confidence intervals.

1.a. (Evaluation weight: 10%)

Estimate by simulation the average packet delay and the average packet loss parameters when the bit error rate of the link is $b = 10^{-6}$ and for the arrival rate values $\lambda = 1500, 1600, 1700, 1800$ and 1900 pps. Plot the results in bar charts with the confidence intervals in error bars (The MathWorks, Inc., 2024). Justify the results and draw all relevant conclusions.

Answer:

After running the simulations, the following results were got



Comentado [PC1]: Atualizar as barras de erro para preto

From the results it's possible to interpret the following:

Average Packet Loss

- **Trend**: The average packet loss increases with higher arrival rates.
- **Explanation**: As the arrival rate increases, more packets are sent to the queue, leading to higher chances of queue overflow and packet drops.

Average Packet Delay

- **Trend**: The average packet delay increases with higher arrival rates.
- **Explanation**: Higher arrival rates cause more congestion in the queue, leading to longer waiting times for packets.

Conclusion

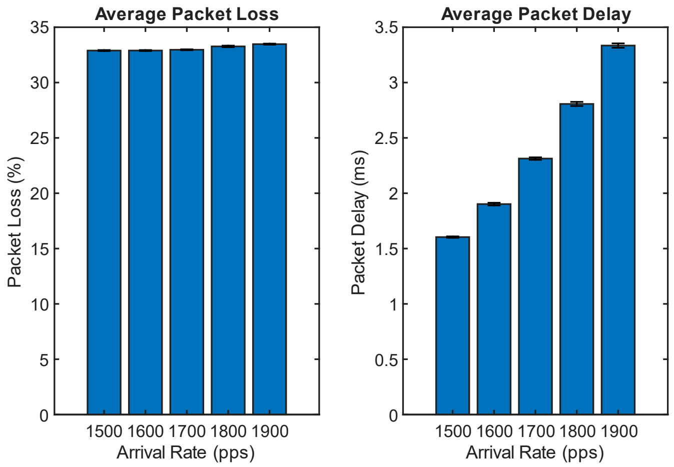
The increase in packet loss with higher arrival rates is primarily due to queue overflow as the system becomes more congested, this conclusion is collaborated by also an increase on the average packet delays which means that each packet takes more time to be sent (i.e. the mean queue size increases).

1.b. (Evaluation weight: 10%)

Repeat experiment **1.a** considering now a bit error rate $b = 10^{-4}$. Justify the differences between these results and the results of experiment **1.a** and draw all relevant conclusions.

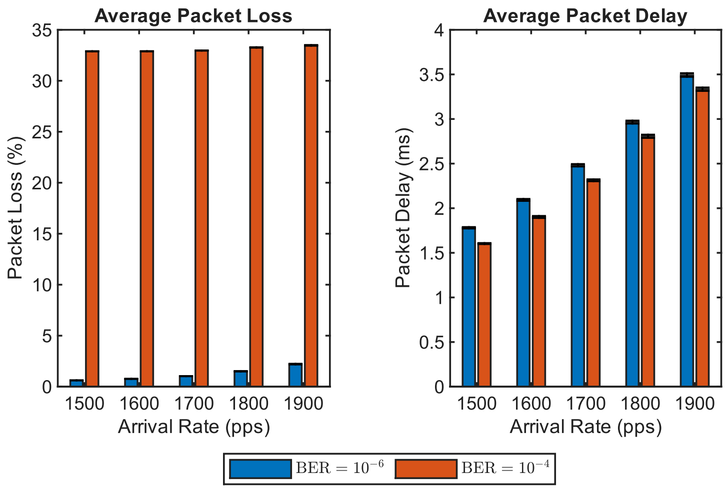
Answer:

Repeating the simulations but now with a higher bit error rate, $b = 10^{-4}$.



Comentado [PC2]: Atualizar as barras de erro

Comparing both results



Comentado [PC3]: Adicionar as barras de erro

it’s possible to observe:

Average Packet Loss

- **Trend**: The average packet loss remains relatively stable with a slight increase at higher arrival rates.
- **Explanation**: At a high BER of (10^{-4}), the packet loss is primarily driven by bit errors during transmission. The high probability of bit errors means that a significant number of packets are corrupted and dropped, regardless of the arrival rate. Thus, showing that the impact of queue overflow is less significant compared to the dominant effect of bit errors.

Average Packet Delay

- **Trend**: The average packet delay increases with higher arrival rates.
- **Explanation**: The average packet delay shows a similar increasing trend in both cases, this shows that the delay is more influenced by queue dynamics than an increase in BER because while bit errors dominate packet loss, they don't influence how the queue is filled being only changed by the arrival rate.

Conclusion

The packet loss is much higher at high BER due to bit errors, while it increases with arrival rates at low BER due to queue overflow. The average packet delays at low BER and high BER remain similar as they are more influenced by queue dynamics.

This comparison highlights the significant impact of an increase on the BER on network performance, particularly in terms of packet loss. The high BER scenario is dominated by bit errors, leading to high packet loss, while the low BER scenario is mostly influenced by queue dynamics and congestion.

A way to fix the high average packet loss was for instance to change the distribution of the packet sizes such that it would be smaller.

Comentado [PC4]: Sem certeza

1.c. (Evaluation weight: 10%)

Determine the theoretical average packet loss (in %) only due to the bit error rate for $b = 10^{-6}$ and $b = 10^{-4}$. Present and explain the MATLAB code developed for these calculations. Compare these values with the results obtained in **1.a** and **1.b**. What do you conclude?

Answer:

To calculate the theoretical average packet loss, it is considered that the bit has only two states available, therefore it can be considered that each bit is a r.v. that follows a binomial distribution whose probability mass function is:

$$f(k|n,p) = P(X = k) = \binom{n}{k} p^k (1-p)^{n-k}, \quad \text{for } k = 0,1,2,\dots,n.$$

In other words, let $\{X\}$ denote the number of bits errors, therefore $\{X\} \sim \mathrm{B}(n = \mathrm{n.\,bits}, p = \mathrm{BER})$.

A packet is considered lost if it has at least one bit with error, so the quantity of interest is the probability of at least one error that, by the application of the 2nd axiom of probability (i.e. $\sum_i P_i = 1$), is the same as

$$P(X \geq 1) = 1 - P(X \geq 1)^c = 1 - P(X < 1) = 1 - P(X = 0),$$

were

$$P(X = 0) = f(0|n = \mathrm{n.\,bits}, p = \mathrm{BER}) = P_n \times (1-p)^n,$$

and that the distribution of the number of bits is 19% for 64 bytes, 23% for 110 bytes, 17% for 1518 bytes and an equal probability for all other values (i.e., from 65 to 109 and from 111 to 1517).

After computing the quantity of interest, $P(X \geq 1)$, the theoretical values for the average packet loss for $b = 10^{-6}$ is $0{,}4937$ and for $b = 10^{-4}$ is $32{,}8278$, the code is presented on appendix A, comparing with the computed values:

Table 1 - Packet loss results comparison for $\mathrm{BER} = 10^{-6}$

λ	Computed Packet loss	Theoretical Packet loss

1500	0.61645	0,4937
1600	0.76369	
1700	1.0288	
1800	1.5076	
1900	2.2087	

- **Computed Packet Loss**: The computed packet loss increases as the arrival rate (λ) increases. This is expected because higher arrival rates lead to more packets being sent through the network, increasing the likelihood of queue overflow and packet drops.
- **Theoretical Packet Loss**: The theoretical packet loss remains constant at $0{,}49371\%$ for all arrival rates. This value is derived from the bit error rate and does not account for queue dynamics.
- **Comparison**: The computed packet loss is higher than the theoretical value for all arrival rates. This discrepancy is due to the additional packet loss caused by queue overflow, which is not considered in the theoretical calculation. As the arrival rate increases, the queue becomes more congested, leading to higher packet loss.

Table 2 - Packet loss results comparison for $\mathrm{BER} = 10^{-4}$

λ	Computed Packet loss	Theoretical Packet loss
1500	32.893	32,8278
1600	32.894	
1700	32.963	
1800	33.266	
1900	33.472	

- **Computed Packet Loss**: The computed packet loss remains relatively stable, with a slight increase as the arrival rate increases. This stability indicates that the packet loss is primarily driven by the high bit error rate rather than queue overflow.
- **Theoretical Packet Loss**: The theoretical packet loss is also high at $32{,}8278\%$ for all arrival rates, reflecting the high bit error rate.
- **Comparison**: The computed packet loss is very close to the theoretical value for all arrival rates. This suggests that the high bit error rate is the dominant factor causing packet loss where the impact of queue overflow is minimal. The slight increase in computed packet loss with higher arrival rates could be due to minor contributions from queue dynamics, but the overall effect is dominated by bit errors.

Concluding at low BER ($b = 10^{-6}$) packet loss increases with higher arrival rates due to queue overflow. The computed packet loss is higher than the theoretical value, indicating that queue dynamics play a significant role. At high BER ($b = 10^{-4}$), packet loss remains relatively stable across different arrival rates, with the computed values closely matching the theoretical values. This indicates that the high bit error rate is the primary cause of packet loss, overshadowing the effects of queue overflow.

Task 2

Consider the event driven simulator *Sim3* developed in *Task 7* of the Practical Guide. Start by developing a new version of *Sim3*, named *Sim3A*, to estimate the same performance parameters as *Sim3* and to consider that the link introduces a bit error rate given by *b* (which should be a new input parameter of *Sim3A*).

2.a. (Evaluation weight: 5%)

Present the developed MATLAB function of Sim3A highlighting and justifying the introduced changes. Using Sim3A, estimate all performance parameters when $\lambda = 1500$ pps, $C = 10$ Mbps, $f = 1.000.000$ Bytes, $b = 10^{-5}$ and $n = 10, 20, 30, 40$ VoIP flows, based on 20 runs of the simulator (with a stopping criterion of $P = 100.000$ on each run) and with 90% confidence intervals.

Answer:

```
        case DEPARTURE          % If first event is a DEPARTURE
            if (rand() < (1-b)^(PacketSize*8))
                if (PacketType == DATA) % Data
                    TRANSBYTES_DATA= TRANSBYTES_DATA + PacketSize;
                    DELAYS_DATA = DELAYS_DATA + (Clock - ArrInstant);
                    if Clock - ArrInstant > MAXDELAY_DATA
                        MAXDELAY_DATA= Clock - ArrInstant;
                    end
                    TRANSPACKETS_DATA= TRANSPACKETS_DATA + 1;
                else % VoIP
                    TRANSBYTES_VOIP= TRANSBYTES_VOIP + PacketSize;
                    DELAYS_VOIP= DELAYS_VOIP + (Clock - ArrInstant);
                    if Clock - ArrInstant > MAXDELAY_VOIP
                        MAXDELAY_VOIP= Clock - ArrInstant;
                    end
                    TRANSPACKETS_VOIP= TRANSPACKETS_VOIP + 1;
                end
            else
                 if (PacketType == DATA)         % Data Packet
                    LOSTPACKETS_DATA = LOSTPACKETS_DATA + 1;
                else                            % VoIP Packet
                    LOSTPACKETS_VOIP = LOSTPACKETS_VOIP + 1;
                end
            end
```

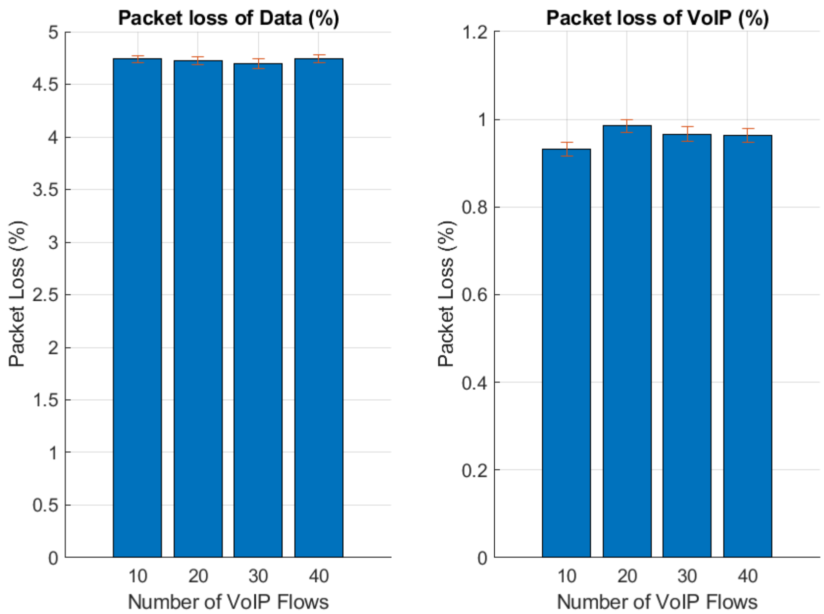
The critical change happens in *line 98* where the bit error rate *b* is introduced. The expression $(1-b)^{PacketSize*8}$ represents the probability of a packet being error-free. To simulate a real scenario, a random number is generated. If this random number is less than the calculated probability, it indicates that the packet is free of errors and can proceed normally. Else, the packet does have an error, and it will be accounted for in variables *LOSTPACKETS_DATA* and *LOSTPACKETS_VOIP*, depending on the packet type.

2.b. (Evaluation weight: 5%)

Present the simulation results of 2.a concerning the packet loss of each service (data and VoIP) in bar charts with the confidence intervals in error bars. Justify the results and draw all relevant conclusions.

Answer:

Executing the simulations the following results were obtained:



From the results it's possible to interpret the following:

1) **The average packet loss stays roughly the same with increasing number of VoIP flows.**

Explanation: The small amount of added VoIP flows is not enough for the packet loss to increase. Considering the big queue size of 1MB, the link capacity of 10 Mbps, and the constant *bit error rate* it is expected that the increase from 10 to 40 VoIP won't make a difference in the resulting packet loss.

Comentado [PC5]: A perda de pacotes é estável dado que a fonte principal de perda de pacotes é atravez do BER e não a partir do overflow da fila

2) **The average packet loss for data packets is considerably higher than for VoIP packets.**

Explanation: The average data packet size is of 620.02 Bytes, while the average VoIP packet size is of 120 Bytes, making them roughly 5 times smaller, Looking at the plots, the VoIP packet loss is also around 5 times less, and considering the differences in packet sizes, this result is to be expected.

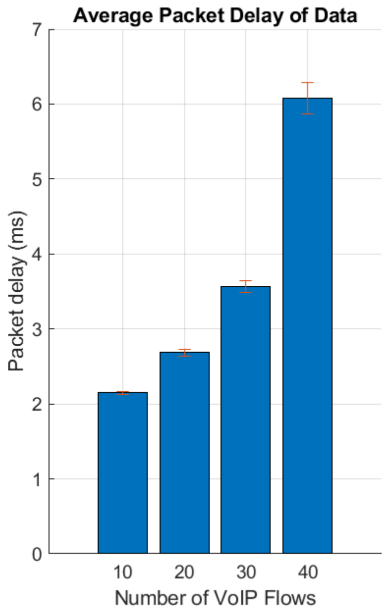
Comentado [PC6]: A relação entre o tamanho e a probabilidade de o packet ser perdido devido a erros não é linear, logo não sei se deva-se colocar que a relação seja linear

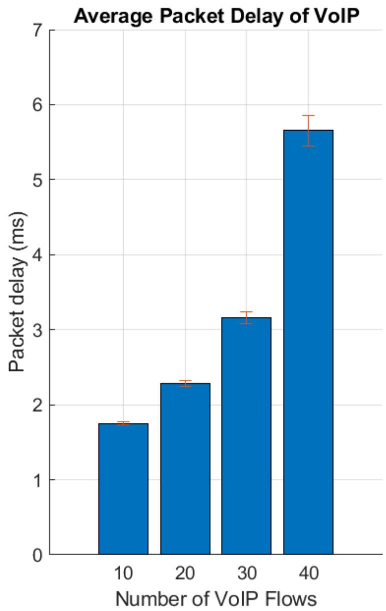
2.c. (Evaluation weight: 5%)

Present the simulation results of 2.a concerning the average packet delay of each service in bar charts with the confidence intervals in error bars. Justify the results and draw all relevant conclusions.

Answer:

Executing the simulations the following results were obtained:





From the results it's possible to interpret the following:

1) **Average packet delay for data packets is slightly higher than for VoIP packets.**

   Explanation: Generally, bigger packets cause higher average delays. As said before, the data packets are around 5 times bigger than VoIP packets, so the higher delay for data packets is to be expected.

2) **Average packet delay exponentially increases with the increase in VoIP flows.**

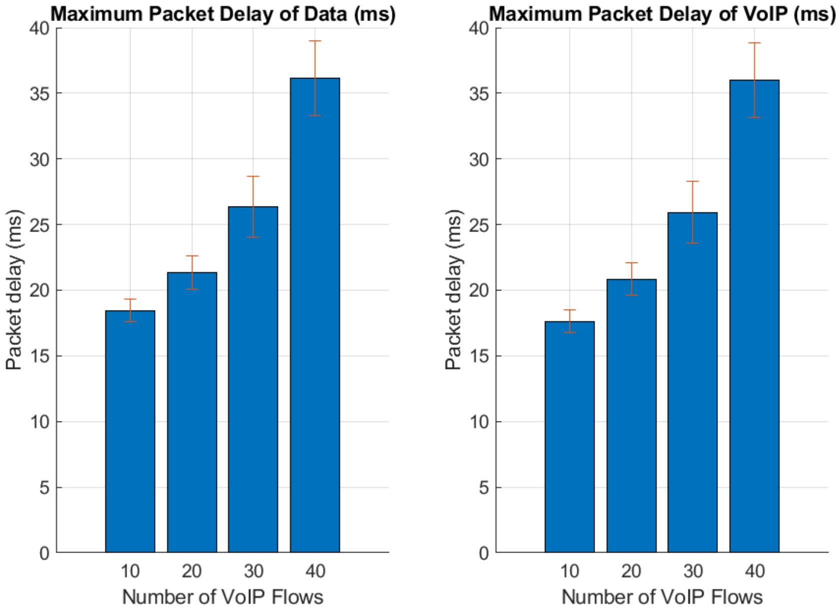
   Explanation: As the number of VoIP packets increases from 10 to 40, it is expected that the average delay will also increase due to greater queueing delays from higher traffic load on the link.

2.d. (Evaluation weight: 5%)

Present the simulation results of 2.a concerning the maximum packet delay of each service in bar charts with the confidence intervals in error bars. Justify the results and draw all relevant conclusions.

Answer:

Executing the simulations the following results were obtained:



From the results it's possible to interpret the following:

1) **The maximum packet delay is the around the same for both data and VoIP packets.**

   Explanation: Since there is no prioritization in the queue, in cases where the system is very congested, the delays for both data and VoIP will be the same, even though the average delays for both are not the same.

2) **The maximum packet delay exponentially increases with the increase in VoIP flows.**

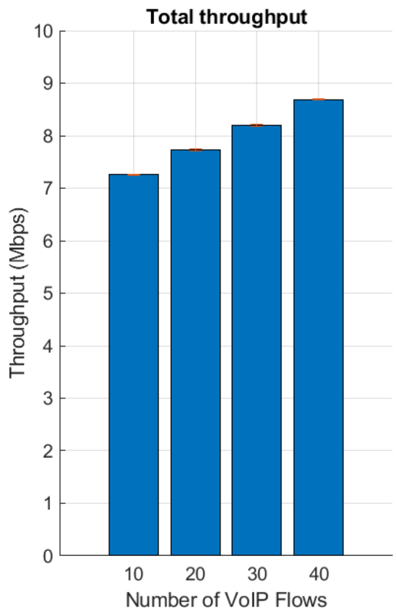
   Explanation: As the number of VoIP packets increases from 10 to 40, it is expected that the maximum delay will also increase due to greater queueing delays from higher traffic load on the link.

2.e. (Evaluation weight: 5%)

Present the simulation results of 2.a concerning the total throughput in bar charts with the confidence intervals in error bars. Justify the results and draw all relevant conclusions.

Answer:

Executing the simulations the following results were obtained:



From the results it's possible to interpret the following:

1) **The total throughput linearly increases with the increase of VoIP flows.**

   Explanation: It is expected that the throughput increases as the number of VoIP flows increase. This means that system is capable handling the additional traffic effectively. However, it is approaching the link capacity (10 Mbps), so if more VoIP flows were to be added, it might not be able to handle it as well, potentially causing more packet loss.

2.f. (Evaluation weight: 5%)

Determine the theoretical value of the total throughput for all cases simulated in experiment **2.a.** Present and explain the MATLAB code developed for these calculations. Compare these values with the results obtained in **2.e.** What do you conclude?

Answer:

To calculate the theoretical throughput of the system in Mbps, we need to consider the bit error rate (BER) and the utilization of the line. In other words, the throughput can be expressed as:

$$\text{Throughput} = \frac{\bar{n}}{10^6} \times \lambda \times P(X = 0),$$

where $\bar{n}$ denotes the average size of the packets (bits), λ the number of packets per second (pps) and $P(X = 0)$ the probability of a packet being transmitted without any errors. From the previous theoretical explanation, it was determined that the probability of a packet being transmitted without errors in given by:

$$P(X = 0) = f(0|n = \text{n. bits}, p = \text{BER}) = P_n \times (1 - p)^n = (1 - p)^{\bar{n}}.$$

As in this system, we have two types of packets: data and VoIP. Considering the mean lengths of the packets being 620 bytes for data and 120 bytes for VoIP, we have:

$$P_{\text{data}}(X = 0) = (1 - \text{BER})^{620\times 8}, \qquad P_{\text{VoIP}}(X = 0) = (1 - \text{BER})^{120\times 8}.$$

The throughput for data and VoIP packets is then:

$$\text{Throughput}_{\text{data}} = \frac{620 \times 8}{10^6} \times 1500\,,$$

$$\text{Throughput}_{\text{VoIP}} = \frac{120 \times 8}{10^6} \times 50\,,$$

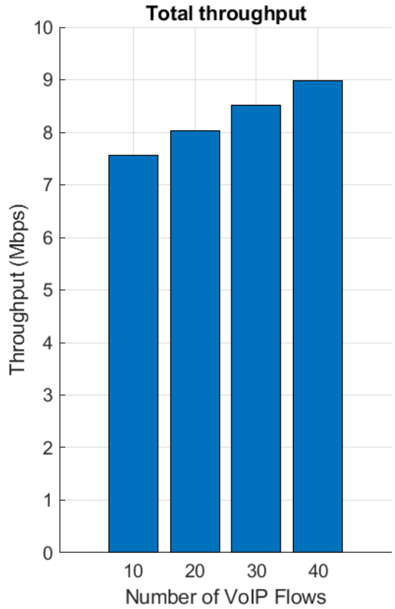
where $\lambda_{\text{data}} = 1500$ pps and $\lambda_{\text{VoIP}} = 50$ pps per VoIP flow.

Finally, the total theoretical throughput, considering the probability of no errors, is simply the sum of the throughput of all the flows:

$$\begin{aligned}\text{Throughput} &= \text{Throughput}_{\text{VoIP}} \times N_{\text{VoIP}} \times P_{\text{VoIP}}(X = 0) \\ &\quad + \text{Throughput}_{\text{data}} \times P_{\text{data}}(X = 0).\end{aligned}$$

Note: the respective code is displayed in appendix c)

With these calculations, the following results were obtained:



From the results it's possible to interpret the following:

a) **Theoretical total throughput is similar but slightly higher than the simulation throughput.**

   Explanation: This slight difference may happen due to the delays and congestion in the queuing. This makes the result expected since queue size was not taken into account for the theoretical total throughput calculations.

Task 3

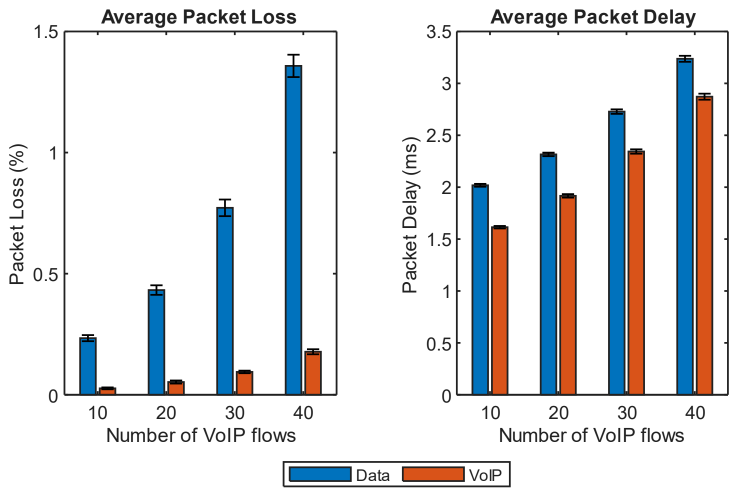
Consider the event driven simulators *Sim3* and *Sim4* developed in Task 7 of the Practical Guide. In all experiments of this task, consider the cases of $\lambda = 1500$ pps, $C = 10$ Mbps, $f = 10.000$ Bytes and $n = 10, 20, 30$ and 40 VoIP flows. All simulation results should be based on 20 runs of the simulator with a stopping criterion of $P = 100.000$ on each run to compute the estimated values and the 90% confidence intervals.

3.a. (Evaluation weight: 10%)

Use *Sim3* to estimate the average packet delay and average packet loss of each service (data and VoIP). Recall that in *Sim3*, both services are statistically multiplexed in a single FIFO queue. Present each of the four performance parameters in a bar chart with the confidence intervals in error bars. Justify the differences in the performance values obtained for each service and draw all relevant conclusions.

Answer:

Executing the simulations the following results were obtained:



by observing the results:

Average Packet Loss

- **Trend:** The average packet loss for data packets increases as the number of VoIP flows increases. Similarly, the average packet loss for VoIP packets also increases with the number of VoIP flows yet is considerable smaller to the average packet loss of the data packets.
- **Explanation:** As the number of VoIP flows increases, the competition for the limited queue space intensifies. Starting with 10 VoIP flows and as the mean time of arrival after the first VoIP packet is 10 milliseconds, for each flow there arrives a total of 100 VoIP pps, this in turn makes that after 20 flows there arrives more VoIP packets than data packets (1500 pps), the cumulative effect of multiple VoIP flows increases the overall load on the queue making it more probable for any arriving packet to be dropped due to buffer overflow, yet the VoIP packets are less impacted than the data packets due to their smaller average size explaining the discrepancy on the magnitudes of the bars although they have the same priority.

Average Packet Delay

- **Trend:** Both the average packet delay for data packets and VoIP packets increases in a similar way as the number of VoIP flows increases, but there exists a discrepancy on the magnitudes being the VoIP lower than the data.
- **Explanation:** Although VoIP packets also experience increased delay with more VoIP flows, the increase is less pronounced compared to data packets due to, once again, their smaller size, this in turn makes them quicker to transmit lowering the average time delay, even when the queue is congested.

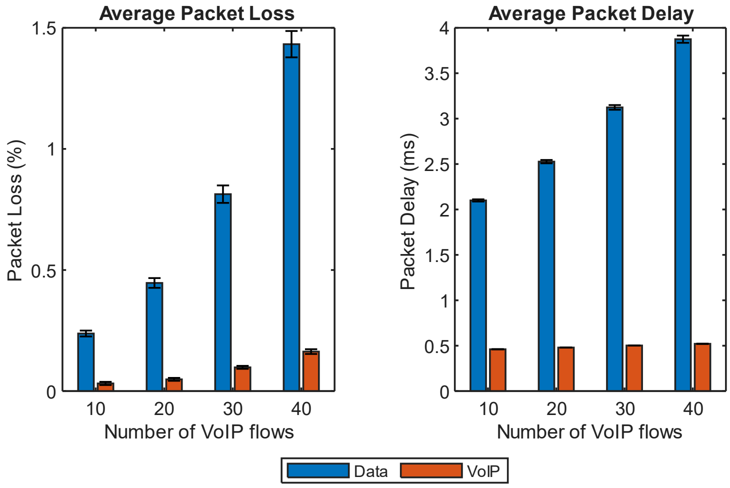
Conclusion

The addition of VoIP flows impacts both packet loss and delay for data and VoIP packets, yet data packets are more adversely affected due to their larger size, for not being able to fit into a congested queue culminating in an increase in packet loss, and for being slower to transmit culminating in a higher average delay in comparison to the VoIP packets. These results highlight a possible unintended effect of no prioritization where, at some specific states of the system, it may prioritize one packet over the other, in this case prioritizing smaller packets when the queue is near full capacity merely because they fit on the available space.

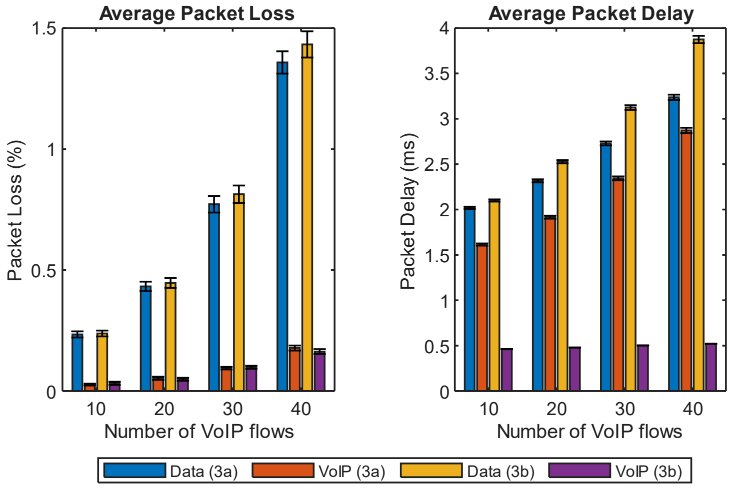
3.b. (Evaluation weight: 10%)

Use *Sim4* to estimate the same performance parameters as in **3.a**. Recall that in *Sim4*, VoIP service has higher priority than data service. Present each of the four performance parameters in a bar chart with the confidence intervals in error bars. Justify the differences in the performance values obtained for each service, and the differences between these results and the results of experiment **3.a**. Draw all relevant conclusions.

Answer:



Making a comparison plot



observing the results:

Average Packet Loss

- **Trend:** Both the average packet loss for data and VoIP packets with and without prioritization increase with the number of flows, being in both cases of prioritization the VoIP having a considerably less average packet loss in comparison to its data packet counterpart.

- **Explanation:** The addition of VoIP priority does not change the packet loss for any of the types of packets, this is due to the loss being attributed to the queue overflow, as mentioned on the previous answer, due to the system already prioritizing, inadvertently, the smaller packets, the addition of a VoIP priority does not change the results.

Average Packet Delay

- **Trend:** The average packet delay for both data and VoIP packets increases with the number of VoIP flows yet the addition of VoIP priority decreases considerably the average delay for the VoIP packets and increases slightly the average data packet delay.
- **Explanation:** Without prioritization, both data and VoIP packets experience increased delays as the queue becomes more congested. Data packets, being larger, take longer to transmit, resulting in higher delays compared to VoIP packets. By prioritizing VoIP packets ensures they are transmitted quickly, maintaining a low average delay. Data packets, however, experience longer delays as they must wait for the VoIP packets to be transmitted first, especially as the number of VoIP flows increases.

Conclusion

Prioritizing VoIP packets significantly improves their performance but at the cost of a slightly longer delay for data packets, the packet loss remains with no significant changes as the packet can only be dropped due to queue overflow and as written in the previous answer, the queue already prioritizes VoIP packets due to their smaller size.

3.c. (Evaluation weight: 5%)

Develop a new version of *Sim4*, named *Sim4A*, to estimate the same performance parameters as *Sim4* changing the queue packet discard algorithm as follows: arriving VoIP packets are always accepted in the queue (if there is enough space) but arriving data packets are accepted in the queue only if the total queue occupation does not become higher than p (in %) of the queue size (parameter p should be a new input parameter of *Sim4A*). Present the developed MATLAB function of *Sim4A* highlighting and justifying the introduced changes.

Answer:

```
69                else
70                    if QUEUEOCCUPATION + PacketSize <= f * (p/100)
71                        QUEUE= [QUEUE;PacketSize , Clock, ARRIVAL];
72                        QUEUEOCCUPATION= QUEUEOCCUPATION + PacketSize;
73                    else        % Data packet is lost if the queue is full
74                        LOSTPACKETS= LOSTPACKETS + 1;
75                    end
76                end
```

The only difference happens in *line 70* where the new parameter *p* is taken into account. Multiplying its value with *f* (queue size) makes it so that data packets will be discarded will be discarded unless the queue occupation is lower than *p* (in %).

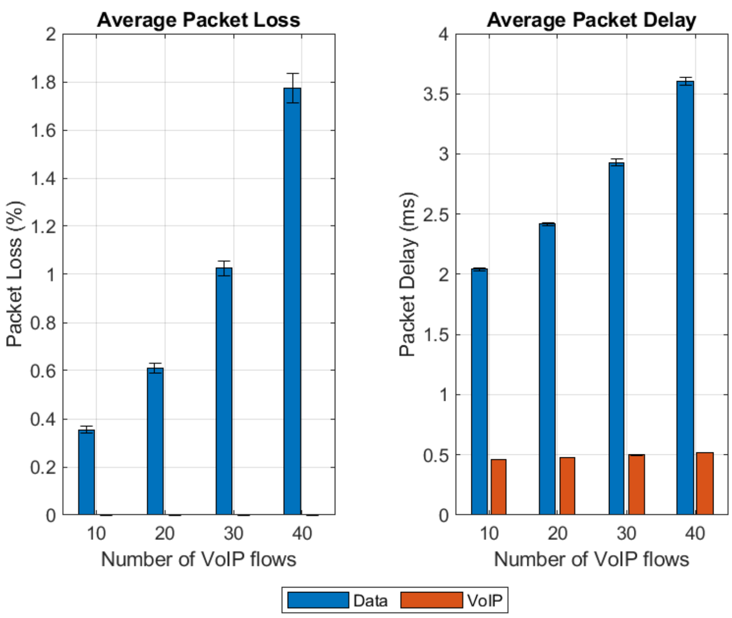
3.d. (Evaluation weight: 5%)

Use simulator *Sim4A* to estimate the same performance parameters as in **3.a** and **3.b** for $p = 90\%$. Justify the differences in the performance values obtained for each service, and the differences between these results and the results of experiment **3.b**. Draw all relevant conclusions.

Answer:

Executing the simulations the following results were obtained:

Comentado [PC7]: Podes utilizar o meu código que corrige o desenho das barras de erro em gráficos de barras agrupados.



From the results it's possible to interpret the following:

1) **The average VoIP packet loss is always at 0 percent.**

   Explanation: Since VoIP packets have 10% of the queue reserved to them as well as having priority, and since there is no *bit error rate*, it is unlikely for the queue to discard any VoIP packet, making these results fully expectable.

2) **The average data packet loss exponentially increases.**

   Explanation: Inversely, since there is less of the queue for the data packets, it is also expected that the packet loss increases as there are more VoIP flows.

3) **VoIP packet delay only slightly increases compared to data packets.**

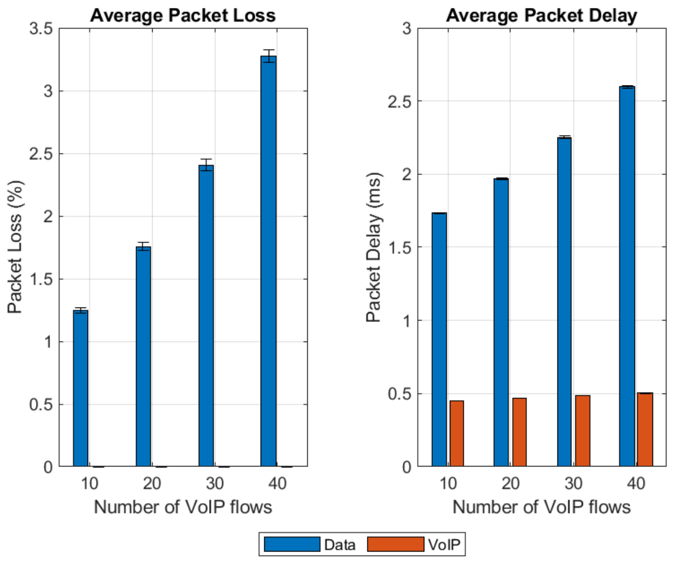
   Explanation: The data packet delay drastically increases compared to VoIP packets; this happens because the VoIP packets have priority over data packets. The VoIP packets being much smaller also helps to keep the delay lower.

3.e. (Evaluation weight: 5%)

Repeat experiment **3.d** considering now $p = 60\%$. Justify the differences in the performance values obtained for each service, and the differences between these results and the results of experiments **3.b** and **3.d**. Draw all relevant conclusions.

Answer:

Executing the simulations the following results were obtained:



Comentado [PC8]: Podes utilizar meu código para as barras de erro ficarem corretamente colocadas

From the results it's possible to interpret the following:

1) **The average VoIP packet loss stays at 0 percent.**

   Explanation: This is fully expected, for the same reasons as in 3.d. With p = 60%, it is even more unlikely that a VoIP packet is discarded, as it has an even bigger reserved queue.

2) **The average data packet loss more than doubles from 3.d.**

   Explanation: Again, for the same reasons as in 3.d this is fully expected. With an even smaller available queue, more data packets will be discarded.

3) **The average VoIP packet delay has stayed the same from 3.d.**

   Explanation: This means that 10% of the queue reserved for VoIP packets was already enough to handle the amount of flows in this simulation.

4) **The average data packet delay has decreased from 3.d.**

   Explanation: Since the queue size for data packets is only at 60%, data packets have a smaller portion of the queue available to them, reducing the maximum waiting time for each packet in the queue, even though the number of lost packets is higher.

References

The MathWorks, Inc. (2024). *Bar Chart with Error Bars*. Retrieved from MathWorks Help Center: https://www.mathworks.com/help/matlab/creating_plots/bar-chart-with-error-bars.html

Appendix:

a) 1.c. code

```
packet_sizes = [64, 110, 1518, 65:109 111:1517];

packet_size_probabilities = [.19, .23, .17, ...
    repelem((1 - (.19 + .23 + .17))/length([65:109 111:1517]), length([65:109
111:1517]))];

packet_sizes_bits = packet_sizes * 8;

% Calculating the probabilities of no bit error for each state
%P_no_bit_error = zeros(length(packet_sizes));

P_no_bit_error_1a = binopdf(0, packet_sizes_bits, b);
P_no_bit_error_1b = binopdf(0, packet_sizes_bits, b_1b);


P_one_bit_error_1a = 1 - sum(packet_size_probabilities .* P_no_bit_error_1a);
P_one_bit_error_1b = 1 - sum(packet_size_probabilities .* P_no_bit_error_1b);

disp(P_one_bit_error_1a*100)
disp(P_one_bit_error_1b*100)
```

b) Sim3A:

Comentado [PC9]: Colocar o código das simulações, digo isto pois lembro-me do professor especificamente referir que queria o código inteiro e nas respostas só se coloca o código que se modificou.

Comentado [PC10R9]: Fazer:

mydoc = publish("Path\to\Sim3A.m","doc");
winopen(mydoc)

Comentado [PC11R9]: Irá dar erro, mas é por ser uma função (i.e. não tem os inputs definidos), deverá abri um word com o código na mesma onde o podes copiar para aqui.

```
function [PL_data, PL_VoIP , APDd, APDv, MPDd, MPDv , TT] = Sim3A(lambda,C,f,P, n, b)
% INPUT PARAMETERS:
%  lambda - packet rate (packets/sec)
%  C      - link bandwidth (Mbps)
%  f      - queue size (Bytes)
%  P      - number of packets (stopping criterium)
% OUTPUT PARAMETERS:
%  PL   - packet loss (%)
%  APD  - average packet delay (milliseconds)
%  MPD  - maximum packet delay (milliseconds)
%  TT   - transmitted throughput (Mbps)
```

```
%Packet type
DATA = 0;
VOIP = 1;

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

%State variables:
STATE = 0;          % 0 - connection is free; 1 - connection is occupied
QUEUEOCCUPATION= 0; % Occupation of the queue (in Bytes)
QUEUE= [];          % Size and arriving time instant of each packet in the queue

%Statistical Counters:
TOTALPACKETS_DATA= 0;     % No. of packets arrived to the system
LOSTPACKETS_DATA= 0;      % No. of packets dropped due to buffer overflow
TOTALPACKETS_VOIP= 0;     % No. of packets arrived to the system
LOSTPACKETS_VOIP= 0;      % No. of packets dropped due to buffer overflow

TRANSPACKETS_DATA = 0; % No. of transmitted data packets
TRANSPACKETS_VOIP = 0; % No. of transmitted voip packets
TRANSBYTES_DATA = 0;   % Sum of the Bytes of transmitted data packets
TRANSBYTES_VOIP = 0;   % Sum of the Bytes of transmitted voip packets
DELAYS_DATA = 0;             % Sum of the delays of transmitted data packets
DELAYS_VOIP = 0;             % Sum of the delays of transmitted voip packets
MAXDELAY_DATA = 0;           % Maximum delay among all transmitted data packets
MAXDELAY_VOIP = 0;

TRANSPACKETS= 0;     % No. of transmitted packets
TRANSBYTES= 0;       % Sum of the Bytes of transmitted packets
DELAYS= 0;           % Sum of the delays of transmitted packets
MAXDELAY= 0;         % Maximum delay among all transmitted packets

% Initializing the simulation clock:
Clock= 0;

% Initializing the List of Events with the first ARRIVAL:
tmp= Clock + exprnd(1/lambda);
EventList = [ARRIVAL, tmp, GeneratePacketSize(), tmp, DATA];

for i = 1:n
    tmp = unifrnd(0, 0.02);              % packet arrivals is unif distrib between 0 ms
and 20 ms % FIRST PACKETS IS 0 TO 20 ms !!  % SEE FOOTNOTE3
    EventList = [EventList; ARRIVAL, tmp, randi([110, 130]), tmp, VOIP];
end


%Simulation loop:
while TRANSPACKETS_DATA + TRANSPACKETS_VOIP<P                        % Stopping
criterium

    EventList= sortrows(EventList,2);       % Order EventList by time
    Event= EventList(1,1);                  % Get first event
    Clock= EventList(1,2);                  %    and all
    PacketSize= EventList(1,3);             %    associated
    ArrInstant= EventList(1,4);             %    parameters.
    PacketType = EventList(1, 5);
    EventList(1,:)= [];                     % Eliminate first event
```

```
    switch Event
        case ARRIVAL         % If first event is an ARRIVAL
            if PacketType == DATA
                TOTALPACKETS_DATA= TOTALPACKETS_DATA + 1;
                tmp= Clock + exprnd(1/lambda);
                EventList = [EventList; ARRIVAL, tmp, GeneratePacketSize(), tmp, DATA];
            else %is voip
                TOTALPACKETS_VOIP= TOTALPACKETS_VOIP + 1;
                tmp= Clock +unifrnd(0.016, 0.024);
                EventList = [EventList; ARRIVAL, tmp, randi([110, 130]), tmp, VOIP];
            end
            %%%%
            if STATE==0
                STATE= 1;
                EventList = [EventList; DEPARTURE, Clock + 8*PacketSize/(C*10^6),
PacketSize, Clock, PacketType];
            else
                if QUEUEOCCUPATION + PacketSize <= f
                    QUEUE= [QUEUE;PacketSize , Clock, PacketType];
                    QUEUEOCCUPATION= QUEUEOCCUPATION + PacketSize;
                else
                    if PacketType == DATA
                        LOSTPACKETS_DATA= LOSTPACKETS_DATA + 1;
                    else
                        LOSTPACKETS_VOIP= LOSTPACKETS_VOIP + 1;
                    end
                end
            end

        case DEPARTURE          % If first event is a DEPARTURE
            if (rand() < (1-b)^(PacketSize*8))
                if (PacketType == DATA) % Data
                    TRANSBYTES_DATA= TRANSBYTES_DATA + PacketSize;
                    DELAYS_DATA = DELAYS_DATA + (Clock - ArrInstant);
                    if Clock - ArrInstant > MAXDELAY_DATA
                        MAXDELAY_DATA= Clock - ArrInstant;
                    end
                    TRANSPACKETS_DATA= TRANSPACKETS_DATA + 1;
                else % VoIP
                    TRANSBYTES_VOIP= TRANSBYTES_VOIP + PacketSize;
                    DELAYS_VOIP= DELAYS_VOIP + (Clock - ArrInstant);
                    if Clock - ArrInstant > MAXDELAY_VOIP
                        MAXDELAY_VOIP= Clock - ArrInstant;
                    end
                    TRANSPACKETS_VOIP= TRANSPACKETS_VOIP + 1;
                end
            else
                 if (PacketType == DATA)         % Data Packet
                    LOSTPACKETS_DATA = LOSTPACKETS_DATA + 1;
                else                            % VoIP Packet
                    LOSTPACKETS_VOIP = LOSTPACKETS_VOIP + 1;
                end
            end


            if QUEUEOCCUPATION > 0
                EventList = [EventList; DEPARTURE, Clock + 8*QUEUE(1,1)/(C*10^6),
QUEUE(1,1), QUEUE(1,2), QUEUE(1, 3)];
```

```
                QUEUEOCCUPATION= QUEUEOCCUPATION - QUEUE(1,1);
                QUEUE(1,:)= [];
            else
                STATE= 0;
            end
    end
end

%Performance parameters determination:
PL_data= 100*LOSTPACKETS_DATA/TOTALPACKETS_DATA;  % in percentage
PL_VoIP= 100*LOSTPACKETS_VOIP/TOTALPACKETS_VOIP;

APDd = 1000*DELAYS_DATA/TRANSPACKETS_DATA;                     % in milliseconds
APDv = 1000*DELAYS_VOIP/TRANSPACKETS_VOIP;                     % in milliseconds
MPDd = 1000*MAXDELAY_DATA;                                       % in milliseconds
MPDv = 1000*MAXDELAY_VOIP;                                       % in milliseconds
TT = 1e-6*(TRANSBYTES_DATA+TRANSBYTES_VOIP)*8/Clock;  % in Mbps

%APD= 1000*DELAYS/TRANSPACKETS;     % in milliseconds
%MPD= 1000*MAXDELAY;                % in milliseconds
%TT= 1e-6*TRANSBYTES*8/Clock;    % in Mbps

end

function out= GeneratePacketSize()
    aux= rand();
    aux2= [65:109 111:1517];
    if aux <= 0.19
        out= 64;
    elseif aux <= 0.19 + 0.23
        out= 110;
    elseif aux <= 0.19 + 0.23 + 0.17
        out= 1518;
    else
        out = aux2(randi(length(aux2)));
    end
end
```

c) Exercise 2.f calculations:

```
nVoip = [10 20 30 40];

lambda_data = 1500; %pps
lambda_voip = 50; % time between packet arrivals is uniformly distributed
                 % between 16 milliseconds and 24 milliseconds

avg_size_data = 620; % bytes
avg_size_voip = 120; % bytes

throughput_per_voip_flow = avg_size_voip*8 * lambda_voip; %bits per second

throughput_voip = zeros(1, length(nVoip));
```

```
throughput_voip(1) = throughput_per_voip_flow * 10 / 1e6; % 10 voip flows
throughput_voip(2) = throughput_per_voip_flow * 20 / 1e6; % 20 voip flows
throughput_voip(3) = throughput_per_voip_flow * 30 / 1e6; % 30 voip flows
throughput_voip(4) = throughput_per_voip_flow * 40 / 1e6; % 40 voip flows

throughput_data =  avg_size_data * 8 * lambda_data / 1e6; %bits per second

ber = 1e-5;
P_no_error_data = (1-ber)^(avg_size_data*8);
P_no_error_voip = (1-ber)^(avg_size_voip*8);


tt = zeros(1, length(nVoip));

tt(1) = throughput_voip(1) * P_no_error_voip + throughput_data * P_no_error_data;
tt(2) = throughput_voip(2) * P_no_error_voip + throughput_data * P_no_error_data;
tt(3) = throughput_voip(3) * P_no_error_voip + throughput_data * P_no_error_data;
tt(4) = throughput_voip(4) * P_no_error_voip + throughput_data * P_no_error_data;

ftt = figure;

hold on;

bar(nVoip, tt');
title('Total throughput')
xlabel('Number of VoIP Flows');
xticks(10:10:40);
ylabel('Throughput (Mbps)');
ylim([0 10]);
grid on;


set(ftt, 'Position', [100, 100, figureWidth, figureHeight]);
saveas(ftt, fullfile('figures', 'ftt.png'));


hold off;
```

d) Sim4A:

```
function [PLdata , APDdata , MPDdata , TTdata, PLVoIP, APDVoIP, MPDVoIP] =
Sim4A(lambda,C,f,P,n,p)
% INPUT PARAMETERS:
%  lambda - packet rate (packets/sec)
%  C      - link bandwidth (Mbps)
%  f      - queue size (Bytes)
%  P      - number of packets (stopping criterium)
%  n      - number of VoIP flows
%  p      - % queue size acceptance limit for data packets, between [0,1]
% OUTPUT PARAMETERS:
%  PL   - packet loss (%)
%  APD  - average packet delay (milliseconds)
%  MPD  - maximum packet delay (milliseconds)
```

```
%  TT   - transmitted throughput (Mbps)

%Events:
ARRIVAL= 0;       % Arrival of a packet
DEPARTURE= 1;     % Departure of a packet
ARRIVALVoIP= 2;       % Arrival of a VoIP
DEPARTUREVoIP= 3;     % Departure of a VoIP

%State variables:
STATE = 0;          % 0 - connection is free; 1 - connection is occupied
QUEUEOCCUPATION= 0; % Occupation of the queue (in Bytes)
QUEUE= [];          % Size and arriving time instant of each packet in the queue

%Statistical Counters:
TOTALPACKETS= 0;     % No. of data packets arrived to the system
LOSTPACKETS= 0;      % No. of data packets dropped due to buffer overflow
TRANSPACKETS= 0;     % No. of transmitted data packets
TRANSBYTES= 0;       % Sum of the Bytes of transmitted data and VoIP packets
DELAYS= 0;           % Sum of the delays of transmitted data packets
MAXDELAY= 0;         % Maximum delay among all transmitted data packets
%VoIP Statistical Counters
TOTALVoIPPACKETS= 0;     % No. of VoIP packets arrived to the system
LOSTVoIPPACKETS= 0;      % No. of VoIP packets dropped due to buffer overflow
TRANSVoIPPACKETS= 0;     % No. of transmitted VoIP packets
DELAYSVoIP= 0;           % Sum of the delays of transmitted VoIP packets
MAXDELAYVoIP= 0;         % Maximum delay among all transmitted VoIP packets

% Initializing the simulation clock:
Clock= 0;

% Initializing the List of Events with the first ARRIVAL:
tmp= Clock + exprnd(1/lambda);
EventList = [ARRIVAL, tmp, GeneratePacketSize(), tmp];

for i=1:n   % VoIP n flows first arrivals
    % First void interval
    tmp= Clock + unifrnd(0,20e-3);
    EventList = [EventList; ARRIVALVoIP, tmp, GenerateVoIPPacketSize(), tmp];
end

%Similation loop:
while TRANSPACKETS+TRANSVoIPPACKETS<P               % Stopping criterium
    EventList= sortrows(EventList,2);  % Order EventList by time
    Event= EventList(1,1);              % Get first event
    Clock= EventList(1,2);              %    and all
    PacketSize= EventList(1,3);        %    associated
    ArrInstant= EventList(1,4);    %    parameters.
    EventList(1,:)= [];                 % Eliminate first event
    switch Event
        case ARRIVAL         % If first event is an ARRIVAL of data packet
            TOTALPACKETS= TOTALPACKETS+1;
            tmp= Clock + exprnd(1/lambda);
            EventList = [EventList; ARRIVAL, tmp, GeneratePacketSize(), tmp];
            if STATE==0
                STATE= 1;
                EventList = [EventList; DEPARTURE, Clock + 8*PacketSize/(C*10^6),
PacketSize, Clock];
            else
```

```
                if QUEUEOCCUPATION + PacketSize <= f * (p/100)
                    QUEUE= [QUEUE;PacketSize , Clock, ARRIVAL];
                    QUEUEOCCUPATION= QUEUEOCCUPATION + PacketSize;
                else       % Data packet is lost if the queue is full
                    LOSTPACKETS= LOSTPACKETS + 1;
                end
            end
        case ARRIVALVoIP        % If first event is an ARRIVAL of VoIP packet
            TOTALVoIPPACKETS= TOTALVoIPPACKETS+1;
            tmp= Clock + unifrnd(16e-3,24e-3);    % Arrivals ~ U(16,24) ms
            EventList = [EventList; ARRIVALVoIP, tmp, GenerateVoIPPacketSize(), tmp];
            if STATE==0
                STATE= 1;
                EventList = [EventList; DEPARTUREVoIP, Clock + 8*PacketSize/(C*10^6), 
PacketSize, Clock];
            else
                if QUEUEOCCUPATION + PacketSize <= f
                    QUEUE= [QUEUE;PacketSize , Clock, ARRIVALVoIP];
                    QUEUEOCCUPATION= QUEUEOCCUPATION + PacketSize;
                else
                    LOSTVoIPPACKETS= LOSTVoIPPACKETS + 1;
                end
            end
        case DEPARTURE          % If first event is a DEPARTURE of data packet
            TRANSBYTES= TRANSBYTES + PacketSize;
            DELAYS= DELAYS + (Clock - ArrInstant);
            if Clock - ArrInstant > MAXDELAY
                MAXDELAY= Clock - ArrInstant;
            end
            TRANSPACKETS= TRANSPACKETS + 1;
            if QUEUEOCCUPATION > 0
                QUEUE= sortrows(QUEUE,3,"descend");  % Order QUEUE by packet type (VoIP 
has priority)
                if QUEUE(1,3) == ARRIVAL    % Check if the packet is data on the queue
                    EventList = [EventList; DEPARTURE, Clock + 8*QUEUE(1,1)/(C*10^6), 
QUEUE(1,1), QUEUE(1,2)];
                elseif QUEUE(1,3) == ARRIVALVoIP    % Check if the packet is VoIP on the 
queue
                    EventList = [EventList; DEPARTUREVoIP, Clock + 
8*QUEUE(1,1)/(C*10^6), QUEUE(1,1), QUEUE(1,2)];
                end
                QUEUEOCCUPATION= QUEUEOCCUPATION - QUEUE(1,1);
                QUEUE(1,:)= [];
            else
                STATE= 0;
            end
        case DEPARTUREVoIP      % If first event is a DEPARTURE of VoIP packet
            TRANSBYTES= TRANSBYTES + PacketSize;
            DELAYSVoIP= DELAYSVoIP + (Clock - ArrInstant);
            if Clock - ArrInstant > MAXDELAYVoIP
                MAXDELAYVoIP= Clock - ArrInstant;
            end
            TRANSVoIPPACKETS= TRANSVoIPPACKETS + 1;
            if QUEUEOCCUPATION > 0
                QUEUE= sortrows(QUEUE,3,"descend");  % Order QUEUE by packet type (VoIP 
has priority)
                if QUEUE(1,3) == ARRIVAL    % Check if the packet is data on the queue
                    EventList = [EventList; DEPARTURE, Clock + 8*QUEUE(1,1)/(C*10^6), 
```

```
QUEUE(1,1), QUEUE(1,2)];
                elseif QUEUE(1,3) == ARRIVALVoIP    % Check if the packet is VoIP on the
queue
                    EventList = [EventList; DEPARTUREVoIP, Clock +
8*QUEUE(1,1)/(C*10^6), QUEUE(1,1), QUEUE(1,2)];
                end
                QUEUEOCCUPATION= QUEUEOCCUPATION - QUEUE(1,1);
                QUEUE(1,:)= [];
            else
                STATE= 0;
            end
    end
end

%Performance parameters determination:
PLdata= 100*LOSTPACKETS/TOTALPACKETS;  % in percentage
APDdata= 1000*DELAYS/TRANSPACKETS;     % in milliseconds
MPDdata= 1000*MAXDELAY;                % in milliseconds
%VoIP Performance parameters determination:
PLVoIP= 100*LOSTVoIPPACKETS/TOTALVoIPPACKETS;  % in percentage
APDVoIP= 1000*DELAYSVoIP/TRANSVoIPPACKETS;     % in milliseconds
MPDVoIP= 1000*MAXDELAYVoIP;                % in milliseconds
% TT considers data an VoIP packets
TTdata= 1e-6*TRANSBYTES*8/Clock;    % in Mbps

end

function out= GeneratePacketSize()
    aux= rand();
    aux2= [65:109 111:1517];
    if aux <= 0.19
        out= 64;
    elseif aux <= 0.19 + 0.23
        out= 110;
    elseif aux <= 0.19 + 0.23 + 0.17
        out= 1518;
    else
        out = aux2(randi(length(aux2)));
    end
end

function out= GenerateVoIPPacketSize()
    out = randi([110, 130]);
end
```