

MODELAÇÃO E DESEMPENHO DE REDES E SERVIÇOS

MINI-PROJETO 1

Eduardo Alves: nºmec 104179

Mariana Silva: nºmec 98392



eduardoalves@ua.pt
marianabarbara@ua.pt

[Link Repositório GitHub](#)

Universidade de Aveiro
Departamento de Electrónica, Telecomunicações e Informática
2025

Contents

1	Notas Gerais	4
2	Introdução	4
3	Task 1	6
3.1	Exercício 1a	6
3.1.1	Contexto	6
3.1.2	Implementação	6
3.1.3	Resultados	7
3.1.4	Conclusões	9
3.2	Exercício 1b	10
3.2.1	Contexto	10
3.2.2	Implementação	10
3.2.3	Resultados	10
3.2.4	Análise dos resultados	12
3.2.5	Comparação com o Exercício 1a	12
3.2.6	Conclusões	12
3.3	Exercício 1c	13
3.3.1	Contexto	13
3.3.2	Implementação	13
3.4	Exercício 1d	15
3.4.1	Contexto	15
3.4.2	Implementação	15
3.4.3	Resultados	16
3.5	Exercicio 1e	18
3.5.1	Contexto	18
3.5.2	Implementação	18
3.5.3	Resultados	18
3.6	Exercicio 1f	19
3.6.1	Contexto	19
3.6.2	Implementação	19
3.6.3	Resultados	19
3.7	Exercicio 1g	20
3.7.1	Contexto	20
3.7.2	Implementação	20
3.7.3	Resultados	20
3.8	Exercicio 1h	21
3.8.1	Contexto	21
3.8.2	Implementação	21
3.8.3	Resultados	21
3.9	Exercicio 1i	22
3.9.1	Contexto	22
3.9.2	Implementação	22
3.9.3	Resultados	22
3.10	Exercicio 1j	23
3.10.1	Contexto	23
3.10.2	Implementação	23

3.10.3	Resultados	23
4	Parte 2	24
4.1	Exercicio 2a	24
4.1.1	Contexto	24
4.1.2	Implementação	24
4.1.3	Resultados	24
4.2	Exercicio 2b	25
4.2.1	Contexto	25
4.2.2	Implementação	25
4.2.3	Resultados	25
4.3	Exercicio 2c	26
4.3.1	Contexto	26
4.3.2	Implementação	26
4.3.3	Resultados	26
4.4	Exercicio 2d	27
4.4.1	Contexto	27
4.4.2	Implementação	27
4.4.3	Resultados	27
4.5	Exercicio 2e	28
4.5.1	Contexto	28
4.5.2	Implementação	28
4.5.3	Resultados	28

List of Figures

1	1a) Perda média de pacotes em função da taxa de chegada (λ)	7
2	1a) Atraso médio de pacotes em função da taxa de chegada (λ)	8
3	1a) Resultados em formato de linha de comandos.	8
4	1b) Perda de pacotes em função da taxa de chegada (λ) para $f = 10.000$ Bytes	10
5	1b) Atraso médio de pacotes em função da taxa de chegada (λ) para $f = 10.000$ Bytes.	11
6	1b) Resultados na linha de comandos.	11
7	1d) Perda média de pacotes para cada tipo de pacote (Exercício 1d).	16
8	1d) Atraso médio dos pacotes para cada tipo de pacote (Exercício 1d).	16
9	1d) Resultados apresentados na linha de comandos.	17

1 Notas Gerais

- Todos os membros do grupo contribuíram de forma equilibrada e ativa para o desenvolvimento deste projeto, participando nas fases de planeamento, implementação e testes. O trabalho foi realizado de forma colaborativa, garantindo uma divisão justa das tarefas e a integração das ideias de todos os elementos.
- O repositório completo do projeto encontra-se disponível no seguinte link:
https://github.com/marianabarbara/MDRS_Projects
- Foram realizados testes funcionais e de integração para garantir o correto funcionamento do sistema e a coerência entre os diferentes módulos.
- É importante salientar que, embora o repositório do projeto tenha permanecido privado durante o seu desenvolvimento, foi disponibilizado publicamente perto do período de entrega para efeitos de avaliação.

2 Introdução

O objetivo deste projeto é analisar e avaliar o desempenho de ligações ponto-a-ponto que suportam serviços de pacotes, recorrendo à modelação e simulação de diferentes cenários em *MATLAB*. Pretende-se compreender o impacto de diversos parâmetros de rede, tais como capacidade do canal, tamanho médio dos pacotes e taxa de chegada no desempenho global do sistema, nomeadamente no que diz respeito à perda média de pacotes, ao atraso médio e à eficiência do enlace.

O projeto será dividido em duas tarefas principais, cada uma com objetivos e componentes específicos. Em todas as simulações, serão realizadas 50 execuções independentes com um critério de paragem de $P = 100\,000$, sendo posteriormente calculados os valores estimados e os intervalos de confiança a 90%, apresentados sob a forma de gráficos de barras com barras de erro.

- **Task 1** Nesta fase será avaliado o desempenho de um sistema de transmissão ponto-a-ponto através do desenvolvimento e adaptação de simuladores orientados a eventos (**Simulator1**, **Simulator1A** e **Simulator1B**). O principal objetivo é analisar o comportamento do sistema sob diferentes condições de tráfego e políticas de enfileiramento, bem como comparar os resultados obtidos por simulação com valores teóricos derivados de modelos analíticos de filas (*queueing models*).
 - **1a:** Estimar, por simulação, a perda média de pacotes e o atraso médio para diferentes taxas de chegada λ , considerando pacotes com tamanho médio de $f = 1\,000\,000$ Bytes.
 - **1b:** Repetir as experiências anteriores considerando $f = 10\,000$ Bytes e analisar as diferenças observadas em relação à alínea anterior.
 - **1c:** Desenvolver uma nova versão do simulador, denominada **Simulator1A**, capaz de calcular separadamente os parâmetros de desempenho para pacotes com tamanhos específicos (64, 110 e 1518 Bytes).
 - **1d:** Utilizar o **Simulator1A** para estimar a perda média e o atraso médio, tanto para o conjunto total de pacotes como para cada um dos três tamanhos especiais.

- 1e: Determinar, através de um modelo analítico M/G/1, os valores teóricos correspondentes e compará-los com os resultados obtidos por simulação.
 - 1f: Repetir as experiências de 1d para $f = 10\,000$ Bytes e justificar as diferenças obtidas.
 - 1g: Desenvolver uma nova versão do simulador, **Simulator1B**, que introduz um sistema de prioridades baseado no tamanho dos pacotes.
 - 1h: Utilizar o **Simulator1B** para estimar o desempenho do sistema com as prioridades definidas, considerando $f = 1\,000\,000$ Bytes.
 - 1i: Repetir as experiências de 1h para $f = 10\,000$ Bytes e analisar as diferenças observadas.
 - 1j: Comparar e justificar as diferenças entre os resultados obtidos com os simuladores **Simulator1A** e **Simulator1B**.
- **Task 2** Nesta etapa será avaliado o desempenho de um sistema que suporta múltiplos serviços de tráfego (dados e VoIP), utilizando e modificando o simulador **Simulator3** para incluir erros de transmissão no enlace. O objetivo é compreender o impacto da taxa de erro de bits e do número de fluxos VoIP no desempenho do sistema e na eficiência do canal.
- 2a: Desenvolver uma nova versão do simulador, **Simulator3A**, que considera a introdução de uma taxa de erro de bits (b) como parâmetro adicional.
 - 2b: Estimar, por simulação, a perda média de pacotes para os serviços de dados e VoIP, variando o número de fluxos VoIP (n).
 - 2c: Estimar o atraso médio de pacotes para cada serviço, nas mesmas condições anteriores.
 - 2d: Calcular o débito total (*throughput*) do enlace em cada cenário e analisar o comportamento do sistema.
 - 2e: Desenvolver um modelo analítico em *MATLAB* para determinar o valor teórico do *throughput* total e compará-lo com os resultados obtidos por simulação.

3 Task 1

3.1 Exercício 1a

3.1.1 Contexto

Neste exercício foi analisado o desempenho de um sistema de transmissão considerando um canal com capacidade $C = 10$ Mbps, um tamanho de ficheiro $f = 10^6$ Bytes e diferentes taxas de chegada de pacotes (λ) correspondentes a 1100, 1300, 1500, 1700 e 1900 pacotes por segundo (pps). O objetivo consiste em estimar, através de simulação, a perda média de pacotes (*Packet Loss*) e o atraso médio de pacotes (*Average Packet Delay*) para cada um dos valores de λ definidos.

Estes parâmetros são fundamentais para avaliar a qualidade de serviço (QoS) em redes de comunicação, uma vez que permitem perceber o impacto da carga oferecida no desempenho do sistema. Para cargas baixas espera-se que o sistema funcione de forma eficiente, com atrasos reduzidos e perdas nulas. À medida que a taxa de chegada se aproxima da capacidade do canal, o atraso tende a aumentar significativamente, podendo eventualmente ocorrer perdas de pacotes.

3.1.2 Implementação

A simulação foi desenvolvida considerando $P = 10^5$ pacotes transmitidos e $N = 50$ execuções independentes, de forma a obter intervalos de confiança de 90% ($\alpha = 0.1$). Os principais parâmetros definidos no código foram:

- $P = 10^5$ – número de pacotes a simular;
- $N = 50$ – número de execuções independentes;
- $\alpha = 0.1$ – nível de significância para o cálculo do intervalo de confiança;
- $C = 10$ Mbps – capacidade do canal;
- $f = 10^6$ Bytes – tamanho de ficheiro transmitido;
- $\lambda = [1100, 1300, 1500, 1700, 1900]$ pps – taxas de chegada dos pacotes.

Durante a execução, para cada valor de λ , foram calculadas as métricas de *Packet Loss* e *Average Packet Delay*, fazendo uso do Simulador1 disponibilizado pelo professor no âmbito da unidade curricular, este, tem como tarefa simular a chegada e saída de pacotes de um canal com fila de tamanho limitado.

O simulador mantém uma lista de eventos composta por chegadas (ARRIVAL) e partidas (DEPARTURE), atualizando o estado do canal e da fila de forma dinâmica. Quando um pacote chega e o canal está ocupado, ele é colocado na fila se houver espaço disponível; caso contrário, é descartado, contribuindo para o *Packet Loss*. Cada pacote transmitido tem o seu atraso calculado como a diferença entre o tempo de chegada e o tempo de saída, permitindo determinar tanto o **atraso médio** (*Average Packet Delay*) como o **atraso máximo** (*Maximum Packet Delay*).

Para cada valor de λ , as métricas *Packet Loss* (PL) e *Average Packet Delay* (APD) foram calculadas para as N execuções, e os resultados médios e respectivos **intervalos**

de confiança de 90% foram obtidos usando:

$$\bar{x} = \frac{1}{N} \sum_{i=1}^N x_i, \quad IC_{90\%} = \bar{x} \pm z_{1-\alpha/2} \cdot \frac{s}{\sqrt{N}}$$

onde s é a variância amostral de cada métrica e $z_{1-\alpha/2}$ o quantil da distribuição normal correspondente a 90% de confiança.

Os resultados foram apresentados em **diagramas de barras** com barras de erro indicando os intervalos de confiança, conforme ilustrado pelo seguinte trecho de código:

```
fprintf('%-30s = %.2e +- %.2e\n', 'PacketLoss (%)', mediaPL, termPL);
fprintf('%-30s = %.2e +- %.2e\n', 'Average Packet Delay (ms)', mediaAPD, termAPD);
```

Este procedimento permite analisar a relação entre a taxa de chegada de pacotes e o desempenho do canal em termos de perda e atraso.

3.1.3 Resultados

As Figuras 1 e 2 apresentam, respectivamente, a evolução da perda de pacotes e do atraso médio em função da taxa de chegada de pacotes. E a figura 3 demonstra os resultados obtidos no formato da linha de comandos.

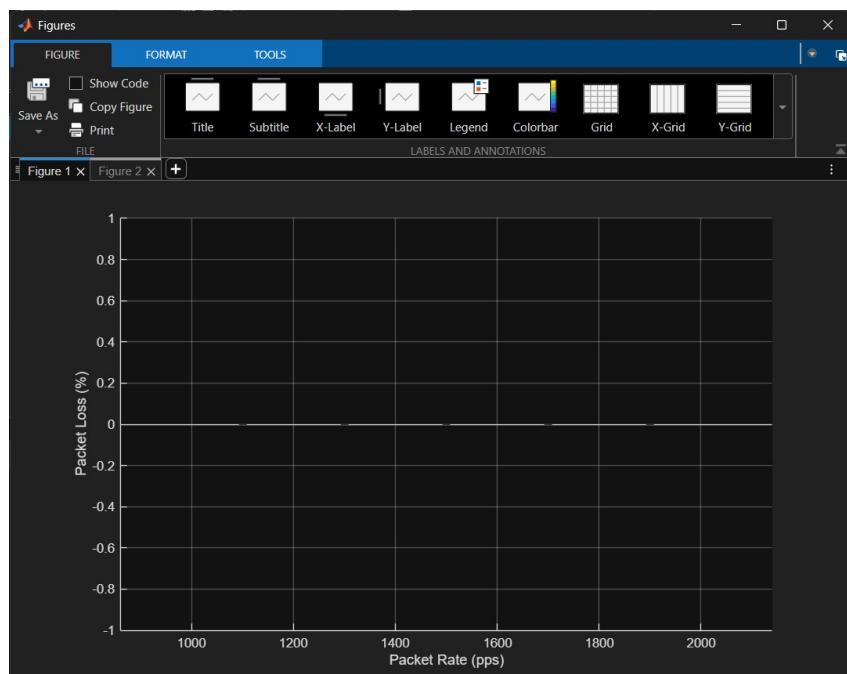


Figure 1: 1a) Perda média de pacotes em função da taxa de chegada (λ).

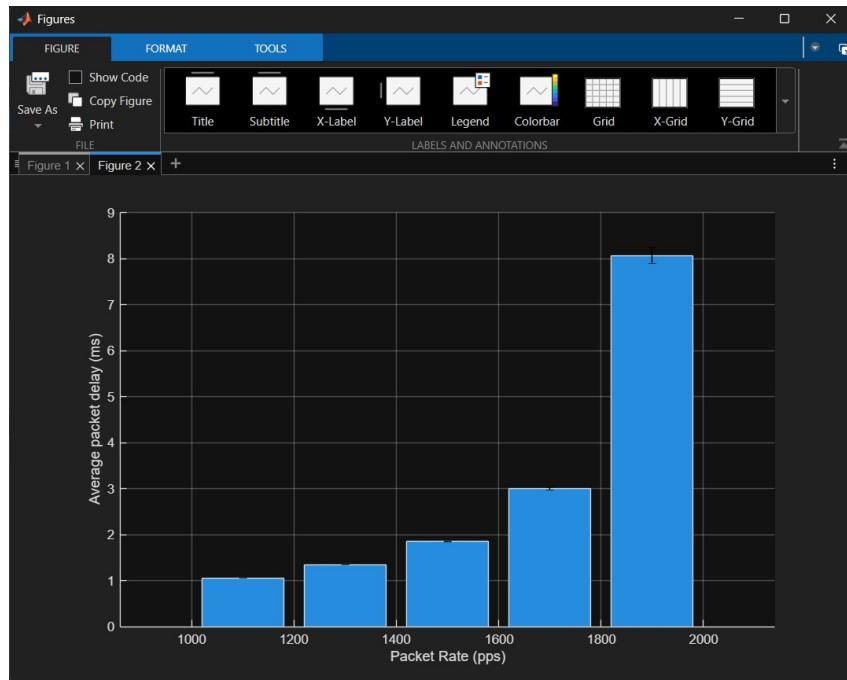


Figure 2: 1a) Atraso médio de pacotes em função da taxa de chegada (λ).

```

Command Window
>> task1
Valor de lambda: 1100
Packet Loss (%) = 0.00e+00 +- 0.00e+00
Average Packet Delay (ms) = 1.05e+00 +- 1.90e-03
Valor de lambda: 1300
Packet Loss (%) = 0.00e+00 +- 0.00e+00
Average Packet Delay (ms) = 1.34e+00 +- 4.33e-03
Valor de lambda: 1500
Packet Loss (%) = 0.00e+00 +- 0.00e+00
Average Packet Delay (ms) = 1.86e+00 +- 9.15e-03
Valor de lambda: 1700
Packet Loss (%) = 0.00e+00 +- 0.00e+00
Average Packet Delay (ms) = 3.00e+00 +- 2.73e-02
Valor de lambda: 1900
Packet Loss (%) = 0.00e+00 +- 0.00e+00
Average Packet Delay (ms) = 8.06e+00 +- 1.71e-01

Simulation ended!
>>

```

Figure 3: 1a) Resultados em formato de linha de comandos.

A análise dos resultados mostra que:

- Para todas as taxas de chegada de pacotes avaliadas, o sistema não apresenta perda de pacotes (0% de Packet Loss).
- É notável um acréscimo lento de atraso médio de envio de pacotes entre a taxa de chegada entre os valores de $\lambda = 1100$ (1.05ms) para $\lambda = 1300$ (1.34ms), o que demonstra um aumento curto de 0,29ms. De $\lambda = 1300$ (1,34ms) para $\lambda = 1500$ (1.86ms) o que demonstra um aumento curto de 0,52ms. E de $\lambda = 1500$ (1,86ms) para $\lambda = 1700$ (3ms), o que já demonstra um aumento curto embora mais relevante de 1.14ms.
- A partir de $\lambda = 1900$ pps, observa-se um aumento significativo no atraso médio dos pacotes, atingindo os 8 ms. O que demonstra um aumento relativamente ao $\lambda = 1700$ de 5.06ms.

3.1.4 Conclusões

Com base nos resultados obtidos, conclui-se que:

- O tamanho da fila $f = 1 \times 10^6$ bytes é relativamente grande (≈ 1 MB), logo, mesmo na taxa mais alta (1900 pacotes/segundo), a capacidade do enlace $C = 10$ Mbps é mais do que suficiente para atender aos pacotes sem ocorrer overflow. Portanto, nenhum pacote é descartado, e o simulador reporta $PacketLoss = 0$.
- Os resultados mostram o crescimento não linear do atraso à medida que a utilização cresce, o que está de acordo com a teoria das filas.
- O aumento abrupto do atraso médio a partir de 1900 pps demonstra a aproximação do limite de capacidade do canal, evidenciando a natureza não-linear da degradação de desempenho à medida que a carga se aproxima da capacidade máxima.
- Apesar da métrica do *Packet Loss* se manter nula em todos os casos simulados, o que indica que o sistema ainda não atingiu o ponto de congestão total, atraso já começa a demonstrar sinais de sobrecarga.

Desta forma, a simulação permitiu validar o impacto da carga oferecida no atraso médio e na perda de pacotes, comprovando que a saturação do canal tem um efeito exponencial no aumento do atraso, mesmo antes de ocorrer perda efetiva de pacotes.

3.2 Exercício 1b

3.2.1 Contexto

Neste exercício foram repetidas as simulações do Exercício 1a, alterando apenas o tamanho máximo da fila para $f = 10.000$ Bytes, mantendo a capacidade do canal $C = 10$ Mbps e as mesmas taxas de chegada $\lambda = [1100, 1300, 1500, 1700, 1900]$ pps. O objetivo é avaliar o impacto da diminuição do tamanho da fila no desempenho do sistema, analisando a perda média de pacotes (*Packet Loss*) e o atraso médio (*Average Packet Delay*).

3.2.2 Implementação

O código utilizado é idêntico ao do Exercício 1a, variando apenas o valor de f . Foram novamente realizadas $N = 50$ execuções para cada valor de λ , simulando $P = 10^5$ pacotes por execução, e calculando intervalos de confiança de 90% para as métricas de desempenho.

As métricas obtidas foram:

- **Perda média de pacotes (%)**: proporção de pacotes descartados devido à fila cheia.
- **Atraso médio (ms)**: tempo médio entre a chegada e o envio de um pacote.

Ambas relativas à taxa de envio de pacotes (λ).

3.2.3 Resultados

Os resultados experimentais obtidos foram os seguintes:

As Figuras 4 e 5 ilustram os resultados obtidos em formato de gráfico e a figura 6.

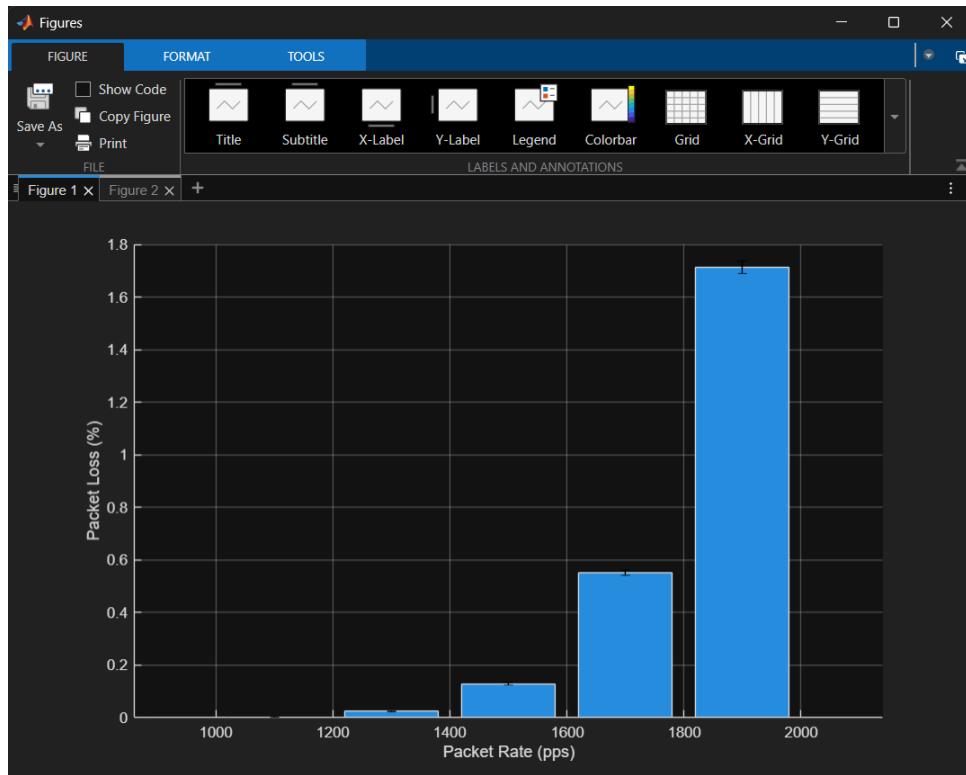


Figure 4: 1b) Perda de pacotes em função da taxa de chegada (λ) para $f = 10.000$ Bytes.

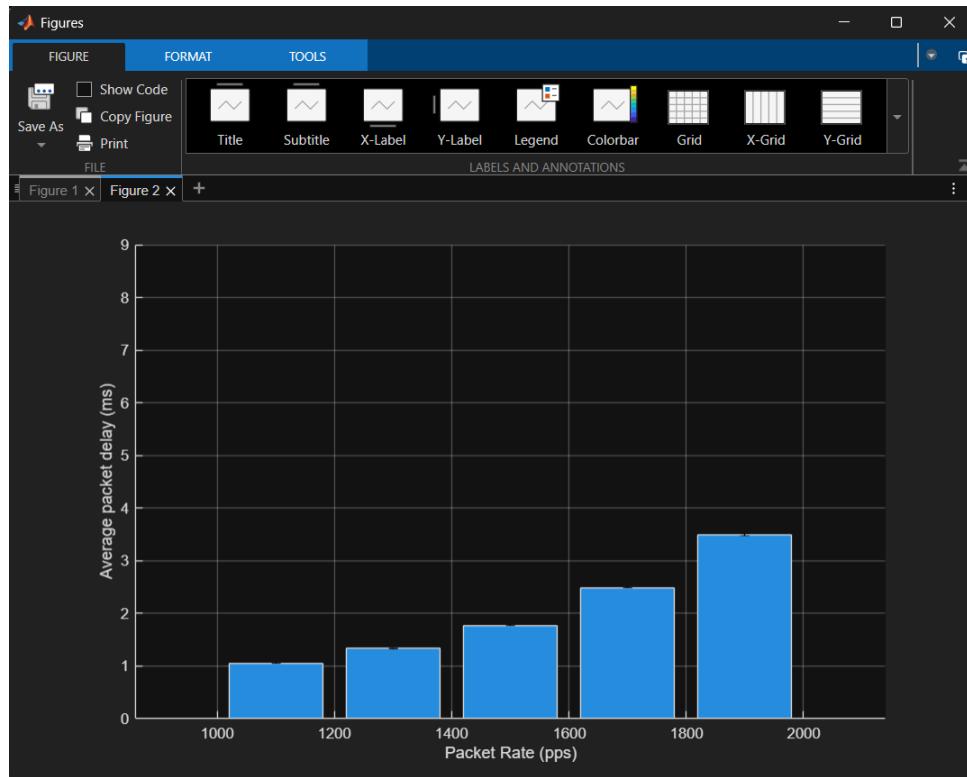


Figure 5: 1b) Atraso médio de pacotes em função da taxa de chegada (λ) para $f = 10.000$ Bytes.

```
Command Window
>> task1
TASK 1b)
Valor de lambda: 1100
Packet Loss (%) = 3.02e-03 +- 5.12e-04
Average Packet Delay (ms) = 1.06e+00 +- 1.72e-03
Valor de lambda: 1300
Packet Loss (%) = 2.40e-02 +- 2.02e-03
Average Packet Delay (ms) = 1.34e+00 +- 4.13e-03
Valor de lambda: 1500
Packet Loss (%) = 1.29e-01 +- 4.37e-03
Average Packet Delay (ms) = 1.77e+00 +- 6.59e-03
Valor de lambda: 1700
Packet Loss (%) = 5.51e-01 +- 1.14e-02
Average Packet Delay (ms) = 2.49e+00 +- 8.82e-03
Valor de lambda: 1900
Packet Loss (%) = 1.71e+00 +- 2.46e-02
Average Packet Delay (ms) = 3.49e+00 +- 1.23e-02

Simulation ended!
>>
```

Figure 6: 1b) Resultados na linha de comandos.

3.2.4 Análise dos resultados

Observa-se que, com a fila muito menor ($f = 10.000$ Bytes), o sistema sofre uma degradação significativa de desempenho à medida que λ aumenta:

- Para $\lambda = 1100$ pps, a perda de pacotes é praticamente nula e o atraso médio é relativamente baixo (cerca de 1 ms), indicando um sistema estável.
- A partir de $\lambda = 1300$ pps, a **perda de pacotes começa a demonstrar um pequeno aumento**. Ao atingir $\lambda = 1500$ a perda de pacotes sobe consideravelmente e para $\lambda = 1900$ sobe já muito acentuadamente.
- Para $\lambda = 1900$ pps, a perda atinge cerca de 1.71%, e o atraso médio sobe para aproximadamente 3.49 ms, mostrando um estado de congestionamento acentuado.

Este comportamento confirma que um **tamanho de fila reduzido limita fortemente a capacidade do sistema de absorver picos de tráfego**, levando à rejeição de pacotes e ao aumento dos tempos de espera.

3.2.5 Comparação com o Exercício 1a

Em comparação com o Exercício 1a (onde $f = 1.000.000$ Bytes), verifica-se uma diferença clara:

- No 1a, o sistema manteve perdas praticamente nulas e atrasos baixos até $\lambda = 1900$ pps.
- No 1b, as perdas e atrasos aumentam de forma acentuada para os mesmos valores de λ .

Esta diferença é consequência direta da redução da capacidade de armazenamento. Com um valor de f menor, a fila satura mais rapidamente, o que leva a um crescimento rápido das perdas e do atraso médio. Além disso, o aumento do atraso antes da perda é típico de sistemas de filas com capacidades mais reduzidas, pois os pacotes começam a acumular até ao ponto de rejeição.

3.2.6 Conclusões

Conclui-se que a redução do tamanho da fila tem um impacto negativo na eficiência do sistema:

- O atraso médio aumenta de forma notável mesmo para taxas moderadas de chegada.
- A perda de pacotes deixou de ser nula e torna-se notável desde inicio.

Estes resultados demonstram a importância do dimensionamento adequado em sistemas de comunicação, de forma a equilibrar a utilização da rede e o tempo de resposta.

3.3 Exercício 1c

3.3.1 Contexto

Nesta secção foi desenvolvido um novo simulador, denominado **Simulator1A**, com base no código do *Simulator1*. O objetivo desta versão é estimar seis parâmetros de desempenho adicionais, correspondentes à **perda média de pacotes (PL)** e ao **atraso médio dos pacotes transmitidos (APD)** para três tamanhos específicos de pacotes: **64, 110 e 1518 Bytes**. Desta forma, além dos parâmetros gerais de desempenho, o simulador passa também a calcular individualmente o comportamento da rede para pacotes de tamanhos distintos, permitindo uma análise mais detalhada do impacto do tamanho do pacote no desempenho do sistema.

3.3.2 Implementação

O código do *Simulator1A* baseia-se na estrutura do *Simulator1*, mas com a introdução de variáveis e cálculos adicionais para monitorizar separadamente as estatísticas associadas a pacotes de 64, 110 e 1518 Bytes. O código deste simulador na sua totalidade está presente [aqui!](#) No entanto, as principais alterações introduzidas foram as seguintes:

1. Criação de novos contadores de estatísticas por tamanho de pacote.

Foram adicionadas variáveis para contabilizar o número total de pacotes gerados, transmitidos, perdidos e o somatório dos atrasos, separadamente para cada tamanho de pacote:

```
DELAYS = 0;  
DELAYS_64 = 0;  
DELAYS_110 = 0;  
DELAYS_1518 = 0;  
  
TOTALPACKETS = 0;  
TOTALPACKETS_64 = 0;  
TOTALPACKETS_110 = 0;  
TOTALPACKETS_1518 = 0;  
  
LOSTPACKETS = 0;  
LOSTPACKETS_64 = 0;  
LOSTPACKETS_110 = 0;  
LOSTPACKETS_1518 = 0;  
  
TRANSPACKETS = 0;  
TRANSPACKETS_64 = 0;  
TRANSPACKETS_110 = 0;  
TRANSPACKETS_1518 = 0;
```

Estas variáveis permitem distinguir os resultados por categoria de tamanho de pacote, sem interferir na lógica geral do simulador.

2. Contabilização dos pacotes totais por tipo no evento *ARRIVAL*.

Logo que um pacote chega, o seu tamanho é verificado e incrementa-se o contador correspondente:

```
TOTALPACKETS = TOTALPACKETS + 1;
if PacketSize == 64
    TOTALPACKETS_64 = TOTALPACKETS_64 + 1;
elseif PacketSize == 110
    TOTALPACKETS_110 = TOTALPACKETS_110 + 1;
elseif PacketSize == 1518
    TOTALPACKETS_1518 = TOTALPACKETS_1518 + 1;
end
```

3. Contabilização de perdas por tipo de pacote.

Quando o buffer está cheio e um pacote é descartado, o simulador identifica o seu tamanho e incrementa o contador de perdas específico:

```
LOSTPACKETS = LOSTPACKETS + 1;
if PacketSize == 64
    LOSTPACKETS_64 = LOSTPACKETS_64 + 1;
elseif PacketSize == 110
    LOSTPACKETS_110 = LOSTPACKETS_110 + 1;
elseif PacketSize == 1518
    LOSTPACKETS_1518 = LOSTPACKETS_1518 + 1;
end
```

4. Acumulação de atrasos e pacotes transmitidos por tipo no evento *DEPARTURE*.

Após a transmissão de cada pacote, o simulador regista o seu atraso e atualiza as contagens de pacotes transmitidos por tipo:

```
if PacketSize == 64
    TRANSPACKETS_64 = TRANSPACKETS_64 + 1;
    DELAYS_64 = DELAYS_64 + (Clock - ArrInstant);
elseif PacketSize == 110
    TRANSPACKETS_110 = TRANSPACKETS_110 + 1;
    DELAYS_110 = DELAYS_110 + (Clock - ArrInstant);
elseif PacketSize == 1518
    TRANSPACKETS_1518 = TRANSPACKETS_1518 + 1;
    DELAYS_1518 = DELAYS_1518 + (Clock - ArrInstant);
end
```

5. Cálculo dos novos parâmetros de desempenho.

No final da simulação, foram adicionadas as expressões para calcular a perda e atraso médio de cada tipo de pacote:

```

PL_64    = 100 * LOSTPACKETS_64 / TOTALPACKETS_64;
APD_64   = 1000 * DELAYS_64 / TRANSPACKETS_64;

PL_110   = 100 * LOSTPACKETS_110 / TOTALPACKETS_110;
APD_110  = 1000 * DELAYS_110 / TRANSPACKETS_110;

PL_1518  = 100 * LOSTPACKETS_1518 / TOTALPACKETS_1518;
APD_1518 = 1000 * DELAYS_1518 / TRANSPACKETS_1518;

```

Estas métricas complementam as globais já calculadas pelo simulador original.

3.4 Exercício 1d

3.4.1 Contexto

Neste exercício, pretende-se utilizar a função **Simulator1A** desenvolvida no exercício anterior para estimar, por simulação, os parâmetros de desempenho **perda média de pacotes** (*Packet Loss*) e **atraso médio dos pacotes** (*Average Packet Delay*). A simulação deve considerar os três tamanhos de pacotes adicionados no exercício anterior (64, 110 e 1518 Bytes) e também o conjunto total de pacotes.

Os parâmetros de simulação são:

$$C = 10 \text{ Mbps}, \quad f = 1,000,000 \text{ Bytes}, \quad \lambda = 1900 \text{ pps}$$

O critério de paragem é de $P = 10^5$ pacotes transmitidos e foram realizadas $N = 50$ execuções independentes para calcular intervalos de confiança a 90%.

3.4.2 Implementação

Como já concretizado previamente, o simulador **Simulator1A** foi desenvolvido de forma a permitir a recolha separada de estatísticas para três tamanhos específicos de pacotes: 64, 110 e 1518 Bytes.

Neste exercício, o código principal tem como objetivo utilizar o **Simulator1A** para comparar o desempenho do sistema entre os diferentes tamanhos de pacotes, mantendo as restantes condições fixas.

Em cada iteração da simulação, o **Simulator1A** devolve oito métricas principais:

- PL e APD : perda e atraso médios considerando todos os pacotes;
- $PL_{64}, PL_{110}, PL_{1518}$: perdas médias específicas por tamanho de pacote;
- $APD_{64}, APD_{110}, APD_{1518}$: atrasos médios específicos por tamanho de pacote;

Após todas as execuções, o código calcula a média e o intervalo de confiança a 90% para cada métrica.

Os resultados são então organizados em dois vetores:

- **PL_means** e **PL_terms** — contêm as médias e os intervalos de confiança da perda de pacotes;
- **APD_means** e **APD_terms** — contêm as médias e intervalos correspondentes ao atraso médio.

Por fim, os resultados são apresentados graficamente, como podermos analisar de seguida.

3.4.3 Resultados

A Figura 7 apresenta as perdas médias de pacotes e a Figura 8 mostra os atrasos médios. A figura 9

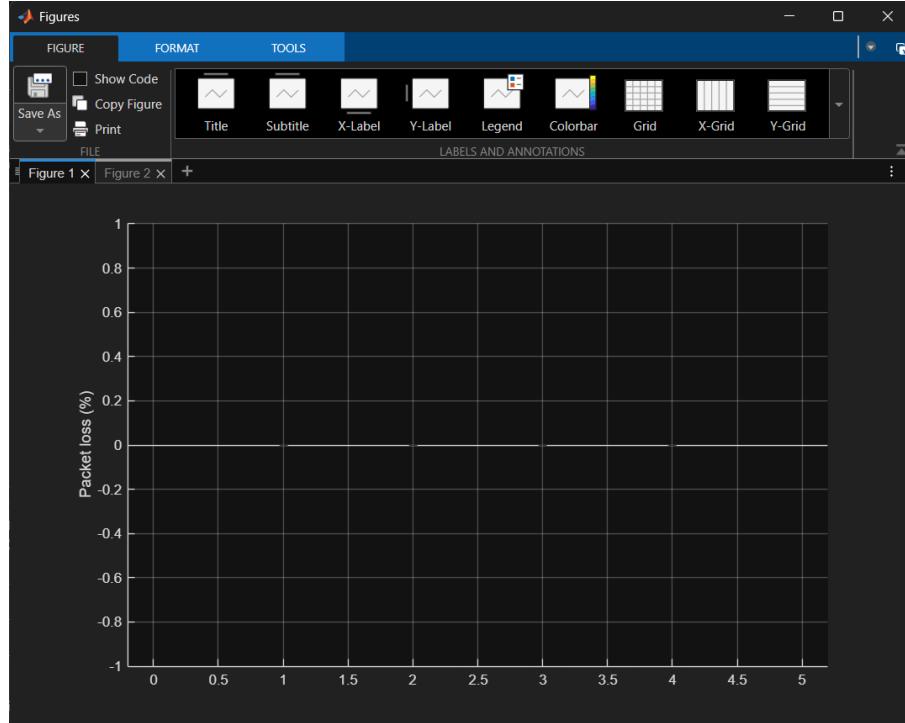


Figure 7: 1d) Perda média de pacotes para cada tipo de pacote (Exercício 1d).

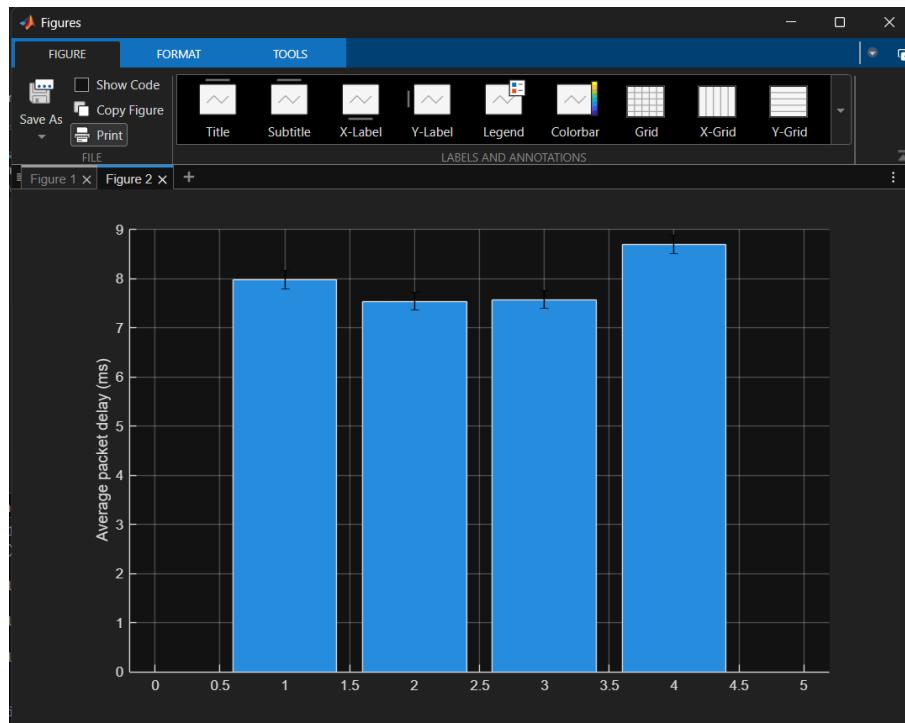


Figure 8: 1d) Atraso médio dos pacotes para cada tipo de pacote (Exercício 1d).

```

Command Window
>> task1
TASK 1d)
Packet Loss (%) = 0.00e+00 +- 0.00e+00
Average Packet Delay (ms) = 7.98e+00 +- 1.86e-01
Packet Loss (%) = 0.00e+00 +- 0.00e+00
Average Packet Delay (ms) = 7.54e+00 +- 1.82e-01
Packet Loss (%) = 0.00e+00 +- 0.00e+00
Average Packet Delay (ms) = 7.58e+00 +- 1.86e-01
Packet Loss (%) = 0.00e+00 +- 0.00e+00
Average Packet Delay (ms) = 8.70e+00 +- 1.91e-01

Simulation ended!
>>

```

Figure 9: 1d) Resultados apresentados na linha de comandos.

Análise e conclusões dos Resultados

Verifica-se que a **perda média de pacotes é nula** para todos os tamanhos, o que se justifica pelo grande tamanho de $f = 1\text{ MB}$ e pela capacidade do enlace ($C = 10\text{ Mbps}$), que permitem processar todas as chegadas sem saturação.

Relativamente ao atraso médio, este mantém-se em torno de 8 ms, sendo ligeiramente superior para pacotes maiores (1518 Bytes), devido ao maior tempo de transmissão. Este comportamento é consistente com a teoria de filas, onde o atraso total é composto pelo tempo de transmissão e o tempo médio de espera na fila, sendo o primeiro dominante neste regime. Assim, conclui-se que para esta taxa de chegada e parâmetros de sistema, o enlace opera **sem perdas** e com **atrasos baixos e estáveis**.

3.5 Exercicio 1e

3.5.1 Contexto

3.5.2 Implementação

3.5.3 Resultados

3.6 Exercicio 1f

3.6.1 Contexto

3.6.2 Implementação

3.6.3 Resultados

3.7 Exercicio 1g

3.7.1 Contexto

3.7.2 Implementação

3.7.3 Resultados

3.8 Exercicio 1h

3.8.1 Contexto

3.8.2 Implementação

3.8.3 Resultados

3.9 Exercicio 1i

3.9.1 Contexto

3.9.2 Implementação

3.9.3 Resultados

3.10 Exercicio 1j

3.10.1 Contexto

3.10.2 Implementação

3.10.3 Resultados

4 Parte 2

4.1 Exercicio 2a

4.1.1 Contexto

4.1.2 Implementação

4.1.3 Resultados

4.2 Exercicio 2b

4.2.1 Contexto

4.2.2 Implementação

4.2.3 Resultados

4.3 Exercicio 2c

4.3.1 Contexto

4.3.2 Implementação

4.3.3 Resultados

4.4 Exercicio 2d

4.4.1 Contexto

4.4.2 Implementação

4.4.3 Resultados

4.5 Exercicio 2e

4.5.1 Contexto

4.5.2 Implementação

4.5.3 Resultados