Redes de Acesso IP: Taxas, Atrasos, Perdas e Duplicação de Pacotes

Catarina Pereira, *PG53733,* Inês Neves, *PG53864,* e Leonardo Martins, *PG53996*

Resumo

Este trabalho refere a conectividade e o desempenho das redes de acesso IP, utilizando ferramentas como `ping`, `traceroute` e `iperf`. São realizados testes de conectividade entre o HOME-PC e o SERVER, analisando os tempos de transmissão e o RTT para diferentes tamanhos de pacotes. Modificações na conexão residencial são feitas para torná-la assimétrica, seguidas por novos testes de conectividade. A análise de desempenho é conduzida utilizando `iperf` para medir a largura de banda máxima em transferências de dados em TCP (Transmission Control Protocol) e UDP (User Datagram Protocol). Os diferentes parâmetros de configuração na conexão residencial são explorados e os seus efeitos no desempenho são avaliados em ambos os protocolos.

Index Terms

Ping, Traceroute, RTT (Round-trip time), iPerf/iPerf3, TCP/UDP, Loss (Perda), Duplicate (Duplicação), Configuração de Parâmetros de Rede, Análise de Desempenho, Emulador Core

I. Introdução

ESTE relatório está inserido no âmbito da Unidade Curricular Redes de Acesso e Núcleo, do 2º semestre do 1º ano do Mestrado em Engenharia de Telecomunicações e Informática, como resposta a um problema apresentado pelo docente. Para realizar este trabalho recorre-se às ferramentas dadas neste trabalho e ao material existente na BlackBoard [1].

II. Estabelecimento das Redes de Acesso e Núcleo

COMO se pode observar na Figura 1, a topologia é composta por uma configuração de Rede de Núcleo (CORE Network), na qual todas as ligações operam em modo full-duplex com uma capacidade de 1 $Gbps$ e apresentam atrasos de 100 us. Adicionalmente, existe um servidor denominado SERVER (Endereço IP 10.0.6.10) que está conectado à rede de núcleo através de uma ligação Ethernet a 100 $Mbps$, com um atraso de 50 us.

A rede residencial de acesso utiliza um Link de Acesso (Access Link - Residencial) com uma taxa de **T** e um atraso de **D**. O computador residencial, HOME-PC (Endereço IP 10.0.5.20), está ligado ao HOME ROUTER (Endereço IP 10.0.5.1) através de uma ligação Ethernet a 100 $Mbps$, apresentando um atraso de 10 us.

Na rede residencial a tecnologia de acesso à rede escolhida foi a fibra ótica mais propriamente, *FTTH (Fiber To The Home)*, utiliza-se taxas de largura de banda de 200 $Mbps$ / 100 $Mbps$ (200 $Mbps$ Downstream; 100 $Mbps$ Upstream) por ser uma das larguras de banda mais usadas no país, quanto ao atraso opta-se por uma latência de 1 ms que é uma latência baixa comum em redes de fibra deste tipo (*FTTH*).

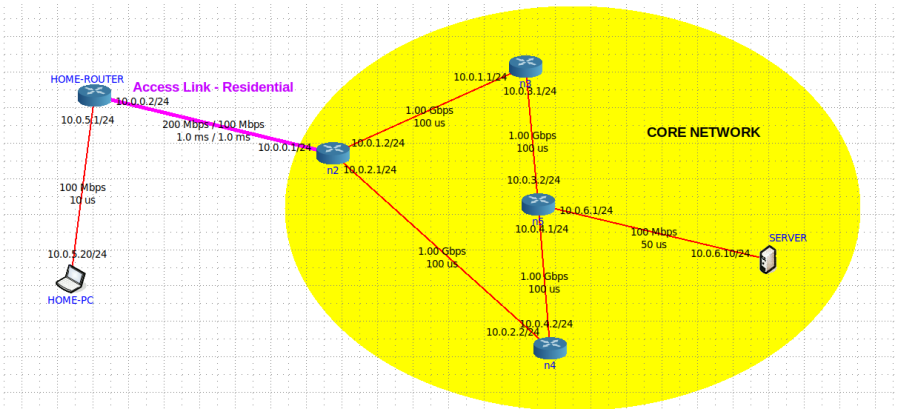

Figura 1: Topologia Redes de Acesso e Redes de Núcleo.

Os alunos do Grupo 3 são do Mestrado de Engenharia de Telecomunicações, lecionado no Campus de Azurém da Universidade do Minho.
Catarina Pereira e-mail: pg537336@alunos.uminho.pt.
Inês Neves e-mail: pg53864@alunos.uminho.pt.
Leonardo Martins e-mail: pg53996@alunos.uminho.pt.

A. Teste de Conectividade e Comparação de Atraso

Para testar a conectividade, recorre-se às ferramentas `ping` e `traceroute`. Inicialmente, conforme representado na Figura 2, utiliza-se o comando `traceroute`, o qual permite mapear a rota que os pacotes percorrem desde o *HOME-PC* até ao *SERVER*. Este comando é também utilizado para identificar a trajetória que os pacotes seguem de um computador para outro, exibindo o endereço IP e o tempo de resposta de cada *router* ao longo desse percurso.

Ao analisar a Figura 2, pode-se concluir que foi possível obter resposta de todos os *routers* para determinados pacotes.

```
root@HOME-PC:/tmp/pycore.32835/HOME-PC.conf# traceroute 10.0.6.10
traceroute to 10.0.6.10 (10.0.6.10), 30 hops max, 60 byte packets
 1  10.0.5.1 (10.0.5.1)  5.880 ms  5.565 ms *
 2  10.0.0.1 (10.0.0.1)  7.482 ms  7.518 ms  7.893 ms
 3  10.0.1.1 (10.0.1.1)  8.688 ms  10.032 ms  10.028 ms
 4  10.0.3.2 (10.0.3.2)  10.409 ms  10.449 ms  10.780 ms
 5  10.0.6.10 (10.0.6.10)  12.064 ms  12.829 ms *
```

Figura 2: `Traceroute` 10.0.6.10 (SERVER).

A seguir, utiliza-se o comando `ping` para testar a conectividade entre o *HOME-PC* e o *SERVER*. Conforme ilustrado na Figura 3 e na Figura 4, envia-se 20 pacotes com 64 bytes e 1024 bytes, obtendo um RTT (Round-trip time) médio de 12,071 ms e 5,657 ms, respetivamente. Calcula-se também os valores teóricos esperados, como evidenciado nos cálculos a seguir.

```
PING 10.0.6.10 (10.0.6.10) 64(92) bytes of data.
72 bytes from 10.0.6.10: icmp_seq=1 ttl=60 time=4.92 ms
72 bytes from 10.0.6.10: icmp_seq=2 ttl=60 time=5.61 ms
72 bytes from 10.0.6.10: icmp_seq=3 ttl=60 time=6.58 ms
72 bytes from 10.0.6.10: icmp_seq=4 ttl=60 time=5.61 ms
72 bytes from 10.0.6.10: icmp_seq=5 ttl=60 time=5.01 ms
72 bytes from 10.0.6.10: icmp_seq=6 ttl=60 time=6.13 ms
72 bytes from 10.0.6.10: icmp_seq=7 ttl=60 time=5.71 ms
72 bytes from 10.0.6.10: icmp_seq=8 ttl=60 time=5.00 ms
72 bytes from 10.0.6.10: icmp_seq=9 ttl=60 time=6.14 ms
72 bytes from 10.0.6.10: icmp_seq=10 ttl=60 time=6.77 ms
72 bytes from 10.0.6.10: icmp_seq=11 ttl=60 time=5.12 ms
72 bytes from 10.0.6.10: icmp_seq=12 ttl=60 time=6.08 ms
72 bytes from 10.0.6.10: icmp_seq=13 ttl=60 time=5.15 ms
72 bytes from 10.0.6.10: icmp_seq=14 ttl=60 time=5.56 ms
72 bytes from 10.0.6.10: icmp_seq=15 ttl=60 time=5.62 ms
72 bytes from 10.0.6.10: icmp_seq=16 ttl=60 time=5.99 ms
72 bytes from 10.0.6.10: icmp_seq=17 ttl=60 time=5.02 ms
72 bytes from 10.0.6.10: icmp_seq=18 ttl=60 time=5.02 ms
72 bytes from 10.0.6.10: icmp_seq=19 ttl=60 time=5.62 ms
72 bytes from 10.0.6.10: icmp_seq=20 ttl=60 time=5.06 ms

--- 10.0.6.10 ping statistics ---
20 packets transmitted, 20 received, 0% packet loss, time 19044ms
rtt min/avg/max/mdev = 4.917/5.584/6.772/0.542 ms
```

Figura 3: Ping *HOME-PC* para *SERVER* (20 pacotes de 64 bytes cada).

```
root@HOME-PC:/tmp/pycore.32835/HOME-PC.conf# ping -c 20 -s 1024 10.0.6.10
PING 10.0.6.10 (10.0.6.10) 1024(1052) bytes of data.
1032 bytes from 10.0.6.10: icmp_seq=1 ttl=60 time=5.75 ms
1032 bytes from 10.0.6.10: icmp_seq=2 ttl=60 time=6.31 ms
1032 bytes from 10.0.6.10: icmp_seq=3 ttl=60 time=5.03 ms
1032 bytes from 10.0.6.10: icmp_seq=4 ttl=60 time=5.09 ms
1032 bytes from 10.0.6.10: icmp_seq=5 ttl=60 time=5.13 ms
1032 bytes from 10.0.6.10: icmp_seq=6 ttl=60 time=5.85 ms
1032 bytes from 10.0.6.10: icmp_seq=7 ttl=60 time=5.92 ms
1032 bytes from 10.0.6.10: icmp_seq=8 ttl=60 time=5.53 ms
1032 bytes from 10.0.6.10: icmp_seq=9 ttl=60 time=5.13 ms
1032 bytes from 10.0.6.10: icmp_seq=10 ttl=60 time=5.44 ms
1032 bytes from 10.0.6.10: icmp_seq=11 ttl=60 time=6.40 ms
1032 bytes from 10.0.6.10: icmp_seq=12 ttl=60 time=5.13 ms
1032 bytes from 10.0.6.10: icmp_seq=13 ttl=60 time=6.17 ms
1032 bytes from 10.0.6.10: icmp_seq=14 ttl=60 time=6.28 ms
1032 bytes from 10.0.6.10: icmp_seq=15 ttl=60 time=7.78 ms
1032 bytes from 10.0.6.10: icmp_seq=16 ttl=60 time=5.04 ms
1032 bytes from 10.0.6.10: icmp_seq=17 ttl=60 time=5.10 ms
1032 bytes from 10.0.6.10: icmp_seq=18 ttl=60 time=5.12 ms
1032 bytes from 10.0.6.10: icmp_seq=19 ttl=60 time=5.04 ms
1032 bytes from 10.0.6.10: icmp_seq=20 ttl=60 time=5.90 ms

--- 10.0.6.10 ping statistics ---
20 packets transmitted, 20 received, 0% packet loss, time 19048ms
rtt min/avg/max/mdev = 5.026/5.657/7.780/0.681 ms
```

Figura 4: Ping *HOME-PC* para *SERVER* (20 pacotes de 1024 bytes cada).

De seguida é apresentado os valores teóricos obtidos para 64 bytes e 1024 bytes, respetivamente, observando que os valores são ligeiramente inferiores. Do *HOME-PC* até ao *SERVER*, os pacotes percorrem cinco ligações, algumas com diferentes larguras de banda e atrasos, exigindo o cálculo do tempo de transmissão para cada uma delas. Para os cálculos de 72 bytes (64 bytes mais 8 bytes do cabeçalho ICMPv4), são apresentados três tempos de transmissão devido a algumas ligações compartilharem a mesma largura de banda. Para cada ligação, calcula-se o RTT e soma-se os valores para determinar o tempo total que cada pacote leva a percorrer do *HOME-PC* ao *SERVER* e retornar.

A equação do tempo de transmissão é apresentada na equação seguinte:

$$\text{Tempo de transmissão} = \frac{L}{R} \tag{1}$$

Utilizando a Equação 1:

$$\mathrm{T_{tx1}} = \frac{72 \text{ bytes} \times 8}{100 \times 10^6} = 5,76 \times 10^{-6} \text{ s}$$

$$\mathrm{T_{tx2}} = \frac{72 \text{ bytes} \times 8}{200 \times 10^6} = 2,88 \times 10^{-6} \text{ s}$$

$$\mathrm{T_{tx3}} = \frac{72 \text{ bytes} \times 8}{1 \times 10^9} = 5,76 \times 10^{-7} \text{ s}$$

A equação do RTT (Round-trip time) é apresentada na equação seguinte:

$$\text{RTT} = 2 \times \text{Tempo de propagação} + \text{Tempo de Transmissão} + \text{Tempo de Receção} \tag{2}$$

Se o tempo de transmissão for igual ao tempo de receção então a formula pode se simplificar para:

$$\text{RTT} = 2 \times (\text{Tempo de propagação} + \text{Tempo de Transmissão}) \tag{3}$$

Utilizando as Equações 2 e 3:

$$\text{RTT}_1 = 2 \times (10 \times 10^{-6} + 5,76 \times 10^{-6}) = 3,152 \times 10^{-5} \text{ s}$$

$$\text{RTT}_2 = 2 \times 1 \times 10^{-3} + 2,88 \times 10^{-6} + 5,76 \times 10^{-6} = 2,01 \times 10^{-3} \text{ s}$$

$$\text{RTT}_3 = 2 \times (100 \times 10^{-6} + 5,76 \times 10^{-7}) = 2,01 \times 10^{-4} \text{ s}$$

$$\text{RTT}_4 = 2 \times (100 \times 10^{-6} + 5,76 \times 10^{-7}) = 2,01 \times 10^{-4} \text{ s}$$

$$\text{RTT}_5 = 2 \times (50 \times 10^{-6} + 5,76 \times 10^{-7}) = 1,01 \times 10^{-4} \text{ s}$$

Sendo o total apresentado com a seguinte equação:

$$\text{TOTAL} = \text{RTT}_1 + \text{RTT}_2 + \text{RTT}_3 + \text{RTT}_4 + \text{RTT}_5 \tag{4}$$

$$\text{TOTAL} = 3,152 \times 10^{-5} + 2,01 \times 10^{-3} + 2,01 \times 10^{-4} + 2,01 \times 10^{-4} + 1,01 \times 10^{-4}$$

$$\Leftrightarrow \text{TOTAL} = 2,54 \times 10^{-3} \text{ s}$$

Para os cálculos de 1032 bytes (1024 bytes mais 8 bytes do cabeçalho ICMPv4), são também apresentados três tempos de transmissão devido a algumas ligações compartilharem a mesma largura de banda. Para cada ligação, calcula-se o RTT e soma-se os valores para determinar o tempo total que cada pacote leva a percorrer do *HOME-PC* ao *SERVER* e retornar.

Utilizando a Equação 1:

$$\text{T}_{\text{tx1}} = \frac{1032 \text{ bytes} \times 8}{100 \times 10^6} = 8,26 \times 10^{-5} \text{ s}$$

$$\text{T}_{\text{tx2}} = \frac{1032 \text{ bytes} \times 8}{200 \times 10^6} = 4,13 \times 10^{-5} \text{ s}$$

$$\text{T}_{\text{tx3}} = \frac{1032 \text{ bytes} \times 8}{1 \times 10^9} = 8,26 \times 10^{-6} \text{ s}$$

Utilizando as Equações 2 e 3:

$$\text{RTT}_1 = 2 \times (10 \times 10^{-6} + 8,26 \times 10^{-5}) = 1,852 \times 10^{-4} \text{ s}$$

$$\text{RTT}_2 = 2 \times 1 \times 10^{-3} + 4,13 \times 10^{-5} + 8,26 \times 10^{-5} = 2,12 \times 10^{-3} \text{ s}$$

$$\text{RTT}_3 = 2 \times (100 \times 10^{-6} + 8,26 \times 10^{-6}) = 2,17 \times 10^{-4} \text{ s}$$

$$\text{RTT}_4 = 2 \times (100 \times 10^{-6} + 8,26 \times 10^{-6}) = 2,17 \times 10^{-4} \text{ s}$$

$$\text{RTT}_5 = 2 \times (50 \times 10^{-6} + 8,26 \times 10^{-6}) = 1,17 \times 10^{-4} \text{ s}$$

Sendo o total apresentado novamente usando a Equação 4:

$$\text{TOTAL} = 1,852 \times 10^{-4} + 2,12 \times 10^{-3} + 2,17 \times 10^{-4} + 2,17 \times 10^{-4} + 1,17 \times 10^{-4}$$

$$\Leftrightarrow \text{TOTAL} = 2,87 \times 10^{-3} \text{ s}$$

Pode-se concluir que o RTT para 1024 bytes é maior do que para 64 bytes devido ao tamanho maior do arquivo enviado. Conforme demonstrado pela equação do tempo de transmissão, quanto maior o tamanho do arquivo, maior é o tempo necessário. Portanto, ao somar todos os tempos, obteu-se um valor maior, resultando numa diferença entre os dois tempos. Neste caso específico, a diferença é de 0,33 ms.

A diferença entre os valores teóricos e os valores práticos é de cerca de 3 ms, esta diferença já é esperada e justificável pois na prática existe mais tráfego na rede para além do pacote que está a enviar (congestionamento de tráfego) e também existem outros intervalos que não são tidos em conta, como por exemplo o intervalo de tempo que o *router* demora até enviar um pacote após a sua receção.

B. Modificação para Ligação Assimétrica e Novo Teste de Atraso

Posteriormente, altera-se as características da Ligação de Acesso – Residencial, de modo a tornar a ligação assimétrica como pode-se observar na Figura 5. Ajusta-se a largura de banda do *downstream* para 1 $Gbps$ e do *upstream* para 256 $Kbps$.

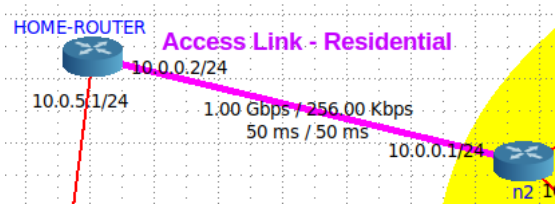

Figura 5: Mudança das características da Ligação de Acesso - Residencial.

A seguir, utiliza-se o comando `ping` para testar a conectividade entre o *HOME-PC* e o *SERVER*. Conforme ilustrado na Figura 6 e na Figura 7, envia-se 20 pacotes com 64 bytes e 1024 bytes, obtendo um RTT (Round-trip time) médio de 117,722 ms e 146,839 ms, respetivamente.

```
root@HOME-PC:/tmp/pycore.33593/HOME-PC.conf# ping -c 20 -s 64 10.0.6.10
PING 10.0.6.10 (10.0.6.10) 64(92) bytes of data.
72 bytes from 10.0.6.10: icmp_seq=1 ttl=60 time=108 ms
72 bytes from 10.0.6.10: icmp_seq=2 ttl=60 time=107 ms
72 bytes from 10.0.6.10: icmp_seq=3 ttl=60 time=107 ms
72 bytes from 10.0.6.10: icmp_seq=4 ttl=60 time=108 ms
72 bytes from 10.0.6.10: icmp_seq=5 ttl=60 time=109 ms
72 bytes from 10.0.6.10: icmp_seq=6 ttl=60 time=107 ms
72 bytes from 10.0.6.10: icmp_seq=7 ttl=60 time=114 ms
72 bytes from 10.0.6.10: icmp_seq=8 ttl=60 time=107 ms
72 bytes from 10.0.6.10: icmp_seq=9 ttl=60 time=202 ms
72 bytes from 10.0.6.10: icmp_seq=10 ttl=60 time=148 ms
72 bytes from 10.0.6.10: icmp_seq=11 ttl=60 time=152 ms
72 bytes from 10.0.6.10: icmp_seq=12 ttl=60 time=107 ms
72 bytes from 10.0.6.10: icmp_seq=13 ttl=60 time=107 ms
72 bytes from 10.0.6.10: icmp_seq=14 ttl=60 time=123 ms
72 bytes from 10.0.6.10: icmp_seq=15 ttl=60 time=108 ms
72 bytes from 10.0.6.10: icmp_seq=16 ttl=60 time=107 ms
72 bytes from 10.0.6.10: icmp_seq=17 ttl=60 time=107 ms
72 bytes from 10.0.6.10: icmp_seq=18 ttl=60 time=112 ms
72 bytes from 10.0.6.10: icmp_seq=19 ttl=60 time=107 ms
72 bytes from 10.0.6.10: icmp_seq=20 ttl=60 time=108 ms

--- 10.0.6.10 ping statistics ---
20 packets transmitted, 20 received, 0% packet loss, time 19042ms
rtt min/avg/max/mdev = 106.544/117.722/201.901/23.180 ms
```

Figura 6: Ping *HOME-PC* para *SERVER* (20 pacotes de 64 bytes cada).

```
root@HOME-PC:/tmp/pycore.33593/HOME-PC.conf# ping -c 20 -s 1024 10.0.6.10
PING 10.0.6.10 (10.0.6.10) 1024(1052) bytes of data.
1032 bytes from 10.0.6.10: icmp_seq=1 ttl=60 time=142 ms
1032 bytes from 10.0.6.10: icmp_seq=2 ttl=60 time=171 ms
1032 bytes from 10.0.6.10: icmp_seq=3 ttl=60 time=185 ms
1032 bytes from 10.0.6.10: icmp_seq=4 ttl=60 time=219 ms
1032 bytes from 10.0.6.10: icmp_seq=5 ttl=60 time=137 ms
1032 bytes from 10.0.6.10: icmp_seq=6 ttl=60 time=138 ms
1032 bytes from 10.0.6.10: icmp_seq=7 ttl=60 time=137 ms
1032 bytes from 10.0.6.10: icmp_seq=8 ttl=60 time=145 ms
1032 bytes from 10.0.6.10: icmp_seq=9 ttl=60 time=138 ms
1032 bytes from 10.0.6.10: icmp_seq=10 ttl=60 time=141 ms
1032 bytes from 10.0.6.10: icmp_seq=11 ttl=60 time=139 ms
1032 bytes from 10.0.6.10: icmp_seq=12 ttl=60 time=139 ms
1032 bytes from 10.0.6.10: icmp_seq=13 ttl=60 time=137 ms
1032 bytes from 10.0.6.10: icmp_seq=14 ttl=60 time=143 ms
1032 bytes from 10.0.6.10: icmp_seq=15 ttl=60 time=139 ms
1032 bytes from 10.0.6.10: icmp_seq=16 ttl=60 time=136 ms
1032 bytes from 10.0.6.10: icmp_seq=17 ttl=60 time=138 ms
1032 bytes from 10.0.6.10: icmp_seq=18 ttl=60 time=139 ms
1032 bytes from 10.0.6.10: icmp_seq=19 ttl=60 time=137 ms
1032 bytes from 10.0.6.10: icmp_seq=20 ttl=60 time=138 ms

--- 10.0.6.10 ping statistics ---
20 packets transmitted, 20 received, 0% packet loss, time 19041ms
rtt min/avg/max/mdev = 136.195/146.839/218.622/20.519 ms
```

Figura 7: Ping *HOME-PC* para *SERVER* (20 pacotes de 1024 bytes cada).

III. Análise de Desempenho

PARA realizar a análise de desempenho das características e funcionalidades da ferramenta *iPerf/iPerf3* no núcleo [2], irá ser executadp o comando `iperf` e o comando `iperf3` para obter a largura de banda máxima (BWmax) alcançável em transferências de dados sobre o IP(v4) entre o *HOME-PC* e o *Server*, tanto em TCP (Transmission Control Protocol) como em UDP (User Datagram Protocol).

A. Medição da Largura de Banda com `iPerf` entre HOME-PC e SERVER

Na Figura 8 observa-se a transferência de dados sobre IP(v4), entre **HOME-PC** e o **SERVER**, em TCP a largura de banda máxima é 3,81 $Mbits/sec$.

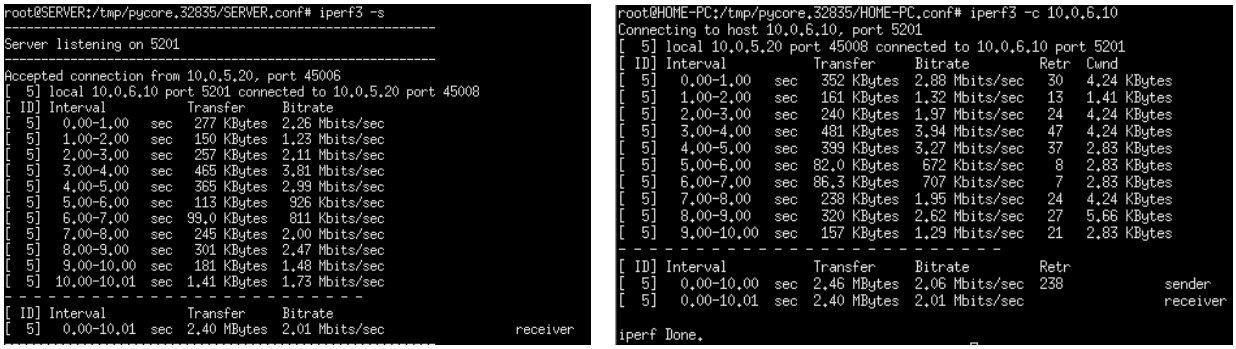

Figura 8: Transferência de dados entre o HOME-PC e o *server* em TCP.

Na Figura 9 observa-se a transferência de dados sobre IP(v4), entre **HOME-PC** e o **SERVER**, em UDP a largura de banda máxima é 1.08 $Mbits/sec$.

Figura 9: Transferência de dados entre o HOME-PC e o *server* em UDP.

B. Exploração dos Parâmetros de Configuração no Link Residencial

1) Configuração de Parâmetros de Rede: Para esta etapa do estudo, realizou-se a modificação de um único parâmetro por vez, incluindo o atraso (*delay*), a perda de pacotes (*loss*) e a duplicação de pacotes (*duplicate*). Foram testados diferentes valores para cada parâmetro: 4% e 10% para perda de pacotes, e 0%, 2% e 5% para duplicação de pacotes. Utiliza-se a ferramenta `iperf` para realizar transferências de dados em ambos os protocolos, UDP e TCP, após cada modificação.

- *Loss* 4% e *Duplicate* 0% (TCP), Figura 10:

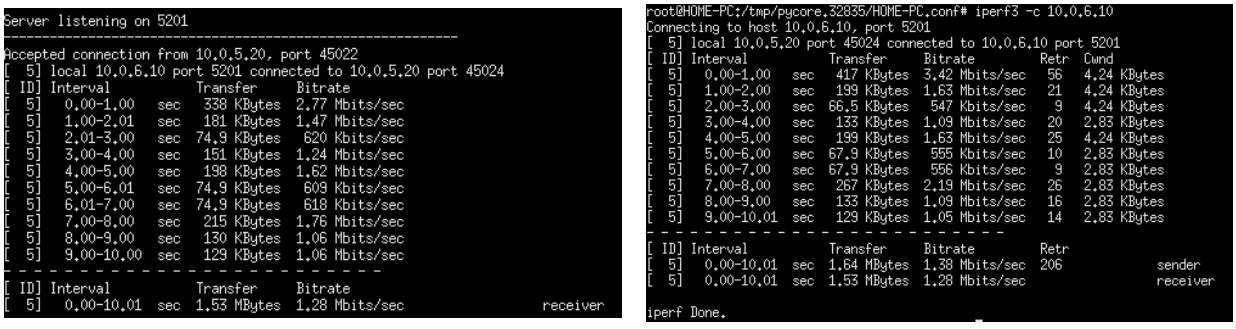

Figura 10: Loss 4% com Duplicate 0% (TCP).

- *Loss* 4% e *Duplicate* 0% (UDP), Figura 11:

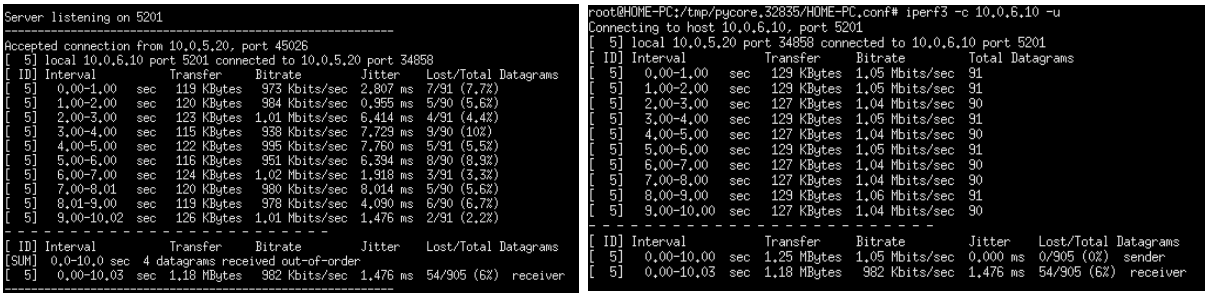

Figura 11: Loss 4% com Duplicate 0% (UDP).

- *Loss* 4% e *Duplicate* 2% (TCP), Figura 12:

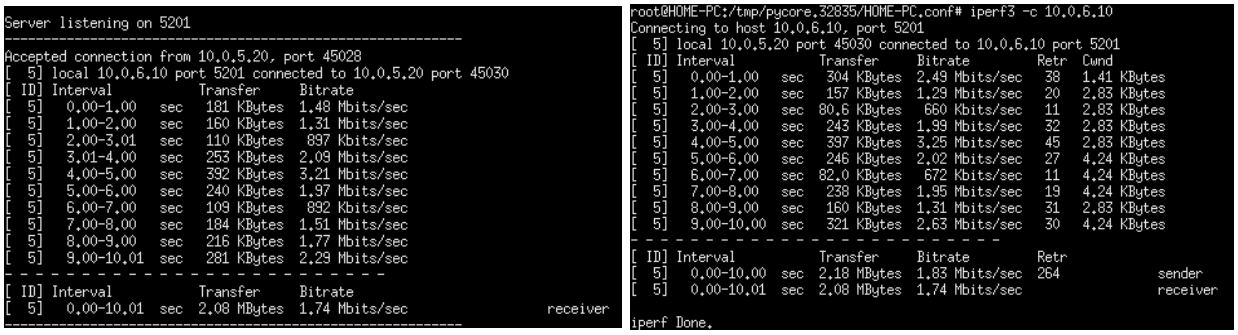

Figura 12: Loss 4% com Duplicate 2% (TCP).

1) **Variações na Taxa de Transferência:** Ambos os lados (servidor e cliente) apresentam variações significativas na taxa de transferência durante os intervalos de tempo. Isto pode ser atribuído a flutuações na rede, congestionamento ou interferência.
2) **Taxa de Transferência Média:** O servidor alcançou uma taxa média de transferência de aproximadamente 1.74 $Mbits/sec$, enquanto o cliente registou uma média ligeiramente superior, em torno de 1.83 $Mbits/sec$.
3) **Impacto do Loss e Duplicate:** Os valores de Loss (4%) e Duplicate (2%) podem afetar a qualidade da conexão, resultando em retransmissões e potencialmente afetando a taxa de transferência e a estabilidade da conexão. Isto pode ser observado nas variações na taxa de transferência ao longo do tempo.
4) **Estabilidade da Conexão:** As variações na taxa de transferência ao longo dos intervalos indicam uma certa instabilidade na conexão entre o cliente e o servidor. Isto pode ser uma consequência direta do Loss e Duplicate, exigindo retransmissões e impactando o desempenho geral.

- *Loss* 4% e *Duplicate* 2% (UDP), Figura 13:

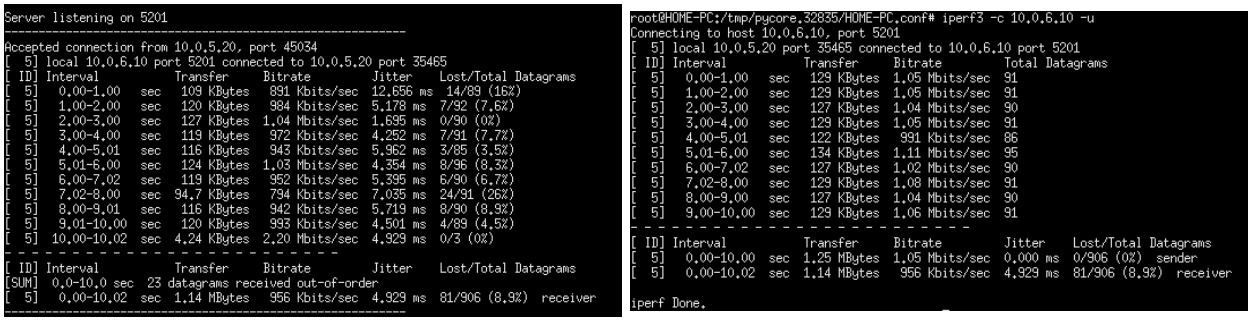

Figura 13: Loss 4% com Duplicate 2% (UDP).

1) **Taxa de Transferência:** A taxa de transferência média registada tanto no servidor quanto no cliente é razoável, variando em torno de 950 a 1050 $Kbits/sec$. Isto sugere que, apesar das perdas e duplicações, a comunicação UDP ainda conseguiu transferir uma quantidade significativa de dados.

2) ***Jitter*:** O *jitter* é a variação no atraso dos pacotes recebidos. Valores mais baixos de *jitter* são desejáveis, pois indicam uma transmissão mais estável. No entanto, o *jitter* registado não é muito alto, o que sugere uma certa consistência na entrega dos pacotes.
3) **Pacotes Perdidos:** O número de pacotes perdidos varia entre os intervalos, com o servidor registando taxas de perda entre 0% e 26%. Isto pode indicar problemas na rede, congestionamento ou outras condições adversas que afetam a entrega dos pacotes.
4) **Impacto de Loss e Duplicate:** A presença de perdas (Loss) e duplicações (Duplicate) afeta diretamente a qualidade da transmissão UDP. Isto pode levar a retransmissões de pacotes e, consequentemente, a uma redução na taxa de transferência e aumento do *jitter*.
5) **Estabilidade da Conexão:** Apesar das perdas e duplicações, a conexão UDP parece manter uma certa estabilidade, como evidenciado pela taxa de transferência e pelo *jitter* relativamente controlado.

- *Loss* 4% e *Duplicate* 5% (TCP), Figura 14:

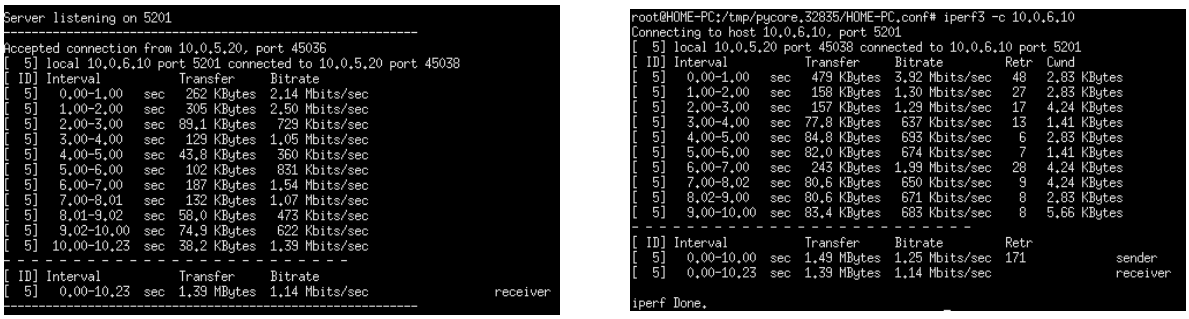

Figura 14: Loss 4% com Duplicate 5% (TCP).

1) **Taxa de Transferência:** Ambos os lados (servidor e cliente) exibem taxas de transferência variadas durante os intervalos de tempo. A média de taxa de transferência do servidor é de aproximadamente 1.14 $Mbits/sec$, enquanto a do cliente é de cerca de 1.25 $Mbits/sec$.
2) **Retransmissões e Congestionamento:** O servidor experimentou um número significativo de retransmissões em diferentes intervalos, variando de 7 a 28 retransmissões. Isto pode indicar congestionamento na rede ou problemas de conectividade entre o cliente e o servidor.
3) **Impacto do Loss e Duplicate:** A combinação de valores de Loss (4 %) e Duplicate (5 %) pode causar interrupções na transferência de dados, resultando em retransmissões e afetando negativamente a taxa de transferência e a eficiência da conexão TCP.
4) **Estabilidade da Conexão:** As flutuações na taxa de transferência e o número variável de retransmissões sugerem que a conexão entre o cliente e o servidor pode não ser muito estável. Isto pode ser atribuído às condições de rede, incluindo perda de pacotes e duplicação.

- *Loss* 4% e *Duplicate* 5% (UDP), Figura 15

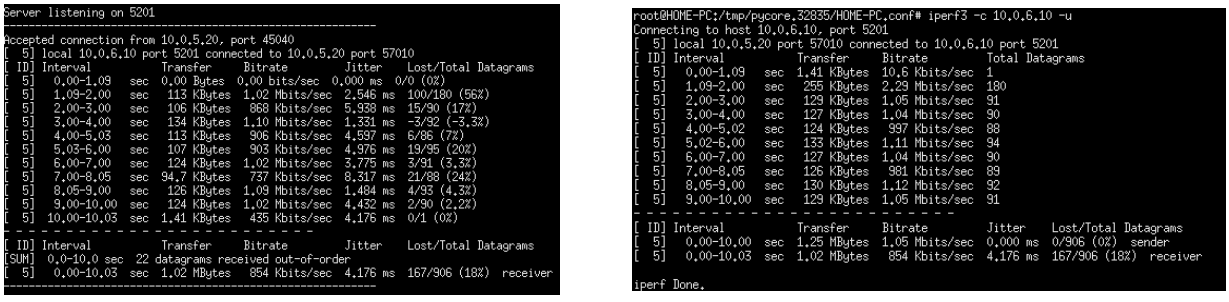

Figura 15: Loss 4% com Duplicate 5% (UDP).

1) **Taxa de Transferência e Perda de Pacotes:** A taxa de transferência e a perda de pacotes variam significativamente entre os intervalos de tempo. Isto indica a instabilidade da conexão UDP entre o cliente e o servidor. A perda de pacotes é alta em alguns intervalos, o que pode ser causado por congestionamento na rede ou problemas de conectividade.

2) ***Jitter*:** O *jitter* é a variação no atraso entre os pacotes recebidos. Valores mais altos de *jitter* podem indicar inconsistências na latência da rede. Os resultados mostram uma variação considerável no *jitter* entre os intervalos, sugerindo uma conexão instável.
3) **Impacto de Loss e Duplicate:** A combinação de valores de Loss (4 %) e Duplicate (5 %) pode afetar a entrega de pacotes e causar interrupções na transmissão de dados. Isto é evidenciado pelos altos índices de perda de pacotes e *jitter* variável.
4) **Performance Geral:** A taxa de transferência média é de aproximadamente 854 $Kbits/sec$, o que pode ser considerado razoável para uma conexão UDP sujeita a perdas e duplicações. No entanto, a alta perda de pacotes e a instabilidade da conexão podem afetar a experiência do utilizador em aplicações sensíveis à latência ou que exigem transmissão contínua de dados.

- *Loss* 10% e *Duplicate* 0% (TCP), Figura 16:

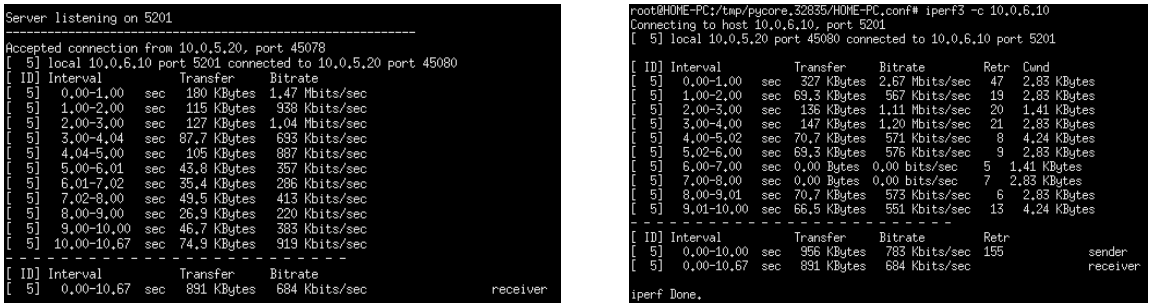

Figura 16: Loss 10% com Duplicate 0% (TCP).

- *Loss* 10% e *Duplicate* 0% (UDP), Figura 17:

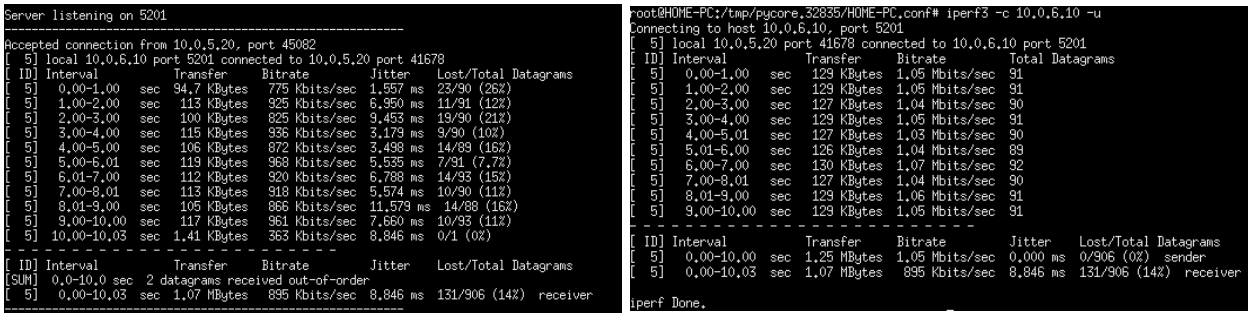

Figura 17: Loss 10% com Duplicate 0% (UDP).

- *Loss* 10% e *Duplicate* 2% (TCP), Figura 18

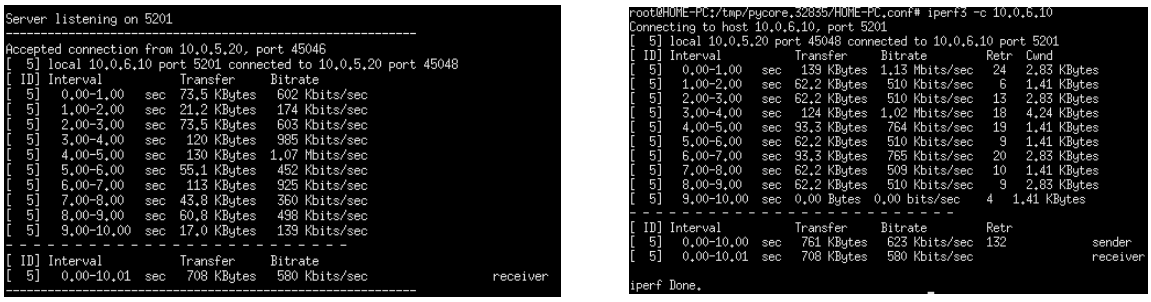

Figura 18: Loss 10% com Duplicate 2% (TCP).

1) **Taxa de Transferência:** A taxa de transferência varia consideravelmente entre os intervalos, oscilando entre valores relativamente baixos e altos. Isto pode indicar flutuações na qualidade da rede ou na carga de tráfego.
2) **Retransmissões e Congestionamento:** O número de retransmissões varia ao longo do tempo, indicando momentos de possível congestionamento ou instabilidade na conexão. O tamanho da janela de congestionamento também varia, sugerindo adaptação dinâmica à condição da rede.

3) **Estabilidade da Conexão:** A conexão parece instável, com flutuações na taxa de transferência e retransmissões. Isto pode ser devido a uma variedade de fatores, incluindo congestionamento na rede, perda de pacotes ou condições variáveis da rede.
4) **Performance Geral:** A taxa de transferência média durante todo o teste foi de aproximadamente 580 $Kbits/sec$.

- *Loss* 10% e *Duplicate* 2% (UDP), Figura 19

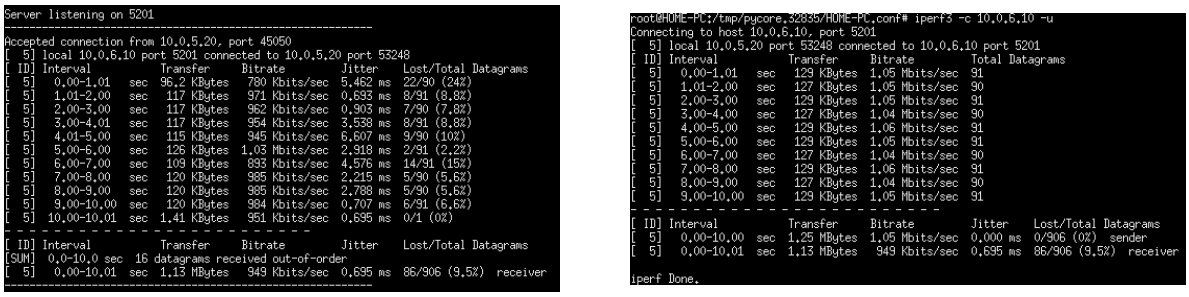

Figura 19: Loss 10% com Duplicate 2% (UDP).

1) **Taxa de Transferência e Perda de Pacotes:** A taxa de transferência média é de cerca de 949 $Kbits/sec$, enquanto a perda média de pacotes é de aproximadamente 9.5 %. Isto sugere uma conexão UDP instável, com uma quantidade significativa de pacotes perdidos.
2) ***Jitter*:** O *jitter* varia entre os intervalos, mas em geral, é mantido em níveis baixos, indicando uma consistência razoável na latência da rede durante a transmissão.
3) **Impacto de Loss e Duplicate:** A combinação de valores de Loss (10 %) e Duplicate (2 %) resulta numa quantidade considerável de pacotes perdidos, o que pode ser atribuído ao alto valor de perda. Os pacotes duplicados também podem estar contribuindo para a perda geral de eficiência da rede.
4) **Performance Geral:** Embora a taxa de transferência seja relativamente alta, a alta taxa de perda de pacotes pode afetar negativamente a qualidade da transmissão de dados em aplicações sensíveis à integridade dos dados, como chamadas de voz ou vídeo em tempo real.

- *Loss* 10% e *Duplicate* 5% (TCP), Figura 20:

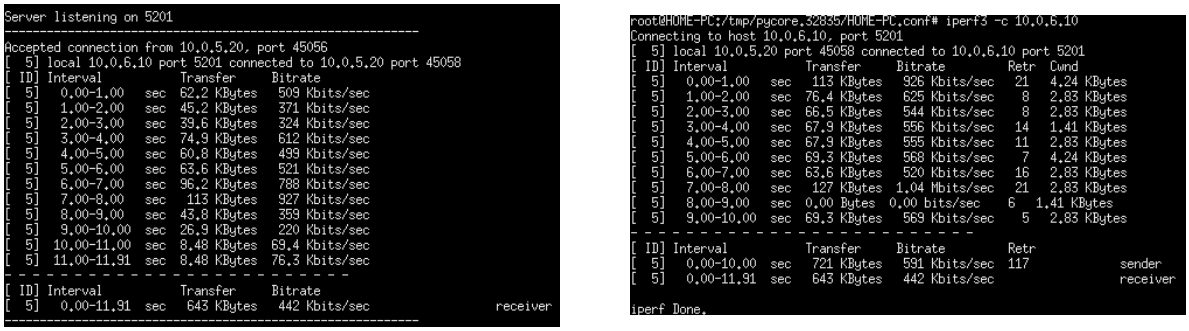

Figura 20: Loss 10% com Duplicate 5% (TCP).

1) **Taxa de Transferência:** A taxa de transferência média é de aproximadamente 442 $Kbits/sec$, o que é relativamente baixo para uma conexão TCP. Isso sugere uma possível interferência na rede ou limitação da largura de banda.
2) **Comportamento Variável:** A taxa de transferência varia significativamente em diferentes intervalos de tempo, o que pode indicar flutuações na qualidade da rede ou na carga de tráfego durante o teste.
3) **Impacto do Loss e Duplicate:** A alta taxa de perda de pacotes (10 %) e a quantidade de pacotes duplicados (5 %) podem estar a contribuir para a baixa eficiência da transmissão de dados em TCP. Isso pode ser visto na inconsistência das taxas de transferência ao longo do tempo.
4) **Retransmissões:** O número de retransmissões é notavelmente alto em alguns intervalos, o que sugere problemas de congestionamento ou perda de pacotes na rede.
5) **Performance Geral:** A taxa de transferência média é abaixo do esperado para uma conexão TCP normal. Isto pode indicar problemas na rede ou na configuração dos hosts.

- *Loss* 10% e *Duplicate* 5% (UDP), Figura 21:

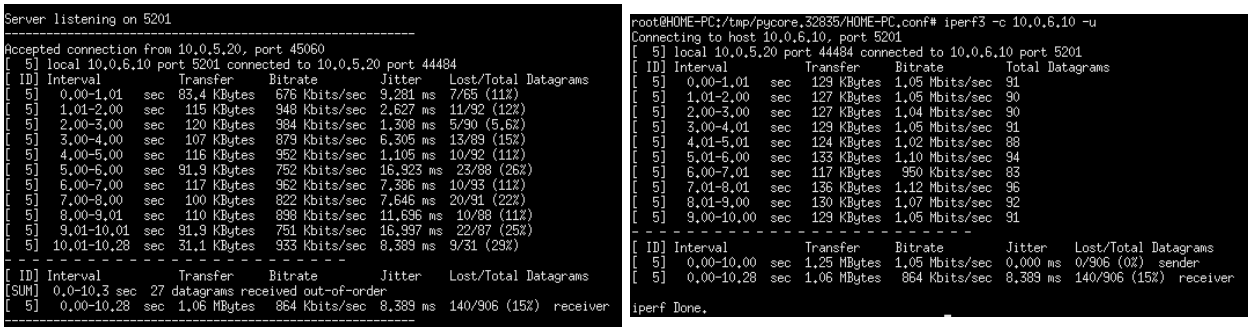

Figura 21: Loss 10% com Duplicate 5% (UDP).

1) **Taxa de Transferência:** A taxa de transferência média é de aproximadamente 864 $Kbits/sec$, o que é razoável para uma conexão UDP. No entanto, poderia ser mais alta em uma rede sem perdas ou duplicações.
2) ***Jitter*:** O *jitter* médio é de 8.389 ms, o que indica alguma variabilidade nos atrasos de entrega dos pacotes. Isto pode ser causado pela perda e duplicação de pacotes na rede.
3) **Perda de Pacotes:** A perda de pacotes é significativa, com uma média de 15% dos pacotes perdidos durante a transmissão. Isto pode ser atribuído à alta taxa de perda configurada.
4) **Impacto de Loss e Duplicate:** Os valores altos de perda (10 %) e duplicação (5 %) estão causando uma diminuição na confiabilidade da transmissão. Isto é evidente pela quantidade significativa de pacotes perdidos e pelo aumento no *jitter*.
5) **Performance Geral:** Apesar da perda de pacotes e duplicações, a taxa de transferência e o *jitter* ainda estão dentro de limites aceitáveis para uma conexão UDP. No entanto, a alta taxa de perda pode impactar negativamente em aplicações sensíveis à latência ou que requerem uma entrega confiável de dados.

2) Modificação de Parâmetros e Testes com iPerf: A Tabela I apresenta os resultados obtidos nas Figuras 10 a 21 que envolveu transmissão de dados utilizando os protocolos TCP e UDP em diferentes condições de perda de pacotes e duplicação de pacotes.

Tabela I: Resultados obtidos nas images das Figuras 10 a 21.

TCP Receiver				**UDP Receiver**			
Loss \ Duplicate	0%	2%	5%	Loss \ Duplicate	0%	2%	5%
0%	2.01 $Mbit/s$	1.83 $Mbit/s$	1.61 $Mbit/s$	0%	1.04 $Mbit/s$	1.05 $Mbit/s$	1.01 $Mbit/s$
4%	1.28 $Mbit/s$	1.74 $Mbit/s$	1.14 $Mbit/s$	4%	982 $Kbit/s$	956 $Kbit/s$	854 $Kbit/s$
10%	684 $Kbit/s$	580 $Kbit/s$	442 $Kbit/s$	10%	895 $Kbit/s$	949 $Kbit/s$	864$Kbit/s$

O aumento do parâmetro *Loss*, como o nome indica, incrementa o número de pacotes perdidos na comunicação. Diferentes protocolos lidam com essa perda de formas distintas:

- TCP, ao detetar a perda de pacotes, entra em modo de controlo de congestão, diminuindo a velocidade de transmissão para prevenir congestionamento e futuras perdas. Isto permite uma melhoria no desempenho da ligação;
- Por outro lado, o UDP, ao contrário do TCP, não dispõe de mecanismos para lidar com essas perdas. As tramas UDP são enviadas sem garantia de chegada (ou chegada ordenada). Quando um pacote é perdido com este protocolo, o transmissor da mensagem não é notificado. Isto resulta em diminuições mais significativas no desempenho da rede durante esses eventos.

Relativamente ao parâmetro *Duplicate*, que representa a duplicação de pacotes já enviados, este também afeta o desempenho da ligação. O comportamento do TCP e do UDP em relação a estas duplicações é semelhante ao das perdas:

- o TCP utiliza mecanismos para tentar lidar com a situação, descartando os duplicados e enviando um ACK para confirmar o pacote original. Isto evita futuras retransmissões, contribuindo para uma melhoria na qualidade da transmissão.
- Por outro lado, o UDP não executa esses mecanismos, não proporcionando garantias de qualidade de serviço ou de chegada ordenada dos pacotes.

3) Avaliação da Conectividade com Parâmetros Extremos: O valor do atraso de ligação foi alterado para um valor muito grande, ou seja 2 s e foram calculadas as velocidades de transmissão.

A Figura 22 mostra os resultados obtidos no TCP.

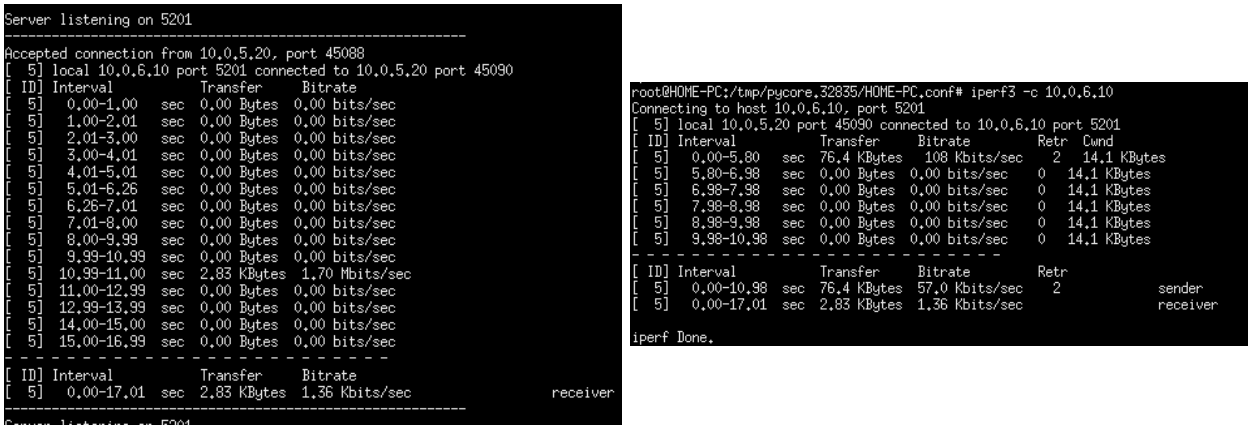

Figura 22: Resultados obtidos para 2 s no TCP.

A Figura 23 mostra os resultados obtidos no UDP.

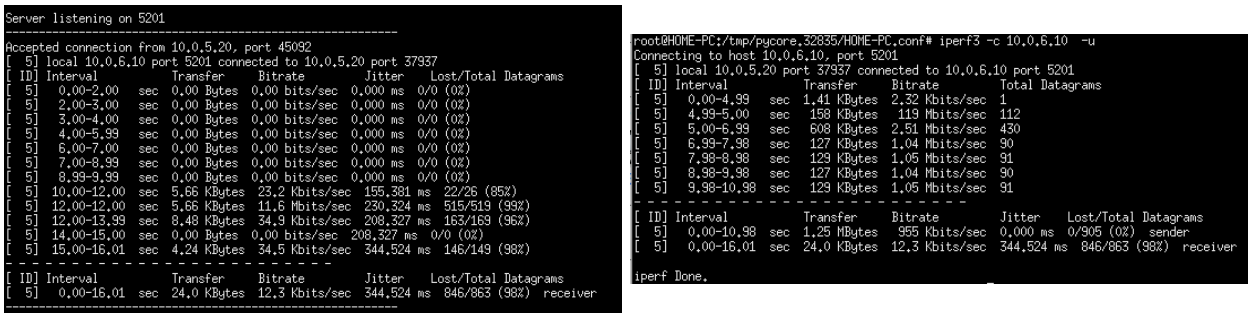

Figura 23: Resultados obtidos para 2 s no UDP.

No caso do TCP, quase nenhum dos pacotes foi recebido, evidenciando uma significativa queda na *bit rate* entre o recetor e o transmissor. Com o UDP chegam mais pacotes em comparação com o TCP. Isto acontece devido ao TCP operar num sistema que exige confirmação atempada dos pacotes. Com um atraso elevado, torna-se praticamente impossível que isso funcione de forma eficaz. O UDP, ao não garantir a chegada nem a ordem dos pacotes, simplesmente os envia sem necessidade de confirmações de receção. Isso possibilita, em situações de qualidade inferior do meio de propagação do sinal, um desempenho mais eficiente.

Dessa forma, conclui-se esta questão, destacando a validade de ambos os protocolos em diferentes situações: o TCP, ao implementar mecanismos de controlo na transmissão e receção, resulta em menos perdas, mas menor velocidade; o UDP, por não implementar tais mecanismos, pode oferecer melhor velocidade, embora sem garantias de chegada de pacotes. Em resumo, é sempre necessário avaliar o meio e a situação específica ao estabelecer uma comunicação, escolhendo o protocolo que melhor se adequa.

IV. Conclusão

O relatório descreve um estudo sobre redes de acesso IP usando o emulador Core. Foram configuradas várias topologias para avaliar características como largura de banda e atraso, com análise de conectividade global usando ferramentas como `ping` e `traceroute`. A ferramenta `iperf` foi utilizada para análise de tráfego TCP e UDP, evidenciando a sensibilidade do TCP às perdas de pacotes e a capacidade superior do UDP em lidar com atrasos de ligação. Apesar de desafios iniciais com o software Core, a orientação da docente foi crucial para alcançar os objetivos.

O projeto proporcionou uma aplicação prática dos conhecimentos teóricos, enriquecendo a compreensão das redes de acesso IP e contribuindo significativamente para o desenvolvimento técnico e prático na área.

Reconhecimento

Os autores desejam expressar a sua sincera gratidão ao Professor Doutor Flávio Oliveira Silva pela orientação e ajuda. O professor não só respondeu a todas as perguntas e esclareceu todas as dúvidas relacionadas ao trabalho prático, mas também forneceu orientação sobre o enunciado.

Referências Bibliográficas

[1] Flávio Oliveira Silva. *Guia de instalação e integração do Visual Studio Code com a máquina virtual CORE 1. Objetivos*. Acedido a 25 de fevereiro de 2024. URL: http://marco.uminho.pt/ferramentas/CORE/xubuncore.html.

[2] Jon Dugan et al. *What is iPerf / iPerf3 ?* Acedido a 28 de fevereiro de 2024. URL: https://iperf.fr.

Catarina Pereira atualmente é aluna do primeiro ano do curso de Mestrado de Engenharia de Telecomunicações e Informática na Universidade do Minho, iniciando a sua formação em licenciatura no mesmo curso no ano letivo de 2020/2021.
Além dos estudos, foi diretora do Departamento de Saídas Profissionais do NETIUM (Núcleo de Engenharia de Telecomunicações e Informática na Universidade do Minho) para o ano de 2023, tendo atuado como colaboradora do Departamento Pedagógico no ano anterior. Em 2022, ela participou de um projeto de handebol em parceria com o ABC, exercendo o papel de analista do projeto.
Catarina também tem experiência em outras áreas, incluindo participação num festival internacional de dança, conclusão de um curso de Língua Gestual Portuguesa, uma residência artística, conclusão do ensino secundário com ênfase em Ciências e Tecnologias, curso de inglês e aulas de música e ballet.

Inês Neves atualmente é aluna do primeiro ano do curso de Mestrado Engenharia de Telecomunicações e Informática na Universidade do Minho, iniciando a sua formação em licenciatura no mesmo curso no ano letivo de 2020/2021. Além dos estudos, ela atuou como vice-diretora da Assembleia Geral do NETIUM (Núcleo de Engenharia de Telecomunicações e Informática na Universidade do Minho) para o ano de 2023, tendo atuado anteriormente como colaboradora do Departamento Pedagógico e do Departamento de Comunicação e Imagem em anos anteriores, em 2021 e em 2022, respetivamente. Em 2022, ela participou num projeto de handebol em parceria com o ABC, exercendo o papel de analista do projeto.
Inês também tem experiência em outras áreas, incluindo a conclusão de um curso de inglês na Royal School of Languages, a prática de natação por muitos anos, bem como ter obtido certificado em ballet. Ela também concluiu o ensino secundário com ênfase em Ciências e Tecnologias.

Leonardo Martins é um estudante do primeiro ano do curso de Mestrado de Engenharia de Telecomunicações e Informática na Universidade do Minho, tendo iniciado a sua formação em licenciatura no mesmo curso no ano letivo de 2020/2021.
Além dos estudos, ele realizou atividades no vice-diretor do Conselho Fiscal do NETIUM (Núcleo de Engenharia de Telecomunicações e Informática na Universidade do Minho) para o ano de 2023, tendo atuado como colaborador do Departamento de Comunicação e Imagem no ano anterior. Em 2022, ele participou de um projeto de handebol em parceria com o ABC, exercendo o papel de *developer* do projeto. Leonardo também tem experiência em outras áreas e conclusão do ensino secundário com ênfase em Ciências e Tecnologias.