



Universidade do Minho
Escola de Engenharia

Redes de Acesso e Núcleo
ENGENHARIA DE TELECOMUNICAÇÕES E INFORMÁTICA
2021/2022

(Docente: Alexandre Júlio Teixeira Santos)

11 de março de 2022

Trabalho Prático 1

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

Hugo Miguel Miranda Reinolds – a83924@alunos.uminho.pt

Inês Barreira Marques – a84913@alunos.uminho.pt

Rui Filipe Ribeiro Freitas – pg47639@alunos.uminho.pt

Tiago João Pereira Ferreira – pg47692@alunos.uminho.pt

Índice

Lista de figuras.....	3
Introdução	4
Desenvolvimento.....	5
Testes e discussão de resultados.....	6
1. Teste da conectividade global	6
2. Análise de desempenho	8
2.1. Comparação da largura de banda máxima	8
2.2. Delay, Loss e Duplicate.....	9
2.3. Combinação dos parâmetros Loss e Duplicate	10
2.4. Alteração do valor do atraso.....	11
2.5. Avaliação das características Best Effort	12
Conclusão.....	13

Lista de figuras

Figura 1 - Topologia de teste.	5
Figura 2 - Ping entre home PC e server.	6
Figura 3 - Traceroute entre home PC e server.	7
Figura 4 - Valores após alteração da largura de banda da ligação residencial.	7
Figura 5 - Teste iperf com TCP entre HOME-PC e SERVER.	8
Figura 6 - Teste iperf com UDP entre HOME-PC e SERVER.	8
Figura 7 - Tabela com os valores de duplicados (TCP).	9
Figura 8 - Tabela com os valores de perdas (TCP).	9
Figura 9 - Tabela com valores de perdas (UDP).	9
Figura 10 - Tabela com os valores de duplicados (UDP).	9
Figura 11 - Tabela com a combinação dos 2 parâmetros (TCP).	10
Figura 12 - Tabela com a combinações dos 2 parâmetros (UDP).	10
Figura 13 - Tabela com os valores para um atraso de 50ms e 2s (TCP).	11
Figura 14 - Tabela com valores para um atraso de 50ms e 2s (UDP).	11

Introdução

Atualmente existem mais de 5.1 mil milhões de utilizadores ativos que acedem à internet e mais de 6.59 mil milhões de Gb atravessam a internet por segundo. Para isto ser possível é necessário uma boa construção e configuração das diferentes redes.

Esta cadeira tem como principais objetivos conhecer e perceber como funcionam as redes de acesso e núcleo. As redes de acesso caracterizam-se por permitirem aos utilizadores finais acederem a uma rede de núcleo. Já as redes de núcleo podem ser equiparadas a uma auto estrada, isto é, tem como objetivo encaminhar tráfego tão rápido quanto possível. Além disso as redes de núcleo conectam-se entre si.

Este trabalho tem como objetivo a aprendizagem de análise de desempenho de redes. A largura de banda foi a propriedade mais utilizada para avaliar a qualidade e desempenho da rede com o auxílio da alteração de parâmetros como *delay*, *loss* e *duplicate*.

Para uma melhor perceção do trabalho realizado este trabalho foi dividido em 3 fases. A primeira que passou pela implementação da topologia no ambiente CORE, a segunda que correspondeu a uma fase de testes para perceber se a topologia implementada era eficiente e funcionava como desejado e a terceira e última fase que corresponde à análise de desempenho da rede manipulando os vários parâmetros pedidos no enunciado.

Desenvolvimento

Para desenvolver a topologia de teste onde foram realizados os vários exercícios propostos no enunciado foi utilizado o ambiente CORE e desenvolvida a topologia apresentada na figura seguinte.

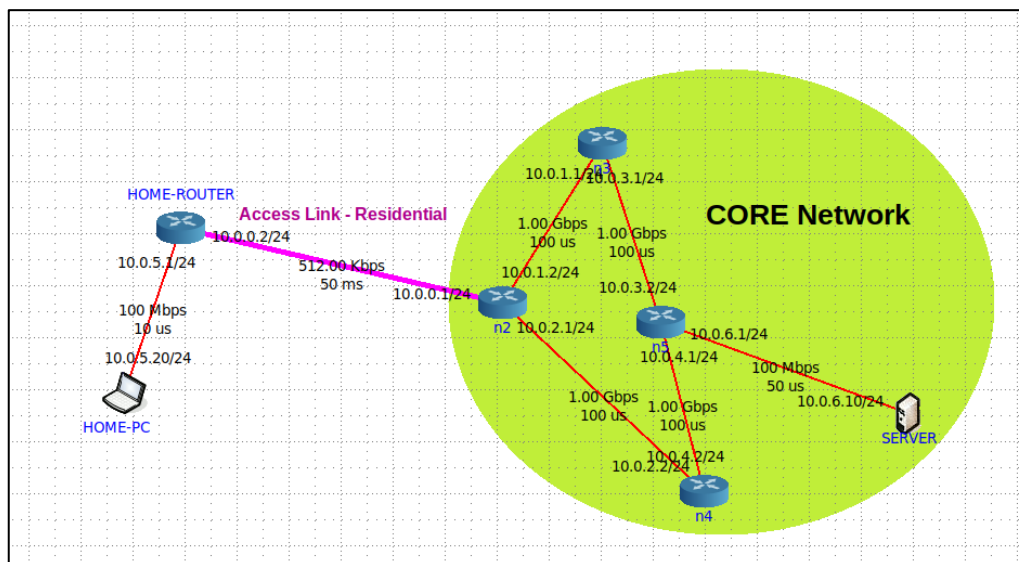


Figura 1 - Topologia de teste.

Na figura anterior podemos observar a topologia com as ligações realizadas com as métricas pedidas como por exemplo a ligação entre os routers da rede core apresentam uma largura de banda de 1 Gbps entre elas e um atraso de 100 microssegundos. Para além disso a ligação entre o home router e a rede core é realizada com uma largura de banda de 512Kbps e um atraso de 50 milissegundos e a ligação ao home PC com uma largura de banda de 100Mbps e um atraso de 10 microssegundos.

Testes e discussão de resultados

1. Teste da conectividade global

De modo a comprovar que a topologia implementada pelo grupo estava a funcionar eficientemente decidimos realizar alguns testes utilizando as ferramentas *ping* e *traceroute*.

Quanto à realização do teste do *ping* entre o home PC e o server é apresentada na figura seguinte uma captura de ecrã onde foram enviados 20 pacotes com um atraso médio de 101.708 ms. Este valor deve-se ao facto de termos adicionado *delay* nas ligações entre os *endpoints*. Em particular a ligação entre o home router e a rede core que apresenta um atraso de 50 ms. Como o tempo calculado tem em conta o valor de *round-trip* time era de esperar que no mínimo o valor fosse de 100 ms pois o tráfego passa na ligação 2 vezes.

```
root@HOME-PC: /tmp/pycore.34401/HOME-PC.conf# ping -c 20 10.0.6.10
PING 10.0.6.10 (10.0.6.10) 56(84) bytes of data.
64 bytes from 10.0.6.10: icmp_seq=1 ttl=60 time=102 ms
64 bytes from 10.0.6.10: icmp_seq=2 ttl=60 time=102 ms
64 bytes from 10.0.6.10: icmp_seq=3 ttl=60 time=102 ms
64 bytes from 10.0.6.10: icmp_seq=4 ttl=60 time=102 ms
64 bytes from 10.0.6.10: icmp_seq=5 ttl=60 time=101 ms
64 bytes from 10.0.6.10: icmp_seq=6 ttl=60 time=102 ms
64 bytes from 10.0.6.10: icmp_seq=7 ttl=60 time=102 ms
64 bytes from 10.0.6.10: icmp_seq=8 ttl=60 time=102 ms
64 bytes from 10.0.6.10: icmp_seq=9 ttl=60 time=101 ms
64 bytes from 10.0.6.10: icmp_seq=10 ttl=60 time=102 ms
64 bytes from 10.0.6.10: icmp_seq=11 ttl=60 time=102 ms
64 bytes from 10.0.6.10: icmp_seq=12 ttl=60 time=102 ms
64 bytes from 10.0.6.10: icmp_seq=13 ttl=60 time=102 ms
64 bytes from 10.0.6.10: icmp_seq=14 ttl=60 time=101 ms
64 bytes from 10.0.6.10: icmp_seq=15 ttl=60 time=102 ms
64 bytes from 10.0.6.10: icmp_seq=16 ttl=60 time=102 ms
64 bytes from 10.0.6.10: icmp_seq=17 ttl=60 time=102 ms
64 bytes from 10.0.6.10: icmp_seq=18 ttl=60 time=102 ms
64 bytes from 10.0.6.10: icmp_seq=19 ttl=60 time=102 ms
64 bytes from 10.0.6.10: icmp_seq=20 ttl=60 time=102 ms

--- 10.0.6.10 ping statistics ---
20 packets transmitted, 20 received, 0% packet loss, time 19036ms
rtt min/avg/max/mdev = 101.440/101.708/101.929/0.167 ms
```

Figura 2 - Ping entre home PC e server.

Quanto aos resultados teóricos realizamos a soma dos tempos de atraso nas ligações e foi obtido um valor de 100.52 ms. Este valor é aproximado ao obtido na prática, mas não igual devido a fatores como a quantidade de tráfego presente na rede.

Relativamente ao teste de *traceroute* podemos observar na figura seguinte os vários caminhos por onde passa o tráfego com o seu respetivo tempo. Podemos também observar que para chegar ao destino 10.0.6.10 este passa por vários saltos como o 10.0.0.1, 10.0.1.1 e 10.0.3.2.

```
root@HOME-PC:/tmp/pycore.34401/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)  0.124 ms  0.086 ms  0.154 ms
 2  10.0.0.1 (10.0.0.1)  101.067 ms  101.055 ms  101.042 ms
 3  10.0.1.1 (10.0.1.1)  101.171 ms  101.164 ms  101.155 ms
 4  10.0.3.2 (10.0.3.2)  101.365 ms  101.356 ms  101.348 ms
 5  10.0.6.10 (10.0.6.10)  101.570 ms  101.561 ms  101.553 ms
```

Figura 3 - Traceroute entre home PC e server.

Se for realizado um ajuste nas características da ligação de acesso alterando a taxa de transmissão para 1 Gbps *downstream* e 256 Kbps *upstream* são obtidos valores bastante parecidos aos obtidos anteriormente. Este acontecimento deve-se ao facto de apesar de ser atribuída uma capacidade de transmissão maior, o atraso continua presente na transmissão, fazendo com que não exista uma diferença notória nos tempos. Para além disso, o pacote que é transmitido no ping tem comprimento relativamente pequeno, o que significa que não afeta muito a largura de banda, o facto de também ser uma rede emulada, faz com que esta não possua tráfego real.

```
root@HOME-PC:/tmp/pycore.42115/HOME-PC.conf# ping -c 20 10.0.6.10
PING 10.0.6.10 (10.0.6.10) 56(84) bytes of data:
64 bytes from 10.0.6.10: icmp_seq=1 ttl=60 time=102 ms
64 bytes from 10.0.6.10: icmp_seq=2 ttl=60 time=102 ms
64 bytes from 10.0.6.10: icmp_seq=3 ttl=60 time=102 ms
64 bytes from 10.0.6.10: icmp_seq=4 ttl=60 time=102 ms
64 bytes from 10.0.6.10: icmp_seq=5 ttl=60 time=102 ms
64 bytes from 10.0.6.10: icmp_seq=6 ttl=60 time=102 ms
64 bytes from 10.0.6.10: icmp_seq=7 ttl=60 time=102 ms
64 bytes from 10.0.6.10: icmp_seq=8 ttl=60 time=101 ms
64 bytes from 10.0.6.10: icmp_seq=9 ttl=60 time=102 ms
64 bytes from 10.0.6.10: icmp_seq=10 ttl=60 time=102 ms
64 bytes from 10.0.6.10: icmp_seq=11 ttl=60 time=101 ms
64 bytes from 10.0.6.10: icmp_seq=12 ttl=60 time=102 ms
64 bytes from 10.0.6.10: icmp_seq=13 ttl=60 time=102 ms
64 bytes from 10.0.6.10: icmp_seq=14 ttl=60 time=102 ms
64 bytes from 10.0.6.10: icmp_seq=15 ttl=60 time=102 ms
64 bytes from 10.0.6.10: icmp_seq=16 ttl=60 time=102 ms
64 bytes from 10.0.6.10: icmp_seq=17 ttl=60 time=102 ms
64 bytes from 10.0.6.10: icmp_seq=18 ttl=60 time=101 ms
64 bytes from 10.0.6.10: icmp_seq=19 ttl=60 time=102 ms
64 bytes from 10.0.6.10: icmp_seq=20 ttl=60 time=102 ms

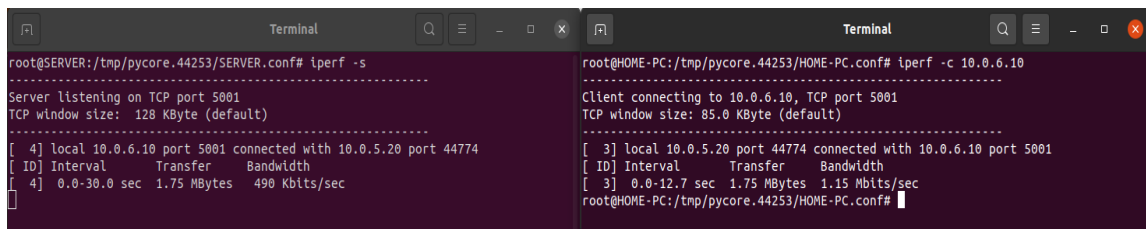
--- 10.0.6.10 ping statistics ---
20 packets transmitted, 20 received, 0% packet loss, time 19035ms
rtt min/avg/max/mdev = 101.266/101.678/101.972/0.166 ms
```

Figura 4 - Valores após alteração da largura de banda da ligação residencial.

2. Análise de desempenho

2.1. Comparação da largura de banda máxima

De modo a analisar o desempenho da rede foi utilizada a ferramenta *iperf*. Para verificar qual a largura de banda máxima em ambos os pontos da topologia decidimos realizar o teste *iperf* utilizando o SERVER como servidor e o HOME PC como cliente. Os testes realizados foram feitos com os dados do exercício 1.1 em que a ligação residencial continha uma taxa de transmissão de 512 Kbps tanto no *upstream* como *downstream*. Na figura seguinte estão apresentados os dados obtidos com a transmissão, a ser feita por TCP, onde podemos aferir que a largura de banda média no servidor é de 490 Kbps e no cliente é de 1.15 Mbps.

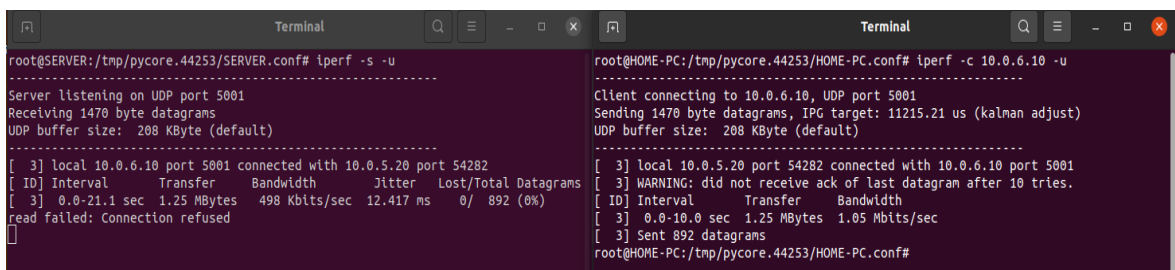


```
root@SERVER:/tmp/pycore.44253/SERVER.conf# iperf -s
Server listening on TCP port 5001
TCP window size: 128 KByte (default)
[ 4] local 10.0.6.10 port 5001 connected with 10.0.5.20 port 44774
[ ID] Interval      Transfer    Bandwidth
[ 4] 0.0-30.0 sec  1.75 MBytes  490 Kbits/sec

root@HOME-PC:/tmp/pycore.44253/HOME-PC.conf# iperf -c 10.0.6.10
Client connecting to 10.0.6.10, TCP port 5001
TCP window size: 85.0 KByte (default)
[ 3] local 10.0.5.20 port 44774 connected with 10.0.6.10 port 5001
[ ID] Interval      Transfer    Bandwidth
[ 3] 0.0-12.7 sec  1.75 MBytes  1.15 Mbits/sec
root@HOME-PC:/tmp/pycore.44253/HOME-PC.conf#
```

Figura 5 - Teste *iperf* com TCP entre HOME-PC e SERVER.

O teste anterior foi feito utilizando o protocolo TCP para a transferência de dados. Na figura seguinte apresentamos o teste realizado com o protocolo UDP, no qual obtivemos valores similares de largura de banda, devido ao facto de não ocorrerem erros na transmissão, o que faz com que a rede não seja substancialmente afetada contrariamente à ligação TCP, que é afetada por erros e duplicação de pacotes.



```
root@SERVER:/tmp/pycore.44253/SERVER.conf# iperf -s -u
Server listening on UDP port 5001
Receiving 1470 byte datagrams
UDP buffer size: 208 KByte (default)
[ 3] local 10.0.6.10 port 5001 connected with 10.0.5.20 port 54282
[ ID] Interval      Transfer    Bandwidth    Jitter    Lost/Tot. Datagrams
[ 3] 0.0-21.1 sec  1.25 MBytes  498 Kbits/sec  12.417 ms  0/ 892 (0%)
read failed: Connection refused

root@HOME-PC:/tmp/pycore.44253/HOME-PC.conf# iperf -c 10.0.6.10 -u
Client connecting to 10.0.6.10, UDP port 5001
Sending 1470 byte datagrams, IPG target: 11215.21 us (kalman adjust)
UDP buffer size: 208 KByte (default)
[ 3] local 10.0.5.20 port 54282 connected with 10.0.6.10 port 5001
[ 3] WARNING: did not receive ack of last datagram after 10 tries.
[ ID] Interval      Transfer    Bandwidth
[ 3] 0.0-10.0 sec  1.25 MBytes  1.05 Mbits/sec
[ 3] Sent 892 datagrams
root@HOME-PC:/tmp/pycore.44253/HOME-PC.conf#
```

Figura 6 - Teste *iperf* com UDP entre HOME-PC e SERVER.

2.2. Delay, Loss e Duplicate

Nesta subsecção de testes foram implementadas várias situações distintas de modo a observar o efeito provocado pela variação de alguns dos parâmetros de comunicação. Em seguida serão apresentadas tabelas com os valores retirados na realização dos testes, de modo a ser mais fácil visualmente a comparação dos resultados.

No primeiro grupo de testes foi utilizado o protocolo TCP com a introdução de perdas e com introdução de duplicação de pacotes.

TCP	BANDWIDTH	
	SERVER	CLIENTE
NO LOSS	490 Kbps	1.15 Mbps
LOSS 4%	452 Kbps	512 Kbps
LOSS 10%	300 Kbps	397 Kbps

Figura 8 - Tabela com os valores de perdas (TCP).

TCP	BANDWIDTH	
	SERVER	CLIENTE
NO DUPLICATE	490 Kbps	1.15 Mbps
DUPLICATE 2%	478 Kbps	1.09 Mbps
DUPLICATE 5%	473 Kbps	978 Kbps

Figura 7 - Tabela com os valores de duplicados (TCP).

Através da observação das tabelas anteriores é possível verificar que os valores obtidos utilizando o protocolo TCP apresentam mais variações na largura de banda quando é introduzida perda do que quando é introduzida duplicação de pacotes.

Em seguida são apresentados os testes utilizando o protocolo UDP de forma a comparar resultados.

UDP	BANDWIDTH	
	SERVER	CLIENTE
NO LOSS	498 Kbps	1.05 Mbps
LOSS 4%	498 Kbps	1.05 Mbps
LOSS 10%	498 Kbps	1.05 Mbps

Figura 9 - Tabela com valores de perdas (UDP).

UDP	BANDWIDTH	
	SERVER	CLIENTE
NO DUPLICATE	498 Kbps	1.05 Mbps
DUPLICATE 2%	498 Kbps	1.05 Mbps
DUPLICATE 5%	498 Kbps	1.05 Mbps

Figura 10 - Tabela com os valores de duplicados (UDP).

Pela análise das tabelas acima é possível concluir que usando o protocolo UDP, os valores obtidos para a largura de banda são sempre os mesmos, o que nos permite concluir que tanto a introdução de perdas como a introdução de duplicação de pacotes não influencia a largura de banda.

Comparando os dois protocolos de comunicação, pode-se constatar que o TCP é o mais afetado com a variação dos parâmetros, isto deve-se ao facto de ser um protocolo orientado à conexão, ao contrário do UDP.

2.3. Combinação dos parâmetros Loss e Duplicate

De seguida são apresentados testes com a combinação dos dois parâmetros (LOSS, DUPLICATE): (4,2), (4,5), (10,2) e (10,5). Abaixo estão apresentados os resultados obtidos usando o protocolo TCP.

TCP	BANDWIDTH	
	SERVER	CLIENTE
NO LOSS	490 Kbps	1.15 Mbps
LOSS 4% / DUPLICATE 2%	433 Kbps	506 Kbps
LOSS 4% / DUPLICATE 5%	471 Kbps	612 Kbps
LOSS 10% / DUPLICATE 2%	327 Kbps	410 Kbps
LOSS 10% / DUPLICATE 5%	283 Kbps	313 Kbps

Figura 11 - Tabela com a combinação dos 2 parâmetros (TCP).

É de salientar que os valores obtidos não são os mais precisos visto que foi utilizada uma versão do core mais antiga, no entanto quando realizado o mesmo teste numa versão mais recente os resultados obtidos eram um pouco diferentes, não sendo possível, deste modo, tirarmos conclusões exatas deste teste. Apesar disso, é possível observar diminuição significativa da largura de banda quando o parâmetro perdas é aumentado, mas quando se faz variar o parâmetro de duplicação de pacotes, a variação não é tão acentuada, havendo casos em que a largura de banda melhora.

Quanto ao protocolo UDP, era de esperar que os valores não variassem muito devido às razões previamente descritas. Com a observação da tabela abaixo, podemos concluir que o que realmente acontece é a não variação dos resultados de largura de banda, que se mantém sempre nos 498 Kbps no lado do server e nos 1.05 Mbps no lado do cliente.

UDP	BANDWIDTH	
	SERVER	CLIENTE
NO LOSS	498 Kbps	1.05 Mbps
LOSS 4% / DUPLICATE 2%	498 Kbps	1.05 Mbps
LOSS 4% / DUPLICATE 5%	498 Kbps	1.05 Mbps
LOSS 10% / DUPLICATE 2%	498 Kbps	1.05 Mbps
LOSS 10% / DUPLICATE 5%	498 Kbps	1.05 Mbps

Figura 12 - Tabela com a combinações dos 2 parâmetros (UDP).

2.4. Alteração do valor do atraso

Quanto ao último teste realizado para a análise de desempenho da rede desenvolvida foi alterado o valor de atraso da ligação residencial da rede de 50 ms para 2 segundos, que é um valor próximo do atraso médio entre a Terra e a Lua. De seguida são apresentados os resultados obtidos utilizando o protocolo TCP.

TCP	BANDWIDTH	
	SERVER	CLIENTE
DELAY 50ms	490 Kbps	1.15 Mbps
DELAY 2000ms	72.8 Kbps	85 Kbps

Figura 13 - Tabela com os valores para um atraso de 50ms e 2s (TCP).

Os resultados obtidos estão dentro do expectável, pois como já foi visto anteriormente, a comunicação por TCP sofre bastante com o atraso, o que faz com que a largura de banda seja relativamente baixa para um atraso de 2 segundos. No protocolo UDP era de esperar obter uns valores melhores a rondar os valores sem atraso e foi precisamente o obtido como demonstrado em seguida.

UDP	BANDWIDTH	
	SERVER	CLIENTE
DELAY 50ms	498 Kbps	1.05 Mbps
DELAY 2000ms	498 Kbps	1.05 Mbps

Figura 14 - Tabela com valores para um atraso de 50ms e 2s (UDP).

2.5. Avaliação das características Best Effort

As redes Best Effort possuem apenas um único requisito que é o de garantir a conectividade entre entidades. A qualidade de comunicação nestas redes é definida pelo congestionamento ou não da rede, ou seja, quanto menos congestionada a rede estiver, melhor é a qualidade. Os recursos são partilhados de forma equitativa entre os diferentes fluxos de tráfego, não havendo mecanismo que permitam priorizar determinados tipos de tráfego em detrimento de outros.

O facto de a largura de banda baixar com o TCP deve-se ao facto de que este protocolo quando se depara com perda de pacotes, interpreta-o como um sinal de que a rede está congestionada e como consequência, diminui a sua taxa de transmissão.

Em relação ao parâmetro de duplicação de pacotes, este não tem um impacto tão grande na largura de banda como o de perdas, pois o TCP quando recebe um pacote duplicado, descarta-o e não assume que a rede possa estar congestionada, ao contrário da perda de pacotes.

Relativamente ao protocolo UDP este não afeta a largura de banda da rede uma vez que não é afetado por perda de pacotes ou duplicação de pacotes.

Por fim, podemos concluir que em certos casos, como quando foi aumentada a perda de pacotes, a duplicação de pacotes até “beneficiou” a rede, porque apesar de existirem perdas, como os pacotes eram duplicados, permitiu uma maior eficiência na comunicação.

Conclusão

Tal como foi proposto no enunciado do trabalho prático, foi realizada a emulação de uma rede de acesso IP, onde foi avaliado o desempenho dessa mesma rede. Essa avaliação de desempenho foi feita através da alteração de taxas de transmissão, introdução de atrasos, aplicação de pacotes duplicados e perda de pacotes.

Com a realização deste trabalho, o grupo foi capaz de concluir que as ligações TCP, por serem orientadas à conexão, são as mais afetadas quanto à largura de banda contrariamente às ligações UDP. Foi também possível aferir que as comunicações TCP são as mais prejudicadas quando introduzidas perdas e duplicação de pacotes, tudo isto pelo motivo acima descrito.

O grupo teve algumas dificuldades na compreensão de alguns dos resultados obtidos, nomeadamente, nos valores de largura de banda quando foram variados os parâmetros de perdas e duplicação de pacote, no entanto, estas foram ultrapassadas com sucesso através de alguma pesquisa e reflexão em grupo.