Trabalho Prático 1 - Grupo 4

João Abreu, *pg55895,*, Luís Vilas, *pg57888,* e Ricardo Pereira, *pg56001*

Index Terms

G.fast, Access Network, Core Network, Redes Assimétrica, Ping, Traceroute, iPerf, Atraso, Banda Larga, Packet Loss, Duplicate, Best Effort.

I. Introdução

Este relatório foi desenvolvido no âmbito da unidade curricular de Redes Fixas e Móveis do Mestrado em Engenharia Informática da Universidade do Minho. Este trabalho prático (TP1) está dividido em duas partes. Na primeira parte, o objetivo é explorar diversas tecnologias para a rede de acesso e construir uma topologia que integre o Access Network e a Core Network, avaliando a sua conectividade e as ligações assimétricas do Acess Network. Na segunda parte, vamos explorar a ferramenta Iperf, onde vamos realizar várias medições com base em diferentes variações do ambiente, como o Loss, Duplicate e a latência, utilizando tanto ligações TCP quanto UDP.

II. Desenvolvimento

A. Tecnologia escolhida para Rede de Acesso

A tecnologia escolhida para o nosso **access link** foi o **G.fast** [1], que foi selecionada por vários motivos, sendo o principal a limitação do **Core Gui**, que suporta velocidades de até **1 Gbps**. Esse limite tecnológico inviabilizou a adoção de tecnologias mais avançadas, como certas implementações de **fibra ótica** (como **XG-PON** e **NG-PON2**) e **coaxiais modernos** (como **DOCSIS 3.1 e 4.0**), que já superam esse limite.

O **G.fast** é uma tecnologia DSL de última geração que opera sobre cabos de cobre em distâncias curtas (geralmente até 250 metros), oferecendo velocidades de **500 Mbps a 1 Gbps**. A velocidade e a latência dessa tecnologia dependem diretamente da **distância**, sendo que quanto maior a distância, maior a atenuação do sinal, o que reduz a velocidade e pode aumentar a latência. No entanto, como nossa topologia não possui restrições significativas de distância, podemos assumir uma velocidade de **800 Mbps** e uma latência de **1 ms** para a implementação.

Distância	Velocidade Máxima	Latência
100 metros	1 Gbps (download + upload)	$<$ 1 ms
150 metros	800 Mbps	1 ms
250 metros	500 Mbps	1-3 ms

TABLE I
Desempenho do G.fast em diferentes distâncias

O **G.fast** oferece a capacidade de **altas velocidades** e **baixas latências** mesmo com a utilização de **cabeamento de cobre**, o que a torna uma solução económica para certos casos de uso.

Uma das principais aplicações do **G.fast** é em arquiteturas **FTTdp (Fibre-to-the-distribution-point)**, onde a fibra ótica chega até um ponto de distribuição, e o trecho final até ao utilizador é feito com **cobre coaxial**. Essa abordagem permite a realização de **upgrades de velocidade** sem a necessidade de substituir toda a infraestrutura de cabos, o que torna o **G.fast** uma solução **económica e prática** para operadoras, especialmente em áreas onde a substituição por fibra ótica seria inviável.

Além disso, o **G.fast** utiliza técnicas avançadas como:

- **Vectoring**: Essa técnica reduz interferências e diafonia entre as linhas vizinhas, garantindo uma maior **estabilidade** e **desempenho** da conexão, mesmo em ambientes com múltiplas linhas ativas.
- **Time-division duplexing (TDD)**: Permite distribuir dinamicamente a largura de banda entre **upload** e **download**. Embora o tópico (1.1) do nosso projeto não requeira uma alocação diferenciada para o **downstream** e **upstream**, o **G.fast** oferece flexibilidade para priorizar o **download**, se necessário, de acordo com as necessidades.
- **Faixas de frequência mais altas (até 212 MHz)**: Essas faixas mais altas permitem a transmissão de dados mais rápidos, superando as limitações de tecnologias **DSL anteriores**, como o **VDSL2**. Além disso, essas frequências ajudam a **reduzir a latência**, permitindo uma **transmissão de dados mais rápida e eficiente**.

Em Portugal, ISPs como a **NOS** ainda utilizam cabos coaxiais, especialmente em regiões não urbanas, onde a substituição total por fibra ótica seria muito dispendiosa. Por esse motivo, o nosso objetivo foi escolher uma tecnologia que respeitasse as restrições impostas, ao mesmo tempo que tivesse aplicação prática no mundo real.

B. Desenvolvimento da Topologia

A topologia desenvolvida é caracterizada por ser constituída por dois tipos de rede: **a rede de Acesso** e **a rede de Núcleo**.

A **rede de Núcleo** representada no círculo a amarelo na Figura 1 é o backbone da rede, e é a responsável por gerir o tráfego de dados entre a/s rede/s de acesso e o/s servidor/es. Na nossa Topologia, esta rede tem apenas um único servidor (SERVER), e cinco routers (R1 ao R5). O objetivo desta rede é garantir o encaminhamento eficiente do tráfego, com uma elevada qualidade de entrega, o que se reflete na baixa latência e na ampla largura de banda disponível.

A **rede de Acesso** tem como função principal estabelecer a ligação entre os dispositivos finais, como o *HOME-PC*, e a rede de Núcleo. Para simular a rede de Acesso com a tecnologia G.fast no CORE (representada na Figura 1 a verde), optámos por adotar a topologia proposta no enunciado, introduzindo limites na Access Network relativos às características do G.fast. Dessa forma, limitamos a ligação entre o HOME-ROUTER e a Core Network a uma banda larga (T) de 800000000 bps, com um atraso (D) de 1000 us, atingindo os valores descritos no tópico anterior.

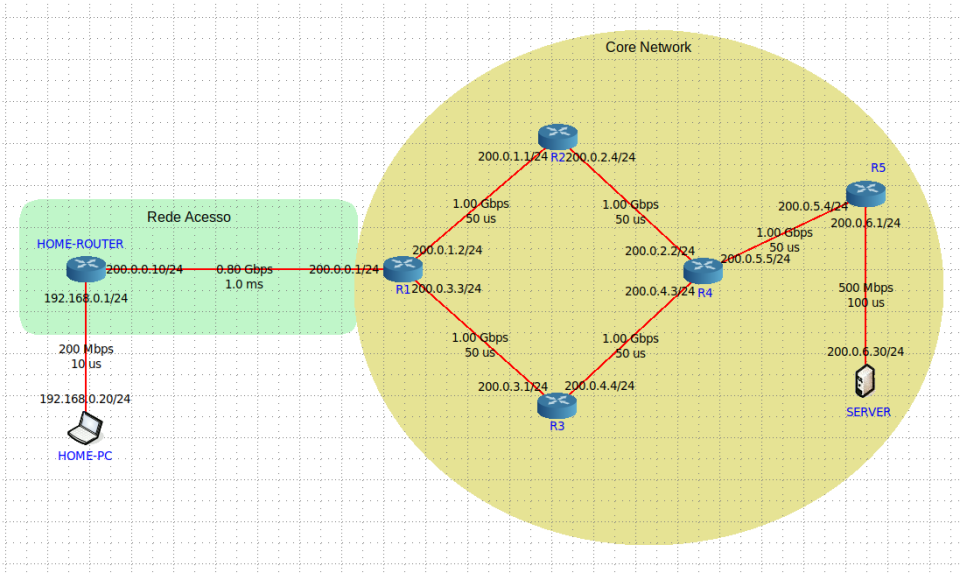

Fig. 1. Topologia com a tecnologia G.fast aplicada

C. Testes de Conectividade e Comparação

1) Testes práticos: Para os testes de conectividade, utilizámos dois comandos principais: o `ping` e o `traceroute`.

No caso do `ping`, definimos dois tamanhos de pacote a enviar: um de 64 bytes (`56` bytes de payload `+ 8` bytes de header) e outro de 1028 bytes (`1020` bytes de payload `+ 8` bytes de header). Enviámos cerca de 20 pacotes para cada teste. A estrutura do comando `ping` utilizada foi a seguinte:

```
ping <Ip-destino> -s <tamanho> -c 20
```

Além disso, utilizámos o comando `traceroute` para observar o caminho seguido até ao destino. A sua configuração foi a seguinte:

```
traceroute <Ip-destino>
```

Realizámos os testes no sentido `Home-PC` para o `Server` assim como no sentido inverso.

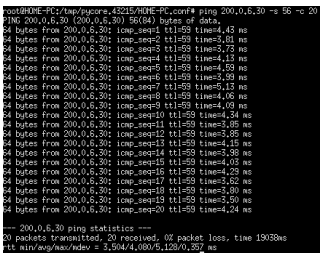

Fig. 2. Ping do HOME-PC para o SERVER (64 bytes)

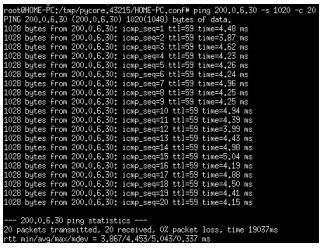

Fig. 3. Ping do HOME-PC para o SERVER (1028 bytes)

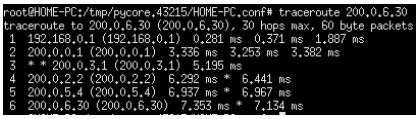

Fig. 4. Traceroute: HOME-PC até o SERVER

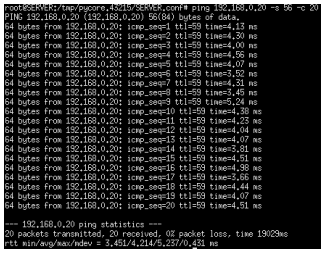

Fig. 5. Ping do SERVER para o HOME-PC (64 bytes)

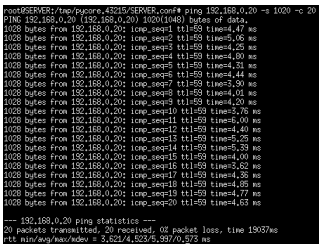

Fig. 6. Ping do SERVER para o HOME-PC (1028 bytes)

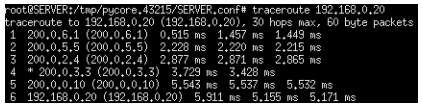

Fig. 7. Traceroute: SERVER até o HOME-PC

O atraso médio (tempo médio de ida e volta) corresponde ao valor **avg** do comando `ping`. A Tabela II apresenta os tempos médios medidos para diferentes tamanhos de pacotes e direções de comunicação.

TABLE II
TEMPOS MÉDIOS DE ATRASO (PING) ENTRE HOME-PC E SERVER

Sentido	Tamanho do Pacote (bytes)	Atraso Médio (ms)
Home-PC → Server	64	4.080
Home-PC → Server	1028	4.353
Server → Home-PC	64	4.214
Server → Home-PC	1028	4.523

As Figuras 4 e 7 apresentam os tempos em cada salto da comunicação, respetivamente no sentido **Home-PC → Server** e **Server → Home-PC**. Nota-se uma pequena variação nos tempos médios, o que pode ser atribuído ao encaminhamento dinâmico dos pacotes que é feito automaticamente pelo OSPF.

Além disso, pode haver diferenças entre os valores obtidos pelo `ping` e pelo `traceroute`, pois estas ferramentas utilizam protocolos distintos:

- O `ping` envia **pacotes ICMP Echo Request**, medindo o tempo total de ida e volta até o destino.
- O `traceroute` determina os tempos em **cada salto individual**, geralmente utilizando **pacotes UDP**, que podem ser tratados de forma diferente pelos routers.

2) Valor Teórico: No cálculo do valor teórico, vamos ignorar o Atraso de Processamento e o Atraso de Enfileiramento, uma vez que não temos como medir esses valores com precisão nem como os estimar. Portanto, a nossa equação para o cálculo do valor teórico de atraso fica:

$$\text{Atraso Total} = 2 \times \left(\sum_{i=1}^{n} \frac{\text{Tamanho Do Pacote (bits)}}{\text{Banda Larga}_i\text{(bps)}} + \sum_{i=1}^{n} \text{Atraso de Propagação}_i \right) \tag{1}$$

A fórmula calcula o Atraso Total de ida e volta de um pacote na rede, levando em consideração os atrasos de transmissão e atrasos de propagação em cada segmento do caminho. A banda larga e o atraso de propagação podem variar ao longo do percurso, e o cálculo é feito para cada segmento, somando os atrasos de transmissão e propagação e multiplicando por 2 para considerar o percurso de ida e volta.

Assumindo o caminho do pacote de 1028 bytes: **HOME-PC → HOME-ROUTER → R1 → R2 → R4 → R5 → SERVER** na topologia representada na Figura 1.

Substituindo na Formula pelos valores da Topologia:

$$\text{Atraso Total} = 2\times \left[\left(\frac{1028}{200\,000\,000} + \frac{1028}{800\,000\,000} + 3 \cdot \frac{1028}{1\,000\,000\,000} + \frac{1028}{500\,000\,000} \right) + (10 + 1000 + 50 + 50 + 50 + 100) \right] \tag{2}$$

$$\text{Atraso Total} \approx 2 \times 1271.565\,\mu\text{s} \approx 2543.13\,\mu\text{s} \approx 2.544\,\text{ms}$$

Portanto, o atraso total calculado é aproximadamente $2544\,\mu$s (ou $2.544\,$ms). Este valor é menor do que os valores práticos porque não conta com outros fatores que podem aparecer na rede, como outros atrasos. Além disso, o modelo teórico assume que a largura de banda e o atraso de propagação são constantes ao longo de todo o caminho, enquanto na prática estes parâmetros podem variar.

D. Rede de acesso assimétrica

No mundo real, as redes de acesso frequentemente não possuem a mesma largura de banda para os dois sentidos de transmissão. Isso ocorre porque se assume que os clientes estão mais interessados em fazer *download* de conteúdo do que em fazer *upload*. Em redes baseadas em cabos coaxiais, esta prática é particularmente comum. No entanto, em tecnologias mais modernas, a largura de banda para *download* e *upload* tende a ser igual, oferecendo um equilíbrio maior entre os dois sentidos de comunicação. No tópico 1.2, será implementada uma alteração na rede de acesso, configurando uma limitação assimétrica da largura de banda, com *downstream* de 1 Gbps e *upstream* de 256 kbps demonstrada na Figura 8.

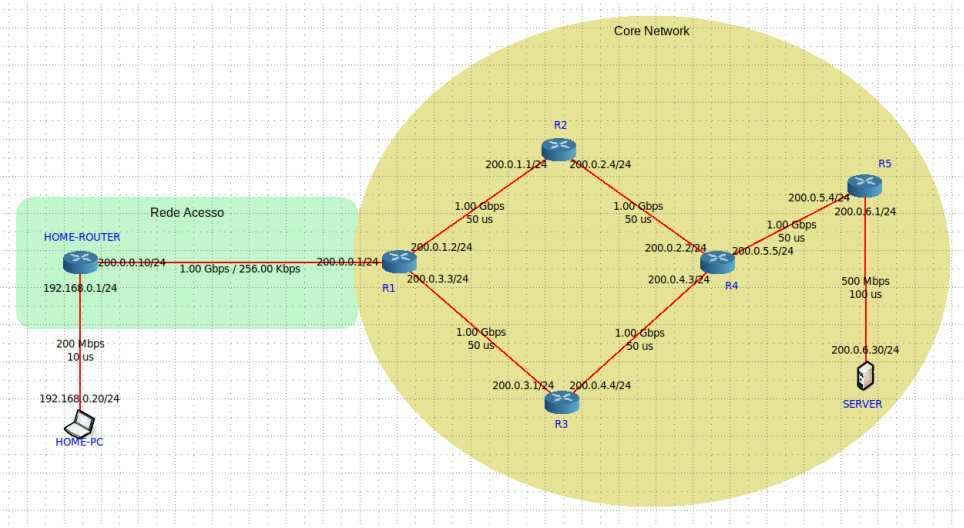

Fig. 8. Topologia com Rede de Acesso assimétrica

Para testar a conexão e a nova configuração, utilizamos os comandos: ping (20 pacotes de 1028 bytes de tamanho), traceroute e o Iperf (será mais elaborado na secção seguinte)

Fig. 9. Ping do HOME-PC para o SERVER

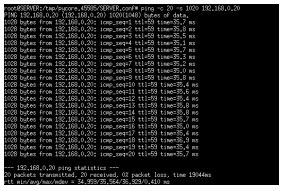

Fig. 10. Ping do SERVER para o HOME-PC

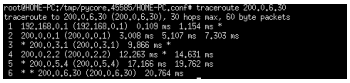

Fig. 11. Traceroute: SERVER até o HOME-PC

Fig. 12. Traceroute: HOME-PC até o SERVER

Como os pings medem o RTT, não são um bom indicador para avaliar separadamente a capacidade *downstream* e *upstream*, uma vez que o protocolo percorre ambos os sentidos. Por esse motivo, os valores de atraso médio são muito semelhantes (35.429 ms versus 35.564 ms), conforme observado nas Figuras 9 e 10. No entanto, é possível notar a influência do upstream reduzido de 256 kbps, que aumenta a latência em comparação com os valores apresentados na Figura 3 e 6 da topologia original.

O comando `traceroute` (Figuras 11 e 12) pode apresentar alguns problemas, como indicado pelos asteriscos (`**`), o que sugere que, ao atravessar a rede de acesso *upstream*, podem ocorrer falhas no envio dos pacotes. Esse comportamento é comum em situações de congestionamento ou lentidão em determinados pontos da rede, pois o `traceroute` envia, por padrão, pacotes UDP sem garantia de entrega.

No contexto da rede de acesso *upstream*, esse tipo de problema pode ser ainda mais relevante, uma vez que a largura de banda do link de *upload* (*upstream*) é tipicamente inferior à do *download*, o que pode levar a um aumento da latência e, consequentemente, a uma maior taxa de perda de pacotes.

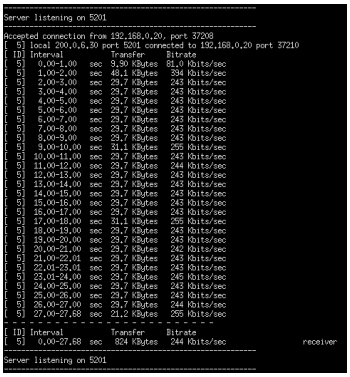

Fig. 13. IPerf: SERVER como Servidor TCP

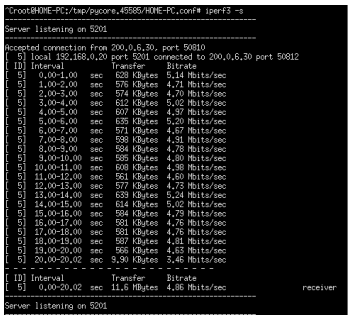

Fig. 14. IPerf: HOME-PC como Servidor TCP

O `Iperf` revela-se uma ferramenta extremamente útil, pois permite observar, nas Figuras 13 e 14, a diferença de largura de banda disponível em cada sentido da rede de acesso. Na Figura 13, o *Server* atua como servidor e o *Home-PC* como cliente. Dado que a taxa de *upload* (*upstream*) é significativamente reduzida, o *bitrate* registado pelo servidor será também muito baixo (244 Kbits/sec). O contrário verifica-se na Figura 14, onde a elevada largura de banda do *download* (*downstream*) permite um desempenho muito superior (4.86 Mbits/sec).

E. IPerf

O iPerf é uma ferramenta amplamente utilizada para medir o desempenho de redes, permitindo testar a largura de banda entre dois dispositivos em uma rede. Ela é usada para gerar tráfego de rede TCP ou UDP e medir a taxa de transferência, latência e perda de pacotes entre o cliente e o servidor.

Optámos por utilizar o iPerf3, porque esta versão, além de outras várias vantagens fornece uma maior precisão na medição da latência e do jitter, em especial nos testes UDP.

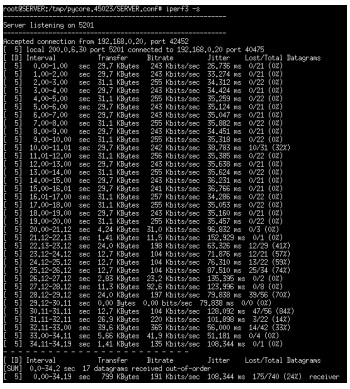

Fig. 15. IPerf: SERVER como Servidor UDP

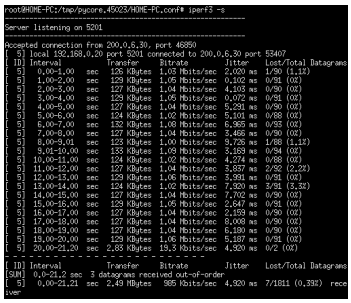

Fig. 16. IPerf: HOME-PC como Servidor UDP

Repetimos as medições mas desta vez com recurso a conexões UDP entre o *Home-PC* e o *Server*. Os resultados foram semelhantes em termos de taxa de transferência, sendo esta bastante mais elevada para *download* (985 Kbits/sec) do que para *upload* (191 Kbits/sec), tal como seria de esperar. Ainda assim, notou-se uma perda considerável de desempenho comparadamente ao protocolo TCP, o que pode ser explicado pelas características de ambos os protocolos. No caso da Figura 16, podemos ainda verificar que a perda de pacotes foi consideravelmente alta (24%), uma vez que o UDP não tem mecanismos que ajustem a taxa de envio consoante as condições de rede. O cliente enviava assim pacotes a uma taxa maior do que a que a ligação de acesso é capaz de suportar, fazendo com que muitos se perdessem aí.

F. Configuração do link de Acesso

Nesta subsecção, apresentamos diversos testes registados para cada alteração efetuada nos seguintes parâmetros:

- **Loss**: Percentagem de pacotes perdidos durante a transmissão, podendo ser causada por congestionamento na rede, interferências ou outros fatores aleatórios. Testamos os seguintes valores: 4%, 7% e 10% de Loss
- **Duplicate**: Percentagem de pacotes que, de forma aleatória, são duplicados na rede, resultando possivelmente de problemas de retransmissão ou falhas na comunicação. Testamos os seguintes valores: 2% e 5% de Duplicate

A Topologia utilizada para os testes é exatamente igual à representada na Figura 8, com a alteração dos parâmetros necessários. Para otimizar a quantidade de testes realizados, focaremos no Home-PC como servidor e no Server como emissor, explorando o sentido downstream para aproveitar a maior largura de banda.

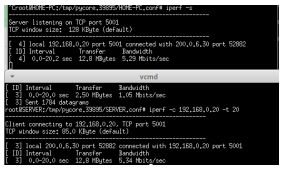

Fig. 17. Duplicate 2% - TCP

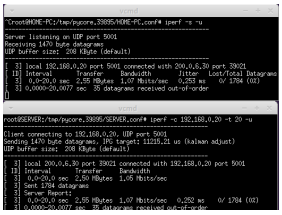

Fig. 18. Duplicate 2% - UDP

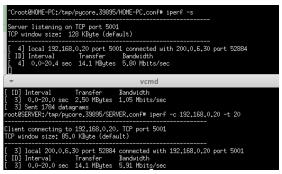

Fig. 19. Duplicate 5% - TCP

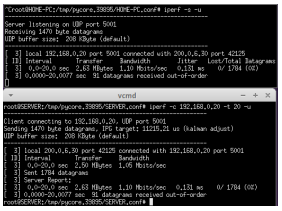

Fig. 20. Duplicate 5% - UDP

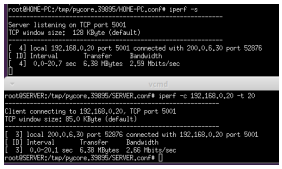

Fig. 21. Loss 4% - TCP

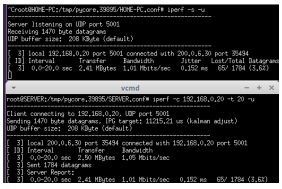

Fig. 22. Loss 4% - UDP

Fig. 23. Loss 7% - TCP

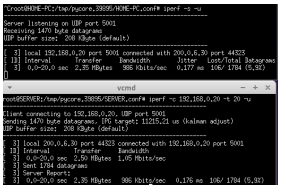

Fig. 24. Loss 7% - UDP

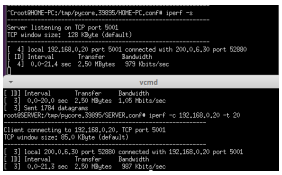

Fig. 25. Loss 10% - TCP

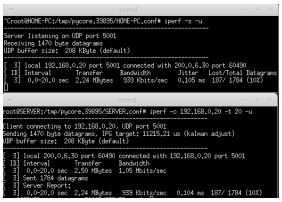

Fig. 26. Loss 10% - UDP

TABLE III
RESULTADOS DE TRANSFERÊNCIA E QUALIDADE DA REDE DO TÓPICO 3.1

Teste	Interval (sec)	Transfer (MBytes)	Bandwidth (MBytes/sec)	Jitter (ms)	Lost/Total Datagrams	Datagrams Out-of-Order
Duplicate 2% TCP	0.0-20.0	12.8	5.34	-	-	-
Duplicate 2% UDP	0.0-20.0	2.55	1.07	0.252	0/1784 (0%)	35
Duplicate 5% TCP	0.0-20.0	14.1	5.91	-	-	-
Duplicate 5% UDP	0.0-20.0	2.63	1.10	0.131	0/1784 (0%)	91
Loss 4% TCP	0.0-20.1	6.38	2.66	-	-	-
Loss 4% UDP	0.0-20.0	2.41	1.01	0.152	65/1784 (3,6%)	0
Loss 7% TCP	0.0-20.7	3.88	1.57	-	-	-
Loss 7% UDP	0.0-20.0	2.35	0.986	0.176	106/1784 (5,9%)	0
Loss 10% TCP	0.0-21.3	2.50	0.987	-	-	-
Loss 10% UDP	0.0-20.0	2.24	0.939	0.104	187/1784 (10%)	0

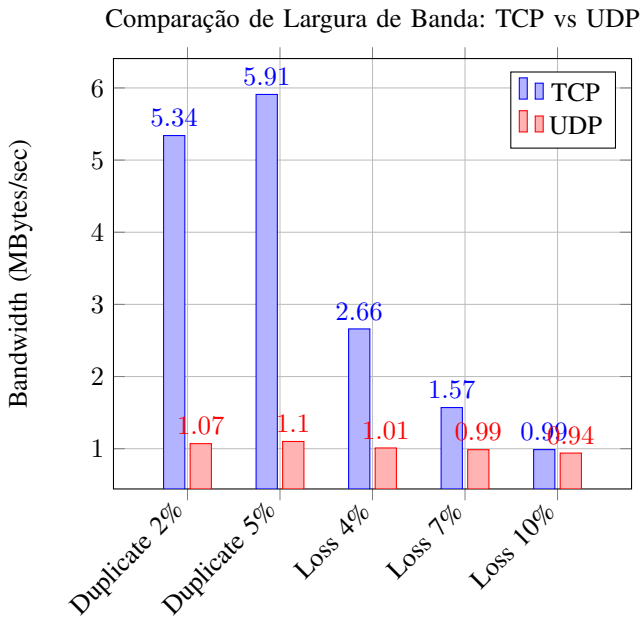

Fig. 27. Largura de banda comparada entre TCP e UDP em diferentes cenários.

1) Modificação de um único parâmetro (Loss ou Duplicate) - Tópico 3.1: Nos testes realizados, a diferença fundamental entre os protocolos TCP e UDP reflete-se na forma como lidam com a perda de pacotes e a sua recuperação.

- **TCP**: Como um protocolo fiável, o TCP possui mecanismos de retransmissão de pacotes perdidos, o que faz com que o tempo total de transmissão seja maior em cenários com perdas de pacotes. Isto ocorre porque, ao detetar a perda, o TCP realiza o reenvio dos pacotes, aumentando a duração do teste para além dos 20 segundos especificados. Por essa razão, não se observa perda de pacotes ou pacotes fora de ordem nos testes TCP, uma vez que o protocolo garante a entrega ordenada dos pacotes. No caso do *Duplicate*, o TCP não é significativamente afetado, uma vez que os seus mecanismos garantem a reordenação e a entrega correta dos pacotes. No entanto, nos testes com *Loss*, o TCP sofre um impacto muito maior, pois precisa de retransmitir os pacotes perdidos, reduzindo drasticamente a largura de banda disponível à medida que a taxa de perda aumenta como é possível visualizar na Figura 36.
- **UDP**: Ao contrário do TCP, o UDP não realiza retransmissões, o que significa que qualquer pacote perdido não será recuperado, resultando numa perda direta de pacotes. No entanto, o tempo de teste permanece em 20 segundos, como esperado, uma vez que não há atrasos causados por retransmissões. A **perda de pacotes** no UDP está em conformidade com os parâmetros de teste definidos (por exemplo, perdas de 4%, 7% e 10%), e o **jitter** (variação do atraso) é registado em alguns casos, devido à natureza não fiável do protocolo.
  Nos testes com *Duplicate*, observa-se que o UDP cria *datagramas fora de ordem*, pois os pacotes podem ser recebidos numa ordem diferente da que foram enviados. Isto ocorre porque o UDP não implementa mecanismos de reordenação, ao contrário do TCP, que assegura que os pacotes chegam na sequência correta. No entanto, a largura de banda do UDP mantém-se relativamente constante, tanto nos testes de *Duplicate* como nos de *Loss*, uma vez que este protocolo não retransmite pacotes e simplesmente aceita as perdas como exibido na Figura 36.
  Portanto, o UDP mantém um desempenho previsível independentemente da perda, e o TCP sofre um impacto severo em cenários de perdas elevadas.

2) Modificação de dois Parâmetros: Esta subsubsecção, representa o tópico 3.2, no qual vamos realizar uma série de testes, onde atribuímos uma série de combinações de Parametros Loss e Duplicate((4,2), (4,5), (10,2), (10,5)).

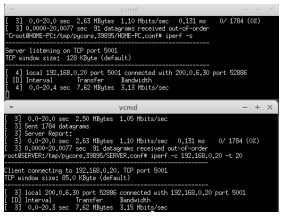

Fig. 28. (4.2) TCP

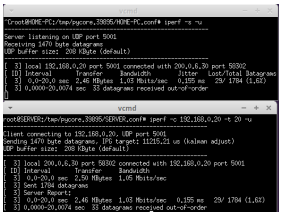

Fig. 29. (4.2) UDP

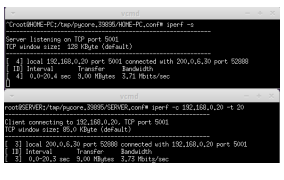

Fig. 30. (4.5) TCP

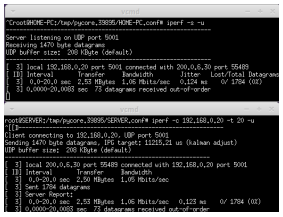

Fig. 31. (4.5) UDP

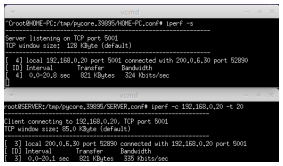

Fig. 32. (10.2) TCP

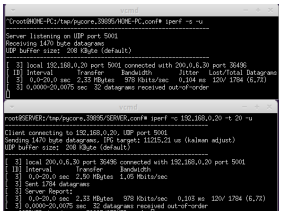

Fig. 33. (10.2) UDP

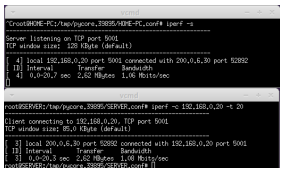

Fig. 34. (10.5) TCP

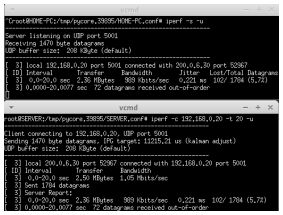

Fig. 35. (10.5) UDP

TABLE IV
RESULTADOS DE TRANSFERÊNCIA E QUALIDADE DA REDE DO TÓPICO 3.2

Teste (Loss, Duplicate)	Interval (sec)	Transfer (MBytes)	Bandwidth (MBytes/sec)	Jitter (ms)	Lost/Total Datagrams	Ooo Datagrams
(4,2) TCP	0.0-20.3	7.62	3.15	-	-	-
(4,2) UDP	0.0-20.0	2.46	1.03	0.155	29/1784 (1,6%)	33
(4,5) TCP	0.0-20.3	9.00	3.73	-	-	-
(4,5) UDP	0.0-20.0	2.53	1.06	0.123	0/1784 (0%)	73
(10,2) TCP	0.0-20.1	0.821	0.335	-	-	-
(10,2) UDP	0.0-20.0	2.33	0.978	0.103	120/1784 (6,7%)	32
(10,5) TCP	0.0-20.3	2.62	1.08	-	-	-
(10,5) UDP	0.0-20.0	2.36	0.989	0.221	102/1784 (5,7%)	72

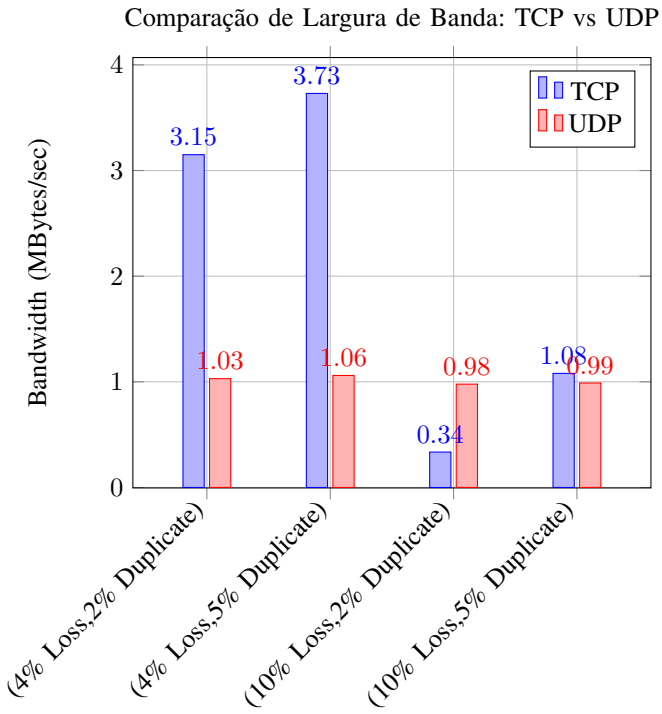

Fig. 36. Largura de banda comparada entre TCP e UDP para diferentes configurações de Loss e Duplicate.

Como discutido no tópico anterior, a perda de pacotes (*Loss*) está associada a uma redução da largura de banda no TCP, enquanto no UDP a taxa de transmissão permanece constante. Um aspecto interessante a destacar é que, no caso do cenário (4,5) UDP, não houve registo de perda de pacotes, o que pode sugerir que todos os pacotes inicialmente perdidos foram, na verdade, duplicados. Essa duplicação de pacotes parece favorecer o TCP, uma vez que ordenar pacotes duplicados pode ser mais eficiente do que retransmitir pacotes perdidos. Além disso, uma maior taxa de duplicação de pacotes (*Duplicate*) e de perda (*Loss*) no UDP pode indicar um aumento na entrega de pacotes fora de ordem.

No geral, as demais observações seguem um padrão semelhante ao descrito anteriormente, incluindo o impacto no tempo de entrega, as diferenças na largura de banda e outros aspetos relevantes.

3) Modificação do parâmetro de Atraso: Nesta subsubsecção, representa o tópico 3.3, onde o parâmetro alterado foi o atraso para 2 segundos, enquanto que as larguras de banda e atrasos das ligações são as representadas na Figura 8.

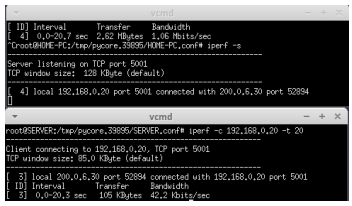

Fig. 37. 3.3) usTCP

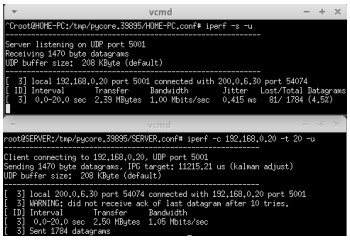

Fig. 38. 3.3) usUDP

TABLE V
RESULTADOS DE TRANSFERÊNCIA E QUALIDADE DA REDE

Teste (Loss, Duplicate)	Interval (sec)	Transfer (MBytes)	Bandwidth (MBytes/sec)	Jitter (ms)	Lost/Total Datagrams	Datagrams Out-of-Order
usec=2 TCP	0.0-20.3	0.105	0.042	-	-	-
usec=2 UDP	0.0-20.0	2.39	1.00	0.415	81/1784 (4,5%)	0

Largura de Banda: No TCP é extremamente baixa (0.042 MBytes/sec) quando comparada com o UDP (1.00 MBytes/sec). Esta diferença deve-se ao facto de o TCP implementar mecanismos de controlo de fluxo. O elevado atraso de 2 segundos introduzido faz com que o tempo de ida e volta (RTT) aumente significativamente, reduzindo a taxa de envio de pacotes. Além disso, o TCP opera com um mecanismo de *acknowledgment*, onde cada segmento enviado precisa de ser reconhecido pelo destinatário antes que novos dados possam ser transmitidos.

Já, por outro lado, o UDP não sofre deste problema, uma vez que não implementa confirmações de entrega. Como resultado, mantém uma largura de banda muito superior.

Packet Loss: Embora a configuração do parâmetro de *Loss* seja teoricamente zero, observa-se que o UDP apresenta uma perda de pacotes de 4,5% (81 pacotes perdidos em 1784). Este fenómeno pode ser explicado pela introdução do atraso de 2 segundos, que pode causar pacotes a chegarem fora do intervalo de tempo esperado para a receção. Como o UDP não implementa retransmissão, qualquer pacote que se perca não será recuperado, refletindo-se nos resultados obtidos na métrica de perda de pacotes.

Redes Best Effort

As redes *Best Effort* (BE) não fornecem garantias explícitas de qualidade de serviço (QoS), tratando todos os pacotes de forma indiferenciada e sem priorização. Neste contexto, a variação de parâmetros como *delay*, *bandwidth*, *packet loss* e *packet duplication* afetam significativamente o desempenho das comunicações na Internet.

1) Efeito do Delay: O *delay* refere-se ao tempo necessário para que um pacote seja transmitido de um emissor para um recetor. Nos testes apresentados no Tópico 3, observou-se que uma latência elevada pode limitar severamente a largura de banda nos protocolos TCP e gerar perdas de pacotes, bem como provocar um jitter elevado nas transmissões UDP. Assim, em redes de *Best Effort* (BE), um aumento do *delay* pode degradar significativamente o QoS e comprometer o desempenho das aplicações interativas, como chamadas VoIP e jogos online, devido ao aumento do tempo de resposta.

2) Efeito da Largura de Banda (Bandwidth): A largura de banda disponível define a quantidade de dados que podem ser transmitidos por unidade de tempo. Nos testes realizados, verificou-se que o desempenho do TCP e do UDP é altamente influenciado por esta limitação. O TCP ajusta dinamicamente a sua taxa de envio com base nas condições da rede, enquanto o UDP mantém uma taxa constante, podendo sofrer perdas se a largura de banda for insuficiente. Quando a largura de banda é reduzida, ambos os protocolos veem a sua taxa de transmissão comprometida, mas o impacto é mais severo para o TCP devido ao aumento do tempo necessário para completar a transmissão, causado pelas retransmissões.

3) Efeito da Perda de Pacotes (Packet Loss): A perda de pacotes ocorre devido a congestionamento ou problemas de rede, afetando de forma diferente os protocolos TCP e UDP. Nos testes realizados para o tópico 3.1 e 3.2 , observou-se que o TCP reage à perda de pacotes com retransmissões, o que reduz drasticamente a largura de banda disponível à medida que a taxa de perda aumenta. Já o UDP, por não retransmitir pacotes, mantém a sua taxa de transmissão constante, mas com uma degradação perceptível na qualidade dos dados recebidos, dependendo da aplicação.

4) Efeito da Duplicação de Pacotes (Packet Duplication): A duplicação de pacotes ocorre quando um mesmo pacote é recebido mais de uma vez, o que pode ser criado por retransmissões desnecessárias, falhas na rede, ou outros fatores. Nos testes, verificou-se que o TCP não é significativamente afetado por este fenómeno, pois implementa mecanismos que identificam e descartam pacotes duplicados. Por outro lado, o UDP pode ser mais impactado, uma vez que não garante a ordem de entrega dos pacotes. Esse comportamento pode comprometer a fluidez da transmissão de certos conteúdos, levando a uma degradação na qualidade da comunicação, especialmente em aplicações sensíveis ao tempo, como áudio e vídeo em tempo real.

III. Conclusões

Concluímos este relatório com um apanhado dos pontos mais relevantes e das conclusões a que chegámos durante as diversas fases de investigação e testes práticos. Num primeiro momento, realizou-se uma pesquisa de vertente mais académica que permitiu identificar possíveis tecnologias adequadas para a Rede de Acesso. A escolha recaiu sobre a tecnologia G.fast, dada a sua capacidade de fornecer uma largura de banda que se enquadra nas nossas restrições e pela sua implementação atual.

Na fase prática, definimos uma topologia conforme descrita no enunciado e testámos os comandos `ping` e `traceroute`, de forma a verificar a conectividade da rede e aprofundar o conhecimento sobre o seu funcionamento.

Posteriormente, procedeu-se à configuração de uma rede assimétrica, com o objetivo de explorar as diferenças entre *upstream* e *downstream*. Nessa etapa, utilizámos o `ping`, `traceroute` e `Iperf` para realizar testes em ambos os sentidos, definindo diferentes limites para cada sentido de dados. Estes testes revelaram que, especialmente no sentido *upstream* e usando ligações UDP, podem ocorrer falhas no envio dos pacotes, devido à baixa banda larga definida para a rede de acesso, precedida de uma ligação com uma capacidade maior (Home-PC para Home-Router).

Segui-se uma fase de experimentação, no Tópico 3, em que alterámos diversos parâmetros da rede, nomeadamente a latência, a taxa de perda (*Loss* – testámos valores de 4% e 10%) e a duplicação de pacotes (*Duplicate* – com valores de 2% e 5%). Esses testes permitiram avaliar de forma aprofundada o impacto desses parâmetros no desempenho dos protocolos TCP e UDP.

Observámos que o TCP beneficia dos seus mecanismos de controlo, como a ordenação e a retransmissão de pacotes, o que lhe permite compensar, em certa medida, os efeitos adversos causados por perdas ou duplicações. Em contraste, o UDP, por não garantir a entrega ordenada dos pacotes, mostrou-se mais suscetível a problemas relacionados com a entrega fora de ordem, traduzindo-se num desempenho menos previsível, especialmente em cenários com elevados índices de *Loss* e *Duplicate*.

De forma geral, conseguimos observar as limitações da tecnologias TCP e UDP perante diversas alterações em parâmetros inerentes a uma rede real, à largura de banda assimétrica, que aliadas aos efeitos da latência elevada, jitter e duplicação de pacotes, podem comprometer significativamente a qualidade do serviço, sobretudo em aplicações interativas e sensíveis à latência, onde qualquer variação nos parâmetros de rede pode afetar a experiência do utilizador.

References

[1] Ram Krishna, DDG (FA), Sidh Kumar, Dir (FA), D. L. Mense, ADG (FA-I) & Avadhesh Singh, ADG (FA-II), *Study Paper on G.fast*. sen.news, 2016. Disponível em: https://sen.news/wp-content/uploads/2024/03/study-paper-gfast.pdf