



Universidade do Minho
Departamento de Informática

Engenharia de Serviços em Rede

Ano Letivo 2024/2025

Trabalho Prático Nº1

Grupo 2

Lucas Oliveira - PG57886

Rafael Gomes - PG56000

Rodrigo Casal Novo - PG56006

Índice

1. Introdução	3
2. Etapa 1 - Streaming HTTP simples sem adaptação dinâmica de débito	3
2.1. Questão 1	3
2.1.1. Identifique a taxa em bps necessária	3
2.1.2. Alínea a - Comente os protocolos utilizados na transferência, bem como a experiência que o utilizador terá caso o link utilizado tenha perdas.	4
2.1.3. Alínea b - Identifique o número total de fluxos gerados e elabore um gráfico que demonstre a evolução do débito dependendo do número de clientes	5
2.1.4. Alínea c - Comente a escalabilidade da solução para 1000 utilizadores, assim como 10000 utilizadores. Crie uma expressão matemática que expresse o débito necessário para que o servidor envie vídeo para N clientes	6
3. Etapa 2 - Streaming adaptativo sobre HTTP (MPEG-DASH)	6
3.1. Questão 2 - Utilize o wireshark para determinar a largura de banda necessária, em bits por segundo, para que o cliente de streaming consiga receber o vídeo no firefox e qual a pilha protocolar usada neste cenário. Explique como obteve esta informação.	6
3.2. Questão 3 - Compare a largura de banda medida na questão anterior com a que é disponibilizada pelo ffmpeg. Qual é a razão para a diferença entre as duas?	7
3.3. Questão 4 - Ajuste o débito dos links da topologia de modo que o cliente no portátil Bela exiba o vídeo de menor resolução e o cliente no portátil Alladin exiba o vídeo com mais resolução. Mostre evidências e justifique a largura de banda necessária para que o stream de vídeo sofra alterações.	8
3.4. Questão 5 - Descreva o funcionamento do DASH neste caso concreto, referindo o papel do ficheiro MPD criado e comparando o modelo de streaming com o que foi utilizado na Questão 1	9
4. Etapa 3 - Streaming RTP/RTCP unicast sobre UDP e multicast com anúncios SAP	10
4.1. Questão 6 - Compare o cenário unicast aplicado com o cenário multicast. Mostre vantagens e desvantagens na solução multicast ao nível da rede, no que diz respeito a escalabilidade (aumento do nº de clientes) e tráfego na rede. Tire as suas conclusões também para os cenários de 1000 e 10000 clientes.	10
4.1.1. Unicast	10
4.1.2. Multicast	10
5. Conclusão	12

1. Introdução

2. Etapa 1 - Streaming HTTP simples sem adaptação dinâmica de débito

Nesta etapa pretendeu-se testar o *streaming* sobre *HTTP*, para isso foi utilizado a topologia da Figura 1.

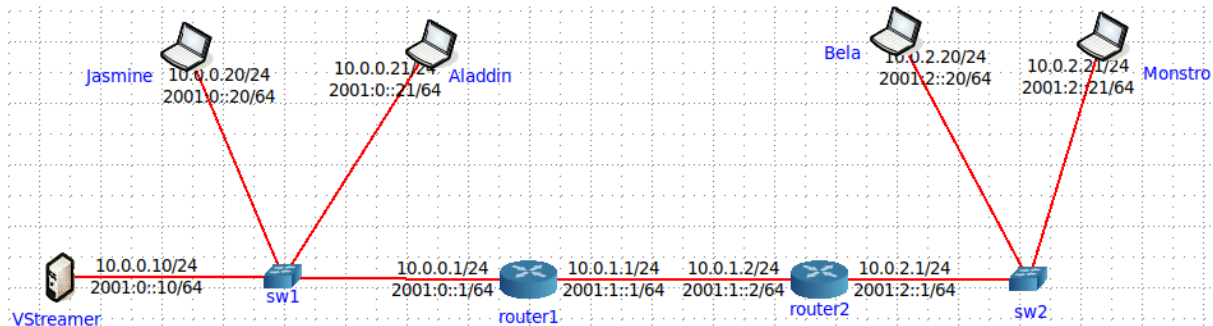


Figura 1: Topologia Utilizada.

Para garantir a correta conectividade da rede, foi utilizado o comando *ping* para verificar a recepção de pacotes entre o computador **Jasmine** e o computador **Monstro**, conforme demonstrado na Figura 2.

```
root@Jasmine:/tmp/pycore.32993/Jasmine.conf# ping 10.0.2.21
PING 10.0.2.21 (10.0.2.21) 56(84) bytes of data:
64 bytes from 10.0.2.21: icmp_seq=1 ttl=62 time=0.128 ms
64 bytes from 10.0.2.21: icmp_seq=2 ttl=62 time=0.136 ms
64 bytes from 10.0.2.21: icmp_seq=3 ttl=62 time=0.135 ms
64 bytes from 10.0.2.21: icmp_seq=4 ttl=62 time=0.051 ms
64 bytes from 10.0.2.21: icmp_seq=5 ttl=62 time=0.058 ms
64 bytes from 10.0.2.21: icmp_seq=6 ttl=62 time=0.049 ms
```

Figura 2: Comando ping entre a Jasmine e o Monstro.

Após esses passos, foram realizadas as configurações necessárias tanto no servidor **VStreamer** como no portátil **Jasmine** para realizar o *streaming* por *HTTP* do vídeo.

2.1. Questão 1

Nota: Devido a problemas associados com o *Firefox*, foi substituído pelo *streaming* de outro *ffplay*.

2.1.1. Identifique a taxa em bps necessária

Para identificar a taxa em *bps* necessária para o *streaming* do vídeo, foi utilizado o comando **ffmpeg -i videoA.mp4**. Conforme ilustrado na Figura 3, o *bitrate* do vídeo é de **15 kb/s (15.000 bps)**. Assim, com base nesta informação, a taxa de transmissão necessária para 2 clientes seria de **30 kb/s (30.000 bps)**, e para 3 clientes, **45 kb/s (45.000 bps)**.

```
Input #0, mov,mp4,m4a,3gp,3g2,mj2, from 'videoA.mp4':
Metadata:
  major_brand      : isom
  minor_version    : 512
  compatible_brands: isomiso2avc1mp41
  encoder         : Lavf58.29.100
Duration: 00:00:16.05, start: 0.000000, bitrate: 17 kb/s
Stream #0:0(und): Video: h264 (High) (avc1 / 0x31637661), yuv420p, 300x226, 15 kb/s, 20 fps, 20 tbr, 10240 tbn, 40 tbc (default)
Metadata:
  handler_name     : VideoHandler
At least one output file must be specified
```

Figura 3: Captura das informações do videoA.

Em seguida, procedeu-se à análise do tráfego gerado, tanto de 2 clientes como para 3 clientes, verificando-se que a taxa de dados transferidos foi, na realidade, diferente da previamente calculada. Como demonstrado na Figura 4 e Figura 5, os dados transferidos durante a transmissão foram de **141 kb/s (141.000 bps)** e **212 kb/s (212.000 bps)**, respetivamente.

Esta discrepância ocorre porque o *Wireshark* não mede apenas o *bitrate* do vídeo, mas também o *overhead* dos protocolos envolvidos, retransmissões e eventuais pacotes de controlo, o que resulta numa taxa de transferência de dados superior à calculada inicialmente.

Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s
▼ Frame	100.0	396	100.0	273900	156 k	0	0	0
▼ Ethernet	100.0	396	2.0	5544	3158	0	0	0
▼ Internet Protocol Version 6	0.5	2	0.0	80	45	0	0	0
Open Shortest Path First	0.5	2	0.0	72	41	2	72	41
▼ Internet Protocol Version 4	99.0	392	2.9	7840	4466	0	0	0
▼ Transmission Control Protocol	97.0	384	94.9	259956	148 k	192	6144	3499
Data	48.5	192	90.4	247668	141 k	192	247668	141 k
Open Shortest Path First	2.0	8	0.1	352	200	8	352	200
Address Resolution Protocol	0.5	2	0.0	56	31	2	56	31

Figura 4: Captura do tráfego de 2 clientes no Wireshark.

Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s
▼ Frame	100.0	587	100.0	410316	234 k	0	0	0
▼ Ethernet	100.0	587	2.0	8218	4694	0	0	0
▼ Internet Protocol Version 6	0.2	1	0.0	40	22	0	0	0
Open Shortest Path First	0.2	1	0.0	36	20	1	36	20
▼ Internet Protocol Version 4	99.5	584	2.8	11680	6672	0	0	0
▼ Transmission Control Protocol	98.1	576	95.0	389934	222 k	288	9216	5264
Data	49.1	288	90.5	371502	212 k	288	371502	212 k
Open Shortest Path First	1.4	8	0.1	352	201	8	352	201
Address Resolution Protocol	0.3	2	0.0	56	31	2	56	31

Figura 5: Captura do tráfego de 3 clientes no Wireshark.

2.1.2. Alínea a - Comente os protocolos utilizados na transferência, bem como a experiência que o utilizador terá caso o link utilizado tenha perdas.

Através da Figura 5, conferimos que a pilha protocolar é constituída pelos protocolos *Ethernet*, *IPv4* para a rede e *TCP* para transporte e transferência de dados. Sendo assim, o protocolo utilizado na transferência de dados é unicamente *TCP*, e este garante:

- **Confiabilidade** - garante que todos os pacotes cheguem ao destino corretamente e na ordem correta. Se um pacote perder-se ou chegar danificado, ele é reenviado.
- **Controlo de Fluxo** - ajusta a taxa de envio de pacotes conforme a capacidade da rede e do cliente, o que ajuda a evitar a sobrecarga da rede.
- **Controlo de Congestionamento** - ajusta a velocidade de transmissão com base no congestionamento da rede, reduzindo a taxa de transmissão se houver sinais do mesmo.

Relativamente a possível perda de pacotes, em teoria seria:

- **1 Cliente (VLC)** - a experiência do utilizador será razoavelmente estável, a menos que as perdas sejam significativas. O *TCP* pode ajustar a taxa de transmissão de forma eficaz, mas qualquer perda de pacotes resultará em pequenas pausas devido ao *buffering*.
- **2 Clientes (VLC e *ffplay*)** - se existirem perdas, ambos os fluxos poderão ser afetados, com uma maior probabilidade de interrupções.
- **3 Clientes (VLC, *ffplay*, *ffplay*)** - com três clientes a situação pode agravar-se, especialmente se o *link* possuir uma largura de banda limitada. O congestionamento aumentará e as perdas de pacotes podem ser mais frequentes, levando a uma má experiência para o utilizador, com mais tempo de *buffering* e má qualidade de transmissão.

2.1.3. Alínea b - Identifique o número total de fluxos gerados e elabore um gráfico que demonstre a evolução do débito dependendo do número de clientes

Wireshark - Conversations - veth1.0.aa													
Ethernet · 3	IPv4 · 2	IPv6 · 1	TCP · 1	UDP									
Address A	Port A	Address B	Port B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
10.0.0.20	47242	10.0.0.10	9090	130	91 k	65	4290	65	86 k	0.000000	11.9349	2875	

Figura 6: Captura do trafego de 1 cliente no Wireshark.

Wireshark - Conversations - veth1.0.aa													
Ethernet · 4	IPv4 · 3	IPv6 · 1	TCP · 2	UDP									
Address A	Port A	Address B	Port B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
10.0.0.20	47242	10.0.0.10	9090	158	113 k	79	5214	79	108 k	0.573520	11.2434	3709	
10.0.2.20	60122	10.0.0.10	9090	158	113 k	79	5214	79	108 k	0.573692	11.2434	3709	

Figura 7: Captura do trafego de 2 clientes no Wireshark.

Wireshark - Conversations - final.pcapng													
Ethernet · 4	IPv4 · 4	IPv6 · 1	TCP · 3	UDP									
Address A	Port A	Address B	Port B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
10.0.0.20	59076	10.0.0.10	9090	176	124 k	88	5808	88	118 k	1.103784	9.8262	4728	
10.0.2.20	34560	10.0.0.10	9090	176	124 k	88	5808	88	118 k	1.103899	9.8261	4728	
10.0.2.21	50276	10.0.0.10	9090	176	124 k	88	5808	88	118 k	1.103976	9.8261	4728	

Figura 8: Captura do trafego de 3 clientes no Wireshark.

Ao analisar o *Wireshark*, verificámos que o número total de fluxos gerados correspondia ao número de clientes ligados ao servidor. Isto significa que é estabelecida uma nova conexão para cada cliente ligado, o que nos permite concluir que existem **3 fluxos**.

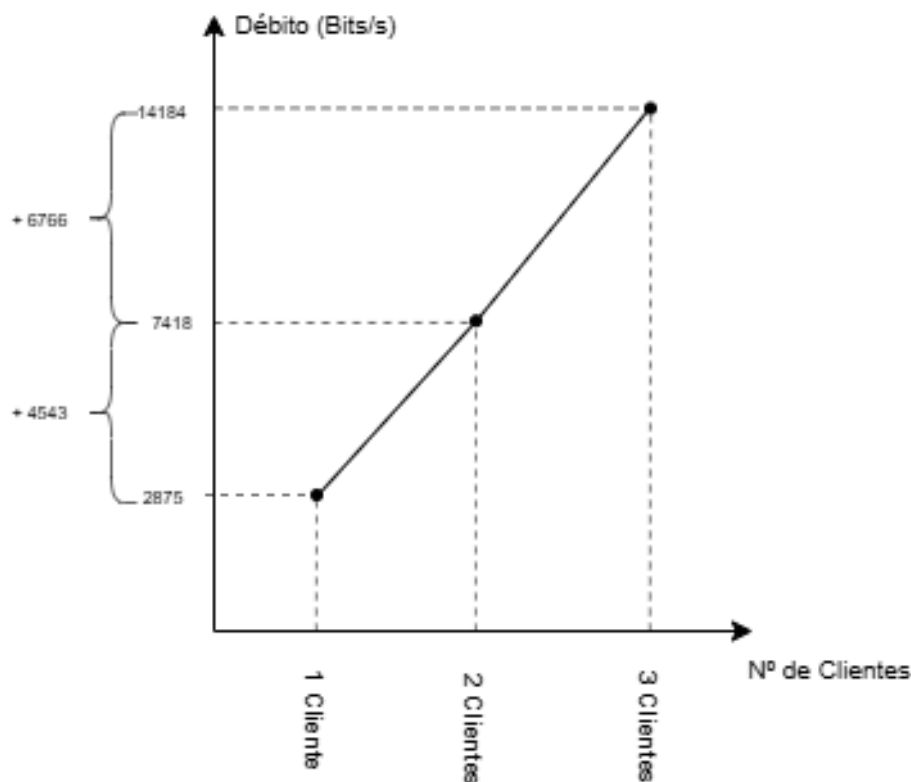


Figura 9: Gráfico da evolução do débito.

Com base nas análises realizadas nas capturas de pacotes Figura 6, Figura 7 e Figura 8, elaboramos o seguinte gráfico representado em Figura 9. Nele, observamos que, com 1 cliente, o débito foi de **2875 bps**; com 2 clientes, o débito subiu para **7418 bps**; e, com 3 clientes, atingiu **14184 bps**. Verifica-se que, ao passar de 1 para 2 clientes, houve um aumento de **4543 bps**, enquanto de 2 para 3 clientes, o acréscimo foi de **6766 bps**. Esses dados indicam que a evolução do débito em função do número de clientes tende a ser aproximadamente linear.

2.1.4. Alínea c - Comente a escalabilidade da solução para 1000 utilizadores, assim como 10000 utilizadores. Crie uma expressão matemática que expresse o débito necessário para que o servidor envie vídeo para N clientes

Para uma grande quantidade de clientes, a escalabilidade pode tornar-se um problema significativo, especialmente se cada cliente gerar um fluxo independente. Podemos dizer então que a largura de banda necessária aumenta, no mínimo, linearmente com o número de clientes. Sendo **Dtotal** o débito total, **N** o número de utilizadores e **4728bps** ($14184 \text{ bps} / 3$) o débito de 1 utilizador, decidimos utilizar o valor atingido com o último teste da questão anterior (3 utilizadores) uma vez que a discrepância tende a ser menor quanto mais utilizador ativos existem:

$$D_{total} \geq N \times 4728 \text{ bps}$$

Para 1000 clientes

sendo $N = 1000$ podemos calcular o D_{total} mínimo:

$$D_{total} \geq N \times 4728$$

$$D_{total} \geq 1000 \times 4728$$

$$D_{total} \geq 4,728 \text{ Mbps}$$

Para 10000 clientes

Podemos calcular o D_{total} mínimo:

$$D_{total} \geq N \times 4728$$

$$D_{total} \geq 10000 \times 4728$$

$$D_{total} \geq 47,28 \text{ Mbps}$$

3. Etapa 2 - Streaming adaptativo sobre HTTP (MPEG-DASH)

3.1. Questão 2 - Utilize o Wireshark para determinar a largura de banda necessária, em bits por segundo, para que o cliente de streaming consiga receber o vídeo no Firefox e qual a pilha protocolar usada neste cenário. Explique como obteve esta informação.

Através da captura do *Wireshark* apresentada na Figura 10, é possível verificar que a pilha protocolar utilizada foi composta por **Ethernet, IPv4, TCP e HTTP**, conforme um dos pacotes *HTTP* capturados. Sendo que, como referido anteriormente, o Ethernet e IPv4 são de rede, o TCP relativo a transporte de dados e HTTP a transferência dos mesmos.

Para determinar a largura de banda necessária, analisámos as informações do protocolo *IPv4* capturado, que se encontra selecionado na Figura 10. Posteriormente, consultámos as *Statistics -> Conversations* no *Wireshark* e identificámos o débito utilizado. Como se pode observar na Figura 10, o débito foi de **132 kbps** (132000 bps), o que indica que, neste cenário, a largura de banda necessária para o cliente receber o vídeo é de **132 kbps**.

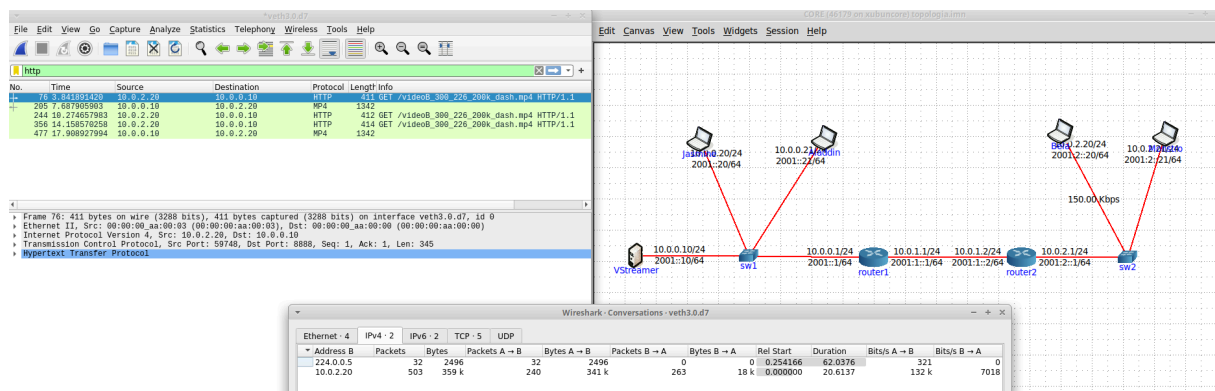


Figura 10: Cenário utilizado para as medições propostas

3.2. Questão 3 - Compare a largura de banda medida na questão anterior com a que é disponibilizada pelo ffplay. Qual é a razão para a diferença entre as duas?

Foi medido no *ffplay* uma taxa de bits de **135 kbps**, enquanto no *Wireshark* o valor obtido foi de **132 kbps**, como mencionado anteriormente. O valor obtido pelo *ffplay*, reflete normalmente a taxa de bits do vídeo, enquanto a medição feita no *Wireshark* pode incluir sobrecarga de protocolo (como cabeçalhos de pacotes, retransmissões, entre outros fatores).

Embora fosse esperado que o *ffplay* apresentasse uma largura de banda inferior, isso não foi observado nestas medições. Como demonstrado na Figura 12, embora dois dos pedidos tenham apresentado um tamanho superior ao registrado no *ffplay*, um terceiro pedido necessitou de menos dados, já que parte das informações já estava em *cache* no *Firefox*.

Sendo assim, essa situação resultou numa largura de banda inferior à medida pelo *ffplay*.

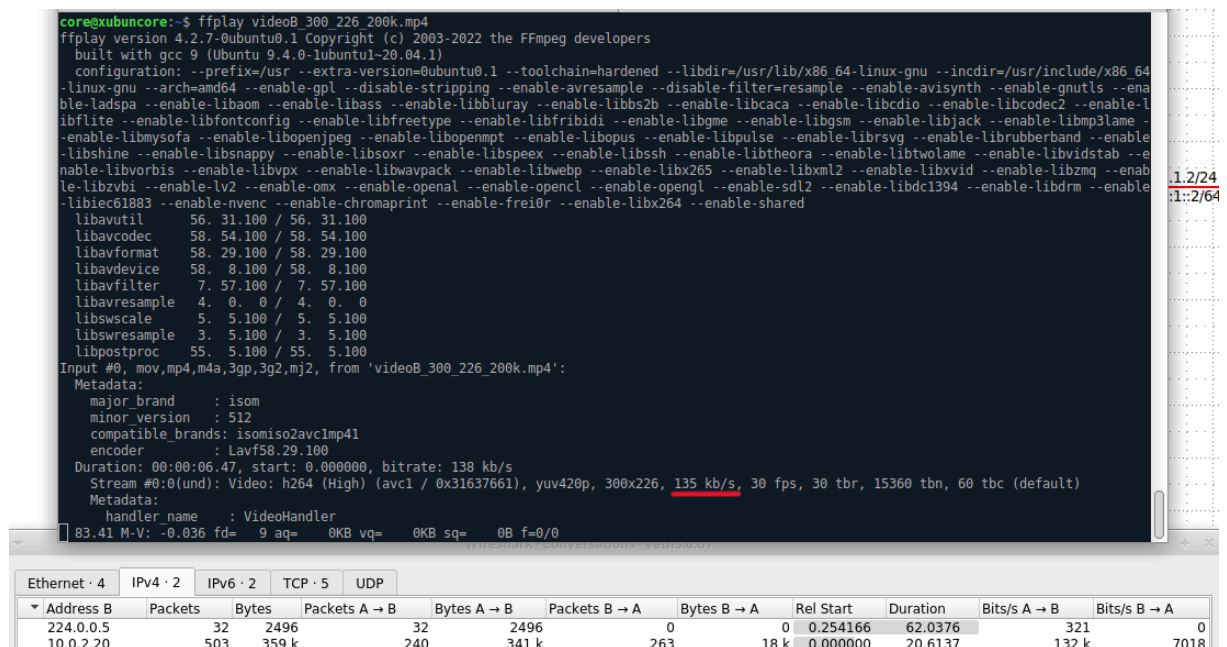


Figura 11: Capturas do ffplay e do wireshark respetivamente

Address A	Port A	Address B	Port B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
10.0.2.20	60046	10.0.0.10	8888	171	125 k	89	6227	82	119 k	1.779880	6.3944	7790	149 k
10.0.2.20	60048	10.0.0.10	8888	98	73 k	50	3318	48	69 k	8.211010	3.7318	7112	149 k
10.0.2.20	60050	10.0.0.10	8888	159	125 k	75	5314	84	119 k	9.811769	8.5455	4974	112 k

Figura 12: Captura do wireshark relativamente aos 3 pedidos.

3.3. Questão 4 - Ajuste o débito dos links da topologia de modo que o cliente no portátil Bela exiba o vídeo de menor resolução e o cliente no portátil Alladin exiba o vídeo com mais resolução. Mostre evidências e justifique a largura de banda necessária para que o stream de vídeo sofra alterações.

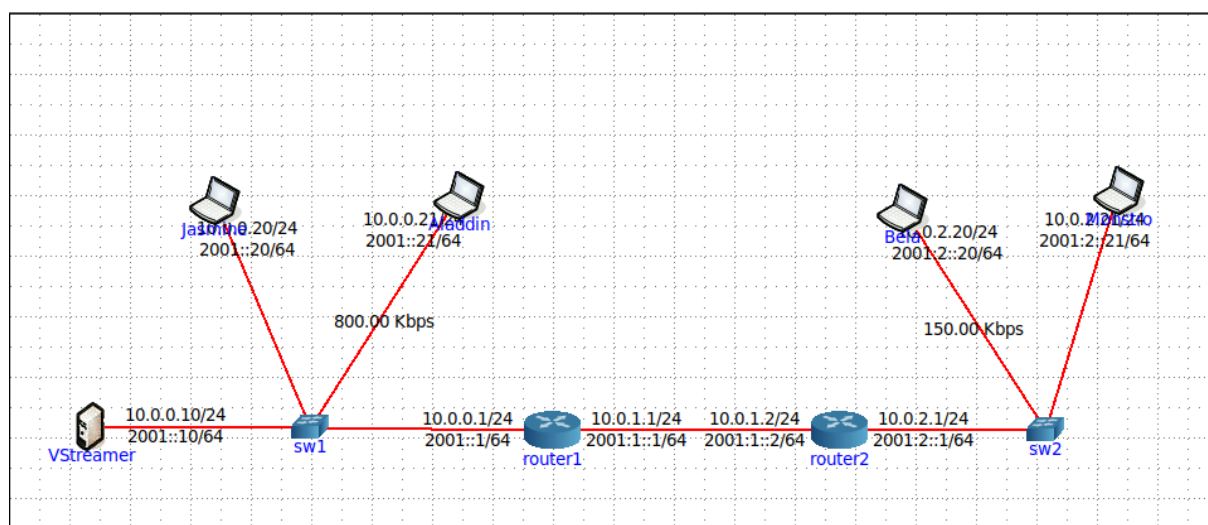


Figura 13: Topologia com as limitações aplicadas.

Para que o portátil **Bela** exiba o vídeo de menor resolução e o cliente no portátil **Alladin** exiba um de maior, é necessário que a largura de banda da **Bela** seja também **menor** que a do **Alladin**. Para isso decidimos definir a largura de banda da bela para **150 kbps** limitando assim a sua *stream* no vídeo mais pequeno.

E para o portátil **Alladin** começamos por definir **700 kbps** porque, sendo o valor do *ffplay* **679kbps**, este seria o mínimo para que fosse possível manter a *stream* no vídeo de maior tamanho. No entanto, observamos que, com **700 kbps**, o cliente a certa altura pedia ao servidor para baixar a resolução do vídeo para a resolução mínima disponível, apesar de não haver nenhuma limitação imposta. Sendo assim, decidimos subir para os **800kbps**. Com isto o cliente passa para a resolução média mantendo uma resolução diferente da **Bela**. Também podemos verificar isso na Figura 14, a partir do *Wireshark*, observamos que com a linha azul corresponde à cliente **Bela**, que começa com uma resolução de **1000k** e reduz para **200k**. Já a linha vermelha corresponde ao cliente **Aladdin**, que também inicia com **1000k** e diminui para **500k**.

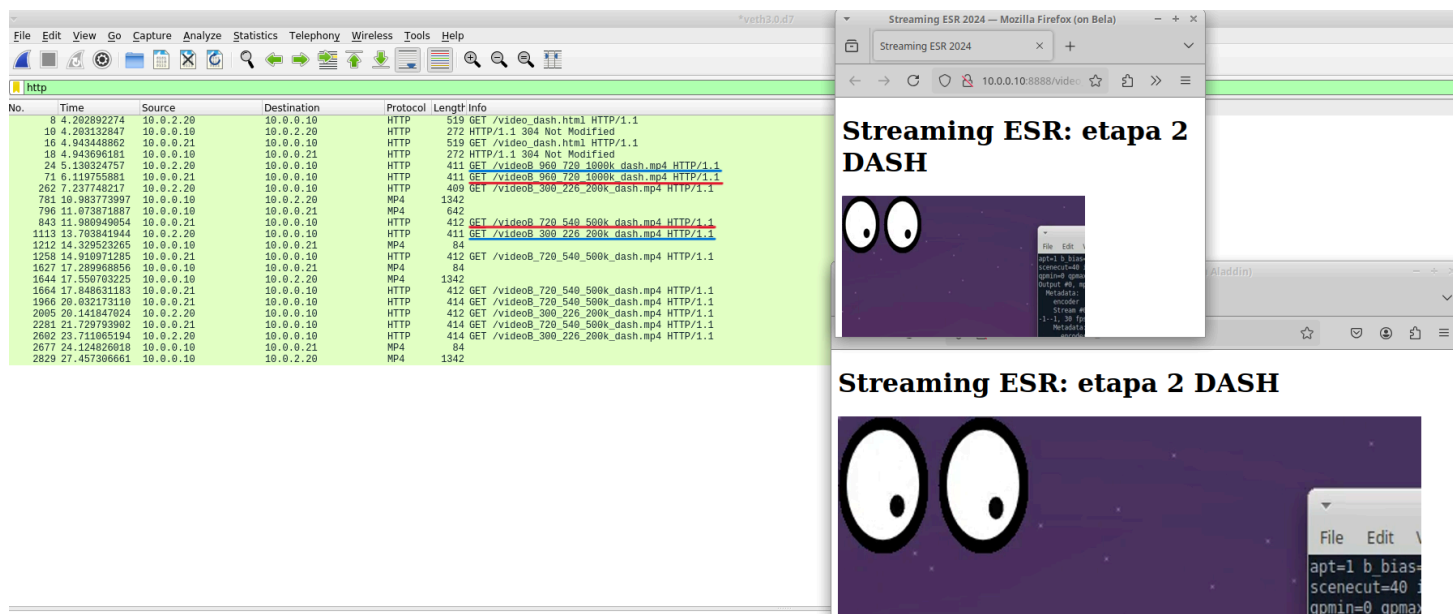


Figura 14: streams e captura do wireshark.

3.4. Questão 5 - Descreva o funcionamento do DASH neste caso concreto, referindo o papel do ficheiro MPD criado e comparando o modelo de streaming com o que foi utilizado na Questão 1

O **DASH** (*Dynamic Adaptive Streaming over HTTP*), é uma tecnologia de transmissão de vídeo de forma adaptativa, em que ajusta a qualidade do *streaming* com base na largura de banda disponível. O **DASH** envolve vários componentes, sendo o **MPD** (*Media Presentation Description*) o mais importante, pois serve como um “manifesto” do conteúdo.

O **MPD**, é um ficheiro que contém uma descrição detalhada de todos os segmentos disponíveis, sendo que descreve as versões do vídeo, a resolução, a taxa de *bits* e *URLs* dos segmentos do vídeo. Também possui informações sobre as regras de adaptação, como também a lógica de selecionar a qualidade apropriada com base nas condições de rede do cliente.

Para comparar este modelo com o modelo utilizado na questão 1, criamos esta seguinte tabela:

	Streaming adaptativo sobre HTTP (MPEG-DASH)	Streaming HTTP simples sem adaptação dinâmica de débito
Resiliência à Rede	Baixa	Alta
Qualidade de Experiência	Menos consistente	Mais consistente
Eficiência da Largura de Banda	Menos eficiente	Mais eficiente
Latência	Pode aumentar com o buffer	Levemente maior, mas contínuo
Implementação	Simple	Complexa
Casos de Uso	Redes estáveis, sem variação	Redes complexas e de escalabilidade alta

Resumidamente, o **Streaming HTTP Simples** funciona bem em redes estáveis e para cenários

simples, enquanto o **MPEG-DASH** é ideal para redes complexas, fornecendo uma experiência de vídeo adaptativa que se ajusta à largura de banda disponível e às condições da rede em tempo real.

4. Etapa 3 - Streaming RTP/RTCP unicast sobre UDP e multicast com anúncios SAP

4.1. Questão 6 - Compare o cenário unicast aplicado com o cenário multicast. Mostre vantagens e desvantagens na solução multicast ao nível da rede, no que diz respeito a escalabilidade (aumento do nº de clientes) e tráfego na rede. Tire as suas conclusões também para os cenários de 1000 e 10000 clientes.

4.1.1. Unicast

No cenário **unicast**, a informação é transmitida exclusivamente para um único endereço de cliente. Cada cliente que pretende aceder ao conteúdo multimédia, é estabelecida uma conexão separada, através da qual o cliente recebe uma cópia individual dos dados, o que resulta num elevado consumo de largura de banda por parte do servidor. Além disso, à medida que o número de clientes aumenta, ocorre uma sobrecarga na infraestrutura da rede. Para ilustrar este caso, foram utilizados os dispositivos **Monstro** e **Bela**, como podemos ver nas Figura 15 e Figura 16.

Ethernet · 4		IPv4 · 2		IPv6 · 2		TCP		UDP · 2					
Address A	Port A	Address B	Port B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
10.0.0.10	60049	10.0.2.21	6666	447	345 k	447	345 k	0	0	0.000000	16.2021	170 k	
10.0.0.10	60050	10.0.2.21	6667	3	210	3	210	0	0	4.596110	10.0979	166	

Figura 15: Captura da conversações em Unicast com 1 cliente.

Ethernet · 6		IPv4 · 3		IPv6 · 4		TCP		UDP · 4					
Address A	Port A	Address B	Port B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
10.0.0.10	45807	10.0.2.21	6666	874	674 k	874	674 k	0	0	0.000000	31.6544	170 k	
10.0.0.10	55688	10.0.2.20	6666	871	664 k	871	664 k	0	0	0.010783	31.6442	168 k	
10.0.0.10	45808	10.0.2.21	6667	7	490	7	490	0	0	0.154912	30.2497	129	
10.0.0.10	55689	10.0.2.20	6667	6	420	6	420	0	0	3.513186	25.1497	133	

Figura 16: Captura da conversações em Unicast com 2 clientes.

Com apenas um cliente, não seria possível demonstrar que era criada uma conexão separada para cada cliente, portanto decidimos acrescentar o dispositivo da **Bela** para o poder demonstrar. Sendo assim, através da Figura 16, podemos visualizar que a cada novo cliente é gerado uma nova conexão, e que cada cliente ocuparia aproximadamente a mesma largura de banda comprovando o que tínhamos referido anteriormente.

4.1.2. Multicast

No cenário **multicast**, a informação é transmitida uma única vez pela rede para um grupo de destinatários. Aqueles que desejam aceder a esse conteúdo inscrevem-se no grupo, o que resulta numa redução de uso da largura de banda e redução do tráfego na rede, como ilustrado na Figura 17.

Ethernet · 2		IPv4 · 2		IPv6		TCP		UDP · 3					
Address A ▾	Port A	Address B	Port B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
10.0.0.10	49522	224.0.0.224	6666	587	419 k	587	419 k	0	0	0.000000	21.5423	155 k	
10.0.0.10	48038	224.2.127.254	9875	4	1464	4	1464	0	0	4.797398	15.0504	778	
10.0.0.10	49523	224.0.0.224	6667	4	280	4	280	0	0	4.797445	15.0504	148	

Figura 17: Captura da conversações em Multicast com 4 clientes.

Ao utilizar o este método, o servidor é capaz de atender os 4 clientes (**Jasmine**, **Aladdin**, **Bela** e **Monstro**) com apenas um único fluxo de dados, ocupando somente **155 kbps** da largura de banda do servidor. Independentemente do número de clientes conectados, a ocupação da largura de banda

permanece praticamente a mesma, pois o fluxo **multicast** é compartilhado entre todos os clientes. Isso contrasta com o cenário **unicast**, onde seria necessário criar uma nova conexão para cada cliente, consumindo largura de banda adicional e sobrecarregando o servidor.

A vantagem de utilizar **multicast** em relação ao **unicast**, é o facto que este permite a transmissão de dados de um único remetente para vários recetores em simultâneo, ao contrário do **unicast** que replica os dados para cada destinatário separadamente. Já a desvantagem de utilizar **multicast** é o pouco controlo de quem acede às informações do grupo, o que pode levar a problemas na segurança da rede.

5. Conclusão

Este trabalho prático permitiu analisar diferentes cenários de *streaming* de vídeo, desde a utilização de *HTTP* simples até técnicas mais avançadas como o *MPEG-DASH* e transmissão *multicast* via *RTP/RTCP*. Através das medições realizadas, foi possível perceber a importância de ajustar corretamente a largura de banda e a resolução dos vídeos conforme as condições da rede e as características dos clientes.

Na Etapa 1, apesar dos problemas encontrados com o *streaming* através do *browser* do *Firefox*, verificamos que o *streaming* via *HTTP* sem adaptação dinâmica é viável em redes com condições estáveis e para um número limitado de clientes. No entanto, quando escalamos para um grande número de utilizadores, a eficiência cai significativamente, tornando essa solução menos adequada para ambientes de grande escala.

Já na Etapa 2, com o uso do *MPEG-DASH*, observamos uma melhoria na adaptação da qualidade de vídeo com base na largura de banda disponível. Esta abordagem também demonstrou ser mais eficiente em redes complexas, garantindo assim uma melhor qualidade de experiência para os utilizadores, mesmo em cenários com variações nas condições de rede.

Finalizando na Etapa 3, evidenciaram-se as vantagens do uso de *multicast* em redes com muitos clientes, comparado ao *unicast*. O *multicast* mostrou-se mais eficiente em termos de tráfego e escalabilidade, uma vez que permitia que múltiplos clientes acessem ao mesmo fluxo de dados, reduzindo a carga sobre o servidor.