

# Universidade do Minho

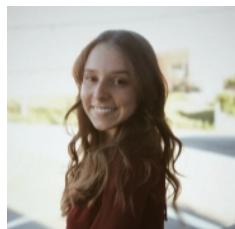
Mestrado em Engenharia Informática

## Engenharia de Serviços em Rede

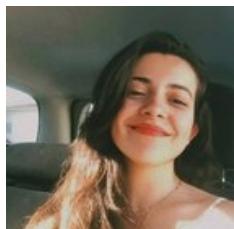
### Trabalho Prático 1:

*Streaming* de áudio e vídeo a pedido e em tempo real

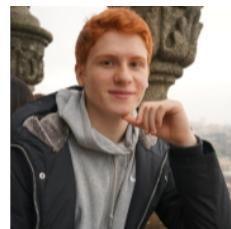
### Grupo 5 – PL7



Ana Murta  
pg50184



Ana Henriques  
pg50196



Francisco Peixoto  
pg47194

outubro, 2022

# Conteúdo

<b>1</b>	<b><u>Etapa 1: Streaming HTTP simples sem adaptação dinâmica de débito</u></b>	<b>4</b>
1.1	Questão 1 . . . . .	4
<b>2</b>	<b><u>Etapa 2: Streaming adaptativo sobre HTTP (MPEG-DASH)</u></b>	<b>7</b>
2.1	Questão 2 . . . . .	7
2.2	Questão 3 . . . . .	8
2.3	Questão 4 . . . . .	9
<b>3</b>	<b><u>Etapa 3: Streaming RTP/RTCP unicast sobre UDP e multicast com anúncios SAP</u></b>	<b>11</b>
3.1	Questão 5 . . . . .	11
<b>4</b>	<b>Conclusões</b>	<b>13</b>

# Listas de Figuras

1.1	Topologia <i>core</i> adotada . . . . .	4
1.2	<b>ffmpeg -i videoA.mp4</b> . . . . .	5
1.3	Fluxo TCP com 1 cliente – VLC . . . . .	5
1.4	Tráfego com 1 cliente – VLC . . . . .	5
1.5	Fluxos TCP com 2 clientes – VLC e Firefox . . . . .	5
1.6	Tráfego com 2 clientes – VLC e Firefox . . . . .	6
1.7	Fluxos TCP com 3 clientes – VLC, Firefox e ffplay . . . . .	6
1.8	Tráfego com 3 clientes – VLC, Firefox e ffplay . . . . .	6
2.1	<b>cat video_manifest.mpd -- dimensão 160x100</b> . . . . .	7
2.2	<b>cat video_manifest.mpd -- dimensão 320x200</b> . . . . .	8
2.3	<b>cat video_manifest.mpd -- dimensão 640x400</b> . . . . .	8
2.4	Topologia com o ajuste do débito dos links . . . . .	8
2.5	Valor atribuído ao débito do <i>link</i> entre switch 2 e Bela . . . . .	9
2.6	Captura <i>wireshark</i> do tráfego no link de saída do servidor . . . . .	9
2.7	Valor atribuído ao débito do <i>link</i> entre switch 1 e Jasmin . . . . .	9
2.8	Captura <i>wireshark</i> do tráfego no link de saída do servidor . . . . .	9
3.1	Nova topologia para cenário <i>multicast</i> . . . . .	12
3.2	Captura <i>wireshark</i> do tráfego num cenário <i>unicast</i> . . . . .	12
3.3	Captura <i>wireshark</i> do tráfego num cenário <i>multicast</i> . . . . .	12

# Capítulo 1

## Etapa 1: *Streaming HTTP simples sem adaptação dinâmica de débito*

### 1.1 Questão 1

Capture três pequenas amostras de tráfego no *link* de saída do servidor, respetivamente com 1 cliente (VLC), com 2 clientes (VLC e Firefox) e com 3 clientes (VLC, Firefox e ffplay). Identifique a taxa em bps necessária (usando o `ffmpeg -i videoA.mp4` e/ou o próprio wireshark), o encapsulamento usado e o número total de fluxos gerados. Comente a escalabilidade da solução. Ilustre com evidências da realização prática do exercício (ex: capturas de ecrã).

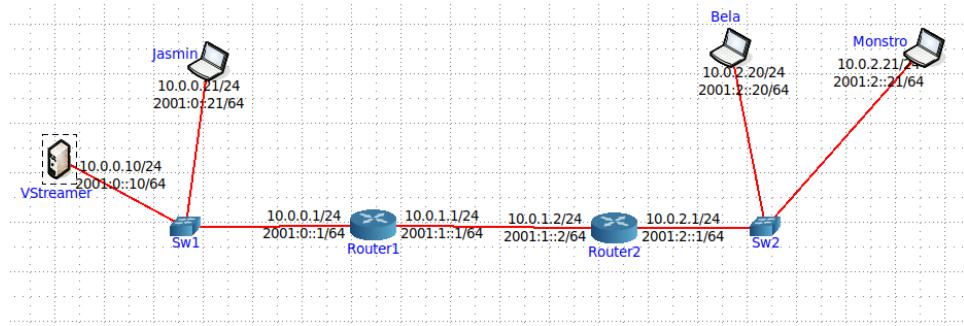


Figura 1.1: Topologia *core* adotada

Ao executar o comando `ffmpeg -i videoA.mp4` – ilustrado na Figura 1.2, podemos ver que o *bitrate* é 18 kb/s, o que, convertendo para bps, significa que a taxa necessária para transmitir o vídeo `videoA.mp4` é de 144 kbps.

Figura 1.2: `ffmpeg -i videoA.mp4`

Recorrendo aos dados estatísticos do *wireshark*, ilustrado na Figura 1.3, conseguimos analisar que, com apenas 1 cliente – VLC, existe somente um fluxo TCP.

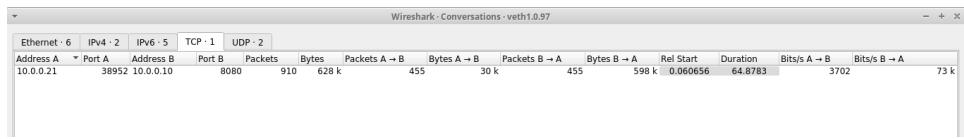


Figura 1.3: Fluxo TCP com 1 cliente – VLC

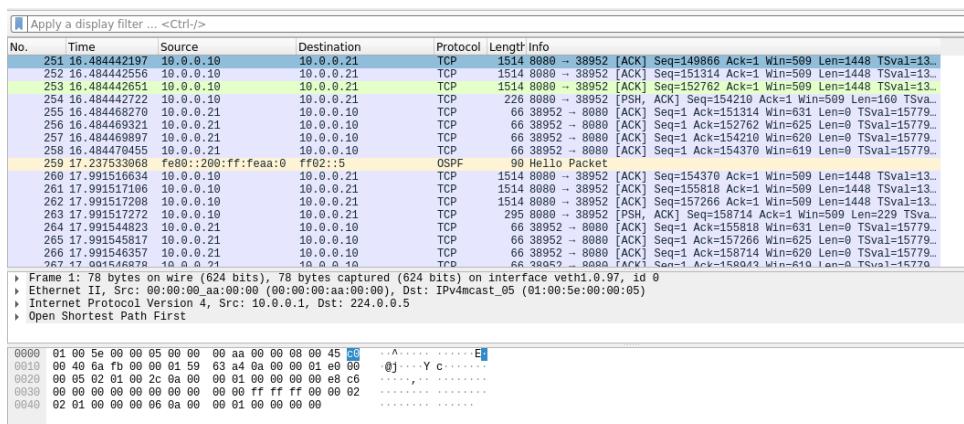


Figura 1.4: Tráfego com 1 cliente – VLC

Posteriormente, capturamos outra amostra de tráfego no *link* de saída do servidor, mas desta vez com 2 clientes – VLC e Firefox. O resultado é a existência de 2 fluxos TCP, tal como apresentado na Figura 1.5.

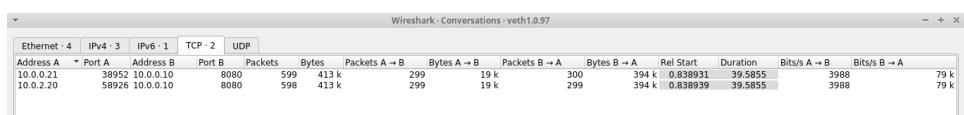


Figura 1.5: Fluxos TCP com 2 clientes – VLC e Firefox

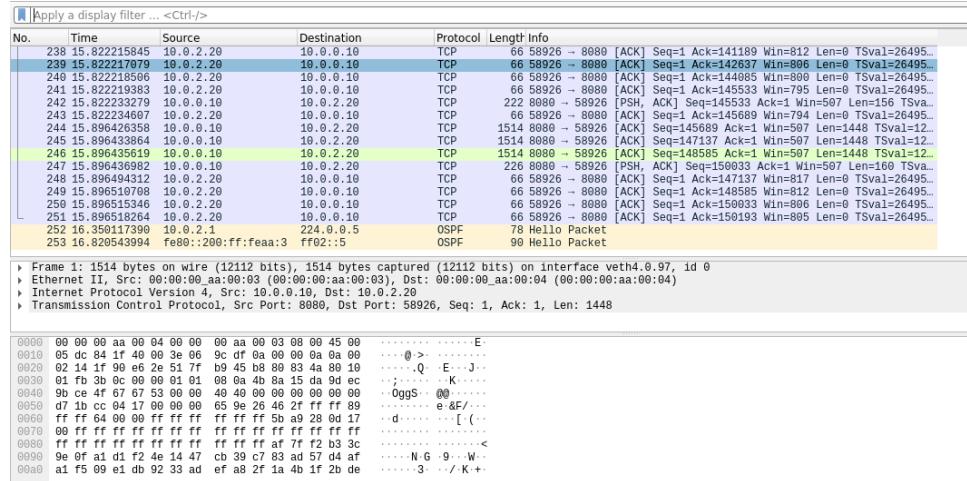


Figura 1.6: Tráfego com 2 clientes – VLC e Firefox

Finalmente, capturamos uma última amostra de tráfego com 3 clientes – VLC, Firefox e ffplay, o que originou a existência de 3 fluxos TCP, comprovada na Figura 1.7.

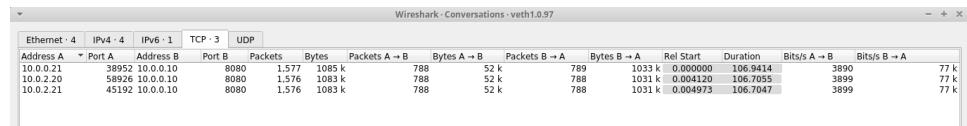


Figura 1.7: Fluxos TCP com 3 clientes – VLC, Firefox e ffplay

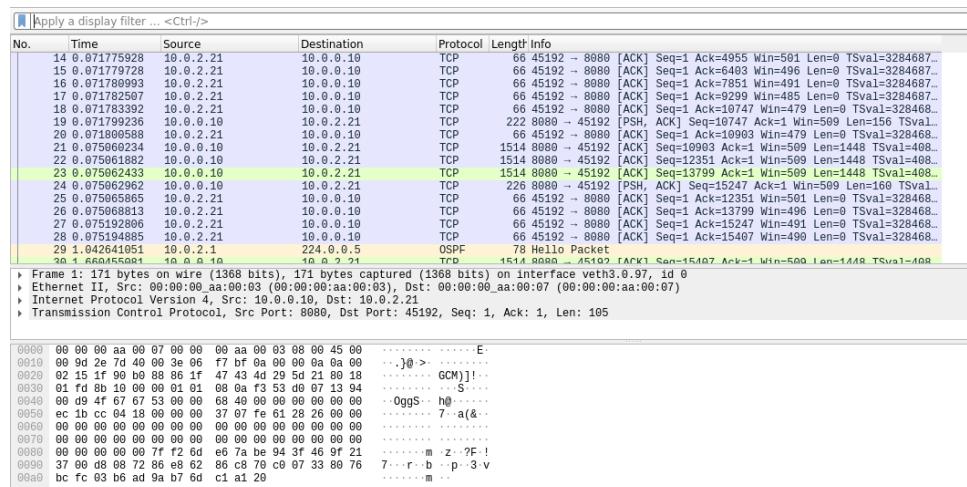


Figura 1.8: Tráfego com 3 clientes – VLC, Firefox e ffplay

## Interpretação dos resultados:

Com as amostras capturadas, e comparando os dados estatísticos retirados do *wireshark*, verifica-se que existem tantos fluxos TCP quantos clientes adicionados ao tráfego no *link* de saída do servidor. Isto indica que o servidor responde a um cliente de cada vez; logo, a solução não é escalável. Quanto maior o número de clientes, menor a qualidade do serviço.

# Capítulo 2

## Etapa 2: *Streaming adaptativo sobre HTTP (MPEG-DASH)*

### 2.1 Questão 2

Diga qual 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.

A pilha protocolar aplicada neste cenário é TCP/IP. Executando o comando `cat video_manifest.mpd`, cujo resultado está ilustrado nas Figuras 2.1, 2.2, e 2.3, descobrimos o seguinte:

- Com a dimensão 160x100, é preciso uma largura de banda (*bandwith*) de 150467 bps.
- Com a dimensão 320x200, é preciso uma largura de banda (*bandwith*) de 389264 bps.
- Com a dimensão 640x400, é preciso uma largura de banda (*bandwith*) de 839974 bps.

Posto isto, a largura de banda mínima **teórica** para que o cliente de *streaming* consiga receber o vídeo é de 839974 bps. Todavia, existe uma taxa percentual de 10% a 15% de largura de banda mínima **superior** a esse valor, ou seja, é sempre preciso um pouco mais do que 839974 bps para que o vídeo seja recebido.

```
wl@wlsurface: ~ cat video_manifest.mpd
<?xml version="1.0"?>
<MPD file Generated with GPAC version 0.5.2-DEV-revision: 0.5.2-426-gc5ad4e4+dfsg5-5 at 2022-10-06T10:34:46.269Z-->
<MPD type="static" mediaPresentationDuration="PT0H0M0.500S" maxSegmentDuration="PT0H0M0.500S" profiles="urn:mpeg:dash:profile:full">
  <ProgramInformation moreInformationURL="http://gpac.sourceforge.net">
    <Title>video manifest.mpd generated by GPAC</Title>
  </ProgramInformation>
  <Period duration="PT0H0M0.6345">
    <AdaptationSet segmentAlignment="true" bitstreamSwitching="true" maxWidth="640" maxHeight="400" maxFrameRate="30" par="8:5" lang="und">
      <SegmentList>
        <SegmentList sourceURL="video_manifest_init.mpd"/>
        <SegmentList>
          <Representation id="1" mimeType="video/mp4" codecs="avc3.64800c" width="160" height="100" frameRate="30" sar="1:1" startWithSAP="0" bandwidth="150467">
            <BaseURL>video1_160_100_200k.dash.mp4</BaseURL>
            <SegmentList type="range">150467-150467</SegmentList>
            <SegmentURL mediaRange="0-927-944" indexRange="927-970"/>
            <SegmentURL mediaRange="7645-13712" indexRange="7645-7688"/>
            <SegmentURL mediaRange="13713-18278" indexRange="13713-13756"/>
            <SegmentURL mediaRange="18279-29584" indexRange="18279-29322"/>
            <SegmentURL mediaRange="29585-39552" indexRange="29585-29322"/>
            <SegmentURL mediaRange="34888-41247" indexRange="34888-34931"/>
            <SegmentURL mediaRange="41248-56782" indexRange="41248-41191"/>
            <SegmentURL mediaRange="41177-64116" indexRange="41177-56826"/>
            <SegmentURL mediaRange="64117-72469" indexRange="64117-56826"/>
            <SegmentURL mediaRange="72469-79933" indexRange="72469-72512"/>
            <SegmentURL mediaRange="79934-95787" indexRange="79934-79977"/>
            <SegmentURL mediaRange="95784-104886" indexRange="95784-95747"/>
            <SegmentURL mediaRange="104887-111453" indexRange="104887-104909"/>
            <SegmentURL mediaRange="111453-111465" indexRange="111453-111496"/>
            <SegmentURL mediaRange="111466-139776" indexRange="111466-131599"/>
            <SegmentURL mediaRange="139777-147677" indexRange="139777-139820"/>
            <SegmentURL mediaRange="147678-162151" indexRange="147678-147721"/>
            <SegmentURL mediaRange="162152-162379" indexRange="162152-162195"/>
          </SegmentList>
        </Representation>
      </SegmentList>
    </AdaptationSet>
  </Period>
</MPD>
```

Figura 2.1: `cat video_manifest.mpd -- dimensão 160x100`

```

<Representation id="2" mimeType="video/mp4" codecs="avc3.640014" width="320" height="200" frameRate="30" sar="1:1" startWithSAP="0" bandwidth="389264">
  <SegmentList timescale="15360" duration="7680">
    <SegmentURL mediaRange="928-16706" indexRange="928-971"/>
    <SegmentURL mediaRange="16707-34197" indexRange="16707-16750"/>
    <SegmentURL mediaRange="34198-45471" indexRange="34198-34241"/>
    <SegmentURL mediaRange="45472-74579" indexRange="45472-45515"/>
    <SegmentURL mediaRange="74580-89487" indexRange="74580-74623"/>
    <SegmentURL mediaRange="89488-106822" indexRange="89488-89531"/>
    <SegmentURL mediaRange="106823-19116" indexRange="106823-106866"/>
    <SegmentURL mediaRange="191167-196956" indexRange="191167-191210"/>
    <SegmentURL mediaRange="196957-199988" indexRange="196957-196980"/>
    <SegmentURL mediaRange="199989-208833" indexRange="199989-208833"/>
    <SegmentURL mediaRange="208833-208978" indexRange="208833-208926"/>
    <SegmentURL mediaRange="253971-276222" indexRange="253971-254014"/>
    <SegmentURL mediaRange="276223-301116" indexRange="276223-276266"/>
    <SegmentURL mediaRange="301117-346415" indexRange="301117-301160"/>
    <SegmentURL mediaRange="346416-365906" indexRange="346416-346459"/>
    <SegmentURL mediaRange="365907-362661" indexRange="365907-365950"/>
    <SegmentURL mediaRange="382662-419844" indexRange="382662-382705"/>
    <SegmentURL mediaRange="419845-429080" indexRange="419845-419888"/>
  </SegmentList>
</Representation>

```

Figura 2.2: cat video\_manifest.mpd -- dimensão 320x200

```

<Representation id="3" mimeType="video/mp4" codecs="avc3.640014" width="640" height="400" frameRate="30" sar="1:1" startWithSAP="0" bandwidth="839944">
  <SegmentList timescale="15360" duration="7680">
    <SegmentURL mediaRange="927-33798" indexRange="927-970"/>
    <SegmentURL mediaRange="33799-71810" indexRange="33799-33842"/>
    <SegmentURL mediaRange="71811-106823" indexRange="71811-106866"/>
    <SegmentURL mediaRange="94413-1348761" indexRange="94413-944567"/>
    <SegmentURL mediaRange="134877-183867" indexRange="134877-184920"/>
    <SegmentURL mediaRange="183868-222033" indexRange="183868-183911"/>
    <SegmentURL mediaRange="222034-320851" indexRange="222034-222077"/>
    <SegmentURL mediaRange="320852-360167" indexRange="320852-320895"/>
    <SegmentURL mediaRange="360168-399108" indexRange="360168-360211"/>
    <SegmentURL mediaRange="399109-443389" indexRange="399109-399152"/>
    <SegmentURL mediaRange="443389-548266" indexRange="443389-443432"/>
    <SegmentURL mediaRange="548267-576376" indexRange="548267-548310"/>
    <SegmentURL mediaRange="576377-631883" indexRange="576377-576420"/>
    <SegmentURL mediaRange="631884-754137" indexRange="631884-631927"/>
    <SegmentURL mediaRange="754138-791969" indexRange="754138-754181"/>
    <SegmentURL mediaRange="791970-826542" indexRange="791970-792013"/>
    <SegmentURL mediaRange="826543-906170" indexRange="826543-826586"/>
    <SegmentURL mediaRange="966171-966471" indexRange="966171-966214"/>
  </SegmentList>
</Representation>
</AdaptationSet>
</Period>

```

Figura 2.3: cat video\_manifest.mpd -- dimensão 640x400

## 2.2 Questão 3

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.

Para exibir o vídeo de menor resolução, definimos o valor 150.47 kbps como o débito do link entre o switch 2 e o portátil Bela. Para o vídeo de maior resolução, definimos o valor 2 Mbps como o débito do link entre o switch 1 e o portátil Jasmin.

Cada um destes ajustes pode ser verificado pela Figura 2.4.

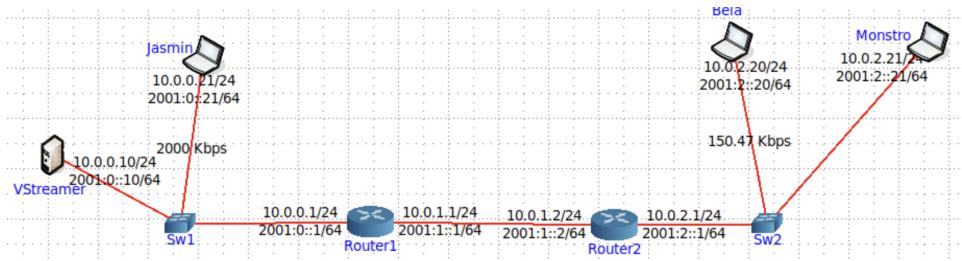


Figura 2.4: Topologia com o ajuste do débito dos links

Nas Figuras 2.6 e 2.8, temos evidências de como, de facto, os vídeos são exibidos com aqueles valores atribuídos ao débito de cada um dos *links*.

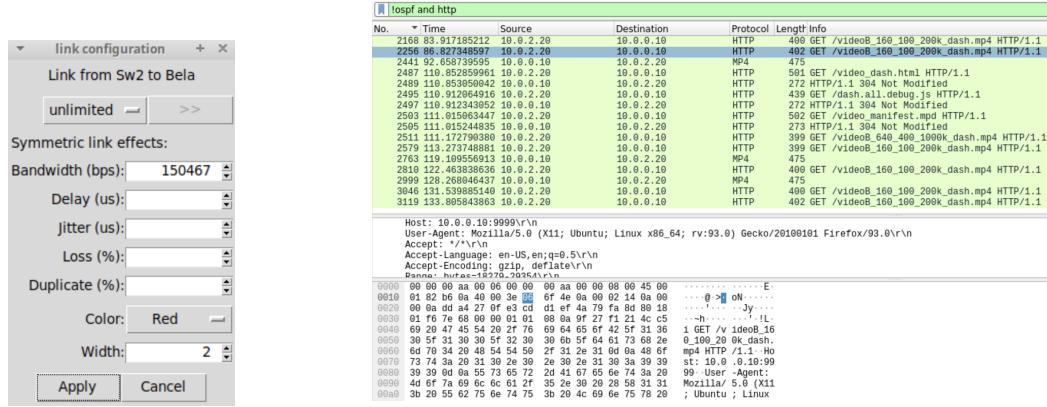


Figura 2.5: Valor atribuído ao débito do link entre switch 2 e Bela

Figura 2.6: Captura wireshark do tráfego no link de saída do servidor

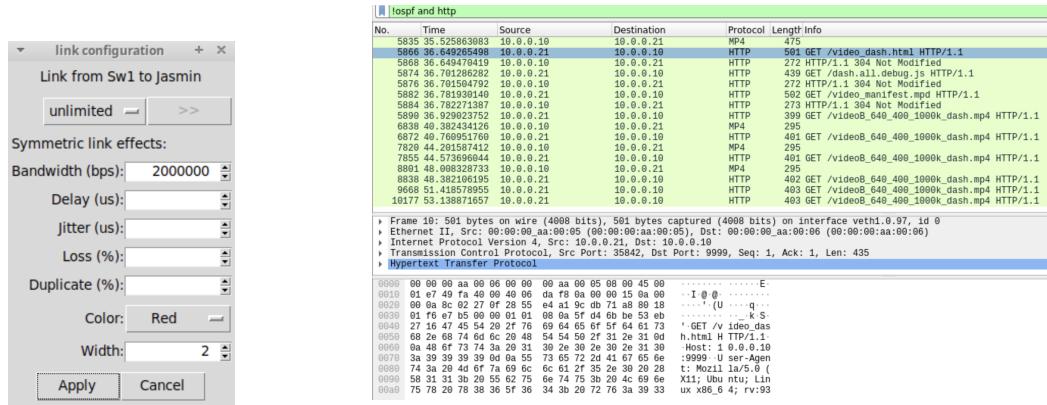


Figura 2.7: Valor atribuído ao débito do link entre switch 1 e Jasmin

Figura 2.8: Captura wireshark do tráfego no link de saída do servidor

## 2.3 Questão 4

Descreva o funcionamento do DASH neste caso concreto, referindo o papel do ficheiro MPD criado.

De acordo com o protocolo DASH:

O servidor divide o arquivo de vídeo em vários *chunks*, sendo estes armazenados e codificados em taxas diferentes.

O cliente é responsável por medir, periodicamente, a largura de banda *server-to-client*. Segundo o ficheiro MPD, este solicita um *chunk* de cada vez e escolhe a taxa máxima de codificação sustentável com base na largura de banda atual. O cliente pode ainda, em momentos distintos, escolher diferentes taxas de codificação dependendo da largura de banda disponível no momento.

Durante a resolução da questão 3, podemos comprovar o que foi referido em cima. Ao ajustar o débito do link entre o switch 1 e o portátil Jasmin para exibir o vídeo com maior resolução, a sua dimensão foi alterada dinamicamente durante a exibição do mesmo. Isto significa que, nos primeiros segundos, era reproduzido o vídeo de maior dimensão e, uma vez que o débito de rede não suportava esta transmissão, o vídeo de resolução média passou a ser exibido. Desta forma,

usando o protocolo DASH, é possível que o *streaming* de vídeo seja dinamicamente otimizado de acordo com as condições da rede.

O objetivo do ficheiro MPD criado é fornecer ao serviço de *streaming* diferentes formatações de vídeo, que poderão ser utilizadas pelo serviço, dependendo das capacidades da rede, definidas pelo protocolo DASH que foi explicado anteriormente.

Neste caso, o vídeo é disponibilizado pelo ficheiro MPD, cuja formatação é apresentada pelo conjunto de pontos abaixo e nele estão identificadas as resoluções com maior ou menor resolução, que serão escolhidas para o *streaming* conforme o débito da rede:

- Ficheiro de menor dimensão 160x100, para largura de banda (*bandwidth*) de 150467 bps.
- Ficheiro de média dimensão 320x200, para largura de banda (*bandwidth*) de 389264 bps.
- Ficheiro de maior dimensão 640x400, para largura de banda (*bandwidth*) de 839974 bps.

# Capítulo 3

## Etapa 3: Streaming RTP/RTCP unicast sobre UDP e multicast com anúnicos SAP

### 3.1 Questão 5

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.

A topologia usada nas questões anteriores é aplicada num cenário *unicast* e, como tal, temos um único canal de transmissão para receber e responder aos pedidos de todos os clientes na rede. No entanto, recorrer a um cenário *unicast* leva a um consumo ineficiente de recursos. Por outras palavras, se considerarmos uma rede que abrange um elevado número de clientes, haverá uma grande taxa de chegada de pedidos para a qual a capacidade da largura de banda terá de aumentar linearmente para acomodar as demandas de tráfego, o que não seria possível num contexto prático, levando ao congestionamento da rede e demonstrando que o cenário em *unicast* não é escalável.

A solução é, então, ponderar um modelo de entrega *multicast* onde, tal como podemos confirmar pela Figura 3.1, existe um único *switch* e, para cada cliente existente na rede, existe um canal de transmissão. Deste modo, é possível receber e responder a múltiplos pedidos simultaneamente porque a transmissão de dados é efetuada para diferentes pontos ao mesmo tempo.

Comparativamente com o modelo *unicast*, o modelo *multicast* tem inúmeras **vantagens**: uma delas é que, como estamos perante uma solução escalável, há uma maior eficácia na transmissão de dados numa rede com um elevado número de clientes; outra vantagem é já não haver um consumo ineficiente dos recursos, ou seja, haver um menor consumo da largura de banda porque os dados são transferidos consoante as necessidades de quem os pediu; finalmente, uma outra vantagem, é a conservação da qualidade do vídeo já que o *switch* apenas o envia para o cliente que o requeri.

Contudo, o cenário *multicast* também apresenta **desvantagens**, nomeadamente o facto de não garantir a entrega fiável dos dados. Isto porque, o *multicast* só altera o endereço IPv4, não muda a pilha protocolar e, portanto, tanto num cenário como no outro, é usado o UDP como protocolo da camada de transporte. Isto remete a que seja possível haver congestão de tráfego na rede.

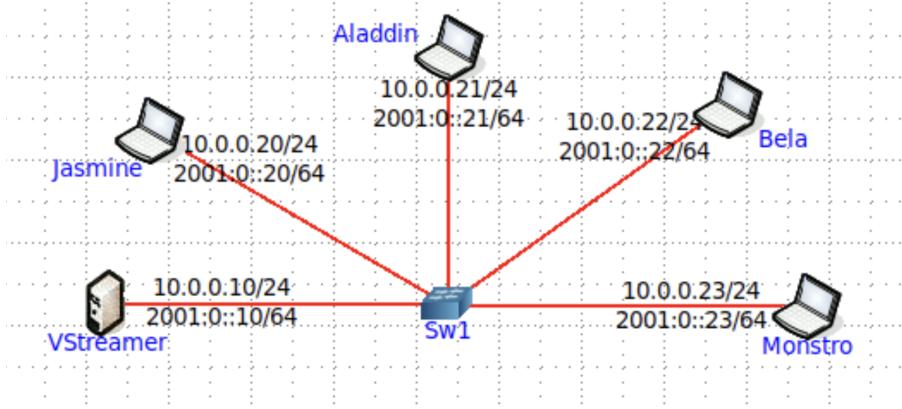


Figura 3.1: Nova topologia para cenário *multicast*

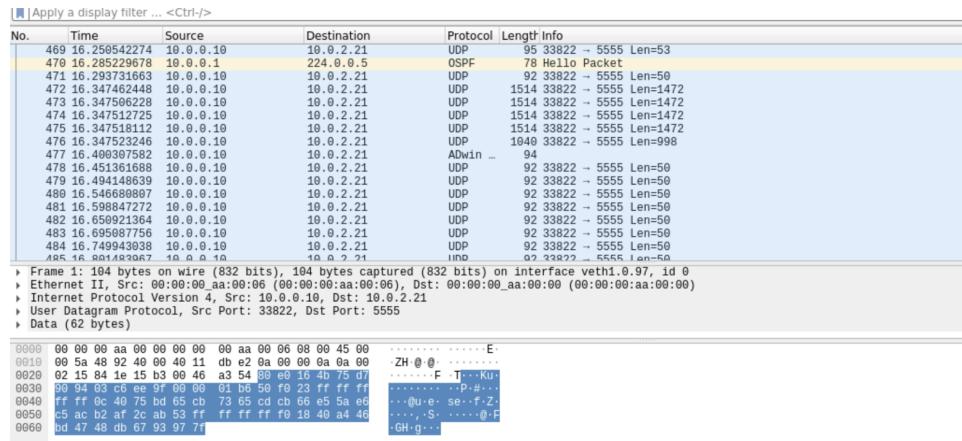


Figura 3.2: Captura *wireshark* do tráfego num cenário *unicast*

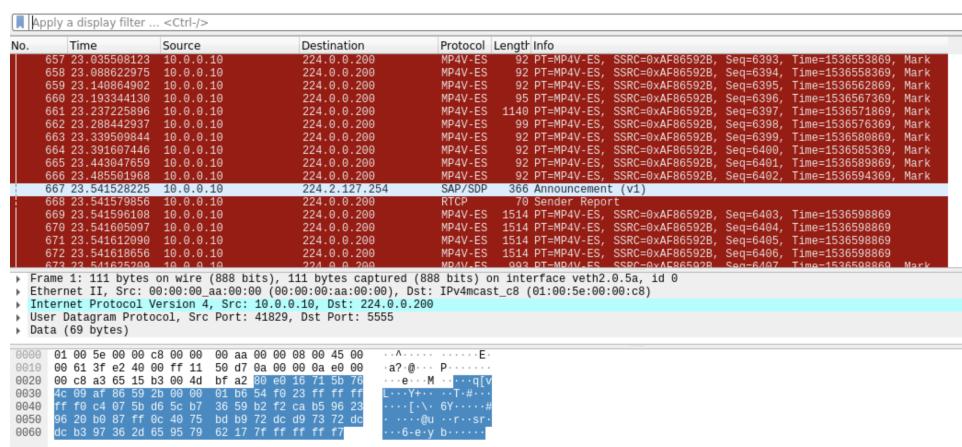


Figura 3.3: Captura *wireshark* do tráfego num cenário *multicast*

# Capítulo 4

## Conclusões

No decorrer deste trabalho prático, consolidou-se a matéria lecionada nas aulas teóricas acerca de *streaming* de áudio e vídeo a pedido e em tempo real.

Com a realização da primeira etapa, onde efetuamos um *streaming* HTTP simples sem adaptação dinâmica de débito, descobrimos que uma solução não escalável não é a melhor solução visto que exige mais a nível de desempenho: quanto maior o número de clientes, mais lento é o serviço e, consequentemente, menor é a sua qualidade.

Posteriormente, executámos a segunda tarefa, onde era solicitado um *streaming* adaptativo sobre HTTP (MPEG-DASH). No entanto, tivemos alguns problemas visto que, apesar da largura de banda necessária para uma determinada dimensão do vídeo ser inferior à largura de banda máxima imposta no *link*, foi preciso atribuir valores elevadíssimos para que todas as dimensões do vídeo fossem recebidas pelo cliente.

Finalmente, na terceira e última etapa, efetuámos um *streaming* RTP/RTCP *unicast* sobre UDP e *multicast* com anúncios SAP e as conclusões retiradas da comparação entre um cenário *unicast* e um cenário *multicast* encontram-se descritas na secção 3.