

# TP1 - Streaming de áudio e vídeo a pedido e em tempo real

---

*Autores:*

João Paulo Sousa Mendes (pg50483)

João Silva Torres (pg50499)

José Diogo Martins Vieira (pg50518)

October 13, 2022



**Universidade do Minho**  
Escola de Engenharia

## Etapa 1 - Streaming HTTP simples sem adaptação dinâmica de débito

**Questão 1 -** Capture três pequenas amostras de tráfego no link de saída do servidor, respectivamente 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).

Para a taxa bps necessária, primeiramente utilizamos o comando `ffmpeg -i videoA.mp4` para visualizar o bit rate do vídeo localmente. O valor para este foi 10kb/s (Figura 1). Contudo ao verificar o bps no Wireshark, verificamos que afinal a taxa real necessária para a transmissão do vídeo foi 19kb/s (VLC), para apenas um cliente (Figura 2). Este valor pode ser explicado pelo *overhead* da transmissão. Observamos também a taxa relativa à captura com 2 clientes: 34kb/s (VLC e Firefox) e com 3 clientes: 52kb/s (VLC, Firefox e ffplay), Fig 3 e 4, respectivamente.

```
ffmpeg version 4.2.7-0ubuntu0.1 Copyright (c) 2000-2022 the FFmpeg developers
  built with gcc 9 (Ubuntu 9.4.0-1ubuntu1~20.04.1)
  configuration: --prefix=/usr --extra-version=0ubuntu0.1 --toolchain=hardened --libdir=/usr/lib/x86_64-linux-gnu --incdir=/usr/include/
  --enable-ladspa --enable-libaom --enable-libass --enable-libbluray --enable-libbs2b --enable-libcaca --enable-libcdio --enable-l
  --enable-libjack --enable-libmp3lame --enable-libmysofa --enable-libopenjpeg --enable-libopenmpt --enable-libopus --enable-libpulse --enable-librsvg
  --enable-libtwolame --enable-libvidstab --enable-libvorbis --enable-libvpx --enable-libwavpack --enable-libwebp --enable-libx264 --enable-li
  --enable-libx265 --enable-libzmq --enable-libzimg --enable-openssl --enable-sdl2 --enable-libdca --enable-libdrm --enable-libiec61883 --enable-nvenc --enable-chromaprint --enable-frei0r --enable-libx264 --
  libavutil      56. 31.100 / 56. 31.100
  libavcodec     58. 54.100 / 58. 54.100
  libavformat    58. 29.100 / 58. 29.100
  libavdevice    58.  8.100 / 58.  8.100
  libavfilter     7. 57.100 /  7. 57.100
  libavresample   4.  0.  0 /  4.  0.  0
  libswscale     5.  5.100 /  5.  5.100
  libswresample   3.  5.100 /  3.  5.100
  libpostproc    55.  5.100 / 55.  5.100
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: 00:00:18.70, start: 0.000000, bitrate: 12 kb/s
  Stream #0:0(und): Video: h264 (High) (avc1 / 0x31637661), yuv420p, 160x100, 10 kb/s, 20 fps, 20 tbr, 10240 tbn, 40 tbc (default)
  Metadata:
    handler_name     : VideoHandler
```

Figure 1: Taxa bps esperada para a transmissão

Ethernet · 1		IPv4 · 1		IPv6		TCP · 1		UDP															
Address A	Port A	Address B	Port B	Packets	Bytes	Bits/s B → A	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration											
10.0.0.21	49612	10.0.0.10	8080	62	89 k	19 k	0	0	62	89 k	0.731446	36.3772											

Figure 2: Taxa bps real para a transmissão com 1 cliente

Ethernet · 2		IPv4 · 2		IPv6		TCP · 6		UDP									
Address A	Port A	Address B	Port B	Packets	Bytes	Bits/s B → A	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s				
10.0.0.21	49612	10.0.0.10	8080	53	75 k	17 k	0	0	53	75 k	0.727542	34.8866					
10.0.2.20	47694	10.0.0.10	8080	4	6056	17 k	0	0	4	6056	0.727834	2.7572					

Figure 3: Taxa bps real para a transmissão com 2 clientes

Address A	Port A	Address B	Port B	Packets	Bytes	Bits/s B → A	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s
10.0.0.21	49612	10.0.0.10	8080	51	72 k	16 k	0	0	51	72 k	0.923482	34.9141	
10.0.2.20	47726	10.0.0.10	8080	17	23 k	20 k	0	0	17	23 k	0.923868	9.1986	
10.0.2.20	47748	10.0.0.10	8080	1	360	—	1	360	0	0	10.976948	0.0000	
10.0.2.21	54248	10.0.0.10	8080	51	72 k	16 k	0	0	51	72 k	0.923713	34.9141	

Figure 4: Taxa bps real para a transmissão com 3 clientes

Para o encapsulamento, verificamos que este utiliza a nível da camada aplicacional o HTTP, da camada de transporte o TCP, a da camada de rede IPv4, da canda de ligação de dados o protocolo *Ethernet*. Este

encapsulamento é utilizado assim tanto para 1 cliente como para 2 e para 3. Verificamos o encapsulamento da análise da captura no *wireshark*, figura 5.

No.	Time	Source	Destination	Protocol	Length	Info
13	0.223371402	10.0.0.10	10.0.0.21	TCP	487	8080 → 49612 [PSH, ACK] Seq=8854 Ack=1 Win=509 Len=421 TSval=3058122622 TSecr=1431091111
14	0.223437783	10.0.0.21	10.0.0.10	TCP	66	49612 → 8080 [ACK] Seq=1 Ack=5958 Win=501 Len=0 TSval=1431091335 TSecr=3058122622
15	0.223439396	10.0.0.21	10.0.0.10	TCP	66	49612 → 8080 [ACK] Seq=1 Ack=7406 Win=496 Len=0 TSval=1431091335 TSecr=3058122622
16	0.223440087	10.0.0.21	10.0.0.10	TCP	66	49612 → 8080 [ACK] Seq=1 Ack=8854 Win=491 Len=0 TSval=1431091335 TSecr=3058122622
17	0.223440898	10.0.0.21	10.0.0.10	TCP	66	49612 → 8080 [ACK] Seq=1 Ack=9275 Win=489 Len=0 TSval=1431091335 TSecr=3058122622
18	0.485806646	10.0.0.10	10.0.0.21	TCP	1514	8080 → 49612 [ACK] Seq=9275 Ack=1 Win=509 Len=1448 TSval=3058122884 TSecr=1431091335
19	0.485801767	10.0.0.10	10.0.0.21	TCP	1514	8080 → 49612 [ACK] Seq=10723 Ack=1 Win=509 Len=1448 TSval=3058122884 TSecr=1431091335
20	0.485801988	10.0.0.10	10.0.0.21	TCP	1514	8080 → 49612 [ACK] Seq=12171 Ack=1 Win=509 Len=1448 TSval=3058122884 TSecr=1431091335
21	0.485802188	10.0.0.10	10.0.0.21	TCP	1514	8080 → 49612 [ACK] Seq=13619 Ack=1 Win=509 Len=1448 TSval=3058122884 TSecr=1431091335
22	0.485802388	10.0.0.10	10.0.0.21	TCP	1514	8080 → 49612 [PSH, ACK] Seq=15067 Ack=1 Win=509 Len=1448 TSval=3058122884 TSecr=1431091335
23	0.485806409	10.0.0.21	10.0.0.10	TCP	66	49612 → 8080 [ACK] Seq=1 Ack=10723 Win=501 Len=0 TSval=1431091597 TSecr=3058122884
24	0.485809064	10.0.0.21	10.0.0.10	TCP	66	49612 → 8080 [ACK] Seq=1 Ack=12171 Win=496 Len=0 TSval=1431091597 TSecr=3058122884
25	0.485809596	10.0.0.21	10.0.0.10	TCP	66	49612 → 8080 [ACK] Seq=1 Ack=13619 Win=491 Len=0 TSval=1431091597 TSecr=3058122884
26	0.4858091849	10.0.0.21	10.0.0.10	TCP	66	49612 → 8080 [ACK] Seq=1 Ack=15067 Win=485 Len=0 TSval=1431091597 TSecr=3058122884
27	0.4858093211	10.0.0.21	10.0.0.10	TCP	66	49612 → 8080 [ACK] Seq=1 Ack=16515 Win=479 Len=0 TSval=1431091597 TSecr=3058122884
28	0.485809296	10.0.0.10	10.0.0.21	TCP	817	8080 → 49612 [PSH, ACK] Seq=16515 Ack=1 Win=509 Len=751 TSval=3058122884 TSecr=1431091335
29	0.485915619	10.0.0.21	10.0.0.10	TCP	66	49612 → 8080 [ACK] Seq=1 Ack=17266 Win=476 Len=0 TSval=1431091597 TSecr=3058122884
30	0.731440310	10.0.0.10	10.0.0.21	TCP	1514	8080 → 49612 [ACK] Seq=17266 Ack=1 Win=509 Len=1448 TSval=3058123130 TSecr=1431091597
31	0.731447708	10.0.0.10	10.0.0.21	TCP	1514	8080 → 49612 [ACK] Seq=10714 Ack=1 Win=509 Len=1448 TSval=3058123130 TSecr=1431091597
32	0.731447818	10.0.0.10	10.0.0.21	TCP	1514	8080 → 49612 [ACK] Seq=20162 Ack=1 Win=509 Len=1448 TSval=3058123130 TSecr=1431091597
33	0.731447918	10.0.0.10	10.0.0.21	TCP	138	8080 → 49612 [PSH, ACK] Seq=21610 Ack=1 Win=509 Len=72 TSval=3058123130 TSecr=1431091597
34	0.731526410	10.0.0.21	10.0.0.10	TCP	66	49612 → 8080 [ACK] Seq=1 Ack=18714 Win=501 Len=0 TSval=1431091843 TSecr=3058123130
35	0.731528253	10.0.0.21	10.0.0.10	TCP	66	49612 → 8080 [ACK] Seq=1 Ack=20162 Win=496 Len=0 TSval=1431091843 TSecr=3058123130
36	0.731529024	10.0.0.21	10.0.0.10	TCP	66	49612 → 8080 [ACK] Seq=1 Ack=21610 Win=491 Len=0 TSval=1431091843 TSecr=3058123130
37	0.731529756	10.0.0.21	10.0.0.10	TCP	66	49612 → 8080 [ACK] Seq=1 Ack=21682 Win=491 Len=0 TSval=1431091843 TSecr=3058123130
38	1.239419279	10.0.0.10	10.0.0.21	TCP	1514	8080 → 49612 [ACK] Seq=21682 Ack=1 Win=509 Len=1448 TSval=3058123638 TSecr=1431091843
39	1.239420140	10.0.0.10	10.0.0.21	TCP	1514	8080 → 49612 [ACK] Seq=23130 Ack=1 Win=509 Len=1448 TSval=3058123638 TSecr=1431091843
40	1.239420259	10.0.0.10	10.0.0.21	TCP	1514	8080 → 49612 [ACK] Seq=24578 Ack=1 Win=509 Len=1448 TSval=3058123638 TSecr=1431091843
41	1.239420359	10.0.0.10	10.0.0.21	TCP	471	8080 → 49612 [PSH, ACK] Seq=26026 Ack=1 Win=509 Len=495 TSval=3058123638 TSecr=1431091843
42	1.239438317	10.0.0.21	10.0.0.10	TCP	66	49612 → 8080 [ACK] Seq=1 Ack=23130 Win=501 Len=0 TSval=1431092351 TSecr=3058123638
43	1.239435590	10.0.0.21	10.0.0.10	TCP	66	49612 → 8080 [ACK] Seq=1 Ack=24578 Win=496 Len=0 TSval=1431092351 TSecr=3058123638
44	1.239436351	10.0.0.21	10.0.0.10	TCP	66	49612 → 8080 [ACK] Seq=1 Ack=26026 Win=491 Len=0 TSval=1431092351 TSecr=3058123638
45	1.239437123	10.0.0.21	10.0.0.10	TCP	66	49612 → 8080 [ACK] Seq=1 Ack=26431 Win=489 Len=0 TSval=1431092351 TSecr=3058123638
46	1.725982819	10.0.0.10	10.0.0.21	TCP	1514	8080 → 49612 [ACK] Seq=26431 Ack=1 Win=509 Len=1448 TSval=3058124124 TSecr=1431092351
47	1.725984682	10.0.0.10	10.0.0.21	TCP	1514	8080 → 49612 [ACK] Seq=27879 Ack=1 Win=509 Len=1448 TSval=3058124124 TSecr=1431092351
48	1.725984862	10.0.0.10	10.0.0.21	TCP	1428	8080 → 49612 [PSH, ACK] Seq=29327 Ack=1 Win=509 Len=1362 TSval=3058124124 TSecr=1431092351
49	1.726048499	10.0.0.21	10.0.0.10	TCP	66	49612 → 8080 [ACK] Seq=1 Ack=27879 Win=501 Len=0 TSval=1431092837 TSecr=3058124124

\* Frame 30: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits) on interface veth1.0.12, id 0  
 \* Ethernet II, Src: 00:00:00:aa:00:00 (00:00:00:aa:00:00), Dst: 00:00:00:aa:00:02 (00:00:00:aa:00:02)  
 \* Internet Protocol Version 4, Src: 10.0.0.10, Dst: 10.0.0.21  
 \* Transmission Control Protocol, Src Port: 8080, Dst Port: 49612, Seq: 17266, Ack: 1, Len: 1448  
 \* Hypertext Transfer Protocol

Figure 5: Encapsulamento da trama TCP

Relativamente ao número de fluxos gerados do protocolo TPC, pertencem ao mesmo fluxo se estes 4 parâmetros forem iguais:

- IP Origem
- IP Destino
- Porta Origem
- Porta Destino

Seria um bocado tedioso verificar todos os frames dos *Wireshark* e comparar um a um. O *Wireshark* tem funcionalidades que nos permitem visualizar a que fluxo pertence. Para tal adicionamos uma coluna que nos indica o índice do fluxo do protocolo tcp, através de *tcp.stream*.

Verifica-se assim que para um cliente apenas um fluxo é gerado, para dois clientes são gerados dois fluxos e como é de esperar, para três clientes serão gerados três fluxos. Como podemos verificar nas figuras

No.	Time	Source	Destination	Protocol	Length	Info
13	0.223371402	10.0.0.10	10.0.0.21	TCP	487	8080 → 49612 [PSH, ACK] Seq=8854 Ack=1 Win=509 Len=421 TSval=3058122622 TSecr=1431091111
14	0.223437783	10.0.0.21	10.0.0.10	TCP	66	49612 → 8080 [ACK] Seq=1 Ack=5958 Win=501 Len=0 TSval=1431091335 TSecr=3058122622
15	0.223439396	10.0.0.21	10.0.0.10	TCP	66	49612 → 8080 [ACK] Seq=1 Ack=7406 Win=496 Len=0 TSval=1431091335 TSecr=3058122622
16	0.223440087	10.0.0.21	10.0.0.10	TCP	66	49612 → 8080 [ACK] Seq=1 Ack=8854 Win=491 Len=0 TSval=1431091335 TSecr=3058122622
17	0.223440898	10.0.0.21	10.0.0.10	TCP	66	49612 → 8080 [ACK] Seq=1 Ack=9275 Win=489 Len=0 TSval=1431091335 TSecr=3058122622
18	0.485806646	10.0.0.10	10.0.0.21	TCP	1514	8080 → 49612 [ACK] Seq=9275 Ack=1 Win=509 Len=1448 TSval=3058122884 TSecr=1431091335
19	0.485801767	10.0.0.10	10.0.0.21	TCP	1514	8080 → 49612 [ACK] Seq=10723 Ack=1 Win=509 Len=1448 TSval=3058122884 TSecr=1431091335
20	0.485801988	10.0.0.10	10.0.0.21	TCP	1514	8080 → 49612 [ACK] Seq=12171 Ack=1 Win=509 Len=1448 TSval=3058122884 TSecr=1431091335
21	0.485802188	10.0.0.10	10.0.0.21	TCP	1514	8080 → 49612 [ACK] Seq=13619 Ack=1 Win=509 Len=1448 TSval=3058122884 TSecr=1431091335
22	0.485802388	10.0.0.10	10.0.0.21	TCP	1514	8080 → 49612 [PSH, ACK] Seq=15067 Ack=1 Win=509 Len=1448 TSval=3058122884 TSecr=1431091335
23	0.485806409	10.0.0.21	10.0.0.10	TCP	66	49612 → 8080 [ACK] Seq=1 Ack=10723 Win=501 Len=0 TSval=1431091597 TSecr=3058122884
24	0.485809064	10.0.0.21	10.0.0.10	TCP	66	49612 → 8080 [ACK] Seq=1 Ack=12171 Win=496 Len=0 TSval=1431091597 TSecr=3058122884
25	0.485809596	10.0.0.21	10.0.0.10	TCP	66	49612 → 8080 [ACK] Seq=1 Ack=13619 Win=491 Len=0 TSval=1431091597 TSecr=3058122884
26	0.4858091849	10.0.0.21	10.0.0.10	TCP	66	49612 → 8080 [ACK] Seq=1 Ack=15067 Win=485 Len=0 TSval=1431091597 TSecr=3058122884
27	0.4858093211	10.0.0.21	10.0.0.10	TCP	66	49612 → 8080 [ACK] Seq=1 Ack=16515 Win=479 Len=0 TSval=1431091597 TSecr=3058122884
28	0.485809296	10.0.0.10	10.0.0.21	TCP	817	8080 → 49612 [PSH, ACK] Seq=16515 Ack=1 Win=509 Len=751 TSval=3058122884 TSecr=1431091335
29	0.485915619	10.0.0.21	10.0.0.10	TCP	66	49612 → 8080 [ACK] Seq=1 Ack=17266 Win=476 Len=0 TSval=1431091597 TSecr=3058122884
30	0.731440310	10.0.0.10	10.0.0.21	TCP	1514	8080 → 49612 [ACK] Seq=17266 Ack=1 Win=509 Len=1448 TSval=3058123130 TSecr=1431091597
31	0.731447708	10.0.0.10	10.0.0.21	TCP	1514	8080 → 49612 [ACK] Seq=10714 Ack=1 Win=509 Len=1448 TSval=3058123130 TSecr=1431091597
32	0.731447818	10.0.0.10	10.0.0.21	TCP	1514	8080 → 49612 [ACK] Seq=20162 Ack=1 Win=509 Len=1448 TSval=3058123130 TSecr=1431091597
33	0.731447918	10.0.0.10	10.0.0.21	TCP	138	8080 → 49612 [PSH, ACK] Seq=21610 Ack=1 Win=509 Len=72 TSval=3058123130 TSecr=1431091597
34	0.731526410	10.0.0.21	10.0.0.10	TCP	66	49612 → 8080 [ACK] Seq=1 Ack=18714 Win=501 Len=0 TSval=1431091843 TSecr=3058123130
35	0.731528253	10.0.0.21	10.0.0.10	TCP	66	49612 → 8080 [ACK] Seq=1 Ack=20162 Win=496 Len=0 TSval=1431091843 TSecr=3058123130
36	0.731529024	10.0.0.21	10.0.0.10	TCP	66	49612 → 8080 [ACK] Seq=1 Ack=21610 Win=491 Len=0 TSval=1431091843 TSecr=3058123130
37	0.731529756	10.0.0.21	10.0.0.10	TCP	66	49612 → 8080 [ACK] Seq=1 Ack=21682 Win=491 Len=0 TSval=1431091843 TSecr=3058123130
38	1.239419279	10.0.0.10	10.0.0.21	TCP	1514	8080 → 49612 [ACK] Seq=21682 Ack=1 Win=509 Len=1448 TSval=3058123638 TSecr=1431091843
39	1.239420140	10.0.0.10	10.0.0.21	TCP	1514	8080 → 49612 [ACK] Seq=23130 Ack=1 Win=509 Len=1448 TSval=3058123638 TSecr=1431091843
40	1.239420259	10.0.0.10	10.0.0.21	TCP	1514	8080 → 49612 [ACK] Seq=24578 Ack=1 Win=509 Len=1448 TSval=3058123638 TSecr=1431091843
41	1.239420359	10.0.0.10	10.0.0.21	TCP	471	8080 → 49612 [PSH, ACK] Seq=26026 Ack=1 Win=509 Len=495 TSval=3058123638 TSecr=1431091843
42	1.239438317	10.0.0.21	10.0.0.10	TCP	66	49612 → 8080 [ACK] Seq=1 Ack=23130 Win=501 Len=0 TSval=1431092351 TSecr=3058123638
43	1.239435590	10.0.0.21	10.0.0.10	TCP	66	49612 → 8080 [ACK] Seq=1 Ack=24578 Win=496 Len=0 TSval=1431092351 TSecr=3058123638
44	1.239436351	10.0.0.21	10.0.0.10	TCP	66	49612 → 8080 [ACK] Seq=1 Ack=26026 Win=491 Len=0 TSval=1431092351 TSecr=3058123638
45	1.239437123	10.0.0.21	10.0.0.10	TCP	66	49612 → 8080 [ACK] Seq=1 Ack=26431 Win=489 Len=0 TSval=1431092351 TSecr=3058123638
46	1.725982819	10.0.0.10	10.0.0.21	TCP	1514	8080 → 49612 [ACK] Seq=26431 Ack=1 Win=509 Len=1448 TSval=3058124124 TSecr=1431092351
47	1.725984682	10.0.0.10	10.0.0.21	TCP	1514	8080 → 49612 [ACK] Seq=27879 Ack=1 Win=509 Len=1448 TSval=3058124124 TSecr=1431092351
48	1.725984862	10.0.0.10	10.0.0.21	TCP	1428	8080 → 49612 [PSH, ACK] Seq=29327 Ack=1 Win=509 Len=1362 TSval=3058124124 TSecr=1431092351
49	1.726048499	10.0.0.21	10.0.0.10	TCP	66	49612 → 8080 [ACK] Seq=1 Ack=27879 Win=501 Len=0 TSval=1431092837 TSecr=3058124124

\* Frame 30: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits) on interface veth1.0.12, id 0  
 \* Ethernet II, Src: 00:00:00:aa:00:00 (00:00:00:aa:00:00), Dst: 00:00:00:aa:00:02 (00:00:00:aa:00:02)  
 \* Internet Protocol Version 4, Src: 10.0.0.10, Dst: 10.0.0.21  
 \* Transmission Control Protocol, Src Port: 8080, Dst Port: 49612, Seq: 17266, Ack: 1, Len: 1448  
 \* Hypertext Transfer Protocol

Figure 6: Fluxos gerados com 1 cliente

http									
No.	Time	Source	Destination	Protocol	Length	Info	StreamNumb		
666	16.652138863	10.0.0.10	10.0.0.21	TCP	1514	8880 → 49612 [ACK] Seq=198849 Ack=1 Win=509 Len=1448 TSval=3058315551 TSecr=1431283778	0		
696	17.653655121	10.0.0.10	10.0.0.21	TCP	1514	8880 → 49612 [ACK] Seq=218567 Ack=1 Win=509 Len=1448 TSval=3058316553 TSecr=1431284519	0		
768	18.891841961	10.0.0.10	10.0.0.21	TCP	1514	8880 → 49612 [ACK] Seq=239657 Ack=1 Win=509 Len=1448 TSval=3058317791 TSecr=1431286260	0		
770	18.891843163	10.0.0.10	10.0.0.21	TCP	1514	8880 → 49612 [ACK] Seq=233553 Ack=1 Win=509 Len=1448 TSval=3058317791 TSecr=1431286260	0		
814	20.395219209	10.0.0.10	10.0.0.21	TCP	1514	8880 → 49612 [ACK] Seq=245984 Ack=1 Win=509 Len=1448 TSval=3058319295 TSecr=1431287151	0		
825	20.659732783	10.0.0.10	10.0.0.21	TCP	1514	8880 → 49612 [ACK] Seq=248737 Ack=1 Win=509 Len=1448 TSval=3058319559 TSecr=1431288008	0		
853	21.179538862	10.0.0.10	10.0.0.21	TCP	1514	8880 → 49612 [ACK] Seq=258253 Ack=1 Win=509 Len=1448 TSval=3058320070 TSecr=1431288518	0		
874	21.666589519	10.0.0.10	10.0.0.21	TCP	1514	8880 → 49612 [ACK] Seq=266296 Ack=1 Win=509 Len=1448 TSval=3058320566 TSecr=1431289763	0		
899	23.177814789	10.0.0.10	10.0.0.21	TCP	1514	8880 → 49612 [ACK] Seq=274686 Ack=1 Win=509 Len=1448 TSval=3058322077 TSecr=1431290502	0		
1015	24.658843884	10.0.0.10	10.0.0.21	TCP	1514	8880 → 49612 [ACK] Seq=301314 Ack=1 Win=509 Len=1448 TSval=3058323558 TSecr=1431292012	0		
1030	25.149211696	10.0.0.10	10.0.0.21	TCP	1514	8880 → 49612 [ACK] Seq=304282 Ack=1 Win=509 Len=1448 TSval=3058324049 TSecr=1431292271	0		
1032	25.149213481	10.0.0.10	10.0.0.21	TCP	1514	8880 → 49612 [ACK] Seq=307178 Ack=1 Win=509 Len=1448 TSval=3058324049 TSecr=1431292271	0		
1059	26.170550705	10.0.0.10	10.0.0.21	TCP	1514	8880 → 49612 [ACK] Seq=313289 Ack=1 Win=509 Len=1448 TSval=3058325070 TSecr=1431293270	0		
1072	26.394394363	10.0.0.10	10.0.0.21	TCP	1514	8880 → 49612 [ACK] Seq=318081 Ack=1 Win=509 Len=1448 TSval=3058325294 TSecr=1431293763	0		
1073	26.394394583	10.0.0.10	10.0.0.21	TCP	1477	8880 → 49612 [PSH, ACK] Seq=320329 Ack=1 Win=509 Len=1411 TSval=3058325294 TSecr=1431293783	0		
1099	27.169175664	10.0.0.10	10.0.0.21	TCP	1514	8880 → 49612 [ACK] Seq=326318 Ack=1 Win=509 Len=1448 TSval=3058326069 TSecr=1431294293	0		
1137	27.656054026	10.0.0.10	10.0.0.21	TCP	1161	8880 → 49612 [PSH, ACK] Seq=342301 Ack=1 Win=509 Len=1095 TSval=3058326558 TSecr=1431295904[Malformed Packet]	0		
1205	29.871980773	10.0.0.10	10.0.0.21	TCP	556	8880 → 49612 [PSH, ACK] Seq=362381 Ack=1 Win=509 Len=490 TSval=3058328771 TSecr=1431297080	0		
1220	30.358958689	10.0.0.10	10.0.0.21	TCP	1514	8880 → 49612 [ACK] Seq=365687 Ack=1 Win=509 Len=1448 TSval=3058329258 TSecr=1431297484	0		
1234	30.621622096	10.0.0.10	10.0.0.21	TCP	1514	8880 → 49612 [ACK] Seq=367141 Ack=1 Win=509 Len=1448 TSval=3058329521 TSecr=1431297971	0		
1250	30.884344434	10.0.0.10	10.0.0.21	TCP	1514	8880 → 49612 [ACK] Seq=371767 Ack=1 Win=509 Len=1448 TSval=3058329784 TSecr=1431298234	0		
1282	31.61782465	10.0.0.10	10.0.0.21	TCP	1514	8880 → 49612 [ACK] Seq=380800 Ack=1 Win=509 Len=1448 TSval=3058330517 TSecr=1431298985	0		
1292	31.617670791	10.0.0.10	10.0.0.21	TCP	880	8880 → 49612 [PSH, ACK] Seq=380840 Ack=1 Win=509 Len=814 TSval=3058330517 TSecr=1431298985	0		
1324	32.371024633	10.0.0.10	10.0.0.21	TCP	1514	8880 → 49612 [ACK] Seq=394723 Ack=1 Win=509 Len=1448 TSval=3058331270 TSecr=1431299475	0		
1325	32.371024753	10.0.0.10	10.0.0.21	TCP	1353	8880 → 49612 [PSH, ACK] Seq=396171 Ack=1 Win=509 Len=1287 TSval=3058331270 TSecr=1431299475	0		
1351	33.374006889	10.0.0.10	10.0.0.21	TCP	1514	8880 → 49612 [ACK] Seq=402124 Ack=1 Win=509 Len=1448 TSval=3058332273 TSecr=1431300476	0		
1382	34.367837701	10.0.0.10	10.0.0.21	TCP	1514	8880 → 49612 [ACK] Seq=413632 Ack=1 Win=509 Len=1448 TSval=3058333267 TSecr=1431301230	0		
1452	35.614095144	10.0.0.10	10.0.0.21	TCP	1514	8880 → 49612 [ACK] Seq=433932 Ack=1 Win=509 Len=1448 TSval=3058334513 TSecr=1431302980	0		
1454	35.614094666	10.0.0.10	10.0.0.21	TCP	1514	8880 → 49612 [ACK] Seq=436828 Ack=1 Win=509 Len=1448 TSval=3058334513 TSecr=1431302980	0		
23	8.727834032	10.0.0.10	10.0.2.20	TCP	1514	8880 → 47694 [ACK] Seq=7082 Ack=1 Win=507 Len=1448 TSval=226393150 TSecr=14411331	1		
94	1.996405259	10.0.0.10	10.0.2.20	TCP	1514	8880 → 47694 [ACK] Seq=77382 Ack=1 Win=507 Len=1448 TSval=226394418 TSecr=41413962	1		
96	1.996405600	10.0.0.10	10.0.2.20	TCP	1514	8880 → 47694 [ACK] Seq=80278 Ack=1 Win=507 Len=1448 TSval=226394418 TSecr=41413962	2		
139	3.485956726	10.0.0.10	10.0.2.20	TCP	1514	8880 → 47694 [ACK] Seq=42709 Ack=2 Win=507 Len=1448 TSval=226395907 TSecr=41414397	1		
156	3.953473771	10.0.0.10	10.0.0.10	HTTP	365	GET / HTTP/1.1	2		
321	7.940079406	10.0.0.10	10.0.0.10	HTTP	365	GET / HTTP/1.1	3		
627	15.637413117	10.0.0.10	10.0.0.10	HTTP	365	GET / HTTP/1.1	4		
992	24.455415658	10.0.0.10	10.0.0.10	HTTP	360	GET / HTTP/1.1	5		

Figure 7: Fluxos gerados com 2 clientes

http									
No.	Time	Source	Destination	Protocol	Length	Info	StreamNumber		
21	0.923482474	10.0.0.10	10.0.0.21	TCP	1514	8880 → 49612 [ACK] Seq=5640 Ack=1 Win=509 Len=1448 TSval=3058405583 TSecr=1431374054	0		
29	0.923713239	10.0.0.10	10.0.2.21	TCP	1514	8880 → 54248 [ACK] Seq=5640 Ack=1 Win=509 Len=1448 TSval=2064831709 TSecr=2442574458	1		
37	0.923807647	10.0.0.10	10.0.2.20	TCP	1514	8880 → 47726 [ACK] Seq=5640 Ack=1 Win=507 Len=1448 TSval=2264680929 TSecr=414517586	0		
155	2.434609126	10.0.0.10	10.0.0.21	TCP	1514	8880 → 49612 [ACK] Seq=32268 Ack=1 Win=509 Len=1448 TSval=3058407094 TSecr=1431375542	0		
163	2.434717143	10.0.0.10	10.0.2.21	TCP	1514	8880 → 54248 [ACK] Seq=32268 Ack=1 Win=509 Len=1448 TSval=2064833220 TSecr=2442575938	1		
171	2.434824118	10.0.0.10	10.0.2.20	TCP	1514	8880 → 47726 [ACK] Seq=32268 Ack=1 Win=507 Len=1448 TSval=226508431 TSecr=414519074	0		
178	2.927180475	10.0.0.10	10.0.0.21	TCP	1514	8880 → 49612 [ACK] Seq=35236 Ack=1 Win=509 Len=1448 TSval=3058407587 TSecr=1431375808	0		
180	2.927181527	10.0.0.10	10.0.0.21	TCP	1514	8880 → 49612 [ACK] Seq=38132 Ack=1 Win=509 Len=1448 TSval=3058407587 TSecr=1431375808	0		
186	2.927268821	10.0.0.10	10.0.2.21	TCP	1514	8880 → 54248 [ACK] Seq=35236 Ack=1 Win=509 Len=1448 TSval=2064833712 TSecr=2442576204	1		
188	2.927269122	10.0.0.10	10.0.2.21	TCP	1514	8880 → 54248 [ACK] Seq=38132 Ack=1 Win=509 Len=1448 TSval=2064833712 TSecr=2442576204	0		
194	2.927367073	10.0.0.10	10.0.2.20	TCP	1514	8880 → 47726 [ACK] Seq=35236 Ack=1 Win=507 Len=1448 TSval=2265080923 TSecr=414519348	2		
196	2.927367363	10.0.0.10	10.0.2.20	TCP	1514	8880 → 47726 [ACK] Seq=38132 Ack=1 Win=507 Len=1448 TSval=2265080923 TSecr=414519348	0		
220	3.924809742	10.0.0.10	10.0.0.21	TCP	1514	8880 → 49612 [ACK] Seq=44243 Ack=1 Win=509 Len=1448 TSval=3058408585 TSecr=1431376810	1		
226	3.925038494	10.0.0.10	10.0.2.21	TCP	1514	8880 → 54248 [ACK] Seq=44243 Ack=1 Win=509 Len=1448 TSval=2064834710 TSecr=2442577206	1		
232	3.925155805	10.0.0.10	10.0.2.20	TCP	1514	8880 → 47726 [ACK] Seq=44243 Ack=1 Win=507 Len=1448 TSval=2265091921 TSecr=414520342	0		
240	4.173606897	10.0.0.10	10.0.0.21	TCP	1514	8880 → 49612 [ACK] Seq=49835 Ack=1 Win=509 Len=1448 TSval=3058408833 TSecr=1431377298	0		
241	4.173606968	10.0.0.10	10.0.0.21	TCP	1477	8880 → 49612 [PSH, ACK] Seq=51283 Ack=1 Win=509 Len=1411 TSval=3058408833 TSecr=1431377298	0		
246	4.173696045	10.0.0.10	10.0.2.21	TCP	1514	8880 → 54248 [ACK] Seq=49835 Ack=1 Win=509 Len=1448 TSval=2064834959 TSecr=2442577694	1		
247	4.173696155	10.0.0.10	10.0.2.21	TCP	1477	8880 → 54248 [PSH, ACK] Seq=51283 Ack=1 Win=509 Len=1411 TSval=2064834959 TSecr=2442577694	0		
252	4.173792553	10.0.0.10	10.0.2.20	TCP	1514	8880 → 47726 [ACK] Seq=49835 Ack=1 Win=507 Len=1448 TSval=226502170 TSecr=414520830	2		
253	4.173792363	10.0.0.10	10.0.2.20	TCP	1477	8880 → 47726 [PSH, ACK] Seq=51283 Ack=1 Win=507 Len=1411 TSval=226502170 TSecr=414520830	0		
281	4.939580096	10.0.0.10	10.0.0.21	TCP	1514	8880 → 49612 [ACK] Seq=57272 Ack=1 Win=509 Len=1448 TSval=3058409590 TSecr=1431377795	0		
289	4.939892985	10.0.0.10	10.0.2.21	TCP	1514	8880 → 54248 [ACK] Seq=57272 Ack=1 Win=509 Len=1448 TSval=2064839716 TSecr=2442578191	1		
297	4.939772793	10.0.0.10	10.0.2.20	TCP	1514	8880 → 47726 [ACK] Seq=57272 Ack=1 Win=507 Len=1448 TSval=226502927 TSecr=414521327	0		
333	5.428042041	10.0.0.10	10.0.0.21	TCP	1161	8880 → 49612 [PSH, ACK] Seq=73255 Ack=1 Win=509 Len=1095 TSval=3058410080 TSecr=1431378547[Malformed Packet]	0		
345	5.428161686	10.0.0.10	10.0.2.21	TCP	1161	8880 → 54248 [PSH, ACK] Seq=73255 Ack=1 Win=509 Len=1095 TSval=2064836285 TSecr=2442578943[Malformed Packet]	1		
357	5.428261149	10.0.0.10	10.0.2.20	TCP	1161	8880 → 47726 [PSH, ACK] Seq=73255 Ack=1 Win=507 Len=1095 TSval=226503416 TSecr=414522079[Malformed Packet]	2		
435	7.626454361	10.0.0.10	10.0.0.21	TCP	556	8880 → 49612 [PSH, ACK] Seq=93255 Ack=1 Win=509 Len=490 TSval=3058412280 TSecr=1431380605	0		
443	7.626956122	10.0.0.10	10.0.2.21	TCP	556	8880 → 54248 [PSH, ACK] Seq=93255 Ack=1 Win=509 Len=490 TSval=2064838406 TSecr=2442581001	1		
451	7.626952100	10.0.0.10	10.0.2.20	TCP	556	8880 → 47726 [PSH, ACK] Seq=93255 Ack=1 Win=507 Len=490 TSval=226509561 TSecr=414524137	2		
459	8.128639725	10.0.0.10	10.0.0.21	TCP	1514	8880 → 49612 [ACK] Seq=96641 Ack=1 Win=509 Len=1448 TSval=3058412788 TSecr=1431380893	0		
467	8.128211430	10.0.0.10	10.0.2.21	TCP	1514	8880 → 54248 [ACK] Seq=96641 Ack=1 Win=509 Len=1448 TSval=2064838913 TSecr=2442581390	1		
476	8.128336192	10.0.0.10	10.0.2.20	TCP	1514	8880 → 47726 [ACK] Seq=96641 Ack=1 Win=507 Len=1448 TSval=226506124 TSecr=414524526	0		
481	8.373415607	10.0.0.10	10.0.0.21	TCP	1514	8880 → 49612 [ACK] Seq=89895 Ack=1 Win=509 Len=1448 TSval=3058413033 TSecr=1431381501	0		
489	8.373671532	10.0.0.10	10.0.0.21	TCP	1514	8880 → 49612 [ACK] Seq=89895 Ack=1 Win=509 Len=1448 TSval=3058431959 TSecr=1431381897	0		
497	8.373839367	10.0.0.10	10.0.2.20	TCP	1514	8880 → 47726 [ACK] Seq=89895 Ack=1 Win=507 Len=1448 TSval=226506370 TSecr=414525633	0		
514	9.026565303	10.0.0.10	10.0.0.21	TCP	1514	8880 → 49612 [ACK] Seq=97722 Ack=1 Win=509 Len=1448 TSval=3058413300 TSecr=1431381981	0		
513	8.467255411	10.0.0.10	10.0.2.21	TCP	1514	8880 → 54248 [ACK] Seq=97722 Ack=1 Win=509 Len=1448 TSval=2064839428 TSecr=2442582143	1		

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

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 informação sobre as diversas resoluções do vídeo está descrito no ficheiro "video\_manifest.mpd". Assim sendo, para se conseguir assistir ao vídeo de menor resolução, 160x100, é necessário uma largura de banda de "115851 bps", tal como é possível ver na imagem que se segue.

```
<Representation id="1" mimeType="video/mp4" codecs="avc3.64000c" width="160" height="100" frameRate="30000/1001" sar="1:1" startWithSAP="0" bandwidth="115851">
  <BaseURL>videoB_160_100_200k_dash.mp4</BaseURL>
  <SegmentList timescale="30000" duration="15000">
    <SegmentUFL mediaRange="926-5219" indexRange="926-969"/>
    <SegmentUFL mediaRange="5220-8328" indexRange="5220-5263"/>
    <SegmentUFL mediaRange="8329-11020" indexRange="8329-8372"/>
    <SegmentUFL mediaRange="11021-18262" indexRange="11021-11064"/>
    <SegmentUFL mediaRange="18263-23920" indexRange="18263-18306"/>
    <SegmentUFL mediaRange="23921-30370" indexRange="23921-23964"/>
    <SegmentUFL mediaRange="30371-41690" indexRange="30371-30414"/>
    <SegmentUFL mediaRange="41691-49600" indexRange="41691-41734"/>
    <SegmentUFL mediaRange="49601-56586" indexRange="49601-49644"/>
    <SegmentUFL mediaRange="56587-63463" indexRange="56587-56630"/>
    <SegmentUFL mediaRange="63464-77212" indexRange="63464-63507"/>
    <SegmentUFL mediaRange="77213-86160" indexRange="77213-77256"/>
    <SegmentUFL mediaRange="86161-94502" indexRange="86161-86204"/>
    <SegmentUFL mediaRange="94503-107411" indexRange="94503-94546"/>
    <SegmentUFL mediaRange="107412-114866" indexRange="107412-107455"/>
    <SegmentUFL mediaRange="114867-122707" indexRange="114867-114910"/>
    <SegmentUFL mediaRange="122708-134790" indexRange="122708-122751"/>
    <SegmentUFL mediaRange="134791-143070" indexRange="134791-134834"/>
    <SegmentUFL mediaRange="143071-149926" indexRange="143071-143114"/>
    <SegmentUFL mediaRange="149927-157757" indexRange="149927-149970"/>
    <SegmentUFL mediaRange="157758-167096" indexRange="157758-157801"/>
    <SegmentUFL mediaRange="167097-174382" indexRange="167097-167140"/>
    <SegmentUFL mediaRange="174383-178379" indexRange="174383-174426"/>
    <SegmentUFL mediaRange="178380-191588" indexRange="178380-178423"/>
    <SegmentUFL mediaRange="191589-199127" indexRange="191589-191632"/>
    <SegmentUFL mediaRange="199128-207155" indexRange="199128-199171"/>
    <SegmentUFL mediaRange="207156-218610" indexRange="207156-207199"/>
    <SegmentUFL mediaRange="218611-226485" indexRange="218611-218654"/>
    <SegmentUFL mediaRange="226486-234274" indexRange="226486-226529"/>
    <SegmentUFL mediaRange="234275-242838" indexRange="234275-234318"/>
    <SegmentUFL mediaRange="242839-255443" indexRange="242839-242882"/>
    <SegmentUFL mediaRange="255444-263162" indexRange="255444-255487"/>
    <SegmentUFL mediaRange="263163-272649" indexRange="263163-263206"/>
    <SegmentUFL mediaRange="272650-285391" indexRange="272650-272693"/>
    <SegmentUFL mediaRange="285392-294738" indexRange="285392-285435"/>
    <SegmentUFL mediaRange="294739-303473" indexRange="294739-294782"/>
    <SegmentUFL mediaRange="303474-315807" indexRange="303474-303517"/>
    <SegmentUFL mediaRange="315808-319800" indexRange="315808-315851"/>
    <SegmentUFL mediaRange="319801-325266" indexRange="319801-319844"/>
    <SegmentUFL mediaRange="325267-329563" indexRange="325267-325310"/>
    <SegmentUFL mediaRange="329564-338831" indexRange="329564-329607"/>
    <SegmentUFL mediaRange="338832-341824" indexRange="338832-338875"/>
    <SegmentUFL mediaRange="341825-343132" indexRange="341825-341868"/>
    <SegmentUFL mediaRange="343133-349722" indexRange="343133-343176"/>
    <SegmentUFL mediaRange="349723-354191" indexRange="349723-349766"/>
    <SegmentUFL mediaRange="354192-358077" indexRange="354192-354235"/>
    <SegmentUFL mediaRange="358078-364114" indexRange="358078-358121"/>
    <SegmentUFL mediaRange="364115-364797" indexRange="364115-364158"/>
    <SegmentUFL mediaRange="364798-365492" indexRange="364798-364841"/>
    <SegmentUFL mediaRange="365493-367490" indexRange="365493-365536"/>
    <SegmentUFL mediaRange="367491-376595" indexRange="367491-367534"/>
    <SegmentUFL mediaRange="376596-380735" indexRange="376596-376639"/>
    <SegmentUFL mediaRange="380736-382207" indexRange="380736-380779"/>
  </SegmentList>
</Representation>
```

Figure 9: Largura de banda no vídeo de menor resolução



Para se conseguir assistir ao vídeo de resolução intermédia, 320x200, é necessário uma largura de banda de "269501 bps", como é possível ver na imagem a seguir.

```
<Representation id="2" mimeType="video/mp4" codecs="avc3.640014" width="320" height="200" frameRate="30000/1001" sar="1:1" startWithSAP="0" bandwidth="269501">
<BaseURL>videoB_320_200_500k_dash.mp4</BaseURL>
<SegmentList timescale="30000" duration="15000">
<SegmentURL mediaRange="926-10223" indexRange="926-969"/>
<SegmentURL mediaRange="10224-17957" indexRange="10224-10267"/>
<SegmentURL mediaRange="17958-23556" indexRange="17958-18001"/>
<SegmentURL mediaRange="23557-39955" indexRange="23557-23600"/>
<SegmentURL mediaRange="39956-53730" indexRange="39956-39999"/>
<SegmentURL mediaRange="53731-71611" indexRange="53731-53774"/>
<SegmentURL mediaRange="71612-99507" indexRange="71612-71655"/>
<SegmentURL mediaRange="99508-118346" indexRange="99508-99551"/>
<SegmentURL mediaRange="118347-134519" indexRange="118347-118390"/>
<SegmentURL mediaRange="134520-152333" indexRange="134520-134563"/>
<SegmentURL mediaRange="152334-185331" indexRange="152334-152377"/>
<SegmentURL mediaRange="185332-205507" indexRange="185332-185375"/>
<SegmentURL mediaRange="205508-223679" indexRange="205508-205551"/>
<SegmentURL mediaRange="223680-254316" indexRange="223680-223723"/>
<SegmentURL mediaRange="254317-271907" indexRange="254317-254360"/>
<SegmentURL mediaRange="271908-290410" indexRange="271908-271951"/>
<SegmentURL mediaRange="290411-318625" indexRange="290411-290454"/>
<SegmentURL mediaRange="318626-337056" indexRange="318626-318669"/>
<SegmentURL mediaRange="337057-353422" indexRange="337057-337100"/>
<SegmentURL mediaRange="353423-372675" indexRange="353423-353466"/>
<SegmentURL mediaRange="372676-394491" indexRange="372676-372719"/>
<SegmentURL mediaRange="394492-412666" indexRange="394492-394535"/>
<SegmentURL mediaRange="412667-422137" indexRange="412667-412710"/>
<SegmentURL mediaRange="422138-452312" indexRange="422138-422181"/>
<SegmentURL mediaRange="452313-469917" indexRange="452313-452356"/>
<SegmentURL mediaRange="469918-487910" indexRange="469918-469961"/>
<SegmentURL mediaRange="487911-514891" indexRange="487911-487954"/>
<SegmentURL mediaRange="514892-534212" indexRange="514892-514935"/>
<SegmentURL mediaRange="534213-552602" indexRange="534213-534256"/>
<SegmentURL mediaRange="552603-570618" indexRange="552603-552646"/>
<SegmentURL mediaRange="570619-599775" indexRange="570619-570662"/>
<SegmentURL mediaRange="599776-620229" indexRange="599776-599819"/>
<SegmentURL mediaRange="620230-640017" indexRange="620230-620273"/>
<SegmentURL mediaRange="640018-670122" indexRange="640018-640061"/>
<SegmentURL mediaRange="670123-692645" indexRange="670123-670166"/>
<SegmentURL mediaRange="692646-712834" indexRange="692646-692689"/>
<SegmentURL mediaRange="712835-742114" indexRange="712835-712878"/>
<SegmentURL mediaRange="742115-751355" indexRange="742115-742158"/>
<SegmentURL mediaRange="751356-761092" indexRange="751356-751399"/>
<SegmentURL mediaRange="761093-772044" indexRange="761093-761136"/>
<SegmentURL mediaRange="772045-793383" indexRange="772045-772088"/>
<SegmentURL mediaRange="793384-799353" indexRange="793384-793427"/>
<SegmentURL mediaRange="799354-801932" indexRange="799354-799397"/>
<SegmentURL mediaRange="801933-817488" indexRange="801933-801976"/>
<SegmentURL mediaRange="817489-827931" indexRange="817489-817532"/>
<SegmentURL mediaRange="827932-836690" indexRange="827932-827975"/>
<SegmentURL mediaRange="836691-850586" indexRange="836691-836734"/>
<SegmentURL mediaRange="850587-851685" indexRange="850587-850630"/>
<SegmentURL mediaRange="851686-852934" indexRange="851686-851729"/>
<SegmentURL mediaRange="852935-856877" indexRange="852935-852978"/>
<SegmentURL mediaRange="856878-878709" indexRange="856878-856921"/>
<SegmentURL mediaRange="878710-886358" indexRange="878710-878753"/>
<SegmentURL mediaRange="886359-889120" indexRange="886359-886402"/>
</SegmentList>
</Representation>
```

Figure 10: Largura de banda no vídeo resolução intermédia

Por fim, para se conseguir assistir ao vídeo de maior resolução, 640x400, é necessário uma largura de banda de "680526 bps", tal como é possível ver na imagem que se segue.

```
<Representation id="3" mimeType="video/mp4" codecs="avc3.64001e" width="640" height="400" frameRate="30000/1001" sar="1:1" startWithSAP="0" bandwidth="680526">
<BaseURL>videoB_640_400_1000k_dash.mp4</BaseURL>
<SegmentList timescale="30000" duration="15000">
<SegmentURL mediaRange="926-20837" indexRange="926-969"/>
<SegmentURL mediaRange="20838-40620" indexRange="20838-20881"/>
<SegmentURL mediaRange="40621-55518" indexRange="40621-40664"/>
<SegmentURL mediaRange="55519-94233" indexRange="55519-55562"/>
<SegmentURL mediaRange="94234-128148" indexRange="94234-94277"/>
<SegmentURL mediaRange="128149-168714" indexRange="128149-128192"/>
<SegmentURL mediaRange="168715-237001" indexRange="168715-168758"/>
<SegmentURL mediaRange="237002-281827" indexRange="237002-237045"/>
<SegmentURL mediaRange="281828-322801" indexRange="281828-281871"/>
<SegmentURL mediaRange="322802-368639" indexRange="322802-322845"/>
<SegmentURL mediaRange="368640-450099" indexRange="368640-368683"/>
<SegmentURL mediaRange="450100-501015" indexRange="450100-450143"/>
<SegmentURL mediaRange="501016-549200" indexRange="501016-501059"/>
<SegmentURL mediaRange="549201-624353" indexRange="549201-549244"/>
<SegmentURL mediaRange="624354-669773" indexRange="624354-624397"/>
<SegmentURL mediaRange="669774-716684" indexRange="669774-669817"/>
<SegmentURL mediaRange="716685-785086" indexRange="716685-716728"/>
<SegmentURL mediaRange="785087-836138" indexRange="785087-785130"/>
<SegmentURL mediaRange="836139-876067" indexRange="836139-836182"/>
<SegmentURL mediaRange="876068-925020" indexRange="876068-876111"/>
<SegmentURL mediaRange="925021-979617" indexRange="925021-925064"/>
<SegmentURL mediaRange="979618-1024803" indexRange="979618-979661"/>
<SegmentURL mediaRange="1024804-1048977" indexRange="1024804-1024847"/>
<SegmentURL mediaRange="1048978-1127101" indexRange="1048978-1049021"/>
<SegmentURL mediaRange="1127102-1171614" indexRange="1127102-1127145"/>
<SegmentURL mediaRange="1171615-1217725" indexRange="1171615-1171658"/>
<SegmentURL mediaRange="1217726-1287540" indexRange="1217726-1217769"/>
<SegmentURL mediaRange="1287541-1337170" indexRange="1287541-1287584"/>
<SegmentURL mediaRange="1337171-1384403" indexRange="1337171-1337214"/>
<SegmentURL mediaRange="1384404-1431619" indexRange="1384404-1384447"/>
<SegmentURL mediaRange="1431620-1509848" indexRange="1431620-1431663"/>
<SegmentURL mediaRange="1509849-1509889" indexRange="1509849-1509892"/>
<SegmentURL mediaRange="1509890-1613167" indexRange="1509890-1509933"/>
<SegmentURL mediaRange="1613168-1689764" indexRange="1613168-1613211"/>
<SegmentURL mediaRange="1689765-1749138" indexRange="1689765-1689808"/>
<SegmentURL mediaRange="1749139-1799557" indexRange="1749139-1749182"/>
<SegmentURL mediaRange="1799558-1874228" indexRange="1799558-1799601"/>
<SegmentURL mediaRange="1874229-1898980" indexRange="1874229-1874272"/>
<SegmentURL mediaRange="1898981-1926453" indexRange="1898981-1899024"/>
<SegmentURL mediaRange="1926454-1951904" indexRange="1926454-1926497"/>
<SegmentURL mediaRange="1951905-2006566" indexRange="1951905-1951948"/>
<SegmentURL mediaRange="2006567-2019778" indexRange="2006567-2006610"/>
<SegmentURL mediaRange="2019779-2025290" indexRange="2019779-2019822"/>
<SegmentURL mediaRange="2025291-2064016" indexRange="2025291-2025334"/>
<SegmentURL mediaRange="2064017-2090736" indexRange="2064017-2064060"/>
<SegmentURL mediaRange="2090737-2114341" indexRange="2090737-2090780"/>
<SegmentURL mediaRange="2114342-2147874" indexRange="2114342-2114385"/>
<SegmentURL mediaRange="2147875-2149623" indexRange="2147875-2147918"/>
<SegmentURL mediaRange="2149624-2152024" indexRange="2149624-2149667"/>
<SegmentURL mediaRange="2152025-2161404" indexRange="2152025-2152068"/>
<SegmentURL mediaRange="2161405-2213421" indexRange="2161405-2161538"/>
<SegmentURL mediaRange="2213422-2238178" indexRange="2213422-2213465"/>
<SegmentURL mediaRange="2238179-2245143" indexRange="2238179-2238222"/>
</SegmentList>
```

Figure 11: Largura de banda no vídeo de maior resolução

No entanto, através do *wireshark* deparamo-nos que os valores diferem um pouco assim sendo:

- Para o vídeo de menor resolução a largura de banda necessária é de 167k, como podemos ver na figura que se segue

Ethernet - 1	IPv4 - 1	IPv6	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
10.0.2.20	60990	10.0.0.10	9999						474	414 k	204	13 k	270	400 k	8.355456	19.0605	5796	167 k

Figure 12: Largura de banda no vídeo de menor resolução pelo wireshark

- Para o vídeo resolução intermédia a largura de banda necessária é de 384k, como podemos ver na figura que se segue

Ethernet - 1	IPv4 - 1	IPv6	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
10.0.2.20	46606	10.0.0.10	9999						1,064	959 k	446	29 k	618	930 k	8.728857	19.3740	12 k	384 k

Figure 13: Largura de banda no vídeo resolução intermédia pelo wireshark

- Para o vídeo de maior resolução a largura de banda necessária é de 996k, como podemos ver na figura que se segue

Ethernet - 1	IPv4 - 1	IPv6	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
10.0.2.20	42198	10.0.0.10	9999						2,470	2402 k	920	61 k	1,550	2341 k	4.085434	18.7939	25 k	996 k

Figure 14: Largura de banda no vídeo de maior resolução pelo wireshark

O aumento da largura de banda visto no Wireshark deve-se *overhead* de transmissão.

Os protocolos usados para a transmissão do vídeo são o protocolo *Ethernet*, o protocolo IP, o protocolo TCP e o protocolo HTTP, como se pode ver na figura seguinte. Posto isto, todos estes protocolos estão associados e dão suporte à rede global Internet, uma vez que o vídeo foi transmitido na *WEB* a partir do *firefox*.

5	2.017537465	10.0.2.20	10.0.0.10	TCP	66	39102 → 9999 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=16174934...
6	2.018034957	10.0.2.20	10.0.0.10	HTTP	477	GET /video_dash.html HTTP/1.1
7	2.018169555	10.0.0.10	10.0.2.20	TCP	66	9999 → 39102 [ACK] Seq=1 Ack=412 Win=64768 Len=0 TSval=287180...
8	2.031338773	10.0.0.10	10.0.2.20	HTTP	272	HTTP/1.1 304 Not Modified
9	2.031551252	10.0.2.20	10.0.0.10	TCP	66	39102 → 9999 [FIN, ACK] Seq=412 Ack=208 Win=64128 Len=0 TSval=...
10	2.031621475	10.0.0.10	10.0.2.20	TCP	66	9999 → 39102 [ACK] Seq=208 Ack=413 Win=64768 Len=0 TSval=2871...
11	2.700905786	10.0.2.20	10.0.0.10	TCP	74	39112 → 9999 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1...
12	2.700924505	10.0.0.10	10.0.2.20	TCP	74	9999 → 39112 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 ...
13	2.700940214	10.0.2.20	10.0.0.10	TCP	66	39112 → 9999 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=16174941...
14	2.701375493	10.0.2.20	10.0.0.10	HTTP	401	GET /videoB_640_400_1000k_dash.mp4 HTTP/1.1
15	2.701381149	10.0.0.10	10.0.2.20	TCP	66	9999 → 39112 [ACK] Seq=1 Ack=336 Win=64896 Len=0 TSval=287180...
16	2.701643459	10.0.0.10	10.0.2.20	TCP	1514	9999 → 39112 [ACK] Seq=1 Ack=336 Win=64896 Len=1448 TSval=287...

▶ Frame 6: 477 bytes on wire (3816 bits), 477 bytes captured (3816 bits) on interface veth5.0.63, id 0  
 ▶ Ethernet II, Src: 00:00:00\_aa:00:03 (00:00:00:aa:00:03), Dst: 00:00:00\_aa:00:00 (00:00:00:aa:00:00)  
 ▶ Internet Protocol Version 4, Src: 10.0.2.20, Dst: 10.0.0.10  
 ▶ Transmission Control Protocol, Src Port: 39102, Dst Port: 9999, Seq: 1, Ack: 1, Len: 411  
 ▶ Hypertext Transfer Protocol

Figure 15: Camada Protocolar



**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.**

Numa primeira fase, tivemos de alterar a topologia elaborada na etapa 1, relativamente aos *links*, consoante a largura de banda necessária para transmitir os diferentes vídeos que se encontrava no ficheiro "video\_manifest.mpd", já referido na pergunta anterior. Assim sendo, a topologia resultante é:

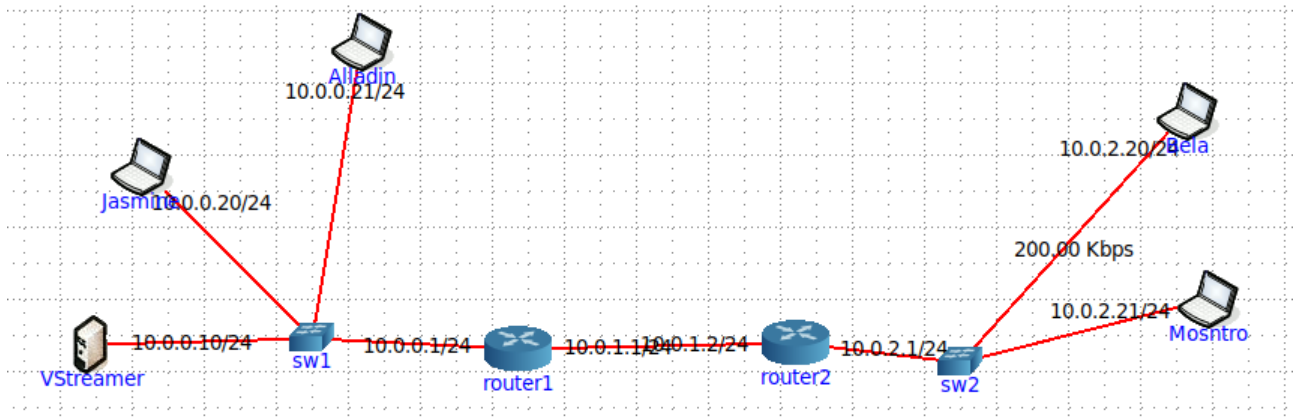


Figure 16: Topologia alterada

Como visto na pergunta anterior, a largura de banda necessária para a transmissão do vídeo de menor resolução é de 167k, por isso, ajustamos o link na Bela para 200kbps e mantivemos o link do Alladin *unlimited* para que fosse possível transmitir o vídeo de maior resolução. Pelo o Wireshark, conseguimos ver a transmissão dos vídeos

No.	Time	Source	Destination	Protocol	Length	Info
7	4.145780155	10.0.2.20	10.0.0.10	HTTP	381	GET /favicon.ico HTTP/1.1
9	4.145990401	10.0.0.10	10.0.2.20	HTTP	741	HTTP/1.1 404 Not Found (text/html)
22	4.717774088	10.0.2.20	10.0.0.10	HTTP	401	GET /videoB_640_400_1000k_dash.mp4 HTTP/1.1
117	7.028057981	10.0.2.20	10.0.0.10	HTTP	402	GET /videoB_320_200_500k_dash.mp4 HTTP/1.1
164	7.727209972	10.0.2.20	10.0.0.10	HTTP	401	GET /videoB_160_100_200k_dash.mp4 HTTP/1.1
191	8.388209462	10.0.0.21	10.0.0.10	HTTP	381	GET /favicon.ico HTTP/1.1
193	8.388639901	10.0.0.10	10.0.0.21	HTTP	741	HTTP/1.1 404 Not Found (text/html)
205	8.732950837	10.0.0.21	10.0.0.10	HTTP	401	GET /videoB_640_400_1000k_dash.mp4 HTTP/1.1
2326	19.369591582	10.0.0.10	10.0.2.20	MP4	207	
2941	22.681584207	10.0.0.10	10.0.0.21	MP4	1016	
3185	27.587382264	10.0.0.21	10.0.0.10	HTTP	403	GET /videoB_640_400_1000k_dash.mp4 HTTP/1.1
5117	39.472715553	10.0.2.20	10.0.0.10	HTTP	402	GET /videoB_160_100_200k_dash.mp4 HTTP/1.1
5577	41.549432688	10.0.0.10	10.0.0.21	MP4	1016	
5987	46.418705455	10.0.0.21	10.0.0.10	HTTP	403	GET /videoB_640_400_1000k_dash.mp4 HTTP/1.1
6711	50.376732399	10.0.0.10	10.0.2.20	MP4	207	
7660	55.506197261	10.0.2.20	10.0.0.10	HTTP	404	GET /videoB_160_100_200k_dash.mp4 HTTP/1.1
8108	58.564895582	10.0.0.21	10.0.0.10	HTTP	407	GET /videoB_640_400_1000k_dash.mp4 HTTP/1.1

Figure 17: Captura de transmissão

**Questão 4 - Descreva o funcionamento do DASH neste caso concreto, referindo o papel do ficheiro MPD criado.**

O DASH (Dynamic, Adaptive Streaming over HTTP) é uma técnica de *streaming* de multimédia por HTTP que permite a transmissão dinâmica e adaptativa às condições da rede. Consequentemente, o DASH vai verificar a largura de banda e verificar qual a resolução do vídeo mais adequada para ser transmitida nesse respetivo *link*. Portanto, o DASH vai avaliando a qualidade de transmissão e conectividade e altera a resolução do vídeo conforme essas características.

Um exemplo deste funcionamento ocorreu durante a stream do video no firefox no portátil Bela e Alladin, onde através do wireshark conseguimos analisar os dados recebidos por ambos os clientes, como é possível ver na figura 17.

Em ambos os casos podemos ver que a *streaming* foi inicializado com um vídeo de maior resolução (640x400). Contudo, ao longo da *stream* deparamo-nos que o vídeo muda a resolução para uma mais baixa no portátil Bela (a vermelho na figura), acabando por transmitir o vídeo de menor resolução, no entanto, no portátil Alladin (a azul na figura) consegue transmitir o vídeo de maior resolução sem qualquer problema. Assim sendo, isto só foi possível ter acontecido por que o DASH verificou a transmissão na rede não são suficientes para manter

a transmissão na resolução mais alta, por isso baixou a resolução do vídeo para que fosse possível ver o vídeo completo sem que note perdas de *frame* ou pausas devido à falta de conexão.

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

**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.**

O cenário *unicast* consiste na transmissão entre um *sender* e um *receiver* enquanto o multicast consiste no envio de pacotes de um *sender* para um grupo de *hosts* na rede.

Para a escalabilidade, o *multicast* é mais escalável do que o *unicast*, pois permite um aumento do número de clientes sem que o aumento dos recursos aumente da mesma forma. Assim, permite que mais clientes recebem o conteúdo utilizando consideravelmente os mesmos recursos que o *unicast* utiliza para fornecer um cliente, podemos verificar estas afirmações com as figuras 18 e 19. A solução *unicast* utiliza 139kb/s enquanto a *multicast* 145kb/s, os números são bastantes semelhantes contudo o número de clientes já difere bastante, apenas 1 cliente em *unicast* enquanto em *multicast* 4 clientes.

Conclui-se assim que tanto a nível de escalabilidade como de tráfego de rede, a solução *multicast* é preferível.

Ethernet · 4		IPv4 · 2		IPv6 · 2		TCP		UDP · 2											
Address A	Port A	Address B	Port B	Packets	Bytes	Bits/s A → B	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s B → A						
10.0.0.10	46563	10.0.2.21	5555	1,422	929 k	139 k	1	1,422	11	770	0	0.000000	53.2565	0					
10.0.0.10	46564	10.0.2.21	5556	11	770	122	1	770	0	0	2.559620	50.2432	0						

Figure 18: Taxa bps *Unicast*

Ethernet · 9		IPv4 · 2		IPv6 · 7		TCP		UDP · 5											
Address A	Port A	Address B	Port B	Packets	Bytes	Bits/s A → B	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s B → A						
fe80::50:6bff:fe7c:7264	5353	ff02::fb	5353	1	107	—	1	107	0	0	36.671094	0.0000	—						
fe80::bc6c:99ff:feac:f500	5353	ff02::fb	5353	1	107	—	1	107	0	0	37.550640	0.0000	—						
10.0.0.10	42788	224.0.0.200	5555	1,634	1,110 k	145 k	1,634	1,110 k	0	0	0.000000	60.8933	0						
10.0.0.10	58863	224.2.127.254	9875	13	4758	630	13	4758	0	0	0.300901	60.4024	0						
10.0.0.10	42789	224.0.0.200	5556	13	910	120	13	910	0	0	0.603672	60.2995	0						

Figure 19: Taxa bps *Multicast*

## Conclusões

Através da realização deste trabalho prático tivemos a oportunidade de consolidar a matéria lecionada nas aulas teóricas relativamente aos vários protocolos associados ao *streaming* de vídeos.

Não só aprofundamos o nosso conhecimento na utilização do CORE e do Wireshark, como também na análise do tráfego, do nível de escalabilidade na transmissão de vídeo em HTTP estático e do de DASH.

Além disso, abordamos também as diferenças entre os cenários de unicast e multicast ao nível da rede.

Concluindo, pensamos ter alcançado os objetivos propostos para este trabalho.