

UNIVERSIDADE DO MINHO

DEPARTAMENTO DE INFORMÁTICA

Engenharia de Serviços em Rede

Trabalho Prático nº 2

Streaming de áudio e vídeo a pedido
e em tempo real

Grupo Nº 1

Ariana Lousada (PG47034)

Carlos Gomes (PG47083)

Tiago Sousa (PG47684)

2 de maio de 2022

Conteúdo

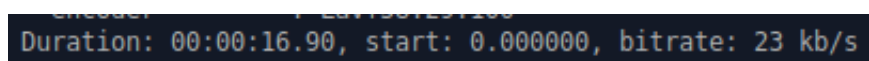
1	Questões e Respostas	3
2	Conclusões	7

Capítulo 1

Questões e Respostas

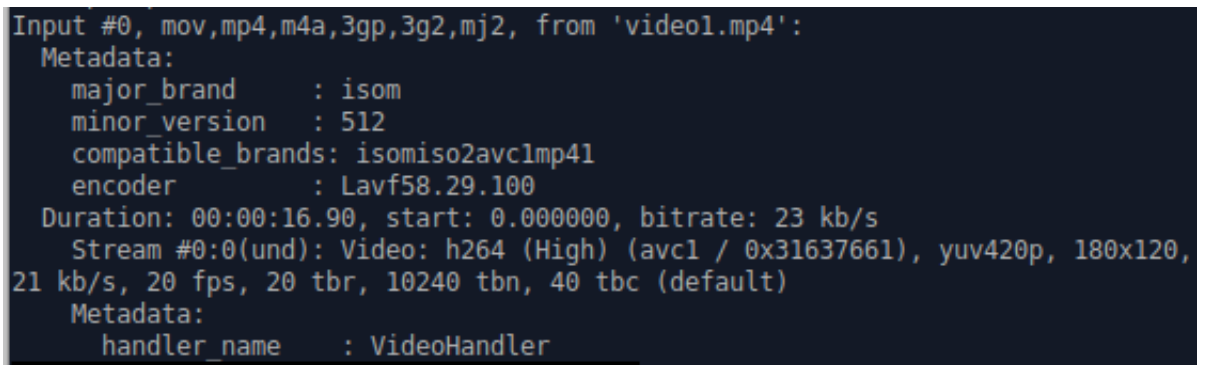
Questão 1: Capture três pequenas amostras de tráfego no link de saída do servidor, respectivamente com 1 cliente (VLC), com 2 clientes (VLC e Firefox) e com 3 clientes (VLC, Firefox e ffmpeg). Identifique a taxa em bps necessária (usando o `ffmpeg -i video1.mp4` e/ou o próprio `wireshark`), o encapsulamento 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).

Através da utilização do comando `ffmpeg -i video1.mp4`, chega-se à conclusão que vão ser necessários 23 kb/s de taxa de bps como demonstra a figura 1.1.



```
Duration: 00:00:16.90, start: 0.000000, bitrate: 23 kb/s
```

Figura 1.1: Taxa em bps necessária



```
Input #0, mov,mp4,m4a,3gp,3g2,mj2, from 'video1.mp4':
  Metadata:
    major_brand      : isom
    minor_version    : 512
    compatible_brands: isomiso2avc1mp41
    encoder          : Lavf58.29.100
  Duration: 00:00:16.90, start: 0.000000, bitrate: 23 kb/s
  Stream #0:0(und): Video: h264 (High) (avc1 / 0x31637661), yuv420p, 180x120,
  21 kb/s, 20 fps, 20 tbr, 10240 tbn, 40 tbc (default)
  Metadata:
    handler_name     : VideoHandler
```

Figura 1.2: `ffmpeg -i video1.mp4`

O encapsulamento consiste em transformar o vídeo compactado num determinado formato: é possível observar-se na figura 1.2 que existem vários tipos de encapsulamento, desde *mov*, até *mj2*.

Em relação ao número total de fluxos gerados, podemos analisar a figura seguinte:

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	44872	10.0.0.10	8080	1,162	903 k	543	35 k	619	867 k	0.000000	46.7856	6128		
10.0.2.20	39464	10.0.0.10	8080	885	886 k	264	17 k	621	869 k	0.000081	46.8725	2977		
10.0.2.21	49368	10.0.0.10	8080	1,192	906 k	572	37 k	620	868 k	0.000122	46.8006	6455		

Figura 1.3: Dados acerca da captura com 3 clientes em simultâneo.

Como podemos ver, no portátil 1 (10.0.0.20) temos um fluxo de 1162 pacotes; no portátil 2 (10.0.2.20) temos um fluxo de 885 pacotes e no portátil 3 (10.0.2.21) de 1192 pacotes.

Para analisar a escalabilidade desta solução podemos observar, por exemplo, o campo de Bits/s na figura 1.3. Apesar do aumento de clientes, podemos dizer que esta taxa mantém-se mais ou menos constante.

Apesar desta solução ser muito utilizada apresenta uma grande desvantagem: todos os clientes recebem a mesma codificação do ficheiro de vídeo, independentemente das variações da largura de banda disponíveis ao cliente, o que limita a qualidade do vídeo em si. Contudo, isto permite um aumento considerável no número de clientes sem afetar demasiado o tráfego na rede.

Com isto, podemos concluir que esta solução é escalável.

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.

Como é possível observar na figura seguinte, a largura de banda necessária para o streaming de vídeo no firefox é de **7.4 Kbit/s**. Podemos também consultar todos os protocolos utilizados, isto é, a pilha protocolar.

Wireshark - Protocol Hierarchy Statistics - etapa2portatil1.pcapng								
Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s
▼ Frame	100.0	615	100.0	724615	206 k	0	0	0
▼ Ethernet	100.0	615	1.2	8610	2458	0	0	0
▼ Internet Protocol Version 6	1.8	11	0.1	440	125	0	0	0
▼ User Datagram Protocol	0.2	1	0.0	8	2	0	0	0
Multicast Domain Name System	0.2	1	0.0	141	40	1	141	40
Open Shortest Path First	0.5	3	0.0	108	30	3	108	30
Internet Control Message Protocol v6	1.1	7	0.0	192	54	7	192	54
▼ Internet Protocol Version 4	97.9	602	1.7	12040	3438	0	0	0
▼ Transmission Control Protocol	95.4	587	96.9	702360	200 k	577	697656	199 k
▼ Hypertext Transfer Protocol	1.6	10	94.3	683496	195 k	8	2422	691
MP4 / ISOBMFF file format	0.2	1	93.9	680194	194 k	1	680399	194 k
Line-based text data	0.2	1	0.1	487	139	1	487	139
Open Shortest Path First	2.4	15	0.1	660	188	15	660	188
Address Resolution Protocol	0.3	2	0.0	56	15	2	56	15

Figura 1.4: Pilha protocolar

Questão 3: Ajuste o débito dos links da topologia de modo que o cliente

no portátil 2 exiba o vídeo de menor resolução e o cliente no portátil 1 exiba o vídeo com mais resolução. Mostre evidências.

25	1.6730...	10.0.0.20	10.0.0.10	TCP	66 50654 → 9999	[FIN, ACK] Seq=437 Ac...
26	1.6730...	10.0.0.10	10.0.0.20	TCP	66 9999 → 50654	[ACK] Seq=209 Ack=438...
27	2.0003...	10.0.0.1	224.0.0.5	OSPF	78 Hello Packet	
28	2.1968...	10.0.0.20	10.0.0.10	TCP	74 50656 → 9999	[SYN] Seq=0 Win=64240...
29	2.1969...	10.0.0.10	10.0.0.20	TCP	74 9999 → 50656	[SYN, ACK] Seq=0 Ack=...
30	2.1969...	10.0.0.20	10.0.0.10	TCP	66 50656 → 9999	[ACK] Seq=1 Ack=1 Win...
31	2.1974...	10.0.0.20	10.0.0.10	HTTP	3... GET /video2_540_360_1000k_dash.mp4...	
32	2.1975...	10.0.0.10	10.0.0.20	TCP	66 9999 → 50656	[ACK] Seq=1 Ack=334 W...
33	2.1977...	10.0.0.10	10.0.0.20	TCP	1... 9999 → 50656	[ACK] Seq=1 Ack=334 W...
34	2.1977...	10.0.0.10	10.0.0.20	TCP	1... 9999 → 50656	[ACK] Seq=1449 Ack=33...
35	2.1977...	10.0.0.10	10.0.0.20	TCP	1... 9999 → 50656	[ACK] Seq=2897 Ack=33...

Figura 1.5: Captura Wireshark do portátil 1

36	40.161...	10.0.0.10	10.0.2.20	TCP	66 9999 → 48232	[ACK] Seq=1 Ack=382 W...
37	40.161...	10.0.0.10	10.0.2.20	HTTP	2... HTTP/1.1 304 Not Modified	
38	40.162...	10.0.2.20	10.0.0.10	TCP	66 48232 → 9999	[FIN, ACK] Seq=382 Ac...
39	40.162...	10.0.0.10	10.0.2.20	TCP	66 9999 → 48232	[ACK] Seq=193 Ack=383...
40	40.174...	10.0.2.20	10.0.0.10	TCP	74 48234 → 9999	[SYN] Seq=0 Win=64240...
41	40.174...	10.0.0.10	10.0.2.20	TCP	74 9999 → 48234	[SYN, ACK] Seq=0 Ack=...
42	40.174...	10.0.2.20	10.0.0.10	TCP	66 48234 → 9999	[ACK] Seq=1 Ack=1 Win...
43	40.175...	10.0.2.20	10.0.0.10	HTTP	4... GET /video2_180_120_200k_dash.mp4 ...	
44	40.175...	10.0.0.10	10.0.2.20	TCP	66 9999 → 48234	[ACK] Seq=1 Ack=410 W...
45	40.175...	10.0.0.10	10.0.2.20	HTTP	2... HTTP/1.1 304 Not Modified	
46	40.175...	10.0.2.20	10.0.0.10	TCP	66 48234 → 9999	[FIN, ACK] Seq=410 Ac...

Figura 1.6: Captura Wireshark do portátil 2

Como podemos observar na figura 1.5, o portátil1 consegue consumir o conteúdo do vídeo com uma resolução 1000K, enquanto na figura 1.6, observa-se o consumo do vídeo de menor qualidade com resolução 200k no portátil2.

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

Neste cenário, o DASH possibilita uma adaptação dinâmica da *stream* de acordo com a largura de banda disponível do cliente. O MPEG-DASH transmite o conteúdo da *stream* dividindo o vídeo em *chunks* de menor dimensão e codifica-os em diferentes níveis de qualidade. Isto possibilita que os vídeos sejam transmitidos em diferentes resoluções, permitindo também a sua mudança durante a transmissão.

O MPEG-DASH importa informação contida num ficheiro MPD (*Media Presentation Description*) e de segmentos armazenados num servidor HTTP. Enquanto que os segmentos contêm *bitstreams*, o ficheiro MPD irá guardar toda a informação necessária para fazer o *pre-render* do conteúdo do vídeo. Este ficheiro contém informações como os tipos de *media*, resoluções de vídeo disponíveis, do espetro da largura de banda, entre outros.

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 protocolo RTP/RTCP pode ser aplicado em cenários *unicast* e *multicast*. Cada cenário tem as suas vantagens e desvantagens:

- O unicast consiste numa conexão de "um para um", visto que é composto por apenas um servidor e um cliente (que corresponde ao tipo de tráfego mais comum em redes TCP/IP), o que por sua vez leva a uma implementação mais simples.
- O multicast consiste numa conexão de "um para muitos", visto que é composto por um servidor e vários clientes. Esta tecnologia tem a capacidade de enviar pacotes da origem para vários destinos simultaneamente, geralmente dentro de uma rede local.
- No cenário unicast só é possível enviar data para um destino específico.
- No cenário multicast o tráfego da rede não aumenta, visto que os pacotes podem ser enviados simultaneamente para vários clientes contidos num determinado grupo utilizando a mesma cópia. Isto reduz significativamente o peso do envio na rede.

Visto isto, em comparação com o cenário unicast o multicast é escalável. Apesar de serem necessárias configurações específicas nas aplicações e rede para utilizar este protocolo, é bastante eficiente.

	Unicast	Multicast
Transmissão	Um para um	Um para vários
Bandwidth	desperdiçado	Utilizado eficientemente
Group Management	Não	Sim
Segurança	Bastante seguro	Seguro

Figura 1.7: diferença entre unicast e multicast

Capítulo 2

Conclusões

Durante a primeira etapa do trabalho prático foi testado o *streaming* HTTP simples sem adaptação dinâmica de débito.

Neste protocolo o vídeo é armazenado num servidor HTTP como um ficheiro normal com um URL específico. Quando o utilizador pretende ver o vídeo, o cliente estabelece uma conexão TCP com o servidor e envia um HTTP GET request do link do ficheiro. O servidor procede a enviar o ficheiro de vídeo numa HTTP response de acordo com as condições da rede no momento. Do lado do cliente, os bytes vão sendo armazenados num buffer. Quando o número de bytes do buffer excede um determinado limite, a aplicação do cliente vai utilizando periodicamente *video frames* do *buffer* da aplicação, descomprimindo-as e apresentando-as no ecrã à medida que o utilizador vai consumindo o vídeo.

O protocolo testado na segunda etapa do trabalho (DASH), apesar de ser baseado em HTTP streaming, já possui débito dinâmico. Neste protocolo, o vídeo é codificado em várias versões diferentes, cada uma com um bit rate distinto (também com resoluções diferentes). O cliente vai mandando requests de chunks de segmentos do vídeo, cujo tamanho vai variando com a largura de banda disponível no momento.

Na terceira etapa foi testado o RTP/RTCP em cenário unicast e multicast. No cenário unicast está presente uma conexão entre um servidor e um cliente, enquanto que no multicast tem-se uma conexão de um servidor para vários clientes. Em geral, a transmissão de pacotes em cenário multicast é bastante mais eficiente comparando com o cenário unicast. A única desvantagem que este protocolo apresenta é a necessidade de configurações específicas, tanto ao nível das aplicações como na rede.

Com a realização deste trabalho foi possível compreender a necessidade da existência de tecnologias tais como HTTP streaming, DASH e RTP/RTCP. Nos dias de hoje a rede através da qual acedemos à Internet poderá ser facilmente congestionada. Várias tecnologias tais como as testadas neste trabalho prático permitem usufruir de um serviço de *streaming* de vídeo com menos falhas. Este número reduzido de falhas custa, por vezes, períodos durante os quais a qualidade do vídeo em streaming é mais baixa, isto é, de menor resolução.

Em suma, com testes realizados podemos concluir que os dois últimos protocolos são os mais eficientes tanto na transmissão como em qualidade de vídeo, apesar de não serem suportados atualmente pela maior parte das aplicações mais utilizadas no mercado.