

# Wstęp do multimediiów

## Sprawozdanie Laboratorium 6

### Transmisja danych multimedialnych w sieciach szerokopasmowych

Amadeusz Lewandowski 331397

Czwartek 14

### Informacje Wstępne

Ćwiczenie zrealizowano na komputerze laboratoryjnym o numerze 14.

### Zadanie 3.1

Progresywne pobieranie danych multimedialnych z wykorzystaniem protokołu HTTP.

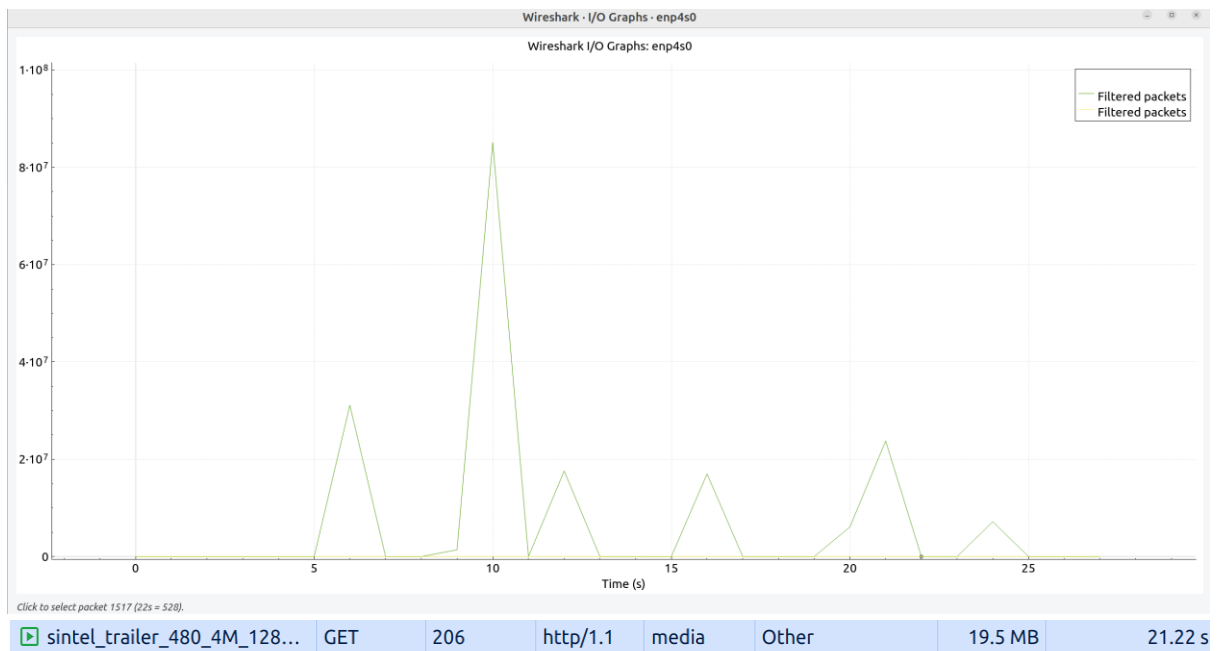
* tcp.port==80 && (ip.src==192.168.11.166    ip.dst== 192.168.11.166)					
No.	Time	Source	Destination	Protocol	Length Info
34	6.850859782	192.168.11.164	192.168.11.166	TCP	74 80 → 52908 [SYN, ACK] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=2907390868 TSecr=0 WS=128
35	6.850839953	192.168.11.166	192.168.11.164	TCP	74 80 → 52908 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM=1 TSval=2512501592 TSecr=2907390868 WS=128
36	6.850915170	192.168.11.164	192.168.11.166	TCP	66 52918 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=2907390868 TSecr=2512501592
37	6.850994134	192.168.11.164	192.168.11.166	TCP	74 52918 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=2907390868 TSecr=0 WS=128
38	6.8509166837	192.168.11.164	192.168.11.166	HTTP	556 GET /video/download.html HTTP/1.1
39	6.850244188	192.168.11.166	192.168.11.164	TCP	74 80 → 52918 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM=1 TSval=2512501592 TSecr=2907390868 WS=128
40	6.850248667	192.168.11.164	192.168.11.166	TCP	66 52918 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=2907390868 TSecr=2512501592
41	6.850270576	192.168.11.166	192.168.11.164	TCP	66 80 → 52908 [ACK] Seq=1 Ack=491 Win=64768 Len=0 TSval=2512501592 TSecr=2907390868
42	6.850461843	192.168.11.166	192.168.11.164	HTTP	622 HTTP/1.1 200 OK (text/html)
43	6.850405428	192.168.11.164	192.168.11.166	TCP	66 52908 → 80 [ACK] Seq=401 Ack=557 Win=83744 Len=0 TSval=2907390868 TSecr=2512501592
44	6.850504414	192.168.11.164	192.168.11.166	HTTP	492 GET /video/video/sintel_trailer_480_4M_128k.mp4 HTTP/1.1
45	6.850607733	192.168.11.166	192.168.11.164	TCP	7386 80 → 52908 [PSH, ACK] Seq=557 Ack=917 Win=64384 Len=7240 TSval=2512501619 TSecr=2907390894 [TCP segment of a reassembled PDU]
46	6.85093508	192.168.11.164	192.168.11.166	TCP	66 52908 → 80 [ACK] Seq=917 Ack=707 Win=78592 Len=0 TSval=2907390895 TSecr=2512501619
47	6.850930902	192.168.11.166	192.168.11.164	TCP	7386 80 → 52908 [PSH, ACK] Seq=7797 Ack=917 Win=64384 Len=7240 TSval=2512501619 TSecr=2907390894 [TCP segment of a reassembled PDU]
48	6.850935448	192.168.11.164	192.168.11.166	TCP	66 52908 → 80 [ACK] Seq=917 Ack=15037 Win=82048 Len=0 TSval=2907390895 TSecr=2512501619

Na powyższym zrzucie znajduje się fragment listy pakietów przechwyconych za pomocą wymaganego filtra związanych z progresywnym pobieraniem danych multimedialnych.

Jakie komunikaty protokołu HTTP zostały użyte podczas transmisji?

38	6.859166837	192.168.11.164	192.168.11.166	HTTP	556 GET /video/download.html HTTP/1.1
39	6.859244188	192.168.11.166	192.168.11.164	TCP	74 80 → 52918 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM=1 TSval=2512501592 TSecr=2907390868 WS=128
40	6.859248667	192.168.11.164	192.168.11.166	TCP	66 52918 → 80 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=2907390868 TSecr=2512501592
41	6.859270576	192.168.11.166	192.168.11.164	TCP	66 80 → 52908 [ACK] Seq=1 Ack=491 Win=64768 Len=0 TSval=2512501592 TSecr=2907390868
42	6.859461843	192.168.11.166	192.168.11.164	HTTP	622 HTTP/1.1 200 OK (text/html)
93	6.891047201	192.168.11.164	192.168.11.166	HTTP	499 GET /video/video/sintel_trailer_480_4M_128k.mp4 HTTP/1.1
94	6.891050955	192.168.11.164	192.168.11.166	TCP	64 52908 → 80 [RST] Seq=917 Win=0 Len=0
95	6.891145348	192.168.11.166	192.168.11.164	TCP	66 80 → 52918 [ACK] Seq=1 Ack=434 Win=64768 Len=0 TSval=2512501624 TSecr=2907390900
96	6.891342541	192.168.11.166	192.168.11.164	TCP	7386 80 → 52918 [PSH, ACK] Seq=1 Ack=434 Win=64768 Len=7240 TSval=2512501624 TSecr=2907390900 [TCP segment of a reassembled PDU]
97	6.891346329	192.168.11.164	192.168.11.166	TCP	66 52918 → 80 [ACK] Seq=434 Ack=7241 Win=78592 Len=0 TSval=2907390900 TSecr=2512501624
98	6.891392745	192.168.11.166	192.168.11.164	TCP	7386 80 → 52918 [PSH, ACK] Seq=7241 Ack=434 Win=64768 Len=7240 TSval=2512501624 TSecr=2907390900 [TCP segment of a reassembled PDU]
99	6.891395405	192.168.11.164	192.168.11.166	TCP	66 52918 → 80 [ACK] Seq=434 Ack=14481 Win=82048 Len=0 TSval=2907390900 TSecr=2512501624
100	6.891594661	192.168.11.166	192.168.11.164	HTTP	5183 HTTP/1.1 206 Partial Content (video/mp4)
118	6.897447119	192.168.11.166	192.168.11.164	HTTP	15254 HTTP/1.1 200 OK (image/x-icon)
119	6.897453801	192.168.11.164	192.168.11.166	TCP	66 52918 → 80 [ACK] Seq=867 Ack=191170 Win=179328 Len=0 TSval=2907390906 TSecr=2512501630
120	6.918524864	192.168.11.164	192.168.11.166	HTTP	497 GET /video/video/sintel_trailer_480_4M_128k.mp4 HTTP/1.1

Jak widać na powyższych zrzutach mamy metody GET w requestach HTTP i odpowiedzi typu OK (status 200), co zgadza się ze stanem faktycznym, gdyż tylko pobieramy dane z serwera i je otrzymujemy, bez napotkania błędów (film się odtwarza, strona się wyświetla).



Wartości oszacowano na podstawie danych z powyższych zrzutów. Maksymalna przepływność z wykresu, natomiast średnia z danych z przeglądarki.

Oszacowana maksymalna przepływność: 85 Mbit/s

Oszacowana średnia przepływność: 7.35 Mbit/s  $((19.5 / 21.22) * 8)$

```
<html>
  <head>
    <meta http-equiv="Pragma" content="no-cache">
    <meta http-equiv="Expires" content="-1">
    <meta http-equiv="CACHE-CONTROL" content="NO-CACHE">
    <title>Video App - test</title>
  </head>

  <body>
    <h1>Progressive download</h1>

    <video controls autoplay>
      <source src="video/sintel_trailer_480_4M_128k.mp4" type="video/mp4">
    </video>

    <hr>
    <a href="index.html">Home</a>
  </body>
</html>
```

Jak widać na powyższym zrzucie fragmentu kodu źródłowego strony do odtworzenia filmu na stronie zostały użyte tagi "video" (odpowiada za odtwarzacz) oraz "source" (umożliwia pobranie filmu z serwera), w którym to mamy film w formacie mp4.

Jakie rodzaje danych multimedialnych mogą być odtwarzane w dokumentach HTML?

Zgodnie [ze źródłem](#) mogą być to między innymi formaty:

- Ogg (koder Theora)
- MP4 (kodery H.264, HEVC)
- WebM (kodery VP8, VP9, AV1)

Jednakże konkretne z tych formaty, które możemy odtworzyć zależą od danej przeglądarki, której używamy.

## Zadanie 3.2

### Adaptacyjne strumieniowanie danych multimedialnych z wykorzystaniem standardu MPEG-DASH.

* tcp.port==80 && ip.src==192.168.11.166   ip.dst == 192.168.11.166						
No.	Time	Source	Destination	Protocol	Length	Info
34	2.89240797	192.168.11.164	192.168.11.166	HTTP	552	GET /video/dash.html HTTP/1.1
35	2.892424011	192.168.11.166	192.168.11.164	HTTP	755	HTTP/1.1 200 OK (text/html)
36	2.900475106	192.168.11.164	192.168.11.166	HTTP	451	GET /video/js/dash-2.0.all.min.js HTTP/1.1
37	2.900831591	192.168.11.164	192.168.11.164	TCP	7306	80 -> 59790 [PSH, ACK] Seq=690 Ack=872 Win=501 Len=7240 TSval=2514874284 TSecr=2909763574 [TCP segment of a reassembled PDU]
38	2.900859970	192.168.11.164	192.168.11.166	TCP	66	59790 -> 80 [ACK] Seq=872 Ack=7308 Win=8987 Len=0 TSval=2909763574 TSecr=2514874284
39	2.900879779	192.168.11.166	192.168.11.164	TCP	7306	80 -> 59790 [PSH, ACK] Seq=7308 Ack=872 Win=501 Len=7240 TSval=2514874284 TSecr=2909763574 [TCP segment of a reassembled PDU]
40	2.901137070	192.168.11.166	192.168.11.164	TCP	10202	80 -> 59790 [PSH, ACK] Seq=15170 Ack=872 Win=501 Len=18136 TSval=2514874285 TSecr=2909763574 [TCP segment of a reassembled PDU]
41	2.901185187	192.168.11.166	192.168.11.164	TCP	4418	80 -> 59790 [PSH, ACK] Seq=25306 Ack=872 Win=501 Len=344 TSval=2514874285 TSecr=2909763574 [TCP segment of a reassembled PDU]
42	2.90166146	192.168.11.164	192.168.11.166	TCP	66	59790 -> 80 [ACK] Seq=872 Ack=29050 Win=8936 Len=0 TSval=2909763575 TSecr=2514874284
43	2.902036671	192.168.11.166	192.168.11.164	TCP	20338	80 -> 59790 [ACK] Seq=29050 Ack=872 Win=501 Len=20272 TSval=2514874286 TSecr=2909763575 [TCP segment of a reassembled PDU]
44	2.902070498	192.168.11.166	192.168.11.164	TCP	1514	80 -> 59790 [PSH, ACK] Seq=49922 Ack=872 Win=501 Len=1448 TSval=2514874286 TSecr=2909763575 [TCP segment of a reassembled PDU]
45	2.902075289	192.168.11.164	192.168.11.166	TCP	66	59790 -> 80 [ACK] Seq=872 Ack=49922 Win=8931 Len=0 TSval=2909763576 TSecr=2514874286
46	2.902229137	192.168.11.166	192.168.11.164	TCP	21706	80 -> 59790 [PSH, ACK] Seq=51370 Ack=872 Win=501 Len=21720 TSval=2514874286 TSecr=2909763575 [TCP segment of a reassembled PDU]
47	2.902250845	192.168.11.164	192.168.11.166	TCP	66	59790 -> 80 [ACK] Seq=872 Ack=73090 Win=8936 Len=0 TSval=2909763576 TSecr=2514874286
48	2.902328151	192.168.11.166	192.168.11.164	TCP	7306	80 -> 59790 [PSH, ACK] Seq=73090 Ack=872 Win=501 Len=7240 TSval=2514874286 TSecr=2909763576 [TCP segment of a reassembled PDU]
49	2.902348833	192.168.11.164	192.168.11.166	TCP	66	59790 -> 80 [ACK] Seq=872 Ack=80330 Win=8936 Len=0 TSval=2909763576 TSecr=2514874286
50	2.902607811	192.168.11.166	192.168.11.164	TCP	31922	80 -> 59790 [PSH, ACK] Seq=80330 Ack=872 Win=501 Len=31956 TSval=2514874286 TSecr=2909763576 [TCP segment of a reassembled PDU]
51	2.902642460	192.168.11.166	192.168.11.164	TCP	2962	80 -> 59790 [ACK] Seq=112186 Ack=872 Win=501 Len=2896 TSval=2514874286 TSecr=2909763576 [TCP segment of a reassembled PDU]
52	2.902660487	192.168.11.164	192.168.11.166	TCP	66	59790 -> 80 [ACK] Seq=872 Ack=112186 Win=8936 Len=0 TSval=2909763576 TSecr=2514874286
53	2.902726286	192.168.11.164	192.168.11.166	TCP	66	59790 -> 80 [ACK] Seq=872 Ack=115082 Win=8936 Len=0 TSval=2909763576 TSecr=2514874286
54	2.902808704	192.168.11.166	192.168.11.164	TCP	30474	80 -> 59790 [PSH, ACK] Seq=115082 Ack=872 Win=501 Len=30408 TSval=2514874286 TSecr=2909763576 [TCP segment of a reassembled PDU]
55	2.902917962	192.168.11.164	192.168.11.166	TCP	66	59790 -> 80 [ACK] Seq=872 Ack=145400 Win=8936 Len=0 TSval=2909763576 TSecr=2514874286
56	2.903000719	192.168.11.166	192.168.11.164	TCP	14546	80 -> 59790 [PSH, ACK] Seq=145400 Ack=872 Win=501 Len=14460 TSval=2514874286 TSecr=2909763576 [TCP segment of a reassembled PDU]
57	2.903021231	192.168.11.164	192.168.11.166	TCP	66	59790 -> 80 [ACK] Seq=872 Ack=159070 Win=8936 Len=0 TSval=2909763576 TSecr=2514874286
58	2.903051832	192.168.11.166	192.168.11.164	TCP	5858	80 -> 59790 [ACK] Seq=159070 Ack=872 Win=501 Len=5792 TSval=2514874286 TSecr=2909763576 [TCP segment of a reassembled PDU]
59	2.903061907	192.168.11.164	192.168.11.166	TCP	66	59790 -> 80 [ACK] Seq=872 Ack=165762 Win=8936 Len=0 TSval=2909763577 TSecr=2514874286
60	2.903121383	192.168.11.166	192.168.11.164	TCP	8754	80 -> 59790 [PSH, ACK] Seq=165762 Ack=872 Win=501 Len=8688 TSval=2514874286 TSecr=2909763576 [TCP segment of a reassembled PDU]
61	2.903155954	192.168.11.164	192.168.11.166	TCP	66	59790 -> 80 [ACK] Seq=872 Ack=174450 Win=8936 Len=0 TSval=2909763577 TSecr=2514874286
62	2.903428698	192.168.11.166	192.168.11.164	TCP	36266	80 -> 59790 [PSH, ACK] Seq=174450 Ack=872 Win=501 Len=36200 TSval=2514874287 TSecr=2909763576 [TCP segment of a reassembled PDU]

Na powyższym zrzucie znajduje się fragment listy pakietów przechwyconych za pomocą wymaganego filtra związanych z progresywnym pobieraniem danych multimedialnych.

### Jakie komunikaty protokołu HTTP zostały użyte podczas transmisji?

34	2.892040797	192.168.11.164	192.168.11.166	HTTP	552	GET /video/dash.html HTTP/1.1
35	2.892424011	192.168.11.166	192.168.11.164	HTTP	755	HTTP/1.1 200 OK (text/html)
36	2.900475106	192.168.11.164	192.168.11.166	HTTP	451	GET /video/js/dash-2.0.all.min.js HTTP/1.1
78	2.904769744	192.168.11.166	192.168.11.164	HTTP	4393	HTTP/1.1 200 OK (application/javascript)
79	2.904777823	192.168.11.164	192.168.11.166	TCP	66	59790 -> 80 [ACK] Seq=872 Ack=366927 Win=8936 Len=0 TSval=2909763578 TSecr=2514874287
80	2.917749232	192.168.11.164	192.168.11.166	HTTP	390	GET /video/js/dash.all.min.js.map HTTP/1.1
81	2.918217199	192.168.11.164	192.168.11.166	HTTP	457	GET /video/video/sintel-dash/sintel.mpd HTTP/1.1
191	2.935112626	192.168.11.164	192.168.11.166	TCP	66	59790 -> 80 [ACK] Seq=1196 Ack=2195751 Win=8725 Len=0 TSval=2909763609 TSecr=2514874317
192	2.935162600	192.168.11.164	192.168.11.166	TCP	66	59790 -> 80 [ACK] Seq=1196 Ack=2200656 Win=8936 Len=0 TSval=2909763609 TSecr=2514874317
193	2.935307861	192.168.11.164	192.168.11.166	HTTP	467	GET /video/video/sintel-dash/sintel_set1_init.mp4 HTTP/1.1
194	2.935335366	192.168.11.164	192.168.11.166	HTTP	479	GET /video/video/sintel-dash/sintel_trailer_128k_dashinit.mp4 HTTP/1.1
195	2.935611030	192.168.11.166	192.168.11.164	MP4	1180	
196	2.935688240	192.168.11.166	192.168.11.164	MP4	1198	
197	2.943999513	192.168.11.164	192.168.11.166	HTTP	476	GET /video/video/sintel-dash/sintel_trailer_128k_dash1.m4s HTTP/1.1
198	2.944071585	192.168.11.164	192.168.11.166	HTTP	478	GET /video/video/sintel-dash/sintel_trailer_480_2k_dash1.m4s HTTP/1.1
278	2.95329994	192.168.11.166	192.168.11.164	HTTP	54738	HTTP/1.1 200 OK
279	2.953576463	192.168.11.164	192.168.11.166	TCP	66	53670 -> 80 [ACK] Seq=1634 Ack=1185502 Win=4470 Len=0 TSval=2909763627 TSecr=2514874336
280	2.957965305	192.168.11.164	192.168.11.166	HTTP	475	GET /video/video/sintel-dash/sintel_trailer_128k_dash3.m4s HTTP/1.1
281	2.958435607	192.168.11.166	192.168.11.164	TCP	31922	80 -> 53670 [PSH, ACK] Seq=1185502 Ack=2044 Win=501 Len=31856 TSval=2514874342 TSecr=2909763631 [TCP segment of a reassembled PDU]
282	2.958576404	192.168.11.166	192.168.11.164	HTTP	916	HTTP/1.1 200 OK
283	2.958599478	192.168.11.164	192.168.11.166	TCP	66	53670 -> 80 [ACK] Seq=2044 Ack=1218208 Win=4261 Len=0 TSval=2909763632 TSecr=2514874342
284	2.962840840	192.168.11.164	192.168.11.166	HTTP	478	GET /video/video/sintel-dash/sintel_trailer_480_2k_dash2.m4s HTTP/1.1
285	2.963096001	192.168.11.166	192.168.11.164	TCP	65226	80 -> 53670 [PSH, ACK] Seq=1218208 Ack=2456 Win=501 Len=65160 TSval=2514874347 TSecr=2909763636 [TCP segment of a reassembled PDU]
286	2.963750620	192.168.11.164	192.168.11.166	TCP	66	53670 -> 80 [ACK] Seq=2456 Ack=1283366 Win=4015 Len=0 TSval=2909763637 TSecr=2514874347
287	2.964037434	192.168.11.166	192.168.11.164	TCP	40610	80 -> 53670 [ACK] Seq=1283366 Ack=2456 Win=501 Len=40544 TSval=2514874347 TSecr=2909763636 [TCP segment of a reassembled PDU]
288	2.964244229	192.168.11.166	192.168.11.164	TCP	24082	80 -> 53670 [PSH, ACK] Seq=132912 Ack=2456 Win=501 Len=24016 TSval=2514874347 TSecr=2909763636 [TCP segment of a reassembled PDU]
289	2.964455965	192.168.11.166	192.168.11.164	TCP	24082	80 -> 53670 [PSH, ACK] Seq=1348528 Ack=2456 Win=501 Len=24016 TSval=2514874347 TSecr=2909763636 [TCP segment of a reassembled PDU]
290	2.964535582	192.168.11.164	192.168.11.166	TCP	66	53670 -> 80 [ACK] Seq=2456 Ack=1373144 Win=3827 Len=0 TSval=2909763638 TSecr=2514874347
291	2.965016435	192.168.11.166	192.168.11.164	TCP	65226	80 -> 53670 [PSH, ACK] Seq=1373144 Ack=2456 Win=501 Len=65160 TSval=2514874348 TSecr=2909763637 [TCP segment of a reassembled PDU]
292	2.965037895	192.168.11.164	192.168.11.166	TCP	66	53670 -> 80 [ACK] Seq=2456 Ack=1438304 Win=4015 Len=0 TSval=2909763638 TSecr=2514874348
293	2.965203586	192.168.11.164	192.168.11.166	HTTP	476	GET /video/video/sintel-dash/sintel_trailer_128k_dash4.m4s HTTP/1.1

Tak samo jak wcześniej (co widać na powyższych zrzutach) mamy metody GET w requestach HTTP i odpowiedzi typu OK (status 200), co zgadza się ze stanem faktycznym, gdyż tylko pobieramy dane z serwera i je otrzymujemy, bez napotkania błędów (film się odtwarza, strona się wyświetla). Z tą różnicą, że jest ich dużo więcej, gdyż film jest pobierany fragmentami co widać dokładnie na zrzucie poniżej.

Name	Method	Status	Protocol	Type	Initiator	Size	Time
□ sintel_trailer_128k_dash12.m4s	GET	200	http/1.1	xhr	FragmentLoader.js:246	32.8 kB	2 ms
□ sintel_trailer_128k_dash13.m4s	GET	200	http/1.1	xhr	FragmentLoader.js:246	32.7 kB	2 ms
□ sintel_trailer_480_4M_dash4.m4s	GET	200	http/1.1	xhr	FragmentLoader.js:246	977 kB	10 ms
□ sintel_trailer_128k_dash14.m4s	GET	200	http/1.1	xhr	FragmentLoader.js:246	32.6 kB	3 ms
□ sintel_trailer_128k_dash15.m4s	GET	200	http/1.1	xhr	FragmentLoader.js:246	32.9 kB	3 ms
□ sintel_trailer_128k_dash16.m4s	GET	200	http/1.1	xhr	FragmentLoader.js:246	32.9 kB	2 ms
□ sintel_trailer_480_4M_dash5.m4s	GET	200	http/1.1	xhr	FragmentLoader.js:246	925 kB	9 ms
□ sintel_trailer_480_4M_dash6.m4s	GET	200	http/1.1	xhr	FragmentLoader.js:246	112 kB	2 ms
□ sintel_trailer_480_4M_dash7.m4s	GET	200	http/1.1	xhr	FragmentLoader.js:246	1.5 MB	13 ms
□ sintel_trailer_480_4M_dash8.m4s	GET	200	http/1.1	xhr	FragmentLoader.js:246	1.1 MB	10 ms
□ sintel_trailer_480_4M_dash9.m4s	GET	200	http/1.1	xhr	FragmentLoader.js:246	442 kB	4 ms
□ sintel_trailer_480_4M_dash10.m4s	GET	200	http/1.1	xhr	FragmentLoader.js:246	1.1 MB	10 ms
□ sintel_trailer_480_4M_dash11.m4s	GET	200	http/1.1	xhr	FragmentLoader.js:246	1.4 MB	13 ms
□ sintel_trailer_480_4M_dash12.m4s	GET	200	http/1.1	xhr	FragmentLoader.js:246	812 kB	8 ms
□ sintel_trailer_480_4M_dash13.m4s	GET	200	http/1.1	xhr	FragmentLoader.js:246	1.2 MB	11 ms
□ sintel_trailer_480_4M_dash14.m4s	GET	200	http/1.1	xhr	FragmentLoader.js:246	1.3 MB	12 ms
□ sintel_trailer_480_4M_dash15.m4s	GET	200	http/1.1	xhr	FragmentLoader.js:246	778 kB	7 ms
□ sintel_trailer_480_4M_dash16.m4s	GET	200	http/1.1	xhr	FragmentLoader.js:246	229 kB	3 ms
□ sintel_trailer_480_4M_dash17.m4s	GET	200	http/1.1	xhr	FragmentLoader.js:246	1.1 MB	11 ms
data:image/svg+xml;...	GET	200	data	svg+xml	Other	(memory cache)	0 ms
■ favicon.ico	GET	200	http/1.1	x-icon	Other	172 kB	6 ms
□ sintel_trailer_480_4M_dash18.m4s	GET	200	http/1.1	xhr	FragmentLoader.js:246	878 kB	12 ms
□ sintel_trailer_128k_dash17.m4s	GET	200	http/1.1	xhr	FragmentLoader.js:246	32.8 kB	4 ms
□ sintel_trailer_480_4M_dash19.m4s	GET	200	http/1.1	xhr	FragmentLoader.js:246	618 kB	11 ms
□ sintel_trailer_128k_dash18.m4s	GET	200	http/1.1	xhr	FragmentLoader.js:246	32.6 kB	2 ms
□ sintel_trailer_480_4M_dash20.m4s	GET	200	http/1.1	xhr	FragmentLoader.js:246	773 kB	12 ms
□ sintel_trailer_128k_dash19.m4s	GET	200	http/1.1	xhr	FragmentLoader.js:246	32.3 kB	4 ms
□ sintel_trailer_480_4M_dash21.m4s	GET	200	http/1.1	xhr	FragmentLoader.js:246	523 kB	8 ms



Wartości oszacowano na podstawie danych z powyższych zrzutów. Maksymalna przepływność z wykresu, natomiast średnia z danych z przeglądarki.

Oszacowana maksymalna przepływność: 109.5 Mbit/s

Oszacowana średnia przepływność: 7.49 Mbit/s ((22.7 / 24.25) \* 8)

```

<?xml version="1.0"?>
<!-- MPD file generated with GPAC version 0.5.2-DEV-revVersion: 0.5.2-426-gc5ad4e4df95-5 at 2023-05-30T00:59:11.103Z-->
<MPD xmlns="urn:mpeg:dash:schema:mpd:2011" minBufferTime="PT1.500S" type="static" mediaPresentationDuration="PT0H0M52.209S" maxSegmentDuration="PT0H0M2.000S" profiles="urn:mpeg:dash:profile:isoff-live:2011,http://dashif.org/guidelines/dash264">
  <ProgramInformation moreInformationURL="http://gpac.sourceforge.net">
    <title>sintel.mpd generated by GPAC</title>
  </ProgramInformation>

  <Period duration="PT0H0M52.209S">
    <AdaptationSet segmentAlignment="true" bitstreamSwitching="true" maxWidth="854" maxHeight="480" maxFrameRate="24" par="854:480" lang="und">
      <SegmentTemplate initialization="sintel_seti_init.mp4"/>
      <Representation id="1" mimeType="video/mp4" codecs="avc1.64001e" width="854" height="480" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="832754">
        <SegmentTemplate timescale="12288" media="sintel_trailer_480_1M_dashNumber$.m4s" startNumber="1" duration="22598"/>
      </Representation>
      <Representation id="2" mimeType="video/mp4" codecs="avc1.64001e" width="854" height="480" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="1628844">
        <SegmentTemplate timescale="12288" media="sintel_trailer_480_2M_dashNumber$.m4s" startNumber="1" duration="22598"/>
      </Representation>
      <Representation id="3" mimeType="video/mp4" codecs="avc1.64001e" width="854" height="480" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="3228059">
        <SegmentTemplate timescale="12288" media="sintel_trailer_480_4M_dashNumber$.m4s" startNumber="1" duration="22598"/>
      </Representation>
    </AdaptationSet>
    <AdaptationSet segmentAlignment="true" lang="und">
      <Representation id="4" mimeType="audio/mp4" codecs="mp4a.40.2" audioSamplingRate="48000" startWithSAP="1" bandwidth="130244">
        <AudioChannelConfiguration schemeIdUri="urn:mpeg:dash:23003:3:audio_channel_configuration:2011" value="2"/>
      <SegmentTemplate timescale="48000" media="sintel_trailer_128k_dashNumber$.m4s" startNumber="1" duration="95232" initialization="sintel_trailer_128k_dashinit.mp4"/>
      </Representation>
    </AdaptationSet>
  </Period>
</MPD>

```

Deskryptor sintel.mpd.

Format to mp4. Mamy trzy warianty dla obrazu i jeden dla dźwięku.

Timescale dla obrazu to 12288, a duration to 22598, czyli w zaokrągleniu wychodzi, że jeden fragment ma 1.84 sekundy, czyli w całym filmie mamy około 29 fragmentów.

Timescale dla dźwięku to 48000, a duration to 95232, czyli w zaokrągleniu wychodzi, że jeden fragment ma 1.98 sekundy, czyli w całym filmie mamy około 27 fragmentów.

## Zadanie 3.3

Na podstawie uzyskanych wyników porównaj analizowane metody strumieniowania danych multimedialnych. Określ możliwości ich zastosowania do udostępniania materiałów multimedialnych, m.in. w telewizji internetowej, usługach Video On Demand (VOD).

Przede wszystkim wyróżniają się ilością wykorzystywanych zapytań HTTP i sposobem implementacji. Do pierwszej metody wystarczy nam zwykły tag HTML, natomiast do drugiej potrzebujemy skryptów JS. W obu metodach średnia przepływność była podobna. Pierwsza metoda nie sprawdzi się w przypadku ograniczonej przepływności sieci, natomiast druga już jak najbardziej w związku z tym wydaje mi się, że metoda ta może być lepsza dla usług typu VOD. Dodatkowo daje to możliwość wybrania sobie jakości filmu przez użytkownika (jak np. na YouTube) co jest dodatkowym plusem.