

Sprawozdanie

Wstęp do multimediów (WMM) Laboratorium #6: Laboratorium – Transmisja danych multimedialnych w sieciach szerokopasmowych

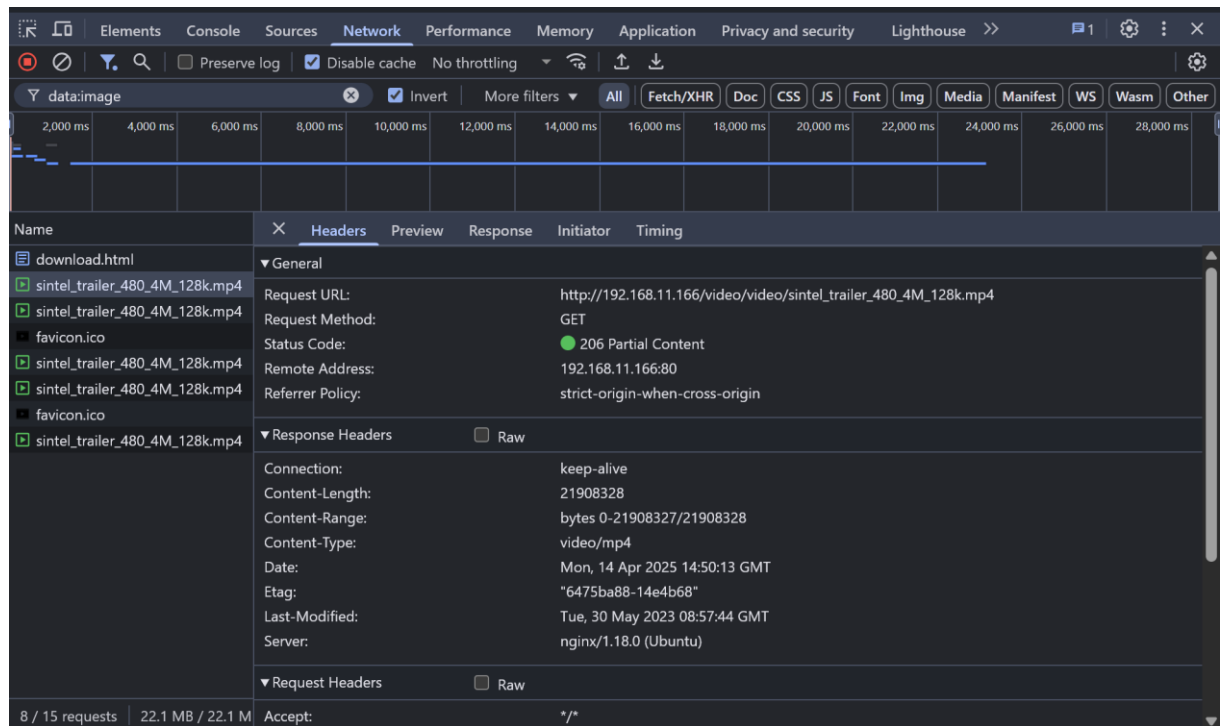
Sprzęt: własny komputer podłączony do sieci laboratoryjnej

1 Cel i zakres ćwiczenia

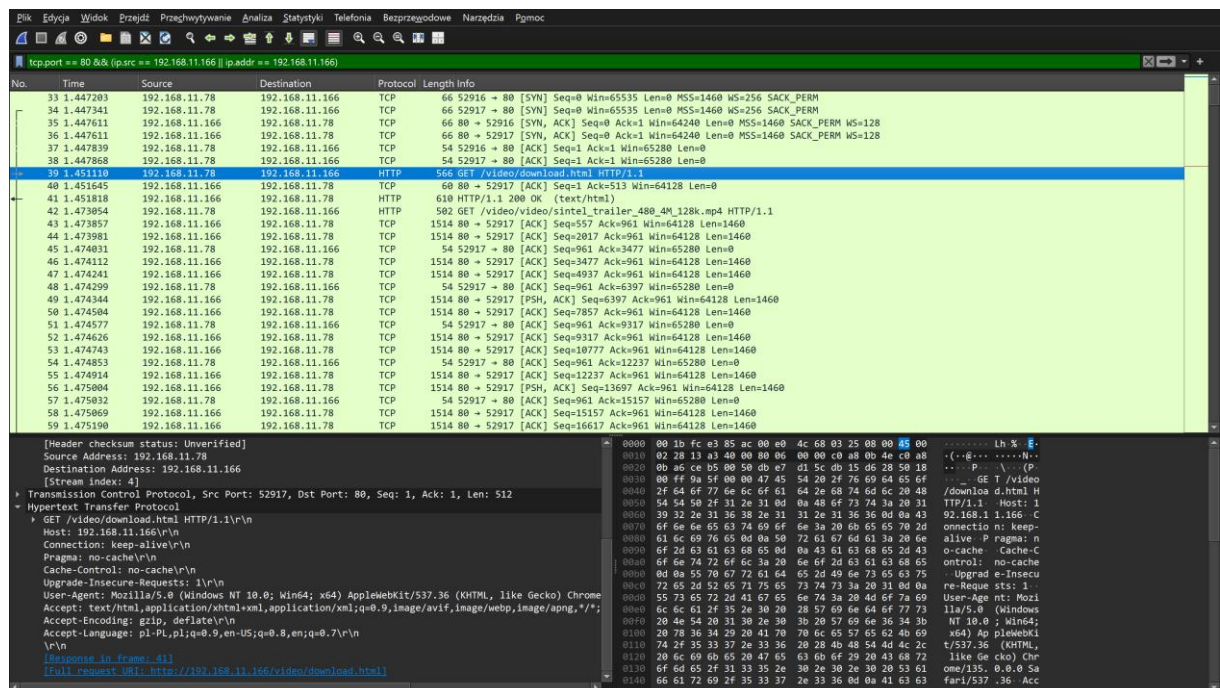
Celem ćwiczenia jest zapoznanie studentów metodami transmisji danych multimedialnych w sieciach szerokopasmowych. W ćwiczeniu zostanie wykorzystany prosty model systemu dystrybucji danych multimedialnych składający się z serwera udostępniającego dane oraz klienta na którym te dane są odtwarzane. Transmisja danych zostanie zrealizowana z wykorzystaniem protokołu HTTP oraz standardu MPEG-DASH

Zad. 3.1 Progresywne pobieranie danych multimedialnych z wykorzystaniem protokołu HTTP

Uruchomienie przeglądarki, odtworzenie pliku i przechwycenie pakietów:



Pakiety:



Komunikaty protokołu HTTP użyte podczas transmisji:

Zostały użyte komunikaty 'GET'

W zapytaniu użyto nagłówka 'Range', aby zadeklarować jaką część pliku pobrać (pobieranie po kawałkach).

W nagłówku odpowiedzi 'Content-Range', podano jaką część filmu pobrano.

Oszacować średnią i maksymalną przepływność strumienia danych podczas transmisji pliku multimedialnego - wynik należy podać w kbit/s lub Mbit/s,

| Statystyki | | | |
|---------------------------|--------------|------------------|------------|
| Pomiary | Przechwycone | Wyświetlane | Zaznaczone |
| Pakiety | 23485 | 23233 (98.9%) | — |
| Okres czasu, s | 56.132 | 54.535 | — |
| Średni pps | 418.4 | 426.0 | — |
| Średni rozmiar pakietu, B | 1022 | 1031 | — |
| Bajty | 23999227 | 23950657 (99.8%) | 0 |
| Średnio bajtów/s | 427 k | 439 k | — |
| Średnio bitów/s | 3420 k | 3513 k | — |

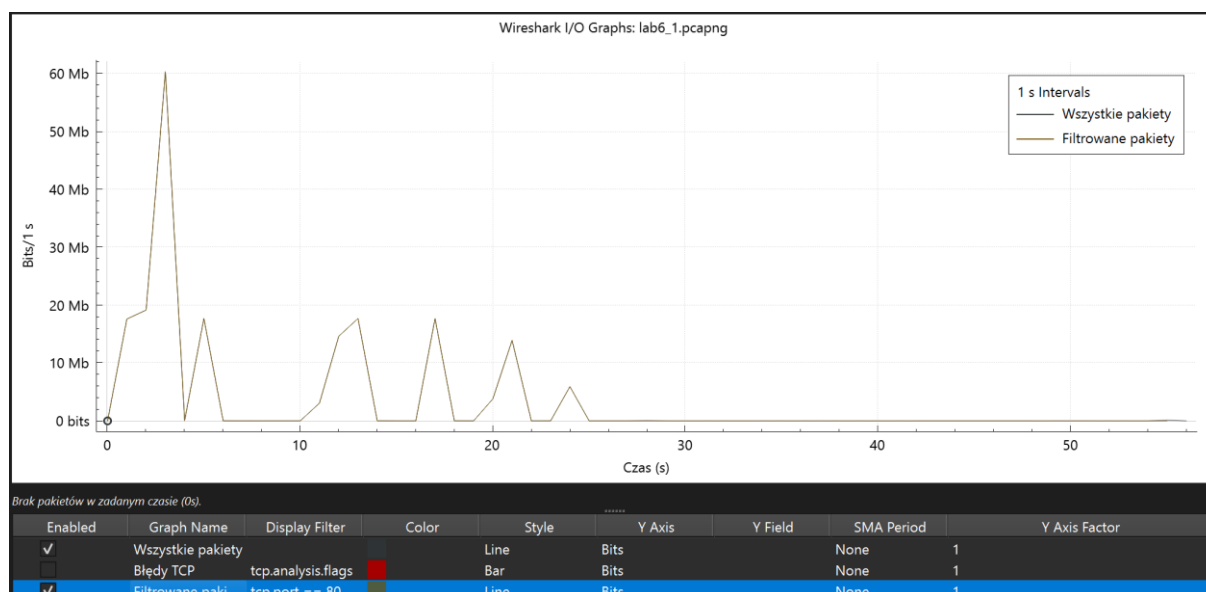
23 999 227 B -> 23 436,74 kB

Średnia przepływność: 3 749,87 kbit/s

23 436,74 kB / 52 = 468,73 kB/s -> **3 749,87 kbit/s**

Maksymalna przepływność: 60 Mbit/s

Wykres przepustowości:



Na podstawie analizy kodu źródłowego dokumentu HTML z pkt. 2 określić jakie elementy języka HTML5 zostały wykorzystane do odtworzenia pliku multimedialnego.

Wykorzystany element to <video> i <src>

Jakie rodzaje danych multimedialnych (format pliku, koder audio/video) mogą być odtwarzane w dokumentach HTML?

Formaty audio: MP3, WAV, OGG

Kodeki audio: MPEG-4, MP3

Formaty wideo: MP4, WebM, OGG

Kodeki wideo: H.264/MPEG-4, AVC, VP8, VP9

Dostępne formaty i kodeki: https://en.wikipedia.org/wiki/HTML_video

```
<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>
```

Zad. 3.2 Adaptacyjne strumieniowanie danych multimedialnych z wykorzystaniem standardu MPEG-DASH

Przechwycone pakiety:

The image shows a Wireshark packet capture of an HTTP session. The top pane displays a list of packets, including TCP segments and an HTTP GET request for a DASH manifest. The middle pane shows the details of the selected packet (HTTP GET), including the request line, headers, and body. The bottom pane shows the raw packet data in hexadecimal and ASCII.

| No. | Time | Source | Destination | Protocol | Length | Info |
|-------|----------|----------------|----------------|----------|--------|---|
| 14076 | 3.988176 | 192.168.11.166 | 192.168.11.78 | TCP | 54 | 59551 → 80 [ACK] Seq=2294233 Ack=2559 Win=501 Len=1460 [TCP PDU reassembled in 14081] |
| 14077 | 3.988201 | 192.168.11.166 | 192.168.11.78 | TCP | 54 | 59551 → 80 [ACK] Seq=2295693 Ack=2559 Win=255 Len=0 |
| 14078 | 3.988343 | 192.168.11.166 | 192.168.11.78 | TCP | 54 | 59551 → 80 [ACK] Seq=2295693 Ack=2559 Win=501 Len=1460 [TCP PDU reassembled in 14081] |
| 14079 | 3.988441 | 192.168.11.166 | 192.168.11.78 | TCP | 54 | 59551 → 80 [ACK] Seq=2297153 Ack=2559 Win=501 Len=1460 [TCP PDU reassembled in 14081] |
| 14080 | 3.988557 | 192.168.11.166 | 192.168.11.166 | TCP | 54 | 59551 → 80 [ACK] Seq=2298613 Ack=2559 Win=255 Len=0 |
| 14081 | 3.988644 | 192.168.11.166 | 192.168.11.78 | HTTP | 382 | HTTP/1.1 200 OK |
| 14082 | 3.998687 | 192.168.11.166 | 192.168.11.78 | HTTP | 487 | GET /video/video/sintel-dash/sintel_trailer_128k_dash11.m4s HTTP/1.1 |
| 14083 | 3.999438 | 192.168.11.166 | 192.168.11.78 | TCP | 54 | 59551 → 80 [ACK] Seq=2298941 Ack=2992 Win=501 Len=1460 [TCP PDU reassembled in 14115] |
| 14084 | 3.999664 | 192.168.11.166 | 192.168.11.78 | TCP | 54 | 59551 → 80 [ACK] Seq=2300401 Ack=2992 Win=501 Len=1460 [TCP PDU reassembled in 14115] |
| 14085 | 3.999664 | 192.168.11.166 | 192.168.11.78 | TCP | 54 | 59551 → 80 [ACK] Seq=2301861 Ack=2992 Win=501 Len=1460 [TCP PDU reassembled in 14115] |
| 14086 | 3.999742 | 192.168.11.166 | 192.168.11.78 | TCP | 54 | 59551 → 80 [ACK] Seq=230292 Ack=2303321 Win=255 Len=0 |
| 14087 | 3.999771 | 192.168.11.166 | 192.168.11.78 | TCP | 54 | 59551 → 80 [ACK] Seq=2303321 Ack=2992 Win=501 Len=1460 [TCP PDU reassembled in 14115] |
| 14088 | 3.999892 | 192.168.11.166 | 192.168.11.78 | TCP | 54 | 59551 → 80 [ACK] Seq=2304781 Ack=2992 Win=501 Len=1460 [TCP PDU reassembled in 14115] |
| 14089 | 3.999914 | 192.168.11.166 | 192.168.11.78 | TCP | 54 | 59551 → 80 [ACK] Seq=2306241 Ack=2992 Win=501 Len=1460 [TCP PDU reassembled in 14115] |
| 14090 | 4.000013 | 192.168.11.166 | 192.168.11.78 | TCP | 54 | 59551 → 80 [ACK] Seq=2307701 Ack=2992 Win=501 Len=1460 [TCP PDU reassembled in 14115] |
| 14091 | 4.000150 | 192.168.11.166 | 192.168.11.78 | TCP | 54 | 59551 → 80 [ACK] Seq=2309161 Ack=2992 Win=501 Len=1460 [TCP PDU reassembled in 14115] |
| 14092 | 4.000182 | 192.168.11.166 | 192.168.11.78 | TCP | 54 | 59551 → 80 [ACK] Seq=2310621 Ack=2992 Win=501 Len=1460 [TCP PDU reassembled in 14115] |
| 14093 | 4.000290 | 192.168.11.166 | 192.168.11.78 | TCP | 54 | 59551 → 80 [ACK] Seq=2312081 Ack=2992 Win=501 Len=1460 [TCP PDU reassembled in 14115] |
| 14094 | 4.000388 | 192.168.11.166 | 192.168.11.78 | TCP | 54 | 59551 → 80 [ACK] Seq=2313541 Ack=2992 Win=501 Len=1460 [TCP PDU reassembled in 14115] |
| 14095 | 4.000448 | 192.168.11.166 | 192.168.11.78 | TCP | 54 | 59551 → 80 [ACK] Seq=2315001 Ack=2992 Win=501 Len=1460 [TCP PDU reassembled in 14115] |
| 14096 | 4.000507 | 192.168.11.166 | 192.168.11.78 | TCP | 54 | 59551 → 80 [ACK] Seq=2316461 Ack=2992 Win=501 Len=1460 [TCP PDU reassembled in 14115] |
| 14097 | 4.000628 | 192.168.11.166 | 192.168.11.78 | TCP | 54 | 59551 → 80 [ACK] Seq=2317921 Ack=2992 Win=501 Len=1460 [TCP PDU reassembled in 14115] |
| 14098 | 4.000660 | 192.168.11.166 | 192.168.11.78 | TCP | 54 | 59551 → 80 [ACK] Seq=2319381 Ack=2992 Win=501 Len=1460 [TCP PDU reassembled in 14115] |
| 14099 | 4.000771 | 192.168.11.166 | 192.168.11.78 | TCP | 54 | 59551 → 80 [ACK] Seq=2320841 Ack=2992 Win=501 Len=1460 [TCP PDU reassembled in 14115] |
| 14100 | 4.000874 | 192.168.11.166 | 192.168.11.78 | TCP | 54 | 59551 → 80 [ACK] Seq=2322301 Ack=2992 Win=501 Len=1460 [TCP PDU reassembled in 14115] |
| 14101 | 4.000894 | 192.168.11.166 | 192.168.11.78 | TCP | 54 | 59551 → 80 [ACK] Seq=2323761 Ack=2992 Win=501 Len=1460 [TCP PDU reassembled in 14115] |
| 14102 | 4.000999 | 192.168.11.166 | 192.168.11.78 | TCP | 54 | 59551 → 80 [ACK] Seq=2325221 Ack=2992 Win=501 Len=1460 [TCP PDU reassembled in 14115] |

Frame 14081: 382 bytes on wire (3056 bits), 382 bytes captured (3056 bits) on interface vDevice\NPF_{8...} Ethernet II, Src: ASUSTekCOMPU_e3:85:ac (08:1b:f1:c3:85:ac), Dst: RealtekSemi_68:03:25 (08:e0:4c:68:03:25) Internet Protocol Version 4, Src: 192.168.11.166, Dst: 192.168.11.78 Transmission Control Protocol, Src Port: 80, Dst Port: 59551, Seq: 2298613, Ack: 2559, Len: 328 [23 Reassembled TCP Segments (32448 bytes): #14048(1460), #14049(1460), #14051(1460), #14052(1460), #1 Hypertext Transfer Protocol

HTTP/1.1 200 OK\r\nServer: nginx/1.16.0 (Ubuntu)\r\nDate: Mon, 14 Apr 2025 15:26:05 GMT\r\nContent-Type: application/octet-stream\r\nContent-Length: 32183\r\nLast-Modified: Tue, 30 May 2023 08:59:11 GMT\r\nConnection: keep-alive\r\nEtag: "6475b9df-7db7"\r\nAccept-Ranges: bytes\r\n\r\n[Time since request: 0.007919000 seconds]\n[Request URI: /video/video/sintel-dash/sintel_trailer_128k_dash10.m4s]

The image shows the Chrome DevTools Network tab. The top pane displays a timeline of network requests. The middle pane shows a list of requests, including the DASH manifest and various video segments. The bottom pane shows the details of the selected request (sintel_trailer_128k_dash1.m4s), including the response headers and body.

| Name | Method | Status | Type | Initiator | Size | Time | Waterfall |
|----------------------------------|--------|--------|----------|-----------------------|---------|--------|-----------|
| dash.html | GET | 200 | document | Other | 689 B | 5 ms | |
| dash-2.0.all.min.js | GET | 200 | script | dash.html:6 | 366 kB | 38 ms | |
| sintel.mpd | GET | 200 | xhr | ManifestUpdater.js:18 | 2.5 kB | 330 ms | |
| sintel_set1_init.mp4 | GET | 200 | xhr | FragmentLoader.js:246 | 1.1 kB | 8 ms | |
| sintel_trailer_128k_dashinit.mp4 | GET | 200 | xhr | FragmentLoader.js:246 | 1.1 kB | 8 ms | |
| sintel_trailer_128k_dash1.m4s | GET | 200 | xhr | FragmentLoader.js:246 | 32.5 kB | 22 ms | |
| sintel_trailer_480_2M_dash1.m4s | GET | 200 | xhr | FragmentLoader.js:246 | 1.0 MB | 191 ms | |
| sintel_trailer_128k_dash2.m4s | GET | 200 | xhr | FragmentLoader.js:246 | 32.9 kB | 26 ms | |
| sintel_trailer_128k_dash3.m4s | GET | 200 | xhr | FragmentLoader.js:246 | 32.7 kB | 27 ms | |
| sintel_trailer_128k_dash4.m4s | GET | 200 | xhr | FragmentLoader.js:246 | 32.7 kB | 27 ms | |
| sintel_trailer_128k_dash5.m4s | GET | 200 | xhr | FragmentLoader.js:246 | 32.6 kB | 25 ms | |
| sintel_trailer_128k_dash6.m4s | GET | 200 | xhr | FragmentLoader.js:246 | 32.7 kB | 24 ms | |
| sintel_trailer_128k_dash7.m4s | GET | 200 | xhr | FragmentLoader.js:246 | 32.3 kB | 20 ms | |
| sintel_trailer_480_2M_dash2.m4s | GET | 200 | xhr | FragmentLoader.js:246 | 490 kB | 78 ms | |
| sintel_trailer_480_4M_dash2.m4s | GET | 200 | xhr | FragmentLoader.js:246 | 1.6 MB | 160 ms | |
| sintel_set1_init.mp4 | GET | 200 | xhr | FragmentLoader.js:246 | 1.1 kB | 6 ms | |
| sintel_trailer_480_4M_dash3.m4s | GET | 200 | xhr | FragmentLoader.js:246 | 563 kB | 57 ms | |

64 / 71 requests | 22.5 MB / 22.5 MB transferred | 22.5 MB / 22.5 MB resources | Finish: 27.12 s | DOMContentLoaded: 86 ms | Load: 88 ms

Jakie komunikaty protokołu HTTP zostały użyte podczas transmisji:

Tak samo jak poprzednio wykorzystano metodę ‘GET’

Teraz mamy ‘Content-Range’, ile bajtów filmu pobrano w danym kawałku

| | |
|---|--|
| ▼ General | |
| Request URL: | http://192.168.11.166/video/video/sintel-dash/sintel_set1_init.mp4 |
| Request Method: | GET |
| Status Code: | 200 OK |
| Remote Address: | 192.168.11.166:80 |
| Referrer Policy: | strict-origin-when-cross-origin |
| ▼ Response Headers <input type="checkbox"/> Raw | |
| Accept-Ranges: | bytes |
| Connection: | keep-alive |
| Content-Length: | 867 |
| Content-Type: | video/mp4 |

Oszacować średnią i maksymalną przepływność strumienia danych podczas transmisji pliku multimedialnego - wynik należy podać w kbit/s lub Mbit/s

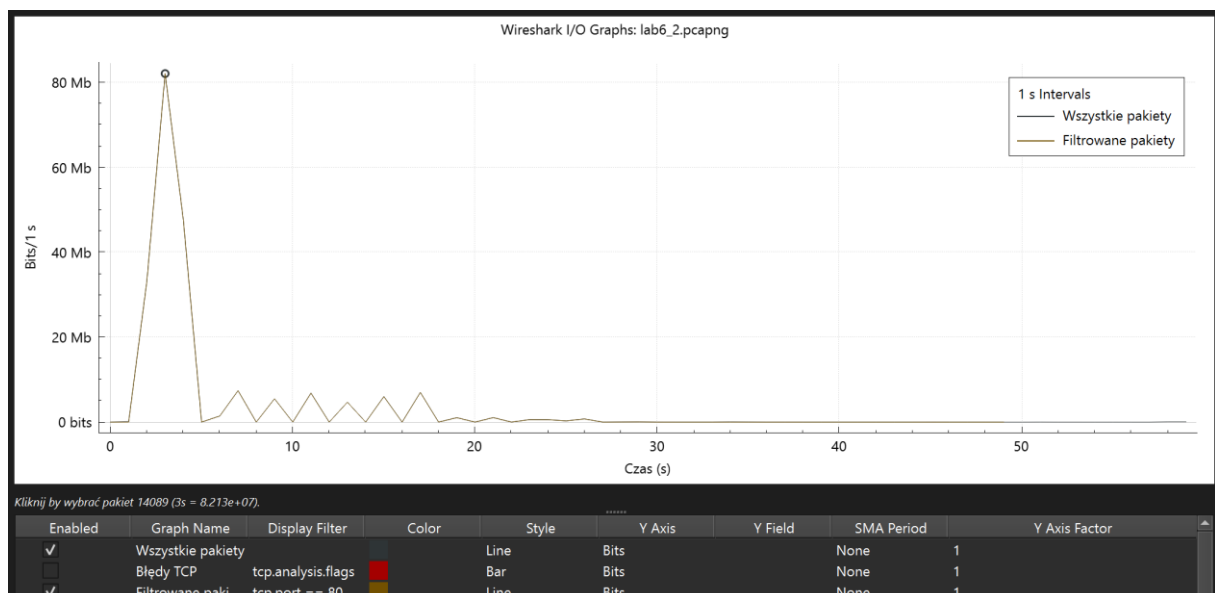
| Statystyki | | |
|---------------------------|--------------|------------------|
| Pomiary | Przechwycone | Wyświetlane |
| Pakiety | 25419 | 25079 (98.7%) |
| Okres czasu, s | 59.674 | 46.794 |
| Średni pps | 426.0 | 535.9 |
| Średni rozmiar pakietu, B | 1017 | 1027 |
| Bajty | 25853756 | 25766510 (99.7%) |
| Średnio bajtów/s | 433 k | 550 k |
| Średnio bitów/s | 3466 k | 4405 k |

25766510 B -> 25 162,6 kB (ze statystyk wyświetlania)

Średnia przepływność: 4301,85 kbit/s

25 162,6 kB / 46,794 = 537,73 kB/s -> **4301,85 kbit/s** (ze statystyk wyświetlania)

Maksymalna przepływność: 82 Mbit/s



Odczytać deskryptor danych multimedialnych (MPD), na jego podstawie określić format danych multimedialnych, liczbę reprezentacji i segmentów.

Plik sintel.mpd:

```
1 <?xml version="1.0"?>
2 <!-- MPD file generated with GPAC version 0.5.2-DEV-rv7version: 0.5.2-426-gc5ade4dfags-5 at 2023-05-30T08:59:11.103Z-->
3 <MPD xmlns="urn:mpeg:dash:schema:mpd:2011" minBufferTime="PT1.500s" type="static" mediaPresentationDuration="PT0H0M52.209S" maxSegmentDuration="PT0H0M2.000S" profile="urn:mpeg:dash:profile:isoff-live:2011:http://dashif.org/guidelines/dash264"/>
4 <ProgramInformation>
5   <Title>sintel.mpd generated by GPAC/71149</Title>
6 </ProgramInformation>
7
8 <Period duration="PT0H0M52.209S">
9   <AdaptationSet segmentAlignment="true" bitstreamSwitching="true" maxWidth="854" maxHeight="480" maxFrameRate="24" par="854:480" lang="und">
10     <SegmentTemplate initialization="sintel_seti_init.mp4"/>
11     <Representation id="1" mimeType="video/mp4" codecs="avc3.64001e" width="854" height="480" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="832754">
12       <SegmentTemplate timescale="12288" media="sintel_trailer_480_1M_dashNumber9.m4s" startNumber="1" duration="22598"/>
13     </Representation>
14     <Representation id="2" mimeType="video/mp4" codecs="avc3.64001e" width="854" height="480" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="1628044">
15       <SegmentTemplate timescale="12288" media="sintel_trailer_480_2M_dashNumber9.m4s" startNumber="1" duration="22598"/>
16     </Representation>
17     <Representation id="3" mimeType="video/mp4" codecs="avc3.64001e" width="854" height="480" frameRate="24" sar="1:1" startWithSAP="1" bandwidth="3228059">
18       <SegmentTemplate timescale="12288" media="sintel_trailer_480_4M_dashNumber9.m4s" startNumber="1" duration="22598"/>
19     </Representation>
20   </AdaptationSet>
21   <AdaptationSet segmentAlignment="true" lang="und">
22     <Representation id="4" mimeType="audio/mp4" codecs="mp4a.40.2" audioSamplingRate="48000" startWithSAP="1" bandwidth="130244">
23       <AudioChannelConfiguration schemeIDURI="urn:mpeg:dash:23003:3:audio_channel_configuration:2011" value="2"/>
24     <SegmentTemplate timescale="48000" media="sintel_trailer_128k_dashNumber9.m4s" startNumber="1" duration="95323" initialization="sintel_trailer_128k_dashinit.mp4"/>
25     </Representation>
26   </AdaptationSet>
27 </Period>
28 </MPD>
```

Format danych: audio – mp4a.40.2, video – avc3.64001e

Liczba reprezentacji: audio – 1, video – 3

Czas trwania segmentu video:

Czas trwania segmentu = duration/timescale = 22598/12288 = 1.839

Liczba segmentów = czas trwania/czas trwania segmentu = 52.209/1.839 = 28.389 -> 29

Czas trwania segmentu audio:

Czas trwania segmentu = duration/timescale = 95323/48000 = 1.986

Liczba segmentów = czas trwania/czas trwania segmentu = 52.209/1.986 = 26.289 -> 27

Zad. 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).

Progresywne pobieranie danych multimedialnych z wykorzystaniem protokołu http:

- pobiera cały film / jego dużą część przed odtworzeniem – dobre, gdyż możemy obejrzeć film bez internetu po jego pobraniu
- nie mamy możliwości zmiany jakości odbieranego filmu – w zależności od przepływności naszego łącza
- wymaga dużego bufora na pobranie części filmu
- słabe zastosowanie do dystrybucji telewizji internetowej i VOD ponieważ:
 - mogłoby występować opóźnienie w filmach na żywo
 - nie jesteśmy w stanie wpływać na jakość podczas przesyłania

Adaptacyjne strumieniowanie danych multimedialnych z wykorzystaniem standardu MPEG-DASH:

- film jest podzielony na mniejsze fragmenty
- dostępne jest kilka jakości – wybierana na podstawie przepływności naszego łącza
- nie wymaga dużego bufora, ściągamy fragmenty dopiero gdy je potrzebujemy
- dobre zastosowanie do dystrybucji telewizji internetowej i VOD ponieważ:
 - programy są dzielone na części, aby pobierać jedynie oglądaną część
 - filmy na żywo mogą być dzielone na fragmenty i wysyłane z małym opóźnieniem
 - każdy fragment może być dostępny w różnych jakościach, która jest dopasowana do przepływności łącza odbiorców