

# Strumieniowanie danych

## 1. Informacje ogólne

Strumieniowanie jako pojęcie odnoszące się do transmisji danych powstało we wczesnych latach osiemdziesiątych. Dopiero pod koniec XX wieku rozpowszechnienie dostępu do sieci oraz zwiększenie jej przepustowości oraz mocy obliczeniowej procesorów sprawiły, że można było mówić o efektywnym strumieniowaniu danych. Follansbee określa strumieniowanie danych jako „ciągły transfer danych z jednego komputera do drugiego w czasie rzeczywistym” przy użyciu pakietów danych. Największą różnicą pomiędzy strumieniowaniem, a pobieraniem danych jest fakt, że przy pobieraniu dane nie mogą zostać odtworzone dopóki pobieranie nie zostanie zakończone. Strumieniowanie pozwala natomiast na odtwarzanie danych w momencie ich odbierania.

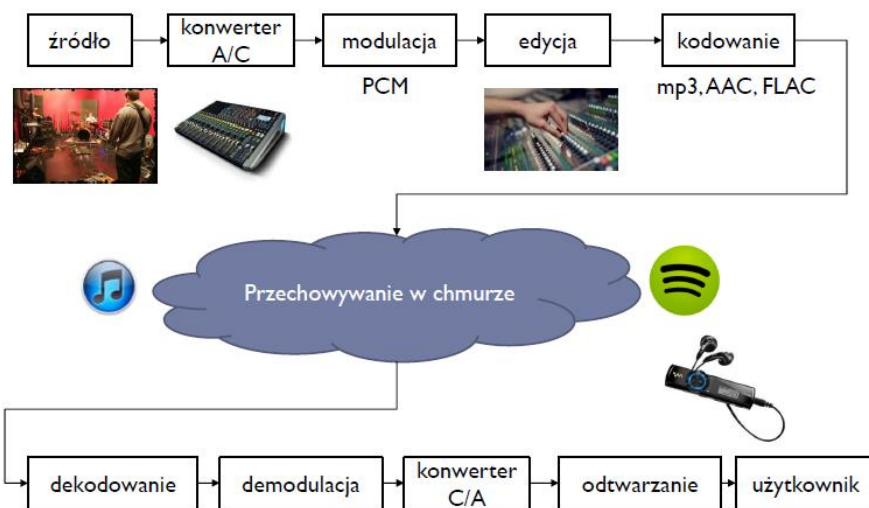
Rozróżniamy dwa sposoby strumieniowania danych:

- na żądanie (on demand) – strumienie są dostępne na serwerze przez dłuższy czas i są gotowe do odtwarzania
- na żywo (live) – dane są dostępne tylko w konkretnym momencie tak jak w przypadku audycji radiowych. Nie ma możliwości przewijania odtwarzanej audycji.

## 2. Rodzaje transmisji wideo

- Transmisja jeden-do-jednego (ang. unicast), w których pojedynczy pakiet danych przesyłany jest od nadawcy do jednego odbiorcy, dokładnie pod jeden adres. Transmisja może być zrealizowana np. w oparciu o protokół RTP. Za pomocą tego protokołu klient kontaktuje się z serwerem. Serwer odpowiada klientowi poprzez RTP, przysyłając informację opisującą materiał wideo jako sesję strumieniowania. Sesja strumieniowania może składać się z jednego lub więcej strumieni, np. strumienia audio i strumienia wideo. Serwer przekazuje klientowi informację, jak dużo strumieni powinien on oczekiwać oraz podaje dokładne dane dotyczące m.in. typu przesyłanych danych i zastosowanego kodeka.
- Transmisja jeden-do-wielu (ang. multicast) składa się z pojedynczego pakietu danych, który adresowany jest do grupy odbiorców, przy czym ruter docelowy może wysyłać pakiety nie tylko do użytkowników końcowych, ale także do innych ruterów.
- Transmisje rozgłoszeniowe (ang. broadcast) składają się z pojedynczego pakietu danych, kopiowanego i przesyłanego do wszystkich węzłów sieciowych. Pakiet jest adresowany przez węzeł źródłowy specjalnym adresem rozsyłającym, a następnie przesyłany do sieci, która tworzy kopie pakietu i wysyła je do każdego węzła sieci.

## 3. Transmisja cyfrowa



Rysunek 1. Przykład transmisji cyfrowej dźwięku.

Czynniki wpływające na jakość transmisji cyfrowej:

- a. Kompresja - dane multimedialne są na ogół transmitowane po kompresji, największe współczynniki uzyskuje się przy kompresji stratnej, która usuwa część informacji z sygnału, przykładami kompresji są np. kodeki mp3, AAC, G.711 dla dźwięku i MPEG-2 lub H.264 dla obrazu, kluczowym parametrem kodeka jest przepływność bitowa, kompresja ma obecnie największy wpływ na jakość sygnału multimedialnego.
- b. Opóźnienie w transmisji - na opóźnienie wpływają następujące czynniki:
  - kodeki (opóźnienie pakietyzacji)
  - prędkość transmisji (opóźnienie transportu)
  - odległość między nadajnikiem a odbiornikiem (czas propagacji)
- c. Jitter - zjawisko polegające na błędach w czasie odebrania (odczytu) transmitowanych danych, powszechne w sieciach IP, gdzie zdarza się, że datagramy docierają do odbiornika ze zmiennym opóźnieniem lub w błędnej kolejności. Rozwiązaniem jest bufor (*jitterbuffer*), w którym przechowywanych jest kilka pakietów, a następnie ustawiane są we właściwej kolejności. Ceną jest jednak wprowadzenie dodatkowego opóźnienia.
- d. Utrata pakietów - w sieciach IP nie wszystkie pakiety docierają do adresata, powodem może być przeciążenie łącza, zbyt duża ilość kolizji pakietów w sieci lokalnej, pakiety mogą gubić się pojedynczo „packetgap” lub paczkami „burst”. W przypadku transmisji multimedialnej skutkuje to przerwami w sygnale audio i video.

#### 4. Protokoły transmisji danych

Idealna transmisja multimedii:

- a. Brak strat – pakiet dociera do miejsca przeznaczenia
- b. Niskie opóźnienia – możliwie szybko dociera do miejsca przeznaczenia
- c. Niewielki narzut protokołu – nagłówki dodawane przez protokół są zanedbywalnie małe w stosunku do danych (payload)
- d. Stabilność transmisji – dane są odbierane z tą samą prędkością z jaką są nadawane – brak jitteru

#### 4.1 Protokoły transmisji UDP (User Datagram Protocol) i TCP (Transmission Control Protocol)

UDP	TCP
Nagłówek 8B	Nagłówek 20B
Bezpołączeniowy	Połączeniowy
Brak mechanizmu retransmisji i kontroli przepływu: <ul style="list-style-type: none"> <li>· Mniejsze opóźnienie</li> <li>· Straty</li> <li>· Zaburzenia sekwencyjności</li> </ul>	Mechanizm retransmisji i kontroli przepływu: <ul style="list-style-type: none"> <li>· Większe opóźnienia</li> <li>· Brak strat</li> <li>· Zachowanie sekwencyjności</li> </ul>
Niewielki nakład pracy w urządzeniach pośredniczących	Transmisja sesyjna – urządzenia pośredniczące utrzymują maszynę stanu dla każdego połączenia
Przekazywanie pakietów odbywa się w sposób bez sesyjny – niewielkie opóźnienia	Mechanizmy kontroli transmisji wymuszają retransmisję utraconych pakietów i mogą ograniczać pasmo – efektem są znaczne opóźnienia
Każdy pakiet wysyłany jest tylko jeden raz	Transmisja z potwierdzeniem – kolejny pakiet wysyłany jest po potwierdzeniu odbioru poprzedniego
Transmisja bez potwierżeń	Transmisja jest ponawiana w przypadku braku potwierdzenia w określonym czasie
Uszkodzenie fragmentu sieci lub tłok w urządzeniu pośredniczącym prowadzą do utraty pakietu	Przewlekłe utraty pakietów prowadzą do zerwania sesji

Narzut protokołu (nagłówek warstwy sieciowej IP, nagłówek warstwy transportowej UDP lub TCP, nagłówek warstwy aplikacyjnej RTP, dane ramki Payload):

**Przykład: bitrate 16kbps**

**ramka 40 B  
+ nagłówek RTP 52B  
+nagłówek 60B  
+nagłówek IP 80B**

**Rzeczywisty bitrate 32kbps**

## 4.2 Protokoły Voice/Video over Data

- VoIP (Voice/Video over Internet Protocol)
- Telefonia IP
- Usługi: głosowe, fax, sms, voice mail
- VoIP wykorzystuje protokoły kontroli sesji i protokoły sygnalizacyjne, potrzebne do zestawienia, rozłączenia i kontroli połączeń
- Wykorzystuje takie protokoły jak: H.323, SIP, RTP, RTCP, SRTP
- Wykorzystuje kodeki mowy takie jak: G.711, G.722 i inne

## 4.3 SIP – Session Initiation Protocol

- Protokół komunikacyjny wykorzystywany do sygnalizacji i kontrolowania podczas sesji wymiany danych multimedialnych
- Negocjacja kodeków
- Wykorzystywany w VoIP, serwisach streamingowych
- Format wiadomości jak w HTTP
- Dostępne szyfrowanie payloadu

## 4.4 RTP – Real-time Transport Protocol

- Otwarty protokół
- Wykorzystywane do dostarczania strumieni audio i video poprzez sieć IP w czasie rzeczywistym
- Wykorzystuje UDP jako protokół warstwy transportowej
- Porządkowanie pakietów
- Wykrywanie strat pakietów
- Sesja RTP jest nawiązywana dla każdego ze strumieni
- Strumienie audio i video mogą używać odrębnych sesji RTP
- Odbiorca może zrezygnować z poszczególnych strumieni (różne języki w filmie, wybór tylko audio lub tylko video)
- Wykorzystuje SIP do zestawienia sesji

## 4.5 RTCP – Real-time Transport Control Protocol

- Synchronizacja strumieni
- Raportowanie QoS - oznacza szereg metryk składających się na *jakość* usług telekomunikacyjnych, w postaci zdefiniowanych parametrów technicznych. (stałą, gwarantowaną przepustowość łącza, scharakteryzowany stopień synchronizacji strumieni, gwarantowany poziom usług (np. ilu użytkowników naraz można obsłużyć), gwarantowane opóźnienie w wyznaczonych granicach)
- Raportowanie o stanie użytkowników w sesji
- Renegocjacja jakości strumienia (kodek, bitrate)
- Raportowanie o danych użytkowników w sesji (imię, adres, numer)

## 4.6 WebRTC

- Otwarty standard przesyłania multimediów w czasie rzeczywistym przez przeglądarki internetowe
- Brak potrzeby instalowania dodatkowych pluginów
- Wykorzystuje HTML5 i javascript

## 4.7 Chmury jako serwery multimediów

- Możliwość stworzenia multimediów za darmo, np.: w chmurze obliczeniowej Amazon AWS
- Zysk – architektura rozproszona – wiele punktów dostępowych – krótsza droga pakietów
- Chmura Amazona wykorzystywana aktualnie jest przez Netflix.

### Literatura:

[1] John G. Apostolopoulos, Wai-tian Tan, Susie J. Wee - „Video Streaming: Concepts, algorithms, and Systems”

[2] K.Choroś - „Strumieniowanie plików wideo w systemach nauczania na odległość”

## 5. Instrukcje do użytego oprogramowania

### Strumieniowanie - VLC media player:

Odtwarzacz multimedialny rozwijany przez VideoLAN, rozpowszechniany na licencji GPL, może być także używany jako serwer w sieciach o dużym przesyśle. Może odtwarzać wiele rodzajów plików bez konieczności instalacji dodatkowych kodeków. Obsługuje on takie formaty jak: Ogg, MPEG, DivX, MP3 i inne.

- Oficjalna strona programu : <http://www.videolan.org/vlc/>
- Video LAN wiki: [http://wiki.videolan.org/Main\\_Page](http://wiki.videolan.org/Main_Page)
- Video LAN dokumentacja: <http://www.videolan.org/doc/>
- Strumieniowanie VLC - [http://wiki.videolan.org/Documentation:Streaming\\_HowTo](http://wiki.videolan.org/Documentation:Streaming_HowTo)

### Strumieniowanie (na serwerze):

1. *Plik* -> *Strumieniuj* -> w zakładce plik, wybór pliku, "Dodaj", wybór pliku do strumieniowania
2. *Strumieniuj (na dole)* -> "*Następny*" -> "*Cele, nowe cel*" (ustawić protokół) -> *Dodaj* -> ustawić port, ewentualnie adres i nazwę strumienia:
  - a. dla protokołu HTTP - tylko przycisk „*Dodaj*”, (bez adresu IP, bez nazwy strumienia, port 8080)
  - b. dla UDP – przycisk *Dodaj* i w polu adresu podajemy adres rozgłoszeniowy 224.0.0.1,
  - c. dla RTP – (jako cel wybrać „RTP/MPEG Transport Stream”) przycisk *Dodaj*, w polu adresu adres rozgłoszeniowy (np. 224.0.0.1), port (np. 5004) i nazwę strumienia (np.: nowy)
3. W "*Opcje transkodowania*" ustawić profil, wybrać odpowiednie kodeki.
4. *Strumieniuj (na dole)*

### Odbieranie strumienia (na klientach)

1. *Plik* -> *Otwórz strumień z sieci* -> *Zakładka "sieć"*, podaj adres URL sieci ->
  - a. Dla HTTP: **`http://adres_ip_serwera:nr_portu`** (nr portu to dla http standardowo 8080), jako adres IP podajemy adres serwera, który udostępni strumień
  - b. Dla UDP: **`udp://@:1234`**
  - c. Dla RTP: **`rtp://@adres_rozgloszeniowy:nr_portu/nazwa_strumienia`**
2. *Odtwarzaj*

Pliki filmowe do pobrania na stronie: <http://heavy.metal.agh.edu.pl/>

## Pomiar obciążenia procesora:

### a) Skrypt Bash:

(wykorzystywany w instrukcji w trybie stacjonarnym – dla laboratorium realizowanego na linuxie)

Przedstawiony skrypt służy do monitorowania obciążenia procesora dla wybranego procesu (VLC). Obciążenie jest przechwytywane łącznie 30 razy, co 1 sekundę od momentu uruchomienia skryptu i zapisywane do pliku.

Otrzymane wyniki mają posłużyć do wygenerowania wykresów zużycia procesora w jednostce czasu (czas od 1 sekundy do 30 sekundy OŚ X, OŚ Y – zużycie procesora).

Skrypt ściągamy ze strony <http://heavy.metal.agh.edu.pl/>

Po ściągnięciu skryptu należy nadać mu uprawnienia do wykonania poleceniem: **chmod a+x skrypt**

Skrypt należy uruchomić w oknie terminalu komendą: **./skrypt nazwa\_pliku\_do\_zapisu**

(uruchamiamy dopiero wtedy gdy program VLC jest uruchomiony !!)

```
#!/bin/bash
plikName=$1
x=1
while [ $x -le 30 ]
do
sleep 1
ps --no-headers -o '%cpu' -C vlc >> $plikName ;
x=$(( $x + 1 ))
done
```

### b) pidstat

Do pomiaru obciążenia procesora wykorzystujemy program pidstat (z pakietu sysstat)

Na początku należy sprawdzić numer PID, jaki został przydzielony dla procesu odtwarzacza VLC. Można to sprawdzić na wiele sposobów, np. za pomocą poleceń **top**, **htop** lub **ps**.

```
sh-4.2$ ps x | grep vlc
3580 ?        Sl      0:40 /usr/bin/vlc --started-from-file
3983 pts/0    S+     0:00 grep vlc
```

Po uzyskaniu informacji po numerze PID procesu VLC średnie zużycie procesora można zmierzyć za pomocą polecenia **pidstat**

```
sh-4.2$ pidstat -p 3580 1 5
Linux 3.10.0-1062.12.1.el7.x86_64 (WL40426)      20.09.2021      _x86_64_      (6 CPU)

09:45:39      UID      PID      %usr  %system  %guest      %CPU   CPU   Command
09:45:40    16777234    3580     25,00    10,00     0,00    35,00     1   vlc
09:45:41    16777234    3580     16,00     3,00     0,00    19,00     1   vlc
09:45:42    16777234    3580     24,00     2,00     0,00    26,00     1   vlc
09:45:43    16777234    3580     23,00     3,00     0,00    26,00     1   vlc
09:45:44    16777234    3580     15,00     3,00     0,00    18,00     1   vlc
Średnia:    16777234    3580     20,60     4,20     0,00    24,80     -   vlc
```

Kolejne parametry polecenia to:

- numer PID procesu, który chcemy monitorować, (opcja **-p**)
- odstępy pomiędzy kolejnymi pomiarami (przedostatni parametr, w sekundach)
- liczba pomiarów po której program ma zakończyć działanie (ostatni parametr)

O ile prowadzący nie wskaże inaczej, w trakcie laboratorium należy każdorazowo wykonać 10 pomiarów co 2 sekundy, co skutkuje wyliczeniem średniej z 20 sekund strumieniowania.

Bardzo ważne jest, by dla każdego pojedynczego filmu rozpoczynać pomiary w tym samym momencie (np. w 30 sekundzie strumieniowania), gdyż obciążenie procesora różni się w zależności od dynamiki sceny w danym materiale video.

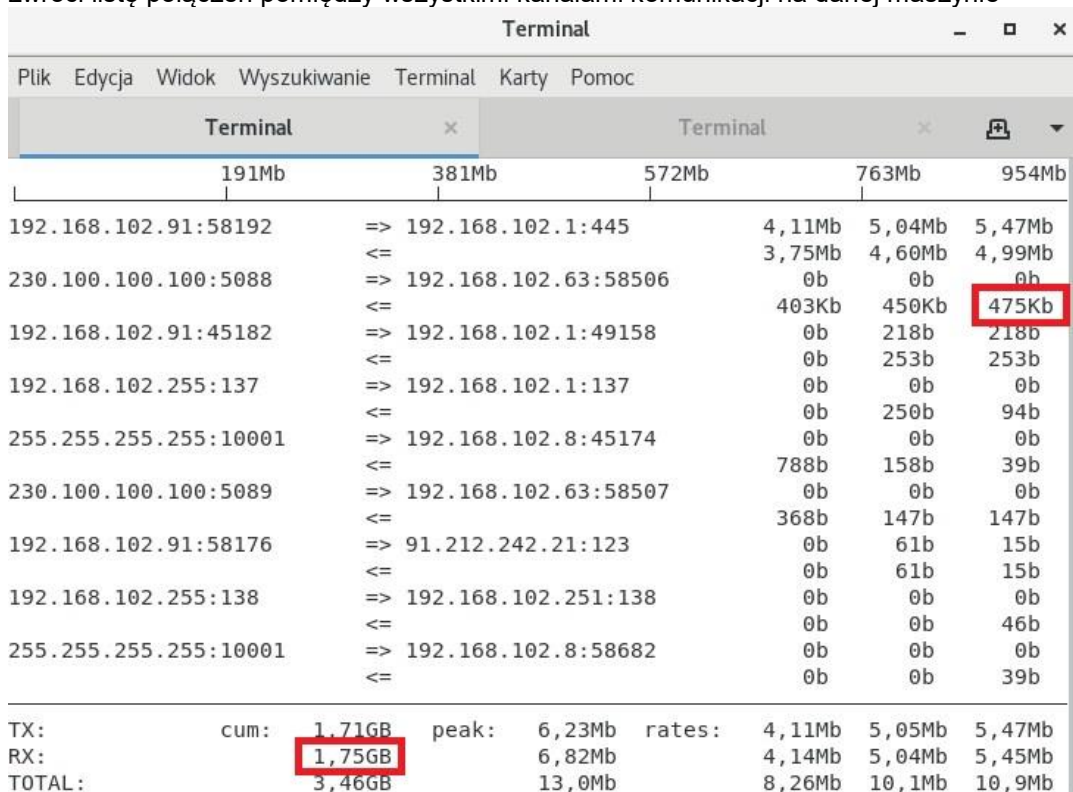
## Pomiar obciążenia sieci:

### a) iftop

Polecenie **iftop** wymaga standardowo użycia trybu administratora, co w warunkach laboratoryjnych możliwe jest przez użycie trybu **sudo**

Standardowo, polecenie uruchomione za pomocą komendy **sudo iftop**

zwróci listę połączeń pomiędzy wszystkimi kanałami komunikacji na danej maszynie



The screenshot shows a terminal window titled "Terminal" with a menu bar (Plik, Edycja, Widok, Wyszukiwanie, Terminal, Karty, Pomoc). The main content displays the output of the iftop command, showing network traffic between various IP addresses. The output is organized into columns for source and destination IP addresses, and then into columns for traffic rates (Tx, Rx, Total). The values are in Mb or Kb. A red box highlights the value "475Kb" in the Rx column for the connection between 192.168.102.91 and 192.168.102.63. At the bottom, summary statistics are shown for TX, RX, and TOTAL, with "1,75GB" highlighted in a red box in the RX row.

Source IP	Direction	Destination IP	Rate 1	Rate 2	Rate 3
192.168.102.91:58192	=>	192.168.102.1:445	4,11Mb	5,04Mb	5,47Mb
	<=		3,75Mb	4,60Mb	4,99Mb
230.100.100.100:5088	=>	192.168.102.63:58506	0b	0b	0b
	<=		403Kb	450Kb	475Kb
192.168.102.91:45182	=>	192.168.102.1:49158	0b	218b	218b
	<=		0b	253b	253b
192.168.102.255:137	=>	192.168.102.1:137	0b	0b	0b
	<=		0b	250b	94b
255.255.255.255:10001	=>	192.168.102.8:45174	0b	0b	0b
	<=		788b	158b	39b
230.100.100.100:5089	=>	192.168.102.63:58507	0b	0b	0b
	<=		368b	147b	147b
192.168.102.91:58176	=>	91.212.242.21:123	0b	61b	15b
	<=		0b	61b	15b
192.168.102.255:138	=>	192.168.102.251:138	0b	0b	0b
	<=		0b	0b	46b
255.255.255.255:10001	=>	192.168.102.8:58682	0b	0b	0b
	<=		0b	0b	39b
<hr/>					
TX:	cum:	1,71GB	peak:	6,23Mb	rates: 4,11Mb 5,05Mb 5,47Mb
RX:		1,75GB		6,82Mb	4,14Mb 5,04Mb 5,45Mb
TOTAL:		3,46GB		13,0Mb	8,26Mb 10,1Mb 10,9Mb

W celu zwiększenia czytelności aplikacji, należy po jej uruchomieniu kliknąć:

- „p” – wyświetlone zostaną porty, na których zachodzi komunikacja
- „n” – wyłączone zostanie mapowanie adresów IP na nazwy serwerów
- „N” – wyłączone zostanie mapowanie numerów portów na nazwy usług

W trakcie realizacji laboratoriów, w celu zwiększenia czytelności zrzutów ekranów, należy zastosować filtry na wybraną komunikację.

Przy strumieniowaniu unicast (http) na porcie 8080 można użyć polecenia

**sudo iftop -f „src port 8080”**

które zaprezentuje jedynie połączenia z portu źródłowego 8080.

Natomiast w trakcie strumieniowania multicast (rtsp) można zastosować filtr na wybrany w laboratorium adres multicastowy:

**sudo iftop -f „host 230.100.100.100”**

(zastępując adres IP na ten wykorzystywany w laboratorium).

Zużycie sieci można odczytać z 3ciej kolumny zużycia sieci dla danego kanału komunikacji (średnia z ostatnich 40 sekund) lub z wartości całkowitej liczby przesłanych danych (tylko w wypadku stosowania filtra na strumieniowanie, oraz z uwzględnieniem czasu monitorowania procesu). W obydwu wypadkach, przy konkretnym materiale wideo monitorowaniu powinny być poddane takie same fragmenty filmu, ze względu na ich różną dynamikę i wynikające z tego różnice w transmisji danych w różnych momentach czasu.

## Scenariusz nr 1: (stacjonarny, realizowany w laboratorium)

### 1. Sprzęt

- 3 komputery PC
- Ćwiczenie wykonywane na Linux CentOS

### 2. Oprogramowanie

- Aplikacja VLC
- Programy do monitorowania:
  - Obciążenia procesora: pidstat lub skrypt
  - Obciążenie sieci: iftop

### 3. Przygotowanie do wykonania ćwiczenia

- Zalogować się na swoje konta systemu Linux
- Sprawdzić czy są zainstalowane polecenia monitorujące (iftop, pidstat)
- Utworzyć na pulpicie folder, w którym będą znajdować się pliki video oraz zrzuty ekranu potrzebne do sprawozdania

### 4. Wykonanie ćwiczenia

- a. Prowadzący zajęcia wyznacza wyjścia, profile i pliki, które zostaną poddane testom
- b. Wyjścia – protokoły (port)
  - a. HTTP (:8080) (unicast)
  - b. RTP (:5004 lub :5005) (multicast)
  - c. UDP (:1234 lub :1235) (multicast)
  - d. Dla RTP i UDP – skonsultować z innymi grupami realizującymi instrukcję Strumieniowanie, aby dwie grupy nie korzystały z tych samych numerów portów !!
- c. Profile strumienia
  - a. H.264 + MP3 (MP4)
  - b. H.265 + MP3 (MP4)
  - c. Theora + Vorbis (OGG)
  - d. MPEG-2 + MP3 (TS)
  - e. Dirac + MP3 (TS)
  - f. własny profil
- d. Testowe pliki video: .....

Pliki do pobrania na stronie: <http://heavy.metal.agh.edu.pl/>

- e. Podczas każdego z testów badamy parametry na komputerze działającym jako serwer (komputer wystawiający strumień do sieci) oraz na jednym z dwóch klientów (komputer odbierający strumień z sieci). Drugi klient obciąża jedynie serwer, ale nie wykonujemy na nim pomiarów.

### 5. Badamy

#### Na serwerze:

1. Uruchamiamy program VLC (z menu „Programy”->”Dźwięk i obraz”)
2. Uruchamiamy pierwsze okno terminala – sprawdzamy PID programu vlc oraz wykonujemy testowe uruchomienie polecenia **pidstat**
3. Uruchamiamy drugie (osobne) okno terminala i wykonujemy testowe uruchomienie polecenia **iftop**.
4. Dla każdego z wariantów strumieniowania wykonywanego w trakcie laboratorium notujemy:
  - a. nazwę pliku + jego rozdzielczość
  - b. parametry wyjścia: protokół, rodzaj strumieniowania (multicast, unicast), adres IP, nr portu, nazwa strumienia
  - c. parametry transkodowania: profil (kodek video, kodek audio, enkapsulacja) - przy definicji strumienia: "opcje transkodowania" ->"edytuj wybrany profil" .
5. Rozpoczynamy strumieniowanie testowych plików dla wariantów protokołu/wyjścia i profili strumienia według instrukcji oprogramowania dla VLC.
6. Podłączamy dwóch klientów do strumienia.



7. Po rozpoczęciu strumieniowania klikamy skrót Ctrl + J -> z zakładki szczegóły kodeka notujemy informację o oryginalnym kodeku i parametrach pliku

8. Po upływie ok. 30 sekund filmu (30 sekunda filmu w programie VLC na serwerze) rozpoczynamy pomiar obciążenia procesora za pomocą polecenia pidstat. Monitorowane dane (średnie zużycie procesora) można zapisywać na kartce papieru, w dokumencie tekstowym na komputerze lub jako zrzut ekranu. W każdym z wariantów należy zwrócić uwagę, by przy mierzonych danych można było jednoznacznie zidentyfikować przeprowadzany wariant testów.

9. W drugiej konsoli uruchamiamy pomiar zużycia sieci (stosując odpowiednie filtry dla odpowiedniego typu strumieniowania).

!!! Obydwa pomiary (obciążenie procesora, zużycie sieci) staramy się wykonywać dla identycznych fragmentów poszczególnych filmów (z uwagi na ich zmienną dynamikę sieci, wpływającą mocno na uzyskiwane rezultaty) !!!

10. Plik wygenerowany przez skrypt oraz zrzut ekranu kopiujemy w znane miejsce, opisujemy je w sposób jednoznaczny aby mieć pewność którego pomiaru dotyczy !!



**Po skończonym pomiarze zamykamy okno programu VLC na serwerze i kliencie, powtarzamy całą procedurę dla innych parametrów / filmów.**

#### Na kliencie:

1. Program VLC oraz dwie konsole uruchamiamy analogicznie jak w punktach 1,2,3 instrukcji dla serwera.
2. Rozpoczynamy odbiór strumieniowanych testowych plików według instrukcji oprogramowania dla VLC.
3. Po rozpoczęciu strumieniowania klikamy w oknie VLC skrót Ctrl + J -> z zakładki szczegóły kodeka notujemy kodek i parametry pliku.

4. Pomiary obciążenia procesora i zużycia sieci wykonujemy analogicznie jak opisano w punktach 8,9 w instrukcji dla serwera.

5. Notujemy dodatkowo parametry dotyczące bitrate strumienia (wejściowy bitrate, bitrate treści) dostępne w oknie Statystyk VLC i porównujemy je z otrzymanymi za pomocą polecenia *iftop*.

6. Dodatkowo notujemy subiektywną jakość obrazu (opóźnienia, artefakty, czas buforowania, jakość obrazu)

7. Plik wygenerowany przez skrypt oraz zrzut ekranu kopiujemy w znane miejsce, opisujemy je w sposób jednoznaczny aby mieć pewność którego pomiaru dotyczy !!

Informacje pliku multimedialnego

Ogólne Metadane Kodek Statystyki

Aktualne statystyki pliku multimedialnego / strumienia

Dźwięk	
Zdekodowane	883 bloki
Odtworzone	883 bufor(yl)ó
Zgubione	0 bufor(yl)ó
Obraz	
Zdekodowane	511 bloki
Wyświetlone	498 klatki
Zgubione	4 klatki
Wejście/Wczytane	
Rozmiar danych pliku multimedialnego	747 KiB
Wejściowy bitrate	
Rozmiar demukowanych danych	172 kb/s
Bitrate treści	609 KiB
165 kb/s	
Odrzucone (uszkodzone)	0
Pominięte (przerwane)	0
Wejście/Zapisaane/Wystane	
Wysłane	0 pakiet(yl)ó
Wysłane	0 KiB
Prędkość transmisji danych	0 kb/s

*Parametry bitrate strumienia*

**Po skończonym pomiarze zamykamy okno programu VLC na serwerze i kliencie, powtarzamy całą procedurę dla innych parametrów / filmów.**



## 6. Dodatkowe uwagi

- Testy wykonujemy dla wszystkich możliwych kombinacji elementów (protokołów, profili, materiałów video) wybranych przez prowadzącego.
- W wypadku nieprawidłowych proporcji wyświetlanego obrazu, zmieniamy go w kliencie przez „**Obraz>stosunek rozmiarów>16:10**”
- Na końcu pracy !!!USUWAMY!!! folder w którym przechowywane były strumieniowane materiały video oraz zrzuty ekranu wykonane podczas laboratorium.
- Przykładowy wpis w tabeli notatek dla jednej serii testów

Protokół	Parametry strumienia	Nazwa i parametry pliku	SERWER	KLIENT
HTTP (:8080)	Video-H.264 + AAC (MP4) przepływowość 300kb/s fps 60 kodek obrazu: H.264 kodek dźwięku: flac, 128kb/s 2 kanały 441000Hz	film1_720.mp4 video: H.264-mpeg4 1020x720 dźwięk: MPEG AAC 48kHz 1536kb/s	Ilość klientów: 1 Wykorzystanie sieci: 3400B/sec Średnie zużycie CPU: 30% Ilość wątków: 8 (intel i7) Uwagi: chwilowe utraty obrazu.....	Wykorzystanie sieci: 720B/sec Średnie zużycie CPU: 42% przepływność wejściowa: 77210kb/s przepływność strumienia: 94321kb/s Uwagi: jakość zadowalająca,.....

- Powyższe notatki należy opracować w osobnych tabelach i opracować wg poniższych wskazówek.
- Na koniec usuwamy wszystkie wygenerowane zrzuty ekranu, oraz ściągnięte pliki !!

## 7. Wyniki pomiarów:

- na podstawie wyników opracować statystycznie i porównać (przy zachowaniu innych jednakowych parametrów)
  - zużycie zasobów klienta i serwera (procesor, sieć),
  - jakość obrazu (artefakty, ilość zgubionych klatek)
- w zależności od
  - zastosowanego protokołu,
  - parametrów kodeka,
  - rozdzielczości materiału video
  - liczby wykorzystywanych klientów
- wykadrować, pomniejszyć i wydrukować odpowiednie zrzuty ekranu (każdy protokół, kodek, materiał video)
- opisać spostrzeżenia, wyciągnąć wnioski

## Scenariusz nr 2 (zdalny)

### Sprzęt:

2 komputery wyposażone w system MS Windows.

(Instrukcje przygotowywane dla MS Windows 10, scenariusz można realizować także na wcześniejszych wersjach systemu)

### Oprogramowanie:

VLC (<https://www.videolan.org/vlc/index.pl.html> )

VPN Hamachi (<https://www.vpn.net> )

### Pliki do strumieniowania:

<http://heavy.metal.agh.edu.pl/>

### Przygotowanie ćwiczenia:

Na wszystkich komputerach biorących udział w laboratorium należy pobrać i zainstalować programy VLC oraz Hamachi.

W wypadku problemów z instalacją i konfiguracją VPNa Hamachi, może być konieczna jego ponowna instalacja poprzedzona wcześniejszym wyłączeniem programów antywirusowych i zapory sieciowej. Sposób zatrzymania ochrony antywirusowej zależy od używanego programu antywirusowego. Zaporę (firewall) w systemie Windows 10 można wyłączyć poprzez:

Start->Ustawienia->Aktualizacja i Zabezpieczenia->Zabezpieczenia Windows->Zapora i ochrona sieci tam wybieramy po kolei Sieci prywatne i Sieci publiczne i wyłączamy zaporę.

lub:

W pasku Wyszukaj wpisujemy *Zapora Windows Defender* uruchamiamy i wybieramy *Włącz/Wyłącz Zaporę Windows Defender* i znowu wyłączamy zaporę dla sieci prywatnych i publicznych.

**Po realizacji ćwiczenia zaleca się ponowne uruchomienie zapory oraz programów antywirusowych.**

VLC instalujemy tradycyjnie jak każdy program.

Hamachi instalujemy w domyślnej lokalizacji na dysku **C:**. Po uruchomieniu Hamachi należy się zarejestrować, podając email oraz hasło (na mailu musimy potwierdzić konto). Po zalogowaniu do Hamachi przechodzimy w tryb online poprzez kliknięcie w ikonę zasilania (kwadratowy przycisk u góry z lewej strony) w oknie programu. Na jednym z komputerów tworzymy nową sieć (*Utwórz nową sieć*): podajemy jej nazwę i hasło dostępu. Na drugim komputerze wybieramy *Dołącz do istniejącej sieci*: podajemy nazwę sieci i hasło dostępu.

W celu sprawdzenia poprawności utworzonego tunelu należy dodatkowo z zakładki programu *Pomoc* wybrać *Diagnostic Tools i Uruchomić test*.

Realizacja ćwiczenia możliwa jest jedynie w wypadku poprawnej realizacji testu na obydwu komputerach. W przypadku wystąpienia błędu należy odczytać informacje diagnostyczne o błędzie (kod błędu, komunikat), i poszukać rozwiązania w oparciu o wyszukiwarkę internetową.

## Przebieg ćwiczenia:

Na jeden z komputerów biorących udział w ćwiczeniu (serwer) należy pobrać materiały do strumieniowania ze strony <http://heavy.metal.agh.edu.pl/>

Numer (wariant) filmu wyznacza prowadzący w trakcie zajęć organizacyjnych: .....

Należy pobrać cztery wersje tego samego materiału nagrane w różnych rozdzielczościach, np.: *film1\_240*, *film1\_480*, *film1\_720* i *film1\_1080*.

Program VLC uruchamiamy na obydwu komputerach. Na serwerze (komputerze na który pobrano materiały video, który będzie udostępniał strumień) strumieniujemy po kolei materiały video postępując zgodnie z instrukcją strumieniowania opisaną w 5 punkcie niniejszej instrukcji.

Jako protokół wybieramy *http* (port i ścieżka bez zmian - 8080).

Profile transmisji do zrealizowania w ramach laboratorium:

1. *H.264 + MP3*,
2. *H.265 + MP3*,
3. *Theora + Vorbis*,
4. *MPEG-2 + MPGA*,
5. *WMV + WMA*.

Każdy plik należy przestrumieniować po protokole *http* z każdym profilem.

Czyli 4 rozdzielczości \* 5 profili \* 1 protokół = 20 serii testów.

Badane pomiary trzeba notować za każdym razem.

## Badane parametry:

a) parametry materiału video

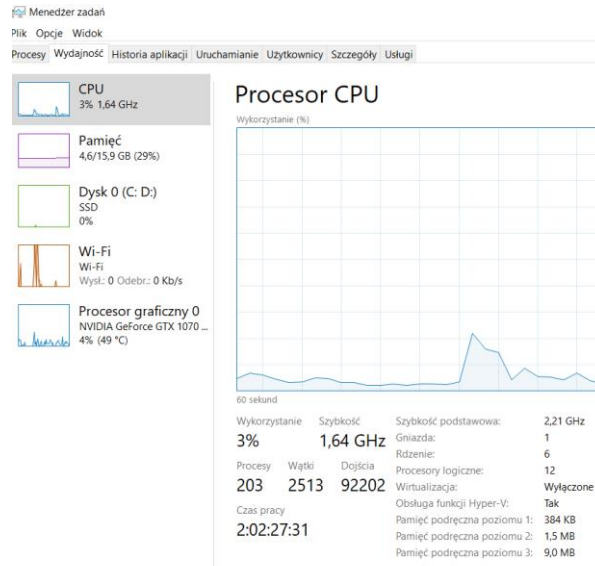
Po uruchomieniu strumieniowania należy (w formie zrzutu ekranu) zanotować informacje o pliku źródłowym (na serwerze) i strumieniu (klient). Należy uruchomić skrót klawiszowy Ctrl+J i w oknie „*Informacje pliku multimedialnego*” w zakładce „*Kodek*” zanotować parametry strumienia (audio i video)

– zarówno na kliencie jak i na serwerze:

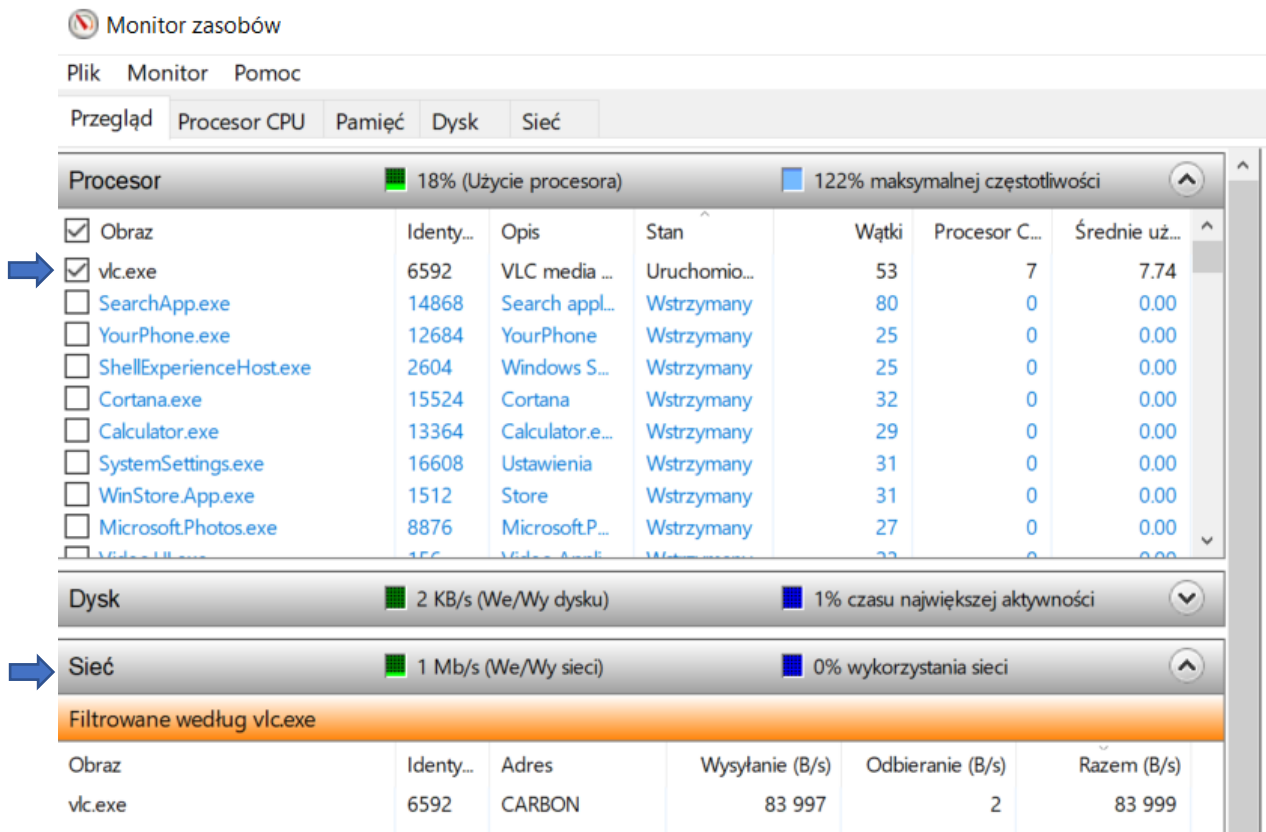


b) pomiar obciążenia procesora oraz zużycia sieci.

W trakcie strumieniowania na obydwu komputerach klikamy **Ctrl+Shift+Esc** co uruchomi *Menedżer zadań*, gdzie w zakładce *Wydajność* wybieramy na samym dole *Otwórz monitor zasobów*.



W oknie „Monitora zasobów” w zakładce *Przegląd*, w części „Procesor” wyszukujemy proces **vlc.exe** i zaznaczamy checkbox po jego lewej stronie, co powinno ustawić filtr na monitorowanie tego procesu:



W trakcie zajęć monitorujemy wartości parametru „Średnie zużycie CPU” z części „Procesor” (na kliencie i serwerze) oraz „Odbieranie (B/s)” (na kliencie) i „Wysyłanie (B/s)” (na serwerze) które obrazują zużycie CPU oraz obciążenie sieci podczas strumieniowania.

Dane do opracowania wniosków zbieramy po ustabilizowaniu strumieniowania (czyli po około 20-40 sekundach od momentu wyświetlenia obrazu na kliencie). Po tym czasie należy rozpocząć monitorowanie parametrów strumienia przez kolejne 30 sekund, w trakcie którego należy obserwować i notować wartości maksymalną, minimalną oraz średnią zarówno dla zużycia sieci jak i obciążenia procesora (zarówno na kliencie jak i na serwerze).

Należy zanotować wartości najmniejsze i największe dla badanych parametrów w danym zakresie czasu. Dodatkowo podczas obserwacji należy wygenerować min. 3 zrzuty ekranu, na podstawie których obliczone zostaną wartości średnie badanych parametrów (przy wyznaczaniu wartości średniej należy dodatkowo uwzględnić zaobserwowaną/zanotowaną wartość średnią).

W zestawieniu końcowym dla każdej serii testów (profil + rozdzielczość) należy podać zaobserwowane (obliczone) wartości: najmniejsza, największa i średnia – zarówno dla zużycia sieci jak i obciążenia procesora

Z uzyskanych wyników (należy umieścić je w czytelnych tabelach) wygenerować wykresy i porównania obrazujące wpływ

- rozdzielczości materiału
- zastosowanego profilu (kodeka audio/video)
- na
- obciążenie procesora
- zużycie sieci.

zarówno po stronie klienta jak i serwera.

Należy dodatkowo zaznaczyć subiektywne uwagi dotyczące jakości transmisji: czy dane wysłane zostały odebrane poprawnie, czy nie było efektów artefaktów graficznych, dodatkowych zacięć czy opóźnień.

Przykładowe notatki dla pojedynczej serii testów mogą wyglądać następująco:

Protokół	Parametry strumienia	Nazwa i parametry pliku	SERWER	KLIENT
HTTP (:8080)	Video-H.264 + AAC (MP4) przepływowość 300kb/s fps 60 kodek obrazu: H.264 kodek dźwięku: flac, 128kb/s 2 kanały 441000Hz	film1_720.mp4 video: H.264-mpeg4 1020x720 dźwięk: MPEG AAC 48kHz 1536kb/s	Ilość klientów: 1 Wykorzystanie sieci: 3400B/sec Średnie zużycie CPU: 30% Ilość wątków: 8 (intel i7) Uwagi: chwilowe utraty obrazu.....	Wykorzystanie sieci: 720B/sec Średnie zużycie CPU: 42% przepływność wejściowa: 77210kb/s przepływność strumienia: 94321kb/s Uwagi: jakość zadowalająca,.....

### Wnioski:

We wnioskach szczegółowych należy umieścić informacje odnośnie wyników poszczególnych serii testów. Wnioski ogólne powinny zawierać podsumowanie porównania poszczególnych serii i analizę wpływu zmieniających się parametrów testów na badane parametry strumienia.