

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.

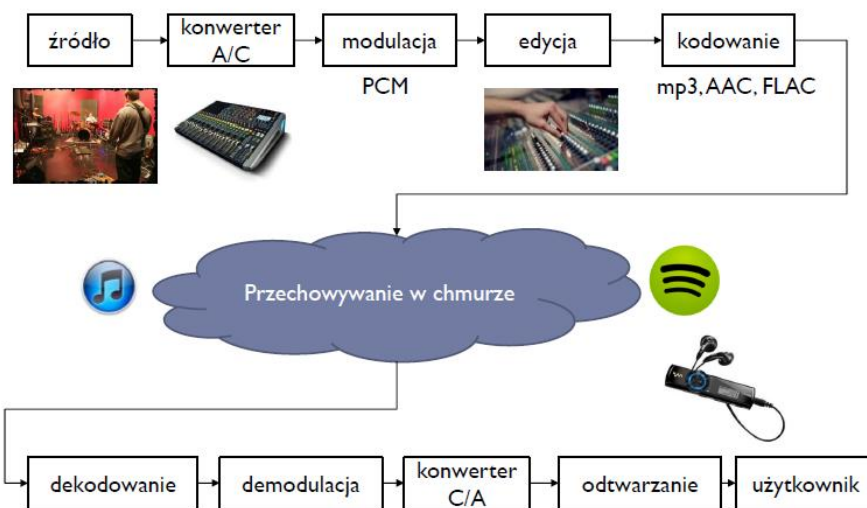
Rozróżniamy dwa sposoby strumieniowania danych:

- 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, przesyłając 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"> · Mniejsze opóźnienie · Straty · Zaburzenia sekwencyjności 	Mechanizm retransmisji i kontroli przepływu: <ul style="list-style-type: none"> · Większe opóźnienia · Brak strat · Zachowanie sekwencyjności
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 potwierdzeń	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 multimediiów w czasie rzeczywistym przez przeglądarki internetowe
- Brak potrzeby instalowania dodatkowych pluginów
- Wykorzystuje HTML5 i javascript

4.7 Chmury jako serwery multimediiów

- Możliwość stworzenia multimediiów 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

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
- Video LAN dokumentacja: <http://www.videolan.org/doc/>
- Strumieniowanie VLC - 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 cele*" (ustawić protokół) -> *Dodaj* -> ustawić port, ewentualnie adres i nazwę strumienia:
 - a. dla protokołu http - tylko przycisk „*Dodaj*”;
 - b. dla UDP – przycisk *Dodaj_i* w polu adresu podajemy adres rozgłoszeniowy 224.0.0.1,
 - c. dla RTP – przycisk *Dodaj*, w polu adresu adres rozgłoszeniowy 224.0.0.1 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://ip_serwera:port*** musimy znać adres IP serwera – komputer puszczający strumień
 - b. Dla UDP: ***udp://@:1234***
 - c. Dla RTP: ***rtp://@:5004/nowy***
2. *Odtwarzaj*

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

Skrypt Bash:

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 !!)

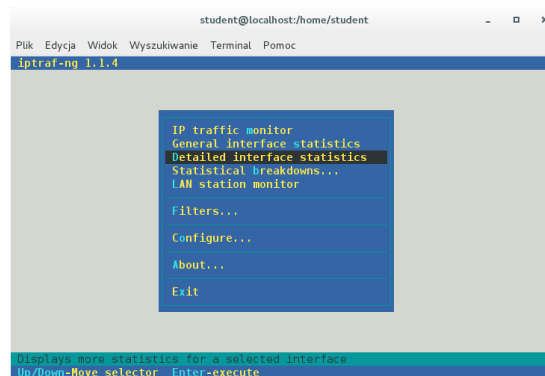
```
#!/bin/bash
plikName=$1
x=1
while [ $x -le 30 ]
do
sleep 1
ps --no-headers -o '%cpu' -C vlc >> $plikName ;
x=$(( $x + 1 ))
done
```

Program iptraf

iptraf jest bazującym na bibliotece ncurses monitorem pakietów IP w sieci lokalnej generującym różnorodne statystyki sieciowe zawierające: informacje o pakietach TCP, zliczenia UDP, ICMP i OSPF, informacje o wykorzystaniu Ethetnetu, stany węzłów, błędy sum kontrolnych IP i inne.

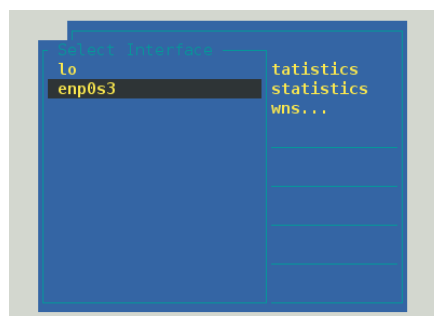
Program iptraf uruchamiamy komendą **sudo iptraf-ng** w oknie terminalu.

Okno programu iptraf-ng:



Następnie wybieramy: **Detailed interface statistics** i potwierdzamy klawiszem Enter.

Wybieramy interfejs: **enp0s3** klawiszem Enter.



Widok finalny potrzebny do zajęć:

```
student@localhost:/home/student
Plik Edycja Widok Wyszukiwanie Terminal Pomoc
iptraf-ng 1.1.4
-----
Total:          Total:          Incoming Incoming  Outgoing  Outgoing
Packets        Bytes        Packets    Bytes    Packets    Bytes
Total:         2916        2717390    1803     2579311    1113     138079
IPv4:          2913        2716964    1800     2578885    1113     138079
IPv6:           3           216        3         216        0         0
TCP:          2280        2654500    1296     2525530    984     128970
UDP:           618         61984      489     52875     129     9109
ICMP:           3           216        3         216        0         0
Other IP:      15          480        15        480        0         0
Non-IP:        0           0          0         0          0         0

Total rates:      0.15 kbps      Broadcast packets: 312
                  1 pps                Broadcast bytes:   26903

Incoming rates:   0.12 kbps
                  1 pps

Outgoing rates:   0.03 kbps
                  0 pps

IP checksum errors: 0

X-exit
```

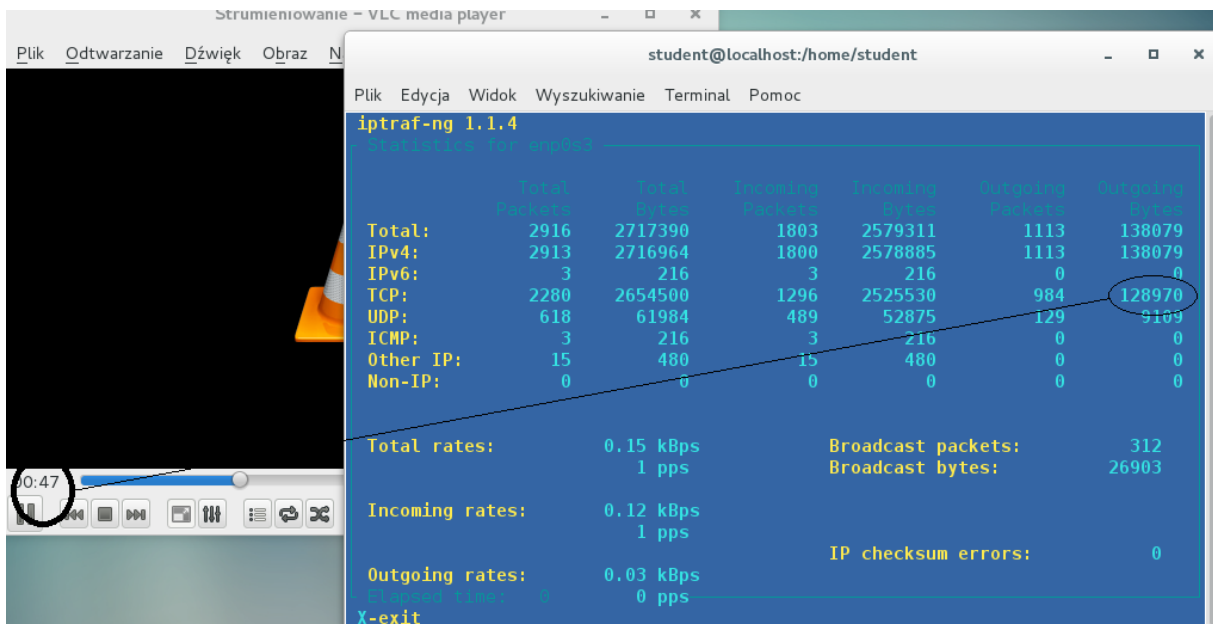
W momencie strumieniowania plików protokołem HTTP zwracamy uwagę na ilość bajtów wyjściowych (outgoing bytes) dla portów TCP.

W trakcie strumieniowania UDP lub RTP zwracamy uwagę na ilość bajtów wyjściowych (outgoing bytes) dla portów UDP.

Program iptraf-ng uruchamiamy tylko na komputerze będącym serwerem, czyli puszczającym strumień w sieć.

Uruchamiamy dokładnie zaraz po uruchomieniu strumieniowania, tak aby zliczane były tylko dane przesłane dla programu VLC !!

Wymagany jest jednorazowy zrzut ekranu zawierający obraz programu VLC i okno programu iptraf-ng:



Podczas strumieniowania w określonej minucie / sekundzie filmu robimy zrzut ekranu, zapisujemy ilość bajtów wysłanych i czas filmu, w celu obliczenia bitrate'u strumienia wyjściowego.

Bitrate = ilość danych wysłanych / czas

Ważne są jednostki: czas podajemy w sekundach, ilość danych: bajty, kilobajty...

Scenariusz 1:

1. Sprzęt

- 2 komputery PC
- System operacyjny Linux CentOS

2. Oprogramowanie

- Aplikacja VLC
- Programy do monitorowania:
 - Obciążenia procesora: skrypt
 - Obciążenie sieci: iptraf-ng

3. Przygotowanie do wykonania ćwiczenia

- Zalogować się na swoje konto systemu Linux
- Sprawdzić czy jest zainstalowany program VLC
- Sprawdzić czy są zainstalowane programy monitorujące (iptraf-ng, ściągnąć skrypt)
- 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)
 - b. RTP (:5004 lub :5005)
 - c. UDP (:1234 lub :1235)
 - 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.264 + MP3 (TS)
 - c. Theora + Vorbis
 - d. MPEG-2 + MPGA (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 puszczający strumień w sieć) oraz na kliencie (komputer odbierający strumień z sieci).

5. Badamy

Na serwerze:

1. Uruchamiamy okno terminalu dla skryptu do pomiaru obciążenia procesora (terminal otwarty w lokalizacji zawierającej plik ze skryptem !!), drugie okno terminalu - program monitorujący obciążenie sieci: iptraf-ng.
2. Uruchamiamy program iptraf-ng według instrukcji do oprogramowania.
3. Notujemy nazwę pliku, protokół wykorzystany do strumieniowania wraz z parametrami, profil (enkapsulacja, kodek video, kodek audio) - przy definicji strumienia: "opcje transkodowania" - >"edytuj wybrany profil" .
4. Rozpoczynamy strumieniowanie testowych plików dla wariantów protokołu i profili strumienia według instrukcji oprogramowania dla VLC.
5. Podłączamy klienta do strumienia.
6. Po rozpoczęciu strumieniowania klikamy skrót Ctrl + J -> z zakładki szczegóły kodeka notujemy:
 - a. oryginalny kodek i parametry pliku



- Po upływie ok. 30 sekund filmu (30 sekunda filmu w programie VLC na serwerze) uruchamiamy skrypt, poprzez wpisanie w oknie terminalu: **./skrypt nazwa_pliku_do_zapisu**. Nazwa_pliku_do_zapisu – indywidualna dla każdego pomiaru, wyniki w niej zapisane posłużą do wygenerowania wykresu zużycia procesora w jednostce czas (opis w instrukcji oprogramowania dla skryptu). Za każdym pomiarem zmieniamy nazwę pliku i jednoznacznie go opisujemy, którego pomiaru dotyczy.
- Pomiar trwa 30 sekund. W trakcie pomiaru robimy 1 zrzut ekranu na którym musi znajdować się okno VLC oraz okno terminalu z programem iptraf-ng (opis w instrukcji oprogramowania dla iptraf-ng).
- Plik wygenerowany przez skrypt oraz zrzut ekranu kopiujemy w znane miejsce, opisujemy je w sposób jednoznaczny aby mieć pewność którego pomiaru dotyczy !!

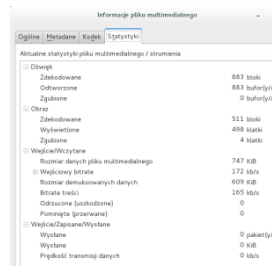
Po skończonym pomiarze zamykamy okno programu VLC na serwerze i kliencie, powtarzamy całą procedurę dla innych parametrów / filmów.

Na kliencie:

- Uruchamiamy okno terminalu dla skryptu do pomiaru obciążenia procesora (terminal otwarty w lokalizacji zawierającej plik ze skryptem !!)
- Rozpoczynamy odbiór strumieniowanych testowych plików według instrukcji oprogramowania dla VLC.
- Po rozpoczęciu strumieniowania klikamy w oknie VLC skrót Ctrl + J -> z zakładki szczegóły kodeka notujemy:
 - kodek i parametry pliku



- W tym samym momencie co na serwerze uruchamiamy skrypt, poprzez wpisanie w oknie terminalu: **./skrypt nazwa_pliku_do_zapisu**. Nazwa_pliku_do_zapisu – indywidualna dla każdego pomiaru, wyniki w niej zapisane posłużą do wygenerowania wykresu zużycia procesora w jednostce czas (opis w instrukcji oprogramowania dla skryptu). Za każdym pomiarem zmieniamy nazwę pliku i jednoznacznie go opisujemy, którego pomiaru dotyczy.
- Pomiar trwa 30 sekund (tak samo jak na serwerze), podczas trwania pomiaru klikamy na oknie VLC skrót Ctrl + J -> wybieramy zakładkę Statystyki, w momencie gdy na serwerze robimy zrzut ekranu zawierający okno programu VLC i iptraf-ng, na kliencie robimy zrzut ekranu z zakładki Statystyki.



Ze zrzutu zapisujemy wejściowy bitrate i bitrate treści.

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 !!

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) przepływność 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
- wykadrować, pomniejszyć i wydrukować odpowiednie zrzuty ekranu (każdy protokół, kodek, materiał video)
- opisać spostrzeżenia, wyciągnąć wnioski