

Usługi i aplikacje internetu

Laboratorium nr 7 - DASH

Adrian Zalewski, Wiktor Zawadzki, Juliusz Kuzyka

Czerwiec 2023

Spis treści

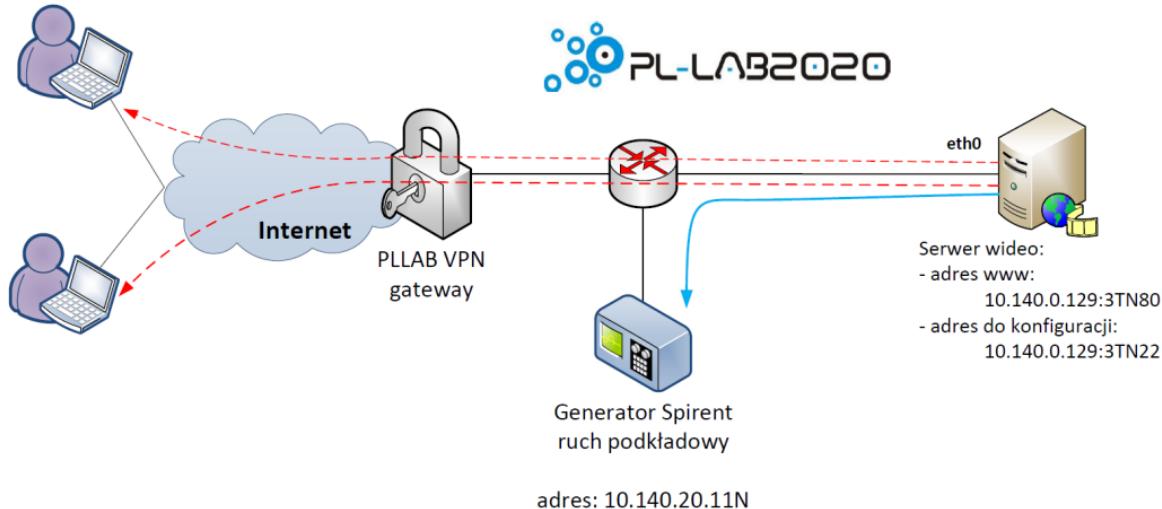
1 Cel laboratorium i infrastruktura sieci	2
2 Przygotowanie filmu wideo zgodnego ze standardem MPEG DASH i opublikowanie na serwerze wideo	2
3 Badanie efektywności algorytmów adaptacji	3
4 Badanie wydajności systemu	6
4.1 Wykresy dla 10 żądań	6
4.2 Wykresy dla 15 żądań	9
5 Wnioski	11

1 Cel laboratorium i infrastruktura sieci

Zapoznanie się z adaptacyjnym strumieniowaniem wideo zgodnego standardem MPEG DASH oraz zasad publikowania zasobów zgodnie z DASH. W ramach ćwiczenia:

- Przygotujemy serwer strumieniujący wideo
- Przygotujemy film zgodnie z standardem MPEG DASH i opublikujemy go na serwerze
- Przebadamy efektywność działania algorytmów adaptacji

Poniżej znajduje się infrastruktura sieci laboratoryjnej:



Rysunek 2: Schemat sieci laboratoryjnej.

Przy czym nasz adres serwera WWW to *10.140.0.129:32380*. A adres do konfiguracji to *10.140.0.129:32322*

2 Przygotowanie filmu wideo zgodnego ze standardem MPEG DASH i opublikowanie na serwerze wideo

Po połączeniu się z siecią laboratoryjną przez VPNa oraz, wrzucamy nasz film. W tym celu wykorzystujemy polecenie:

```
scp -P 32322 ./KochamUAI.mp4 root@10.140.0.129:/root/my_data
```

Następnie logujemy się na serwer (my w tym celu wykorzystaliśmy PowerShella) przez ssh - wykorzystujemy polecenie:

```
ssh root@10.140.0.129 -p 32322.
```

Przechodzimy następnie do zmiany długości wrzuconego materiału w tym celu posługujemy się poleceniem (z poziomu /root/my_data):

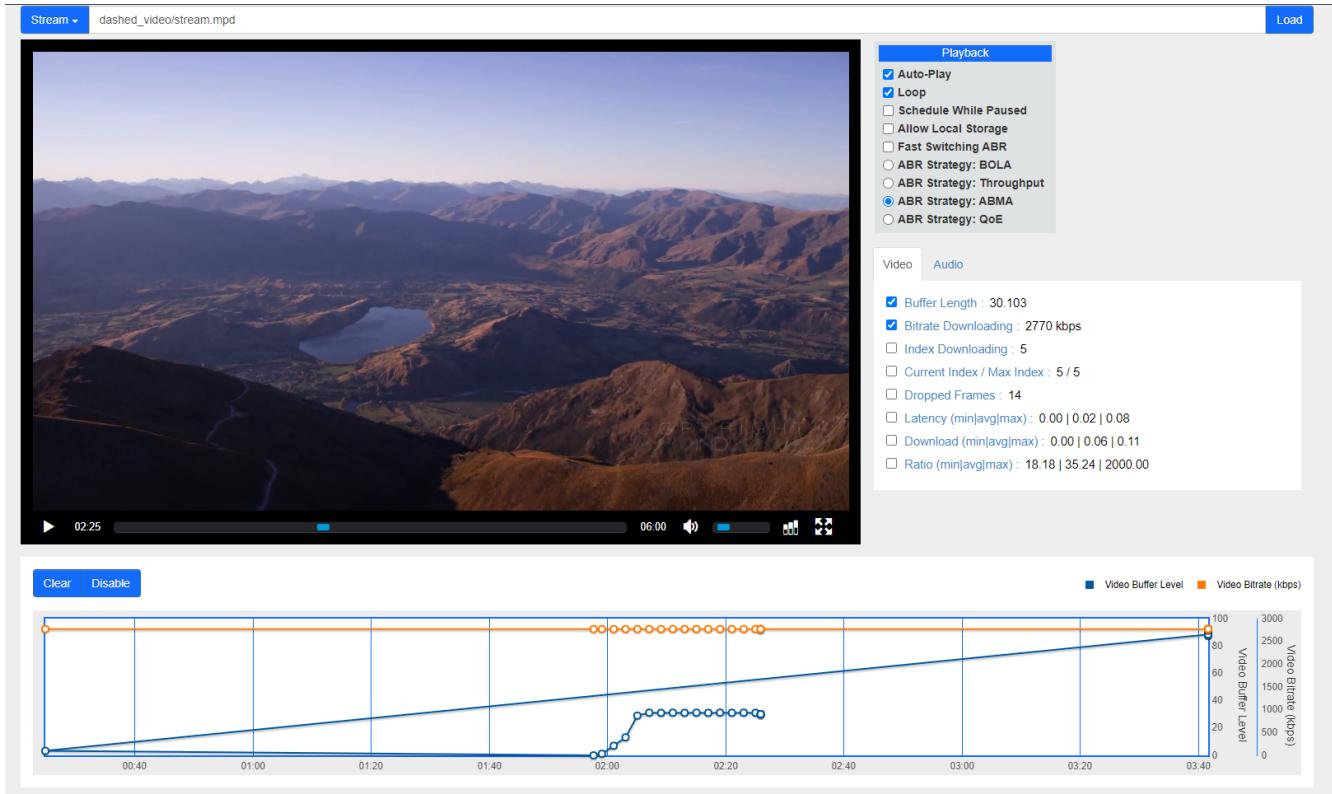
```
ffmpeg -i KochamUAI.mp4 -ss 00:01:30 -t 00:06:00 -acodec copy -vcodec copy Shorten.mp4
```

Po skróceniu filmu rozpoczynamy proces rekodowania, korzystamy z zapewnionego skryptu *recode.sh*, wywołujemy polecenie (z poziomu /root):

```
./recode.sh ./my_data/Shorten.mp4
```

Wrzucamy teraz plik na serwer www, wykorzystujemy do tego polecenie (z poziomu /root directory):

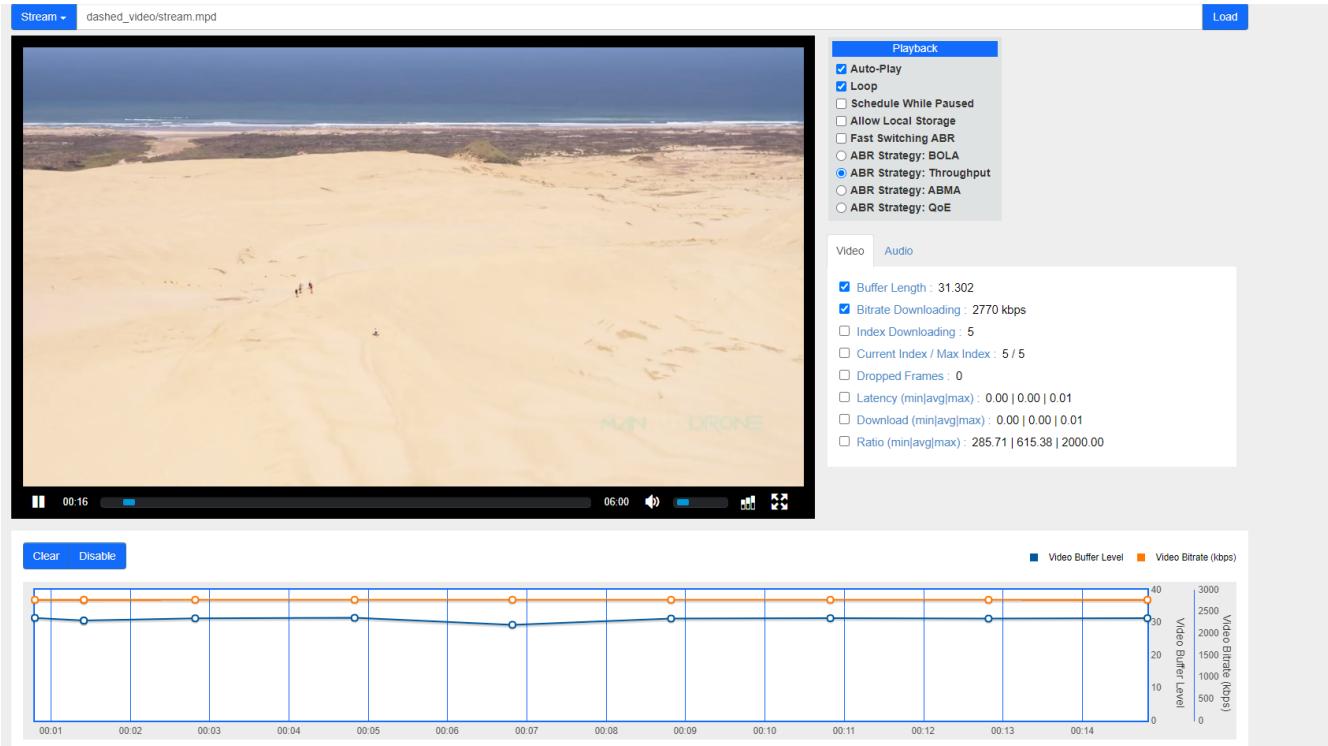
```
cp -r ./dashed_video /var/www/html/
```



Rysunek 1: Działające wideo.

3 Badanie efektywności algorytmów adaptacji

Przetestowaliśmy przepustowość swojego połączenia i możemy stwierdzić, że pozawala ono na renderowanie filmów w najwyższej rozdzielcości, czyli możemy korzystać z bitrate'u 2770 kbps.



Rysunek 2: Stabilny wykres bufora dla bitrate'u 2770 kbps

Następnie wprowadzamy ograniczenie przepustowości na interfejsie sieciowym serwera *eth0*.

```
root@wvideo-server23:~# ./impairment.sh add eth0 2.324
impairment on eth0 - set!
root@wvideo-server23:~# ./impairment.sh show eth0

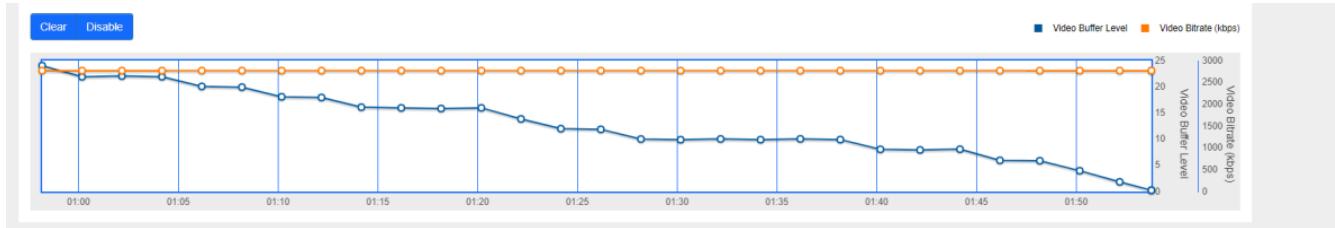
Queue configuration
class htb 1:10 parent 1:1 leaf 10: prio rate 2324Kbit ceil 2324Kbit burst 1599b cburst 1599b
class htb 1:1 root rate 2324Kbit ceil 2324Kbit burst 1599b cburst 1599b
qdisc htb 1: root refcnt 33 r2q 10 default 16 direct_packets_stat 0 direct_qlen 1000
qdisc netem 10: parent 1:10 limit 1000

Configuration of packet filters

Traffic statistics
qdisc htb 1: root refcnt 33 r2q 10 default 16 direct_packets_stat 0 direct_qlen 1000
  Sent 132927438 bytes 96658 pkt (dropped 0, overlimits 46637 requeues 0)
  backlog 0b 0p requeues 0
qdisc netem 10: parent 1:10 limit 1000
  Sent 132927438 bytes 96658 pkt (dropped 0, overlimits 0 requeues 0)
  backlog 0b 0p requeues 0
root@wvideo-server23:~#
```

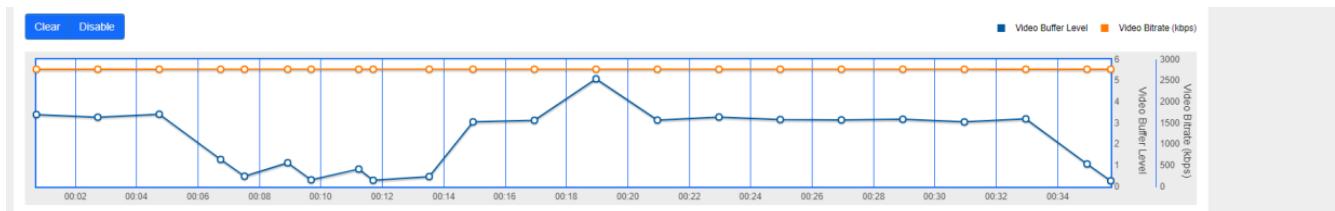
Rysunek 3: Wywołane polecenia

BOLA (Buffer-Based Optimized Dynamic Adaptive Streaming over HTTP) - ten algorytm dostosowuje jakość wideo w oparciu o wypełnienie bufora odtwarzania. Monitoruje poziom bufora i dynamikę zmian, aby określić optymalną jakość wideo do odtworzenia. Gdy bufor jest zbyt niski, algorytm może obniżyć jakość, aby uniknąć zacięć. Jeśli bufor jest wysoki, algorytm może podnieść jakość w celu poprawy doświadczenia użytkownika. W naszym przypadku, algorytm BOLA jest bardzo dobrym algorytmem do odtwarzania video, ponieważ Video Buffer przez dłuższy czas nie spadał do 0, co pozwoliło na oglądanie video bez zacięć. Dopiero po około 1:55 minuty Video Buffer spadł do 0.



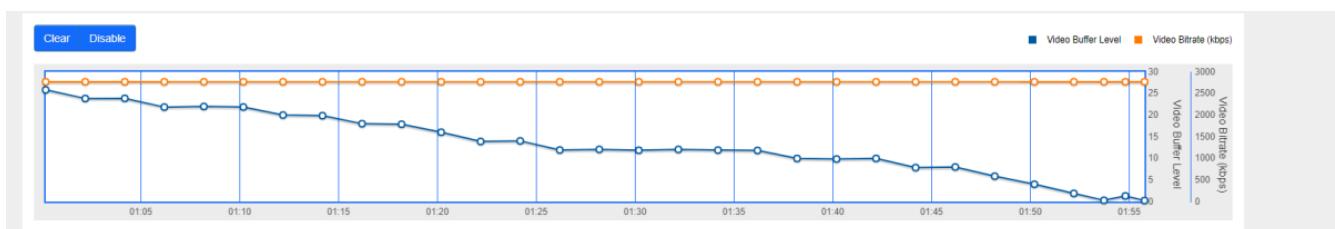
Rysunek 4: Wykres poziomu Video Buffer dla BOLA

Throughput - ten algorytm dostosowuje jakość wideo w oparciu o aktualne oszacowanie przepustowości sieciowej. Monitoruje szybkość pobierania strumienia video i na podstawie tych danych podejmuje decyzję o odpowiedniej jakości do odtworzenia. Jeśli przepustowość sieci jest wysoka, algorytm wybierze wyższą jakość wideo. Jeśli przepustowość jest niska, algorytm wybierze niższą jakość, aby uniknąć buforowania. W naszej sytuacji, algorytm Throughput jest słabym algorytmem do odtwarzania video, ponieważ dopiero po 8 sekundach Video Buffer spadł do 0 i utrzymywał się na tym poziomie przez kolejnych 6 sekund. Następnie wrócił on do efektywnego odtwarzania video, jednak po krótkim czasie znowu Video Buffer spadł do 0.



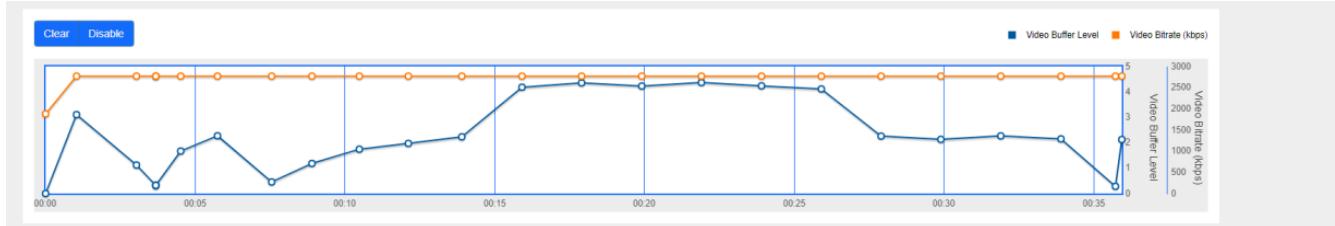
Rysunek 5: Wykres poziomu Video Buffer dla Throughput

ABMA (Adaptive Bitrate Multicast Algorithm) - ten algorytm jest stosowany w przypadku transmisji multicastowych, gdzie dane są wysyłane jednocześnie do wielu odbiorców. ABMA dostosowuje jakość wideo, uwzględniając zarówno warunki sieciowe, jak i preferencje użytkowników. Algorytm może monitorować opóźnienie, straty pakietów, dostępne pasmo i inne parametry, aby wybrać optymalną jakość wideo dla każdego odbiorcy. W naszym przypadku, algorytm ABMA, podobnie jak algorytm BOLA jest bardzo dobrym algorytmem do odtwarzania video, ponieważ dopiero po około 1:55 minuty Video Buffer spadł do 0.



Rysunek 6: Wykres poziomu Video Buffer dla ABMA

QoE (Quality of Experience) - odnosi się do metody oceny jakości doświadczenia użytkownika podczas odtwarzania treści multimedialnych, takich jak wideo. W przeciwieństwie do innych algorytmów adaptacji wideo, które skupiają się głównie na dostosowaniu jakości wideo do warunków sieciowych, algorytm QoE koncentruje się na subiektywnym odczuciu użytkownika. W naszym scenariuszu QoE okazał się kiepskim algorytmem, ponieważ podobnie jak Throughput już po paru sekundach Video Buffer spadł do 0, a nawet jak wrócił do prawidłowego bufferowania video to po kilkunastu sekundach filmik ponownie się zacinał.



Rysunek 7: Wykres poziomu Video Buffer dla QoE

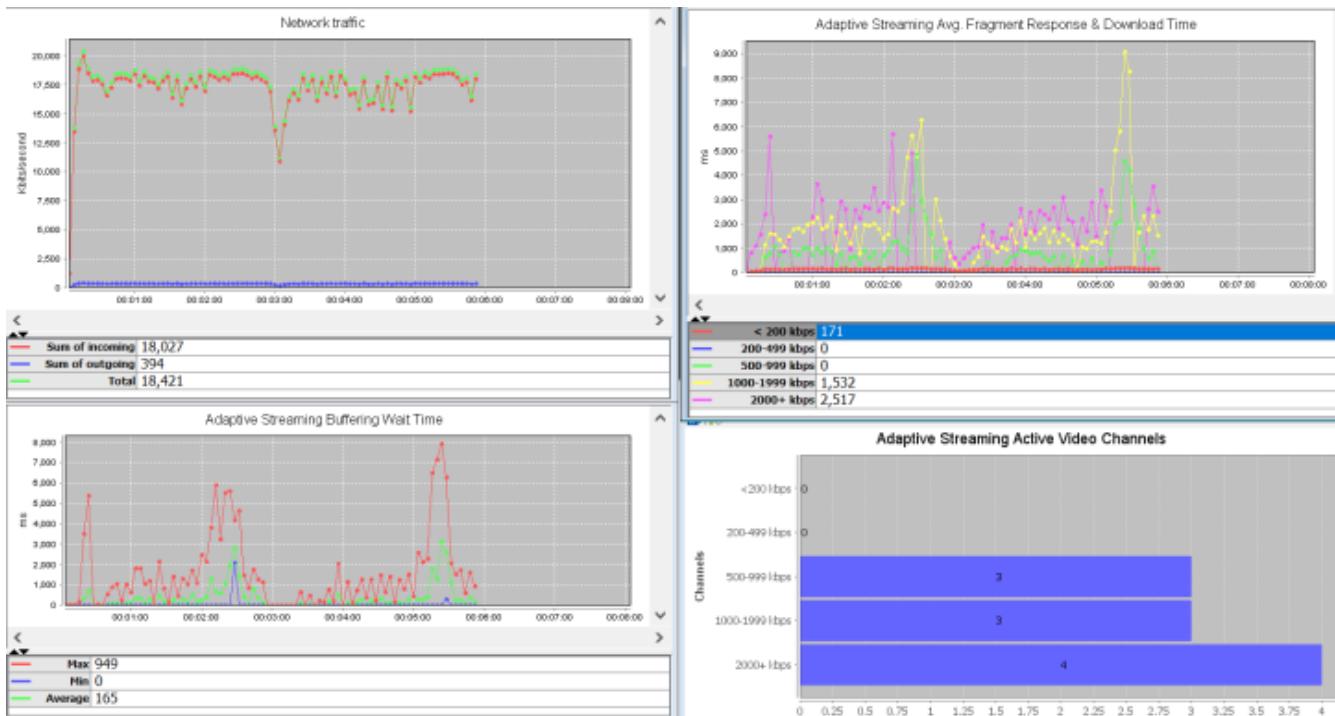
4 Badanie wydajności systemu

Ustawiamy przepustowość na interfejsie eth0 polecienniem: `./impairment.sh add eth0 20`.

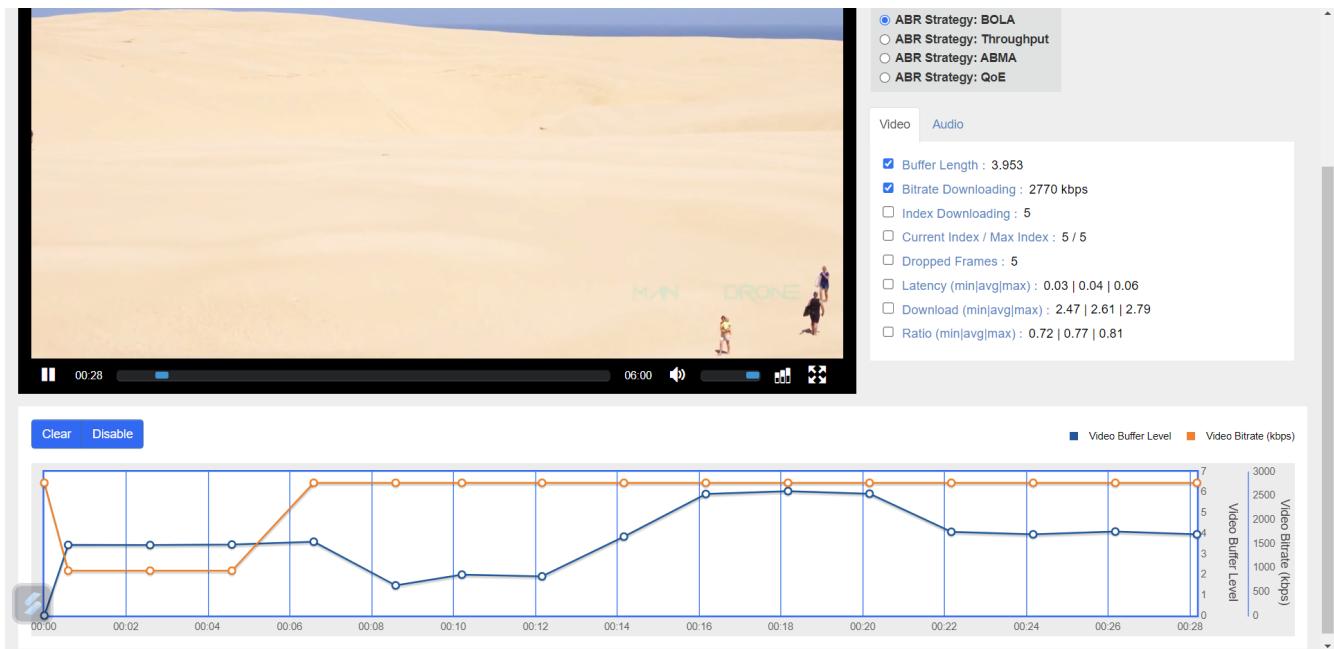
Następnie przy pomocy Spirenta generujemy ruch podkładowy, początkowo ustawiamy liczbę dodatkowych żądań na 10 i uruchamiamy nasz film oraz sprawdzamy zachowanie poszczególnych algorytmów.

4.1 Wykresy dla 10 żądań

Poniżej znajdują się wykresy z Spirenta, która zawierają liczbę aktualnych żądań oraz z jaką przepustowością strumieniowane jest wideo (dla danego żądania).

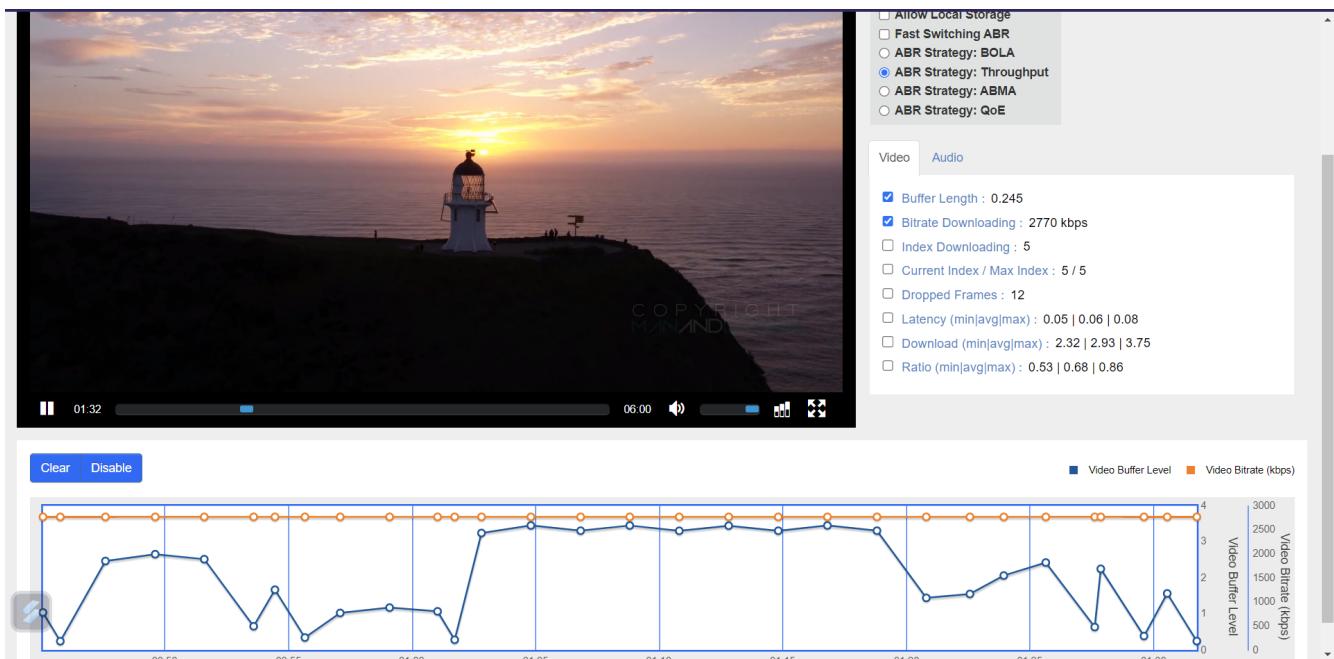


Rysunek 8: Wykresy Spirent



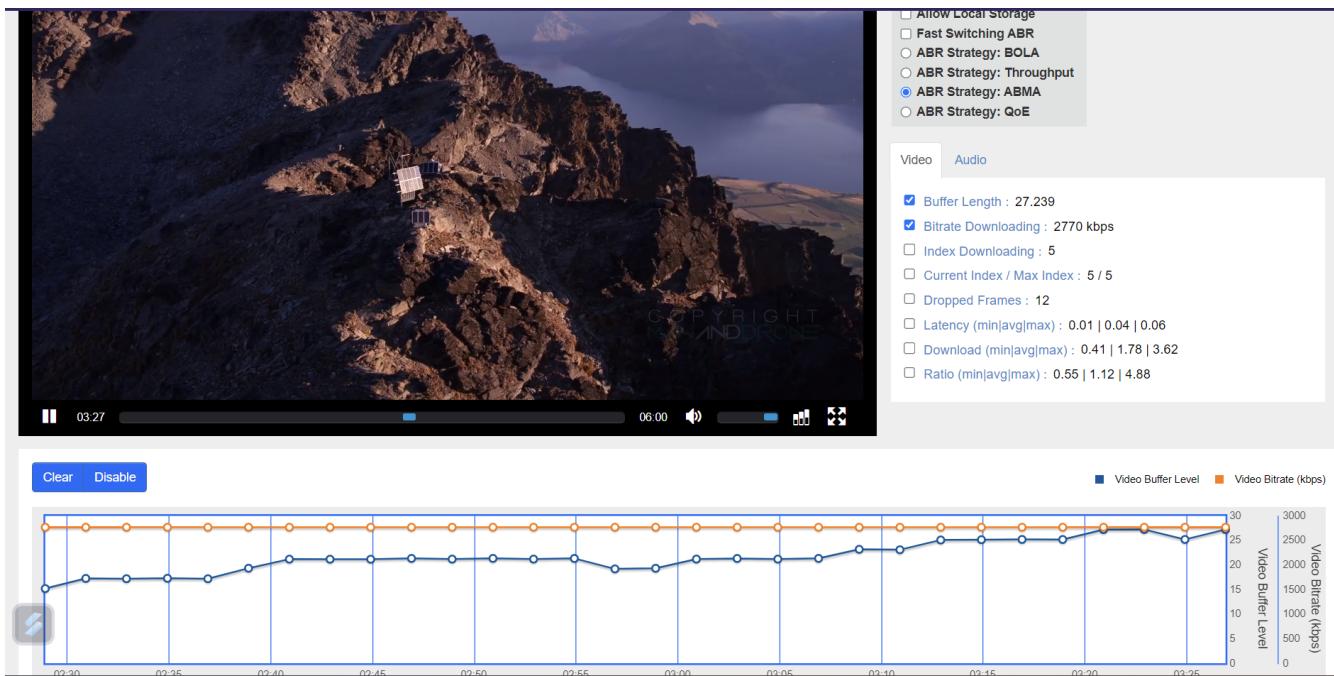
Rysunek 9: Wykres poziomu Video Buffer dla BOLA dla 10 klientów

Jak widać powyżej, algorytm BOLA był dość stabilnym algorytmem nawet przy dużej ilości klientów na stronie.



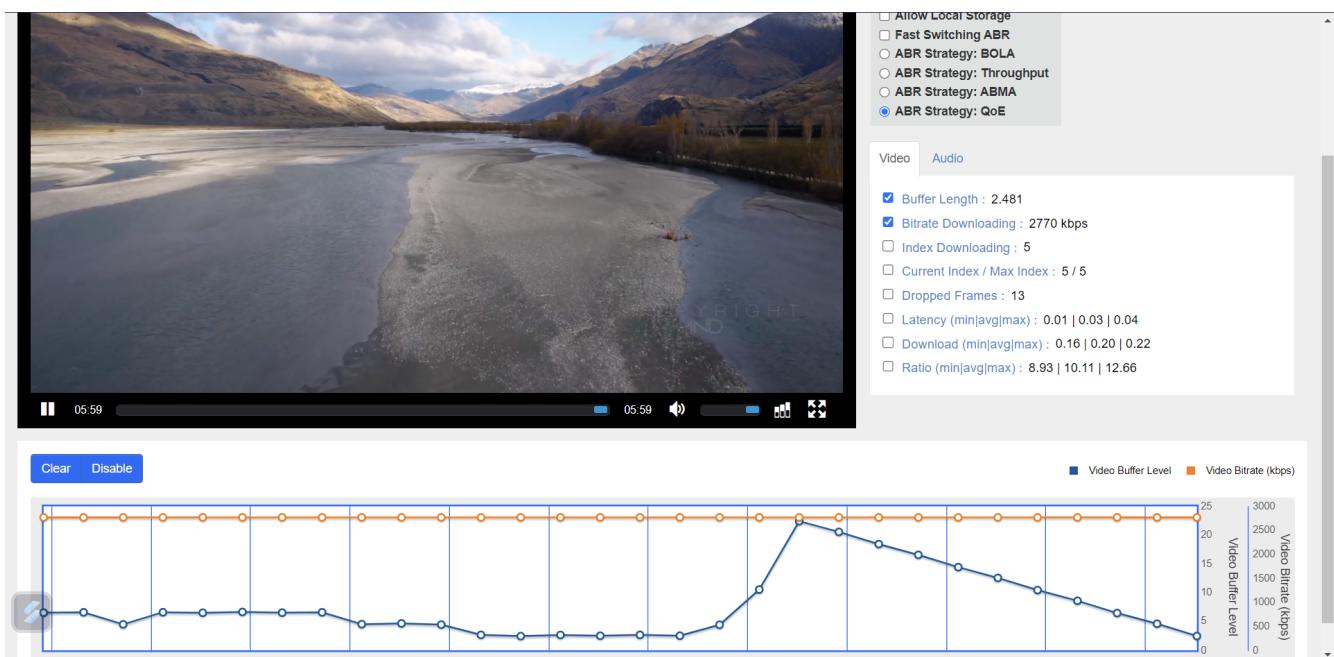
Rysunek 10: Wykres poziomu Video Buffer dla Throughput dla 10 klientów

Algorytm Throughput jest niestabilnym algorytmem przy większej ilości osób oglądających, ponieważ często występują spadki poziomu Buffora.



Rysunek 11: Wykres poziomu Video Buffer dla ABMA dla 10 klientów

ABMA okazał się najskuteczniejszym algorytmem przy liczbie 10 widzów, spadki poziomu Buffora nie występują podczas oglądania filmu.

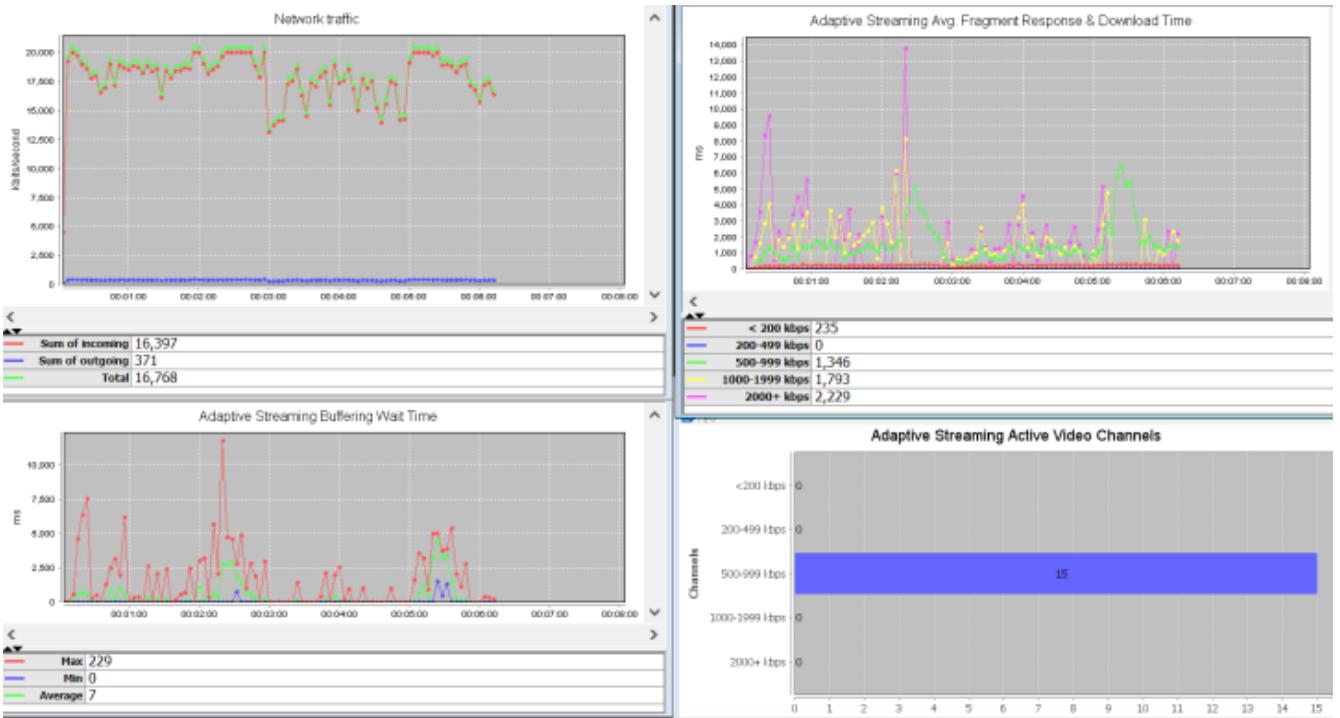


Rysunek 12: Wykres poziomu Video Buffer dla QoE dla 10 klientów

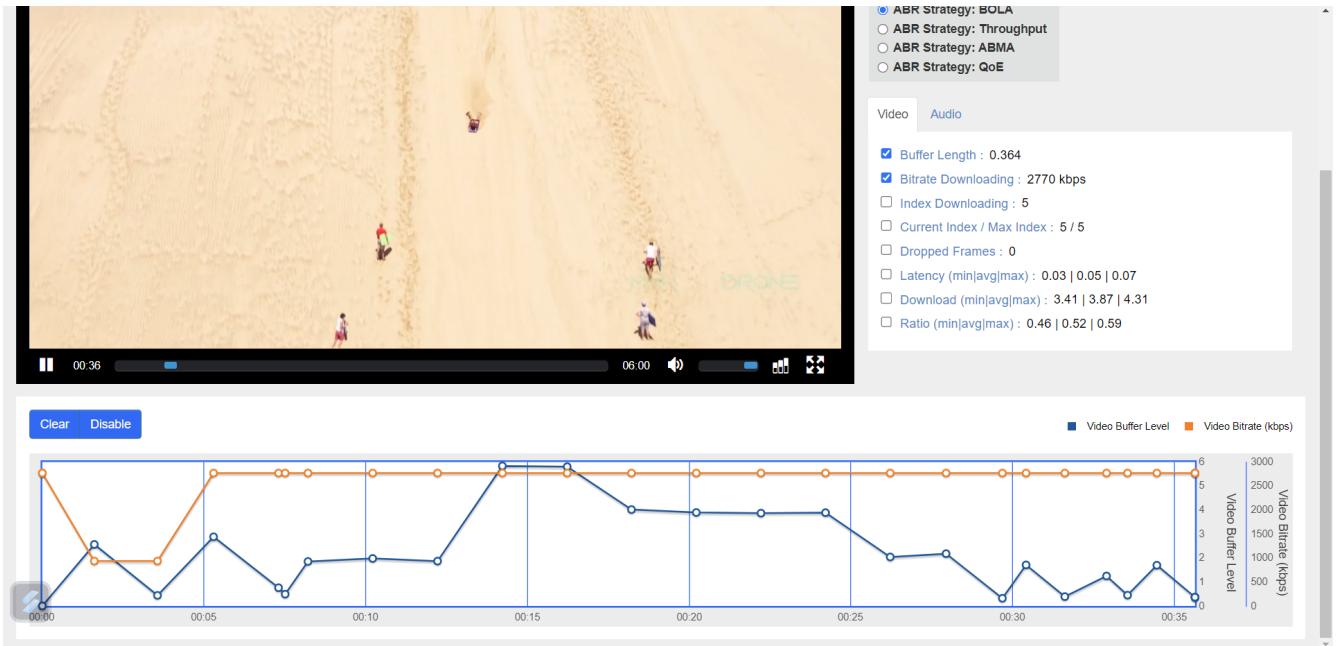
QoE w przypadku 10 widzów jest słabym algorytmem, ponieważ poziom Buffora filmu utrzymywał się długim czasem przy zerze.

4.2 Wykresy dla 15 żądań

Poniżej znajdują się wykresy z Spirenta, która zawierają liczbę aktualnych żądań oraz z jaką przepustowością strumieniowane jest wideo (dla danego żądania). Możemy zauważyć że po wzroście, praktycznie każdy klient korzysta z takiego samego bitrate'u.



Rysunek 13: Wykresy z Spirenta dla 15 żądań



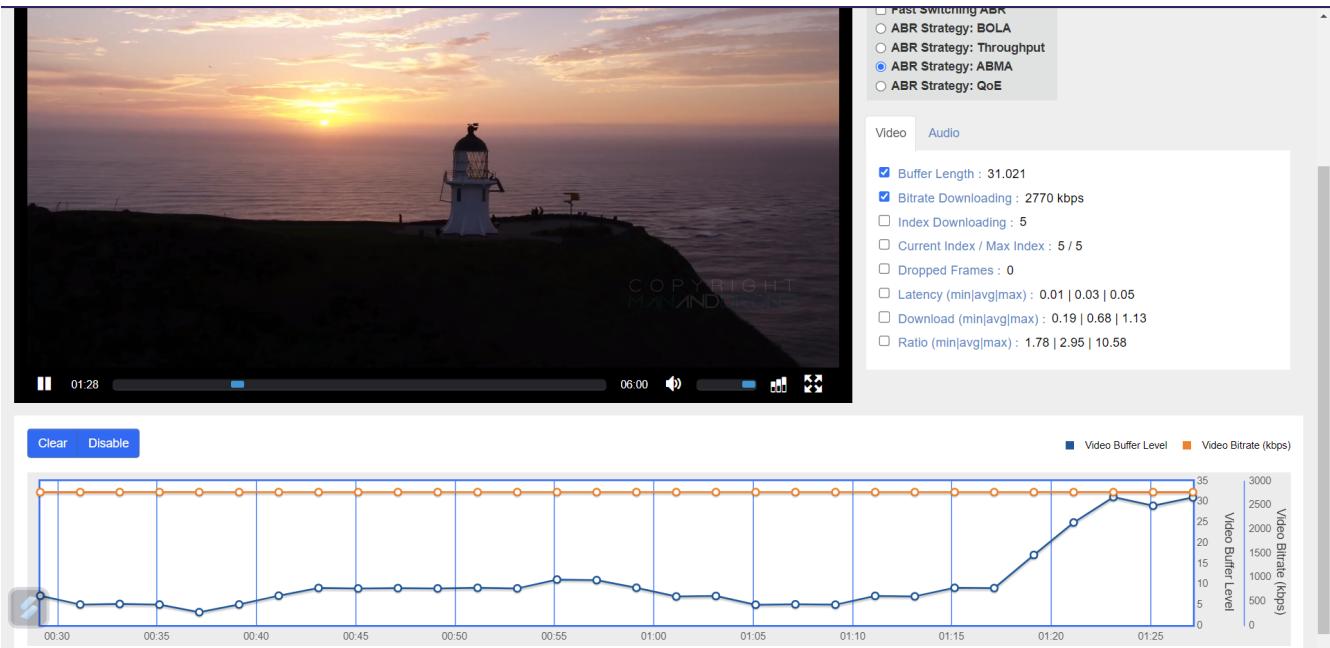
Rysunek 14: Wykres poziomu Video Buffer dla BOLA dla 15 klientów

Algorytm BOLA był średnim algorytmem, spadki poziomu Buffora występuły dość często i przez jakiś czas trzymały się blisko najniższej wartości.



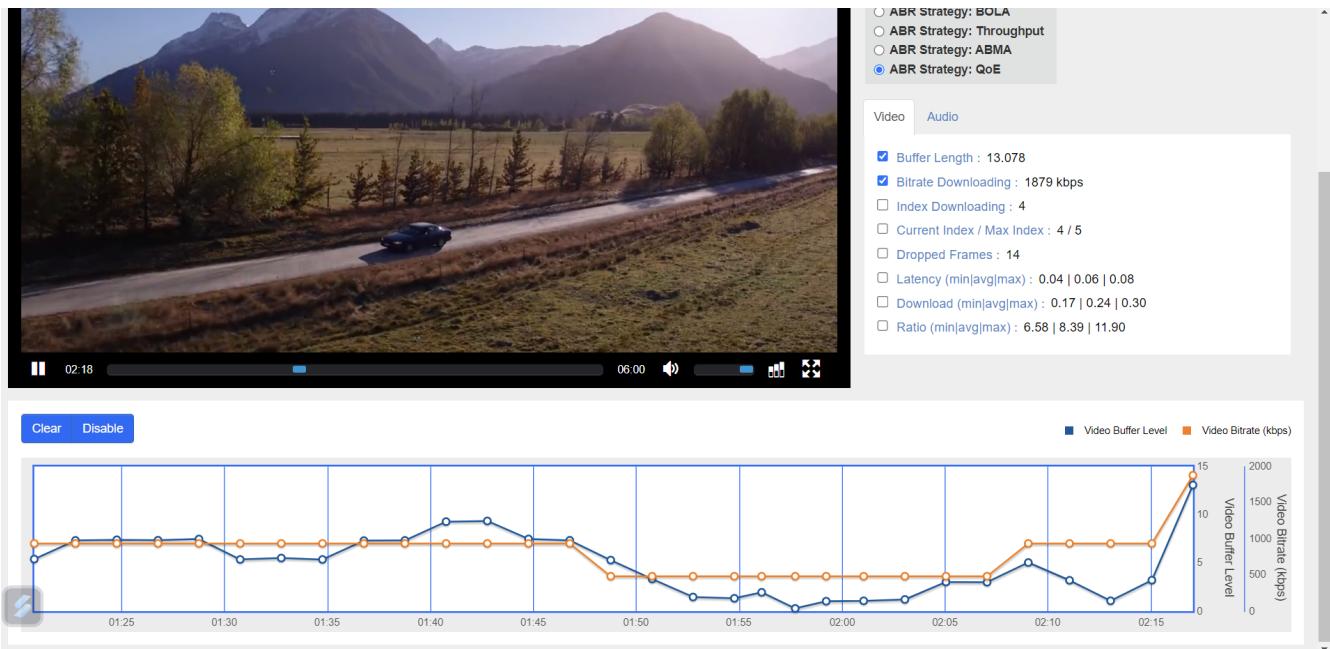
Rysunek 15: Wykres poziomu Video Buffer dla Throughput dla 15 klientów

Algorytm Throughput przez ponad 30 sekund dobrze sobie radził z utrzymaniem 15 klientów na wysokim poziomie Buffora, jednak po tym czasie jego poziom zaczął stopniowo maleć, do momentu aż osiągnął wartość 0.



Rysunek 16: Wykres poziomu Video Buffer dla ABMA dla 15 klientów

Algorytm ABMA przez dłuższy czas utrzymywał się na dość niskim poziomie Buffora filmu, jednak po 30 sekundach osiągnął on najwyższy poziom i tam się przez jakiś czas utrzymywał.



Rysunek 17: Wykres poziomu Video Buffer dla QoE dla 15 klientów

Ogólnie rzecz biorąc, QoE okazał się bardzo chwiejnym algorytmem, ponieważ w czasie jego trwania często następowały spadki do najniższego poziomu i tam się dugo utrzymywały.

5 Wnioski

Podczas laboratorium mieliśmy okazję zapoznać się z czterema różnymi rodzajami algorytmów adaptacji. Testowaliśmy te algorytmy w różnych scenariuszach, szczególnie związanych z liczbą żądań na serwerze.

Przeprowadzenie testów pozwoliło nam zobaczyć, jak różne algorytmy radzą sobie z różnymi scenariuszami. Niektóre algorytmy, takie jak BOLA, mogą być bardziej efektywne w przypadku niewielkiej liczby żądań, podczas gdy inne, takie jak ABMA, mogą lepiej działać w przypadku dużego obciążenia serwera. Ważne jest, aby dostosować algorytm do konkretnych warunków i potrzeb systemu.

Podsumowując, testowanie różnych algorytmów adaptacji w laboratorium pozwoliło nam na lepsze zrozumienie ich działania i skuteczności w zależności od konkretnych warunków. Wybór odpowiedniego algorytmu adaptacji jest kluczowy dla zapewnienia optymalnego działania systemu w zmiennych warunkach operacyjnych. Jesteśmy przekonani, że wiedza oraz doświadczenia z tego laboratorium przydadzą się w przyszłości podczas wykonywania projektów, a nawet pracy.