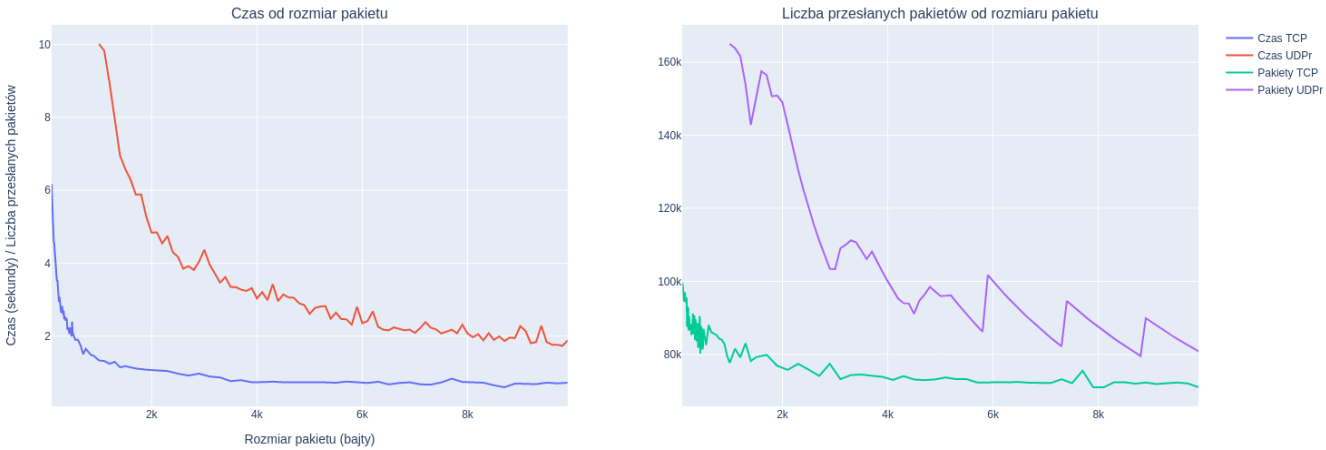
Piotr Pływacz (448454) - raport (1. zad. zal. SIK)

Testowanie TCP i UDPr

Na początku porównamy ze sobą wydajność dwóch protokołów z retransmisją - TCP oraz UDPr (UDP z retransmisją). Wykonane i opisane zostaną następujące testy (w prawie wszystkich poniższych testach mamy: $MAX\_WAIT = 1$, $MAX\_RETRANSMITS = 1000$ - chcemy zagwarantować niezawodność UDPr):

1. Wykresy zależności czasu i liczby przesłanych pakietów od rozmiaru pojedynczego pakietu:

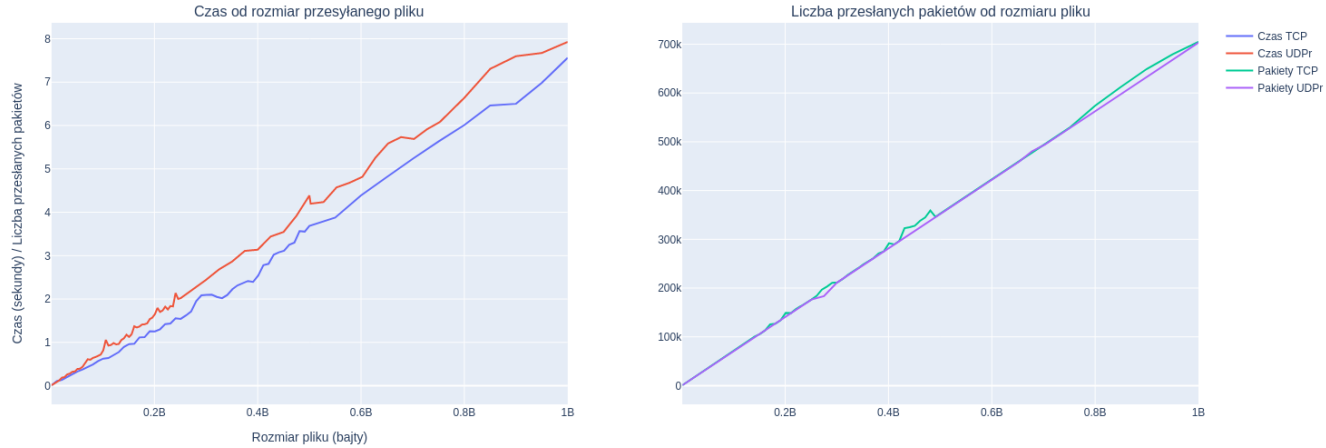


Rozważamy tutaj kolejne wykonania programów klienta i serwera, gdzie (domyślna) liczba bajtów w każdym pakiecie rośnie (oczywiście ostatni pakiet może mieć mniej bajtów). Wszędzie przesyłamy plik wielkości 100MB. Czasy wykonania obydwu protokołow stabilizują się wraz wielkością pakietów. Jest to spowodowane wartością MTU ustawioną na 1500B w przypadku obydwu maszyn witrualnych. Zatem realnie pakiety większe niż 1500-bajtowe przez tę sieć nie przejdą. Istotnie, czasy działania, gdy rozmiar pakietu wynosi 64000B są niemal identyczne jak te, gdy wynosi on 10000B - jak na wykresie.

Dodatkowo dane dla pakietów wielkości od 100 do 1000B dla UDPr zostały pominięte, ponieważ czasy działania w tych przypadkach wyniosił około 80 sekund i zaburzały wykres.

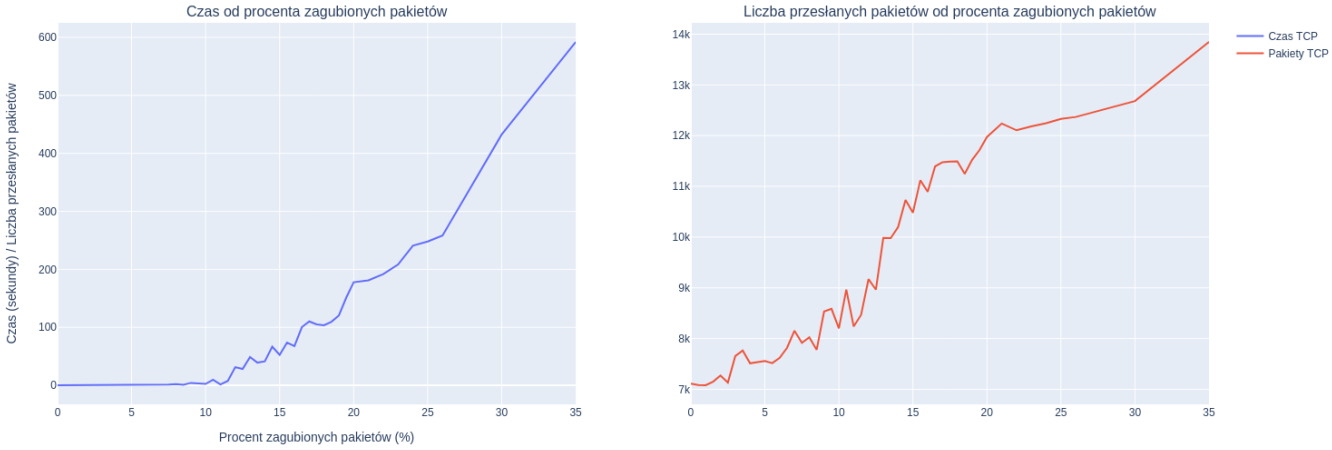
W dalszych testach będziemy wysyłali pakiety jedynie wielkości 64000B, ponieważ w przypadku mniejszych MTU nie tracimy na czasie, w w przypadku większych - zyskamy.

2. Wykresy zależności czasu i liczby przesłanych pakietów od rozmiaru przesyłanego pliku:



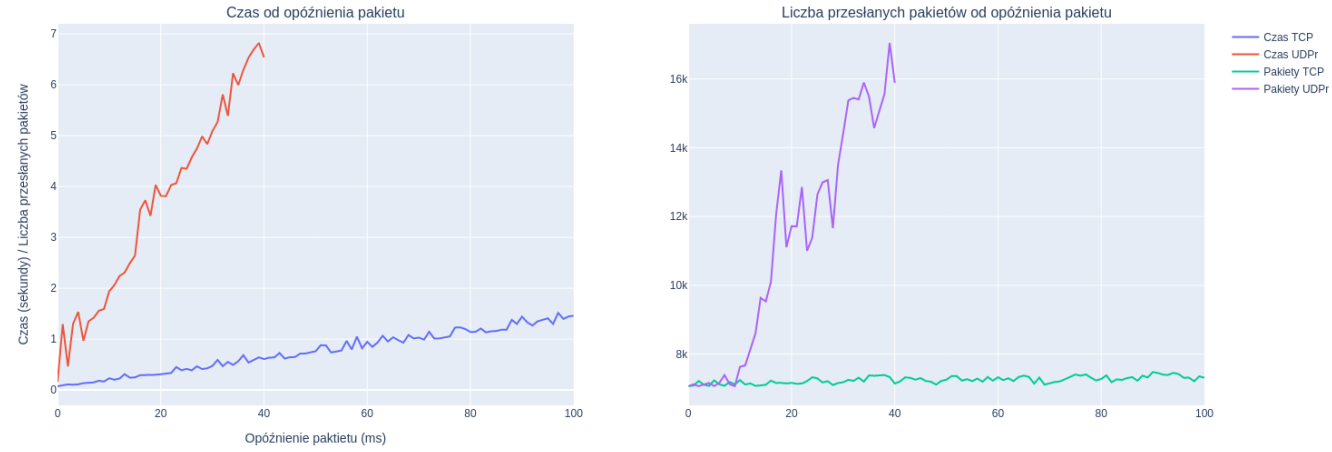
Wysyłamy tutj pliki o rozmiarach od 1MB do 1GB. Zależności te, czego można było sie spodziewaać, są liniowe i obserwujemy tą samą, $0.5 - 1$-sekundową, różnicę w czasie działania UDPr, a TCP.

3. Wykres zależności czasu i liczby przesłanych pakietów od procenta zagubionych pakietów:



W tym przypadku wysyłamy plik o rozmiarze 10MB lecz wykresy doyczą jedynie protokołu TCP. To dlatego, że (korzystając z powyższych wyliczeń) przy takim rozmierze pliku, protokół UDPr wysyła około 7000 pakietów, więc nawet przy gubieniu prakietów z prawdopodobieństwem 10%, dawałoby minimum 700 sekund, więc wykres taki w tym przypadku nie miałby sensu.

4. Wykres zależności czasu i liczby przesłanych pakietów od opóźnienia pakietów:



Tym razem przesyłany był plik wielkości 10MB. Wydajność protokołu UPDr na testach tego typu oczywiście głównie zależy od wartości MAX_WAIT. W przypadku jak ten, gdy czas działania programu bez opóźnień wynosi poniżej sekundy, każde zagubienie pakietu kosztowałoby nas minimum właśnie tyle. Dlatego czas oczekiwania na pakiet w tym teście został zmniejszony do 1ms. Wówczas oczywiście klient wysyła bardzo dużo pakietów, jednak czas działania jest wciąż mniejszy. Aby lepiej pokazać jak zachowuje się TCP, dane UDPr zostały zakrojone. Czas działania UDPr z opóźnieniem 100ms wyniół 16.646 sekund.

Testowanie UDP

Testowanie protokołu UDP na testach autmatycznych nie ma dużego sensu nawet bez emulowania gubienia pakietów, ponieważ aby być w stanie wygenerować jakikolwiek wykres zależności czasu od innych parametrów, programy muszą działać pewną ilość czasu na tyle dużą, aby szum nie zaburzał mocno danych. Gdy jednak programy używające tego protokołu wysyłają dane przez na przykład sekundę, co zaledwie parę prób gubi się jakiś pakiet, a wówczas nie dojdzie do pełnego wysłania danych.

Użyte narzędzia

Do emulowania sieci został użyty moduł jądra *netem*,
Porgramy były uruchamiane automatycznie prostymi skryptami napisanymi w Pythonie. Czas był mierzony funkcją *time*() Pyhonowej biblioteki *time*. Liczba pakietów natomiast została zmierzona przez odjęcie sumy wartości w plikach

$$/sys/class/net/eth1/statistics/tx\_packets \text{ i } /sys/class/net/eth1/statistics/rx\_packets$$

przed uruchomieniem programów od sumy wartości w tych samych plików po zakończeniu działania programów.
Wszystkie programy były uruchamiane na połączonych ze sobą maszynach wirtualnych przez *vagrant* (maszyny z zajęć).