

Programowanie Sieciowe

Zadanie 1.1

Radosława Żukowska - Lider Zespołu

Aleksandra Szczypawka

Małgorzata Grzanka

20.11.2025r

Wersja sprawozdania: 1

Zadanie 1 - Komunikacja UDP

Napisz zestaw dwóch programów – klienta i serwera wysyłające datagramy UDP. Proszę napisać jedno zadanie w konfiguracji klient/server Python/C, a drugie w konfiguracji klient/server C/Python – do wyboru.

1.1

Klient wysyła, a serwer odbiera datagramy oraz odsyła ustaloną odpowiedź. Klient powinien wysyłać kolejne datagramy o przyrostającej wielkości, tj. 2, 4, 8, 16, 32, itd. bajtów. Ustalić eksperymentalnie z dokładnością do jednego bajta jak duży datagram jest obsługiwany. Wyjaśnić. Zmierzyć czas pomiędzy wysłaniem wiadomości a odebraniem odpowiedzi po stronie klienta i zestawić wyniki na wykresie.

Rozwiążanie

Link do Repozytorium

Do realizacji zadania powstał:

- serwer UDP w języku C,
- klient UDP w Pythonie.

W skrypcie klienta można ustawić wartości: początkowy rozmiar datagramu, sposób inkrementacji rozmiaru oraz liczbę powtórzeń pomiaru dla dokładniejszego uśrednienia czasów.

Konfiguracja testowa

- Każdy kontener w sieci z36_network dostaje dynamicznie przydzielony prywatny adres IP z podsieci Docker. Kontenery mogą komunikować się między sobą zarówno po tym adresie, jak i po aliasie nadanym przez –network-alias
- Serwer nasłuchuje na porcie 54070. Klient łączy się na ten port.

- Klient używa bufora 4096B do odbioru odpowiedzi, ale wysyła pakiety większe – eksperyment ma na celu ustalenie maksymalnego rozmiaru datagramu obsługiwanej przez UDP.
- Komunikacja ograniczona do sieci Dockera, więc opóźnienia sieci fizycznej praktycznie nie występują.

Wyniki testów

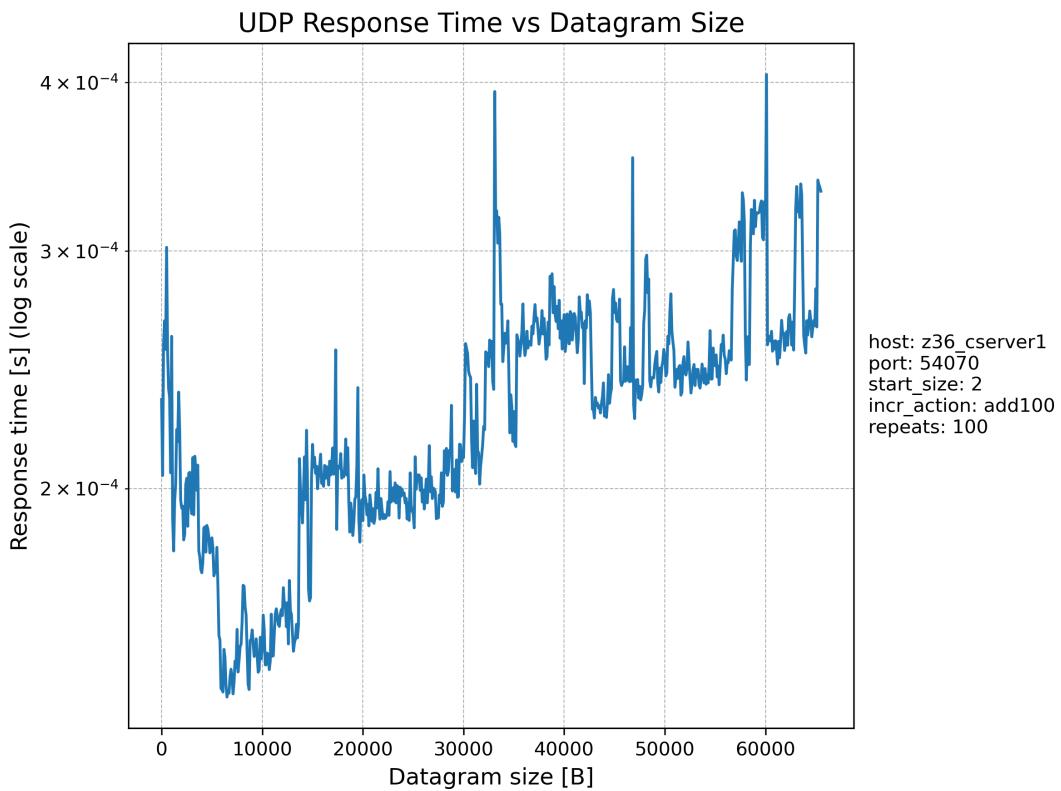
Testy zostały przeprowadzone poprzez uruchamianie aplikacji klienckiej z różnymi parametrami – początkowym rozmiarem datagramu oraz sposobem jego zwiększenia. Eksperyment rozpoczęto od pakietów o wielkości 2B, sukcesywnie podwajając ich rozmiar. W kolejnych przebiegach startowano od większych wartości, zmieniając krok zwiększania, aż udało się dokładnie określić maksymalny rozmiar obsługiwanej aplikacji UDP.

```
For datagram size of 65202B average time: 0.0003386560920625925s
For datagram size of 65302B average time: 0.00033614323241636156s
For datagram size of 65402B average time: 0.00033453969517722727s
For datagram size of 65502B average time: 0.0003322647185996175s
Failed for 65602B
[Errno 90] Message too long
Client finished.
Times saved into /app/z36_dgram_times_add100_100.json
aszczyta@bigubu:~/PSI_lab_z36/1.1/client$
```

```
For datagram size of 65503B average time: 0.00042458903044462204s
For datagram size of 65504B average time: 0.0003683888353407383s
For datagram size of 65505B average time: 0.0003732750192284584s
For datagram size of 65506B average time: 0.00030147610232234s
For datagram size of 65507B average time: 0.0003190329298377037s
Failed for 65508B
[Errno 90] Message too long
Client finished.
Times saved into /app/z36_dgram_times_inc_1.json
aszczyta@bigubu:~/PSI_lab_z36/1.1/client$
```

Rysunek 1: Doświadczalne wyznaczanie największego rozmiaru pakietu przesyłanego przez klienta UDP. Wyniki z uruchomienia aplikacji klienckiej.

Wyniki Pomiarów



Rysunek 2: Średni pomiędzy wysłaniem wiadomości a odebraniem odpowiedzi po stronie klienta na 100 uruchomień.

Wnioski i uwagi

- Największy obsługiwany datagram to 65507B.
- Wynik jest zgodny z przewidywaniami, gdyż dla IPv4 największy pakiet jaki można przesyłać bez fragmentacji to $65535 - (20 + 8) = 65507$ (20B - nagłówek IP, 8B - nagłówek UDP)
- Czas odpowiedzi, nawet po uśrednieniu wyników z wielu pomiarów, wykazuje nie-regularności i wahania. Mogą wynikać z systemowego planowania procesów, pracy Dockera lub drobnych opóźnień w obsłudze buforów sieciowych.
- Najkrótsze czasy odpowiedzi uzyskiwano dla średnich rozmiarów pakietów (np. 10 000B), co może wynikać z optymalizacji buforów UDP w systemie operacyjnym.