

PSI - Zadanie 1.1

18.11.2024

Zespół Z39 w składzie:

Adrian Murawski

Kacper Straszak

Michał Brzeziński

Komunikacja UDP

Napisz zestaw dwóch programów – klienta i serwera wysyłające datagramy UDP. Wykonaj ćwiczenie w kolejnych inkrementalnych wariantach (rozszerzając kod z poprzedniej wersji). Klient jak i serwer powinien być napisany zarówno w C jak i Pythonie (4 programy).

Sprawdzić i przetestować działanie „między-platformowe”, tj. klient w C z serwerem Python i vice versa.

Treść zadania

Klient wysyła, a serwer odbiera datagramy o stałym rozmiarze (rzędu kilkuset bajtów). Datagramy powinny posiadać ustaloną formę danych. Przykładowo: pierwsze dwa bajty datagramu mogą zawierać informację o jego długości, a kolejne bajty kolejne litery A-Z powtarzające się wymaganą liczbę razy (ale można przyjąć inne rozwiązanie). Odbiorca powinien weryfikować odebrany datagram i odsyłać odpowiedź o ustalonym formacie. Klient powinien wysyłać kolejne datagramy o przyrastającej wielkości np. 1, 100, 200, 1000, 2000... bajtów. Sprawdzić, jaki był maksymalny rozmiar wysłanego (przyjętego) datagramu. Ustalić z dokładnością do jednego bajta jak duży datagram jest obsługiwany. Wyjaśnić.

Rozwiązanie

Postanowiliśmy stworzyć funkcje w C oraz w Pythonie do obsługi zarówno klienta, jak i serwera. Przyjęliśmy koncepcję zaproponowaną w poleceniu, czyli dwa pierwsze bajty datagramu to informacja o długości przesyłanej wiadomości, a pozostałe bajty to przesyłane dane (wiadomość) - one też są zgodne z proponowanymi, czyli są to kolejne wielkie litery od A do Z (w razie potrzeby powtarzając sekwencję).

Warto zaznaczyć, iż przyjęliśmy, że zaproponowana tablica `sizes=[1, 100, 200, 1000, 2000, ...]` to długość „treści” wiadomości, czyli kolejnych liter A-Z, a dwa pierwsze bajty oznaczające długość wiadomości to długość tychże liter A-Z (nie uwzględniając tutaj dodatkowych dwóch bajtów na początku). Stąd też w komunikatach od serwera pojawiają się dwie informacje - o długości wysyłanej treści w bajtach oraz o długości całego datagramu (czyli łącznie z dwoma bajtami na początku).

Serwer C

Start `server()` inicjalizuje zmienne przechowujące informacje o adresach klienta i serwera, oraz danych wejściowych, oraz gniazdo sieciowe.

- Następnie łączymy gniazdo z podanym adresem serwera i odpalamy pętlę nasłuchującą na sygnały wejściowe, gdzie w przypadku otrzymania, sprawdzamy stan danych i wysyłamy odpowiednią informację zwrotną do klienta

Klient C

Program został podzielony na trzy funkcje:

- `resolve_host()`, której celem jest “przetłumaczenie” nazwy hosta na adres IP

(Powstał po odkryciu problemów z połączeniem z serwerem na podstawie wyłącznie nazwy serwera)

- `start_client()`, wysyłająca sygnały o podanej długości na adres hosta danym portem, zatrzymując się na czas otrzymania odpowiedzi zwrotnej lub skończenia się czasu
- `main()` czytająca argumenty startowe w formacie `[host, port]` i wykorzystująca je do uruchomienia programu

Serwer Python

- `start server()` tworzy gniazdo sieciowe i łączy je z podanym adresem sieciowym
- Następnie tworzy pętlę czekającą na i testującą sygnały wchodzące i odpowiednio powiadamia klienta o rezultacie

Klient Python

- Tworzymy gniazdo sieciowe, następnie wysyłamy sygnały na serwer o wcześniej określonej długości, czekając pewien czas na odpowiedź serwera po każdym sygnale

Kod do każdego serwera / klienta jest dostępny w niniejszym repozytorium.

Każdy klient i każdy serwer wyświetla w konsoli informację o przychodzących / wychodzących datagramach.

Prezentacja działania

Testowanie przeprowadziliśmy zarówno lokalnie na własnych komputerach, jak i na serwerze BigUbu (wykorzystując stworzoną dla naszego zespołu sieć).

Poniżej prezentujemy zrzuty ekranu, które pokazują działanie programów (pochodzą one z serwera BigUbu).

serwer pythonowy, klient pythonowy:

```
mbrzezi3@bigubu:~/final/psi-2024z-main/zadanie1.1/python/server$ cat run.sh
docker build -t pserver .
docker run -it --network-alias pserver --hostname pserver --network z39_network --name pserver pserver 8001
mbrzezi3@bigubu:~/final/psi-2024z-main/zadanie1.1/python/server$ docker build -t pserver .
[+] Building 1.7s (7/7) FINISHED                                docker:default
=> [internal] load build definition from dockerfile              0.3s
=> => transferring dockerfile: 113B                             0.0s
=> [internal] load metadata for docker.io/library/python:3      0.5s
=> [internal] load .dockerignore                                0.2s
=> => transferring context: 2B                                   0.0s
=> [internal] load build context                                0.1s
=> => transferring context: 35B                                  0.0s
=> [1/2] FROM docker.io/library/python:3@sha256:bc78d3c007f86dbb87d711b8b082d9d564b8025487e7 0.0s
=> => resolve docker.io/library/python:3@sha256:bc78d3c007f86dbb87d711b8b082d9d564b8025487e7 0.0s
=> CACHED [2/2] ADD udp_server.py /                             0.0s
=> exporting to image                                           0.1s
=> => exporting layers                                          0.0s
=> => writing image sha256:860c7e4f02956c0043085719adaac45d8120c3f24f85c2f50f49b1e644551e82 0.0s
=> => naming to docker.io/library/pserver                       0.0s
mbrzezi3@bigubu:~/final/psi-2024z-main/zadanie1.1/python/server$ docker run -it --network-alias pserver --hostname pserver --network z39_network --name pserver pserver 8001
Serwer nastuchuje na pserver:8001
Odebrano poprawny datagram od ('172.21.39.3', 34033) o rozmiarze wiadomości 1B (cały datagram 3B)
Odebrano poprawny datagram od ('172.21.39.3', 34033) o rozmiarze wiadomości 2B (cały datagram 4B)
Odebrano poprawny datagram od ('172.21.39.3', 34033) o rozmiarze wiadomości 100B (cały datagram 102B)
Odebrano poprawny datagram od ('172.21.39.3', 34033) o rozmiarze wiadomości 200B (cały datagram 202B)
Odebrano poprawny datagram od ('172.21.39.3', 34033) o rozmiarze wiadomości 1000B (cały datagram 1002B)
Odebrano poprawny datagram od ('172.21.39.3', 34033) o rozmiarze wiadomości 2000B (cały datagram 2002B)
```

```
mbrzezi3@bigubu:~/final/psi-2024z-main/zadanie1.1/python/client$ docker build -t pclient .
[+] Building 1.6s (7/7) FINISHED                                docker:default
=> [internal] load build definition from dockerfile              0.1s
=> => transferring dockerfile: 113B                             0.0s
=> [internal] load metadata for docker.io/library/python:3      0.8s
=> [internal] load .dockerignore                                0.2s
=> => transferring context: 2B                                   0.0s
=> [internal] load build context                                0.1s
=> => transferring context: 35B                                  0.0s
=> [1/2] FROM docker.io/library/python:3@sha256:bc78d3c007f86dbb87d711b8b082d9d564b8025487e7 0.0s
=> => resolve docker.io/library/python:3@sha256:bc78d3c007f86dbb87d711b8b082d9d564b8025487e7 0.0s
=> CACHED [2/2] ADD udp_client.py /                             0.0s
=> exporting to image                                           0.0s
=> => exporting layers                                          0.0s
=> => writing image sha256:1eb430951e03378e36a9b2947ca6bd7927ac89ecb86e6ae624cf635ab002b363 0.0s
=> => naming to docker.io/library/pclient                       0.0s
mbrzezi3@bigubu:~/final/psi-2024z-main/zadanie1.1/python/client$ docker run -it --network z39_network --name pclient pclient pserver 8001
Wysyłanie wiadomości - rozmiar wiadomości 1B (całego datagramu 3B)...
Otrzymano potwierdzenie dla wiadomości o rozmiarze 1B (całego datagramu 3B)

Wysyłanie wiadomości - rozmiar wiadomości 2B (całego datagramu 4B)...
Otrzymano potwierdzenie dla wiadomości o rozmiarze 2B (całego datagramu 4B)

Wysyłanie wiadomości - rozmiar wiadomości 100B (całego datagramu 102B)...
Otrzymano potwierdzenie dla wiadomości o rozmiarze 100B (całego datagramu 102B)

Wysyłanie wiadomości - rozmiar wiadomości 200B (całego datagramu 202B)...
Otrzymano potwierdzenie dla wiadomości o rozmiarze 200B (całego datagramu 202B)

Wysyłanie wiadomości - rozmiar wiadomości 1000B (całego datagramu 1002B)...
Otrzymano potwierdzenie dla wiadomości o rozmiarze 1000B (całego datagramu 1002B)

Wysyłanie wiadomości - rozmiar wiadomości 2000B (całego datagramu 2002B)...
Otrzymano potwierdzenie dla wiadomości o rozmiarze 2000B (całego datagramu 2002B)

mbrzezi3@bigubu:~/final/psi-2024z-main/zadanie1.1/python/client$ █
```

serwer c, klient c:

```
mbrzezi3@bigubu:~/final/psi-2024z-main/zadanie1.1/c/client$ docker build -t cclient .
[+] Building 1.2s (9/9) FINISHED                                docker:default
=> [internal] load build definition from dockerfile             0.1s
=> => transferring dockerfile: 125B                             0.0s
=> [internal] load metadata for docker.io/library/gcc:latest   0.4s
=> [internal] load .dockerignore                               0.2s
=> => transferring context: 2B                                    0.0s
=> [1/4] FROM docker.io/library/gcc:latest@sha256:561e2a4b76c441215d26625fbf2d5e345da60ef22b7d46e6 0.0s
=> [internal] load build context                                0.1s
=> => transferring context: 348B                                  0.0s
=> CACHED [2/4] WORKDIR /Sockets                               0.0s
=> CACHED [3/4] COPY udp_client.c .                             0.0s
=> CACHED [4/4] RUN gcc -o udp_client udp_client.c             0.0s
=> exporting to image                                           0.1s
=> => exporting layers                                           0.0s
=> => writing image sha256:01a86c41b61f35840f6e48e173aebbc813073463b51f64fd479d219eb9b488 0.0s
=> => naming to docker.io/library/cclient                       0.0s
mbrzezi3@bigubu:~/final/psi-2024z-main/zadanie1.1/c/client$ docker run -it --network z39_network --name cclient
lient cclient ./udp_client cserver 8001
Wysyłanie wiadomości - długość zawartości 1 bajtów...
Otrzymano potwierdzenie dla datagramu o długości 1 bajtów
Wysyłanie wiadomości - długość zawartości 2 bajtów...
Otrzymano potwierdzenie dla datagramu o długości 2 bajtów
Wysyłanie wiadomości - długość zawartości 100 bajtów...
Otrzymano potwierdzenie dla datagramu o długości 100 bajtów
Wysyłanie wiadomości - długość zawartości 200 bajtów...
Otrzymano potwierdzenie dla datagramu o długości 200 bajtów
Wysyłanie wiadomości - długość zawartości 500 bajtów...
Otrzymano potwierdzenie dla datagramu o długości 500 bajtów
Wysyłanie wiadomości - długość zawartości 1000 bajtów...
Otrzymano potwierdzenie dla datagramu o długości 1000 bajtów
Wysyłanie wiadomości - długość zawartości 2000 bajtów...
Otrzymano potwierdzenie dla datagramu o długości 2000 bajtów
Wysyłanie wiadomości - długość zawartości 4000 bajtów...
Otrzymano potwierdzenie dla datagramu o długości 4000 bajtów
mbrzezi3@bigubu:~/final/psi-2024z-main/zadanie1.1/c/client$
```

```
mbrzezi3@bigubu:~/final/psi-2024z-main/zadanie1.1/c/server$ docker build -t cserver .
[+] Building 3.5s (9/9) FINISHED                                docker:default
=> [internal] load build definition from dockerfile             0.1s
=> => transferring dockerfile: 124B                             0.0s
=> [internal] load metadata for docker.io/library/gcc:latest   0.5s
=> [internal] load .dockerignore                               0.1s
=> => transferring context: 2B                                    0.0s
=> [1/4] FROM docker.io/library/gcc:latest@sha256:561e2a4b76c441215d26625fbf2d5e345da60ef22b7d46e6 0.0s
=> [internal] load build context                                0.1s
=> => transferring context: 2.93kB                                0.0s
=> CACHED [2/4] WORKDIR /Server                                 0.0s
=> [3/4] COPY udp_server.c .                                    0.7s
=> [4/4] RUN gcc -o udp_server udp_server.c                    1.4s
=> exporting to image                                           0.2s
=> => exporting layers                                           0.2s
=> => writing image sha256:792a58abf0cc2b48d373bf8343409cc27ce83e75329c67981c77e38c97ae2ed3 0.0s
=> => naming to docker.io/library/cserver                       0.0s
mbrzezi3@bigubu:~/final/psi-2024z-main/zadanie1.1/c/server$ docker run -it --network-alias cserver --hostname cserver --network z39_network --name cserver cserver ./udp_server cserver 8001
Serwer nasłuchuje na cserver:8001
Odebrano poprawny datagram od 172.21.39.3:46122 o rozmiarze 1 bajtów
Odebrano poprawny datagram od 172.21.39.3:46122 o rozmiarze 2 bajtów
Odebrano poprawny datagram od 172.21.39.3:46122 o rozmiarze 100 bajtów
Odebrano poprawny datagram od 172.21.39.3:46122 o rozmiarze 200 bajtów
Odebrano poprawny datagram od 172.21.39.3:46122 o rozmiarze 500 bajtów
Odebrano poprawny datagram od 172.21.39.3:46122 o rozmiarze 1000 bajtów
Odebrano poprawny datagram od 172.21.39.3:46122 o rozmiarze 2000 bajtów
Odebrano poprawny datagram od 172.21.39.3:46122 o rozmiarze 4000 bajtów
mbrzezi3@bigubu:~/final/psi-2024z-main/zadanie1.1/c/server$
```

serwer c, klient pythonowy:

```
mbrzezi3@bigubu:~/final/psi-2024z-main/zadanie1.1/python/client$ docker build -t pclient .
[+] Building 1.3s (7/7) FINISHED                                docker:default
=> [internal] load build definition from dockerfile            0.1s
=> => transferring dockerfile: 113B                             0.0s
=> [internal] load metadata for docker.io/library/python:3    0.4s
=> [internal] load .dockerignore                             0.2s
=> => transferring context: 2B                                   0.0s
=> [1/2] FROM docker.io/library/python:3@sha256:bc78d3c007f86dbb87d711b8b082d9d564b8025487e780d24c 0.0s
=> => resolve docker.io/library/python:3@sha256:bc78d3c007f86dbb87d711b8b082d9d564b8025487e780d24c 0.0s
=> [internal] load build context                             0.1s
=> => transferring context: 35B                                   0.0s
=> CACHED [2/2] ADD udp_client.py /                          0.0s
=> exporting to image                                         0.0s
=> => exporting layers                                           0.0s
=> => writing image sha256:1eb430951e03378e36a9b2947ca6bd7927ac89ecb86e6ae624cf635ab002b363 0.0s
=> => naming to docker.io/library/pclient                     0.0s
mbrzezi3@bigubu:~/final/psi-2024z-main/zadanie1.1/python/client$ docker run -it --network z39_network --name pclient pclient cserver 8001
Wysyłanie wiadomości - rozmiar wiadomości 1B (całego datagramu 3B)...
Otrzymano potwierdzenie dla wiadomości o rozmiarze 1B (całego datagramu 3B)

Wysyłanie wiadomości - rozmiar wiadomości 2B (całego datagramu 4B)...
Otrzymano potwierdzenie dla wiadomości o rozmiarze 2B (całego datagramu 4B)

Wysyłanie wiadomości - rozmiar wiadomości 100B (całego datagramu 102B)...
Otrzymano potwierdzenie dla wiadomości o rozmiarze 100B (całego datagramu 102B)

Wysyłanie wiadomości - rozmiar wiadomości 200B (całego datagramu 202B)...
Otrzymano potwierdzenie dla wiadomości o rozmiarze 200B (całego datagramu 202B)

Wysyłanie wiadomości - rozmiar wiadomości 1000B (całego datagramu 1002B)...
Otrzymano potwierdzenie dla wiadomości o rozmiarze 1000B (całego datagramu 1002B)

Wysyłanie wiadomości - rozmiar wiadomości 2000B (całego datagramu 2002B)...
Otrzymano potwierdzenie dla wiadomości o rozmiarze 2000B (całego datagramu 2002B)

mbrzezi3@bigubu:~/final/psi-2024z-main/zadanie1.1/python/client$
```

```
mbrzezi3@bigubu:~/final/psi-2024z-main/zadanie1.1/c/server$ docker build -t cserver .
[+] Building 1.1s (9/9) FINISHED                                docker:default
=> [internal] load build definition from dockerfile            0.2s
=> => transferring dockerfile: 124B                             0.0s
=> [internal] load metadata for docker.io/library/gcc:latest   0.4s
=> [internal] load .dockerignore                             0.1s
=> => transferring context: 2B                                   0.0s
=> [1/4] FROM docker.io/library/gcc:latest@sha256:561e2a4b76c441215d26625fbf2d5e345da60ef22b7d46e6 0.0s
=> [internal] load build context                             0.1s
=> => transferring context: 34B                                   0.0s
=> CACHED [2/4] WORKDIR /Server                               0.0s
=> CACHED [3/4] COPY udp_server.c .                          0.0s
=> CACHED [4/4] RUN gcc -o udp_server udp_server.c            0.0s
=> exporting to image                                         0.1s
=> => exporting layers                                           0.0s
=> => writing image sha256:792a58abf0cc2b48d373bf8343409cc27ce83e75329c67981c77e38c97ae2ed3 0.0s
=> => naming to docker.io/library/cserver                     0.0s
mbrzezi3@bigubu:~/final/psi-2024z-main/zadanie1.1/c/server$ docker run -it --network-alias cserver --hostname cserver --network z39_network --name cserver cserver ./udp_server cserver 8001
Serwer nasłuchuje na cserver:8001
Odebrano poprawny datagram od 172.21.39.3:58214 o rozmiarze 1 bajtów
Odebrano poprawny datagram od 172.21.39.3:58214 o rozmiarze 2 bajtów
Odebrano poprawny datagram od 172.21.39.3:58214 o rozmiarze 100 bajtów
Odebrano poprawny datagram od 172.21.39.3:58214 o rozmiarze 200 bajtów
Odebrano poprawny datagram od 172.21.39.3:58214 o rozmiarze 1000 bajtów
Odebrano poprawny datagram od 172.21.39.3:58214 o rozmiarze 2000 bajtów
mbrzezi3@bigubu:~/final/psi-2024z-main/zadanie1.1/c/server$
```

serwer pythonowy, klient c:

```
mbrzezi3@bigubu:~/final/psi-2024z-main/zadaniel.1/python/server$ docker build -t pserver .
[+] Building 1.7s (7/7) FINISHED                                docker:default
=> [internal] load build definition from dockerfile             0.2s
=> => transferring dockerfile: 113B                             0.0s
=> [internal] load metadata for docker.io/library/python:3     0.8s
=> [internal] load .dockerignore                               0.1s
=> => transferring context: 2B                                   0.0s
=> [internal] load build context                               0.1s
=> => transferring context: 35B                                  0.0s
=> [1/2] FROM docker.io/library/python:3@sha256:bc78d3c007f86dbb87d711b8b082d9d564b8025487e780d24c 0.0s
=> => resolve docker.io/library/python:3@sha256:bc78d3c007f86dbb87d711b8b082d9d564b8025487e780d24c 0.0s
=> => CACHED [2/2] ADD udp_server.py /                          0.0s
=> => exporting to image                                         0.0s
=> => exporting layers                                           0.0s
=> => writing image sha256:2fce5515ad3eb93fc8db5637236bf8651b1bb164d67a0d91b9ac03ab0e49969c 0.0s
=> => naming to docker.io/library/pserver                      0.0s
mbrzezi3@bigubu:~/final/psi-2024z-main/zadaniel.1/python/server$ docker run -it --network-alias pserver --
hostname pserver --network z39_network --name pserver pserver 8001
Serwer nastuchuje na pserver:8001
Odebrano poprawny datagram od ('172.21.39.3', 46514) o rozmiarze wiadomości 1B (cały datagram 3B)
Odebrano poprawny datagram od ('172.21.39.3', 46514) o rozmiarze wiadomości 2B (cały datagram 4B)
Odebrano poprawny datagram od ('172.21.39.3', 46514) o rozmiarze wiadomości 100B (cały datagram 102B)
Odebrano poprawny datagram od ('172.21.39.3', 46514) o rozmiarze wiadomości 200B (cały datagram 202B)
Odebrano poprawny datagram od ('172.21.39.3', 46514) o rozmiarze wiadomości 500B (cały datagram 502B)
Odebrano poprawny datagram od ('172.21.39.3', 46514) o rozmiarze wiadomości 1000B (cały datagram 1002B)
Odebrano poprawny datagram od ('172.21.39.3', 46514) o rozmiarze wiadomości 2000B (cały datagram 2002B)
Odebrano poprawny datagram od ('172.21.39.3', 46514) o rozmiarze wiadomości 4000B (cały datagram 4002B)
mbrzezi3@bigubu:~/final/psi-2024z-main/zadaniel.1/python/server$
```

```
mbrzezi3@bigubu:~/final/psi-2024z-main/zadaniel.1/c/client$ docker build -t cclient .
[+] Building 1.1s (9/9) FINISHED                                docker:default
=> [internal] load build definition from dockerfile             0.1s
=> => transferring dockerfile: 125B                             0.0s
=> [internal] load metadata for docker.io/library/gcc:latest    0.5s
=> [internal] load .dockerignore                               0.2s
=> => transferring context: 2B                                   0.0s
=> [1/4] FROM docker.io/library/gcc:latest@sha256:561e2a4b76c441215d26625fbf2d5e345da60ef22b7d46e6 0.0s
=> [internal] load build context                               0.1s
=> => transferring context: 34B                                  0.0s
=> CACHED [2/4] WORKDIR /Sockets                               0.0s
=> CACHED [3/4] COPY udp_client.c .                             0.0s
=> CACHED [4/4] RUN gcc -o udp_client udp_client.c              0.0s
=> => exporting to image                                         0.0s
=> => exporting layers                                           0.0s
=> => writing image sha256:01a86c41b61f35840f6e48e173aebbc813073463b51f64fd4d479d219eb9b488 0.0s
=> => naming to docker.io/library/ccclient                     0.0s
mbrzezi3@bigubu:~/final/psi-2024z-main/zadaniel.1/c/client$ docker run -it --network z39_network --name cc
lient cclient ./udp_client pserver 8001
Wysyłanie wiadomości - długość zawartości 1 bajtów...
Otrzymano potwierdzenie dla datagramu o długości 1 bajtów
Wysyłanie wiadomości - długość zawartości 2 bajtów...
Otrzymano potwierdzenie dla datagramu o długości 2 bajtów
Wysyłanie wiadomości - długość zawartości 100 bajtów...
Otrzymano potwierdzenie dla datagramu o długości 100 bajtów
Wysyłanie wiadomości - długość zawartości 200 bajtów...
Otrzymano potwierdzenie dla datagramu o długości 200 bajtów
Wysyłanie wiadomości - długość zawartości 500 bajtów...
Otrzymano potwierdzenie dla datagramu o długości 500 bajtów
Wysyłanie wiadomości - długość zawartości 1000 bajtów...
Otrzymano potwierdzenie dla datagramu o długości 1000 bajtów
Wysyłanie wiadomości - długość zawartości 2000 bajtów...
Otrzymano potwierdzenie dla datagramu o długości 2000 bajtów
Wysyłanie wiadomości - długość zawartości 4000 bajtów...
Otrzymano potwierdzenie dla datagramu o długości 4000 bajtów
mbrzezi3@bigubu:~/final/psi-2024z-main/zadaniel.1/c/client$
```

Widać, że w każdej parze serwer - klient (w obu językach) komunikacja działa poprawnie, tzn. datagramy (zgodne z odpowiednimi parametrami - więcej we wnioskach) są przesyłane od klienta do serwera, a następnie serwer odsyła informację o odbiorze danego datagramu.

Wnioski

Zadanie udało się zrealizować z sukcesem, wszystkie programy działają, serwer i klient komunikują się.

Wnioski wynikające z testowania różnych wartości długości datagramów wyglądają następująco:

Poprawnie zostaną przesłane i odebrane te datagramy, których długość nie przekracza rozmiaru bufora w serwerze (jest on zdefiniowany jako *bufsize* w kodzie serwerów).

Największy możliwy datagram do wysłania (obsługiwany) ma długość 65507 bajtów, większy nie jest możliwy, z powodu *OSError*-a. Oznacza to, że nawet dla większych rozmiarów bufora w serwerze, taki datagram nie będzie mógł zostać przesłany.

```
mbrzezi3@bigubu:~/final/psi-2024z-main/zadanie1.1/python/server$ docker run -it --network-alias pserver --hostname pserver --network z39_network --name pserver pserver 8001
Serwer nasłuchuje na pserver:8001
Odebrano poprawny datagram od ('172.21.39.3', 46270) o rozmiarze wiadomości 1B (cały datagram 3B)
Odebrano poprawny datagram od ('172.21.39.3', 46270) o rozmiarze wiadomości 2B (cały datagram 4B)
Odebrano poprawny datagram od ('172.21.39.3', 46270) o rozmiarze wiadomości 100B (cały datagram 102B)
Odebrano poprawny datagram od ('172.21.39.3', 46270) o rozmiarze wiadomości 200B (cały datagram 202B)
Odebrano poprawny datagram od ('172.21.39.3', 46270) o rozmiarze wiadomości 1000B (cały datagram 1002B)
Odebrano poprawny datagram od ('172.21.39.3', 46270) o rozmiarze wiadomości 2000B (cały datagram 2002B)
Odebrano poprawny datagram od ('172.21.39.3', 46270) o rozmiarze wiadomości 65502B (cały datagram 65504B)
Odebrano poprawny datagram od ('172.21.39.3', 46270) o rozmiarze wiadomości 65503B (cały datagram 65505B)
Odebrano poprawny datagram od ('172.21.39.3', 46270) o rozmiarze wiadomości 65504B (cały datagram 65506B)
Odebrano poprawny datagram od ('172.21.39.3', 46270) o rozmiarze wiadomości 65505B (cały datagram 65507B)
```

```
Wysyłanie wiadomości - rozmiar wiadomości 65502B (całego datagramu 65504B)...
Otrzymano potwierdzenie dla wiadomości o rozmiarze 65502B (całego datagramu 65504B)

Wysyłanie wiadomości - rozmiar wiadomości 65503B (całego datagramu 65505B)...
Otrzymano potwierdzenie dla wiadomości o rozmiarze 65503B (całego datagramu 65505B)

Wysyłanie wiadomości - rozmiar wiadomości 65504B (całego datagramu 65506B)...
Otrzymano potwierdzenie dla wiadomości o rozmiarze 65504B (całego datagramu 65506B)

Wysyłanie wiadomości - rozmiar wiadomości 65505B (całego datagramu 65507B)...
Otrzymano potwierdzenie dla wiadomości o rozmiarze 65505B (całego datagramu 65507B)

Wysyłanie wiadomości - rozmiar wiadomości 65506B (całego datagramu 65508B)...
Traceback (most recent call last):
  File "///.udp_client.py", line 35, in <module>
    start_client(sys.argv[1], int(sys.argv[2]))
    ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
  File "///.udp_client.py", line 18, in start_client
    client_socket.sendto(message, (host, port))
    ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
OSError: [Errno 90] Message too long
mbrzezi3@bigubu:~/final/psi-2024z-main/zadanie1.1/python/client$
```