Technologie sieciowe

Sprawozdanie

Lista 4 TCP/IP

Autor: Grzegorz Bara 236542

1 Cele Listy

Celem listy była modyfikacja podanych przez prowadzącego programów w taki sposób aby symulowały one przesyłanie danych przy pomocy protokołu TCP/IP

2 Wstęp teoretyczny

**TCP** - połączeniowy, niezawodny, strumieniowy [protokół komunikacyjny](https://pl.wikipedia.org/wiki/Protok%C3%B3%C5%82_komunikacyjny) stosowany do przesyłania danych między [procesami](https://pl.wikipedia.org/wiki/Proces_(informatyka)) uruchomionymi na różnych maszynach, będący częścią szeroko wykorzystywanego obecnie [stosu TCP/IP](https://pl.wikipedia.org/wiki/Model_TCP/IP).

**TCP/IP** - to zbiór protokołów służących do transmisji danych przez sieci komputerowe. Model TCP/IP implementuje najważniejsze funkcjonalności siedmiu warstw standardowego [modelu OSI](http://www.crypto-it.net/pl/teoria/protokoly-tcp-ip.html).

**Datagram** - podstawowa jednostka przekazu powiązana z siecią komutacyjną pakietów. Datagramy zwykle są zbudowane z sekcji nagłówka i ładunku.Datagramy dostarczają możliwość bezpołączeniowej komunikacji w sieci komutacyjnej pakietów. Dostarczenie, czas dostarczenia i kolejność datagramów nie musi być gwarantowana przez sieć.

3 Opis klas

**Z2Sender** – Program symulujący działanie hosta wysyłającego dane.

**Z2Packet** – Klasa imitująca pakiet w datagramie.

**Z2Forwarder** – Program symulujący medium transmisyjne i wywołujący różne sytuacje mogące nastąpić przy transmisji danych (zmiana kolejności, utrata lub duplikacja pakietów).

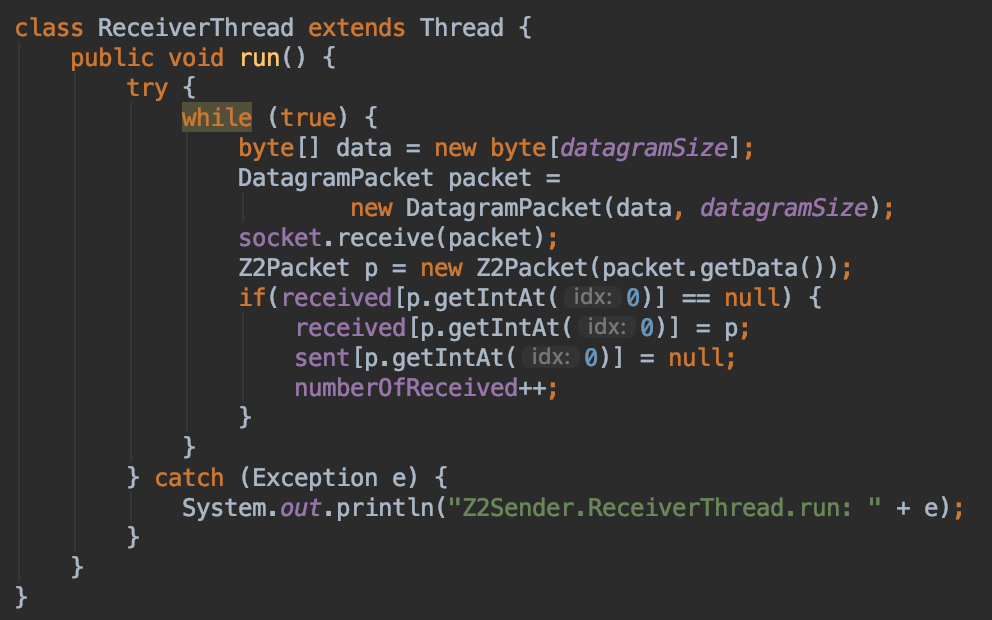
**Z2Receiver** – Program symulujący działanie hosta odbierającego dane.

4 Realizacja

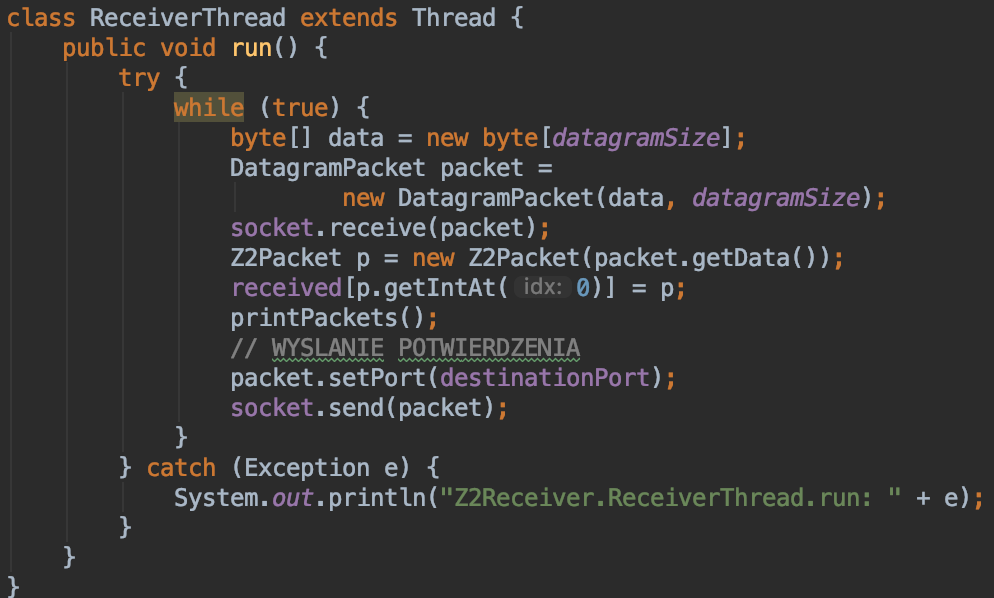
Aktualna struktura klas nie daje nam pewności w odpowiednim przesyłaniu pakietów. Należy zmodyfikować klasę Sender oraz Receiver tak, aby pierwsza klasa upewniła się, że wysyłane dane zostaną odebrane natomiast druga z klas ma za zadanie upewnić się, że dane zostały wysłane w dobrej kolejności oraz są kompletne.



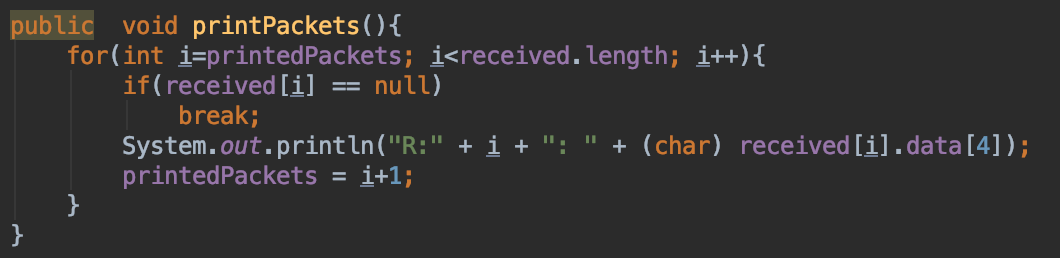
Wątek SenderThread wewnątrz klasy Z2Sender wysyłający pakiety danych do podanego socketa. Po modyfikacji wszystkie wysłane pakiety są zapisywane do tablicy sent[] dzięki czemu mogą zostać w przyszłości wysłane ponownie, jeżeli nie otrzymamy potwierdzenia od drugiego hosta, że otrzymał nasze pakiety. Pętla while() zajmuje się ponownym wysyłaniem pakietów dla których nie otrzymaliśmy potwierdzenia odebrania.



Wątek ReceiverThread wewnątrz klasy Z2Sender odpowiadający za odbieranie potwierdzeń dostarczenia pakietów. W przypadku odebrania potwierdzenia o dostarczeniu danego pakietu, jest on zapisywany do tablicy received[] oraz usuwany z tablicy send[].



Wątek ReceiverThread wewnątrz klasy Z2Receiver odpowiadający za odbiór i drukowanie danych oraz wysyłanie potwierdzeń. Odebrane pakiety zapisywane są w tabeli received[] co umożliwia drukowanie ich w poprawnej kolejności oraz bez duplikatów po wywołaniu metody printPackets().



Metoda printPackets() wewnątrz klasy Z2Receiver odpowiadająca za drukowanie pakietów w odpowiedniej kolejności i bez duplikatów. Pakiety drukowane są tylko wtedy, gdy nie były jeszcze drukowane oraz wszystkie wcześniejsze pakiety zostały odebrane i wydrukowane.

Po wykonaniu powyższych modyfikacji pakiety są odpowiednio odbierane oraz wysyłane dzięki czemu program może zostać poprawnie wywołany.

5 Wnioski

Pomimo kilkukrotnego przesłania tych samych danych, mamy pewność, że dane będą drukowane w tej samej kolejności nie będą się dublować. Protokół TCP/IP pomaga w transmisji danych, ponieważ sprawdza on czy dotarły wszystkie pakiety, a medium transmisyjne nie musi być stałe, a to znaczy, że mogą występować w nim zakłócenia powodujące gubienie pakietów podczas transmisji danych.