

POLITECHNIKA WROCŁAWSKA

ANNA MODRZEJEWSKA

Sprawozdanie nr 4

Technologie Sieciowe

15 czerwca 2018

1 Cel

Zmodyfikowanie programu do wysyłania i odbierania danych tak, aby wszystkie dane zostały dostarczone do odbiorcy w kolejności ich wysyłania.

2 Wyjaśnienie

2.1 TCP

Programy po modyfikacji mają symulować protokół TCP. Pakiety po wysłaniu są śledzone, czy dane nie zostały zgubione lub uszkodzone. Osiąga się to poprzez wysyłanie potwierdzenia odbioru od odbiorcy do nadawcy. W przypadku gdy odbiorca otrzyma niepoprawne dane, nadawca retransmituje je. TCP jest używany między innymi na stronach internetowych lub chatach, gdzie największe znaczenie ma otrzymanie wszystkich danych w kolejności ich wysyłania.

2.2 UDP

Istnieją sytuacje, w których bardziej uzasadnione będzie zastosowanie protokołu UDP. W odróżnieniu od TCP, pakiety są wysyłane do odbiorcy z pominięciem kontroli przepływu. UDP jest używany w przypadku, gdy prędkość przepływu danych jest bardziej pożądana niż ich stuprocentowa poprawność. Wykorzystuje się go między innymi w transmisjach na żywo, gdzie można porzucić oczekiwanie na dane, które nie dotarły na czas, aby szybciej otrzymać aktualne dane.

3 Realizacja

Uwaga. Kody źródłowe i opis działania programów do modyfikacji są podane na stronie dra Krzywieckiego: <http://cs.pwr.edu.pl/krzywiecki/teaching.html>

Program pracuje w trzech trybach:

1. **Sender:** `java Z2Sender <port1> <port2>`
2. **Receiver:** `java Z2Receiver <port1> <port2>`
3. **Forwarder:** `java Z2Forwarder <port1> <port2>`

Przykładowe uruchomienie:

```
java Z2Receiver 6002 6003 & java Z2Forwarder 6001 6002 & java  
Z2Forwarder 6003 6000 & java Z2Sender 6000 6001
```

Sender wysyła pakiety do portu 2 i oczekuje potwierdzenia na porcie 1.

Receiver odbiera pakiety z portu 1, drukuje je w odpowiedniej kolejności i wysyła potwierdzenia do portu 2.

Forwarder przekazuje pakiety z portu 1 do portu 2 (z losową kolejnością, z szansą na zgubienie lub zdublowanie pakietu).

Do programu został dodany bufor przy klasie Sender i Receiver oraz wątek retransmitujący.

Sender będzie wysyłał pakiet dopóki nie otrzyma potwierdzenia.

Sender składa się z następujących wątków:

- **SenderThread** wysyła dane podane w standardowym wejściu.

- **ReceiverThread** odbiera nadesłane potwierdzenia odbioru.
- **RetransmitThread** wysyła ponownie dane, które nie mają potwierdzenia odbioru.

Forwarder składa się z następujących wątków:

- **Receiver** odbiera dane i wrzuca je do bufora.
- **Sender** wysyła dane z bufora.

Receiver składa się z następujących wątków:

- **ReceiverThread** odbiera dane i wysyła potwierdzenia odbioru.
- **PrintThread** wypisuje po kolei odebrane dane.

4 Wnioski

- W Internecie pakiety wysyłane przez nadawcę mogą być tracone, przybywać z różnymi opóźnieniami, w zmienionej kolejności czy zduplikowane.
- Poprawna implementacja retransmisji i potwierdzenia odbioru zapewnia bezstratną komunikację mimo niepewnego łącza.
- W przypadku aplikacji działających w rzeczywistym czasie (chaty video, gry online) stosuje się protokół UDP, który pomija oczekiwanie na wszystkie pakiety, dzięki czemu działa sprawniej niż TCP.
- TCP jest przydatny wtedy, gdy największe znaczenie ma kompletność pakietów.