

Zadanie 3: Protokół do przesyłania plików z kontrolą przepustowości SKJ (2017)

Wstęp

Przesyłanie plików przez sieć jest jedną z częściej wykonywanych czynności. Przesyłanie plików musi być procesem niezawodnym, przesyłającym dane bez przekłamań. Potoki transferu zwykle dążą do maksymalizacji przepustowości, jednak nie jest to jedyny możliwy sposób postępowania. Celem zadania jest opracowanie protokołu transmisji danych wykorzystującego UDP. Protokół ma zapewniać niezawodne przesyłanie danych oraz zawierać mechanizmy kontroli prędkości przepływu danych.

Specyfikacja

Głównym celem zadania jest opracowanie protokołu przesyłu danych.

1. Jako protokół transmisyjny warstwy 4-tej musi być wykorzystany protokół UDP. Oznacza to, że do wysyłania/odbierania danych można wykorzystywać **wyłącznie** z klas `DatagramSocket` i `DatagramPacket` oraz ich metod.
2. Protokół musi być niezawodny (czego nie zapewnia UDP), w szczególności musi umieć przeprowadzić kontrolę spójności przesyłanych danych (w tym weryfikacja treści), retransmisję danych w przypadku zagubienia pakietów i zachowywanie ich kolejności zgodnie z kolejnością danych wysłanych.
3. Protokół pozwala na określenie średniej prędkości wysyłanych danych (dotyczy wyłącznie przesyłanego pliku, bez uwzględnienia danych dodatkowych wymaganych do pracy protokołu), która to prędkość musi być przestrzegana przez proces wysyłający dane. Średnia prędkość musi być zachowana w interwale długości 1s (w szczególności nie jest dozwolone przesłanie całości pliku a następnie odczekanie czasu do uzyskania żądanej średniej, jeśli plik jest większy niż rozmiar danych przewidziany do wysłania w czasie 1s). Parametr określający prędkość wysyłania danych może zostać zmieniony w trakcie transmisji i proces wysyłający musi się do tej zmiany dynamicznie dopasować.

4. Obsługuje weryfikację przesłanych danych poprzez porównanie sumy kontrolnej MD5 wyliczonej na oryginalnym i odebranych pliku (na całości, aczkolwiek obliczenie może być wykonywane na bieżąco). Do wyznaczania sumy MD5 można użyć bibliotecznych klas Javy.

Zaprojektowany protokół należy zaimplementować w postaci pary aplikacji: **klienta** i **serwera**.

Klient

Klient jest procesem służącym do wysyłania danych. Podczas uruchomienia klient przyjmuje następujące parametry uruchomienia (kolejność parametrów w linii poleceń może być dowolna):

- **-server <address>** – określa adres, na którym działa aplikacja serwera.
- **-port <port number>** – określa numer portu UDP na którym pracuje aplikacja serwera.
- **-file <file name>** – określa nazwę pliku, który ma zostać przesłany.

Po uruchomieniu, klient nawiązuje połączenie z serwerem. Po nawiązaniu połączenia przesyła do serwera nazwę pliku (plus opcjonalne dane dodatkowe – wedle projektu), odbiera od serwera wartość początkowej prędkości przesyłania danych i rozpoczyna przesyłanie pliku określonego jako parametr. Podczas transmisji musi reagować na odpowiedzi serwera – żądanie zmiany prędkości przesyłu danych, informacje o skutecznym/nieskutecznym przesłaniu danych itp. Po skutecznym przesłaniu pliku i weryfikacji jego treści klient wysyła do serwera komunikat o poprawnym/błędnym transferze i kończy pracę.

Podczas przesyłania danych klient wyświetla na bieżąco informacje dotyczące transmisji:

- liczbę skutecznie przesłanych bajtów i liczbę bajtów pozostałych do przesłania,
- aktualną średnią prędkość przesyłania danych liczoną od początku wysyłania pliku,
- aktualną średnią prędkość przesyłania danych liczoną w ciągu ostatnich 10s,
- aktualną średnią prędkość przesyłania danych liczoną w ciągu ostatniej 1s lub o problemach transmisji, jeśli wystąpią.

Po zakończeniu transmisji powinien wyświetlić podsumowanie transmisji: liczbę przesłanych bajtów, czas transmisji, średnią prędkość przesyłu danych, zweryfikowaną sumę kontrolną MD5.

Serwer

Zadaniem serwera jest przyjęcie połączenia klienta oraz odebranie przesyłanego do niego pliku. Podczas uruchomienia serwer przyjmuje następujące parametry (kolejność parametrów w linii poleceń może być dowolna):

- `-port <port number>` – określa numer portu UDP na którym pracuje aplikacja serwera.
- `-speed <initial speed>` – określa wartość początkowej prędkości przesyłania danych wyrażoną w KB/s (1KB=1024B).

Po uruchomieniu serwer oczekuje na połączenie od klienta. Po nawiązaniu połączenia serwer otrzymuje od klienta nazwę pliku docelowego (plus opcjonalne dane dodatkowe – wedle projektu protokołu), wysyła do klienta wartość początkowej prędkości przesyłania danych i rozpoczyna odbieranie zawartości pliku, zgodnie z zaprojektowanym protokołem. Zadaniem serwera jest kontrola poprawności dostarczonych danych oraz reakcja na ewentualne błędy (brak dostarczonych danych, błędy transmisji, itp.). Po zakończeniu transferu serwer przesyła do klienta wyliczoną na podstawie zawartości pliku sumę kontrolną MD5, a po odebraniu potwierdzenia o poprawnym/błędym transferze, wypisuje odpowiedni komunikat i kończy pracę. Odbierany plik należy zapisywać na bieżąco na dysku w bieżącym katalogu (jeśli plik o odpowiedniej nazwie istnieje, jego zawartość jest usuwana i nadpisywana).

Podczas pracy serwer może wczytać z klawiatury nową wartość prędkości przesyłu danych (w KB/s). Jeśli taka wartość zostanie wczytana, serwer przekazuje ją do klienta, który ma obowiązek przystosować się do nowej wartości prędkości. Tego typu zmiany mogą nastąpić w dowolnej chwili i wielokrotnie w trakcie przesyłania pliku.

Wymagania i sposób oceny

1. Poprawny i pełny projekt wart jest **7 punktów**. Za zrealizowanie każdej z poniższych funkcjonalności można otrzymać punkty do podanej wartości.

- (a) Projekt poprawnego protokołu zapewniającego funkcjonalność opisaną w treści zadania: 5 punktów.

Opis projektu musi zawierać informacje o:

- sposobie nawiązywania połączenia: jakie komunikaty są wysyłane, co przychodzi w odpowiedzi (nie jest wymagana autoryzacja a jedynie wymiana danych niezbędnych do transmisji) (1p.),
- sposobie transmisji danych: jakie dane dodatkowe są przesyłane, w jaki sposób zapewnia się właściwą kolejność danych po stronie odbiorcy, jaka jest reakcja na niepoprawne dane lub ich niedostarczenie (2p.),
- metodzie weryfikacji poprawności przesłanych danych w trakcie i/lub na końcu transmisji (1p.),

- sposobie realizacji kontroli prędkości przesyłu danych: w jaki sposób aplikacja klienta dopasowuje się do wymagań przepustowości narzuconych przez aplikację serwera (1p.).
- (b) Implementacja aplikacji klienta i serwera, zgodnie z powyższym opisem: 2 punkty.
- (c) Dodatkowy punkt (poza skalą) można uzyskać, jeśli zaprojektowany i zaimplementowany protokół będzie uodporniony na zaburzenia transmisji wynikające z przesłania danych na określony port (zarówno do klienta jak i do serwera) z komputera nieuczestniczącego w tej transmisji.
2. Aplikację piszemy w języku Java zgodnie ze standardem Java 8 (JDK 1.8). Do komunikacji przez sieć można wykorzystać jedynie podstawowe klasy do komunikacji z wykorzystaniem protokołu UDP.
3. Projekty powinny zostać zapisane do odpowiednich katalogów w systemie EDUX w nieprzekraczalnym terminie 28 stycznia 2018 (termin może zostać zmieniony przez prowadzącego grupę).
4. Spakowany plik projektu powinien obejmować:
- Plik *Dokumentacja(nr.indeksu)Zad3.pdf*, opisujący protokół, co zostało zrealizowane, co się nie udało, gdzie ewentualnie są błędy, których nie udało się poprawić.
 - Pliki źródłowe (dla JDK 1.8) (włącznie z wszelkimi bibliotekami nie należącymi do standardowej instalacji Javy, których autor użył) - aplikacja musi dać się bez problemu skompilować na komputerach w laboratorium w PJA.

UWAGA: PLIK Z DOKUMENTACJĄ JEST WARUNKIEM KONIECZNYM PRZYJĘCIA PROJEKTU DO OCENY.

5. Prowadzący oceniać będą w pierwszym rzędzie poprawność działania programu i zgodność ze specyfikacją, ale na ocenę wpływać będzie także zgodność wytworzonego oprogramowania z zasadami inżynierii oprogramowania i jakością implementacji.
6. JEŚLI NIE WYSZCZEGÓLNIŁO INACZEJ, WSZYSTKIE NIEJASNOŚCI NALEŻY PRZEDYSKUTOWAĆ Z PROWADZĄCYM ZAJĘCIA POD GROŻBĄ NIEZALICZENIA PROGRAMU W PRZYPADKU ICH NIEWŁAŚCIWEJ INTERPRETACJI.