AKADEMIA GÓRNICZO-HUTNICZA

Dokumentacja do projektu

Biblioteka do emulacji łącza fizycznego

z przedmiotu

Języki Programowania Obiektowego

EiT, 3 rok

Hubert Makara

środa 8.00

prowadzący: mgr inż. Jakub Zimnol

09.01.2026

1. Opis projektu i zastosowania

Celem projektu jest stworzenie biblioteki do symulacji działania rzeczywistego łącza sieciowego. Program pozwala na emulację transmisji pakietów danych oraz dodanie typowych zakłóceń sieciowych, takich jak, opóźnienie, utrata pakietów czy uszkodzenie danych.

1.1 Zastosowania:

-Testowanie odporności protokołów sieciowych na błędy transmisji.

-Symulacja zachowania aplikacji w warunkach słabego łącza

-Edukacja w zakresie działania sieci komputerowych i mechanizmów obsługi błędów

2. Opis zaimplementowanych klas

Packet: Szablonowa klasa reprezentująca pakiet sieciowy. Przechowuje nagłówki (IP, porty), dane (payload) oraz flagi statusu (np. czy pakiet został zgubiony lub zduplikowany)

LinkEmulator: Główna klasa zarządzająca symulacją. Przechowuje kolekcję aktywnych zakłóceń i sekwencyjnie aplikuje je na przesłane pakiety.

Disturbance: Abstrakcyjna klasa bazowa dla wszystkich zakłóceń.

Klasy zakłóceń (dziedziczące po Disturbance):

Delay: Reprezentuje efekt opóźnienia transmisji.

PacketLost: Symuluje utratę pakietu z określonym prawdopodobieństwem.

Tamper: Symuluje uszkodzenie pakietu z określonym prawdopodobieństwem (zmiana danych).

Duplicate: Oznacza pakiet jako zduplikowany, co symuluje podwójne odebranie ramki.

Throttle: Symuluje ograniczenie przepustowości łącza – opóźnienie jest zależne od wielkości przesyłania danych.

2.1 Bezpieczeństwo:

Program zabezpiecza przed:

Wpisaniem w Delay najpierw wartości większej później mniejszej, oraz wpisanie wartości ujemnej.

W klasach z prawdopodobieństwem wpisanie wartości mniejszej niż 0 ustawia wartość na 0 a, większej od 1 ustawia wartość 1.

W projekcie zastosowano std::make_shared w celu zapewnienia bezpieczeństwa pamięci i eliminacji wycieków (Memory Leaks) poprzez automatyczne zarządzanie cyklem

życia obiektów. Rozwiązanie to gwarantuje tzw. Exception Safety (bezpieczeństwo wyjątków), ponieważ alokacja pamięci dla obiektu i jego licznika referencji odbywa się w jednym, niepodzielnym kroku. Dzięki temu aplikacja jest stabilna i odporna na błędy wynikające z ręcznego zarządzania pamięcią.

3. Opis uruchomienia

mkdir build

cd build

cmake ..

cmake –build .

.\Program.exe ← uruchomienie

4. Przykładowe działanie